Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 1

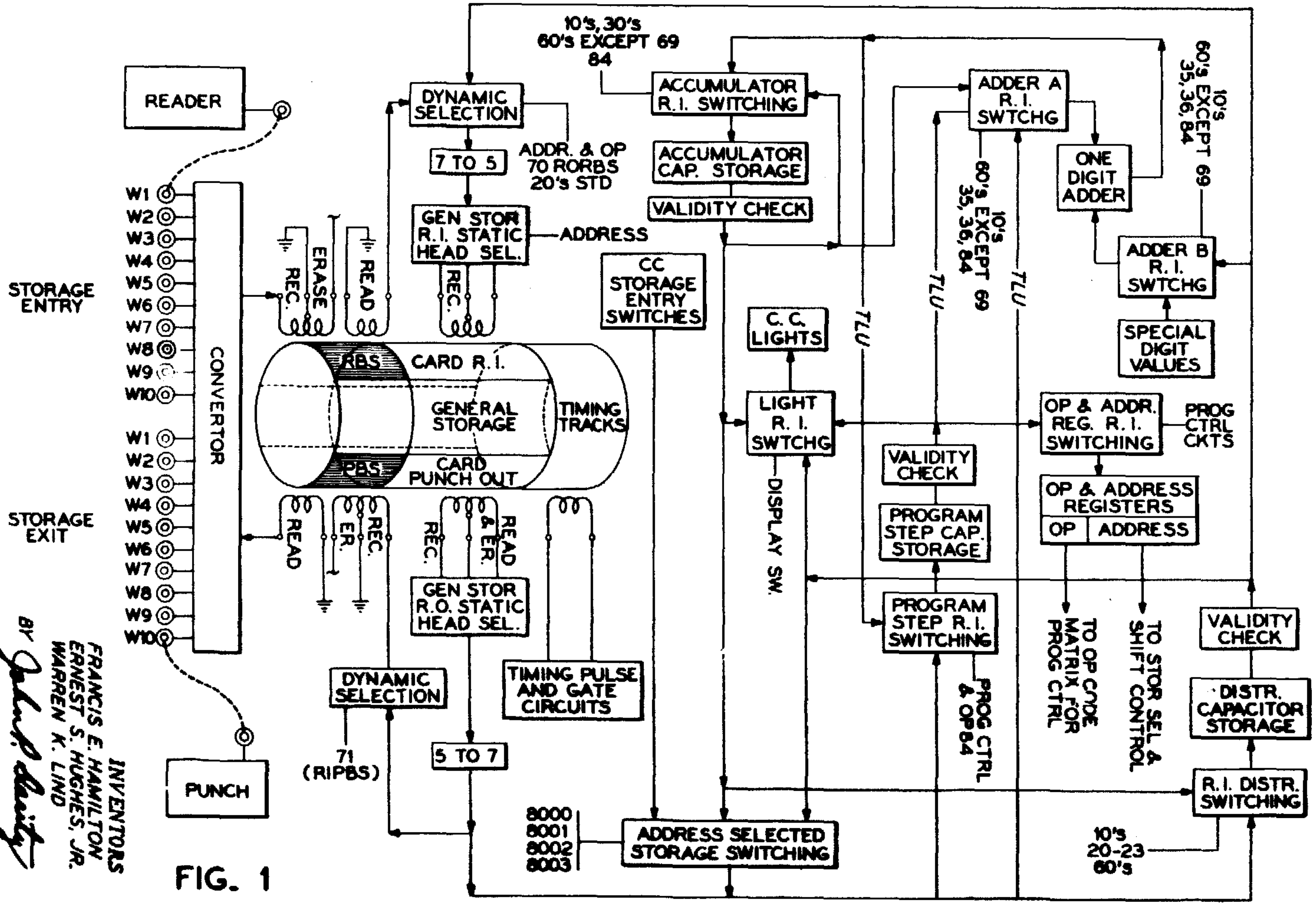


FIG. 1

INVENTORS
FRANCIS E. HAMILTON
ERNEST S. HUGHES, JR.
WARREN K. LIND

BY

ATTORNEY

	BINARY LEVEL		QUINARY LEVEL				
DECIMAL DIGIT VALUE	VALUE ASSIGNED TO EACH BINARY TYPE STORAGE DEVICE						
	B5	B0	Q4	Q3	Q2	Q1	Q0
0	0	1	0	0	0	0	1
1	0	1	0	0	0	1	0
2	0	1	0	0	1	0	0
3	0	1	0	1	0	0	0
4	0	1	1	0	0	0	0
5	1	0	0	0	0	0	1
6	1	0	0	0	0	1	0
7	1	0	0	0	1	0	0
8	1	0	0	1	0	0	0
9	1	0	1	0	0	0	0
	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5	TRACK 6	TRACK 7

FACTOR STORAGE BAND ON DRUM

FIG. 2

DECIMAL DIGIT VALUE	VALUE ASSIGNED TO EACH BINARY DEVICE				
	0	1	2	3	6
0	0	1	1	0	0
1	1	1	0	0	0
2	1	0	1	0	0
3	1	0	0	1	0
4	0	1	0	1	0
5	0	0	1	1	0
6	1	0	0	0	1
7	0	1	0	0	1
8	0	0	1	0	1
9	0	0	0	1	1
	TRACK 1	TRACK 2	TRACK 3	TRACK 4	TRACK 5

1 GENERAL STORAGE BAND ON DRUM

FIG. 3

DRUM TIMING DIVISIONS

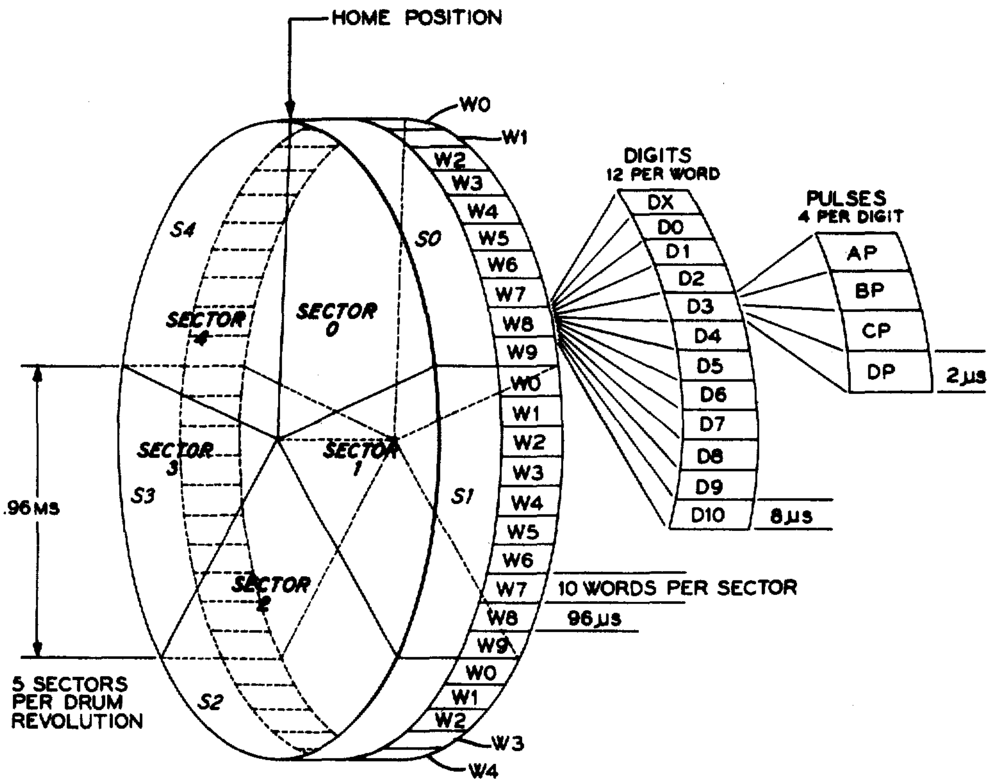


FIG. 4

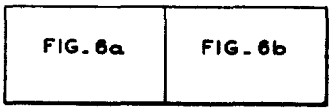


FIG. 10

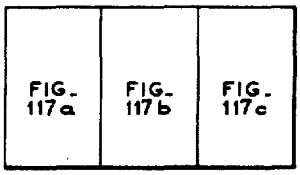


FIG. 11

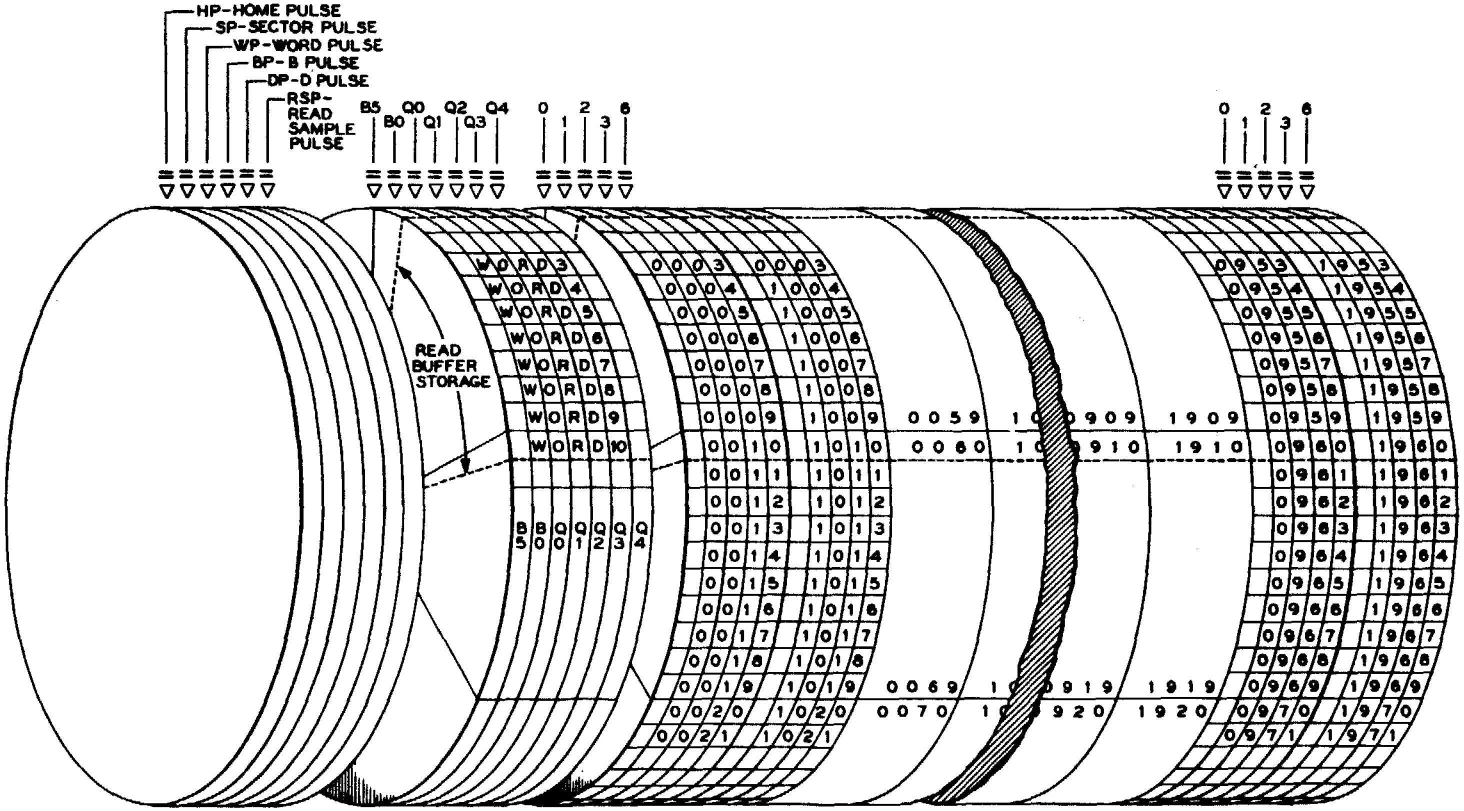
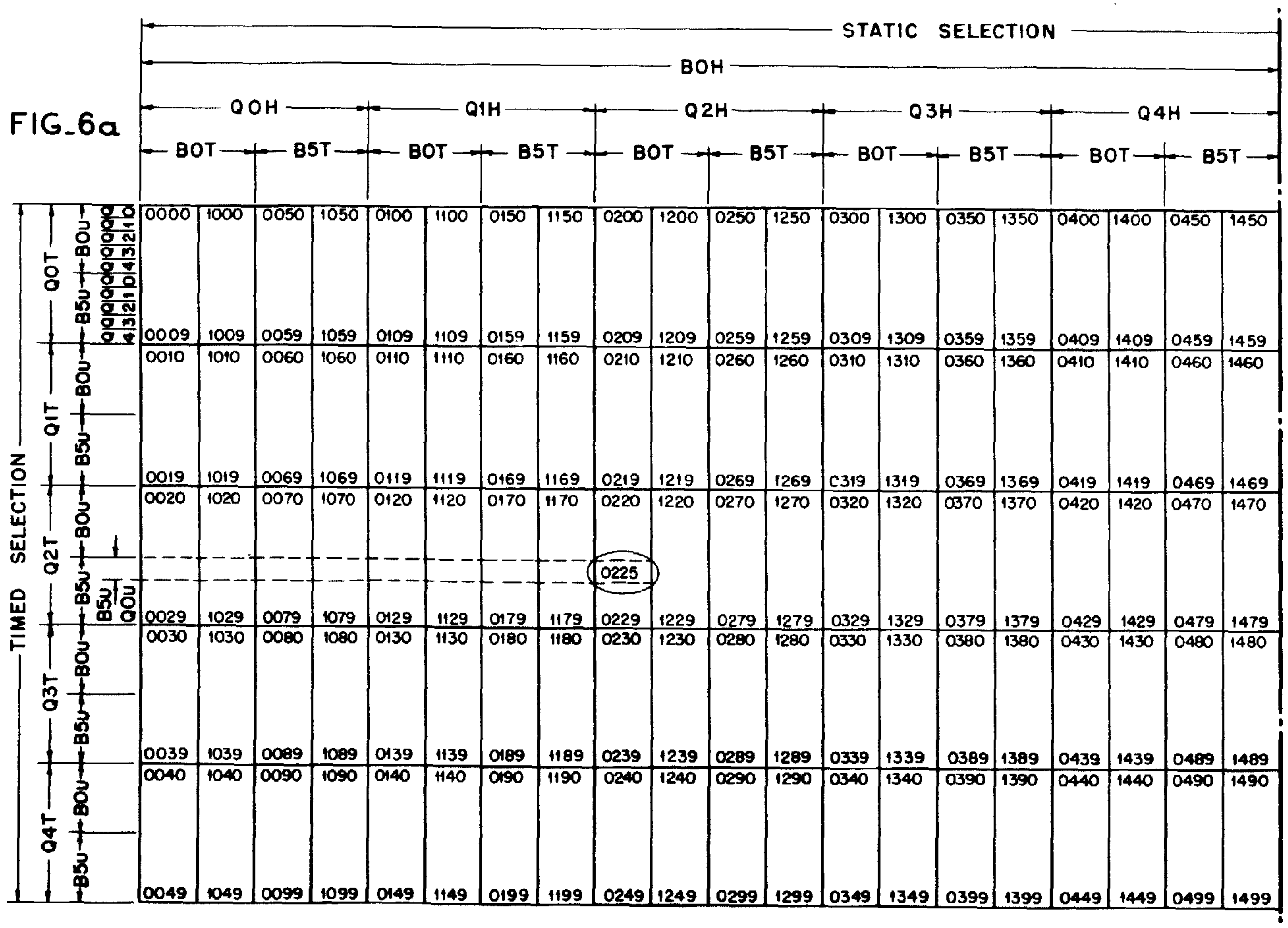

HP-HOME PULSE
SP-SECTOR PULSE
WP-WORD PULSE
BP-B PULSE
DP-D PULSE
RSP-READ SAMPLE PULSE
B5
B0
Q0
Q1
Q2
Q3
Q4
0
1
2
3
6
READ BUFFER STORAGE
WORD 3
WORD 4
WORD 5
WORD 6
WORD 7
WORD 8
WORD 9
WORD 10
B5
B0
Q0
Q1
Q2
Q3
Q4
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
0059
0060
0069
0070
0909
0910
0919
0920
1909
1910
1919
1920
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971

FIG. 5

FIG. 6a

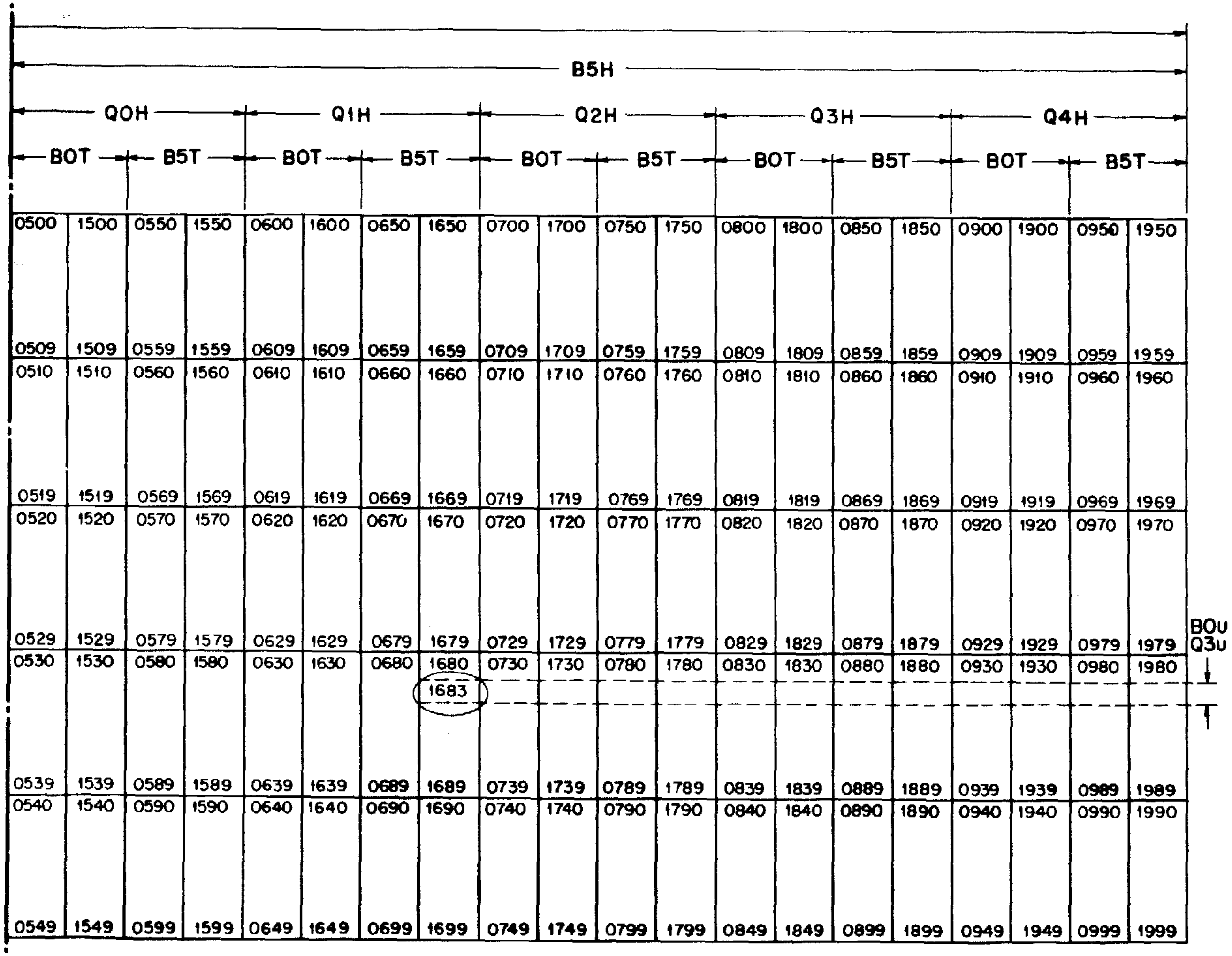


	STATIC SELECTION																			
	BOH																			
	Q0H				Q1H				Q2H				Q3H				Q4H			
	BOT		B5T		BOT		B5T		BOT		B5T		BOT		B5T		BOT		B5T	
Q0T	0000	1000	0050	1050	0100	1100	0150	1150	0200	1200	0250	1250	0300	1300	0350	1350	0400	1400	0450	1450
	0009	1009	0059	1059	0109	1109	0159	1159	0209	1209	0259	1259	0309	1309	0359	1359	0409	1409	0459	1459
Q1T	0010	1010	0060	1060	0110	1110	0160	1160	0210	1210	0260	1260	0310	1310	0360	1360	0410	1410	0460	1460
	0019	1019	0069	1069	0119	1119	0169	1169	0219	1219	0269	1269	C319	1319	0369	1369	0419	1419	0469	1469
Q2T	0020	1020	0070	1070	0120	1120	0170	1170	0220	1220	0270	1270	0320	1320	0370	1370	0420	1420	0470	1470
									0225											
	0029	1029	0079	1079	0129	1129	0179	1179	0229	1229	0279	1279	0329	1329	0379	1379	0429	1429	0479	1479
Q3T	0030	1030	0080	1080	0130	1130	0180	1180	0230	1230	0280	1280	0330	1330	0380	1380	0430	1430	0480	1480
	0039	1039	0089	1089	0139	1139	0189	1189	0239	1239	0289	1289	0339	1339	0389	1389	0439	1439	0489	1489
Q4T	0040	1040	0090	1090	0140	1140	0190	1190	0240	1240	0290	1290	0340	1340	0390	1390	0440	1440	0490	1490
	0049	1049	0099	1099	0149	1149	0199	1199	0249	1249	0299	1299	0349	1349	0399	1399	0449	1449	0499	1499

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 6



B5H																				
Q0H				Q1H				Q2H				Q3H				Q4H				
BOT		B5T		BOT		B5T		BOT		B5T		BOT		B5T		BOT		B5T		
0500 0509	1500 1509	0550 0559	1550 1559	0600 0609	1600 1609	0650 0659	1650 1659	0700 0709	1700 1709	0750 0759	1750 1759	0800 0809	1800 1809	0850 0859	1850 1859	0900 0909	1900 1909	0950 0959	1950 1959	
0510 0519	1510 1519	0560 0569	1560 1569	0610 0619	1610 1619	0660 0669	1660 1669	0710 0719	1710 1719	0760 0769	1760 1769	0810 0819	1810 1819	0860 0869	1860 1869	0910 0919	1910 1919	0960 0969	1960 1969	
0520 0529	1520 1529	0570 0579	1570 1579	0620 0629	1620 1629	0670 0679	1670 1679	0720 0729	1720 1729	0770 0779	1770 1779	0820 0829	1820 1829	0870 0879	1870 1879	0920 0929	1920 1929	0970 0979	1970 1979	BOU Q3U
0530 0539	1530 1539	0580 0589	1580 1589	0630 0639	1630 1639	0680 0689	1680 1683 1689	0730 0739	1730 1739	0780 0789	1780 1789	0830 0839	1830 1839	0880 0889	1880 1889	0930 0939	1930 1939	0980 0989	1980 1989	
0540 0549	1540 1549	0590 0599	1590 1599	0640 0649	1640 1649	0690 0699	1690 1699	0740 0749	1740 1749	0790 0799	1790 1799	0840 0849	1840 1849	0890 0899	1890 1899	0940 0949	1940 1949	0990 0999	1990 1999	

FIG_6b

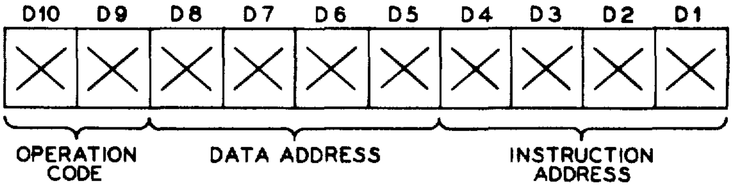

D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
OPERATION CODE
DATA ADDRESS
INSTRUCTION ADDRESS

FIG. 7

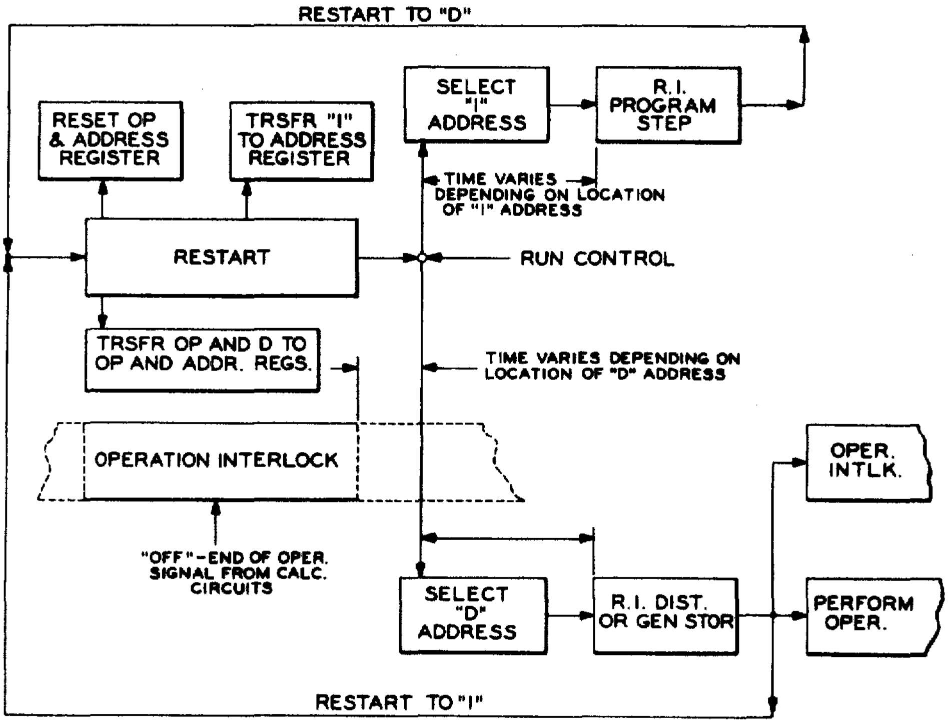

RESTART TO "D"
RESET OP & ADDRESS REGISTER
TRSFR "I" TO ADDRESS REGISTER
SELECT "I" ADDRESS
R.I. PROGRAM STEP
TIME VARIES DEPENDING ON LOCATION OF "I" ADDRESS
RESTART
RUN CONTROL
TRSFR OP AND D TO OP AND ADDR. REGS.
TIME VARIES DEPENDING ON LOCATION OF "D" ADDRESS
OPERATION INTERLOCK
OPER. INTLK.
"OFF"-END OF OPER. SIGNAL FROM CALC. CIRCUITS
SELECT "D" ADDRESS
R.I. DIST. OR GEN STOR
PERFORM OPER.
RESTART TO "I"

FIG. 8

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 8

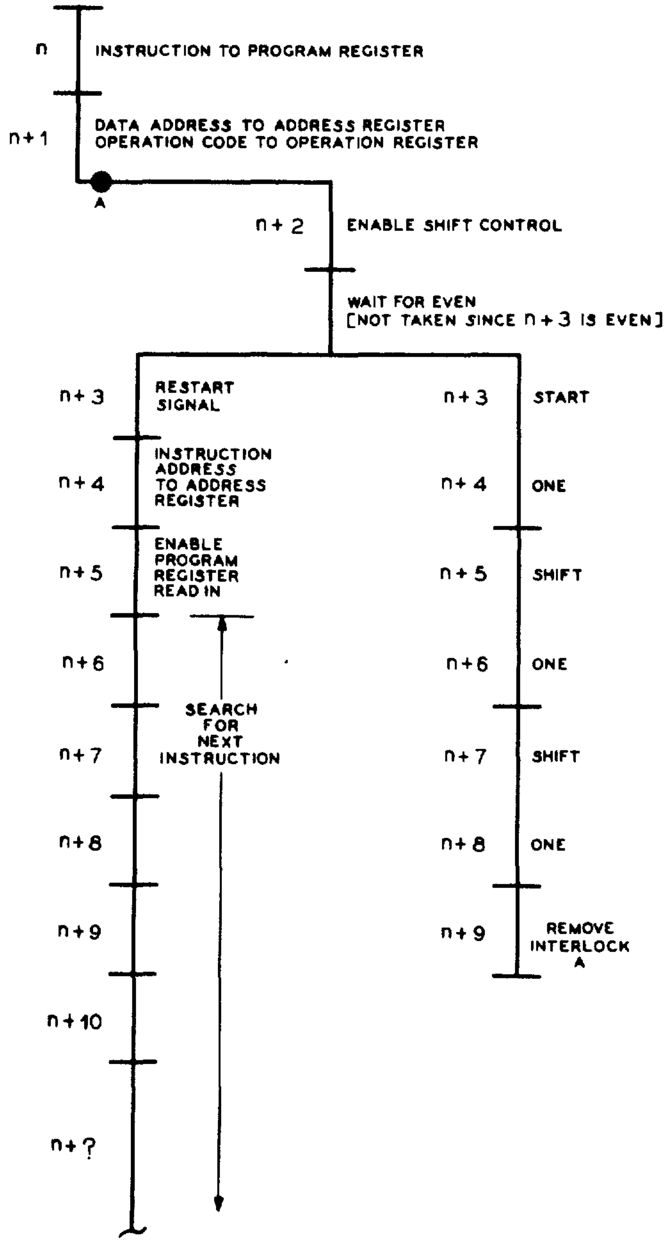

n
INSTRUCTION TO PROGRAM REGISTER
n+1
DATA ADDRESS TO ADDRESS REGISTER
OPERATION CODE TO OPERATION REGISTER
A
n+2
ENABLE SHIFT CONTROL
WAIT FOR EVEN
[NOT TAKEN SINCE n+3 IS EVEN]
n+3
RESTART SIGNAL
n+4
INSTRUCTION ADDRESS TO ADDRESS REGISTER
n+5
ENABLE PROGRAM REGISTER READ IN
n+6
n+7
SEARCH FOR NEXT INSTRUCTION
n+8
n+9
n+10
n+?
n+3
START
n+4
ONE
n+5
SHIFT
n+6
ONE
n+7
SHIFT
n+8
ONE
n+9
REMOVE INTERLOCK A

FIG_9

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

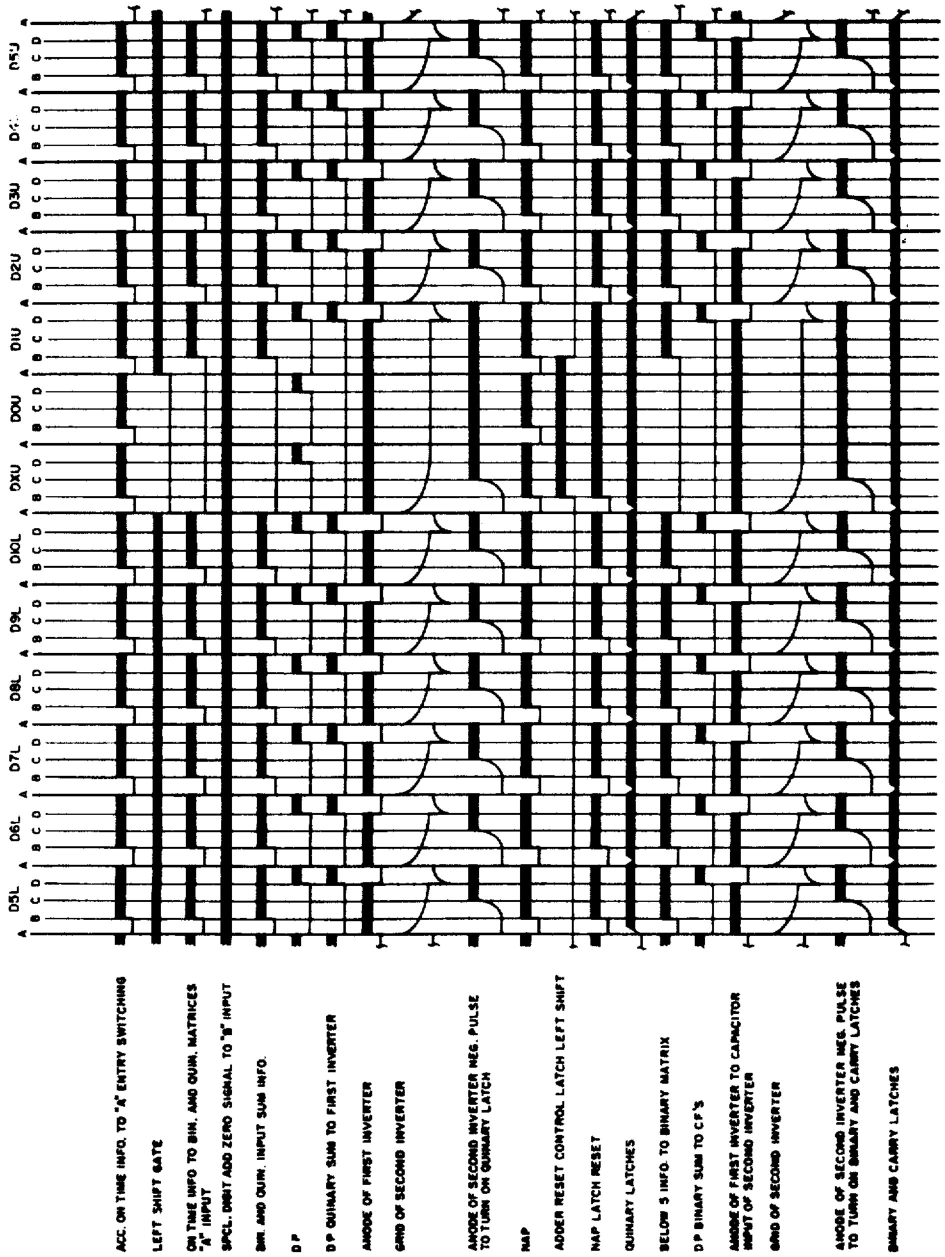

D5L A B C D
D6L A B C D
D7L A B C D
D8L A B C D
D9L A B C D
D10L A B C D
DXU A B C D
D0U A B C D
D1U A B C D
D2U A B C D
D3U A B C D
D4U A B C D
D5U A B C D
ACC. ON TIME INFO. TO "A" ENTRY SWITCHING
LEFT SHIFT GATE
ON TIME INFO TO BIN. AND QUIN. MATRICES "A" INPUT
SPCL. DIGIT ADD ZERO SIGNAL TO "B" INPUT
BIN. AND QUIN. INPUT SUM INFO.
D P
D P QUINARY SUM TO FIRST INVERTER
ANODE OF FIRST INVERTER
GRID OF SECOND INVERTER
ANODE OF SECOND INVERTER NEG. PULSE TO TURN ON QUINARY LATCH
NAP
ADDER RESET CONTROL LATCH LEFT SHIFT
NAP LATCH RESET
QUINARY LATCHES
BELOW 5 INFO. TO BINARY MATRIX
D P BINARY SUM TO CF'S
ANODE OF FIRST INVERTER TO CAPACITOR INPUT OF SECOND INVERTER
GRID OF SECOND INVERTER
ANODE OF SECOND INVERTER NEG. PULSE TO TURN ON BINARY AND CARRY LATCHES
BINARY AND CARRY LATCHES

FIG. 12

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 10

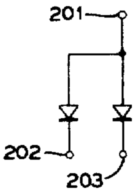


FIG. 13

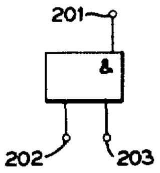


FIG. 13a

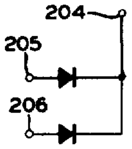


FIG. 14

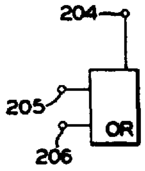


FIG. 14a

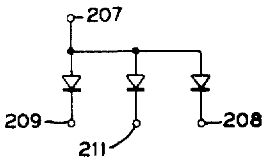


FIG. 15

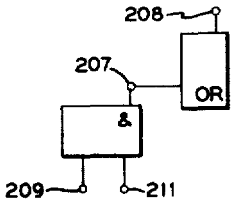


FIG. 15a

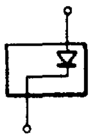


FIG. 16

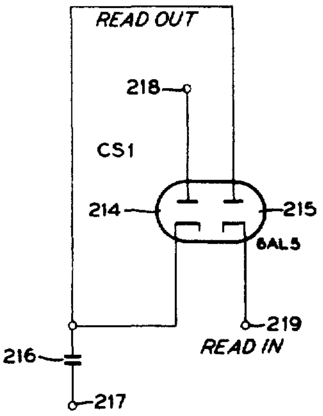


FIG. 18

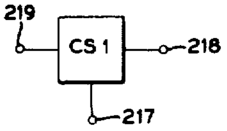


FIG. 18a

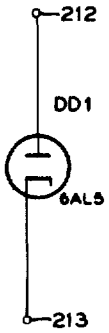


FIG. 17

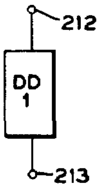


FIG. 17a

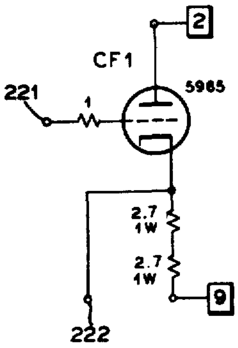


FIG. 19

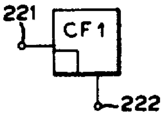


FIG. 19a

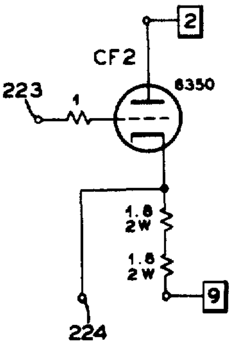


FIG. 20

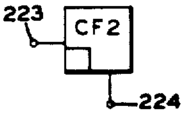


FIG. 20a

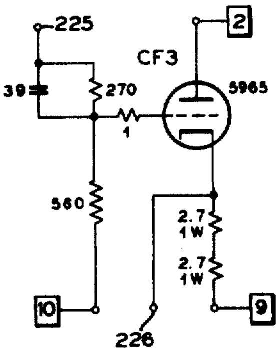


FIG. 21

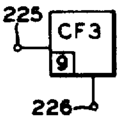


FIG. 21a

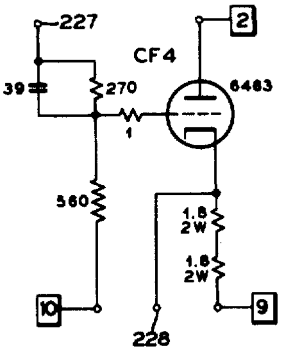


FIG. 22

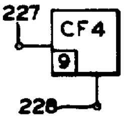


FIG. 22a

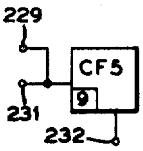


FIG. 23a

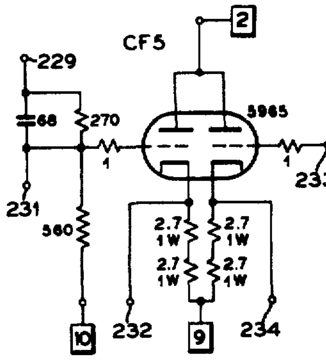


FIG. 23

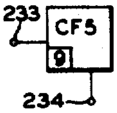


FIG. 23b

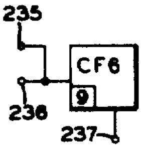


FIG. 24a

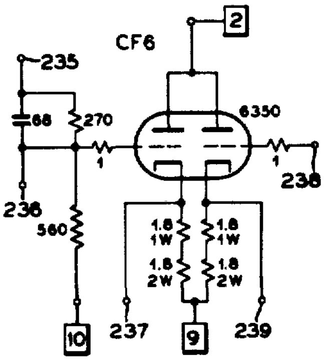


FIG. 24

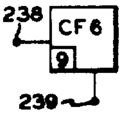


FIG. 24b

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 12

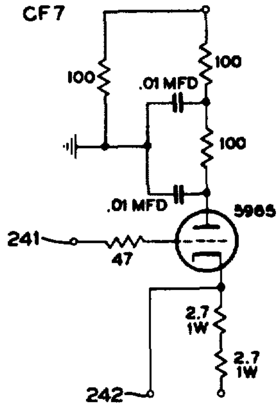


FIG. 25

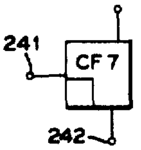


FIG. 25a

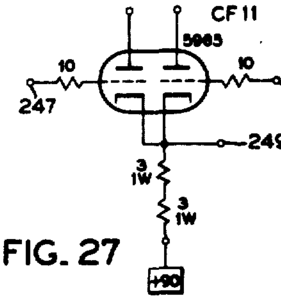


FIG. 27

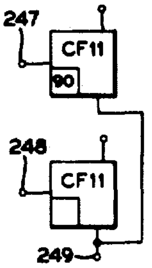


FIG. 27a

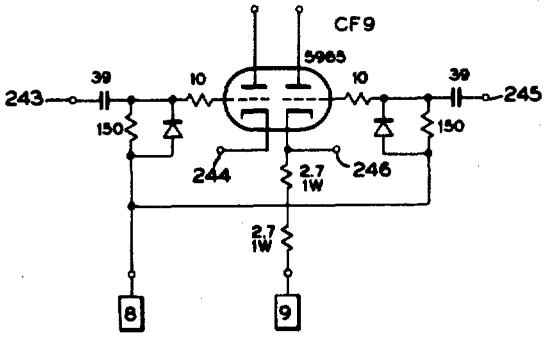


FIG. 26

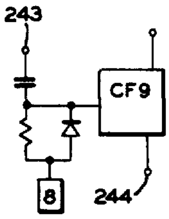


FIG. 26a

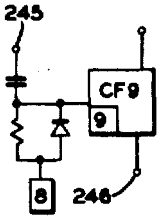


FIG. 26b

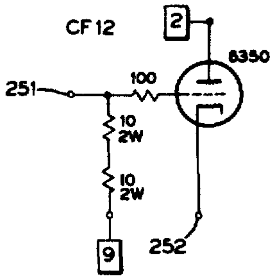


FIG. 28

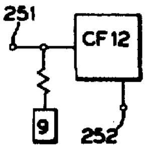


FIG. 28a

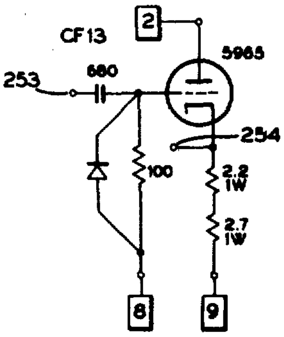


FIG. 29

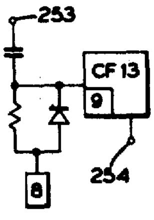


FIG. 29a

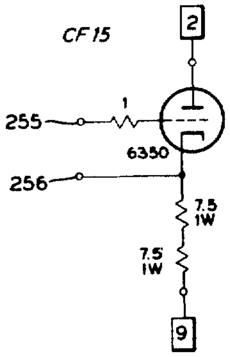


FIG. 30

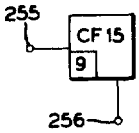


FIG. 30a

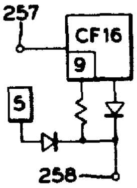


FIG. 31a

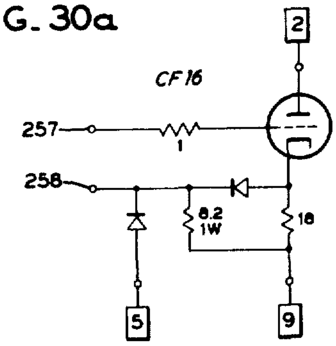


FIG. 31

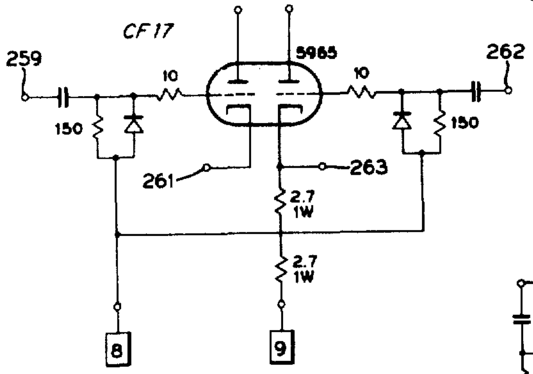


FIG. 32

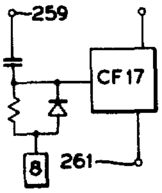


FIG. 32a

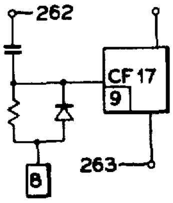


FIG. 32b

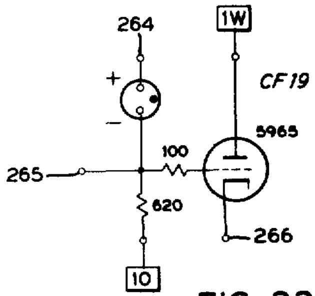


FIG. 33

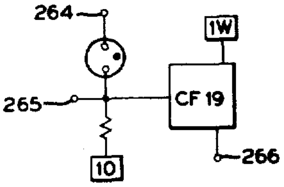


FIG. 33a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 14

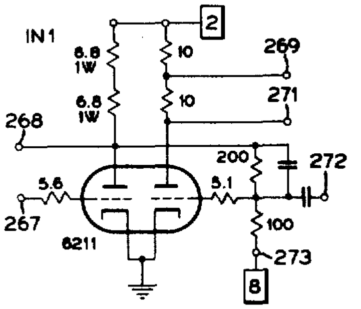


FIG. 34

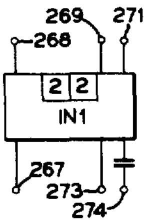


FIG. 34a

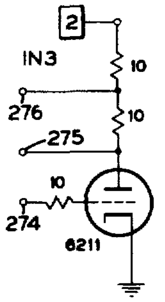


FIG. 35

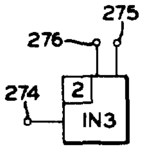


FIG. 35a

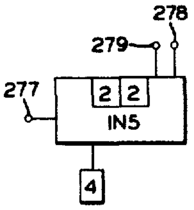


FIG. 36a

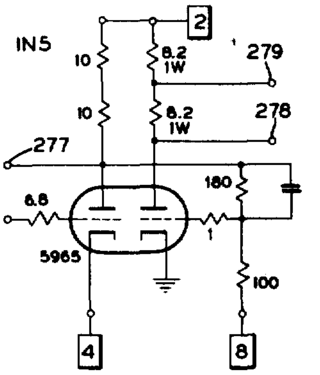


FIG. 36

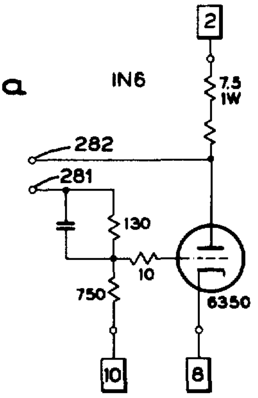


FIG. 37

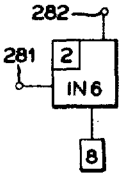


FIG. 37a

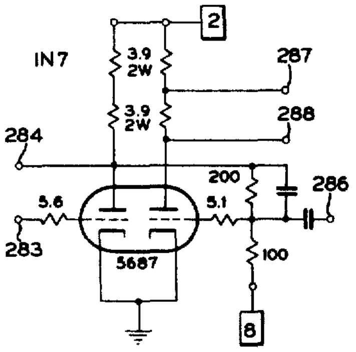


FIG. 38

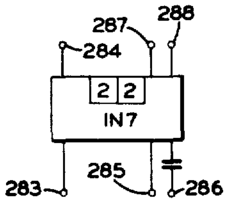


FIG. 38a

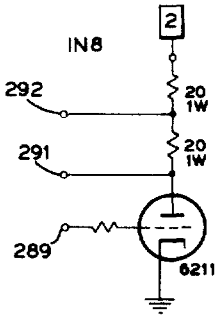


FIG. 39

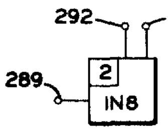


FIG. 39a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 15

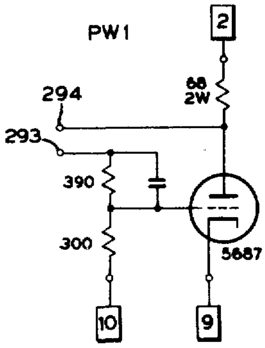


FIG. 40

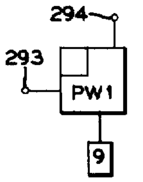


FIG. 40a

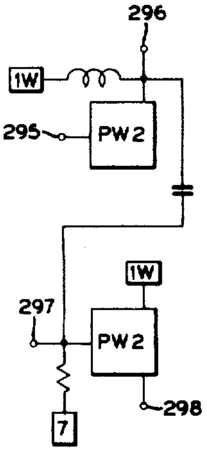


FIG. 41a

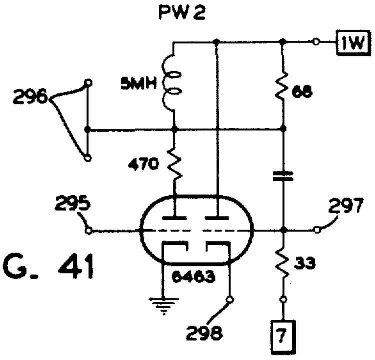


FIG. 41

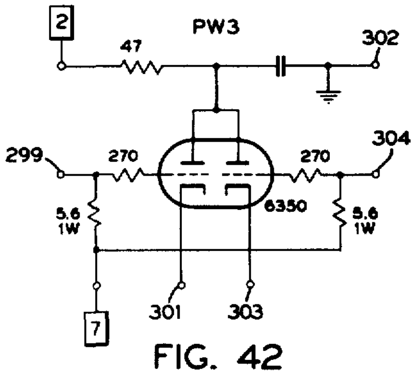


FIG. 42

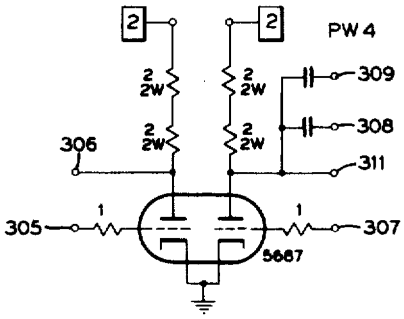


FIG. 43

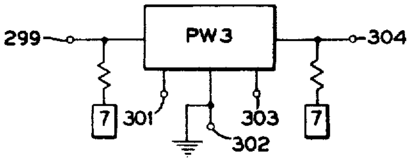


FIG. 42a

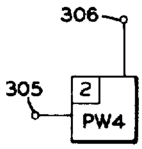


FIG. 43a

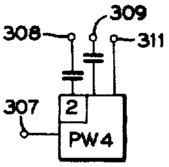


FIG. 43b

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 16

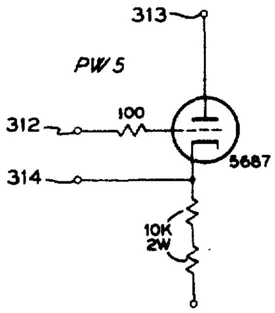


FIG. 44

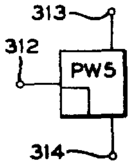


FIG. 44a

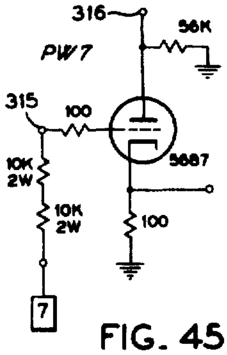


FIG. 45

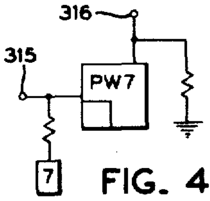


FIG. 45a

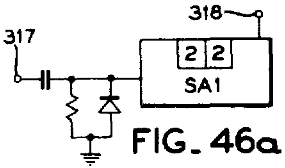


FIG. 46a

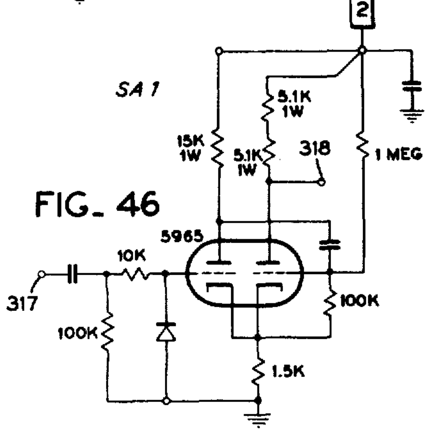


FIG. 46

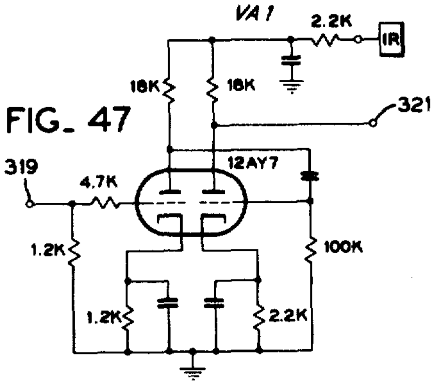


FIG. 47

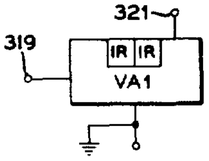


FIG. 47a

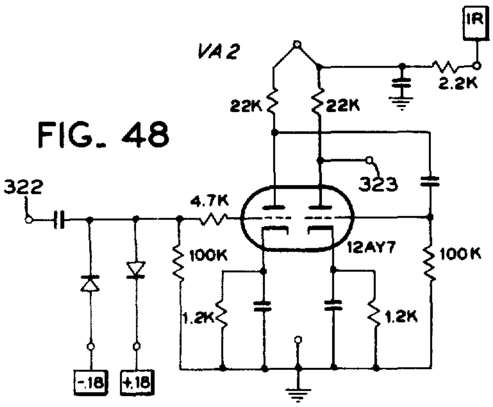


FIG. 48

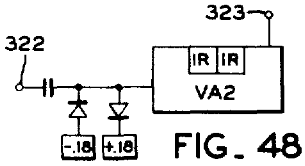


FIG. 48a

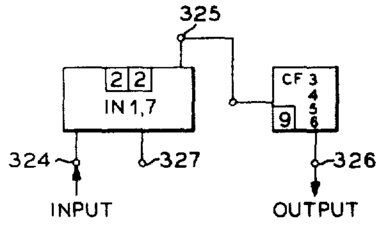


FIG_49

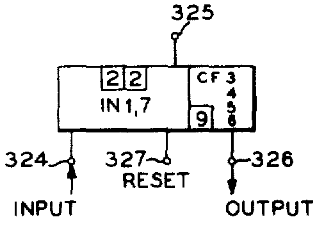


FIG_49a

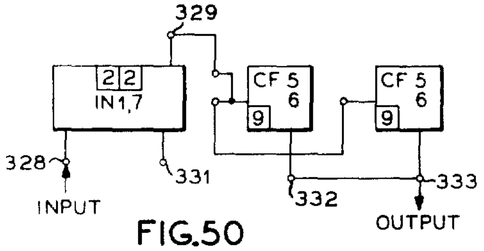


FIG.50

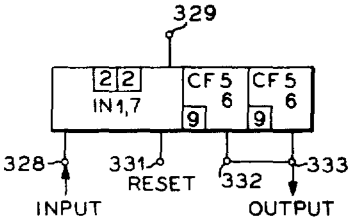


FIG_50a

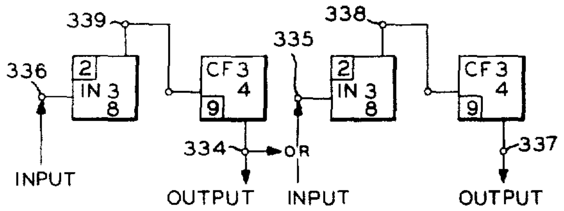


FIG_51

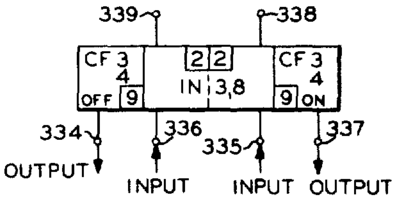


FIG_51a

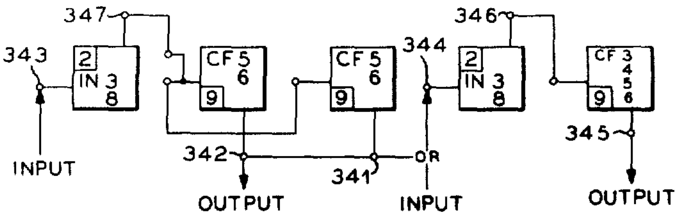


FIG_52

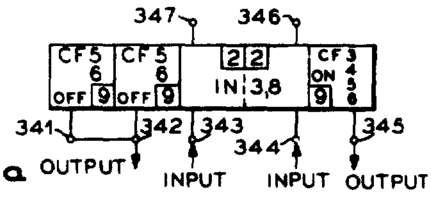


FIG_52a

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

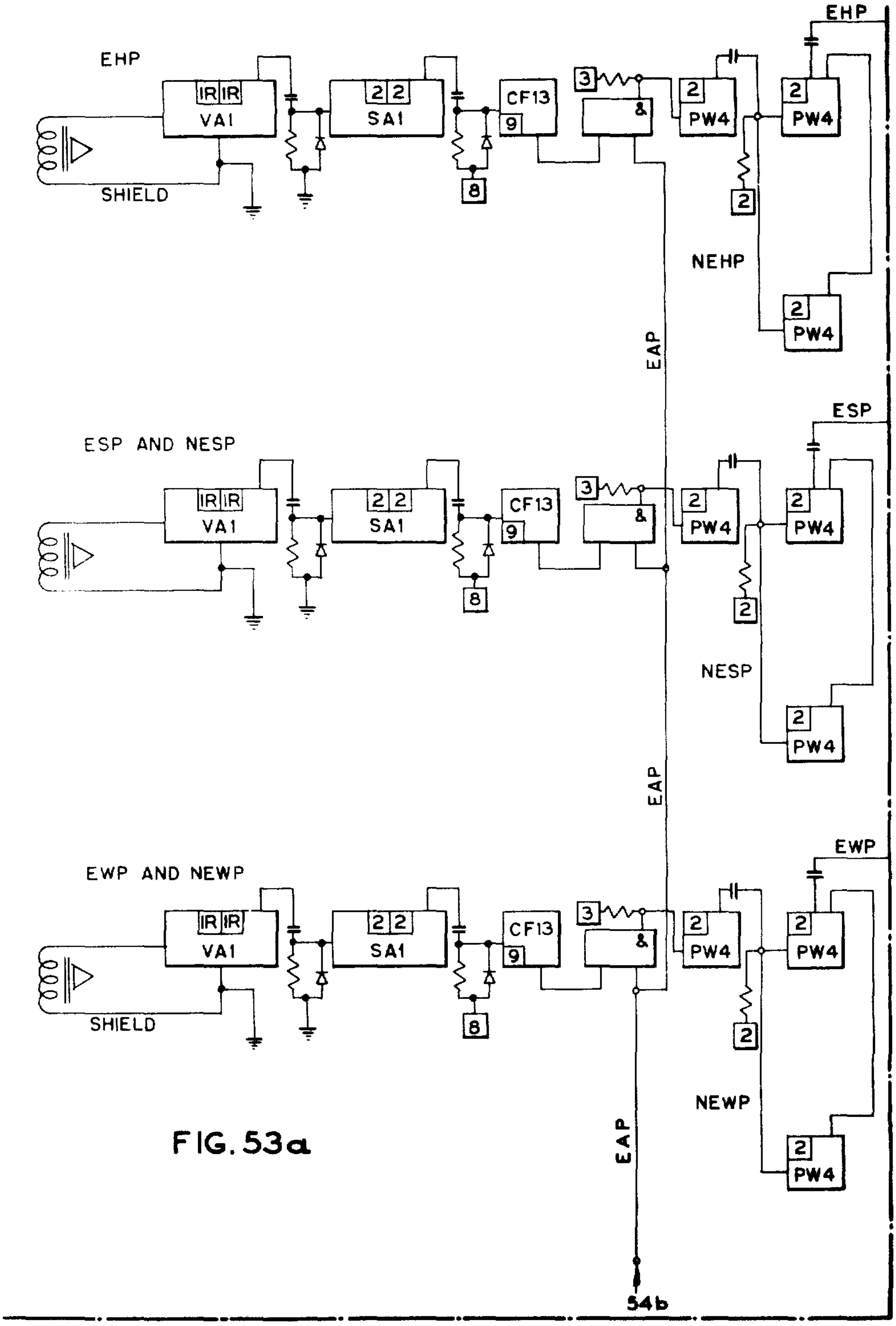
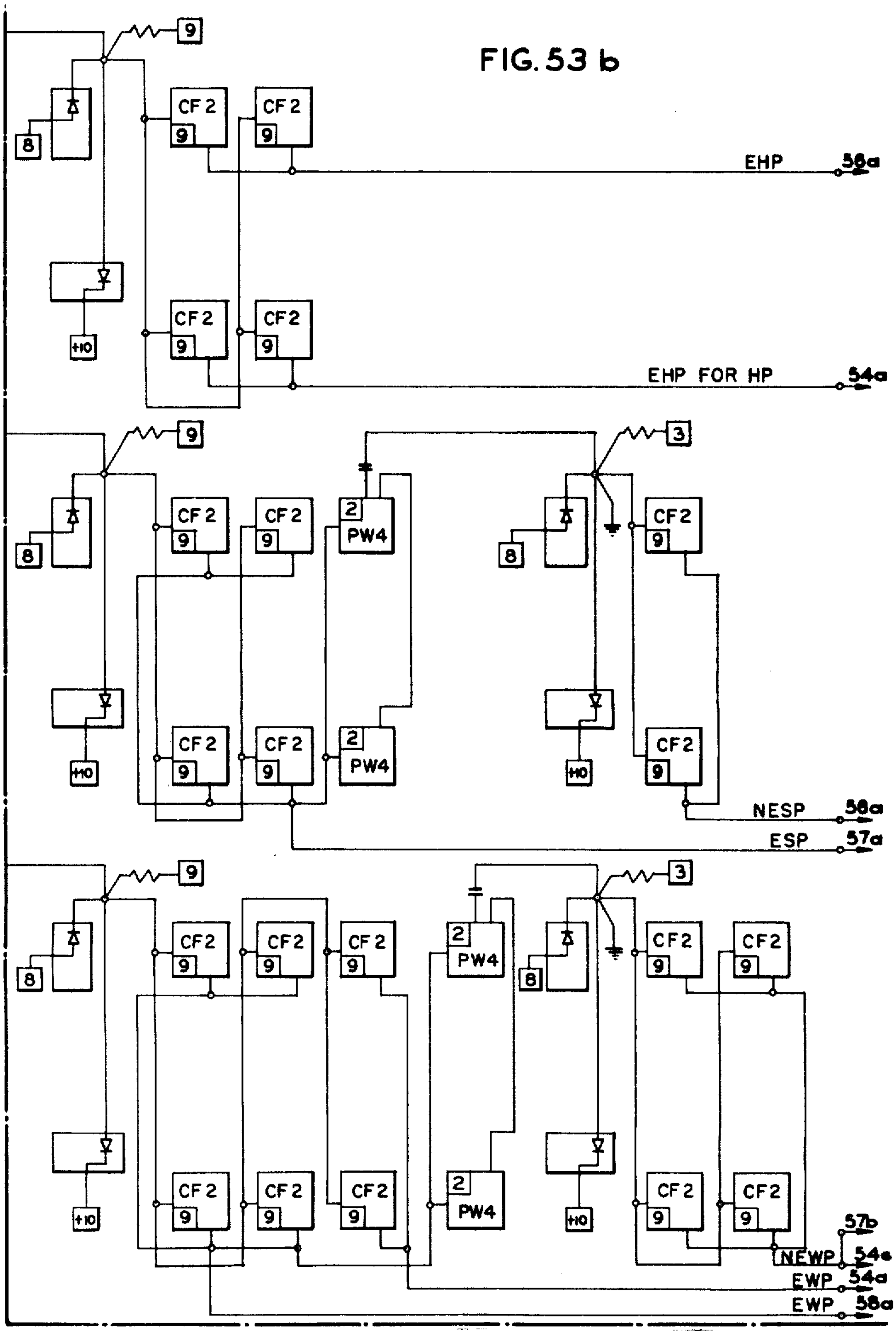


FIG. 53a

FIG. 53 b


9
8
CF 2
9
+10
EHP
58a
EHP FOR HP
54a
2
PW4
3
NESP
58a
ESP
57a
57b
NEWP
54e
EWP
54a
EWP
58a

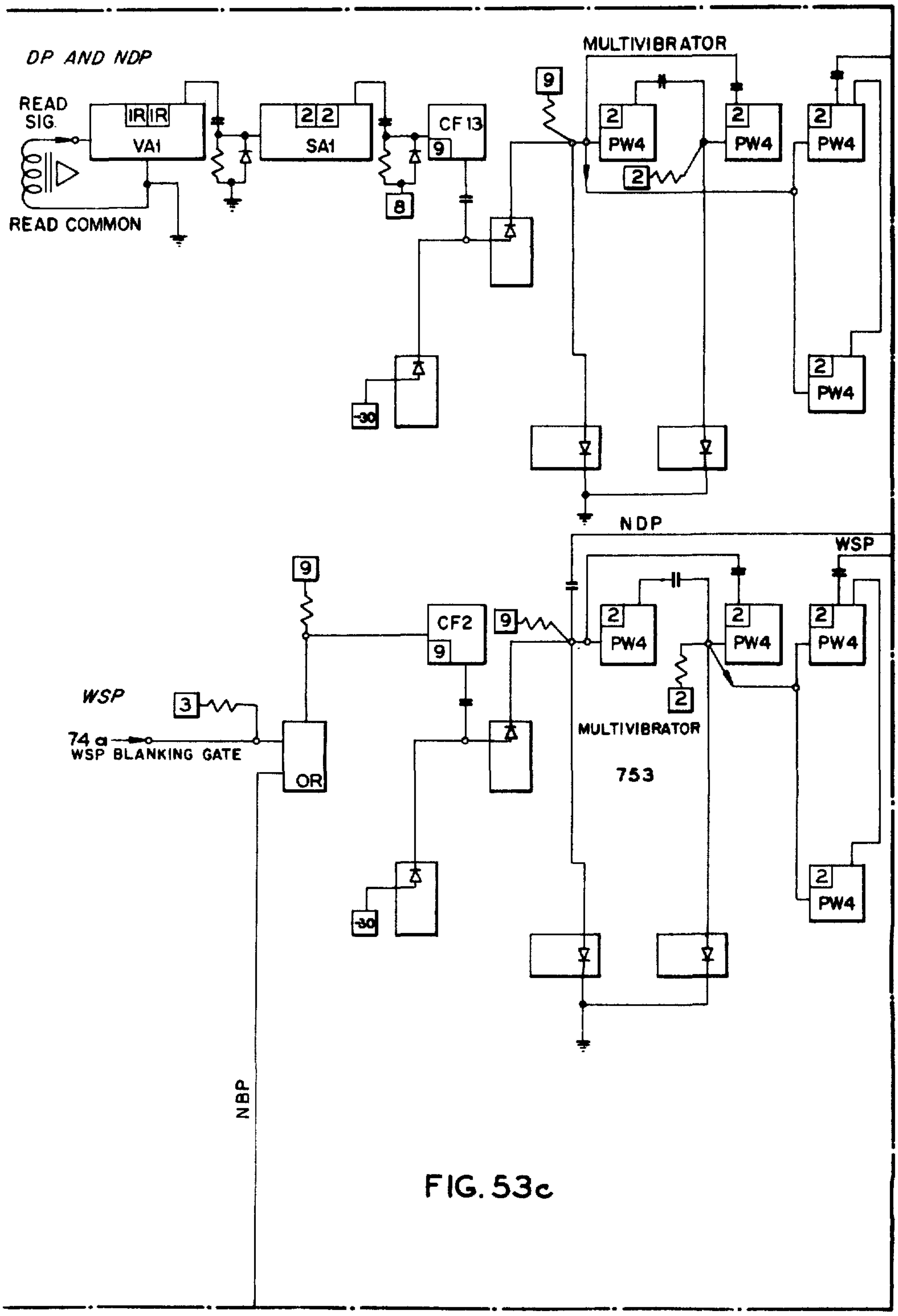

DP AND NDP
READ SIG.
IR IR
VA1
READ COMMON
2 2
SA1
CF 13
9
8
9
MULTIVIBRATOR
2
PW4
2
PW4
2
PW4
2
2
PW4
-30
NDP
WSP
9
CF2
9
9
2
PW4
2
PW4
2
PW4
2
MULTIVIBRATOR
753
WSP
3
74 a
WSP BLANKING GATE
OR
-30
2
PW4
NBP

FIG. 53c

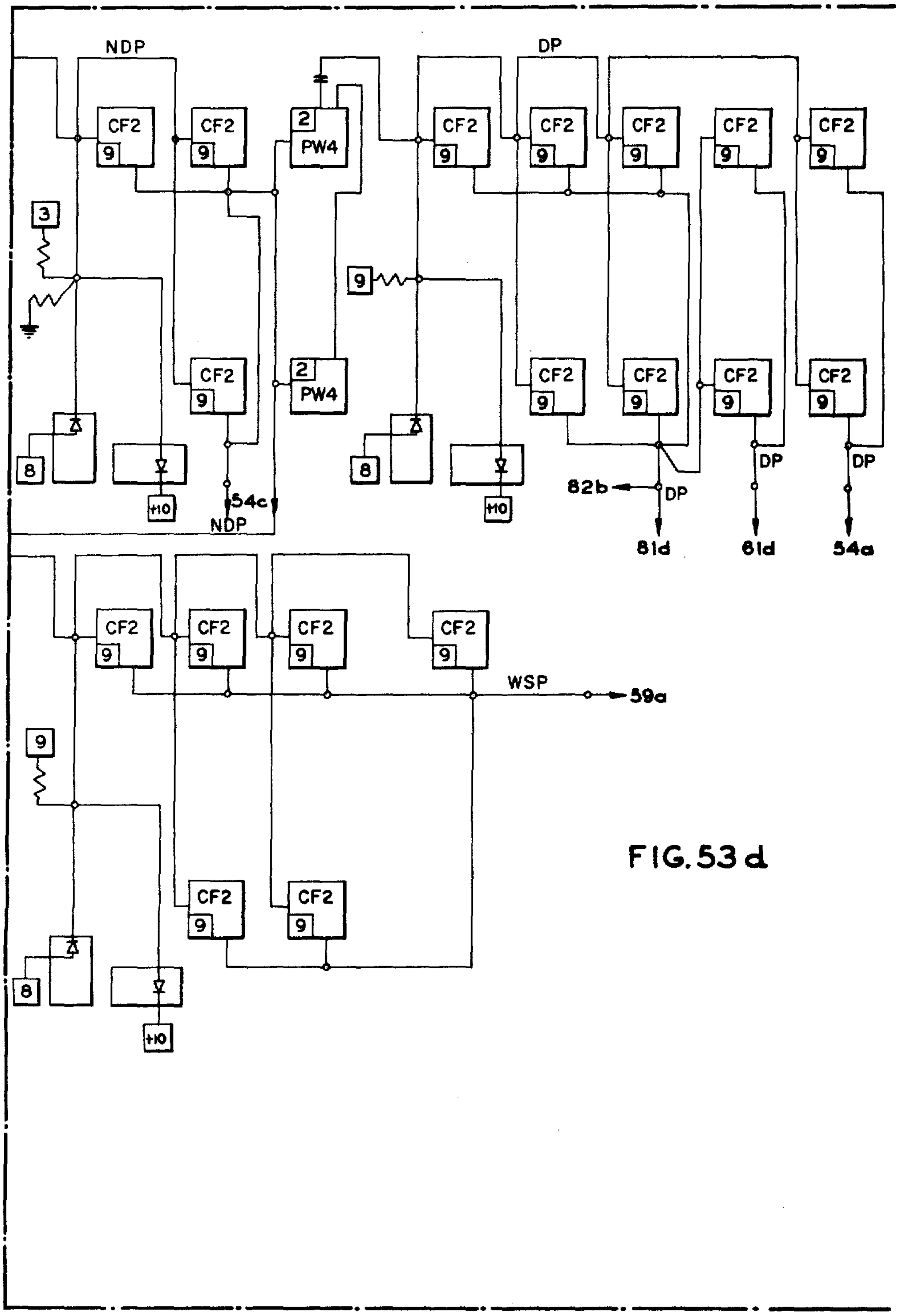


FIG. 53 d

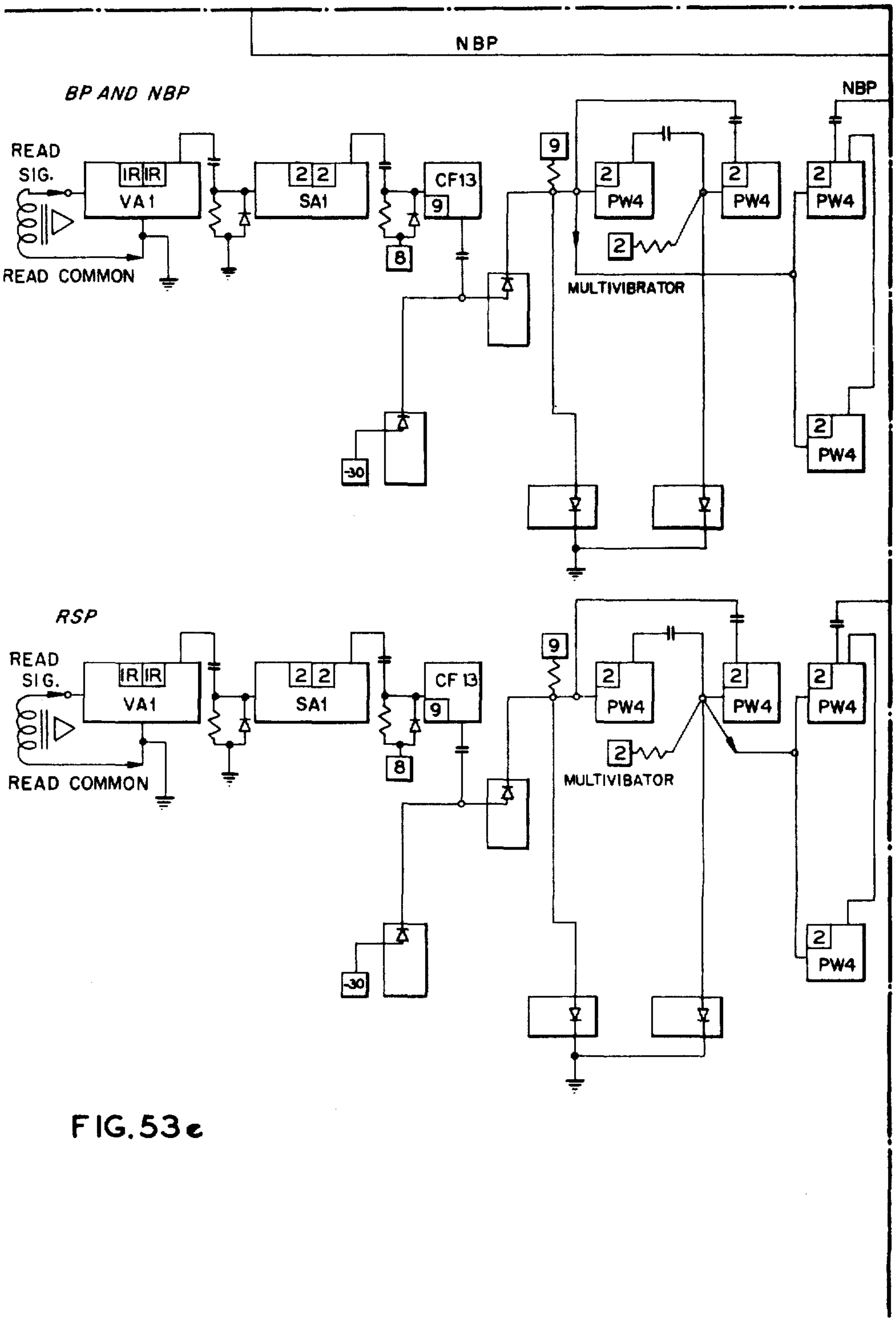

NBP
BP AND NBP
NBP
READ
SIG.
IR IR
VA1
READ COMMON
2 2
SA1
CF13
9
8
9
2
PW4
2
PW4
2
PW4
2
MULTIVIBRATOR
-30
2
PW4
RSP
READ
SIG.
IR IR
VA1
READ COMMON
2 2
SA1
CF 13
9
8
9
2
PW4
2
PW4
2
PW4
2
MULTIVIBATOR
-30
2
PW4

FIG. 53e

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 23

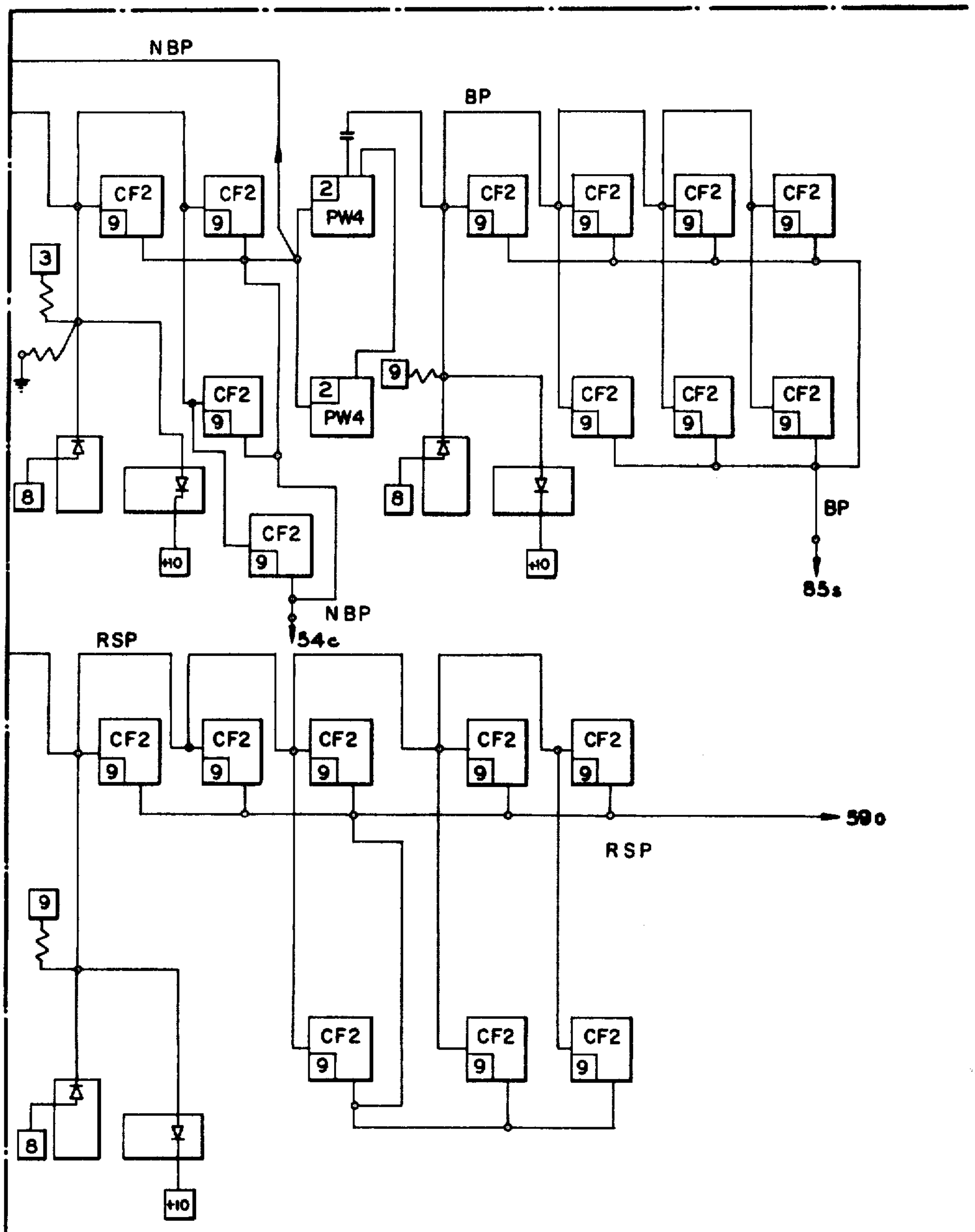


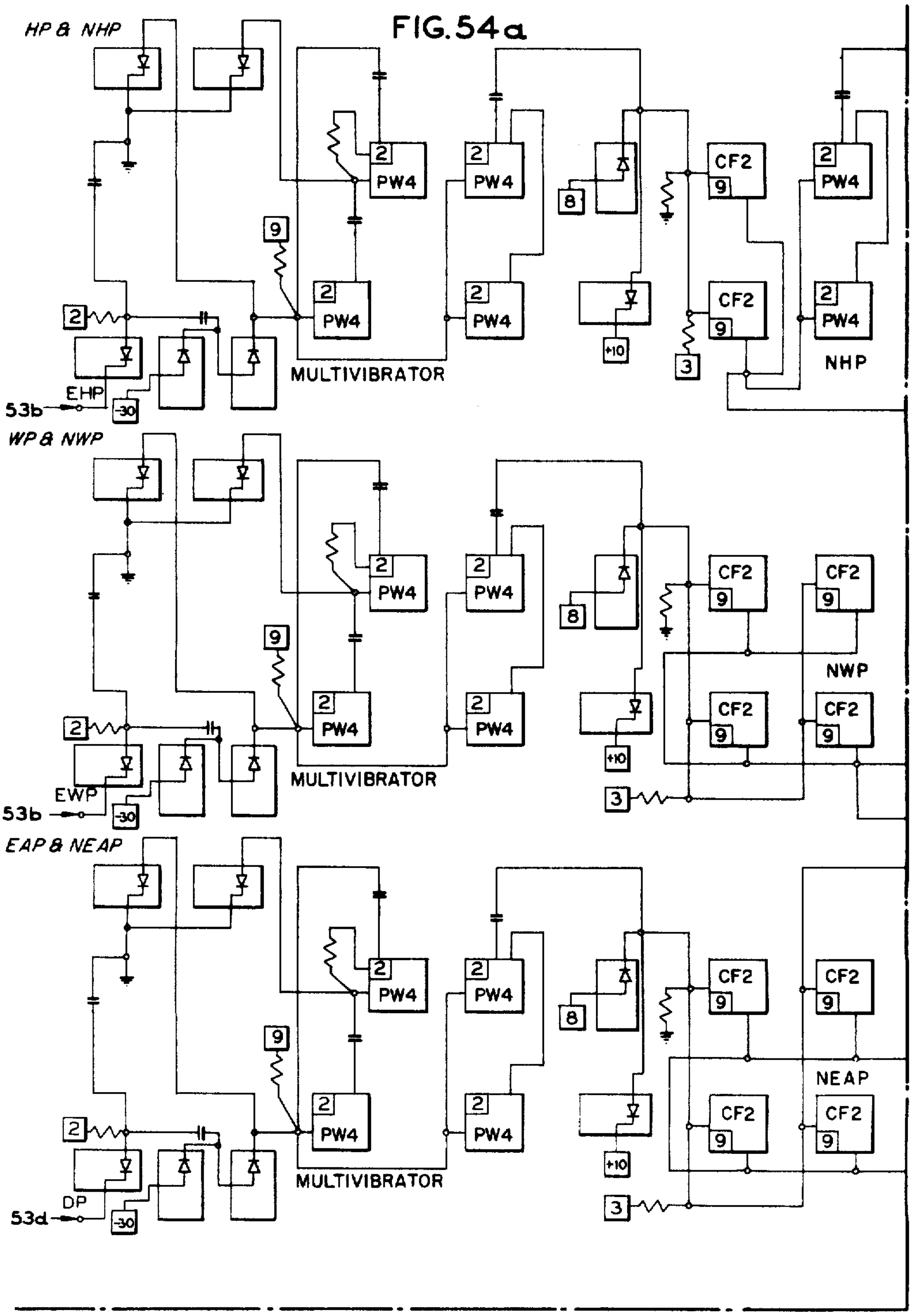
FIG.53f

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 24

FIG. 54a

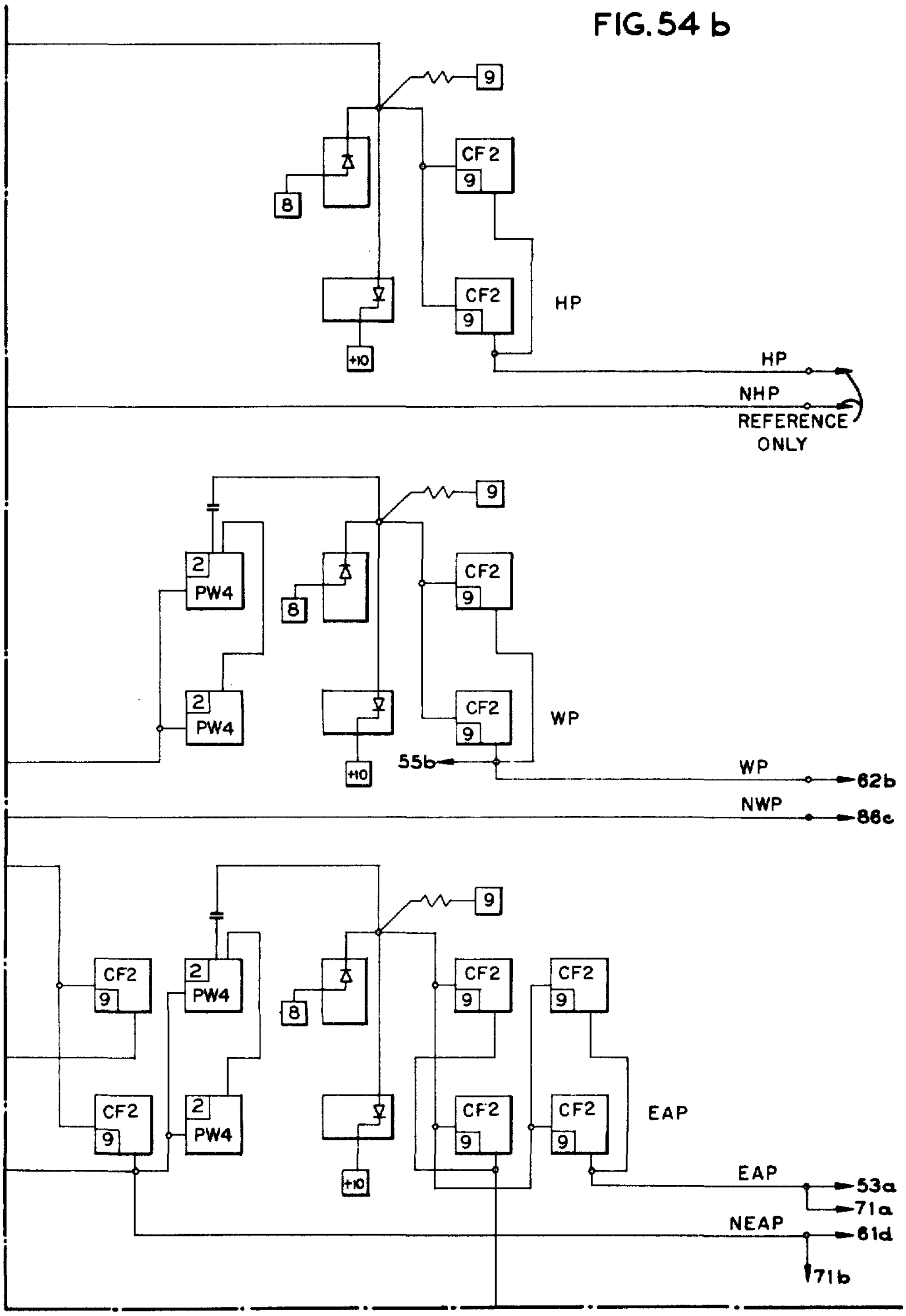


Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 25

FIG. 54 b



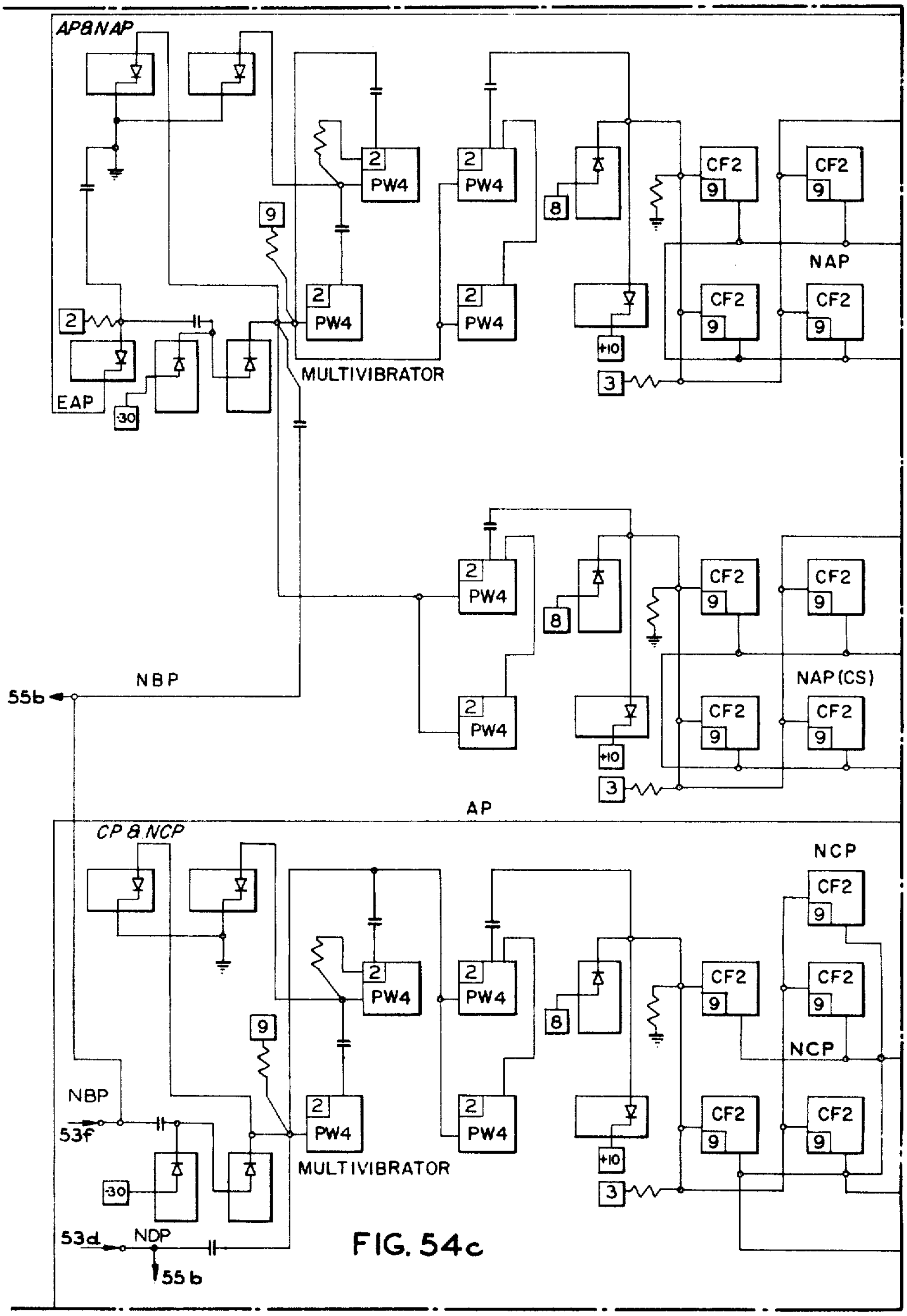
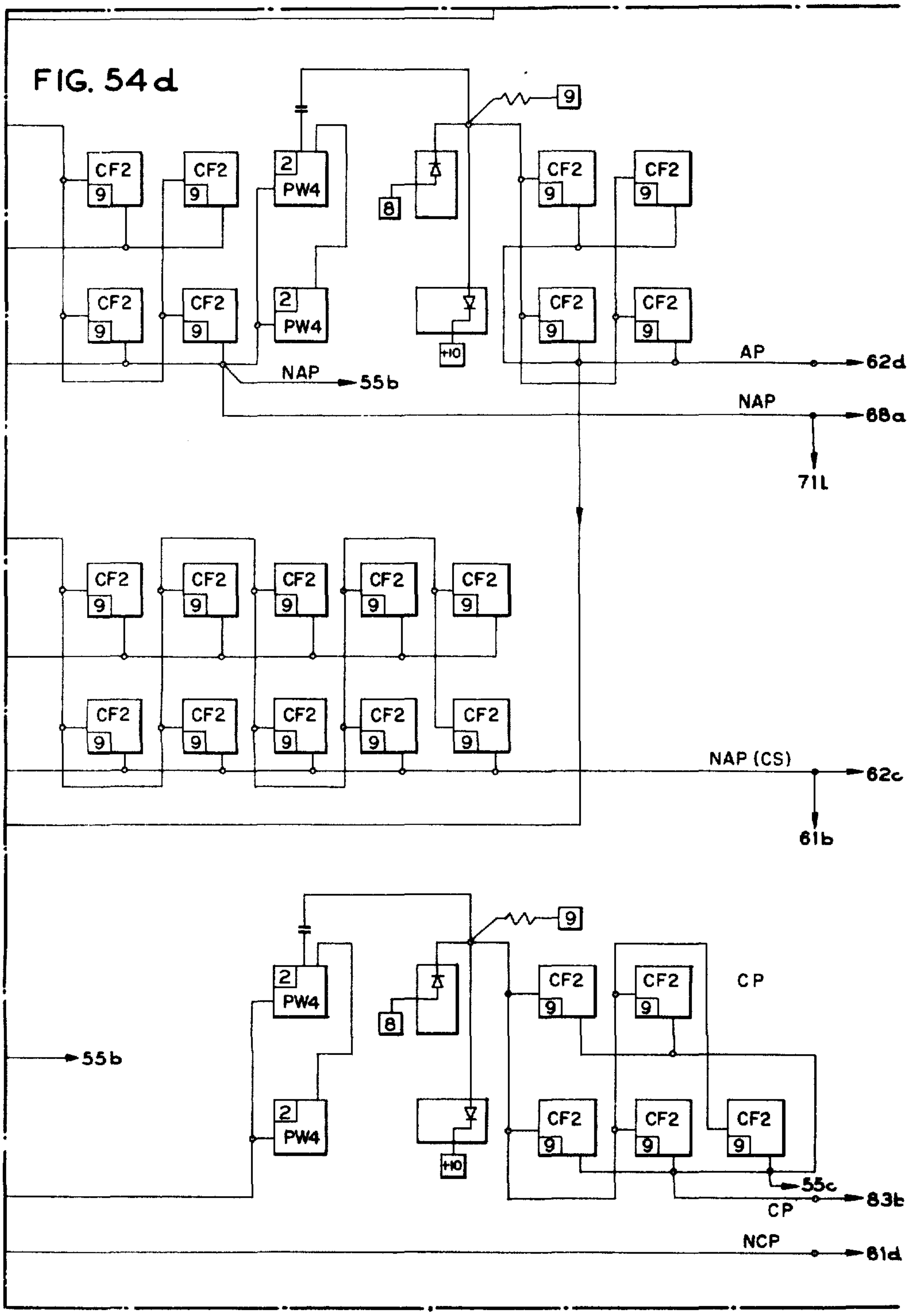

AP&NAP
2
PW4
2
PW4
9
2
PW4
2
PW4
8
CF2
9
CF2
9
NAP
CF2
9
CF2
9
2
EAP
-30
MULTIVIBRATOR
+10
3
2
PW4
8
CF2
9
CF2
9
2
PW4
NAP (CS)
CF2
9
CF2
9
55b
NBP
+10
3
AP
CP & NCP
NCP
CF2
9
2
PW4
2
PW4
8
CF2
9
CF2
9
9
NCP
NBP
53f
2
PW4
2
PW4
CF2
9
CF2
9
-30
MULTIVIBRATOR
+10
3
53d
NDP
55b

FIG. 54c

FIG. 54d


CF2
9
2
PW4
8
+10
AP
62d
NAP
55b
NAP
68a
71L
NAP (CS)
62c
61b
55b
CP
55c
CP
83b
NCP
81d

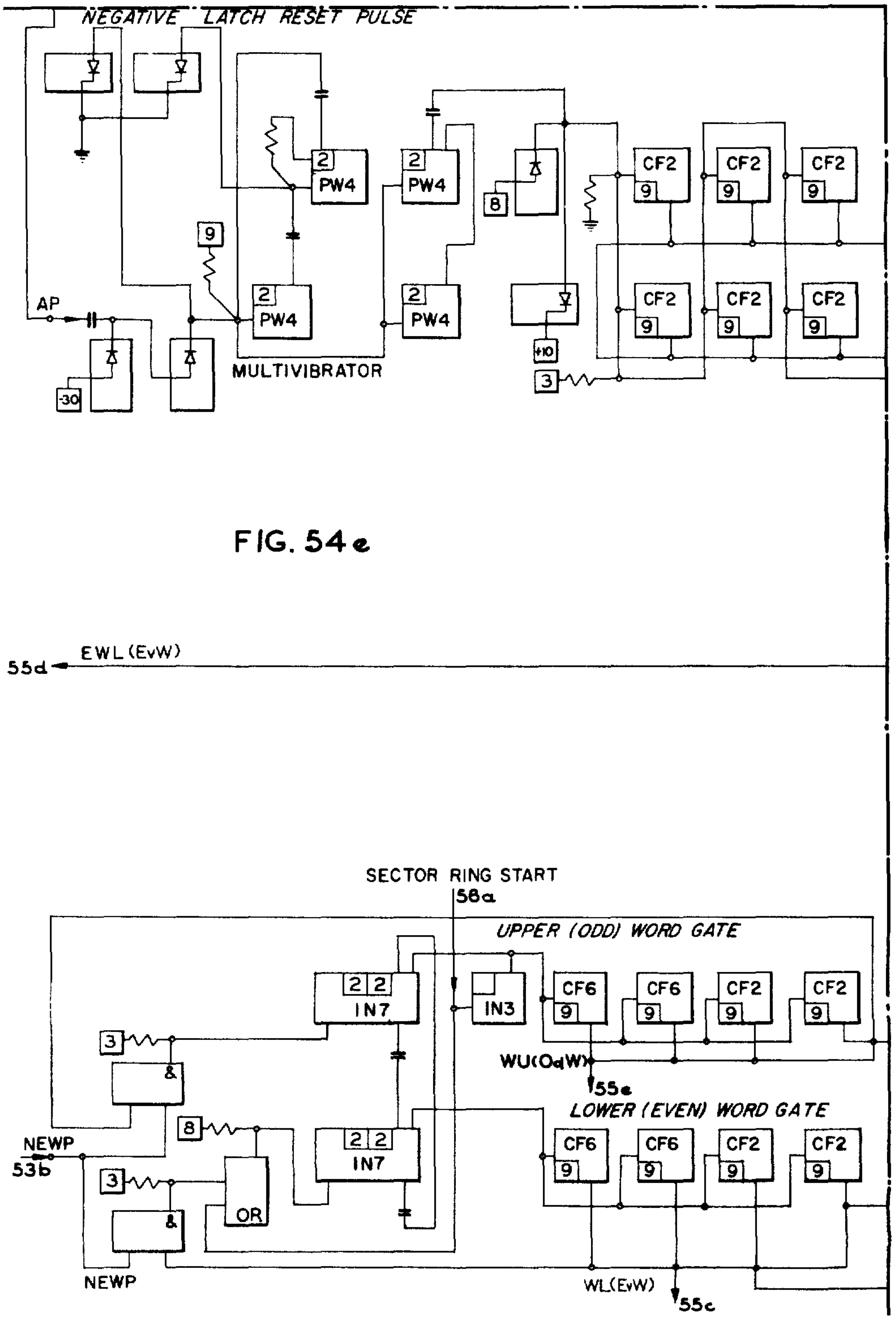

NEGATIVE LATCH RESET PULSE
2
PW4
2
PW4
8
9
CF2
9
CF2
9
CF2
9
2
PW4
2
PW4
CF2
9
CF2
9
CF2
9
AP
-30
MULTIVIBRATOR
+10
3
EWL (EvW)
55d
SECTOR RING START
58a
UPPER (ODD) WORD GATE
2 2
IN7
IN3
CF6
9
CF6
9
CF2
9
CF2
9
3
&
WU(OdW)
55e
LOWER (EVEN) WORD GATE
8
NEWP
53b
2 2
IN7
CF6
9
CF6
9
CF2
9
CF2
9
3
OR
&
NEWP
WL(EvW)
55c

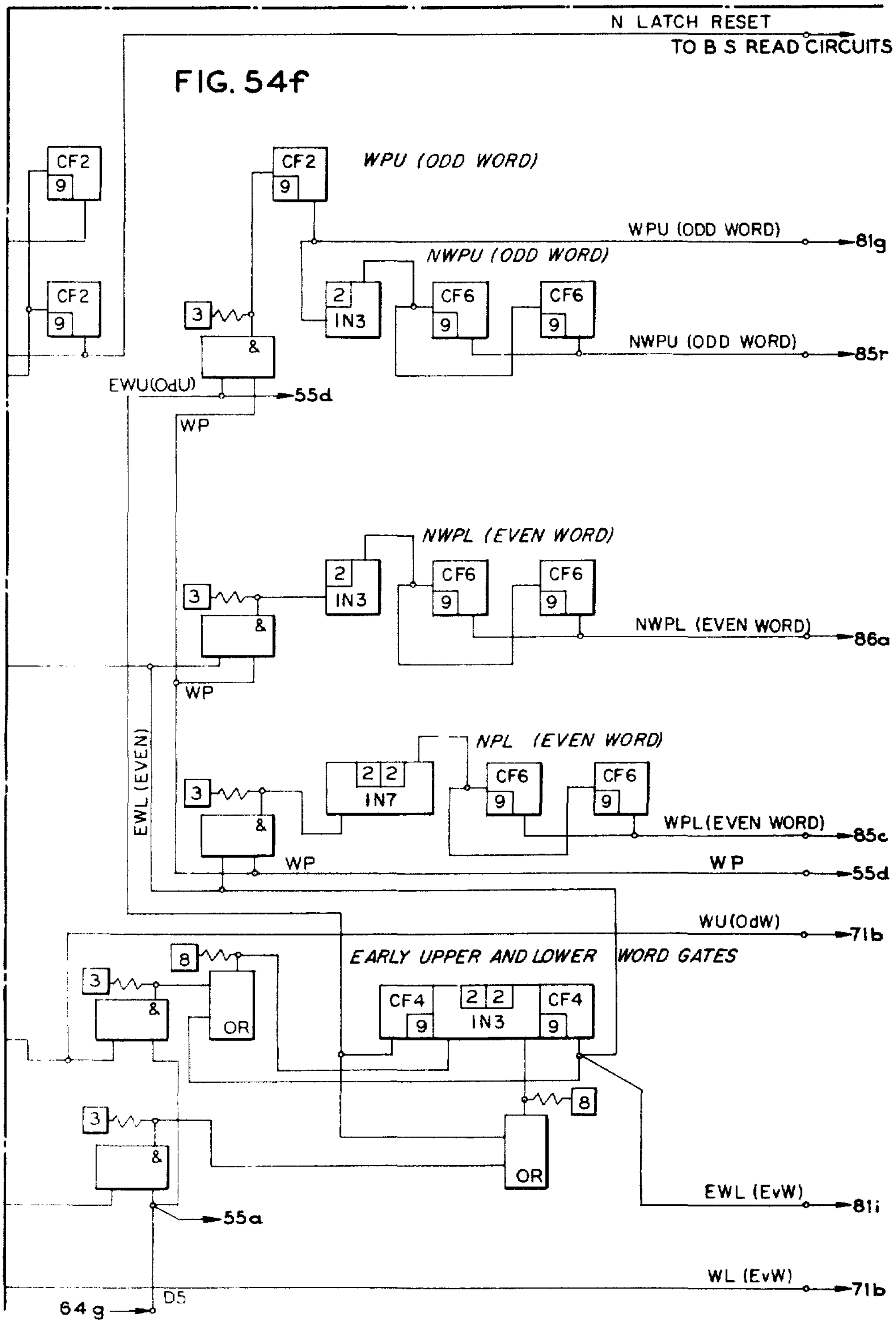
FIG. 54e

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 29

FIG. 54f



Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 30

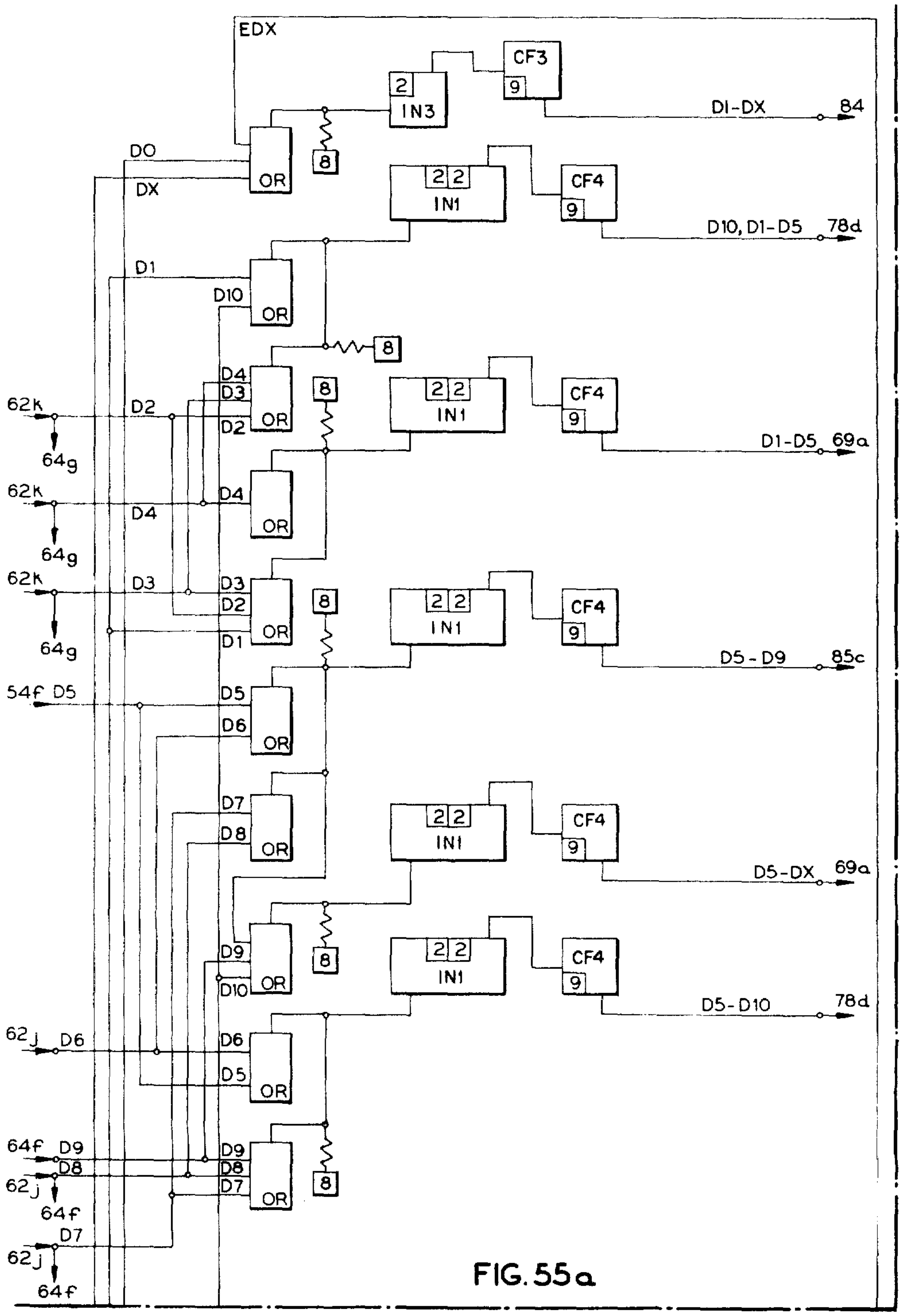


FIG. 55a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 31

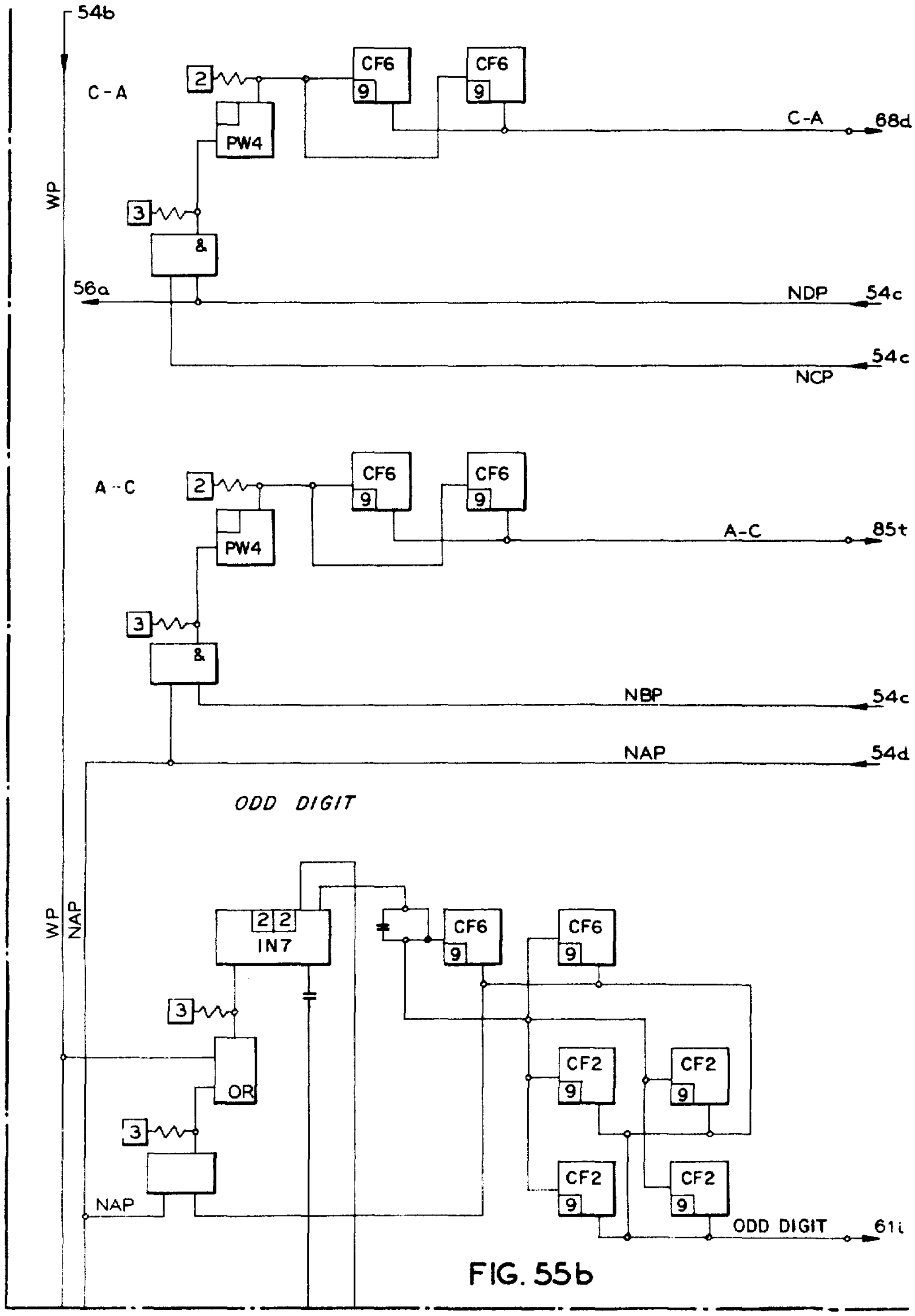


FIG. 55b

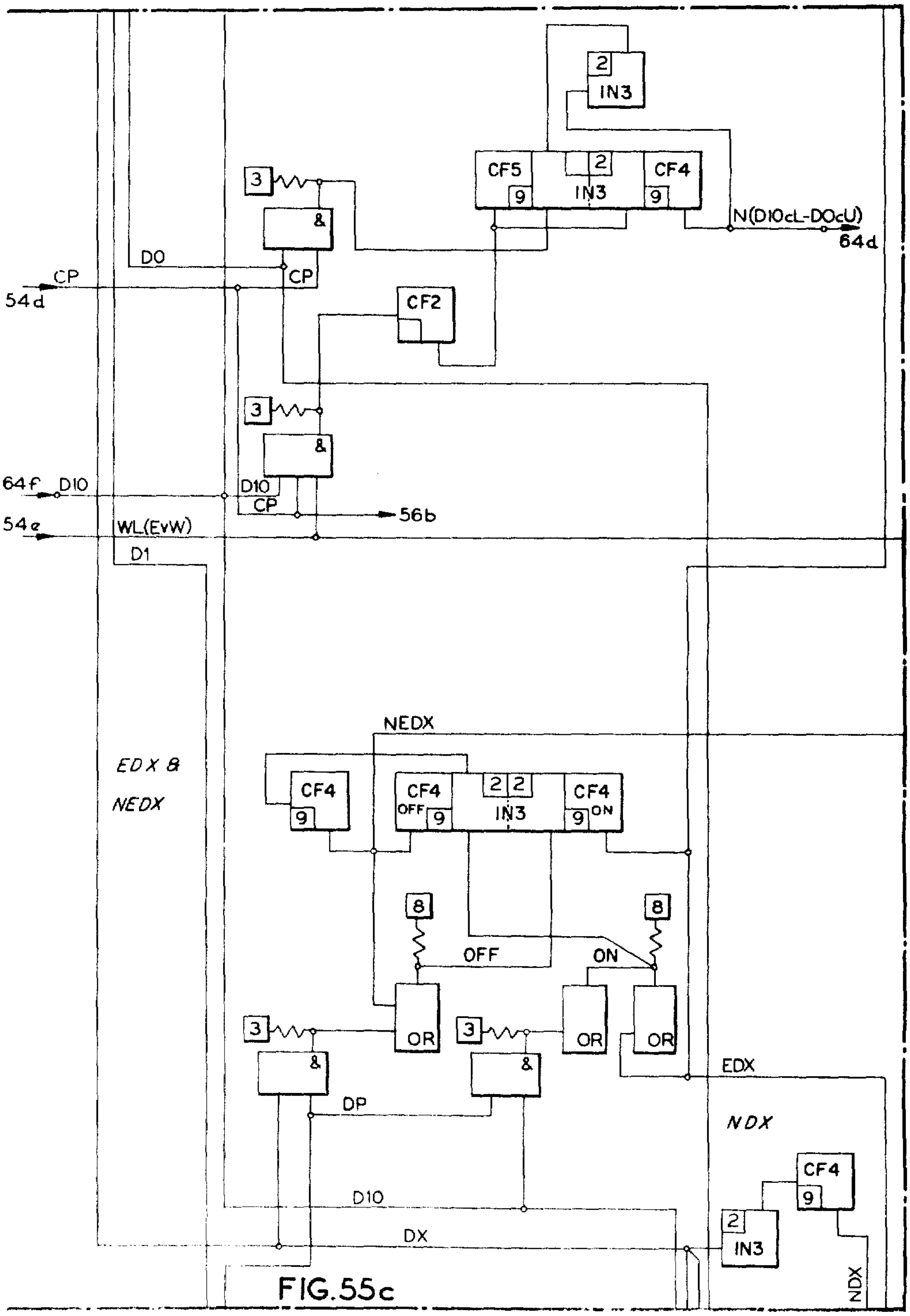

2
IN3
CF5
9
2
IN3
CF4
9
N(D10cL-D0cU)
64d
3
&
D0
CP
CP
54d
CF2
3
&
64f
D10
D10
CP
56b
54e
WL(EvW)
D1
NEDX
EDX &
NEDX
CF4
9
CF4
OFF
9
2 2
IN3
CF4
9
ON
8
8
OFF
ON
3
OR
3
OR
OR
&
&
EDX
DP
NDX
CF4
9
D10
2
DX
IN3
NDX

FIG.55c

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 33

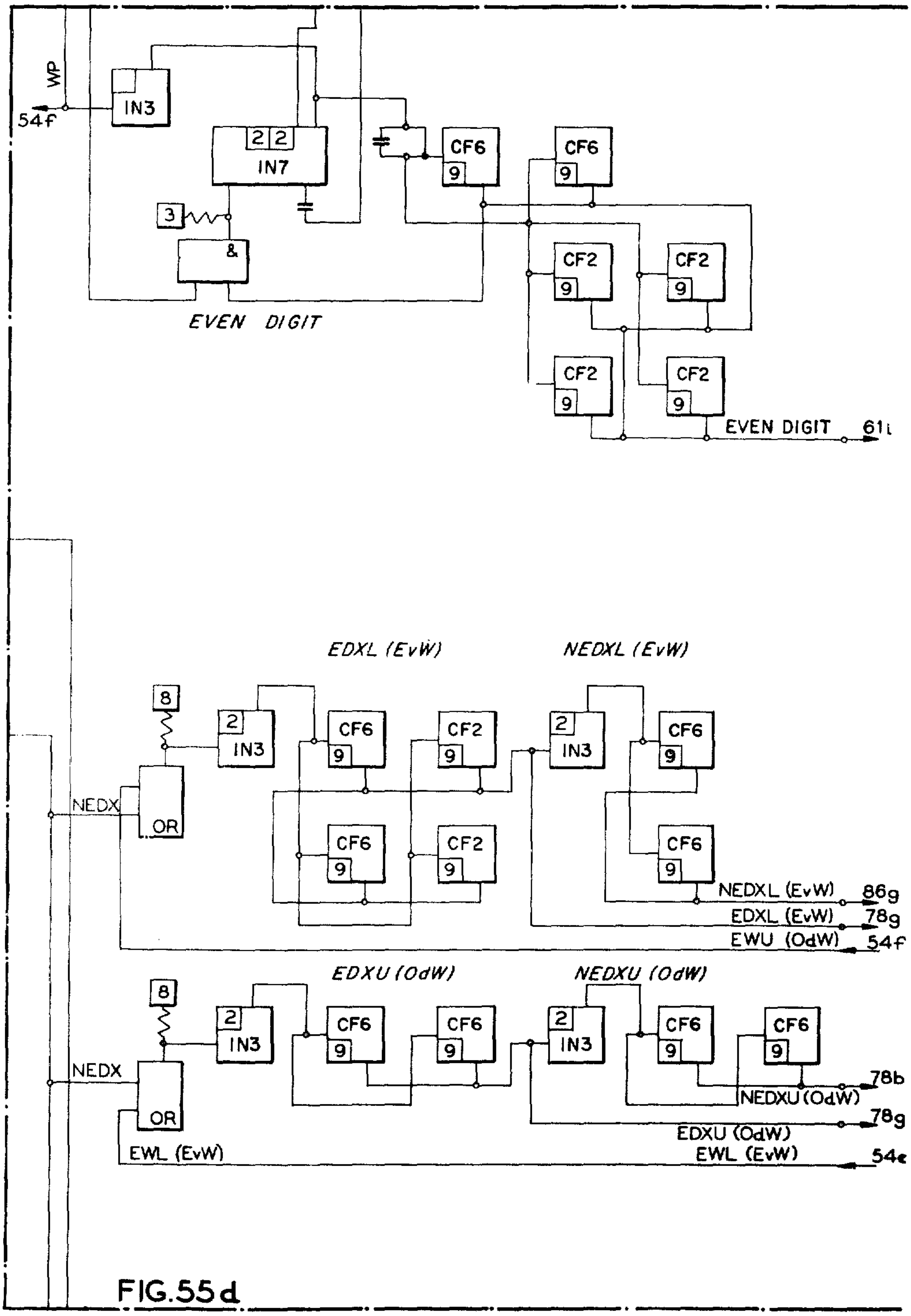


FIG.55d

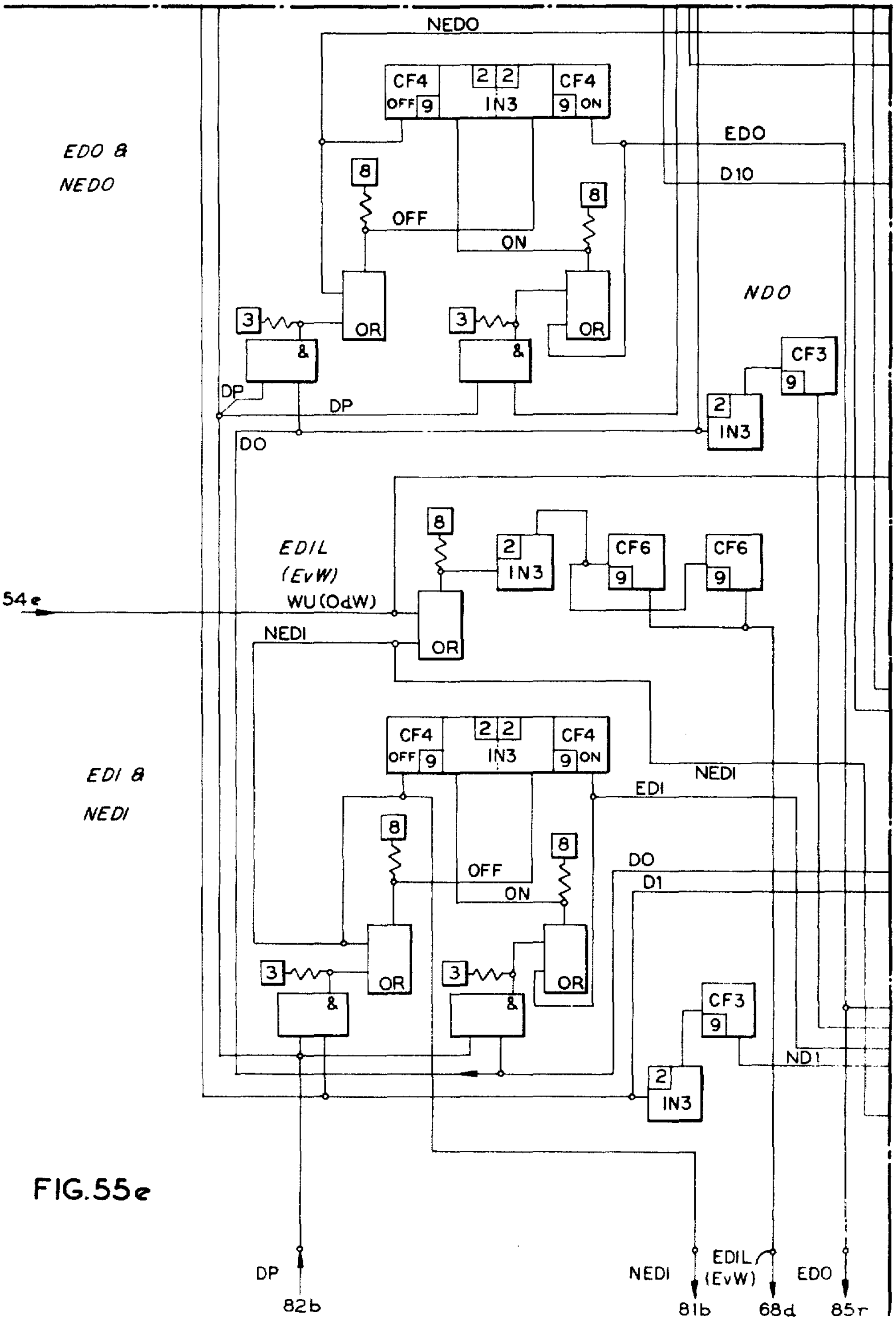

NEDO
CF4
OFF
9
2 2
IN3
CF4
9
ON
EDO
D10
EDO &
NEDO
8
OFF
8
ON
NDO
3
OR
3
OR
&
&
CF3
9
DP
DP
2
IN3
DO
8
2
IN3
CF6
9
CF6
9
EDIL
(EvW)
54e
WU (OdW)
NEDI
OR
CF4
OFF
9
2 2
IN3
CF4
9
ON
NEDI
EDI
EDI &
NEDI
8
8
OFF
ON
DO
D1
3
OR
3
OR
&
&
CF3
9
ND1
2
IN3
DP
82b
NEDI
81b
EDIL
(EvW)
68d
EDO
85r

FIG.55e

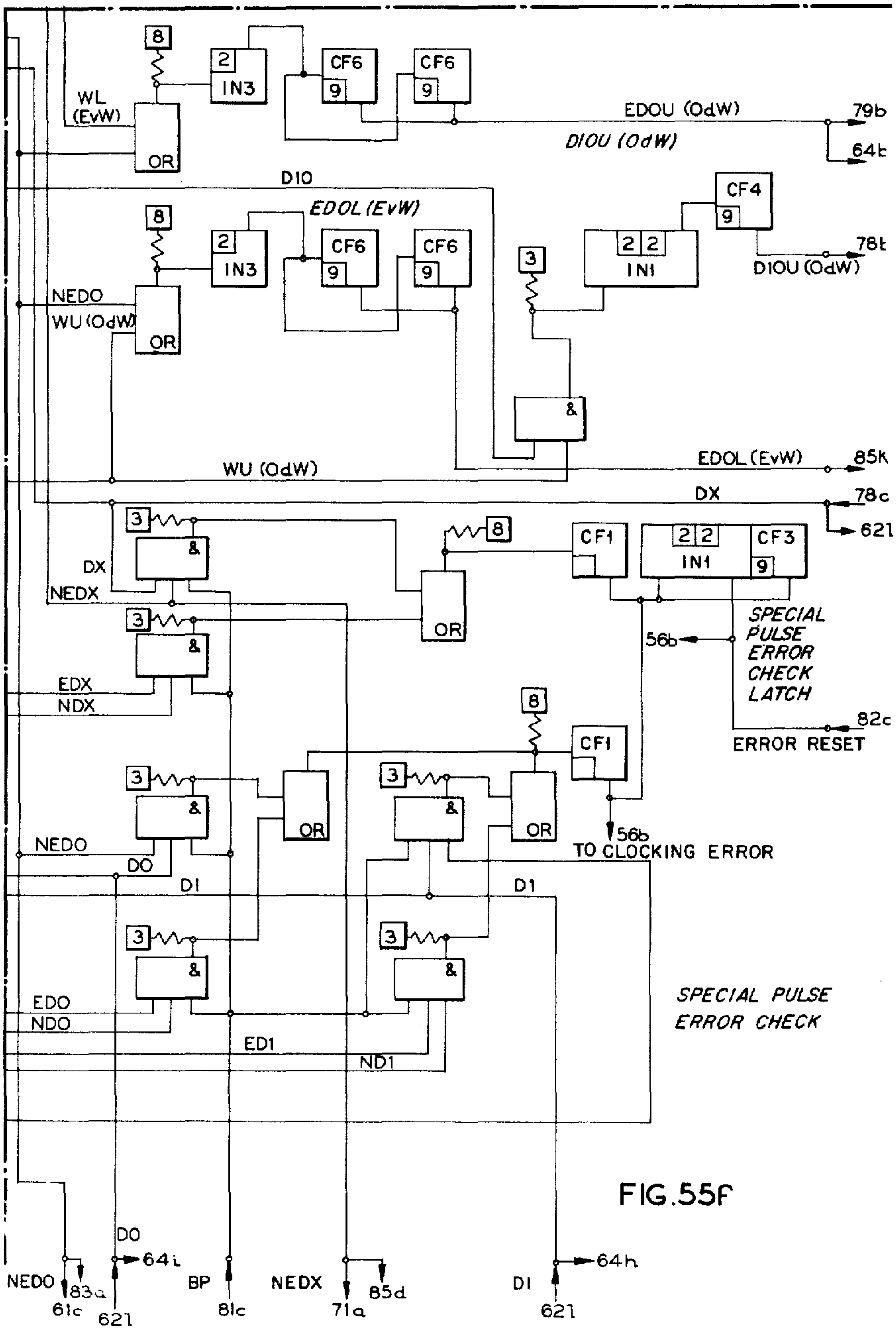

8
2
IN3
CF6
9
CF6
9
WL
(EvW)
OR
EDOU (OdW)
79b
DIOU (OdW)
64t
D10
8
EDOL (EvW)
2
IN3
CF6
9
CF6
9
CF4
9
3
2 2
IN1
78t
D10U (OdW)
NEDO
WU (OdW)
OR
&
EDOL (EvW)
85K
WU (OdW)
DX
78c
621
3
&
DX
NEDX
8
OR
CF1
2 2
IN1
CF3
9
3
&
SPECIAL
PULSE
ERROR
CHECK
LATCH
56b
EDX
NDX
8
CF1
82c
ERROR RESET
3
&
OR
3
&
OR
56b
TO CLOCKING ERROR
NEDO
DO
D1
D1
3
&
3
&
SPECIAL PULSE
ERROR CHECK
EDO
NDO
ED1
ND1
DO
64i
NEDO
83a
61c
621
BP
81c
NEDX
85d
71a
D1
64h
621

FIG.55F

DATA STORAGE AND PROCESSING MACHINE

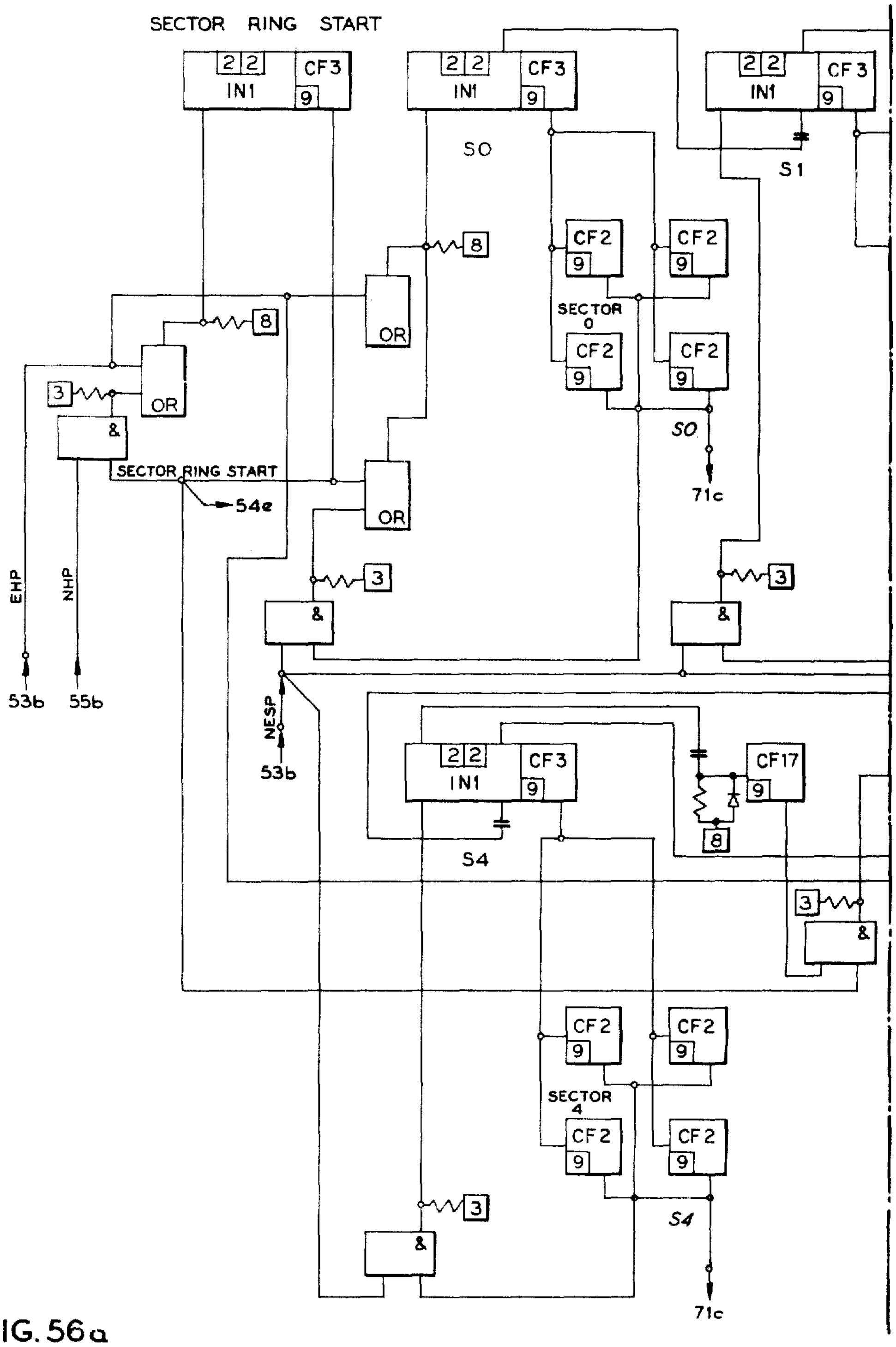


FIG. 56a

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 37

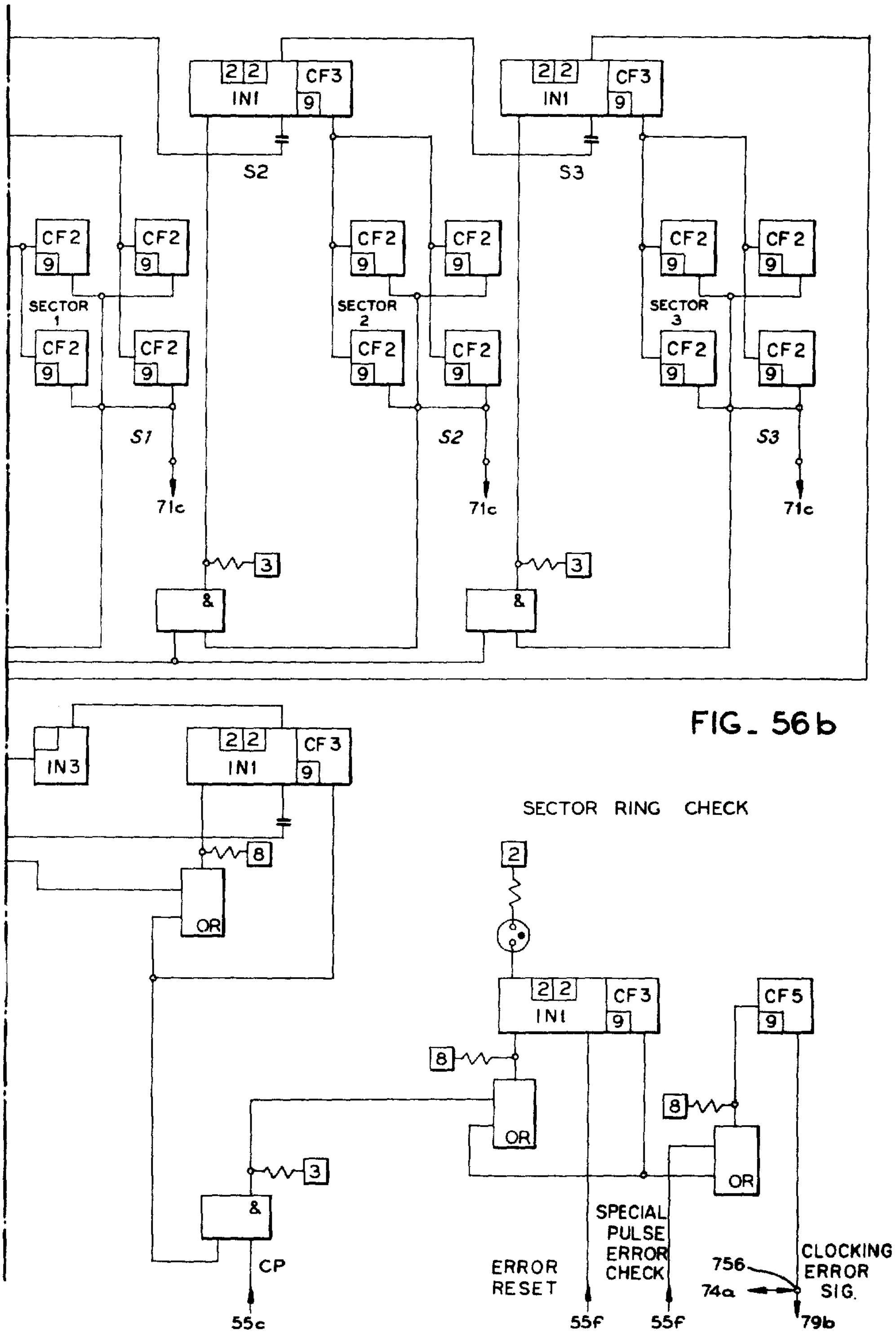


FIG. 56b

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 38

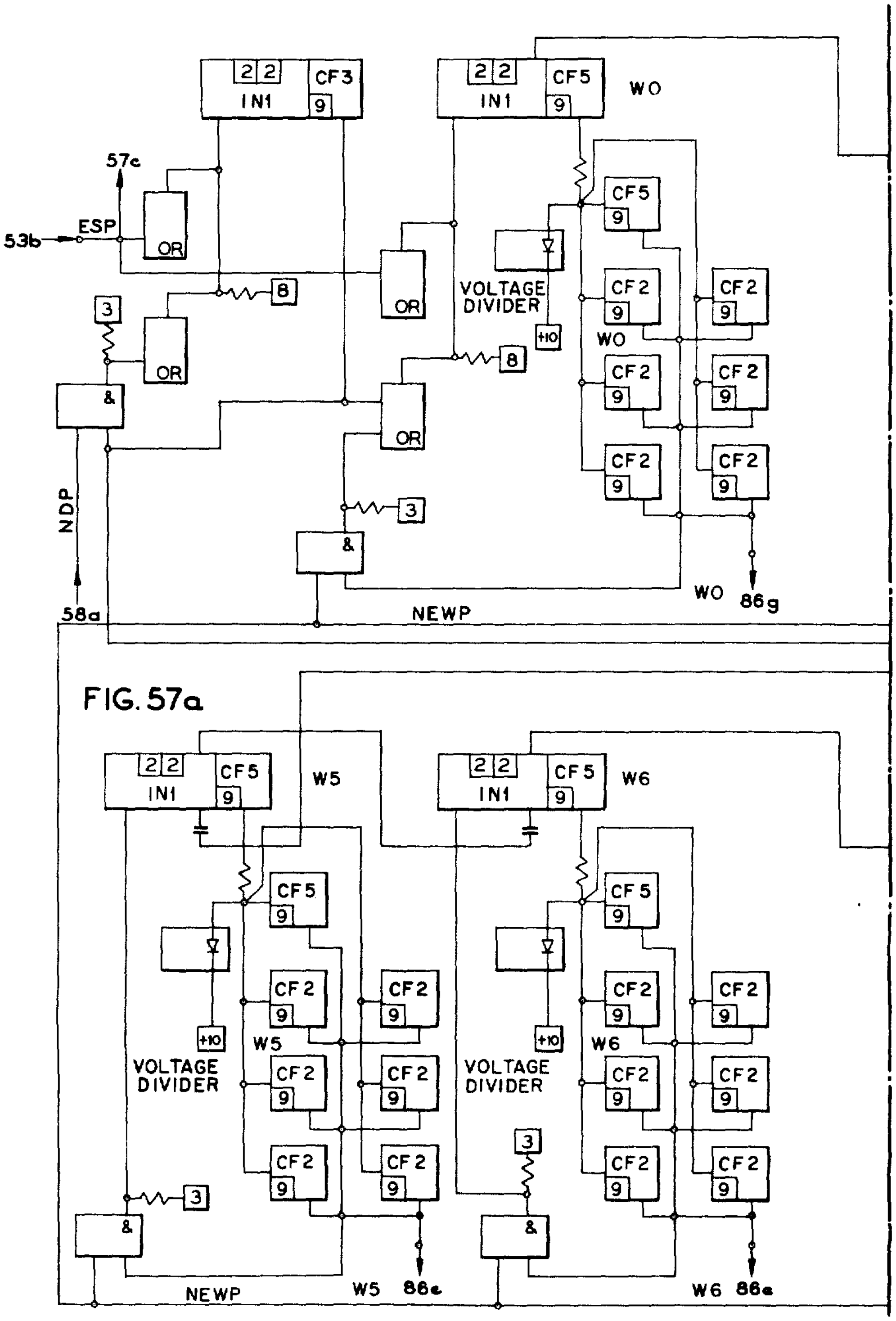


FIG. 57a

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 39

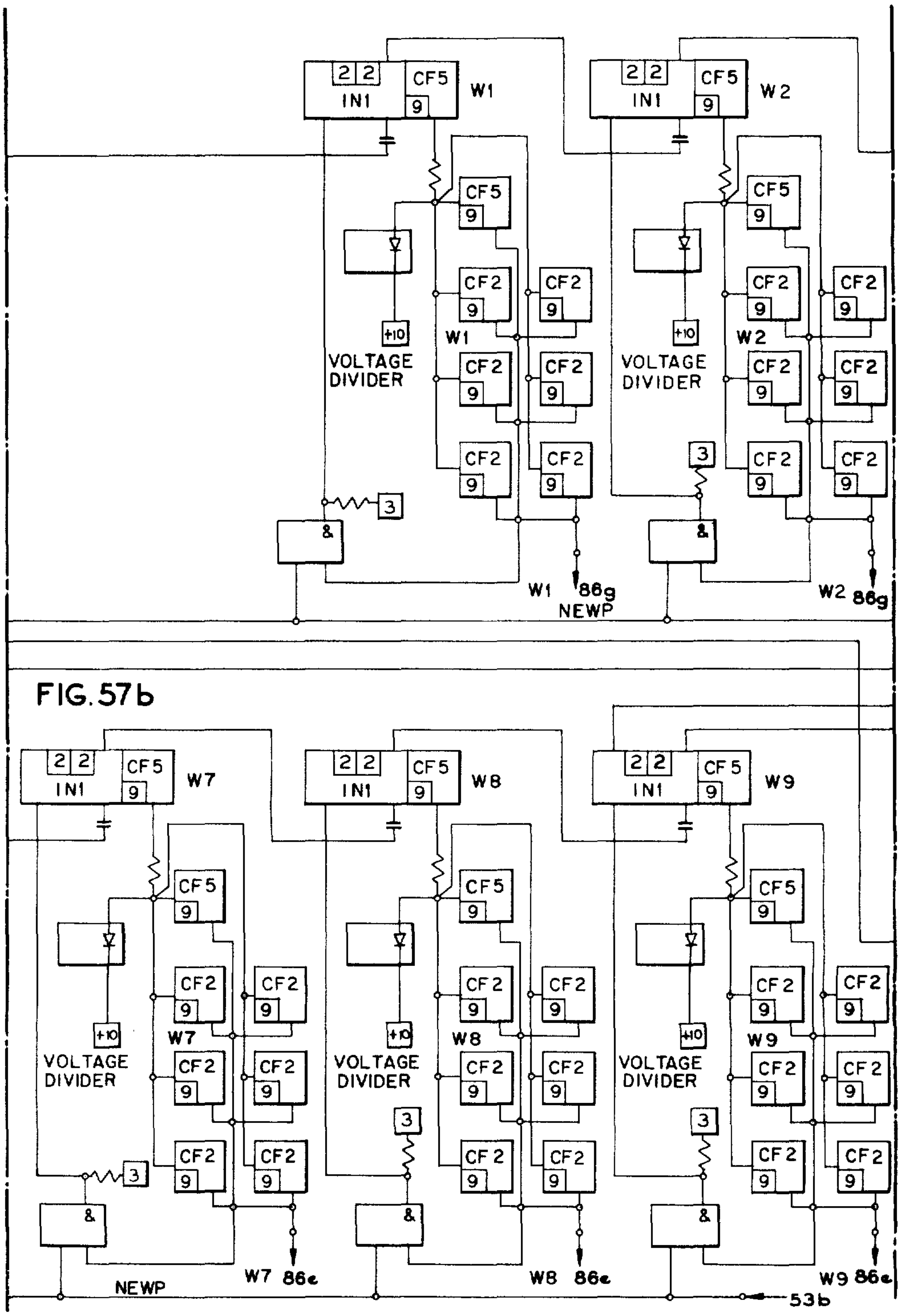


FIG. 57b

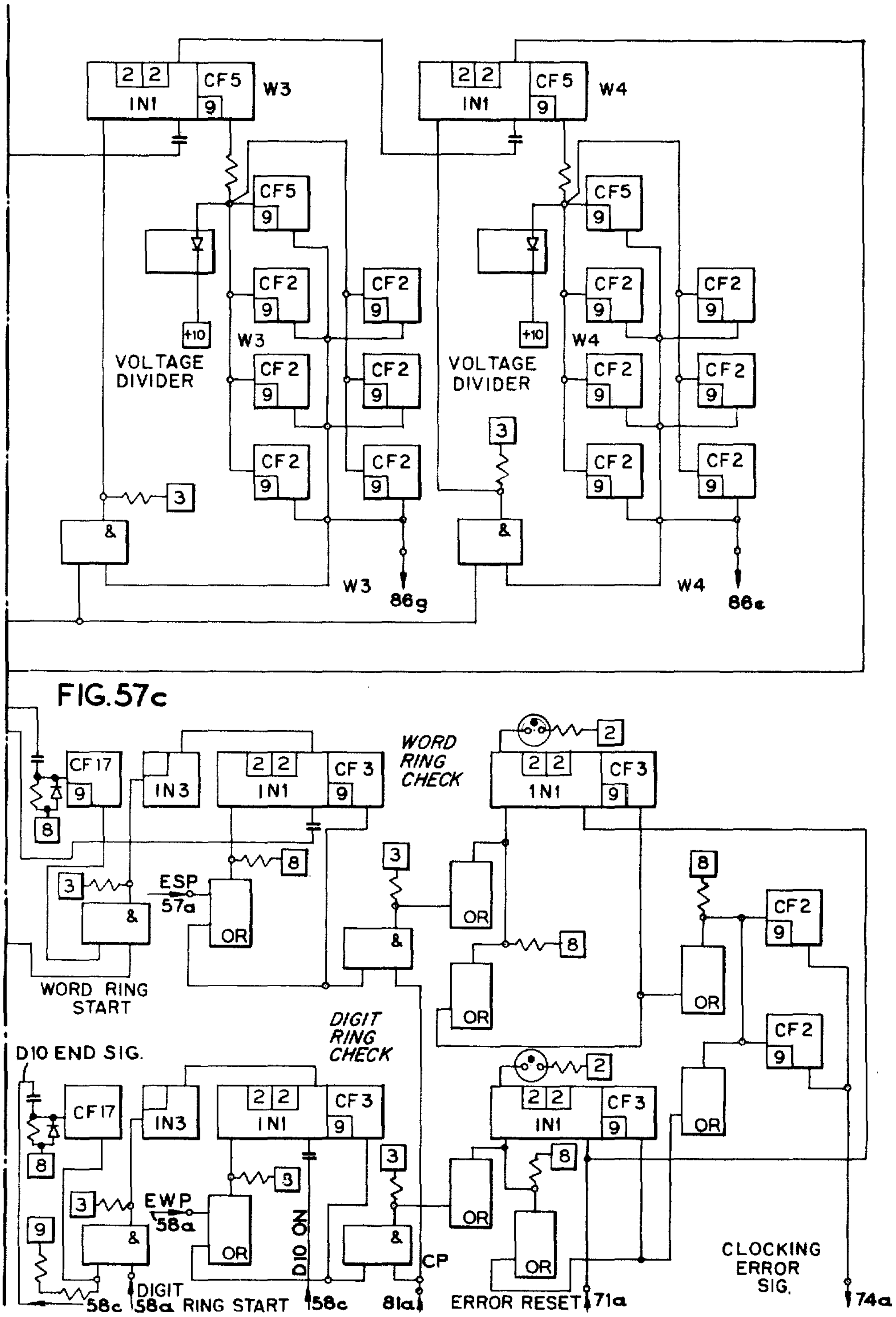


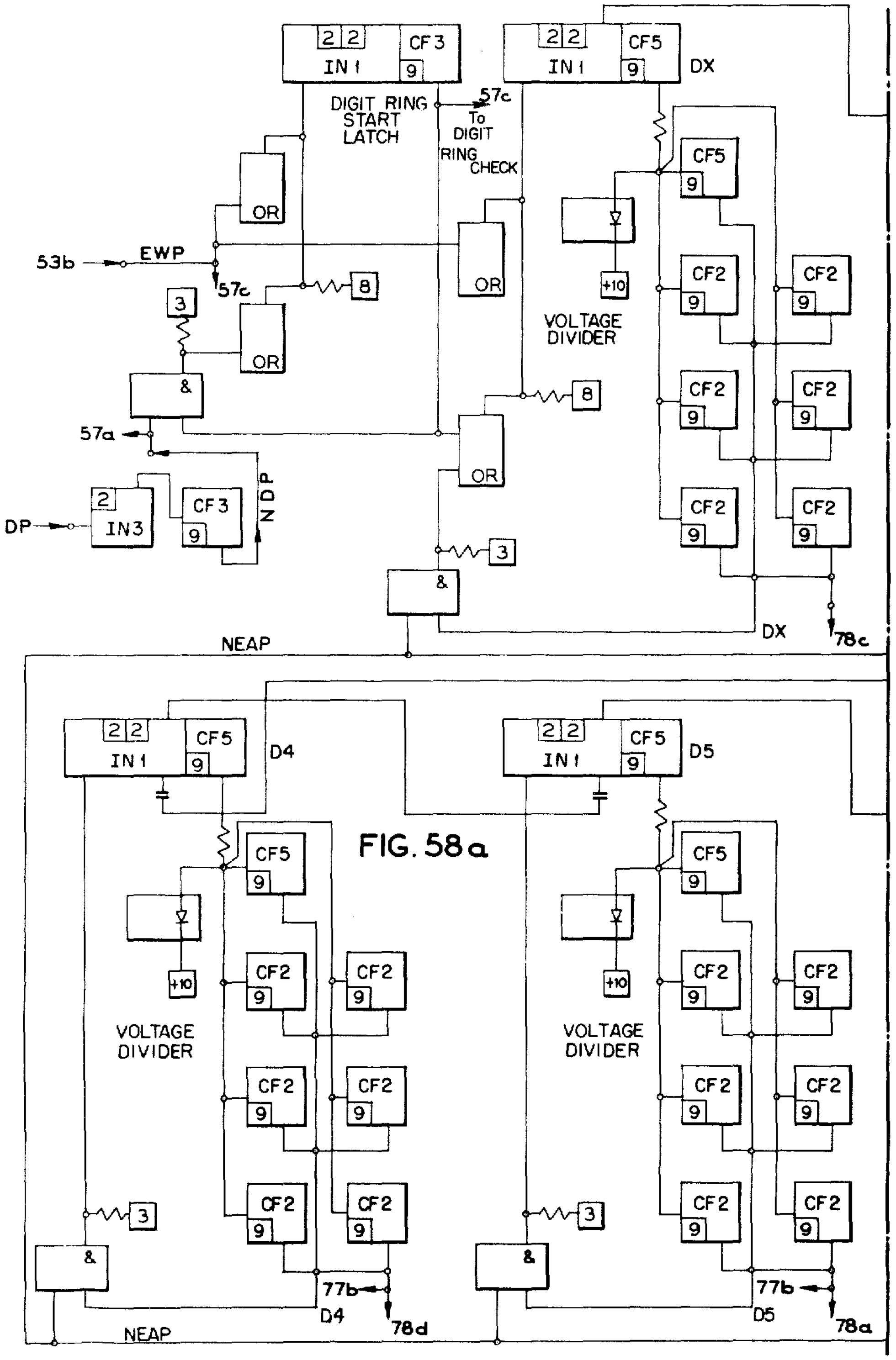
FIG. 57c

2 2
IN1
CF5
9
W3
W4
CF5
CF2
+10
VOLTAGE DIVIDER
3
&
86g
86e
CF17
8
IN3
CF3
WORD RING CHECK
ESP
57a
OR
WORD RING START
DIGIT RING CHECK
D10 END SIG.
EWP
58a
D10 ON
CP
58c
DIGIT
58a RING START
81a
ERROR RESET
71a
CLOCKING ERROR SIG.
74a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 41



Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 42

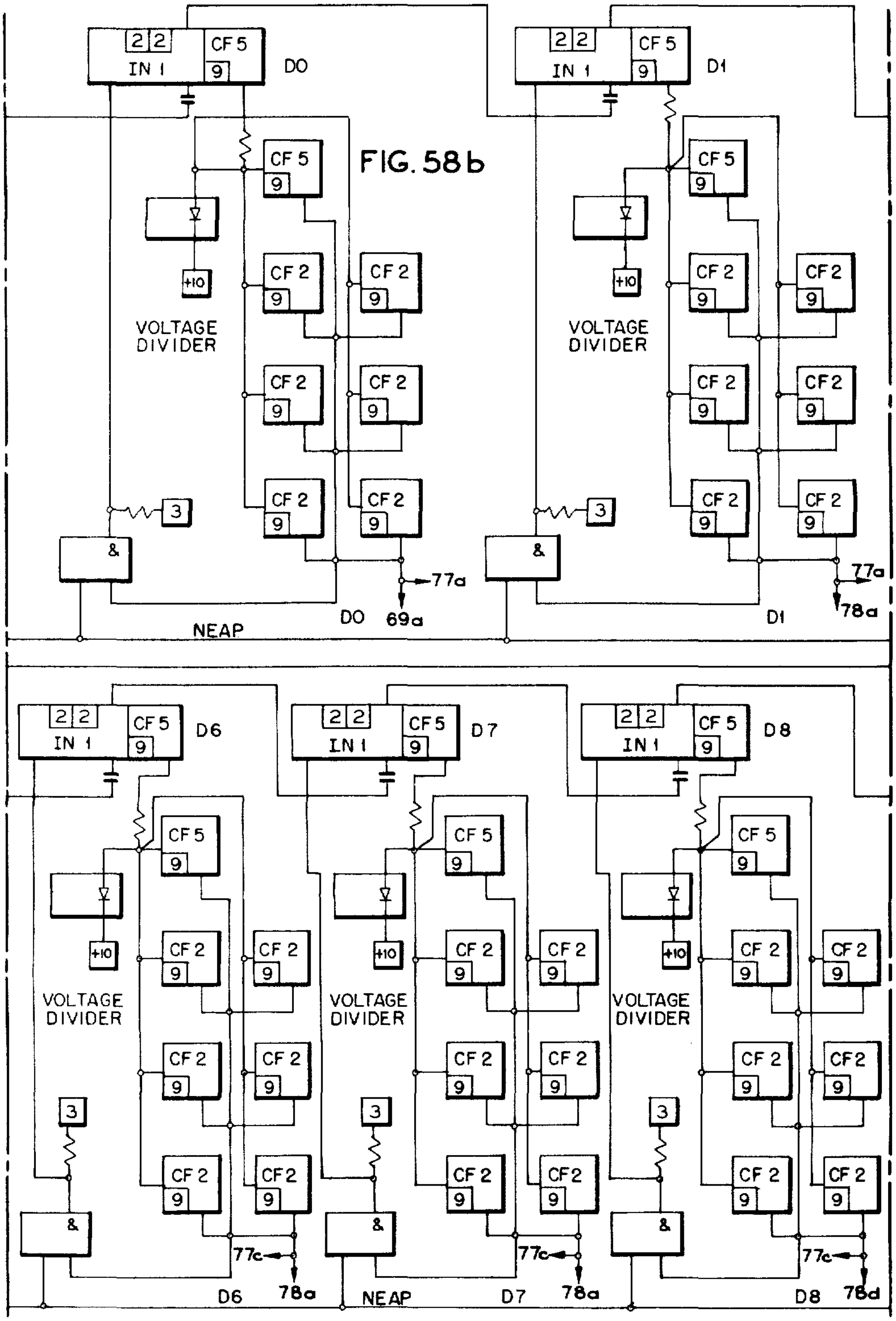


FIG. 58b

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 43

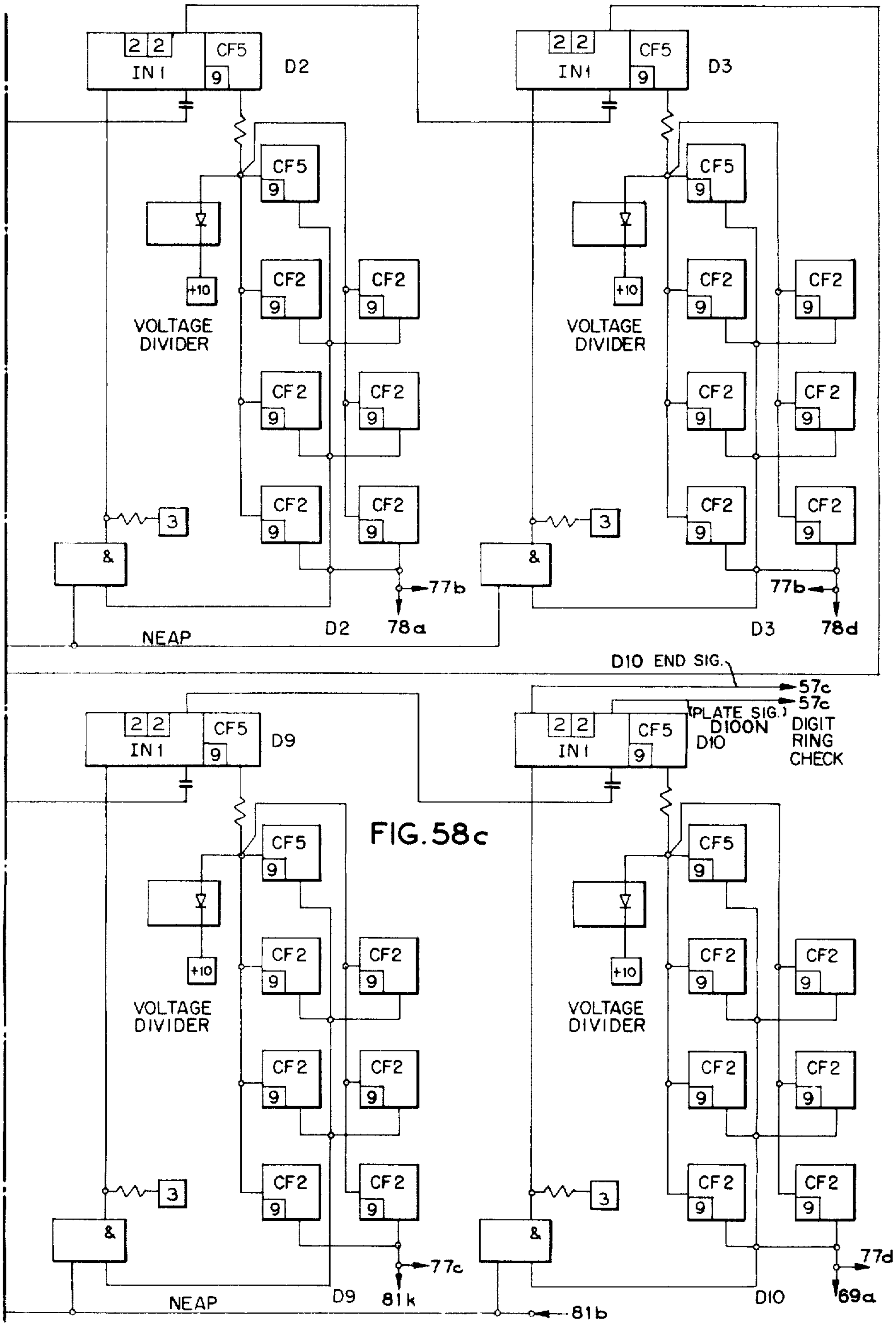


FIG. 58c

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 44

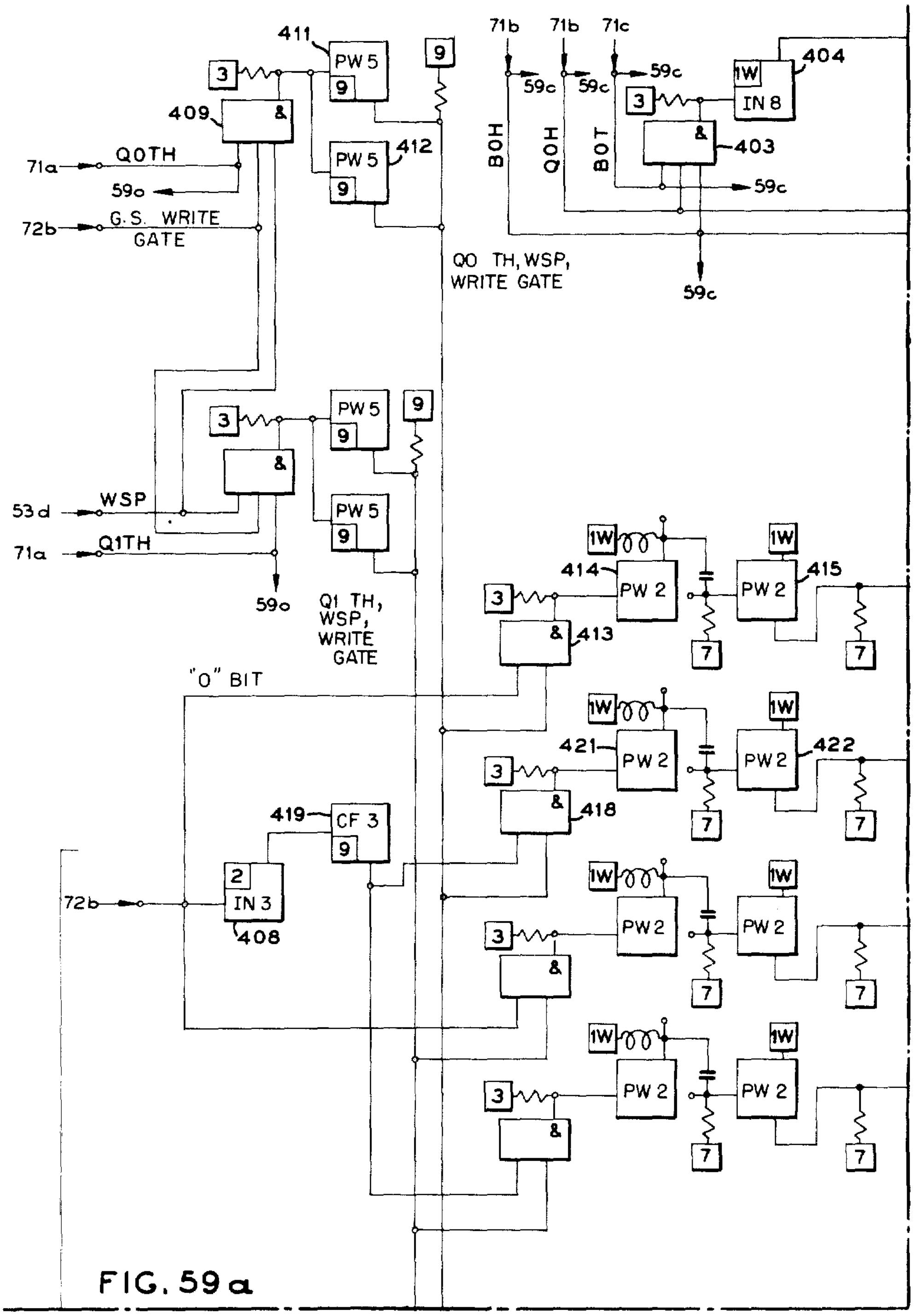


FIG. 59a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 45

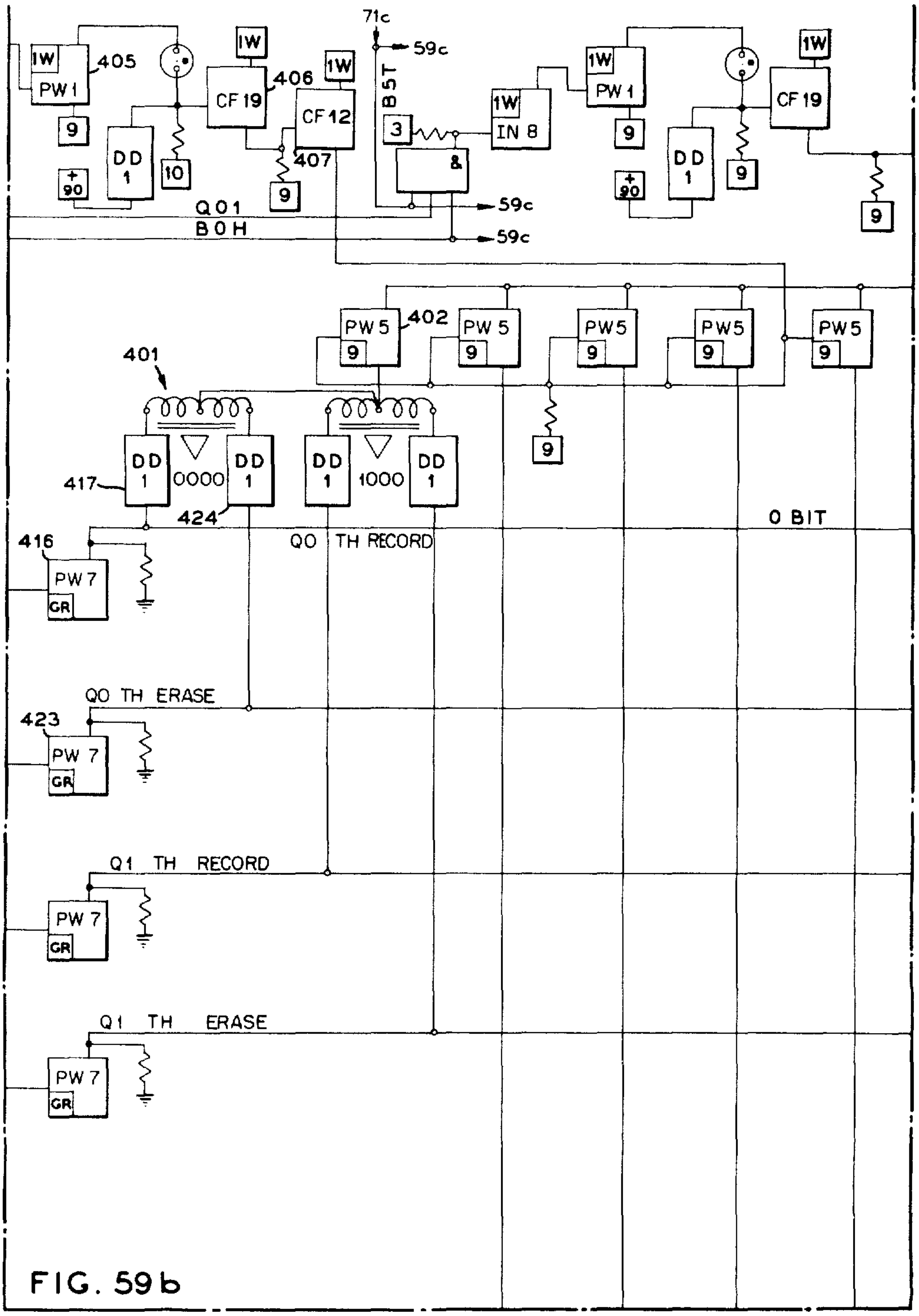


FIG. 59b

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 46

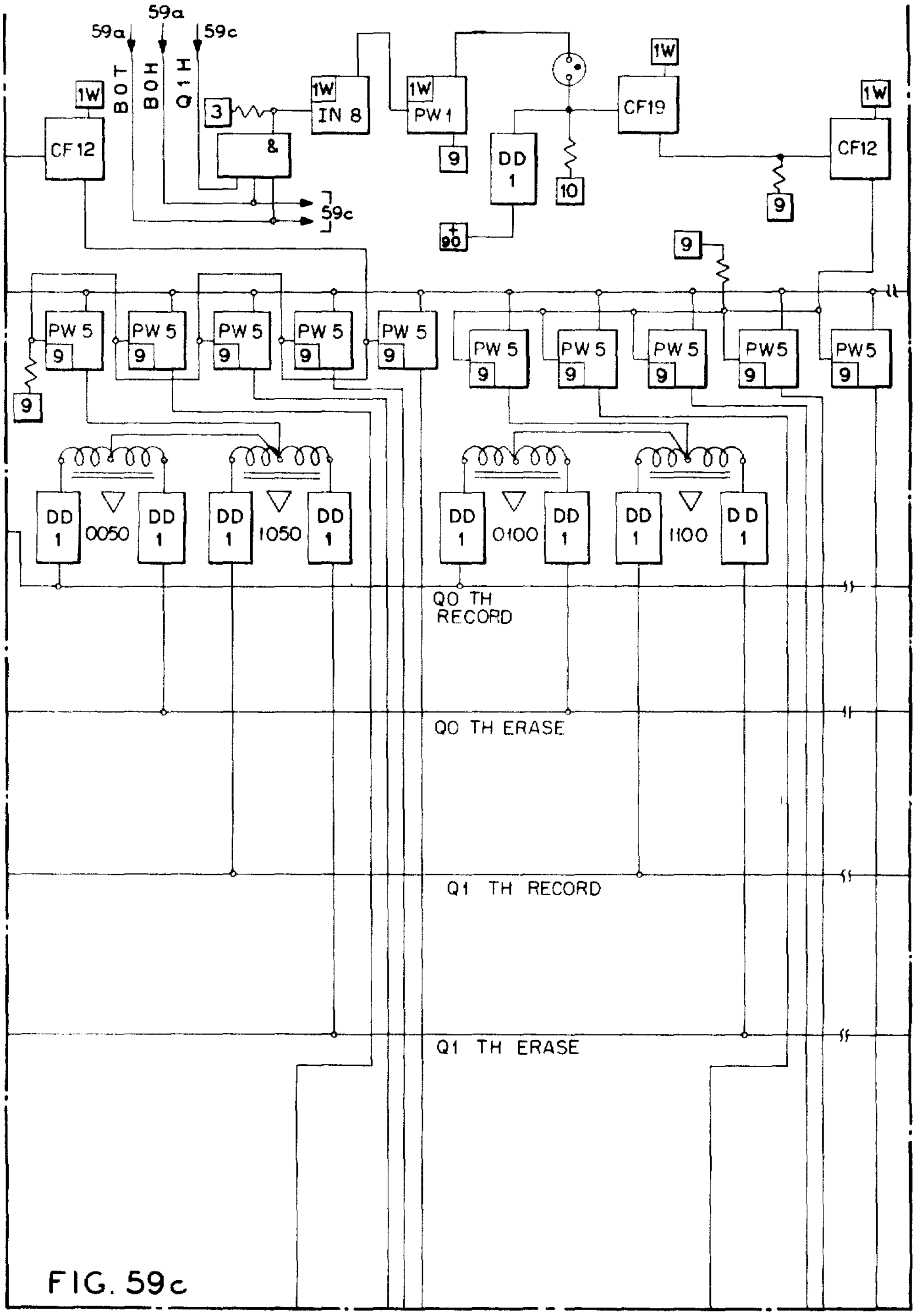


FIG. 59c

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 47

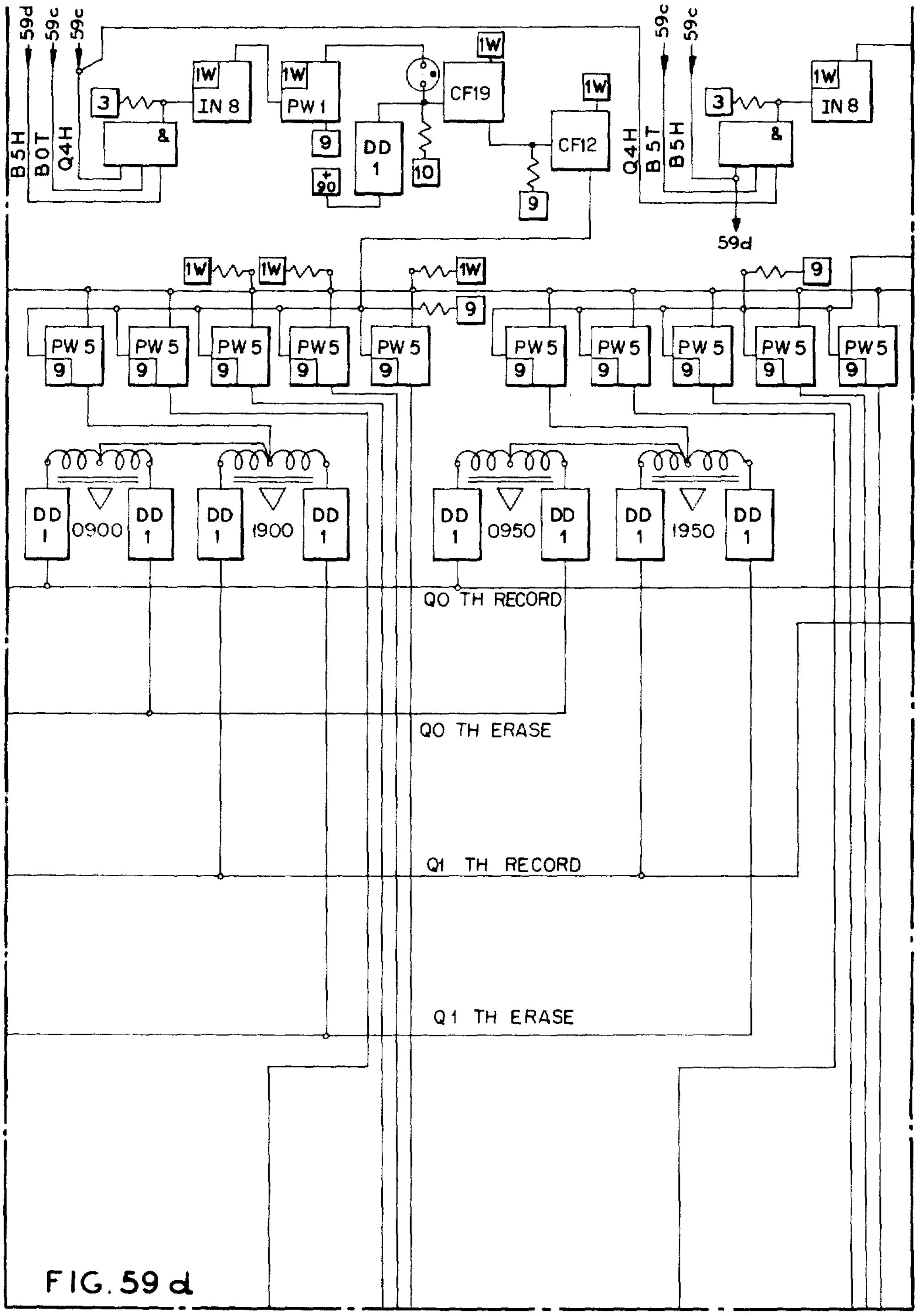


FIG. 59 d

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 43

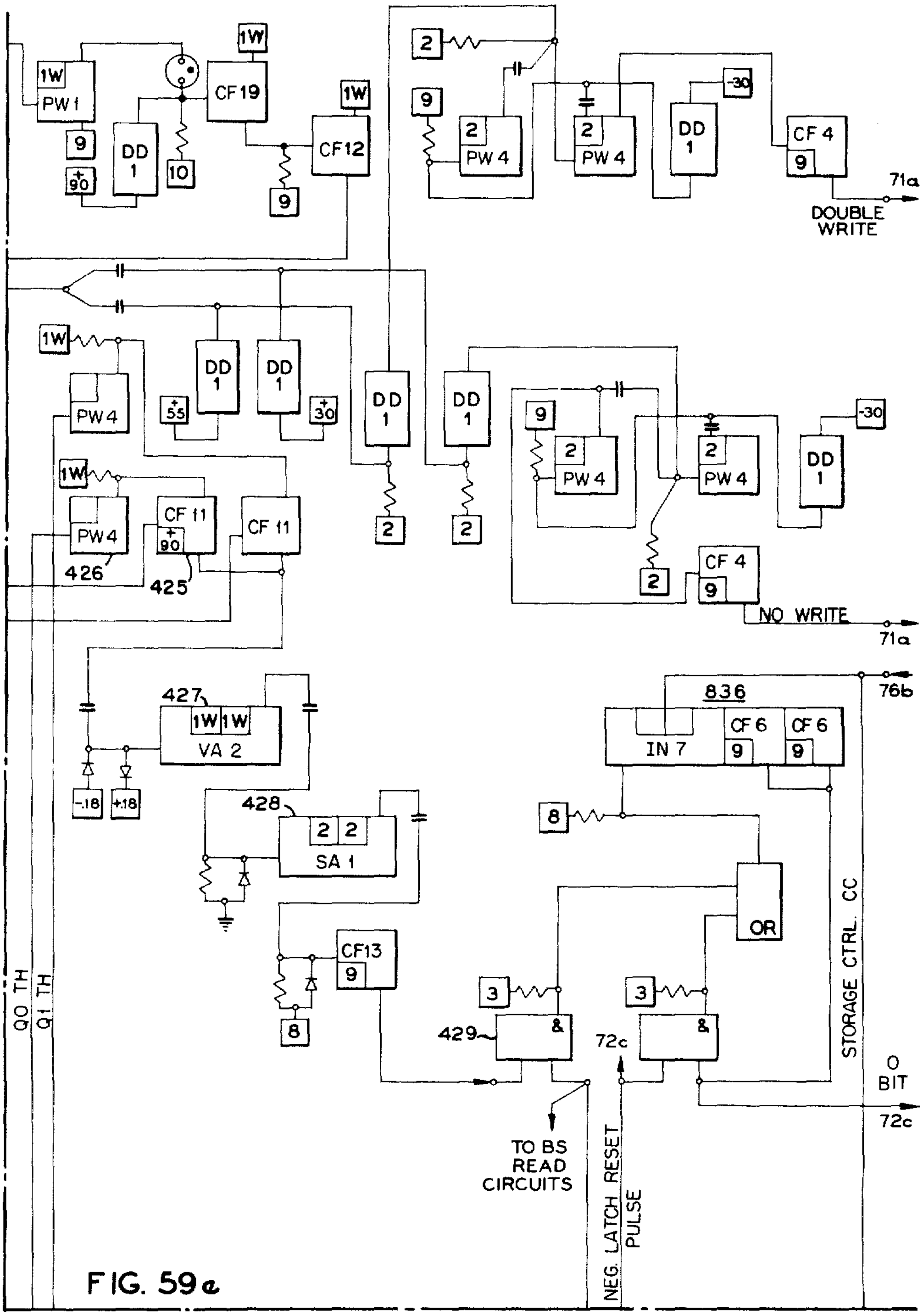


FIG. 59a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 49

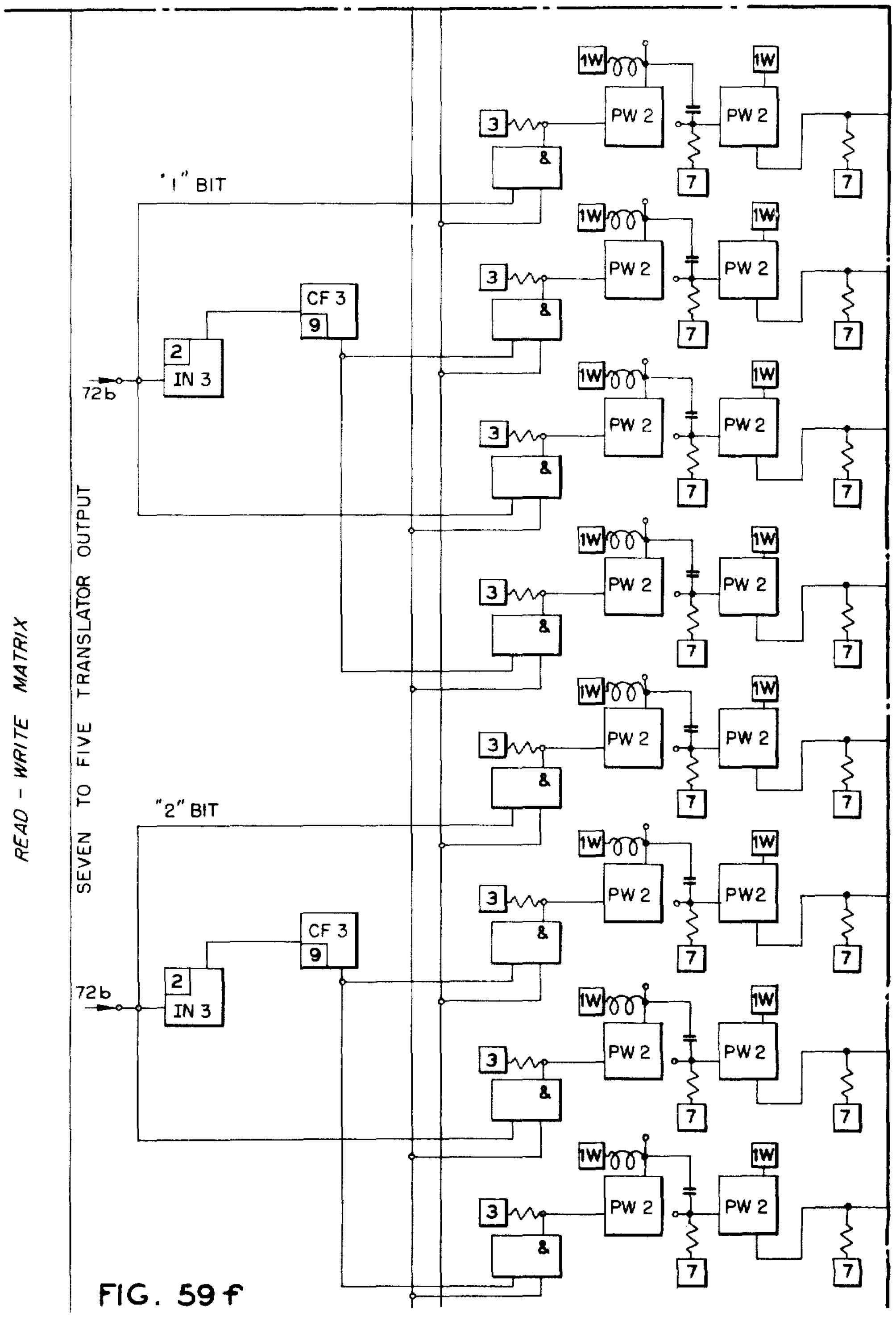


FIG. 59f

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 50

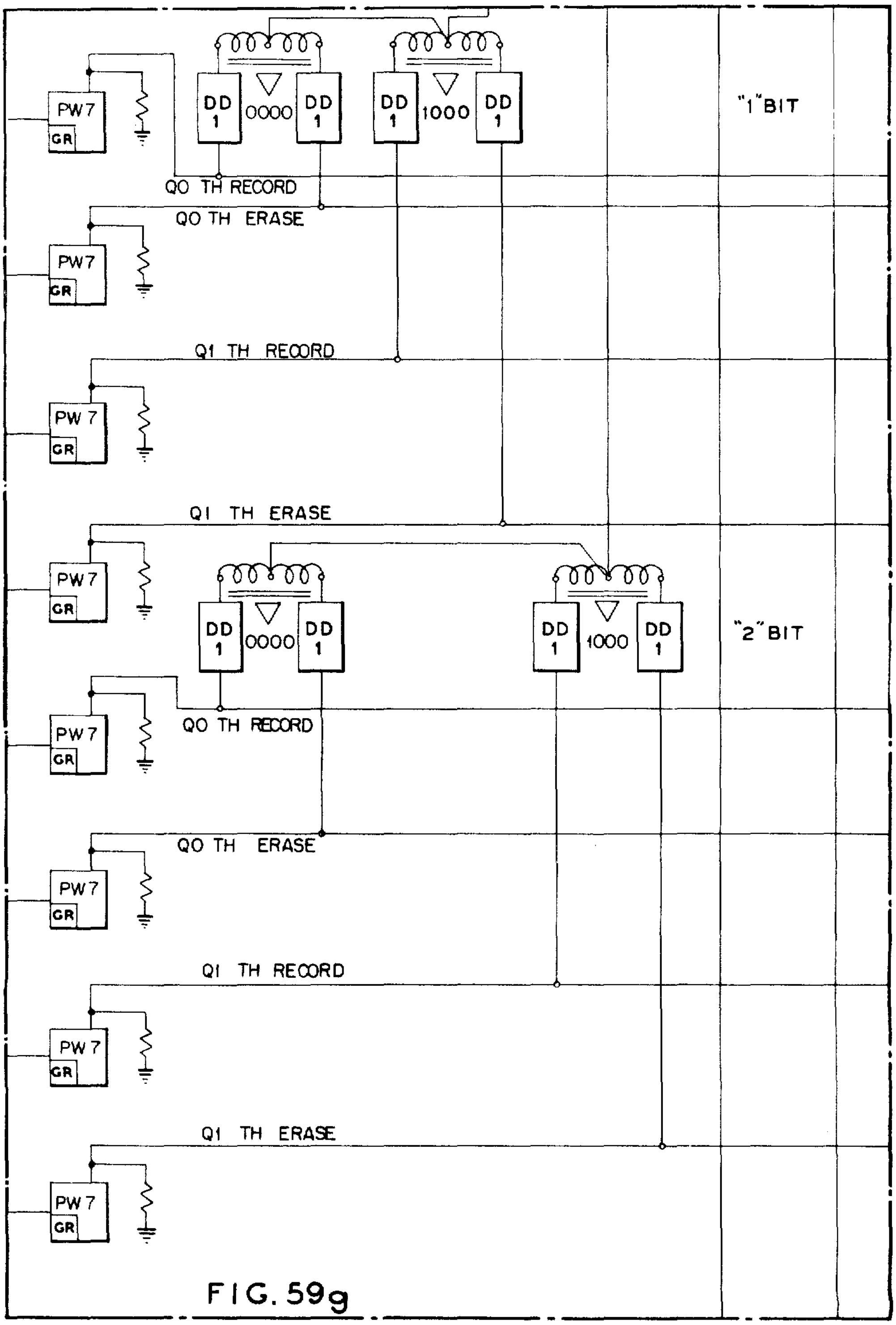


FIG. 59g

DATA STORAGE AND PROCESSING MACHINE

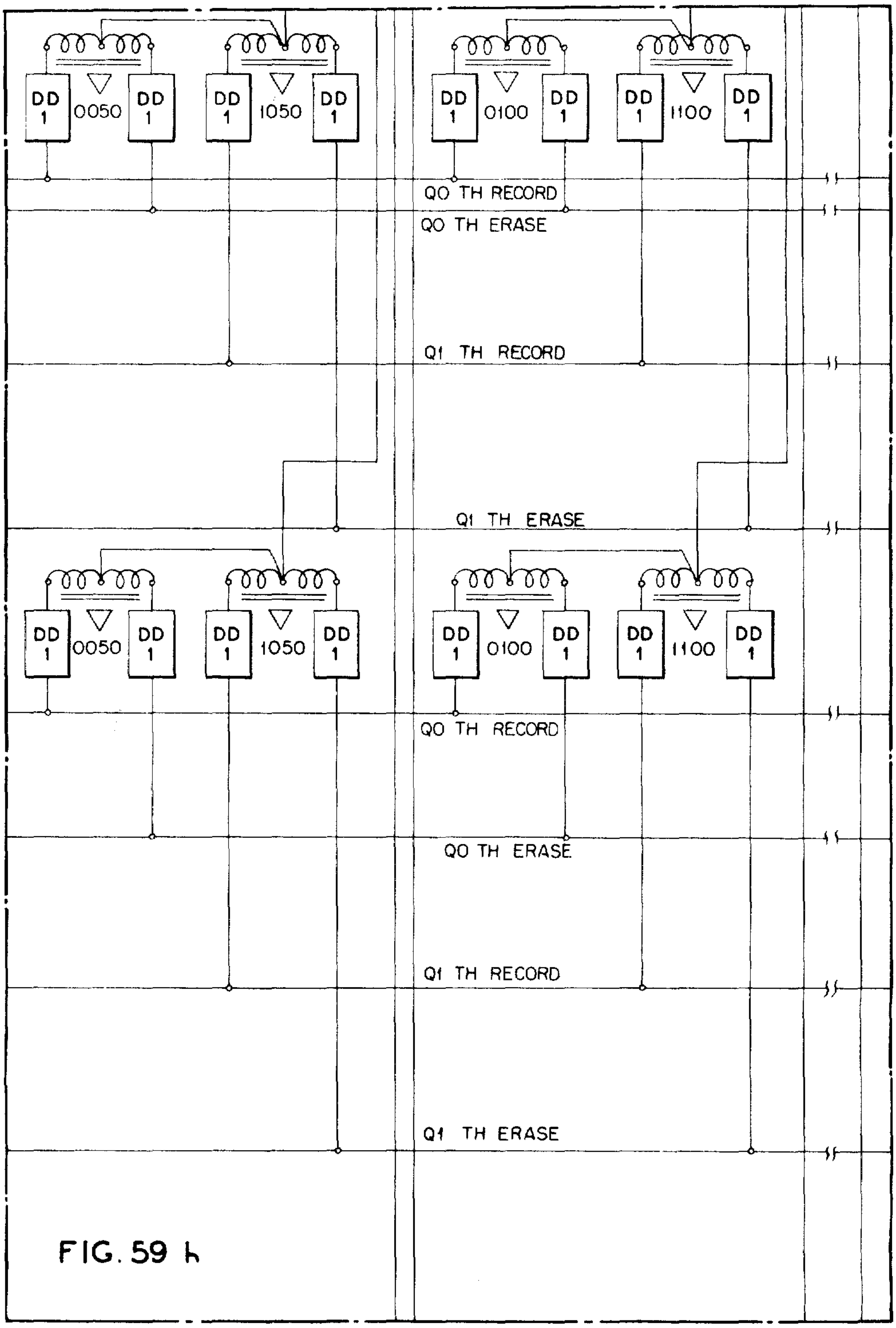


FIG. 59 h

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 52

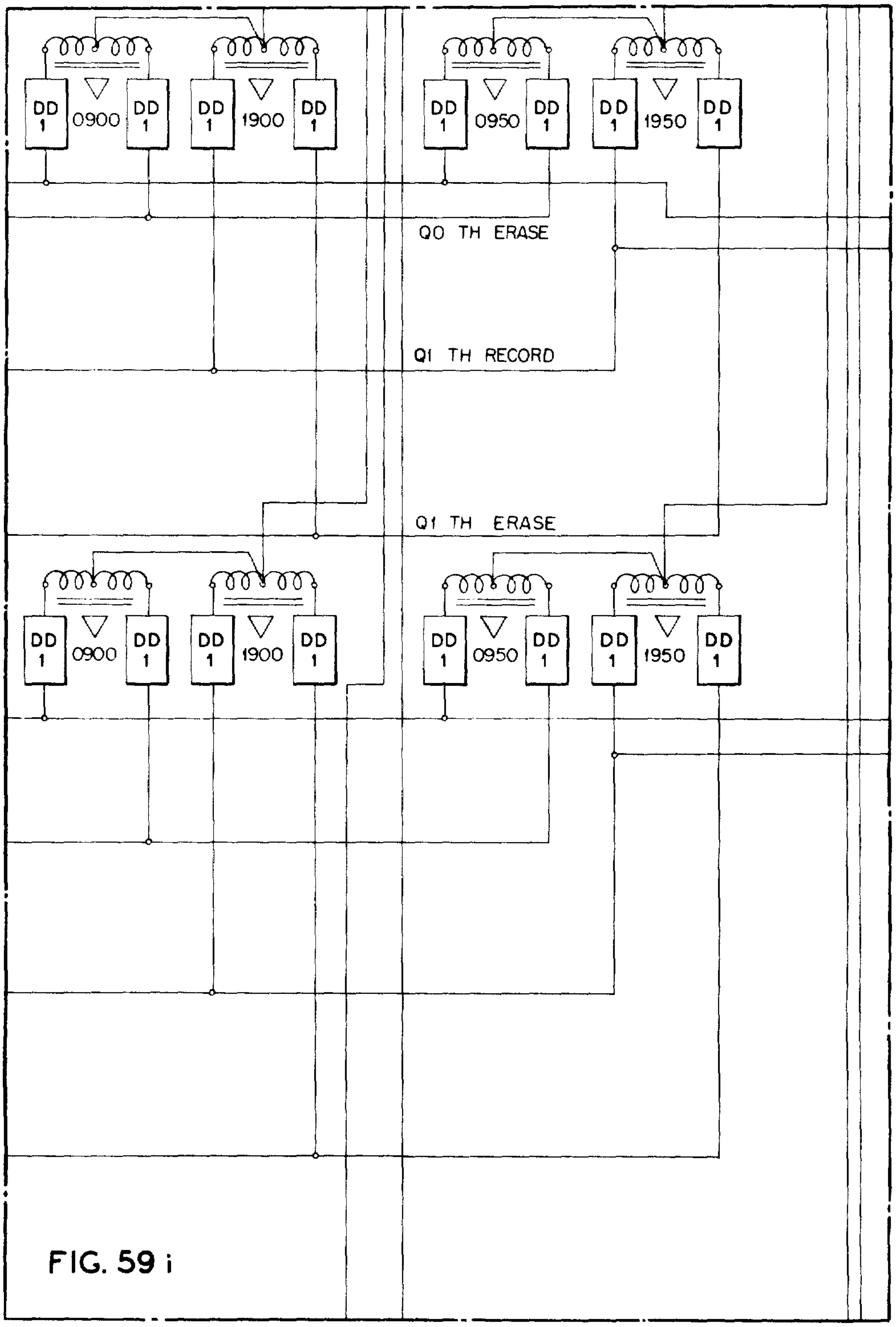


FIG. 59 i

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 53

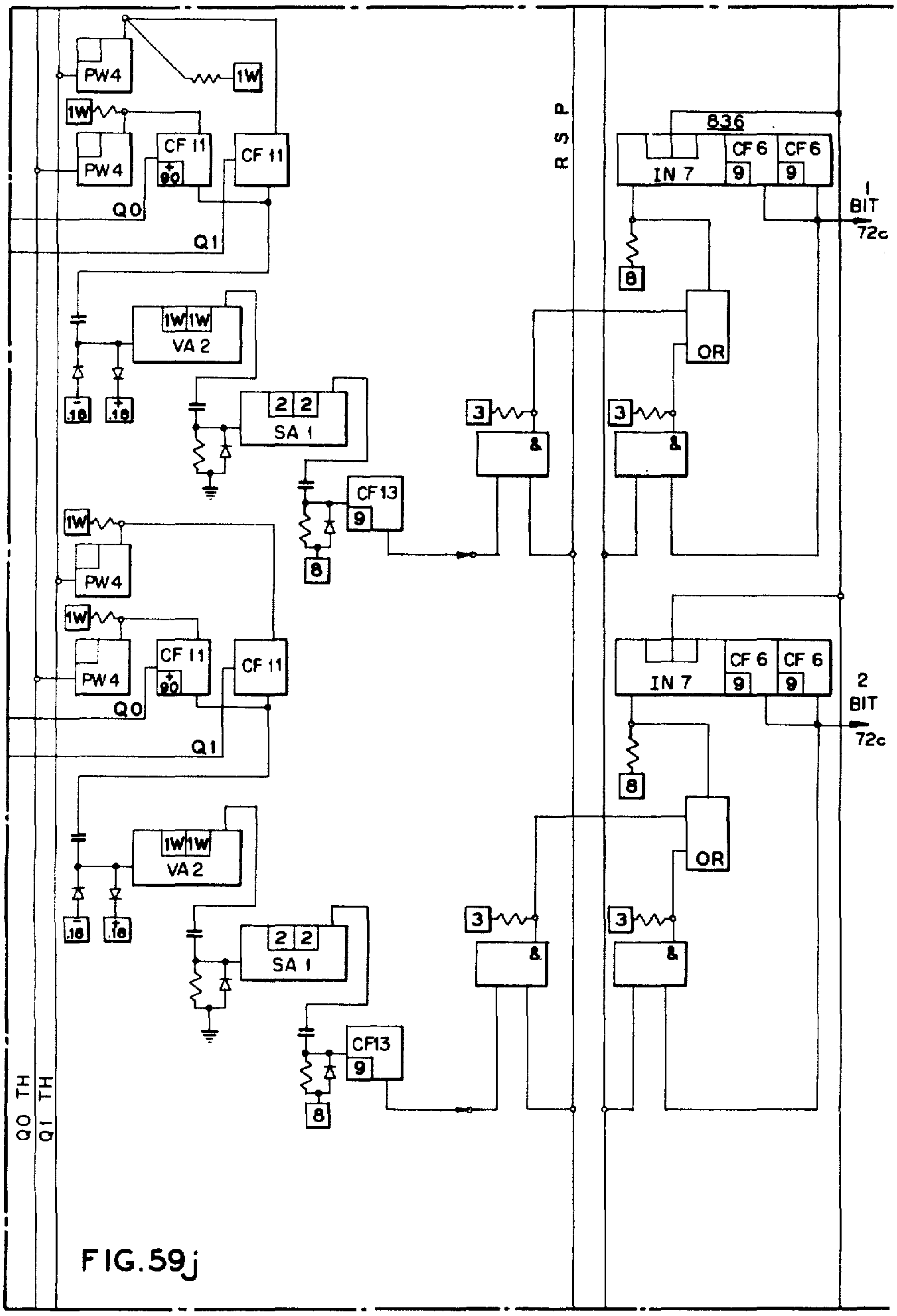


FIG.59j

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 54

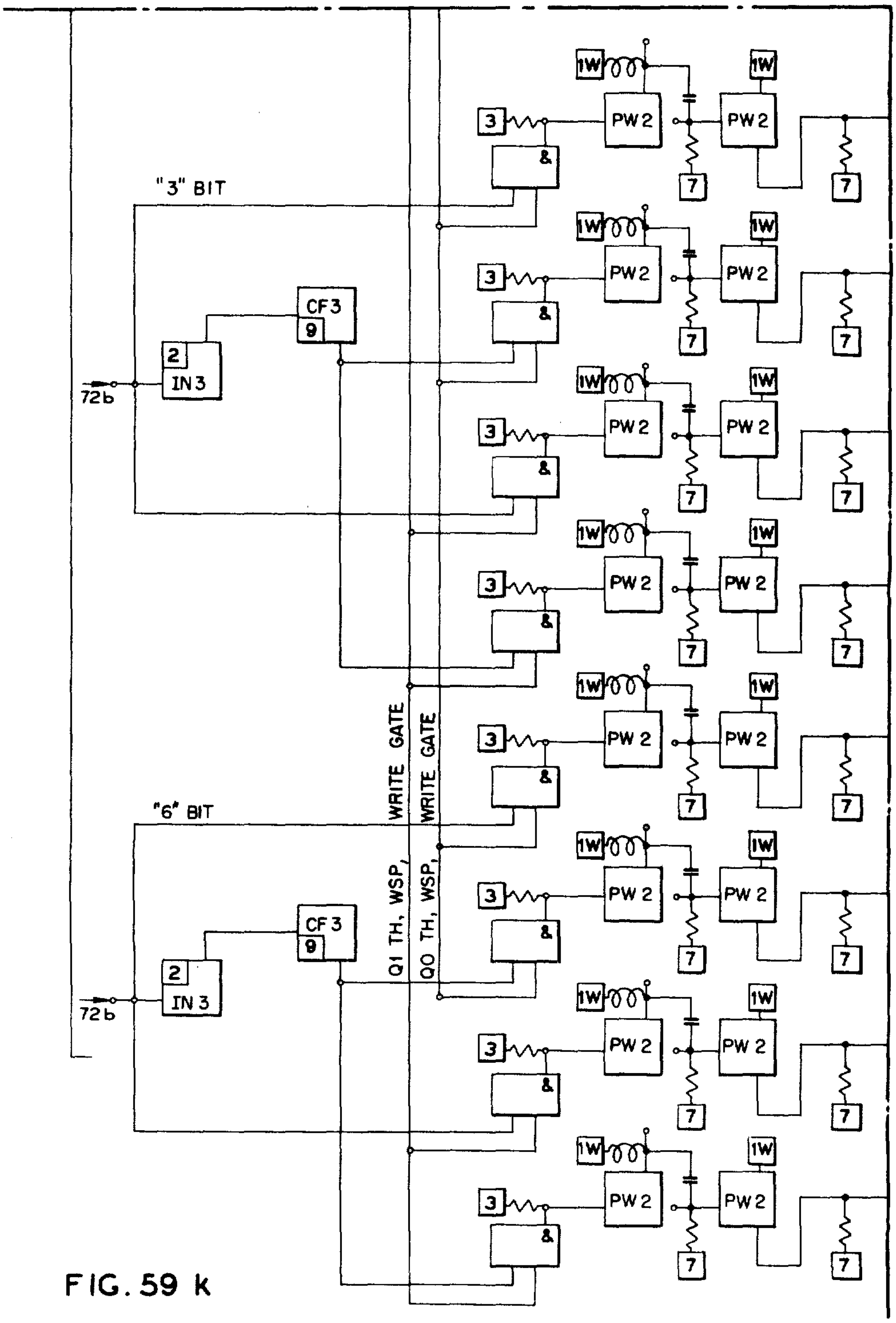


FIG. 59 k

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 55

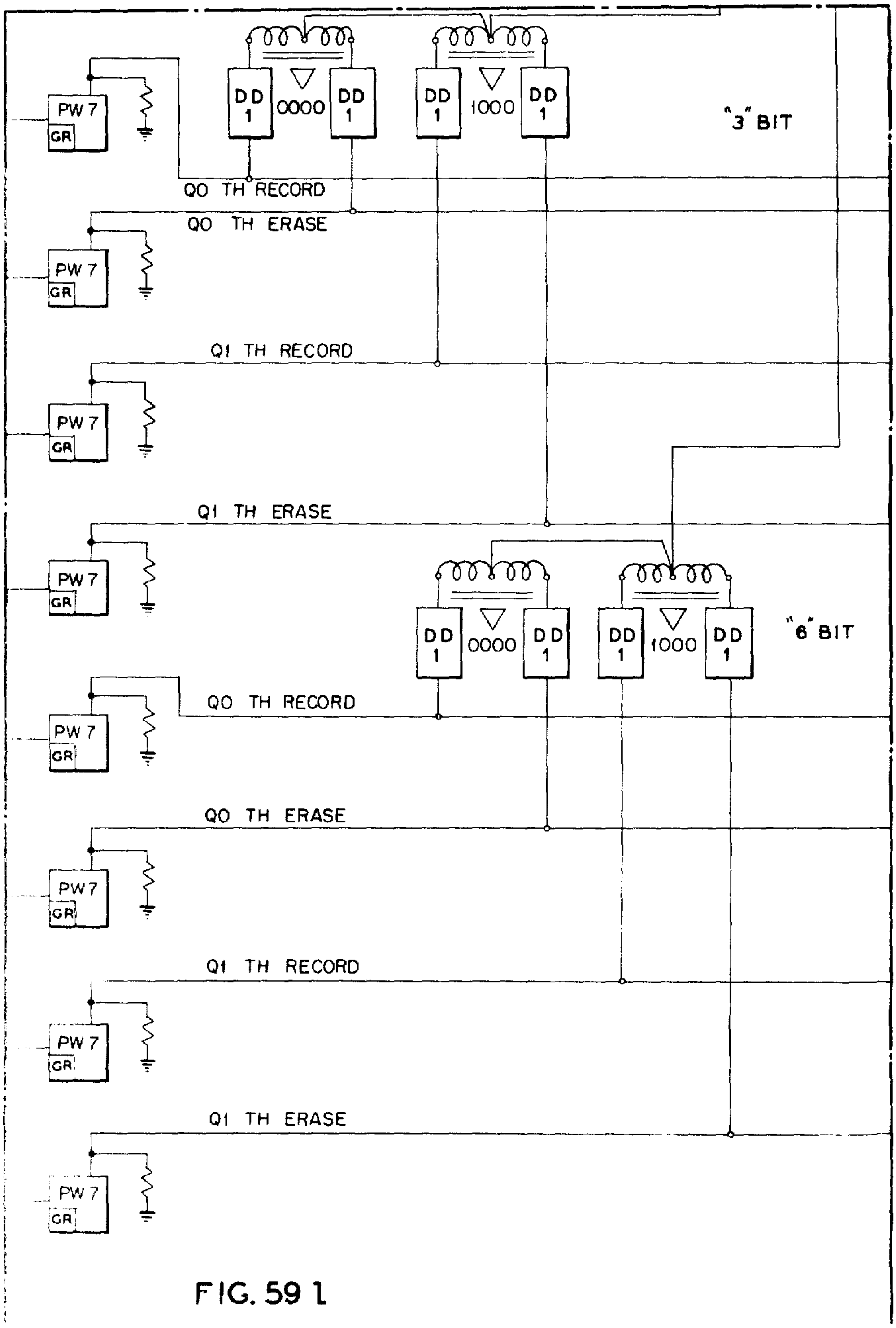


FIG. 59 L

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 56

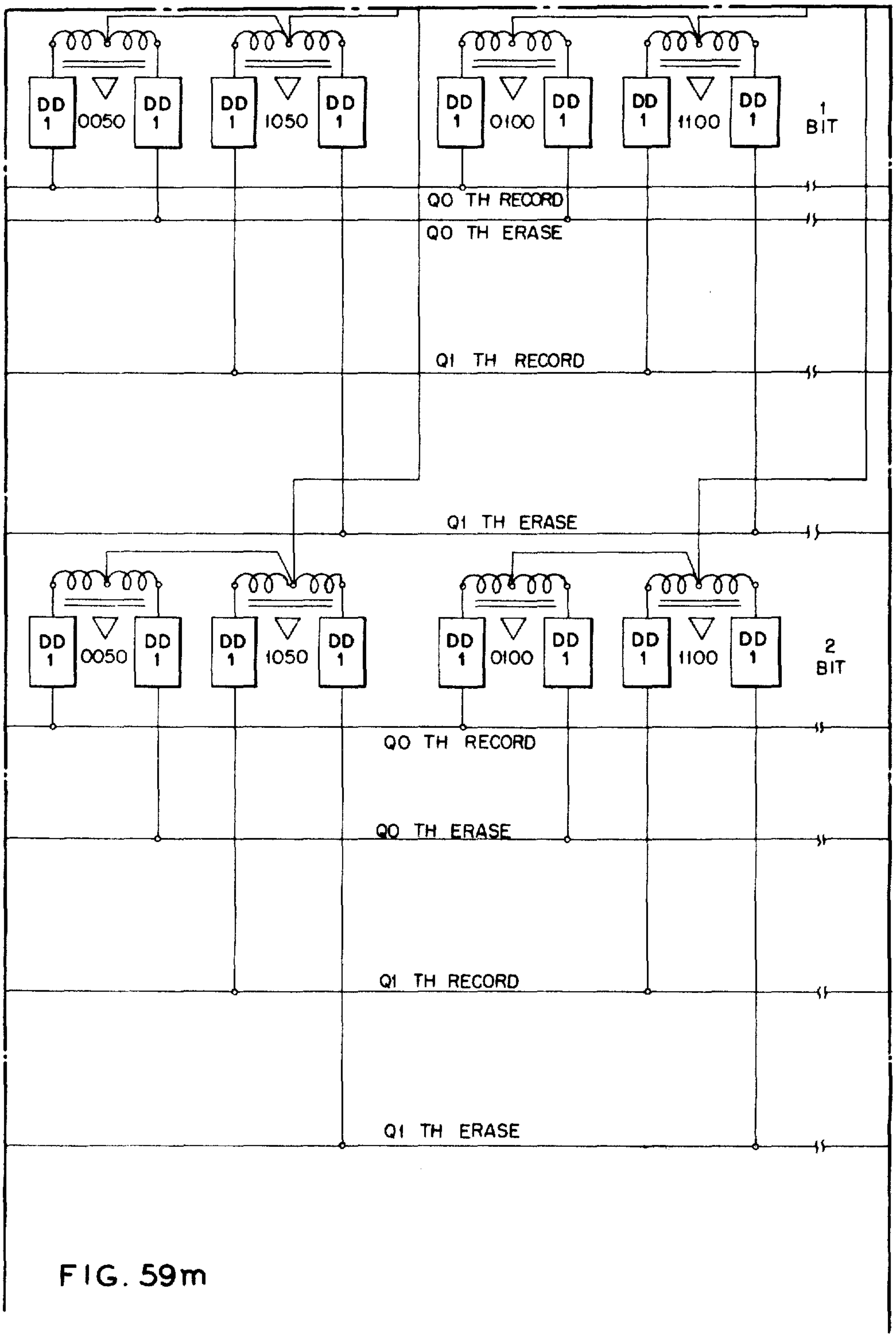


FIG. 59m

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

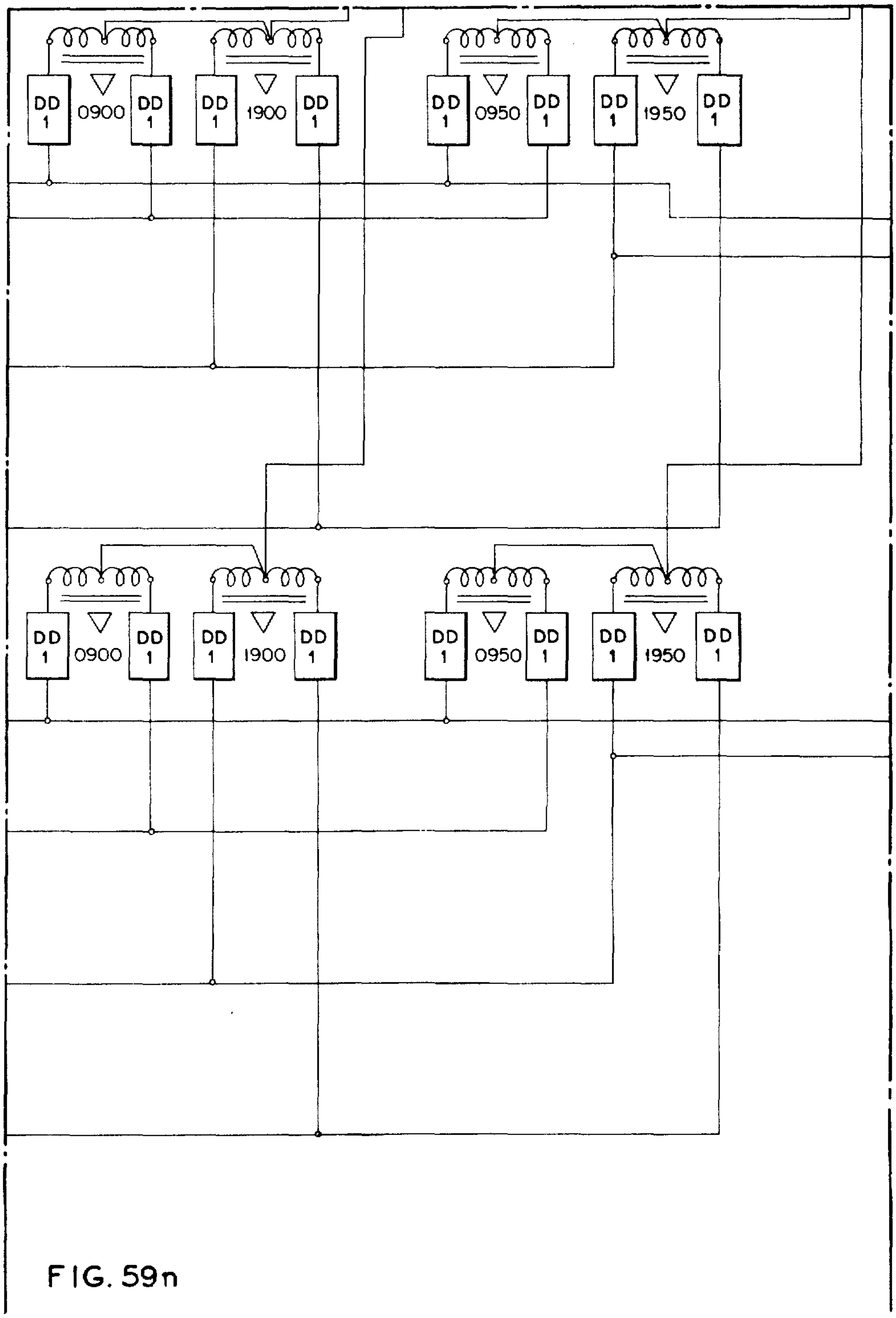


FIG. 59n

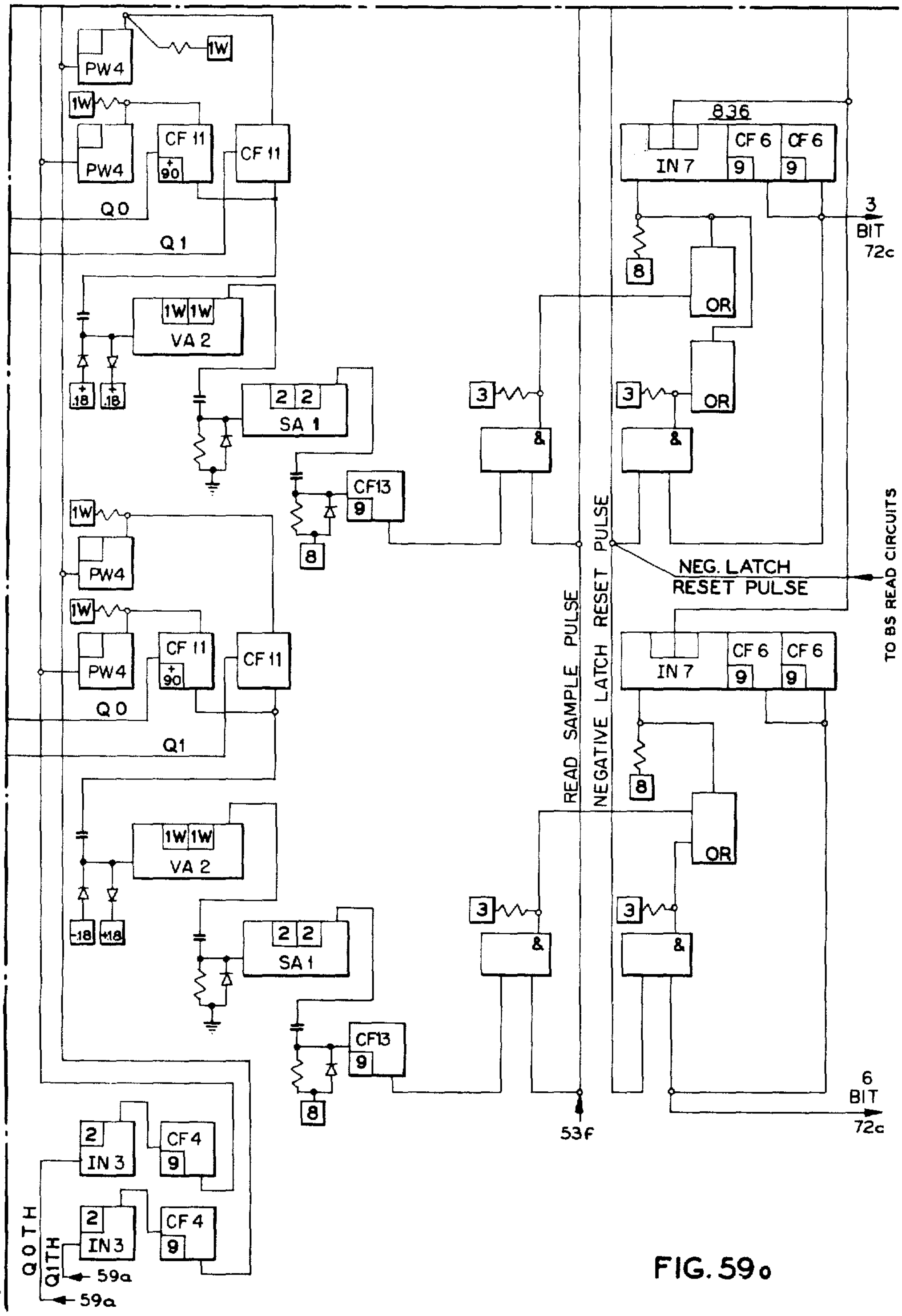

PW4
1W
1W
PW4
CF 11
+90
CF 11
Q0
Q1
1W 1W
VA 2
+18
+18
2 2
SA 1
CF13
9
8
3
&
836
IN 7
CF 6
9
CF 6
9
3
BIT
72c
8
OR
3
OR
&
READ SAMPLE PULSE
NEGATIVE LATCH RESET PULSE
NEG. LATCH
RESET PULSE
TO BS READ CIRCUITS
1W
PW4
1W
PW4
CF 11
+90
CF 11
Q0
Q1
1W 1W
VA 2
-18
+18
2 2
SA 1
CF13
9
8
IN 7
CF 6
9
CF 6
9
8
OR
3
&
3
&
6
BIT
72c
53F
2
IN 3
CF 4
9
2
IN 3
CF 4
9
Q0TH
Q1TH
59a
59a

FIG. 59o

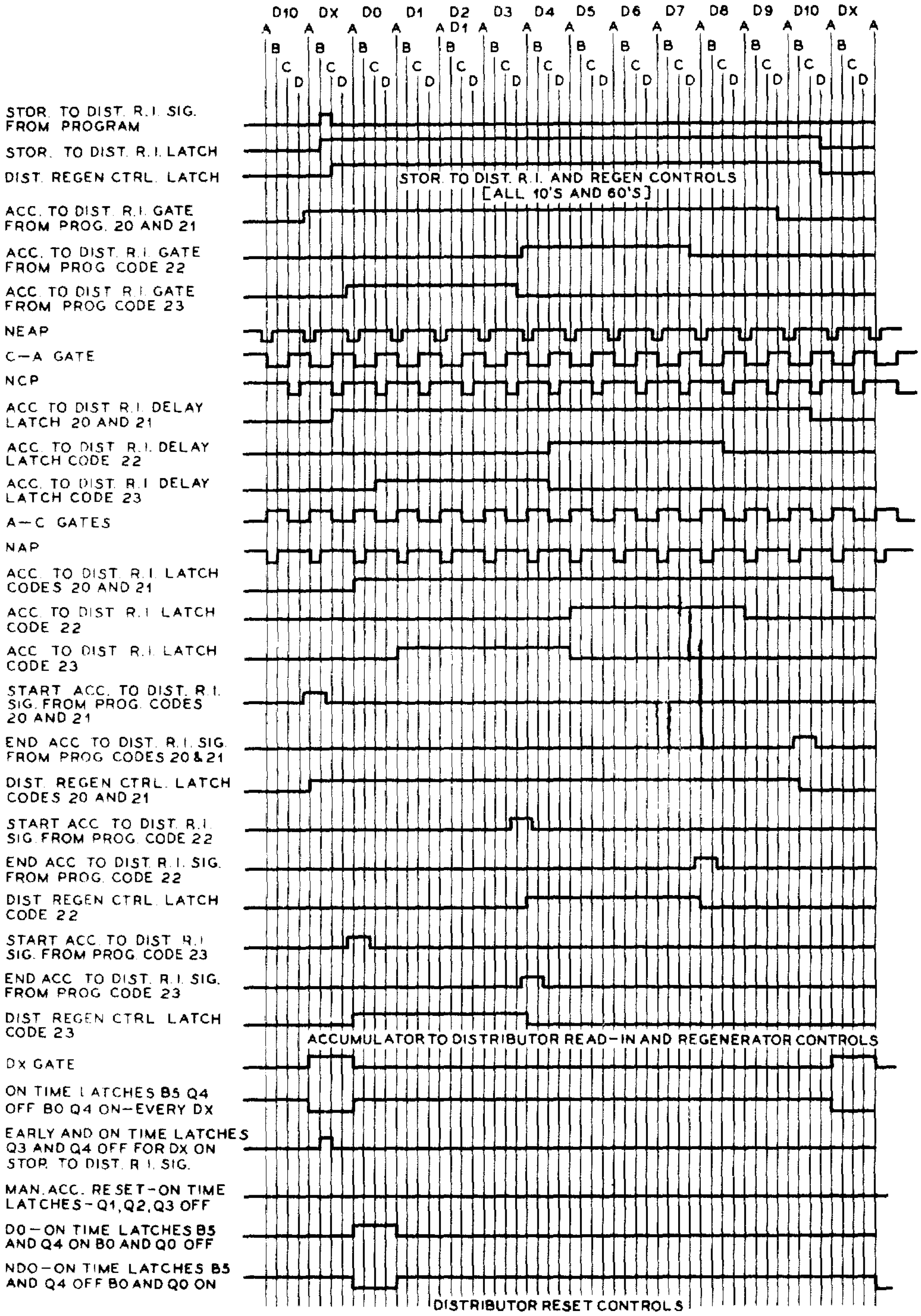

D10
DX
D0
D1
D2
D1
D3
D4
D5
D6
D7
D8
D9
D10
DX
A
B
C
D
STOR. TO DIST. R.I. SIG. FROM PROGRAM
STOR. TO DIST. R.I. LATCH
DIST. REGEN CTRL. LATCH
STOR. TO DIST. R.I. AND REGEN CONTROLS
[ALL 10'S AND 60'S]
ACC. TO DIST. R.I. GATE FROM PROG. 20 AND 21
ACC. TO DIST. R.I. GATE FROM PROG. CODE 22
ACC. TO DIST. R.I. GATE FROM PROG. CODE 23
NEAP
C—A GATE
NCP
ACC. TO DIST. R.I. DELAY LATCH 20 AND 21
ACC. TO DIST. R.I. DELAY LATCH CODE 22
ACC. TO DIST. R.I. DELAY LATCH CODE 23
A—C GATES
NAP
ACC. TO DIST. R.I. LATCH CODES 20 AND 21
ACC. TO DIST. R.I. LATCH CODE 22
ACC. TO DIST. R.I. LATCH CODE 23
START ACC. TO DIST. R.I. SIG. FROM PROG. CODES 20 AND 21
END ACC. TO DIST. R.I. SIG. FROM PROG. CODES 20 & 21
DIST. REGEN CTRL. LATCH CODES 20 AND 21
START ACC. TO DIST. R.I. SIG. FROM PROG. CODE 22
END ACC. TO DIST. R.I. SIG. FROM PROG. CODE 22
DIST. REGEN CTRL. LATCH CODE 22
START ACC. TO DIST. R.I. SIG. FROM PROG. CODE 23
END ACC. TO DIST. R.I. SIG. FROM PROG. CODE 23
DIST. REGEN CTRL. LATCH CODE 23
ACCUMULATOR TO DISTRIBUTOR READ-IN AND REGENERATOR CONTROLS
DX GATE
ON TIME LATCHES B5 Q4 OFF B0 Q4 ON—EVERY DX
EARLY AND ON TIME LATCHES Q3 AND Q4 OFF FOR DX ON STOR. TO DIST. R.I. SIG.
MAN. ACC. RESET—ON TIME LATCHES—Q1, Q2, Q3 OFF
D0—ON TIME LATCHES B5 AND Q4 ON B0 AND Q0 OFF
ND0—ON TIME LATCHES B5 AND Q4 OFF B0 AND Q0 ON
DISTRIBUTOR RESET CONTROLS

FIG. 60

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 60

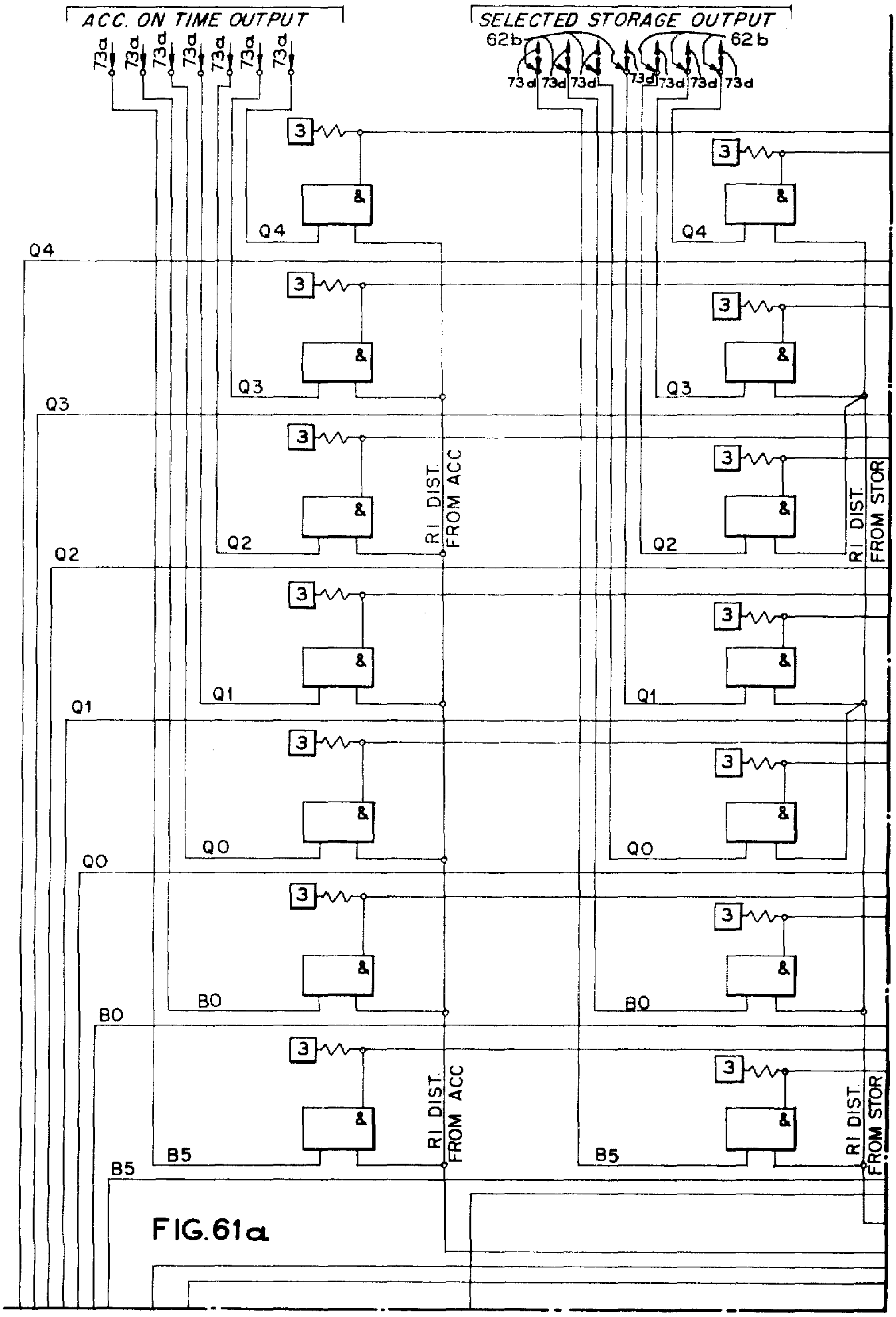


FIG.61a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 61

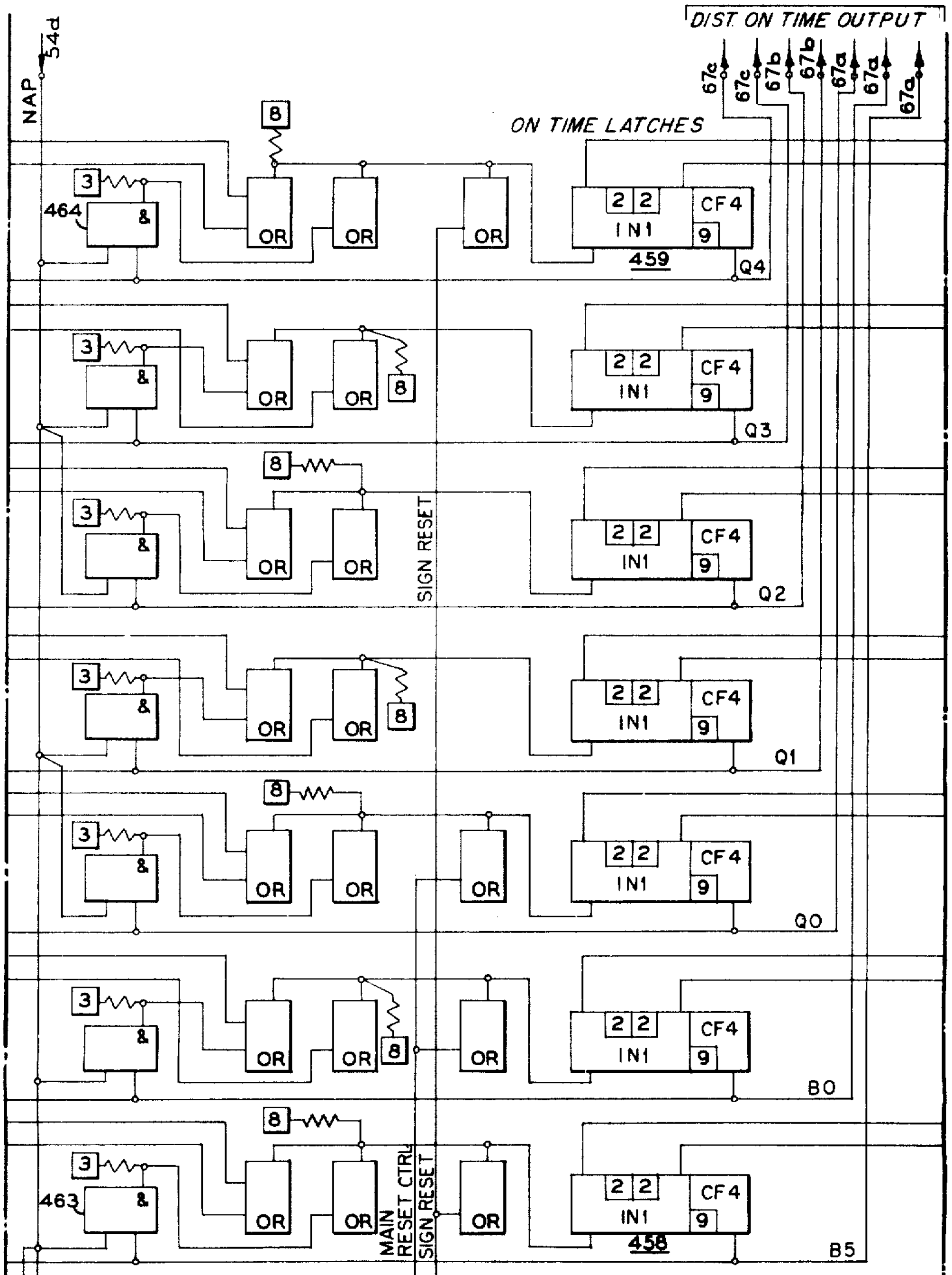


FIG. 61b

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 62

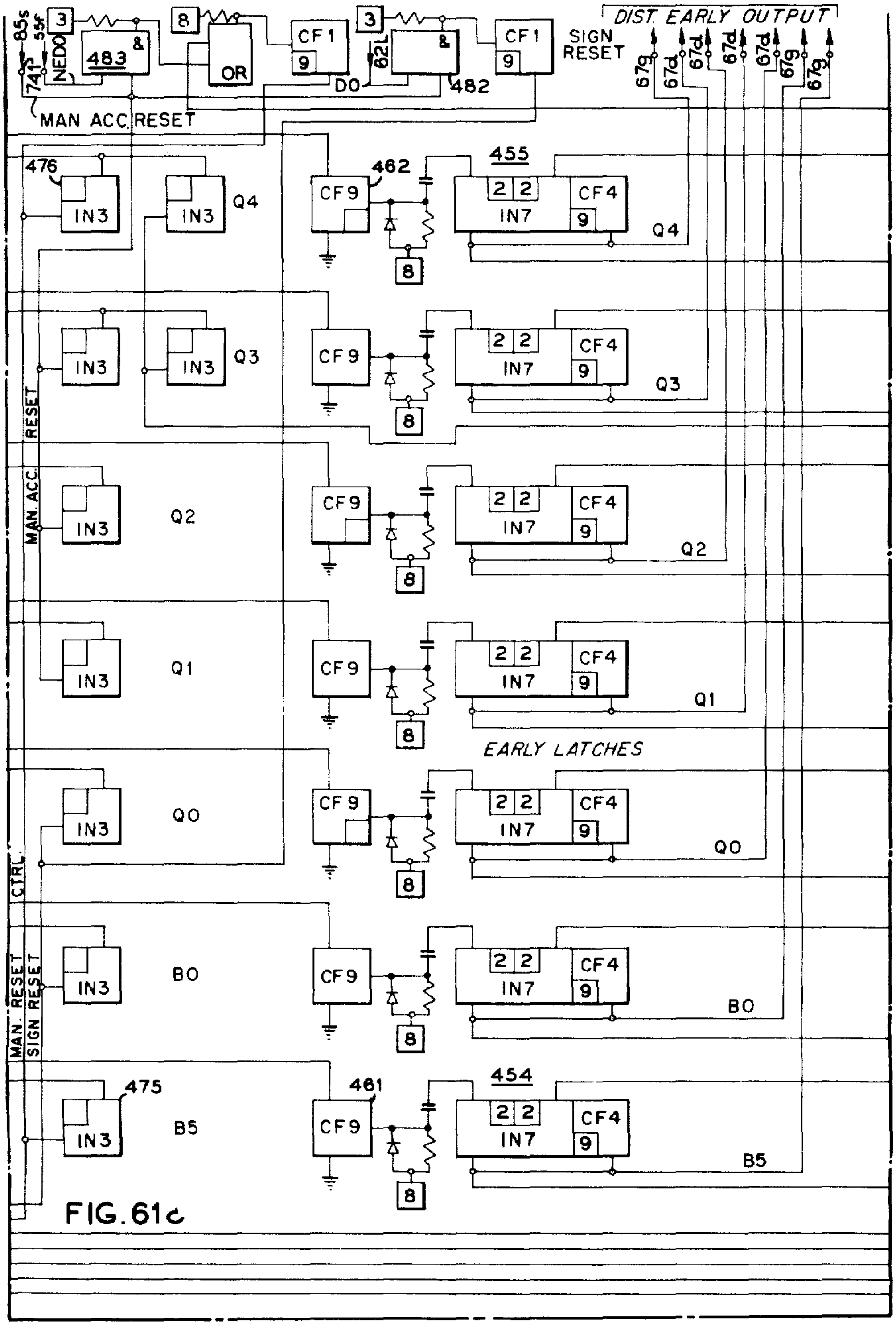


FIG. 61c

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 63

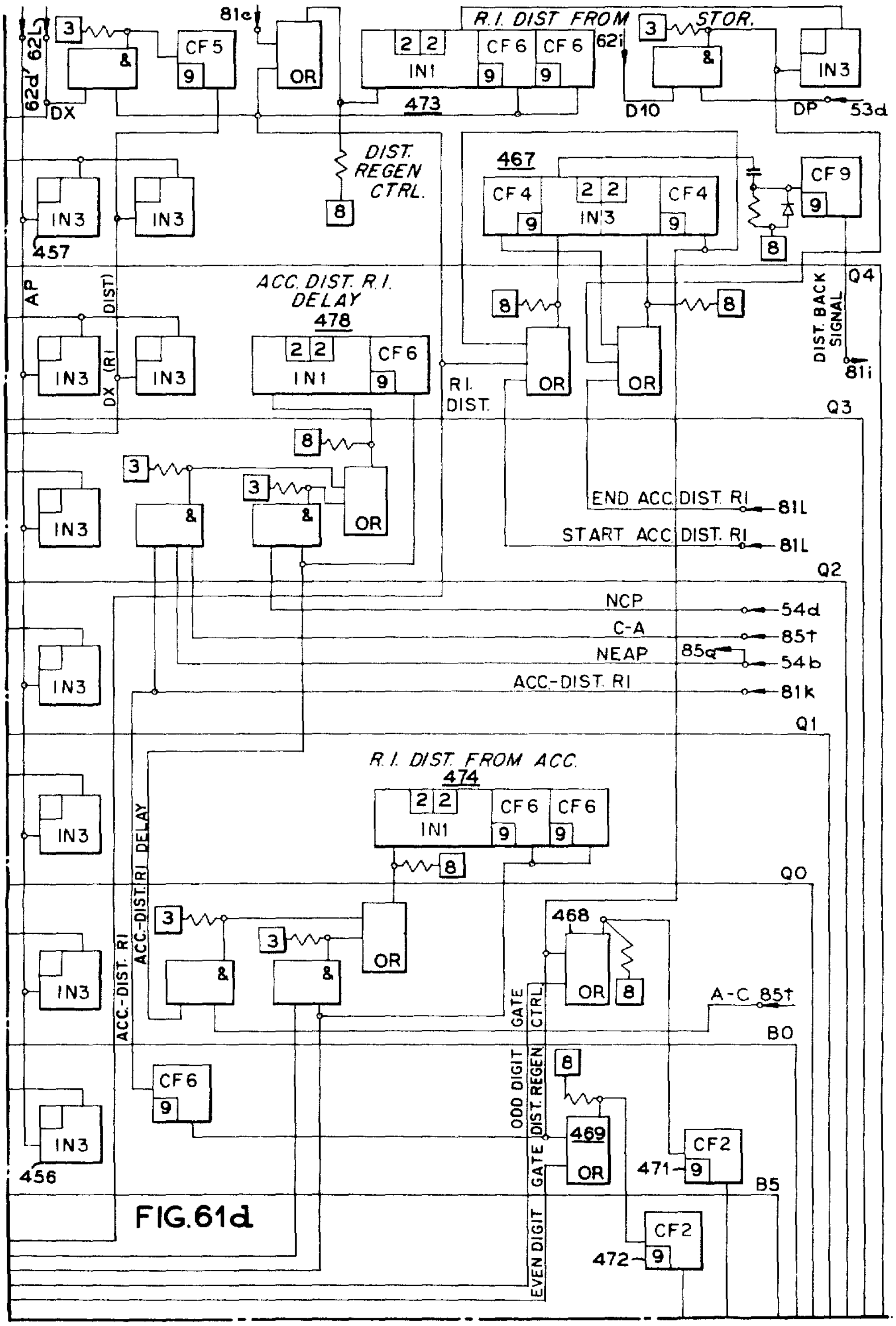
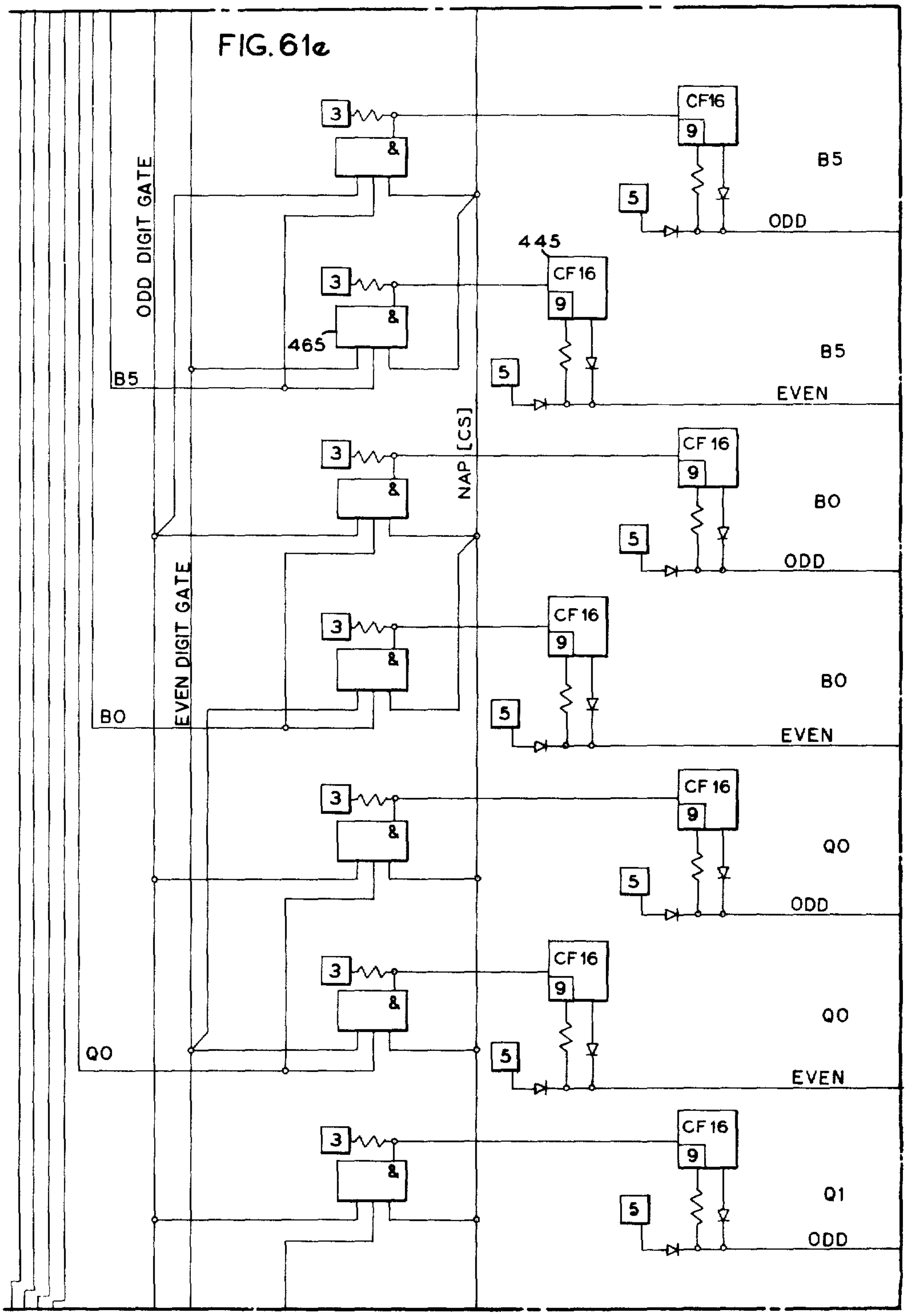


FIG. 61d

FIG. 61e


ODD DIGIT GATE
EVEN DIGIT GATE
NAP [CS]
3
&
CF16
9
5
B5
ODD
445
465
B5
EVEN
BO
ODD
BO
EVEN
QO
ODD
QO
EVEN
Q1
ODD

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 65

FIG. 61 f

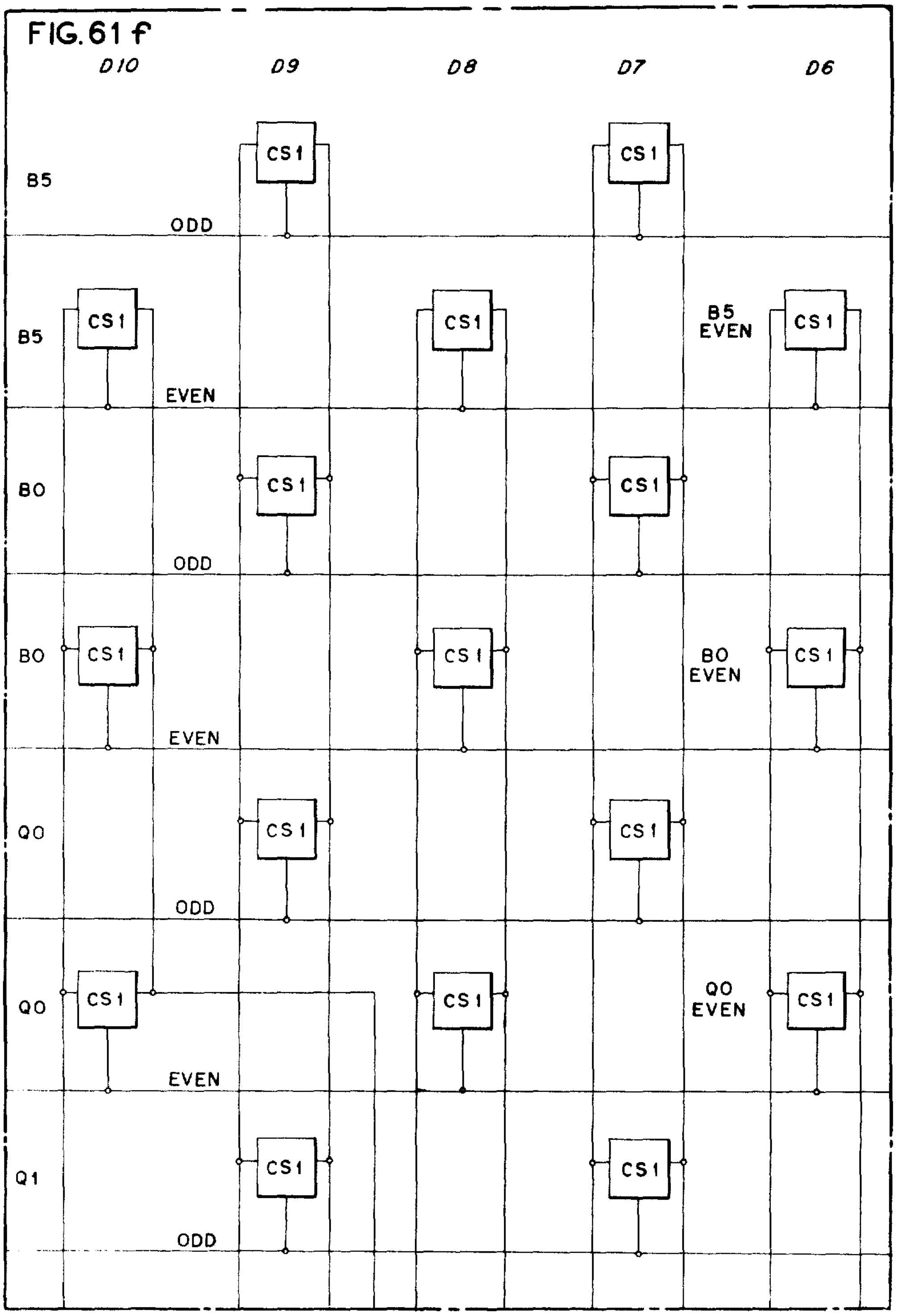
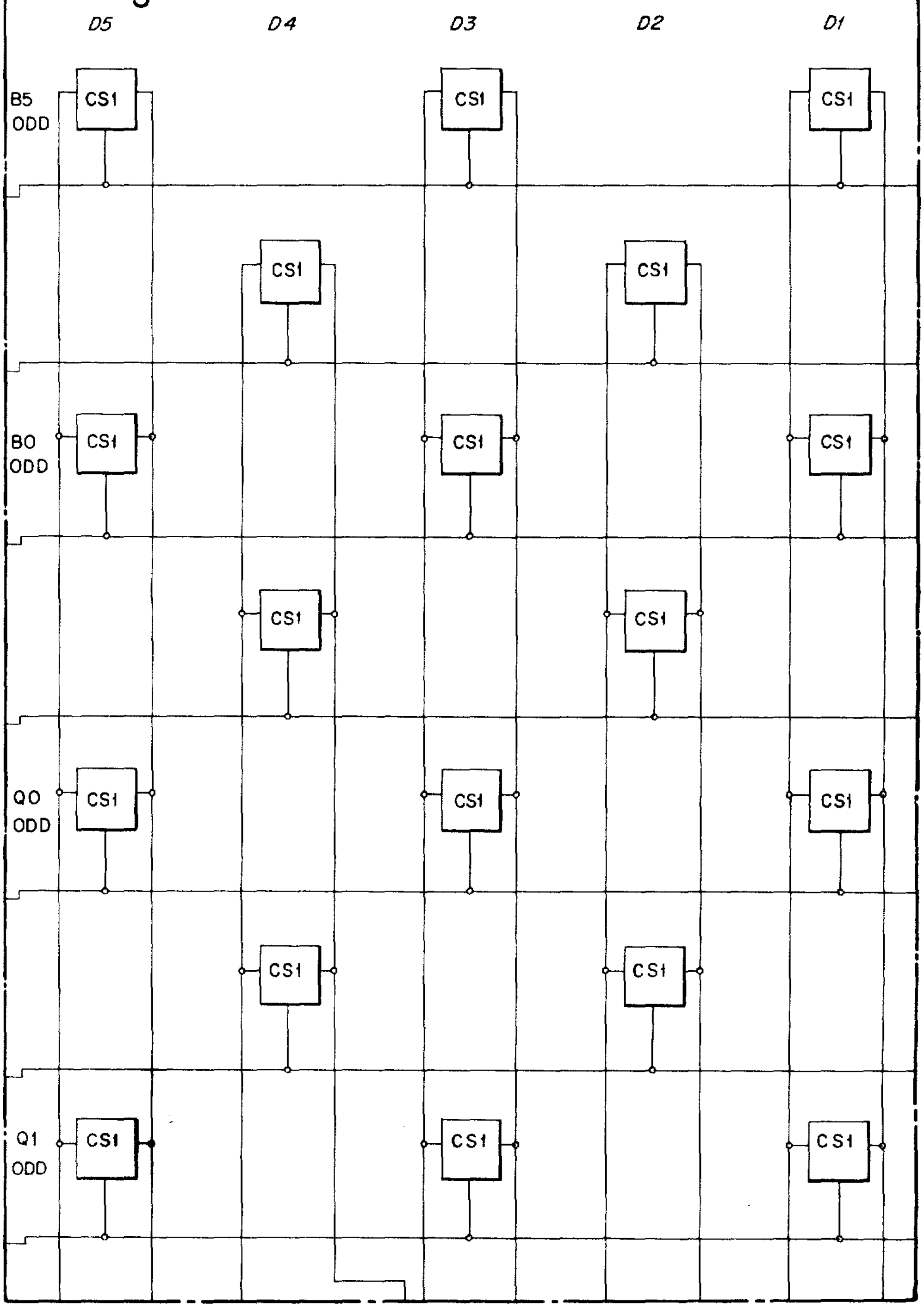


FIG. 61g

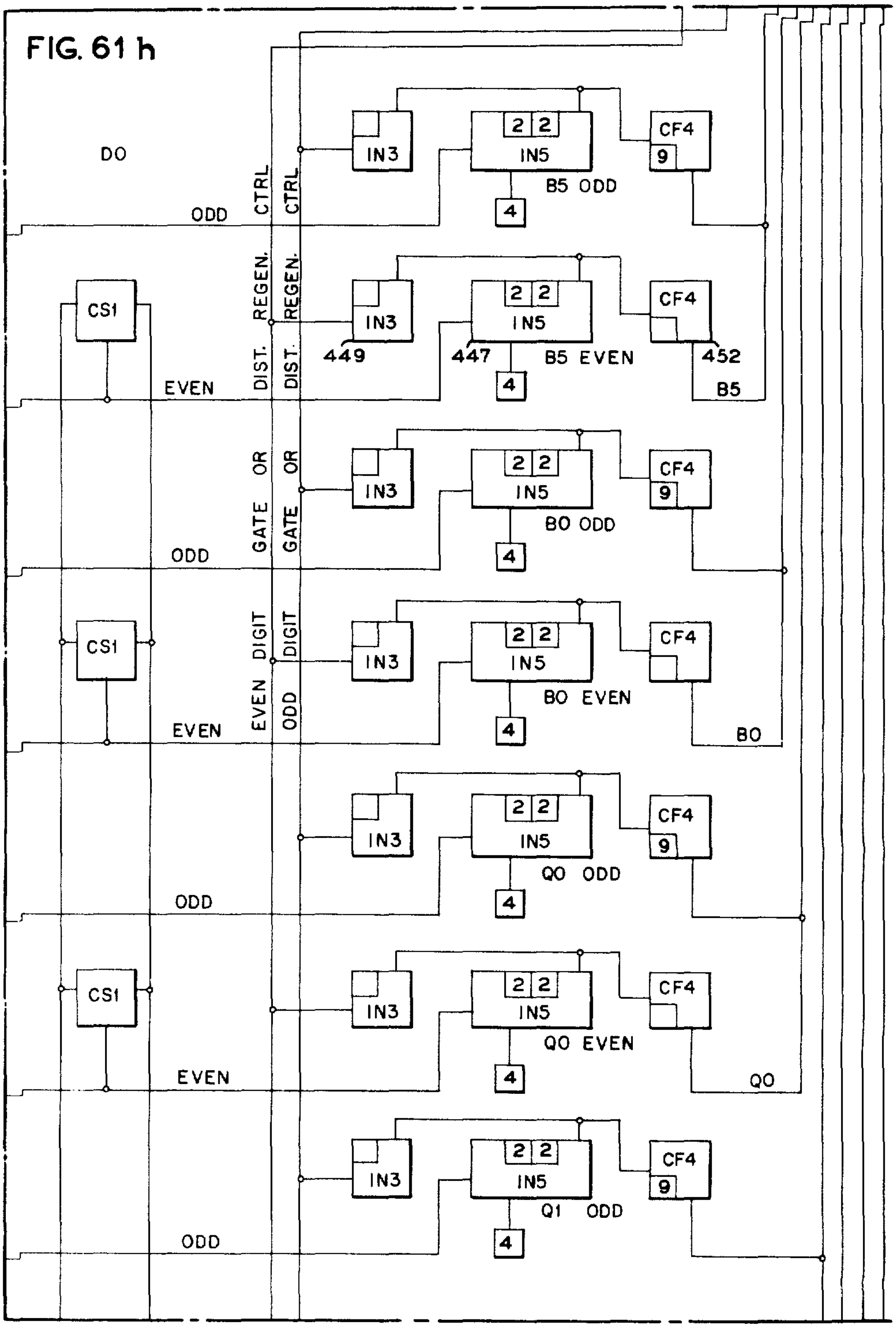


Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 67

FIG. 61 h

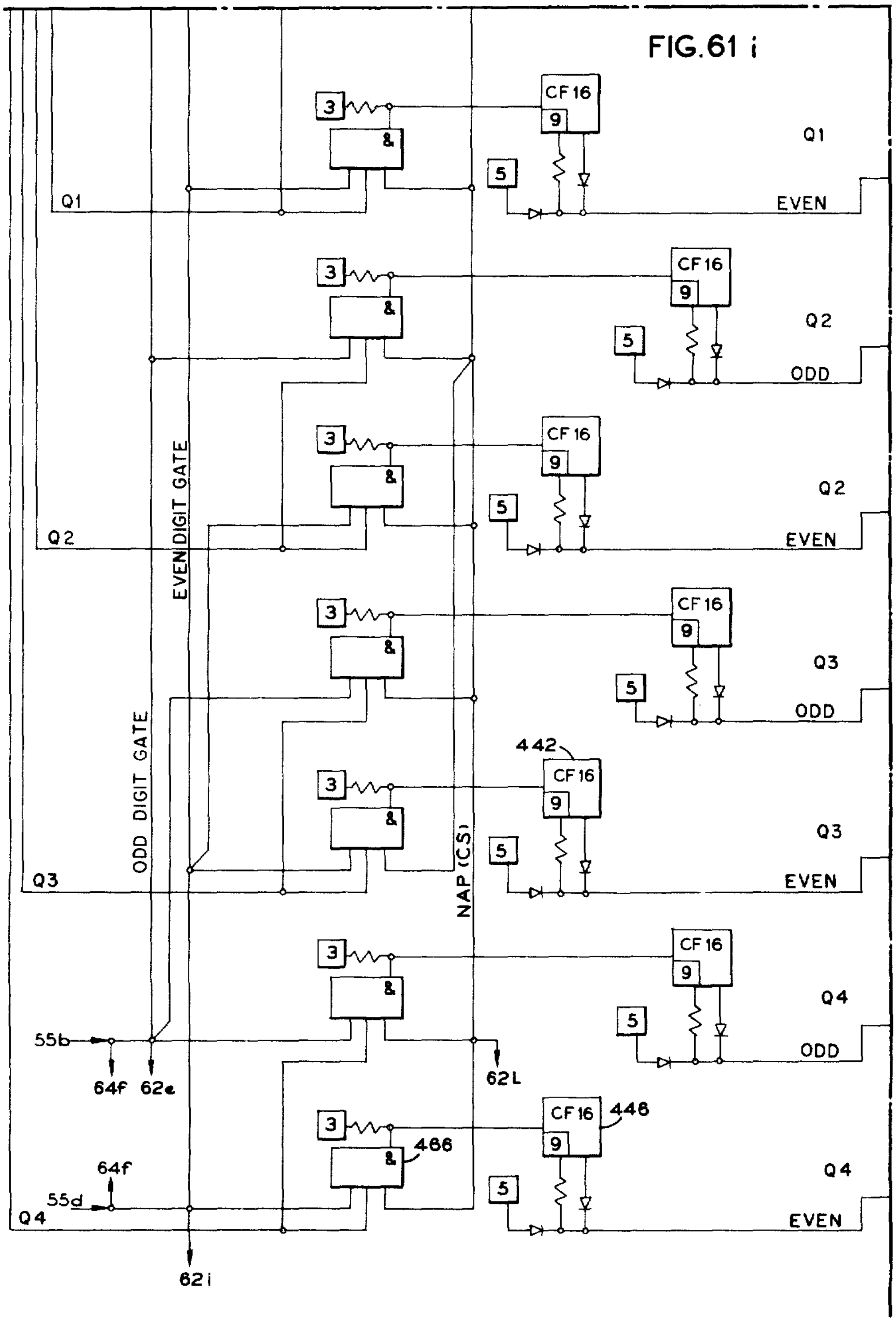


Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 68

FIG. 61 i



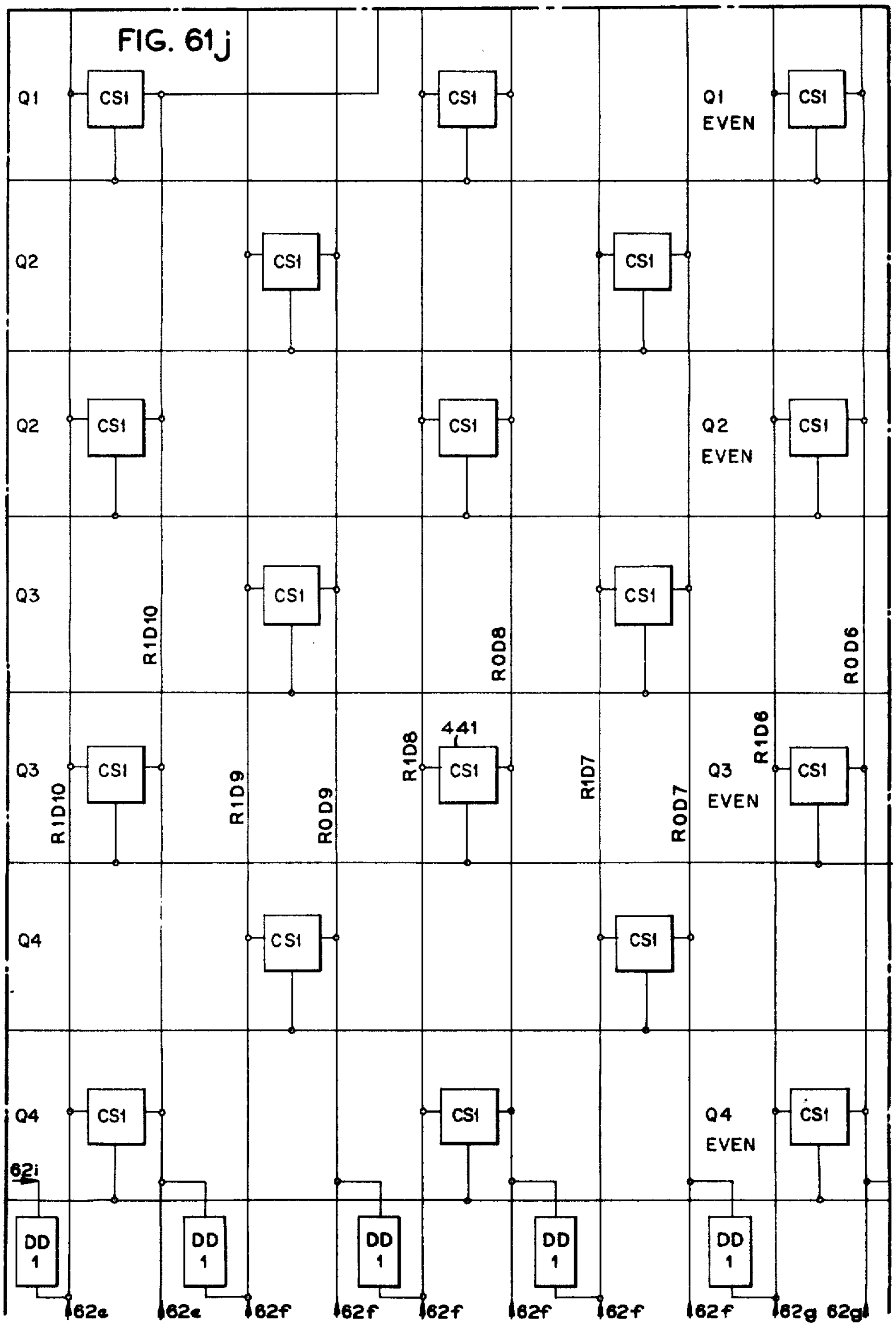


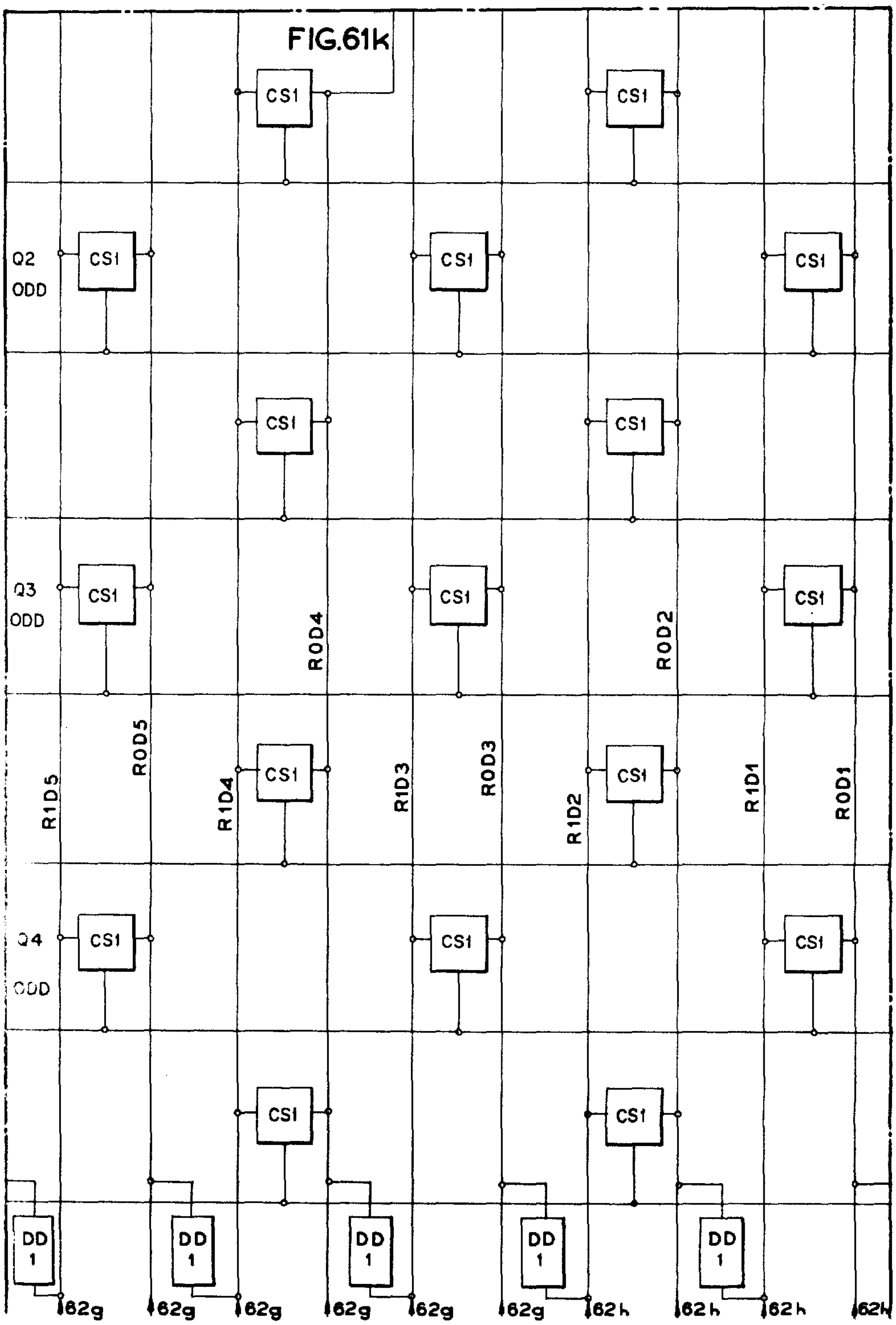
FIG. 61j
Q1
CS1
CS1
Q1
EVEN
CS1
Q2
CS1
CS1
Q2
CS1
CS1
Q2
EVEN
CS1
Q3
CS1
R1D10
ROD8
CS1
ROD6
Q3
CS1
R1D10
R1D9
ROD9
R1D8
441
CS1
R1D7
ROD7
R1D6
Q3
EVEN
CS1
Q4
CS1
CS1
Q4
CS1
CS1
Q4
EVEN
CS1
62i
DD
1
DD
1
DD
1
DD
1
DD
1
62e
62e
62f
62f
62f
62f
62f
62f
62g
62g

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 70

FIG. 61k

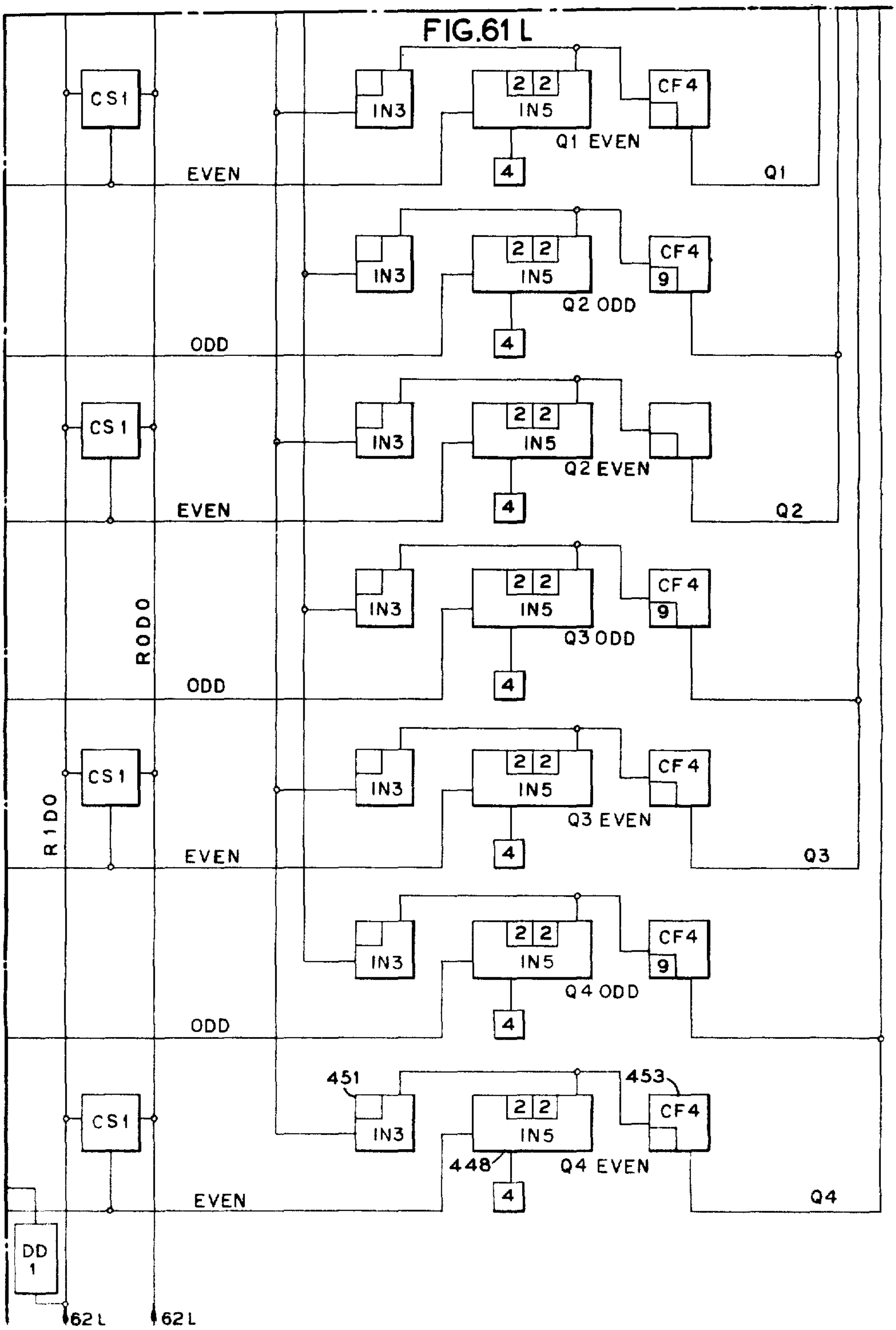


Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 71

FIG.61 L



Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 72

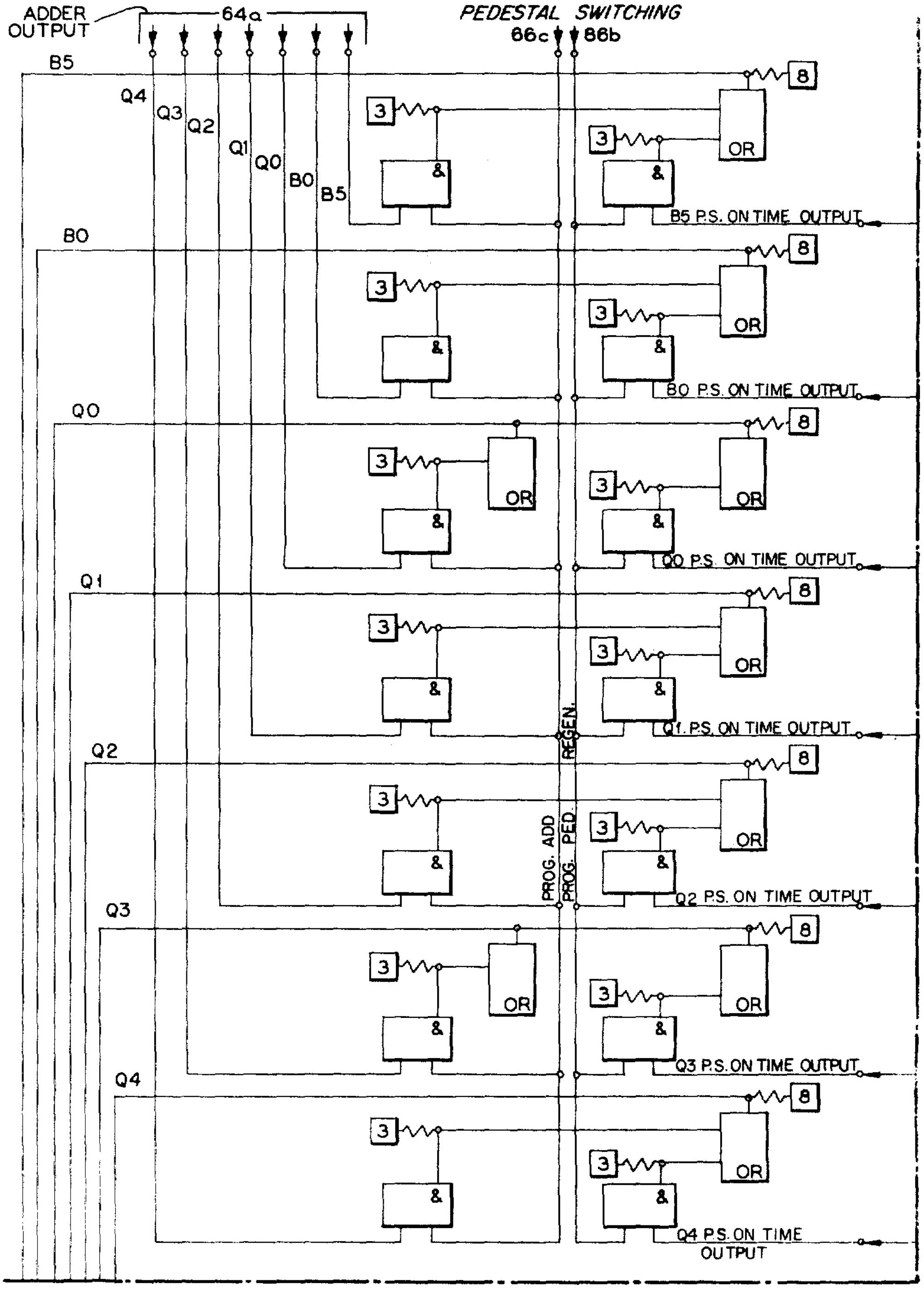


FIG. 62a

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

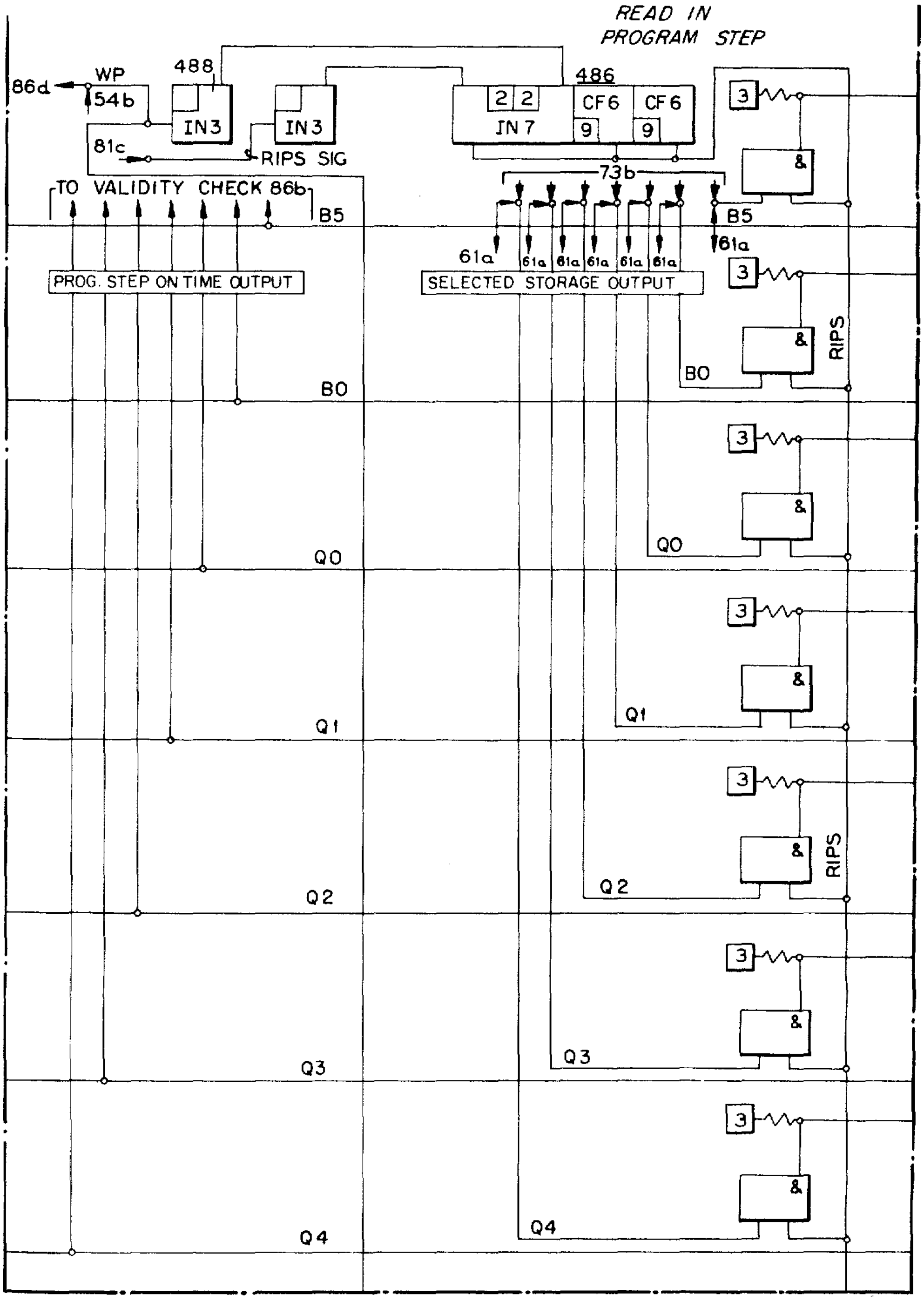


FIG. 62 b

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 74

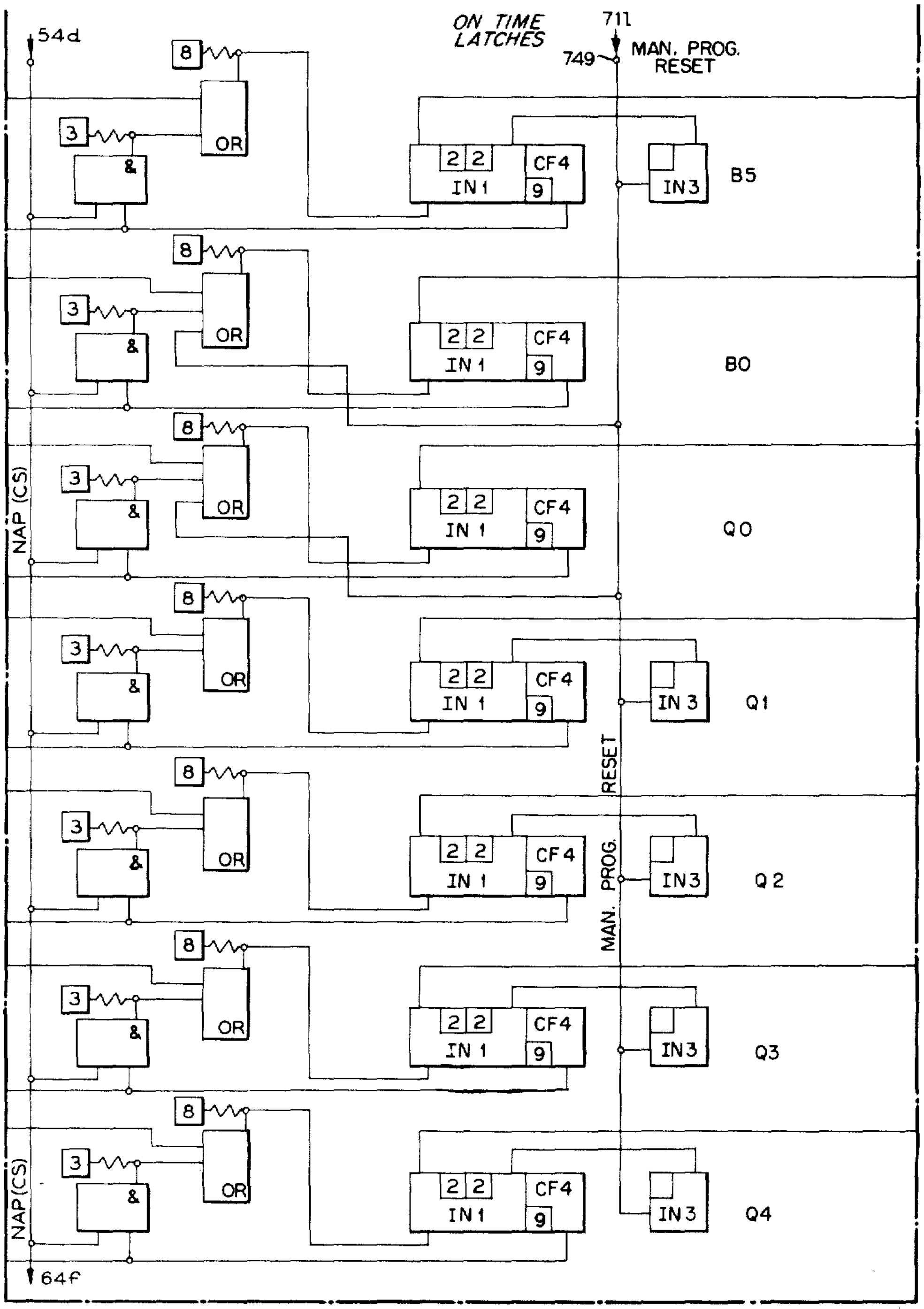


FIG. 62c

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 75

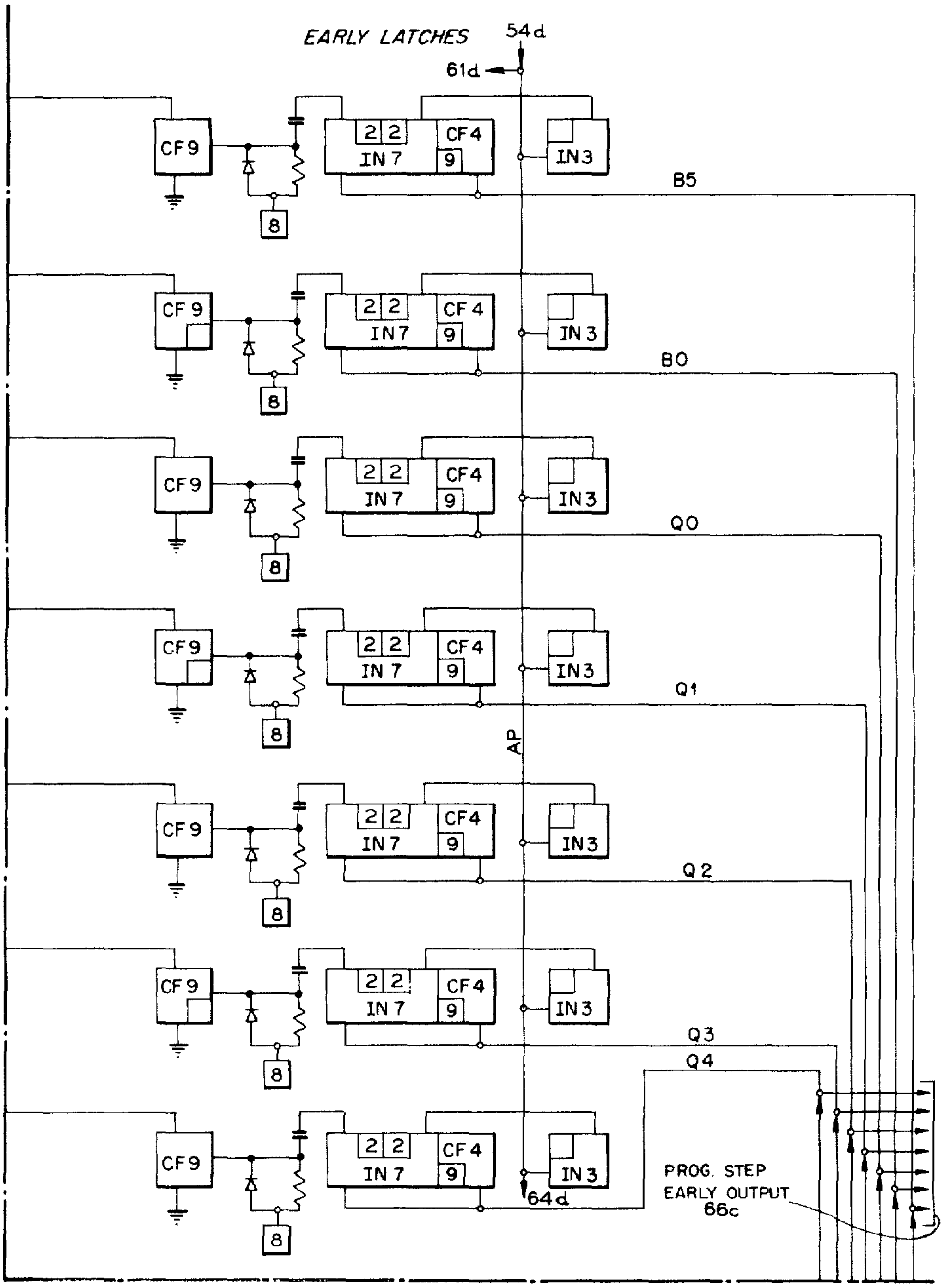


FIG. 62 d

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 76

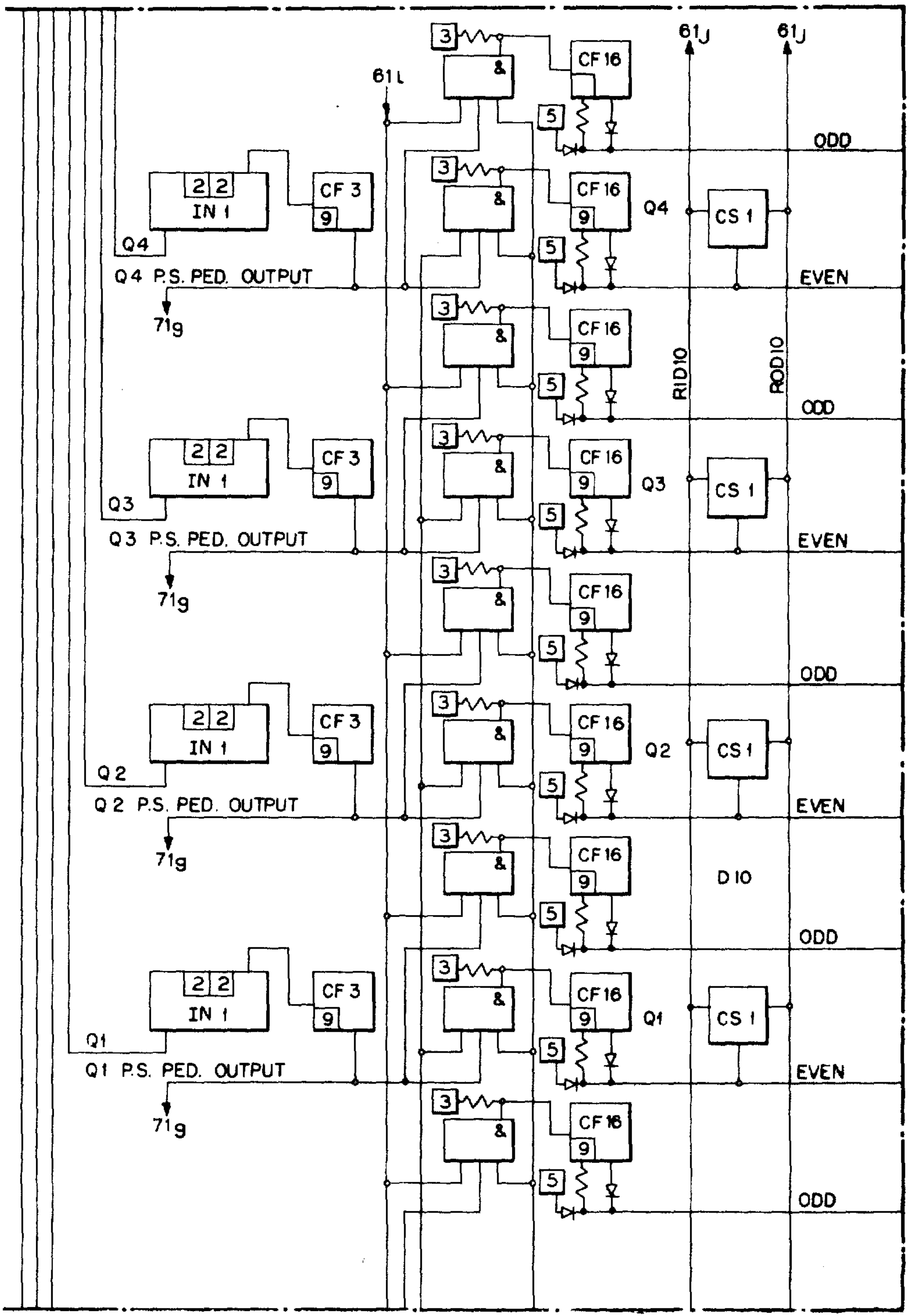


FIG. 62e

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 77

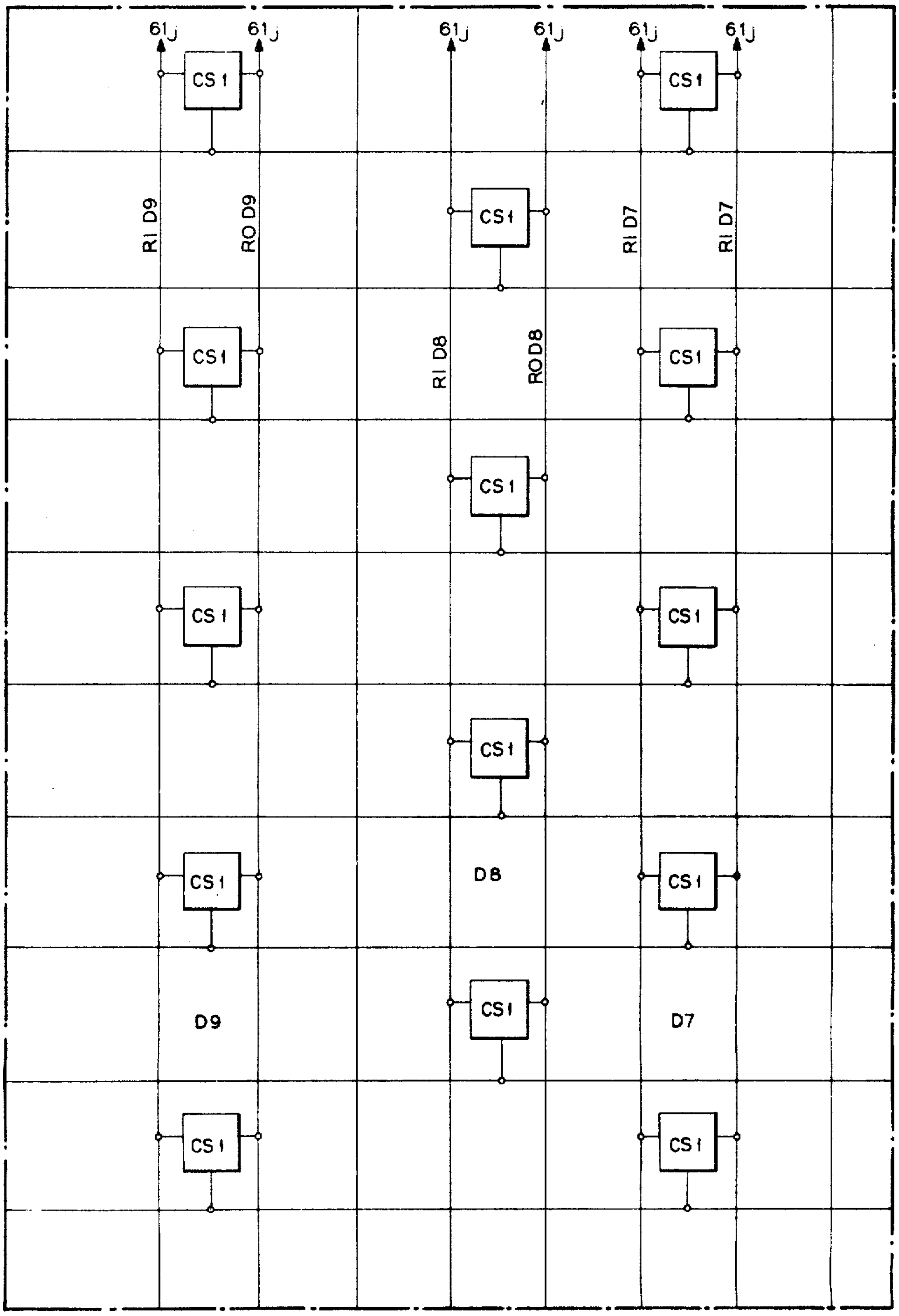


FIG.62F

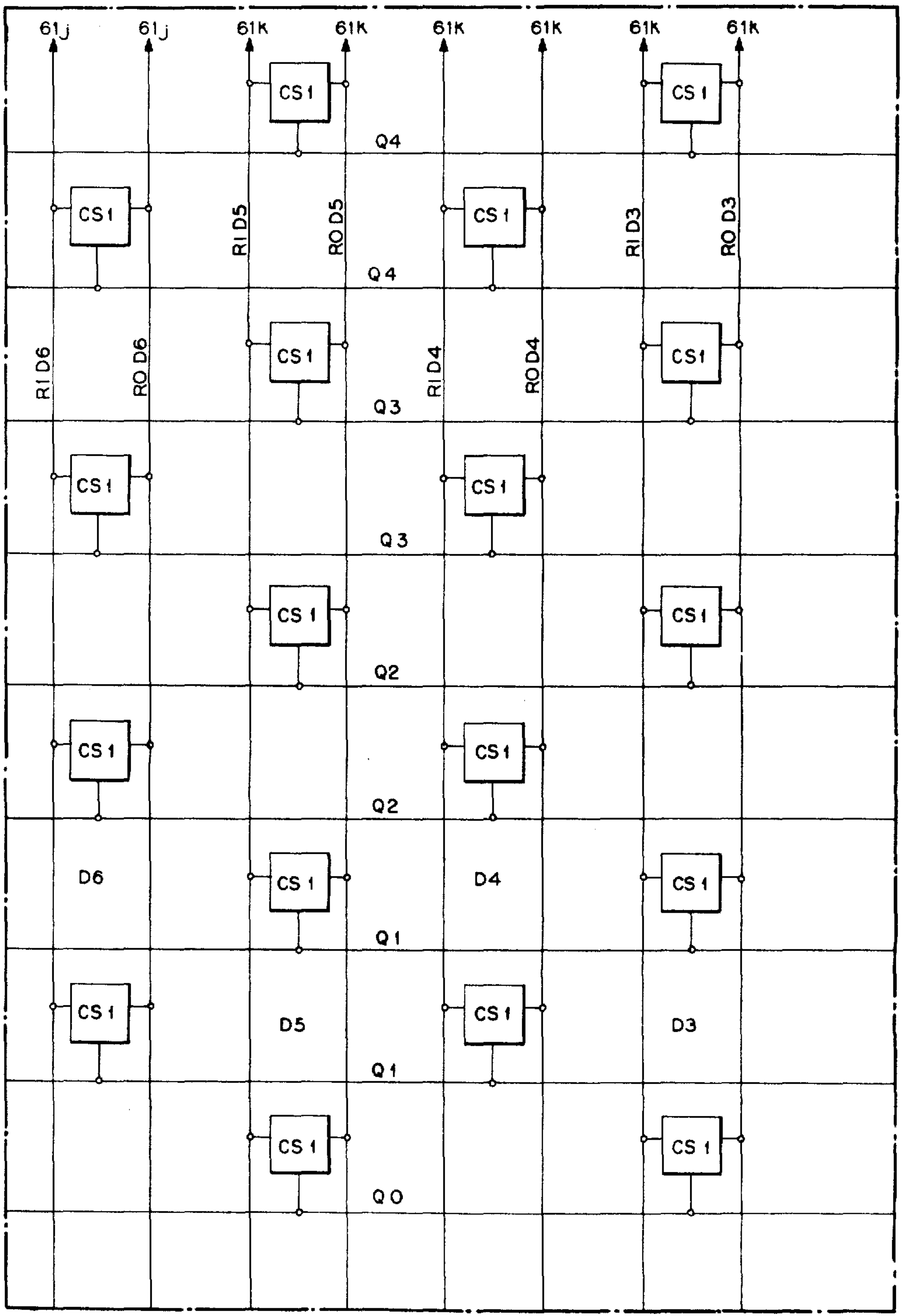

61j
61j
61K
61K
61K
61K
61K
61K
CS1
Q4
RI D6
RO D6
RI D5
RO D5
RI D4
RO D4
RI D3
RO D3
Q4
Q3
Q3
Q2
Q2
D6
D5
D4
D3
Q1
Q1
Q0

FIG.62g

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 79

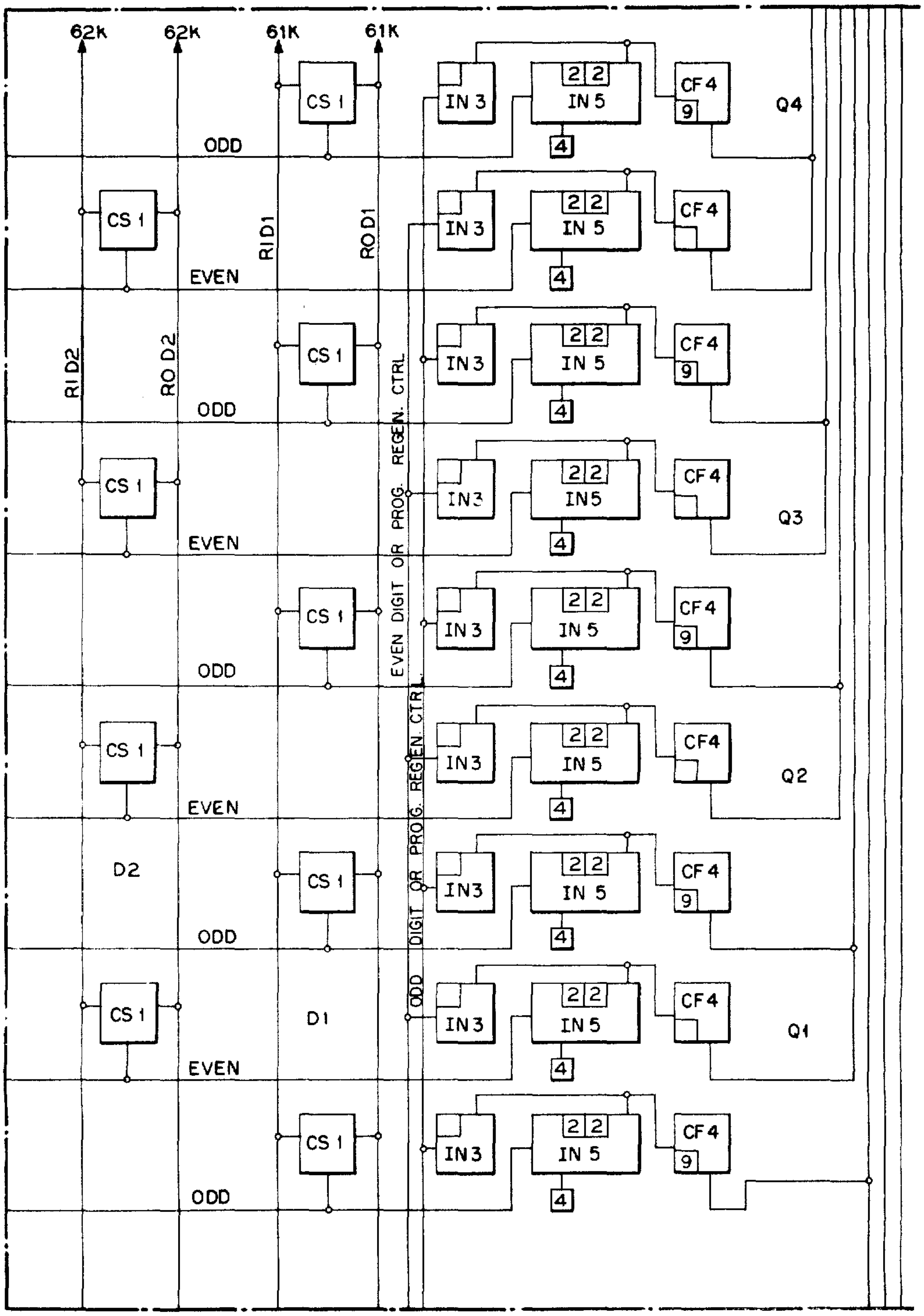


FIG. 62h

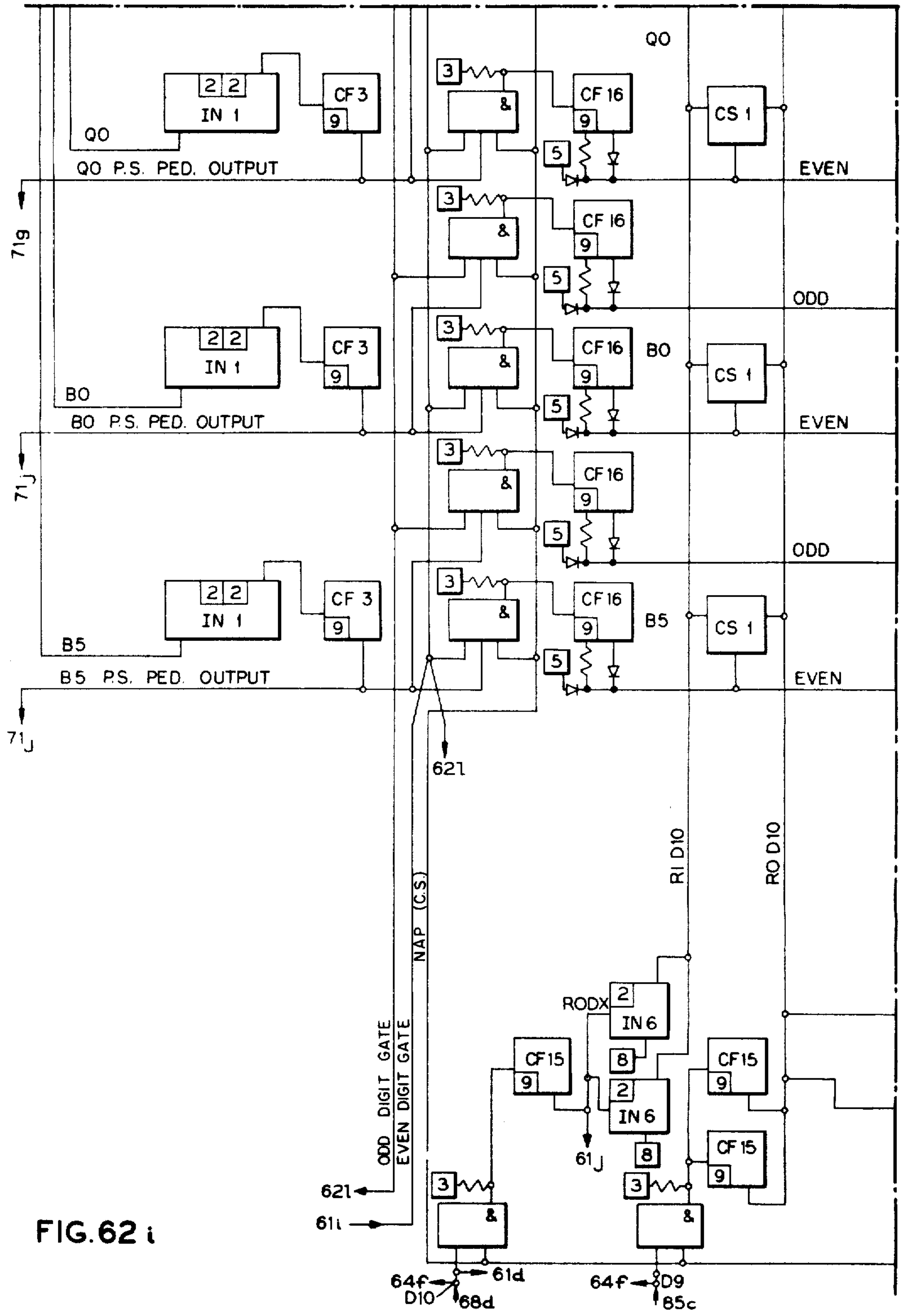

QO
2 2
IN 1
CF 3
9
QO
QO P.S. PED. OUTPUT
71g
3
&
CF 16
9
5
CS 1
EVEN
ODD
BO
BO P.S. PED. OUTPUT
71J
B5
B5 P.S. PED. OUTPUT
71J
621
NAP (C.S.)
ODD DIGIT GATE
EVEN DIGIT GATE
RI D10
RO D10
RODX
2
IN 6
8
CF15
61J
621
61i
64f
D10
68d
61d
64f
D9
85c

FIG. 62 i

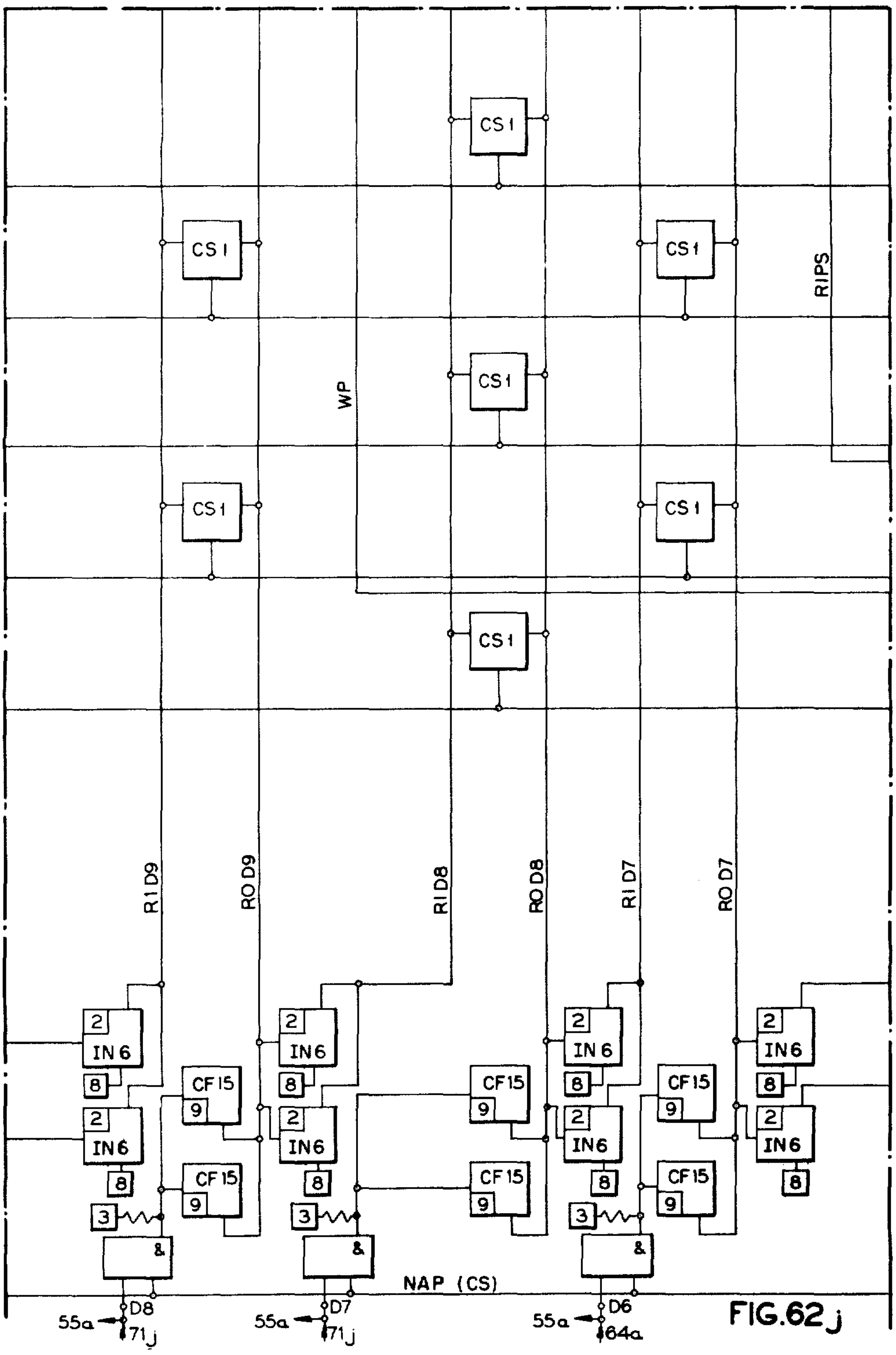

CS 1
CS 1
CS 1
CS 1
CS 1
CS 1
CS 1
CS 1
RIPS
WP
RI D9
RO D9
RI D8
RO D8
RI D7
RO D7
2
IN 6
8
CF 15
9
3
&
NAP (CS)
D8
D7
D6
55a
71j
64a

FIG. 62j

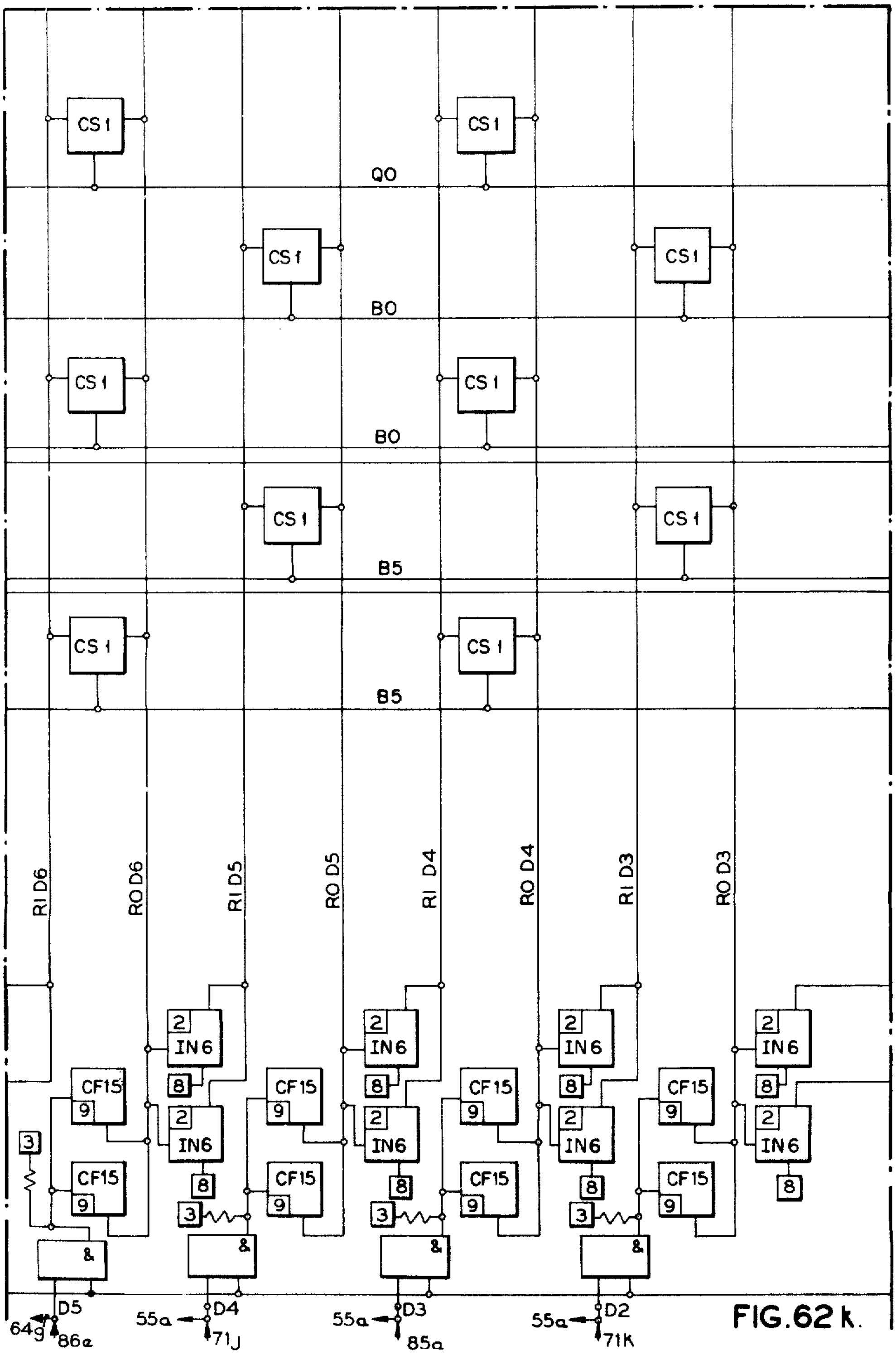

CS1
QO
BO
BO
B5
B5
RI D6
RO D6
RI D5
RO D5
RI D4
RO D4
RI D3
RO D3
IN6
CF15
2
8
9
3
&
D5
D4
D3
D2
64g
86a
55a
71J
85a
71K

FIG. 62k.

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 83

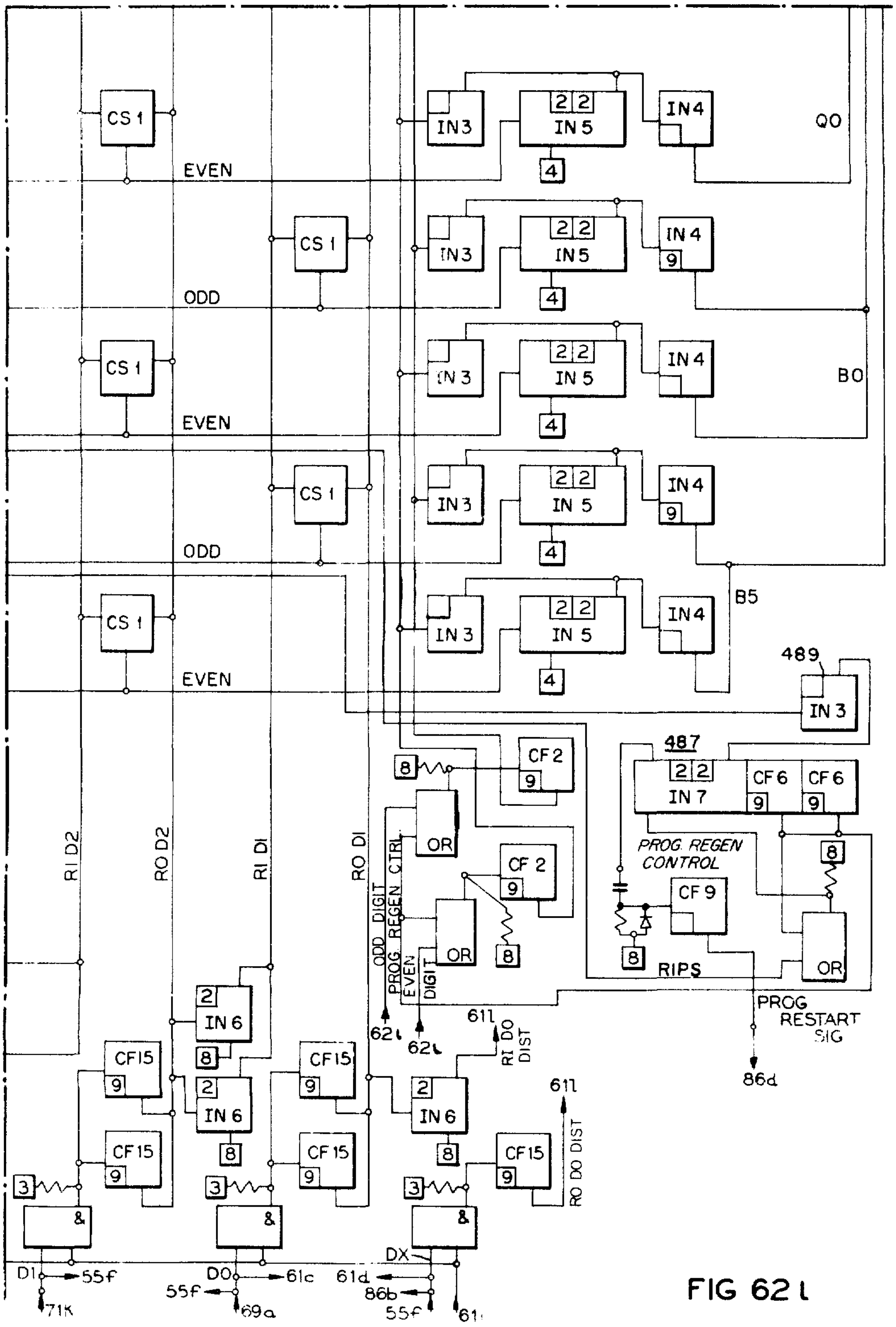


FIG 62 l

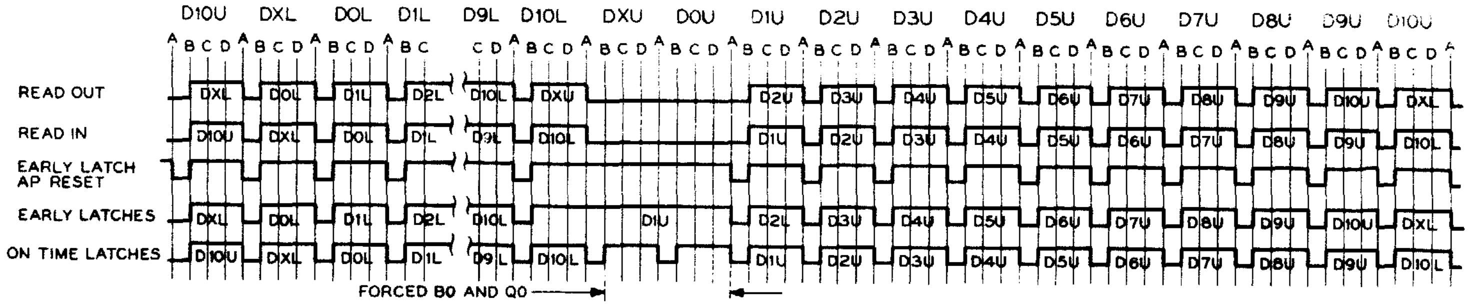
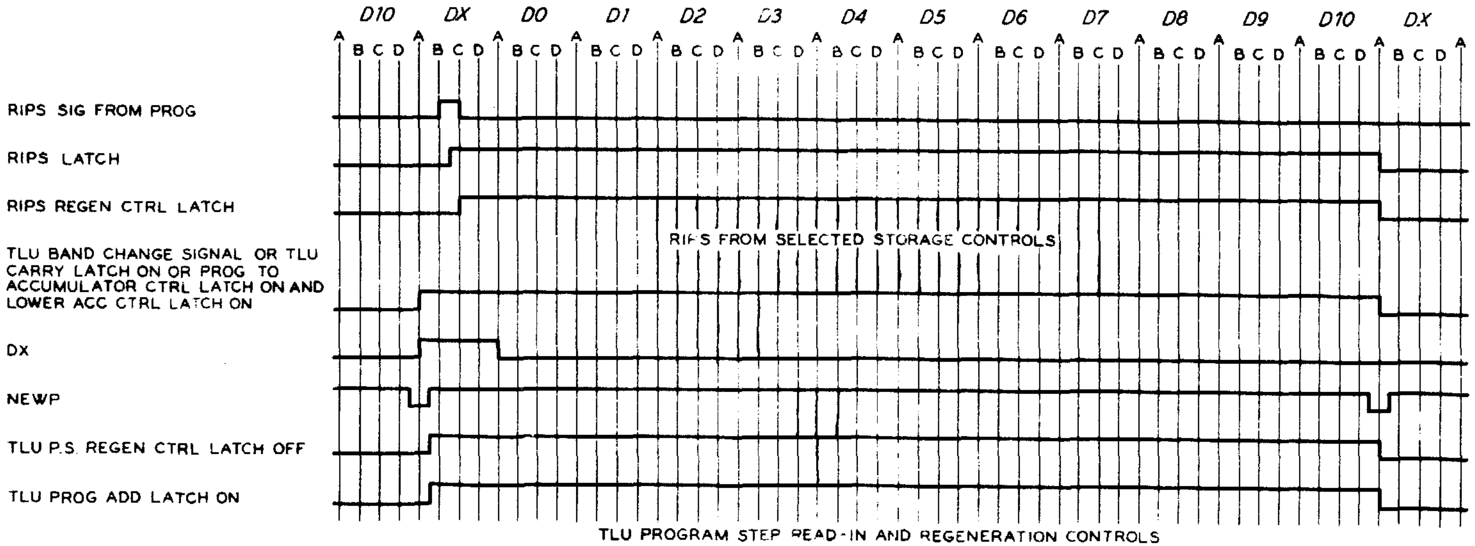

D10U
DXL
D0L
D1L
D9L
D10L
DXU
D0U
D1U
D2U
D3U
D4U
D5U
D6U
D7U
D8U
D9U
D10U
A B C D
READ OUT
READ IN
EARLY LATCH AP RESET
EARLY LATCHES
ON TIME LATCHES
DXL D0L D1L D2L D10L DXU D2U D3U D4U D5U D6U D7U D8U D9U D10U DXL
D10U DXL D0L D1L D9L D10L D1U D2U D3U D4U D5U D6U D7U D8U D9U D10L
DXL D0L D1L D2L D10L D1U D2L D3U D4U D5U D6U D7U D8U D9U D10U DXL
D10U DXL D0L D1L D9L D10L D1U D2U D3U D4U D5U D6U D7U D8U D9U D10L
FORCED B0 AND Q0

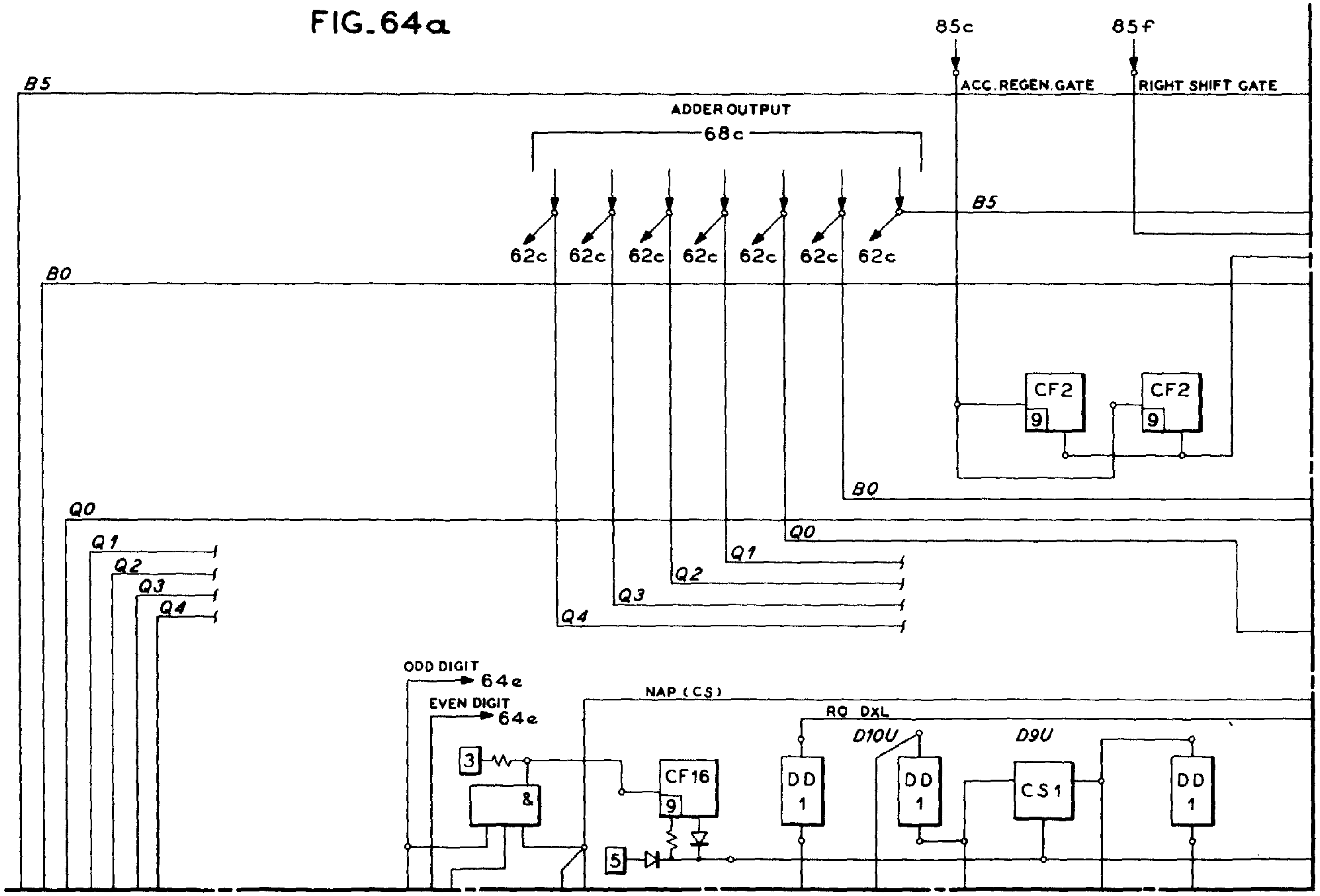
FIG. 65


D10
DX
D0
D1
D2
D3
D4
D5
D6
D7
D8
D9
D10
DX
A B C D
RIPS SIG FROM PROG
RIPS LATCH
RIPS REGEN CTRL LATCH
RIPS FROM SELECTED STORAGE CONTROLS
TLU BAND CHANGE SIGNAL OR TLU CARRY LATCH ON OR PROG TO ACCUMULATOR CTRL LATCH ON AND LOWER ACC CTRL LATCH ON
DX
NEWP
TLU P.S. REGEN CTRL LATCH OFF
TLU PROG ADD LATCH ON

TLU PROGRAM STEP READ-IN AND REGENERATION CONTROLS

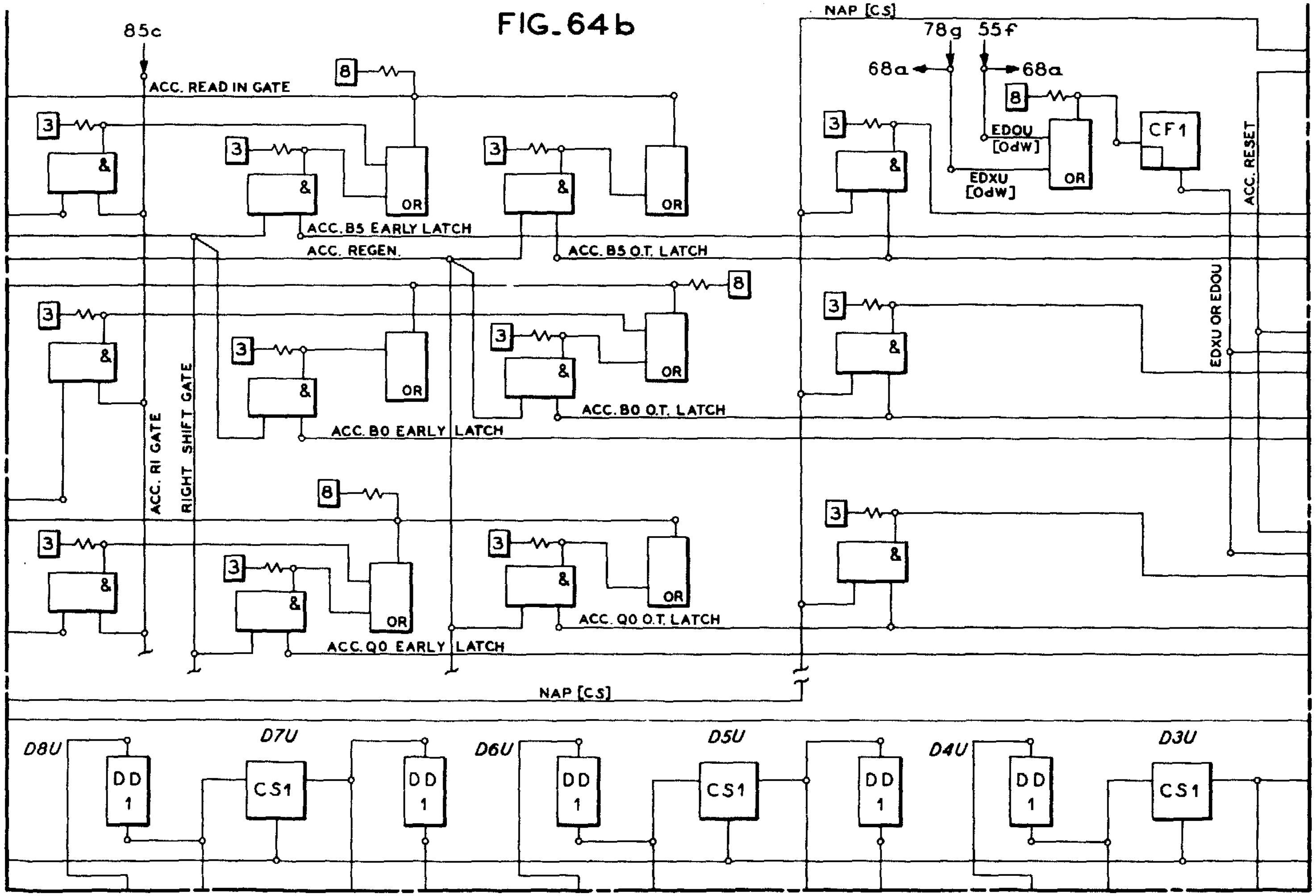
FIG. 63

FIG. 64a


85c
85f
ACC. REGEN. GATE
RIGHT SHIFT GATE
B5
ADDER OUTPUT
68c
B5
62c
62c
62c
62c
62c
62c
62c
B0
CF2
9
CF2
9
B0
Q0
Q0
Q1
Q2
Q3
Q4
Q1
Q2
Q3
Q4
ODD DIGIT
64e
EVEN DIGIT
64e
NAP (CS)
RO DXL
D10U
D9U
3
&
CF16
9
5
DD
1
DD
1
CS1
DD
1

DATA STORAGE AND PROCESSING MACHINE

FIG_64b



Nov. 8, 1960

F. E. HAMILTON ET AL

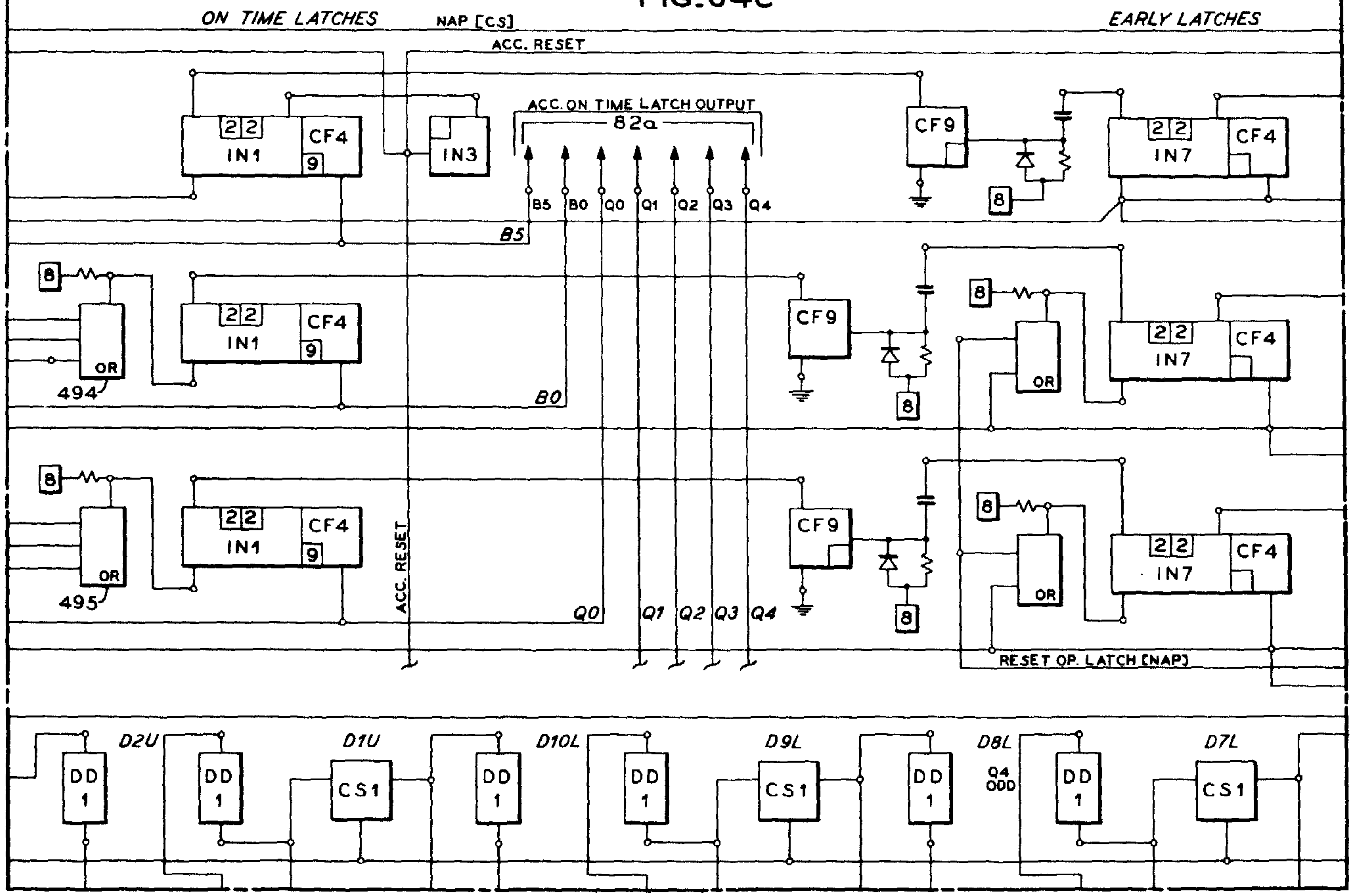
2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

228 Sheets-Sheet 87

FIG. 64c



Nov. 8, 1960

F. E. HAMILTON ET AL

2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

228 Sheets-Sheet 88

FIG_64d

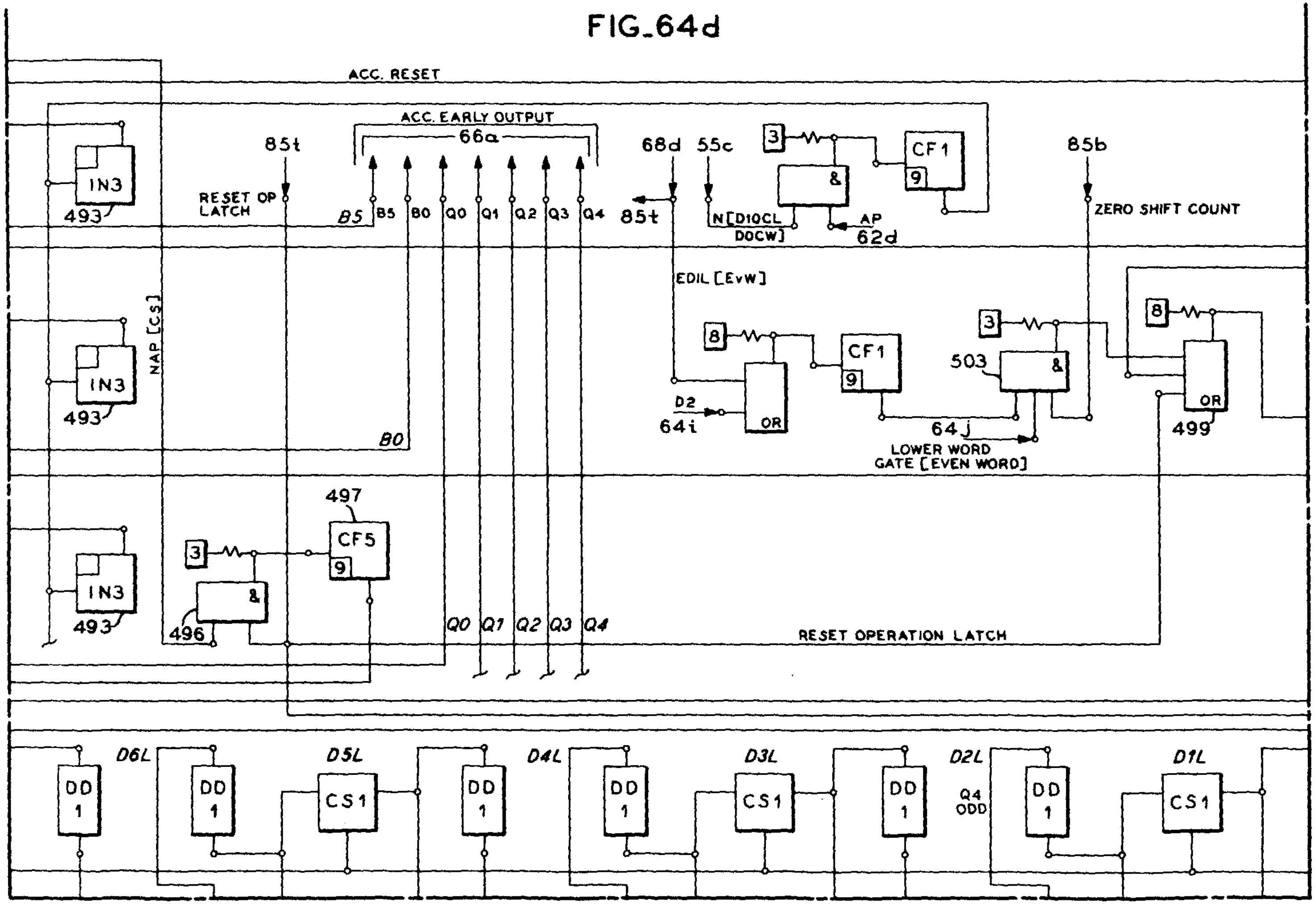
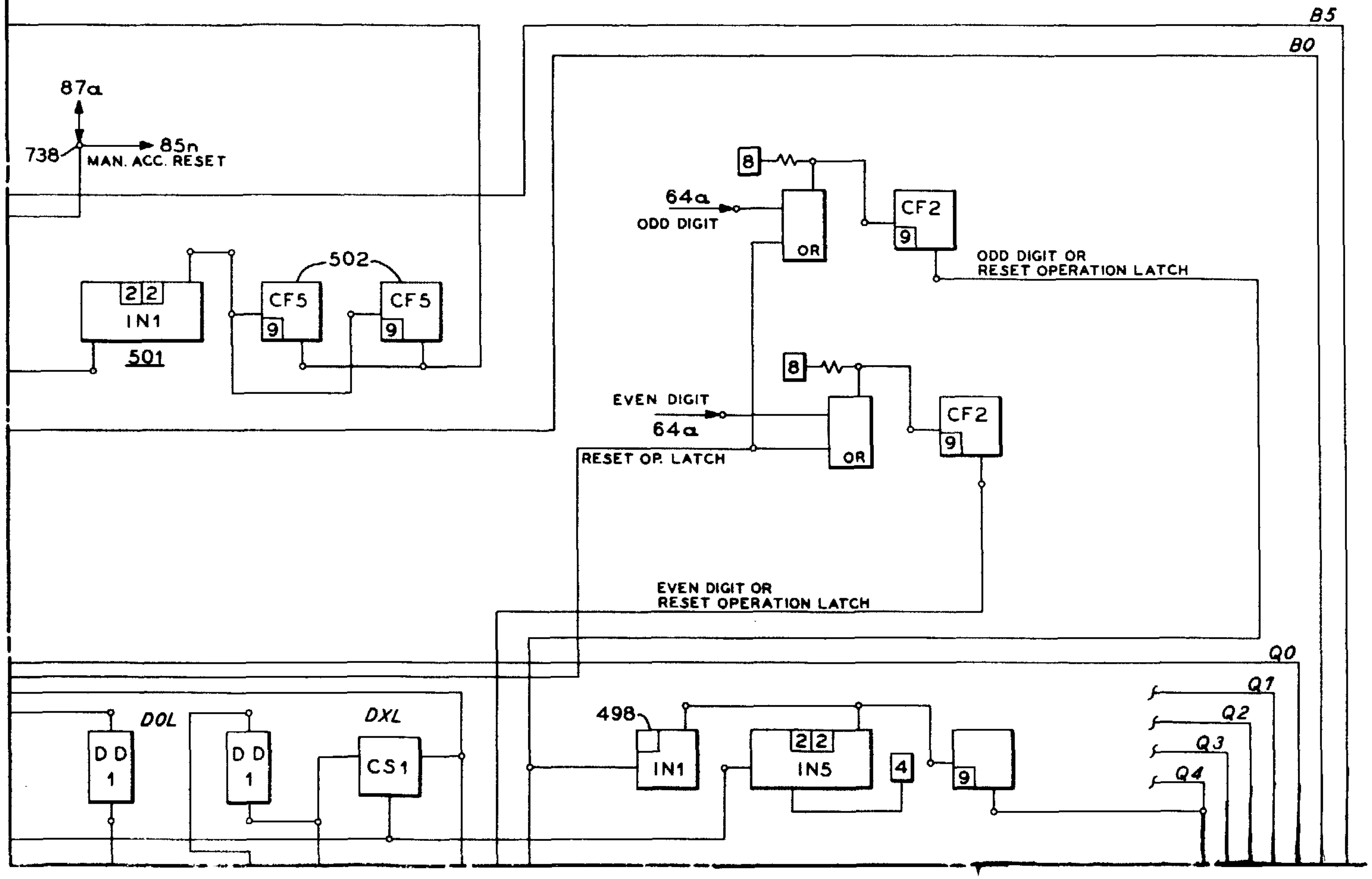


FIG_64e


B5
B0
87a
738
85n
MAN. ACC. RESET
2 2
IN1
501
502
CF5
9
CF5
9
8
64a
ODD DIGIT
OR
CF2
9
ODD DIGIT OR
RESET OPERATION LATCH
8
EVEN DIGIT
64a
OR
CF2
9
RESET OP. LATCH
EVEN DIGIT OR
RESET OPERATION LATCH
Q0
Q1
Q2
Q3
Q4
DOL
DD
1
DD
1
DXL
CS1
498
IN1
2 2
IN5
4
9

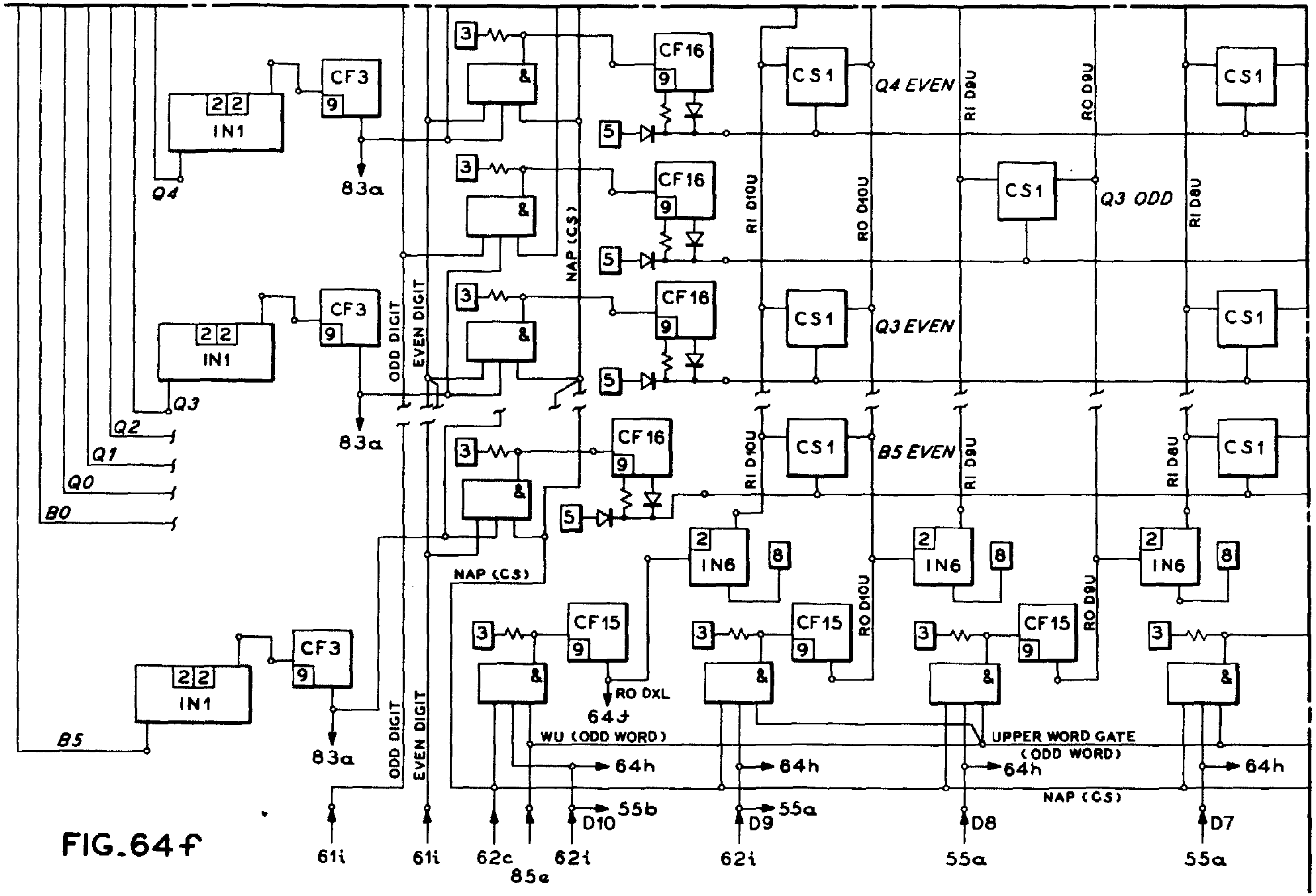

3
&
CF16
9
5
CS1
Q4 EVEN
RI D9U
RO D9U
CS1
22
IN1
CF3
9
Q4
83a
3
&
CF16
9
5
NAP (CS)
RI D10U
RO D10U
CS1
Q3 ODD
RI D8U
3
&
CF16
9
5
CS1
Q3 EVEN
CS1
22
IN1
CF3
9
Q3
83a
ODD DIGIT
EVEN DIGIT
Q2
Q1
Q0
B0
3
&
CF16
9
5
RI D10U
CS1
B5 EVEN
RI D9U
RI D8U
CS1
2
IN6
8
2
IN6
8
2
IN6
8
NAP (CS)
RO D10U
RO D9U
22
IN1
CF3
9
B5
83a
3
CF15
9
&
3
CF15
9
&
3
CF15
9
&
3
&
RO DXL
64J
WU (ODD WORD)
UPPER WORD GATE
(ODD WORD)
64h
64h
64h
64h
NAP (CS)
ODD DIGIT
EVEN DIGIT
55b
D10
55a
D9
D8
D7
61i
61i
62c
85e
62i
62i
55a
55a

FIG. 64f

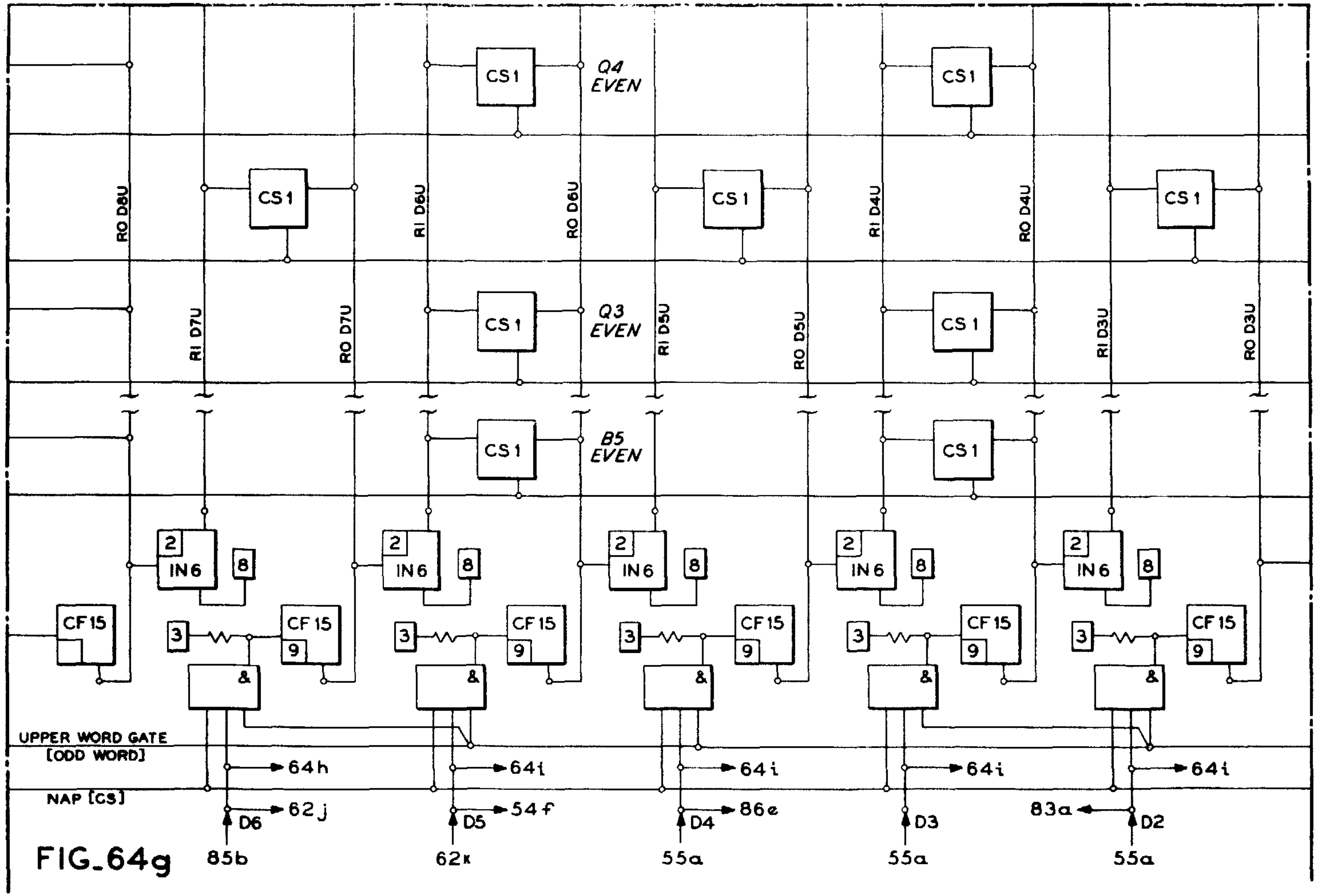

CS1
Q4
EVEN
CS1
RO D8U
RI D6U
RO D6U
RI D4U
RO D4U
Q3
EVEN
RI D7U
RO D7U
RI D5U
RO D5U
RI D3U
RO D3U
B5
EVEN
2
IN6
8
CF15
3
9
&
UPPER WORD GATE
[ODD WORD]
NAP [CS]
64h
64i
62j
54f
86e
83a
D6
D5
D4
D3
D2
85b
62k
55a

FIG_64g

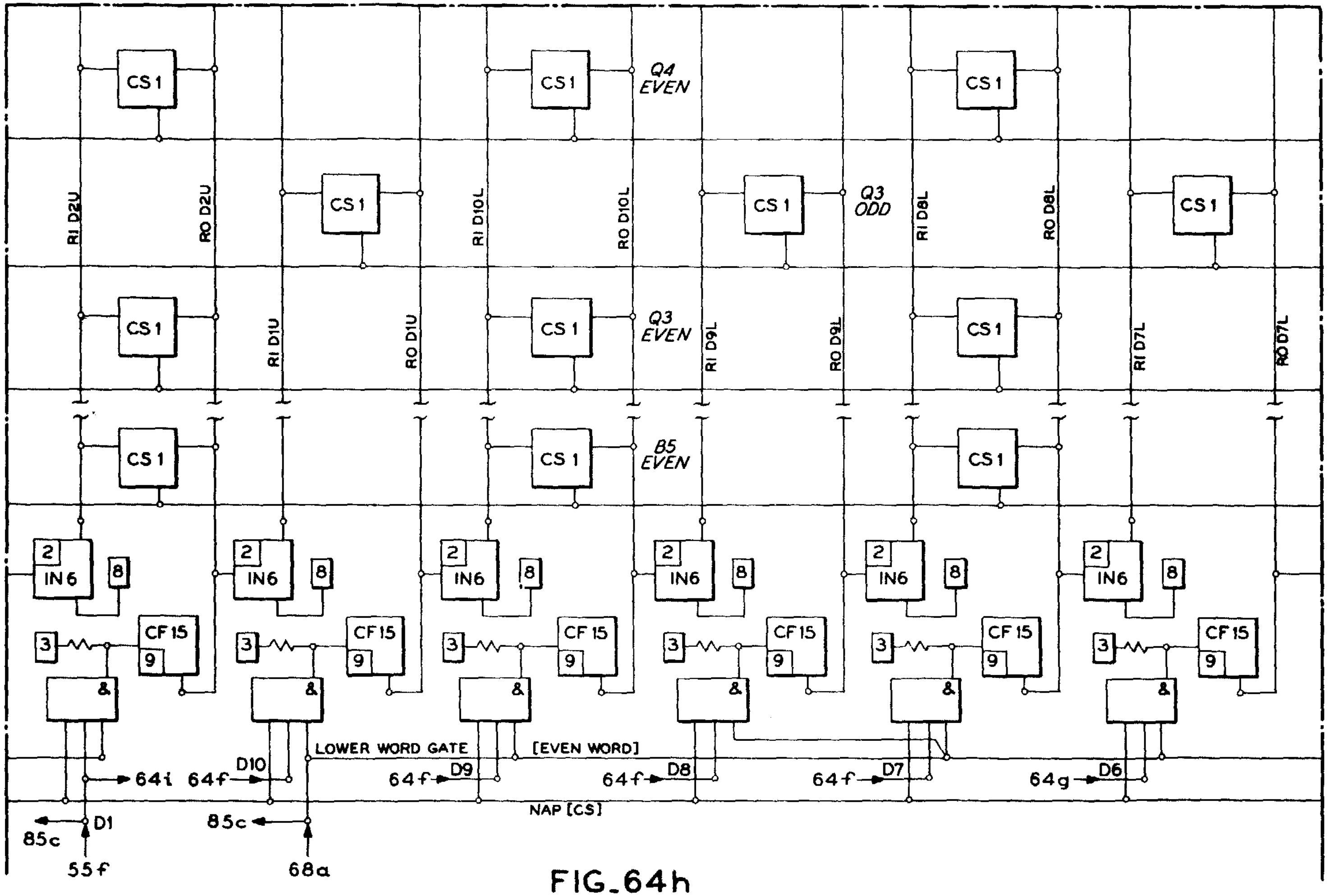

CS 1
Q4
EVEN
CS 1
Q3
ODD
CS 1
Q3
EVEN
CS 1
B5
EVEN
RI D2U
RO D2U
RI D1U
RO D1U
RI D10L
RO D10L
RI D9L
RO D9L
RI D8L
RO D8L
RI D7L
RO D7L
2
IN6
8
3
CF 15
9
&
LOWER WORD GATE
[EVEN WORD]
NAP [CS]
64i
64f
D10
64f
D9
64f
D8
64f
D7
64g
D6
D1
85c
85c
55f
68a

FIG. 64h

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 93

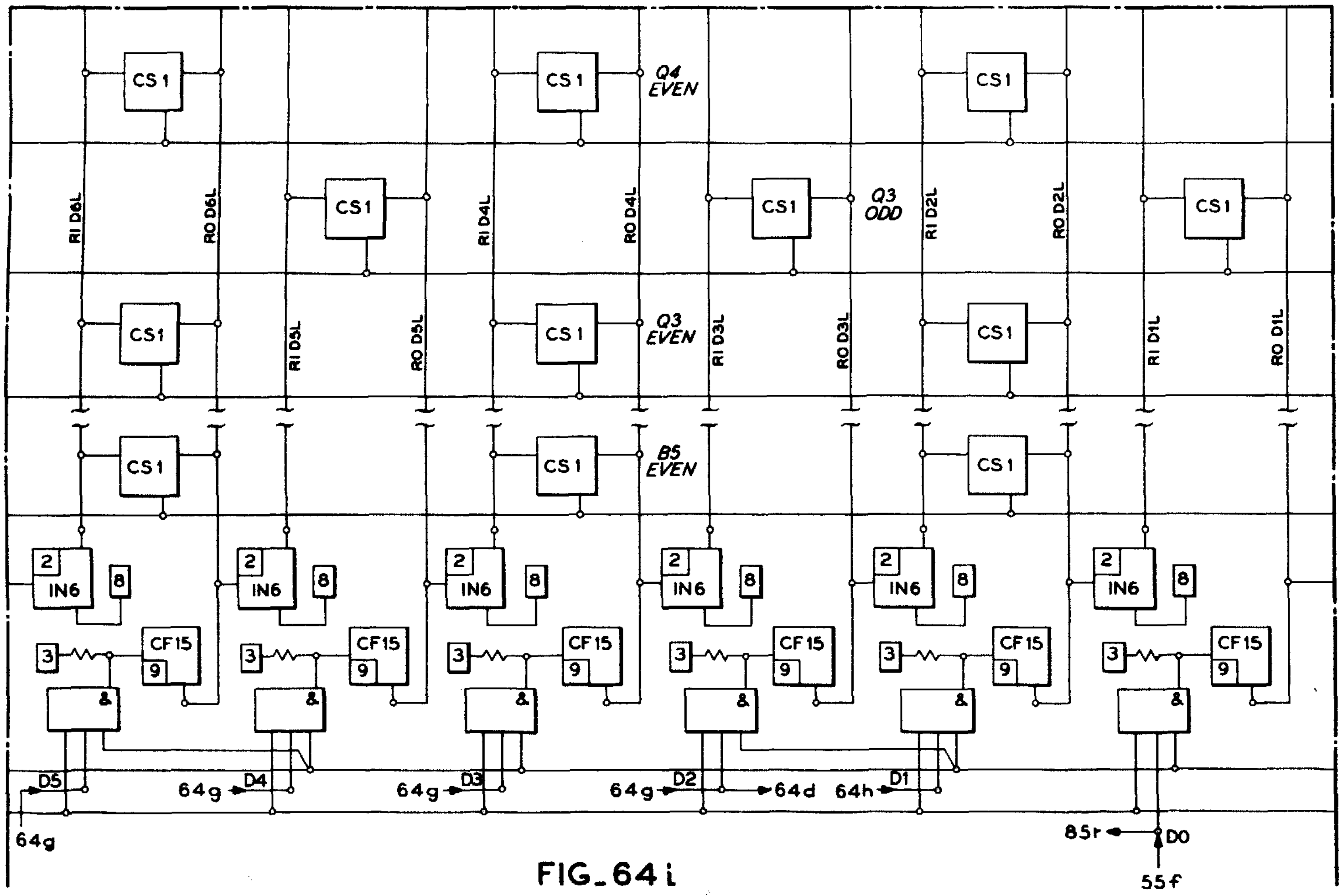


FIG_64 i

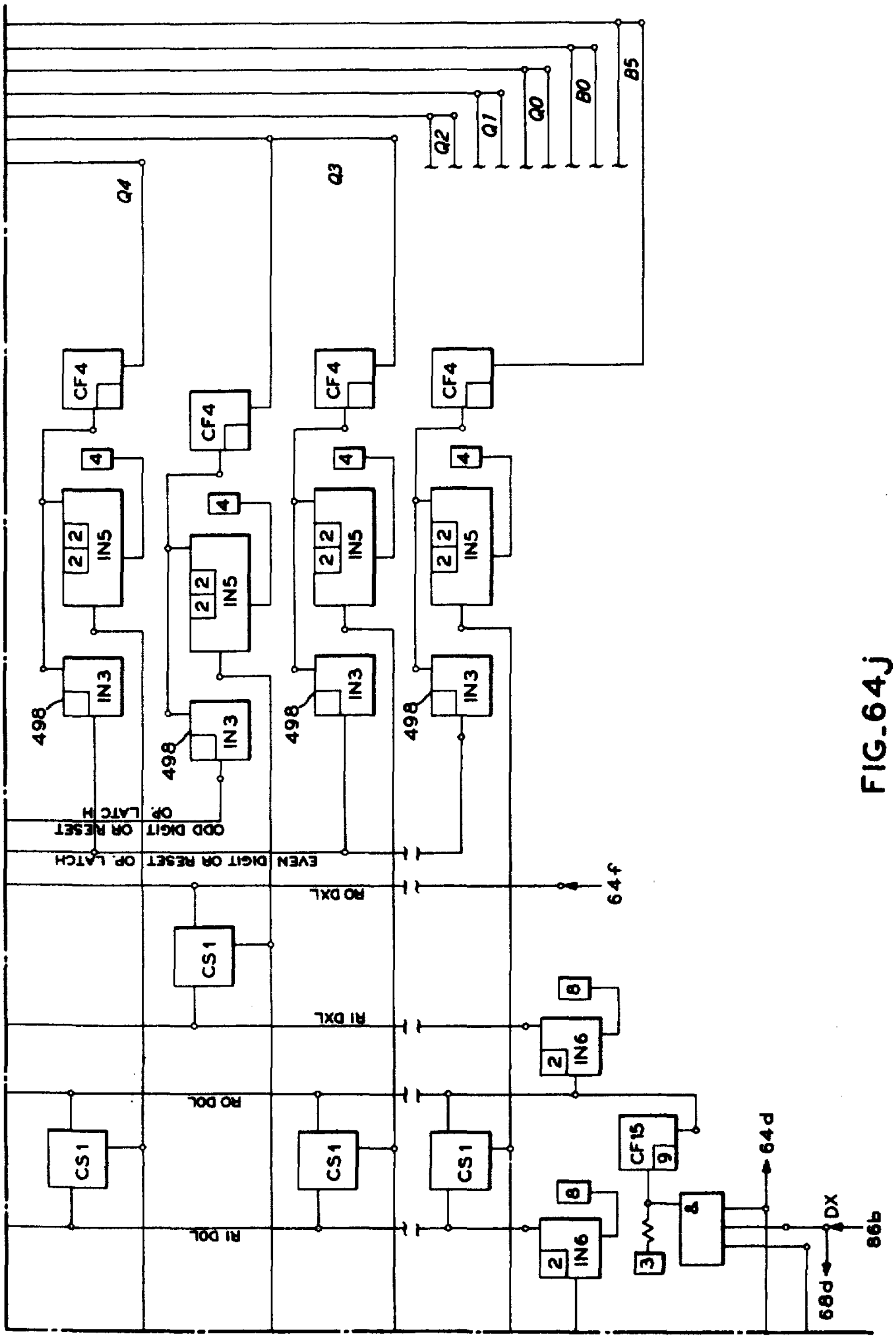
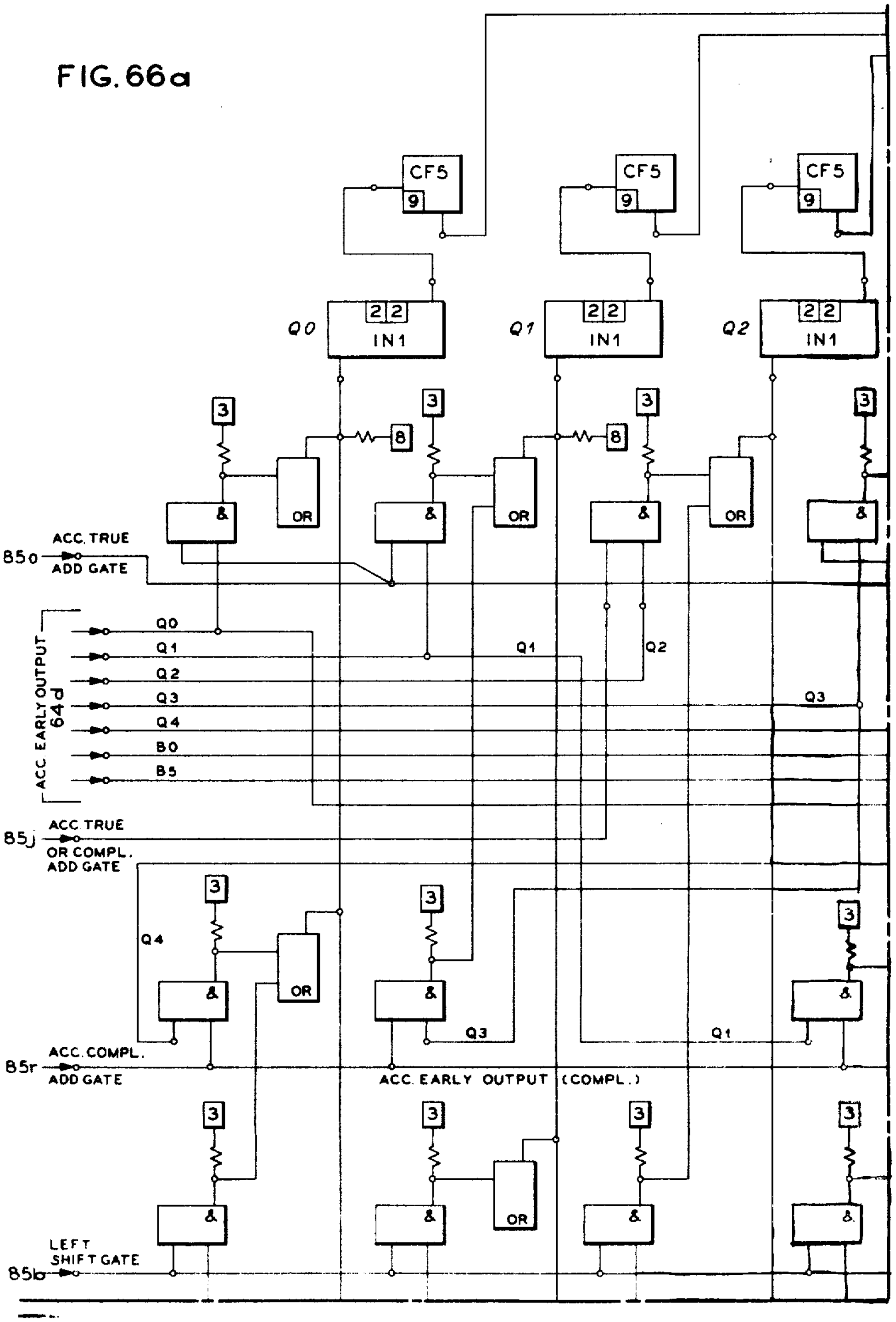

Q4
Q3
Q2
Q1
Q0
B0
B5
CF4
CF4
CF4
CF4
4
4
4
4
2 2 IN5
2 2 IN5
2 2 IN5
2 2 IN5
IN3
IN3
IN3
IN3
498
498
498
498
OP. LATCH
ODD DIGIT OR RESET
EVEN DIGIT OR RESET OP. LATCH
RO DXL
64f
CS1
RI DXL
8
2 IN6
RO DOL
CS1
CS1
CS1
CF15
9
64d
&
DX
86b
68d
3
8
2 IN6
RI DOL

FIG_64j

FIG. 66a


CF5
9
CF5
9
CF5
9
Q0
2 2
IN1
Q1
2 2
IN1
Q2
2 2
IN1
3
8
3
8
3
3
&
OR
&
OR
&
OR
&
85o
ACC. TRUE
ADD GATE
ACC. EARLY OUTPUT
64d
Q0
Q1
Q2
Q3
Q4
B0
B5
Q1
Q2
Q3
85j
ACC. TRUE
OR COMPL.
ADD GATE
Q4
3
3
3
&
OR
&
&
Q3
Q1
85r
ACC. COMPL.
ADD GATE
ACC. EARLY OUTPUT (COMPL.)
3
3
3
3
&
&
OR
&
&
85b
LEFT
SHIFT GATE

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 96

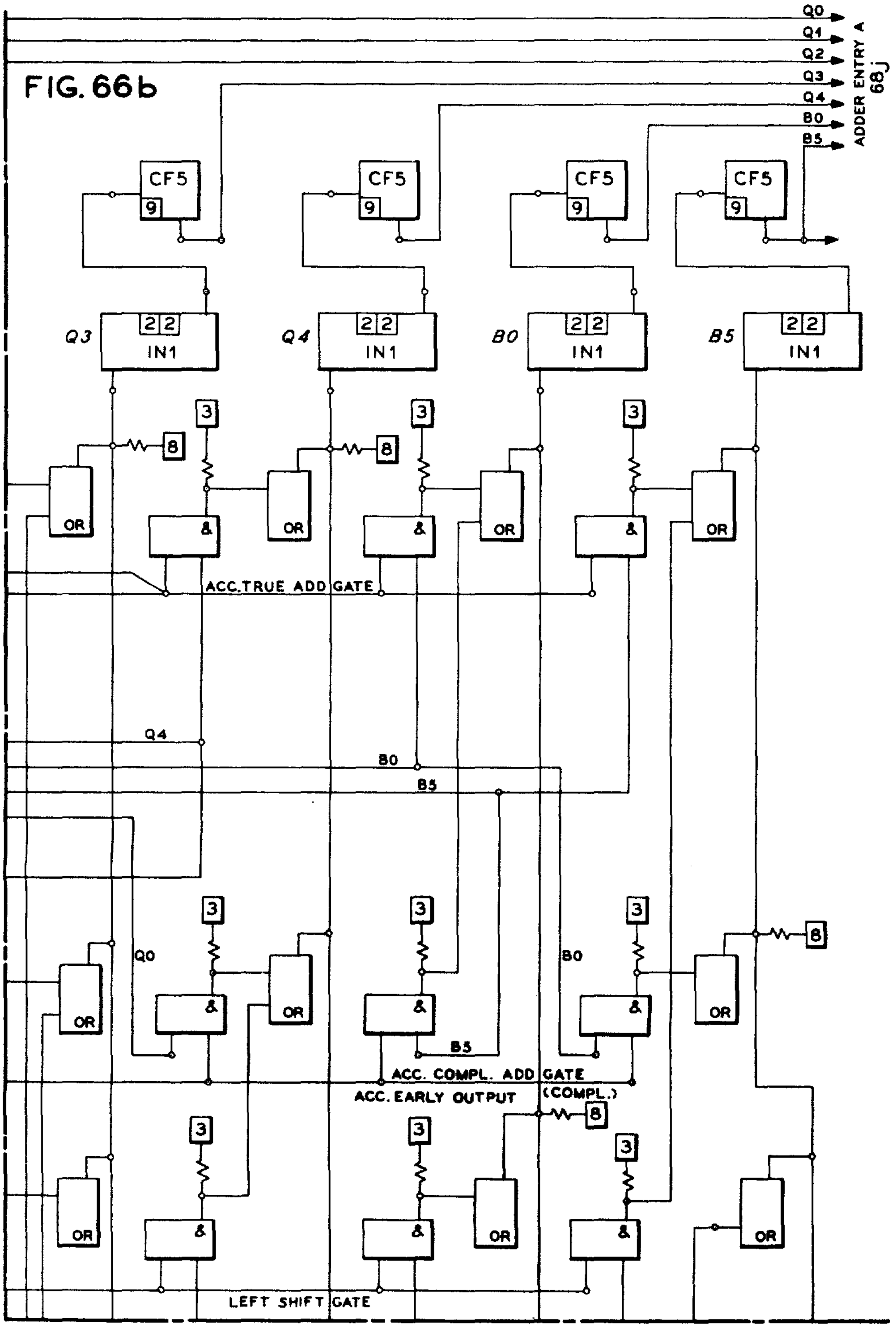


FIG. 66b

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 97

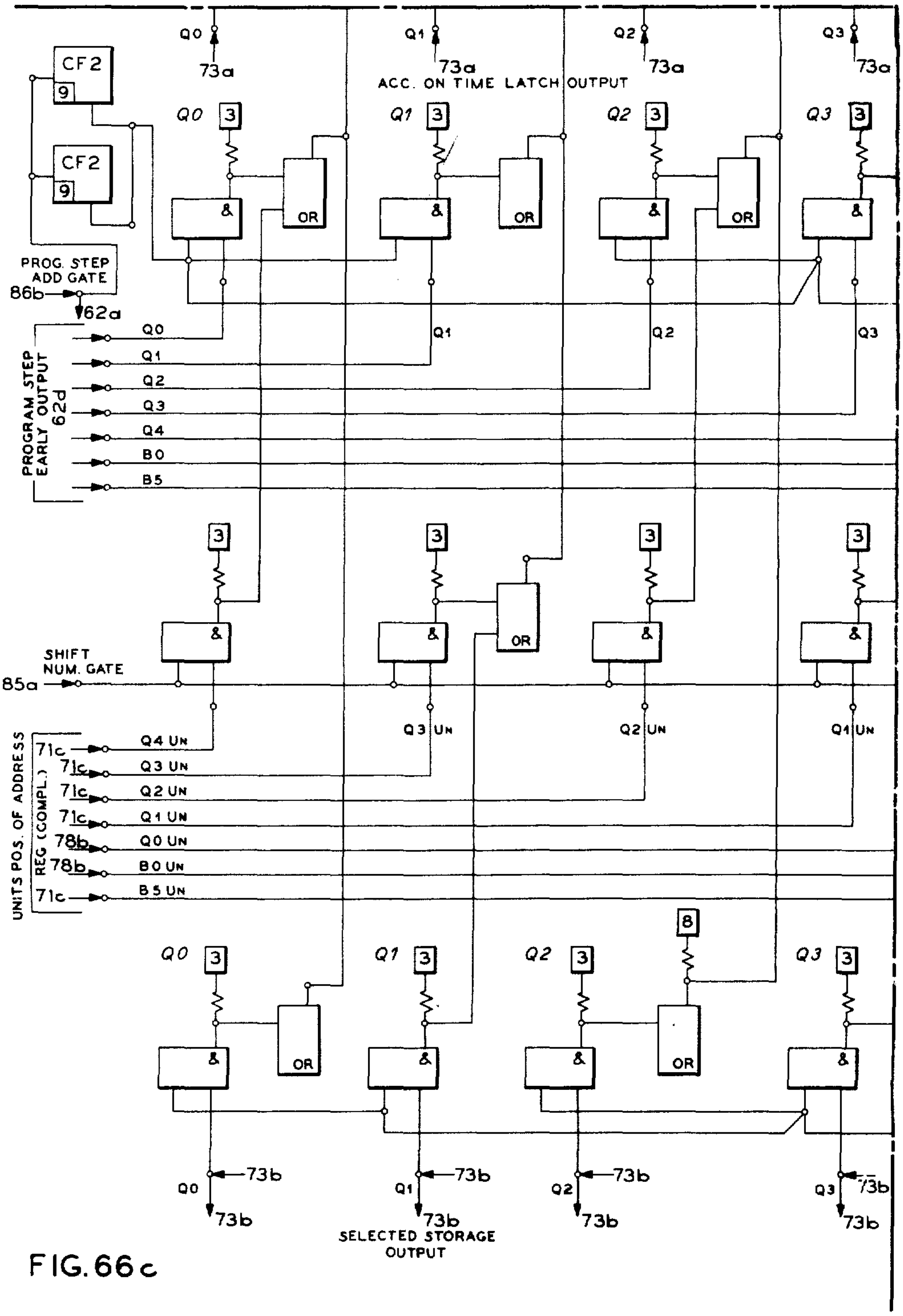


FIG. 66c

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

228 Sheets-Sheet 98

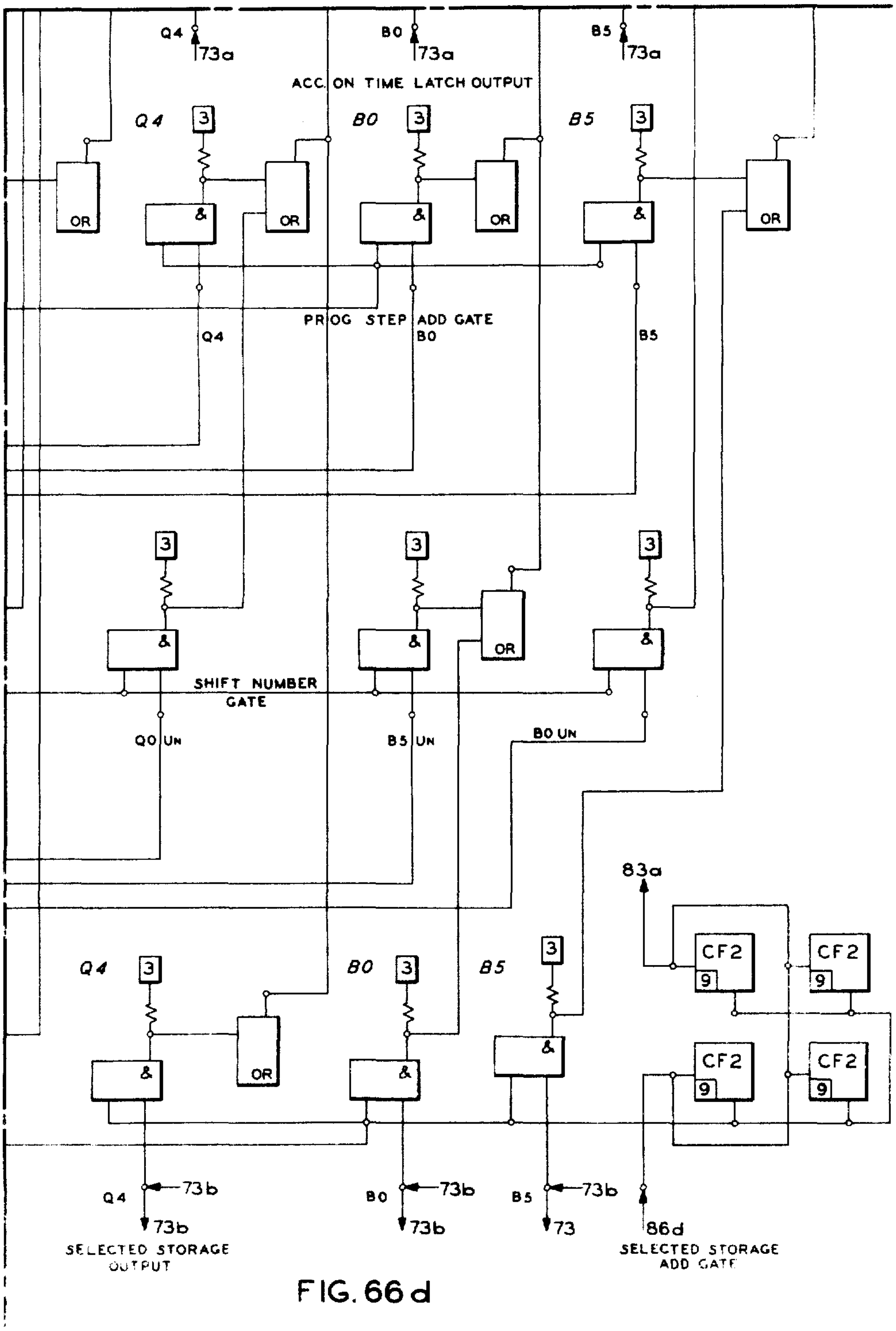


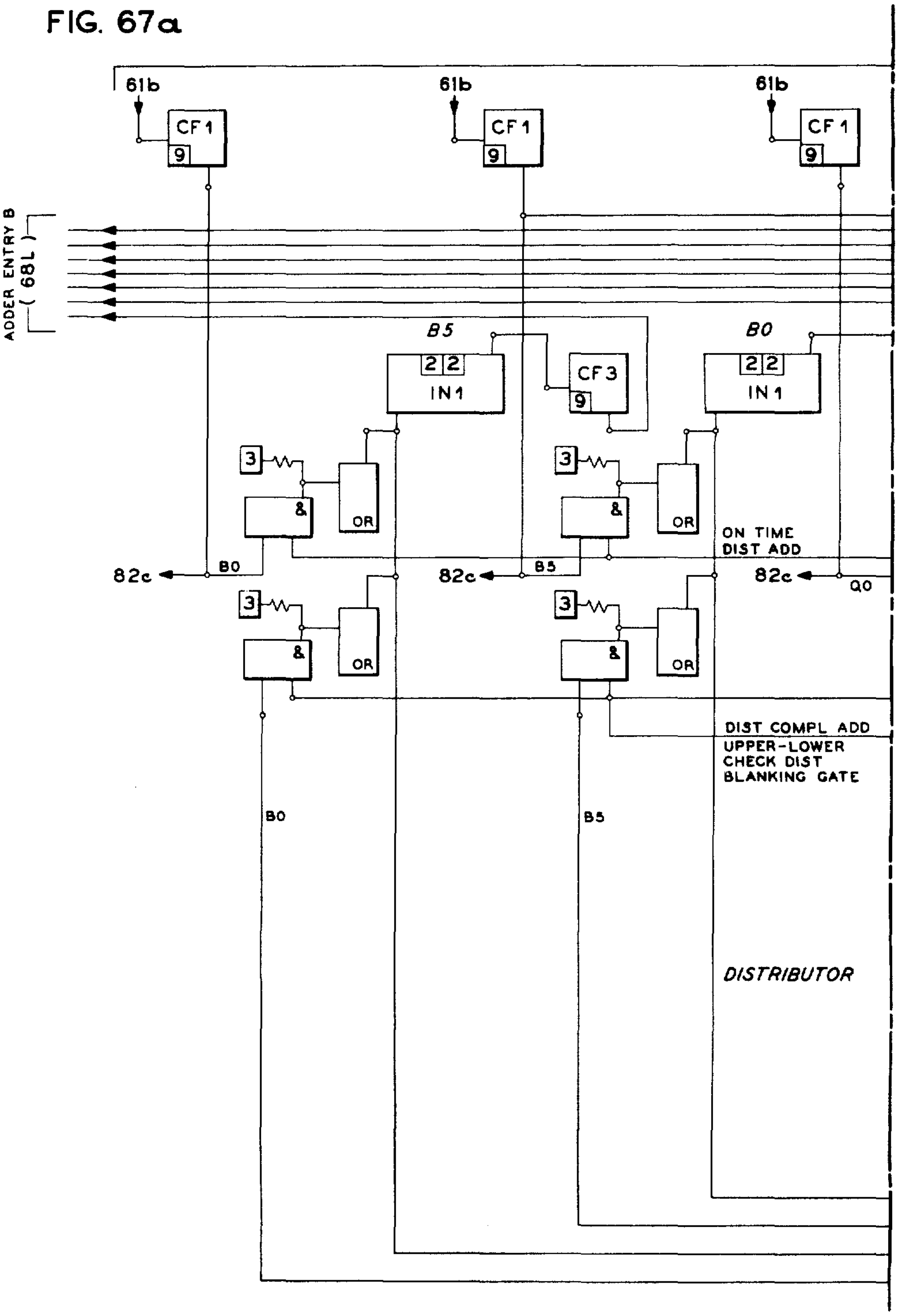
FIG. 66 d

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 99

FIG. 67a

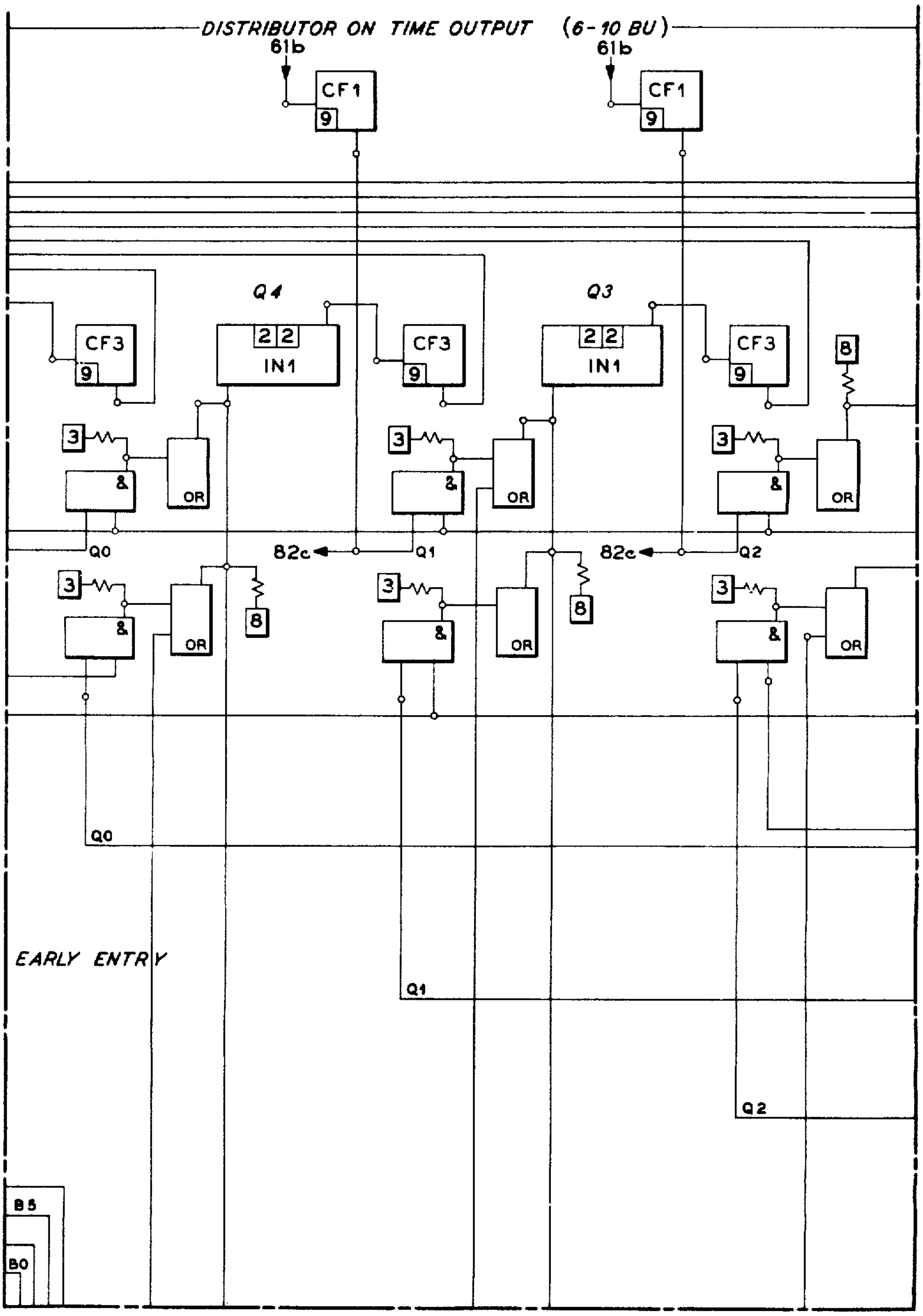


Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 100

FIG. 67b

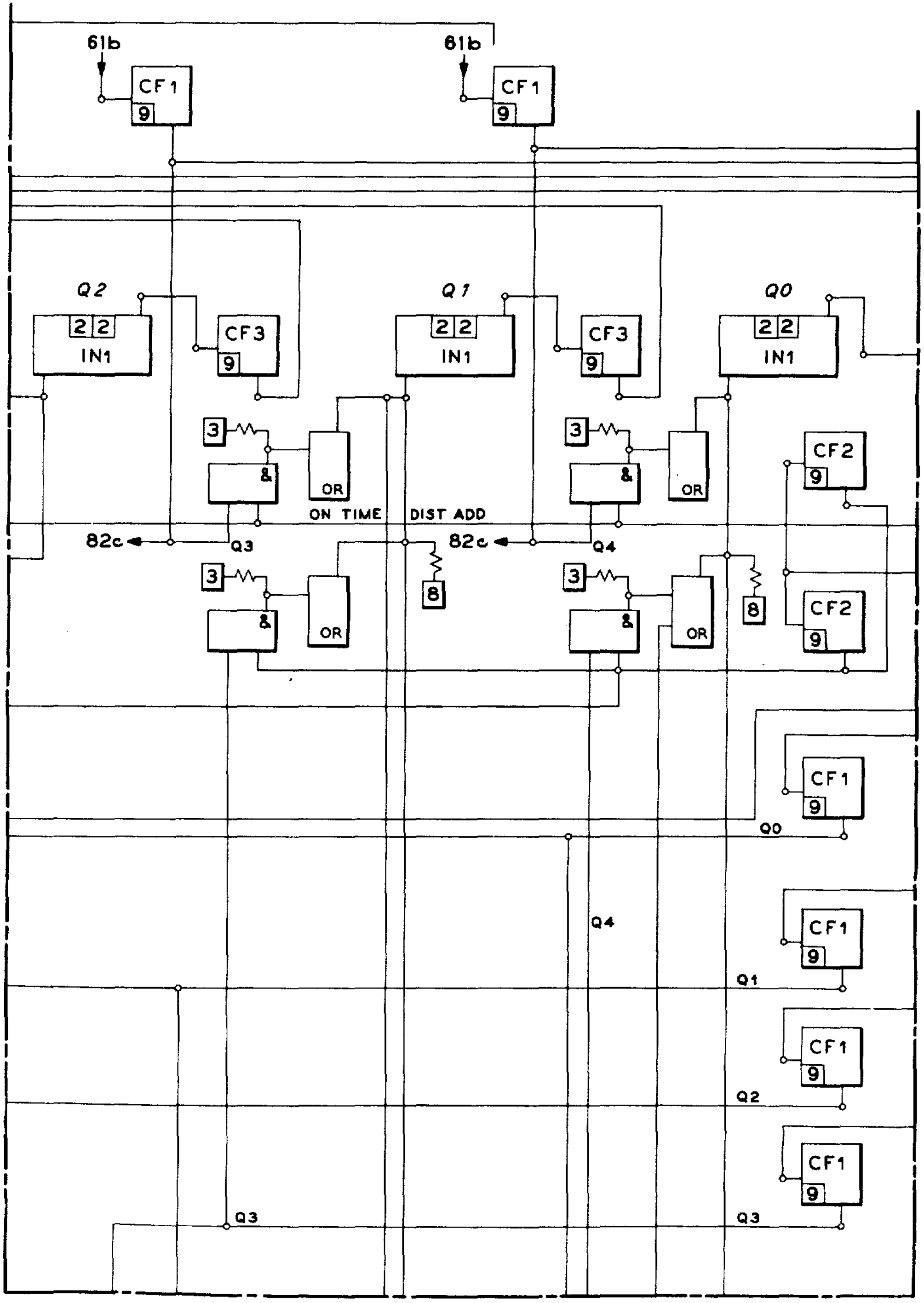


Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 101

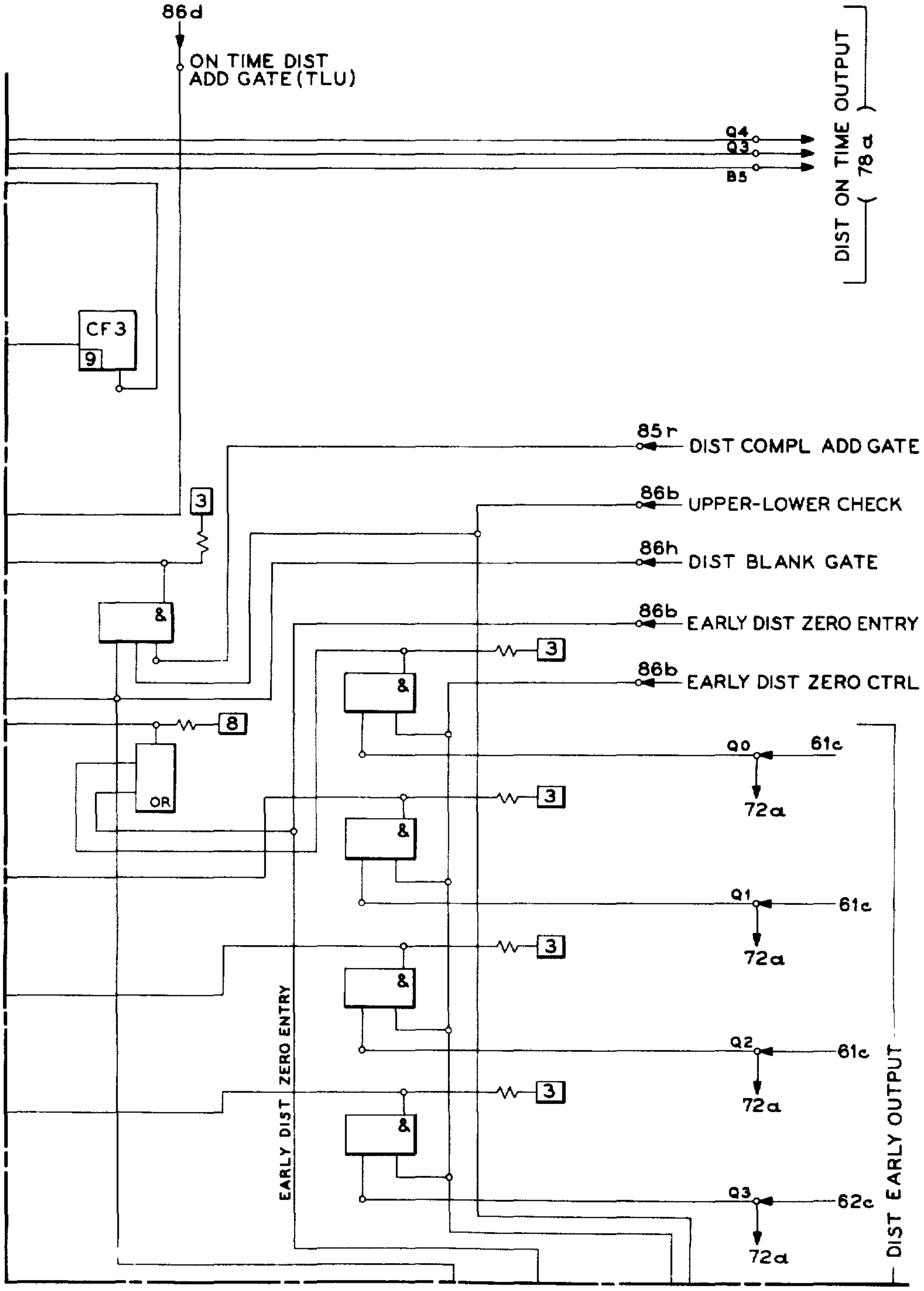
FIG. 67c



DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

FIG. 67d



Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 103

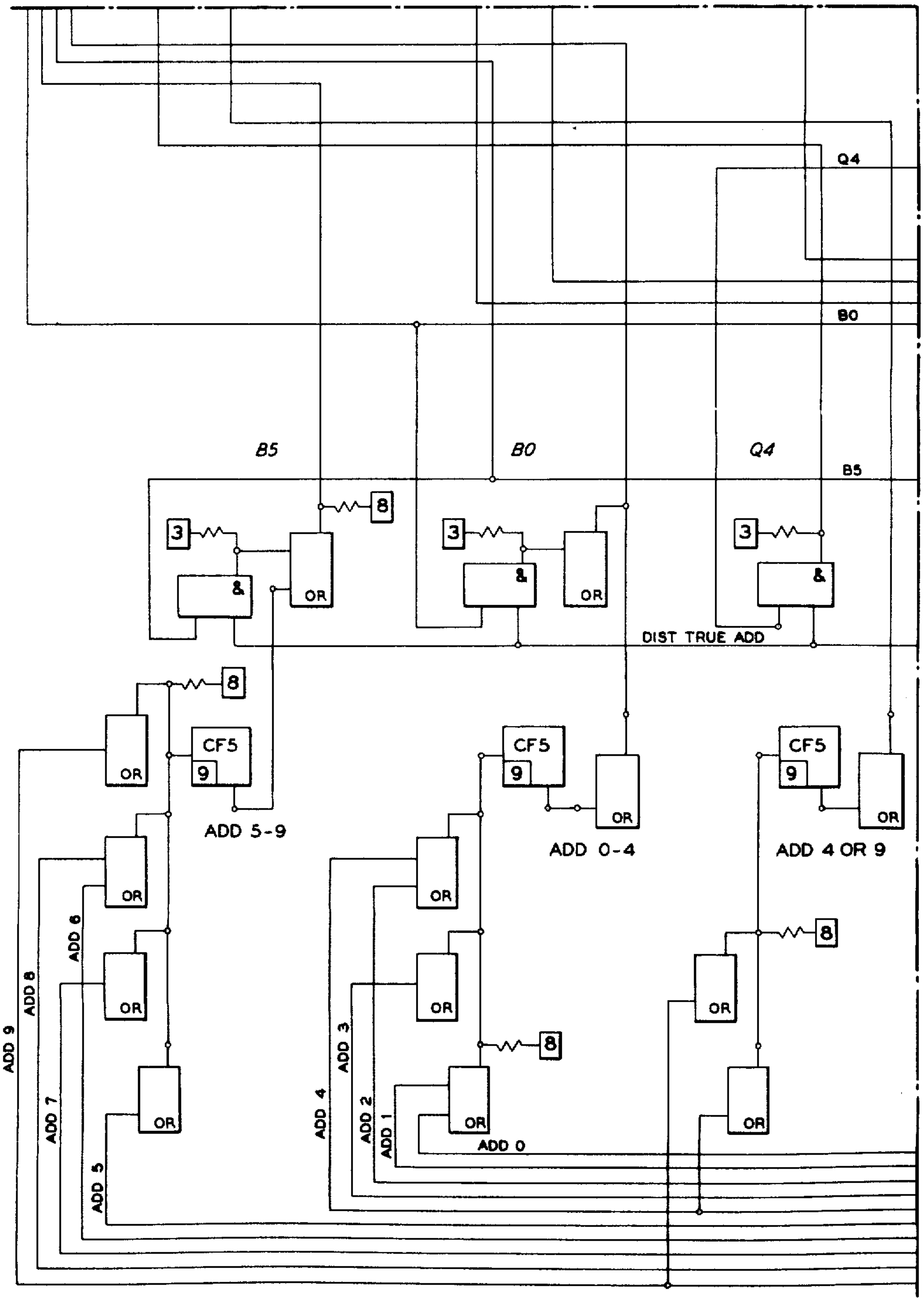


FIG. 67e

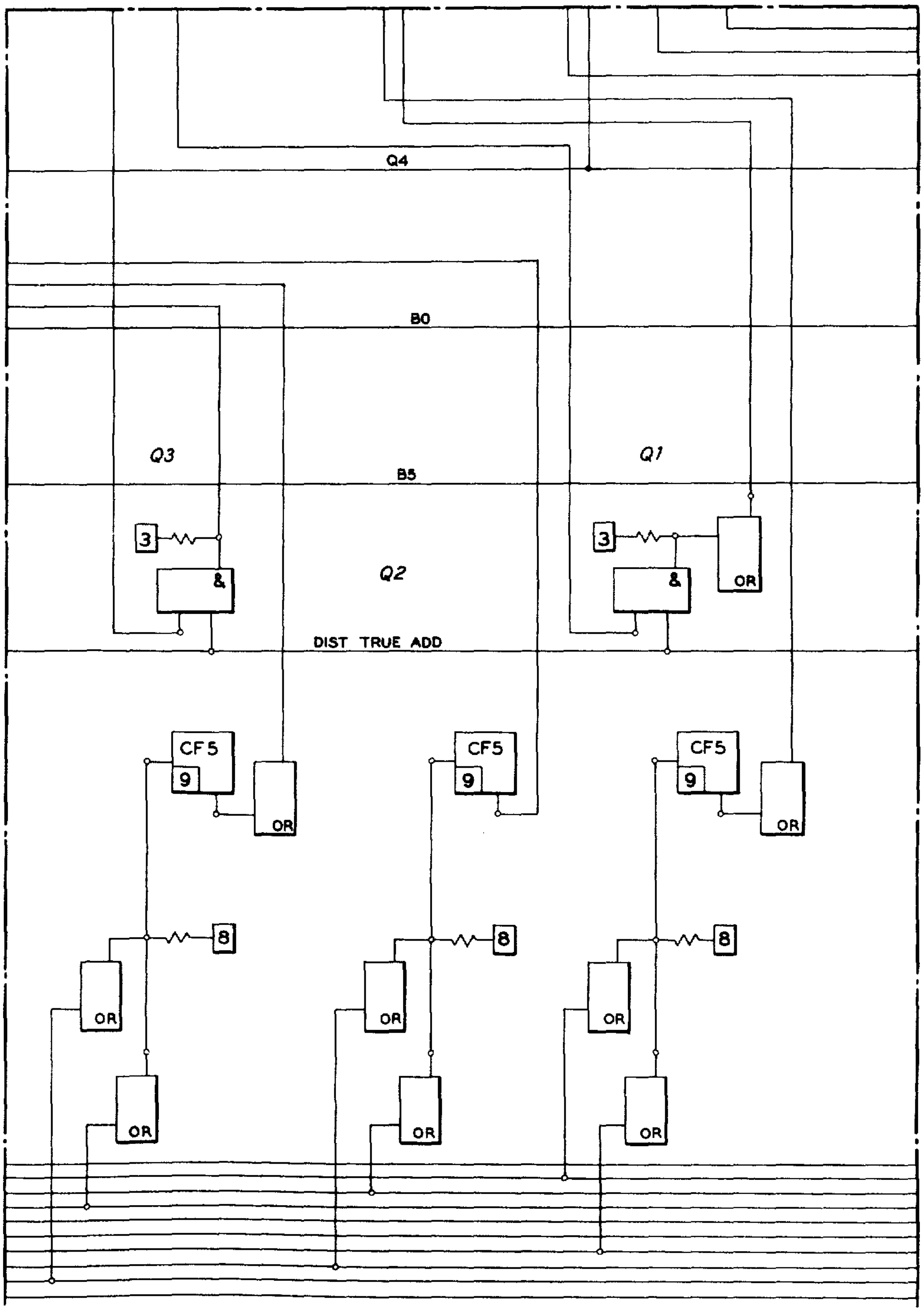

Q4
B0
Q3
Q1
B5
3
&
Q2
3
&
OR
DIST TRUE ADD
CF5
9
OR
CF5
9
CF5
9
OR
8
OR
OR
8
OR
OR
8
OR
OR

FIG. 67f

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 105

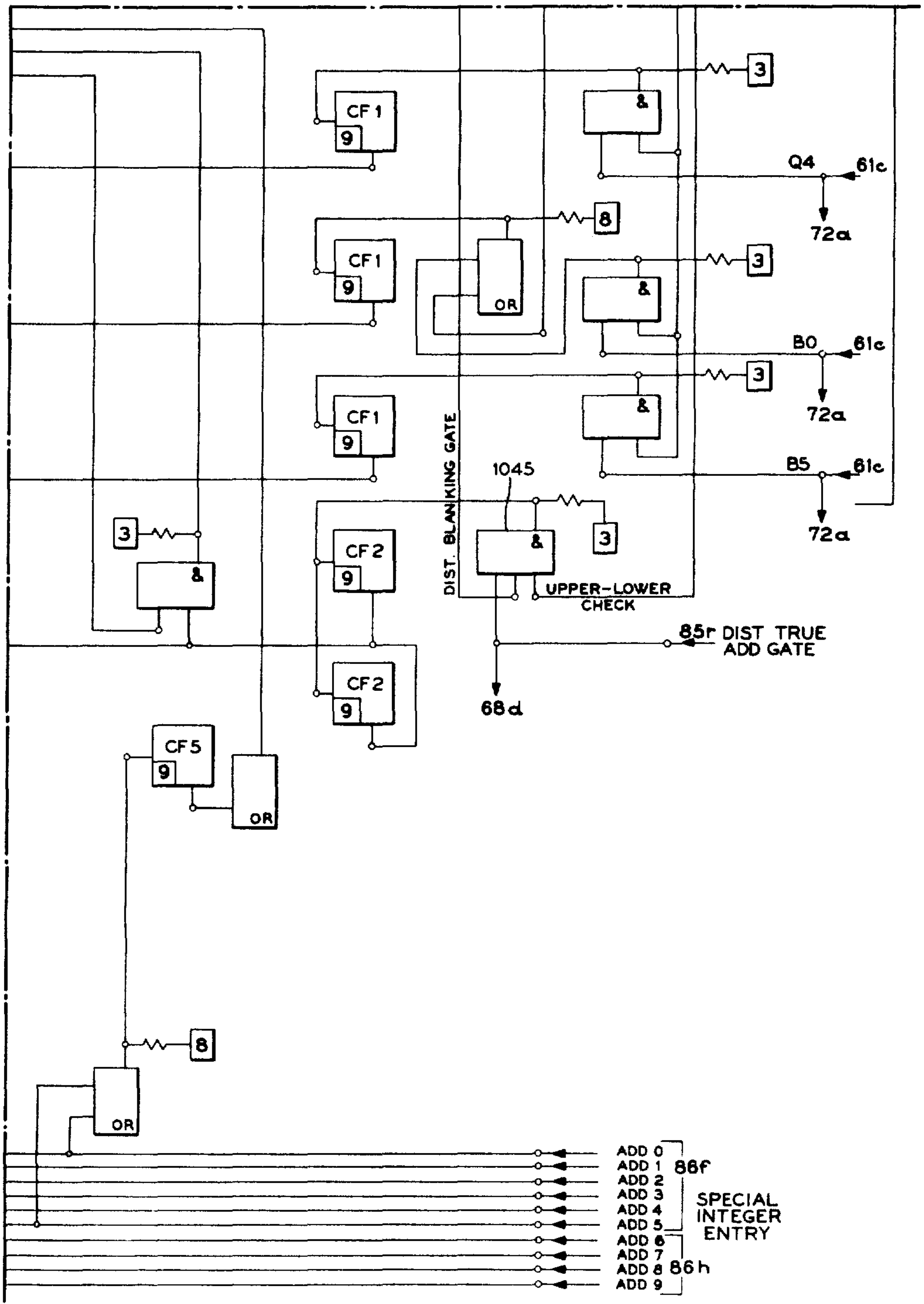


FIG. 67g

FIG_ 68a

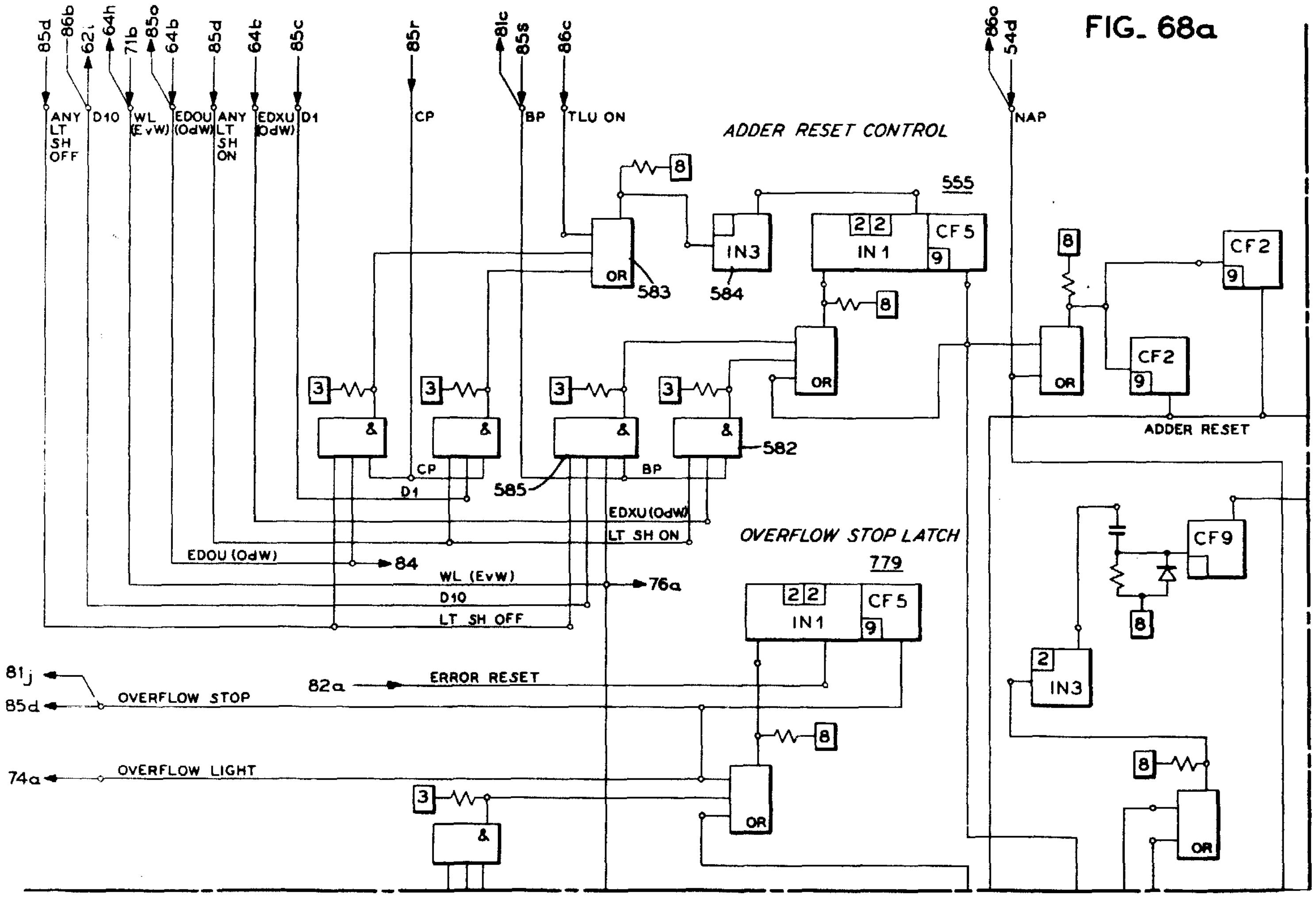

85d
86b
62i
64h
71b
85o
64b
85d
64b
85c
85r
81c
85s
86c
86o
54d
ANY LT SH OFF
D10
WL (EvW)
EDOU (OdW)
ANY LT SH ON
EDXU (OdW)
D1
CP
BP
TLU ON
NAP
ADDER RESET CONTROL
555
8
IN3
22
IN1
CF5
9
OR
583
584
CF2
3
&
582
585
EDXU (OdW)
LT SH ON
EDOU (OdW)
84
WL (EvW)
76a
D10
LT SH OFF
ADDER RESET
OVERFLOW STOP LATCH
779
CF9
2
81j
85d
OVERFLOW STOP
82a
ERROR RESET
74a
OVERFLOW LIGHT

FIG. 68b

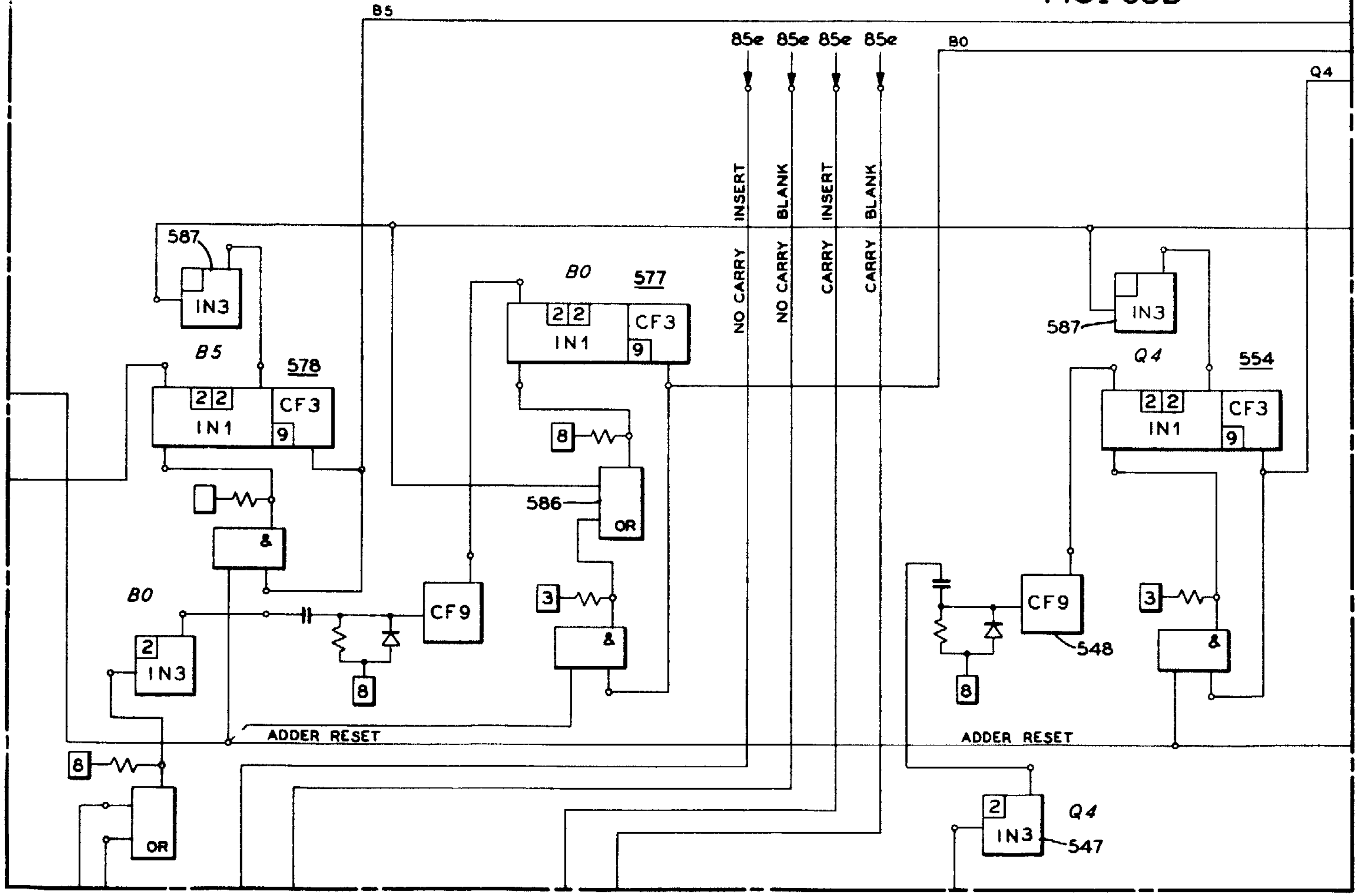
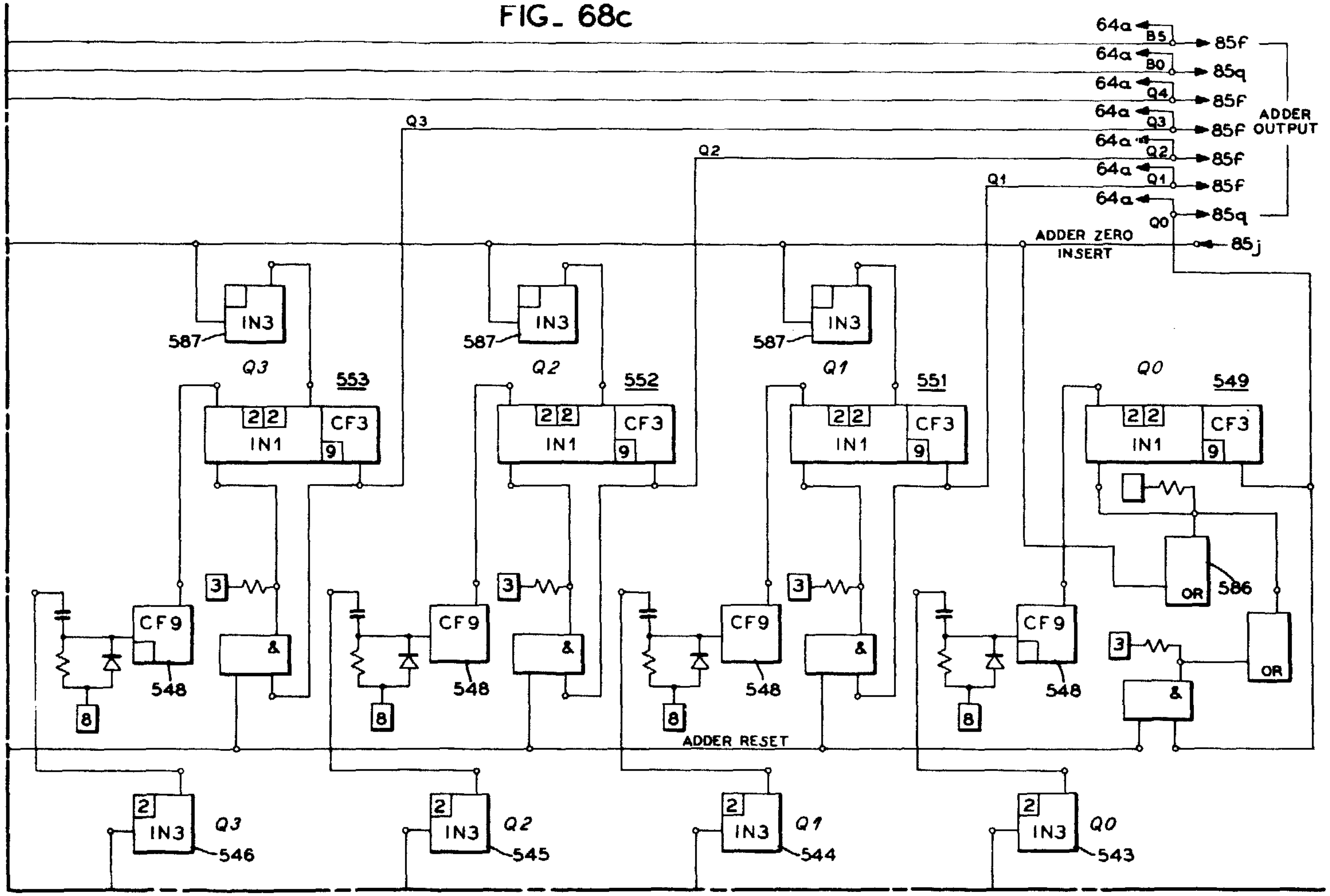

B5
85e
85e
85e
85e
B0
Q4
INSERT
BLANK
INSERT
BLANK
NO CARRY
NO CARRY
CARRY
CARRY
587
IN3
B5
578
2 2
IN1
CF3
9
B0
577
2 2
IN1
CF3
9
8
586
OR
&
3
&
B0
2
IN3
CF9
8
ADDER RESET
8
OR
587
IN3
Q4
554
2 2
IN1
CF3
9
CF9
548
8
3
&
ADDER RESET
2
IN3
Q4
547

FIG. 68c


64a
B5
85f
64a
B0
85q
64a
Q4
85F
64a
Q3
85F
ADDER OUTPUT
64a
Q2
85F
64a
Q1
85F
64a
Q0
85q
ADDER ZERO INSERT
85j
Q3
Q2
Q1
IN3
587
553
552
551
549
Q0
2 2
IN1
CF3
9
3
CF9
&
8
548
OR
586
ADDER RESET
2
IN3
Q3
546
Q2
545
Q1
544
Q0
543

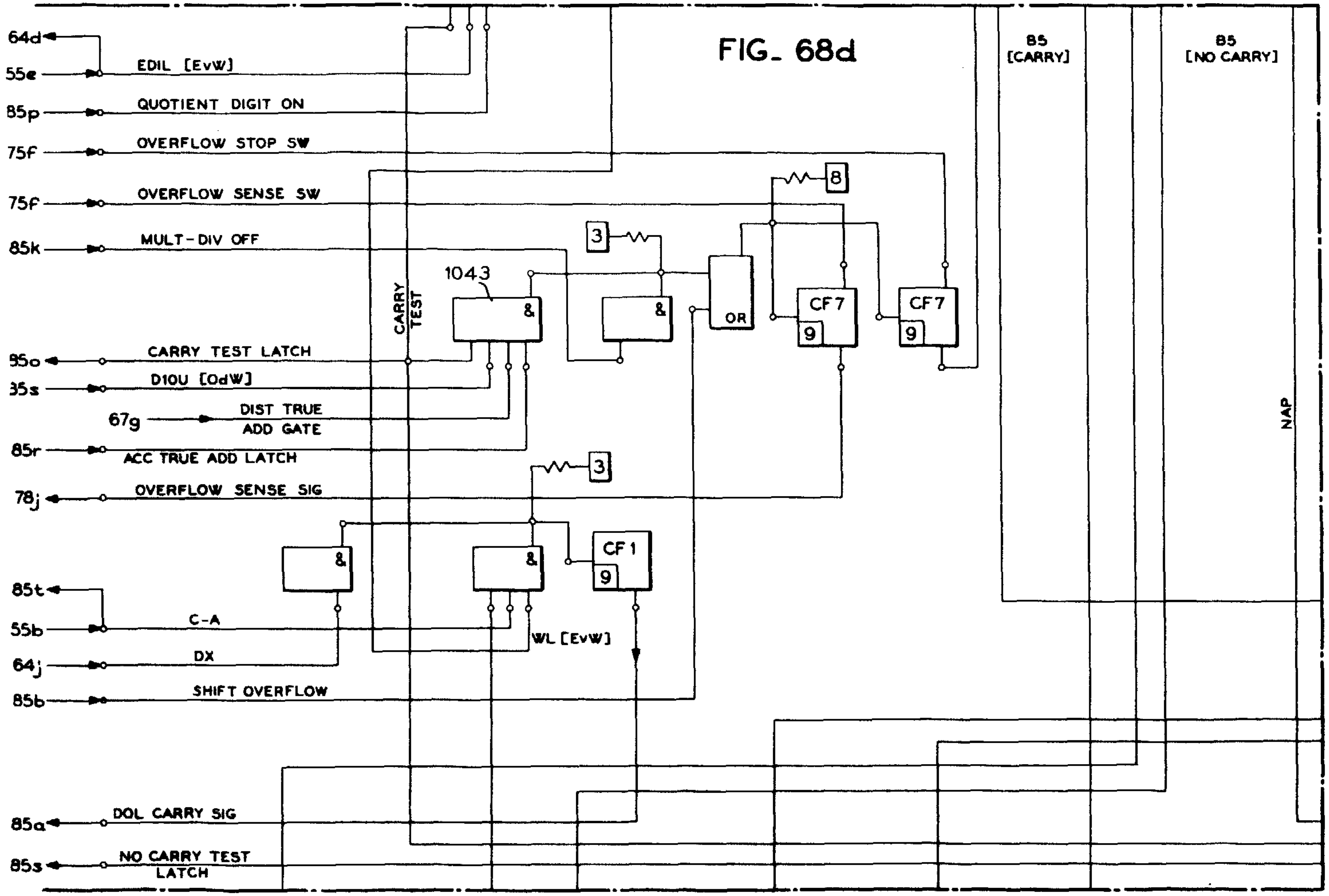

64d
55e
EDIL [EvW]
85p
QUOTIENT DIGIT ON
75f
OVERFLOW STOP SW
75f
OVERFLOW SENSE SW
85k
MULT-DIV OFF
CARRY TEST
1043
&
&
OR
3
8
CF7
9
CF7
9
85
[CARRY]
85
[NO CARRY]
NAP
85o
CARRY TEST LATCH
85s
D10U [OdW]
67g
DIST TRUE
ADD GATE
85r
ACC TRUE ADD LATCH
78j
OVERFLOW SENSE SIG
3
&
&
CF1
9
85t
55b
C-A
WL [EvW]
64j
DX
85b
SHIFT OVERFLOW
85a
DOL CARRY SIG
85s
NO CARRY TEST
LATCH

FIG_ 68d

Nov. 8, 1960

F. E. HAMILTON ET AL

2,959,351

Filed Nov. 2, 1955

DATA STORAGE AND PROCESSING MACHINE

228 Sheets-Sheet 110

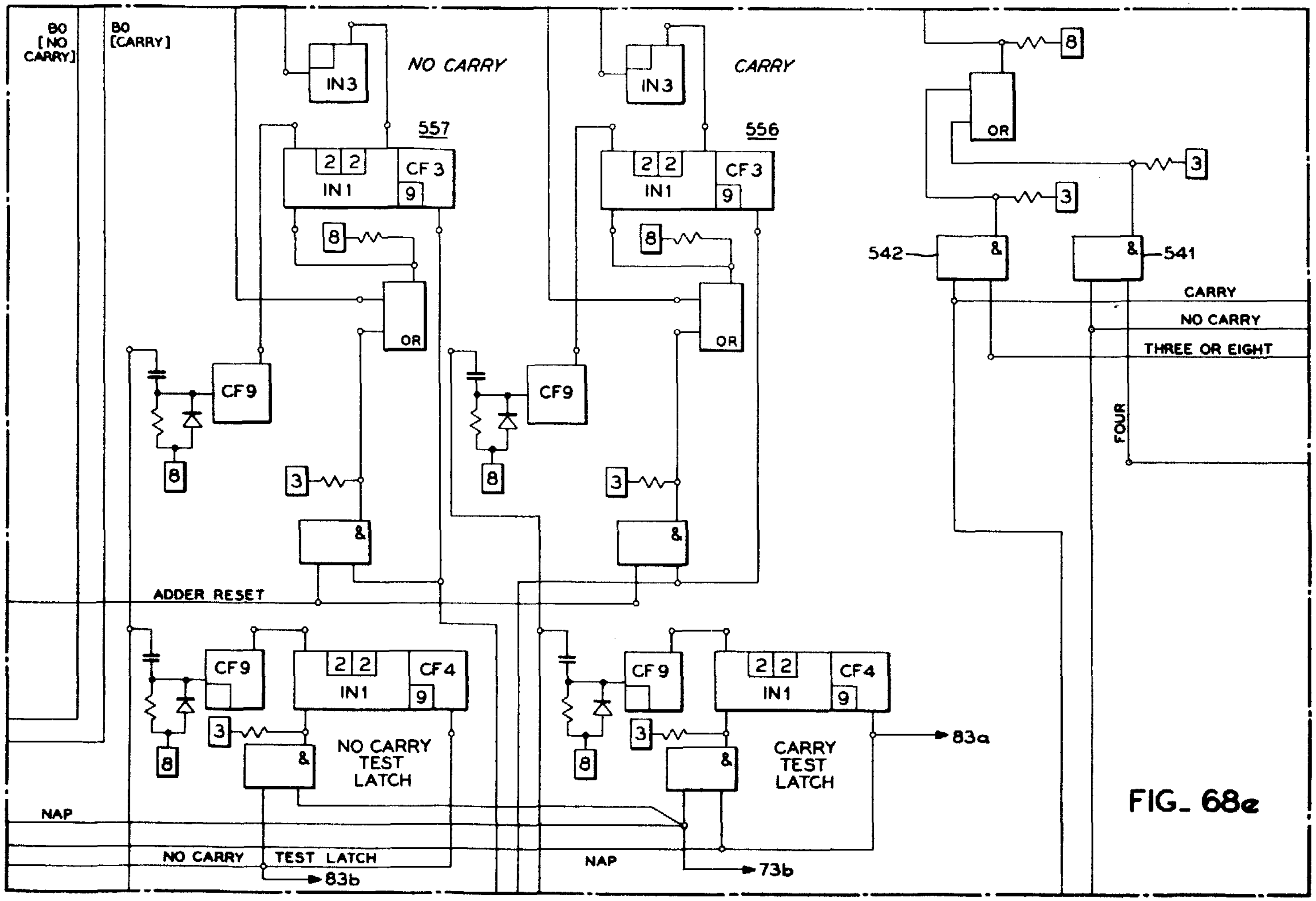


FIG_ 68e

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 111

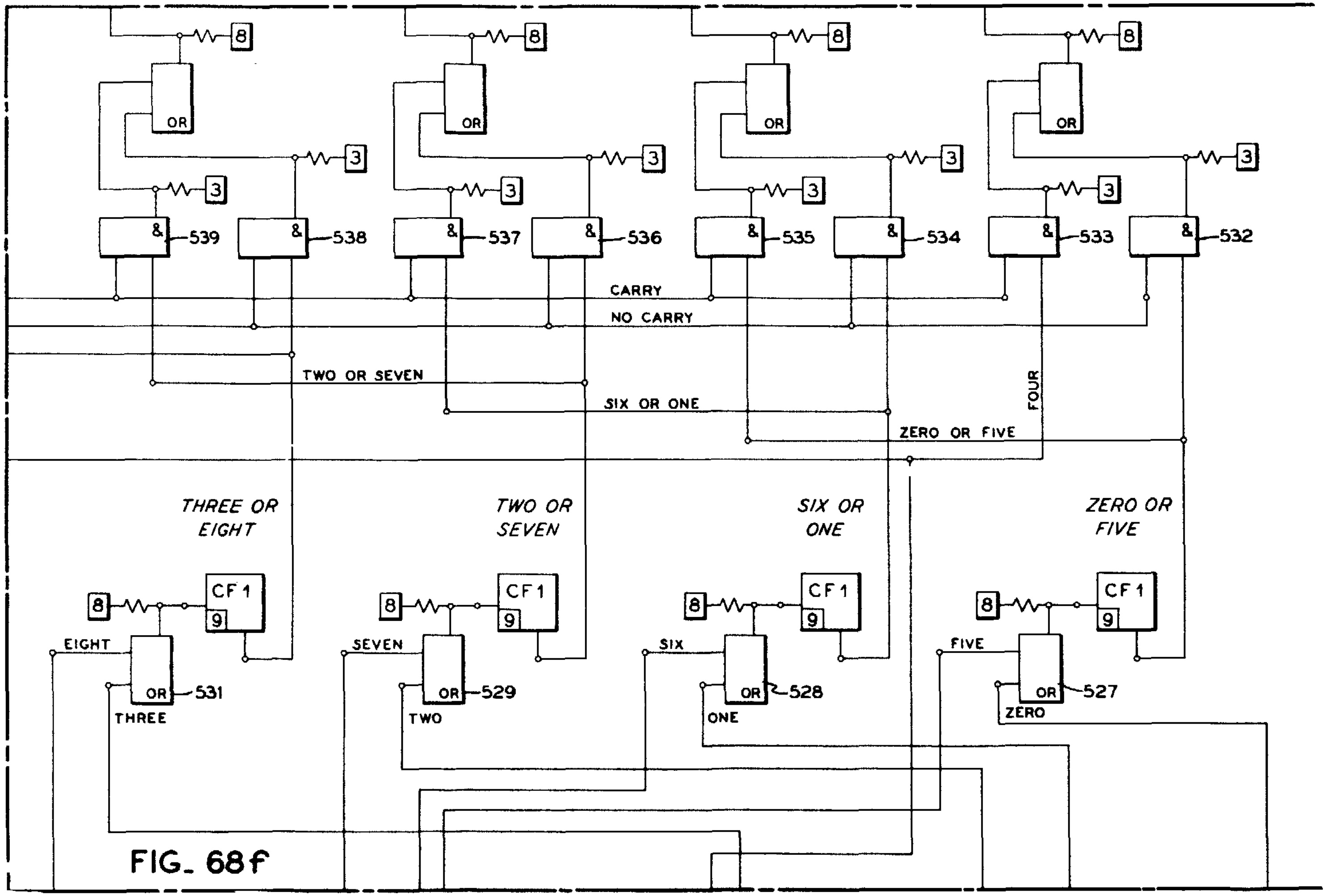


FIG. 68f

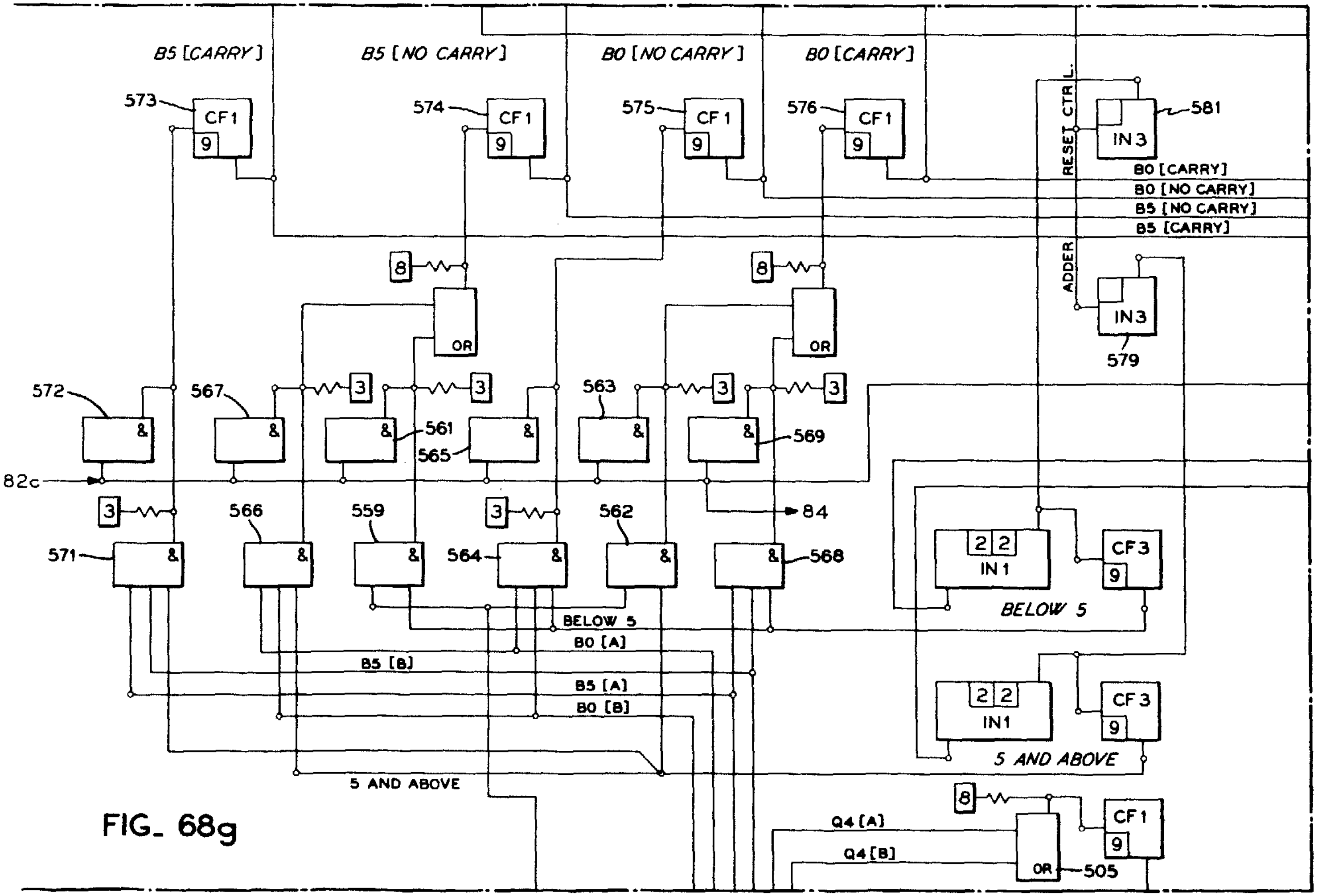

B5 [CARRY]
B5 [NO CARRY]
B0 [NO CARRY]
B0 [CARRY]
573
574
575
576
CF1
9
RESET CTRL.
581
IN3
B0 [CARRY]
B0 [NO CARRY]
B5 [NO CARRY]
B5 [CARRY]
ADDER
579
8
OR
3
572
567
561
565
563
569
82c
84
566
559
562
571
564
568
&
22
IN1
CF3
BELOW 5
BELOW 5
B0 [A]
B5 [B]
B5 [A]
B0 [B]
5 AND ABOVE
5 AND ABOVE
Q4 [A]
Q4 [B]
505

FIG_ 68g

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

FIG_ 68h

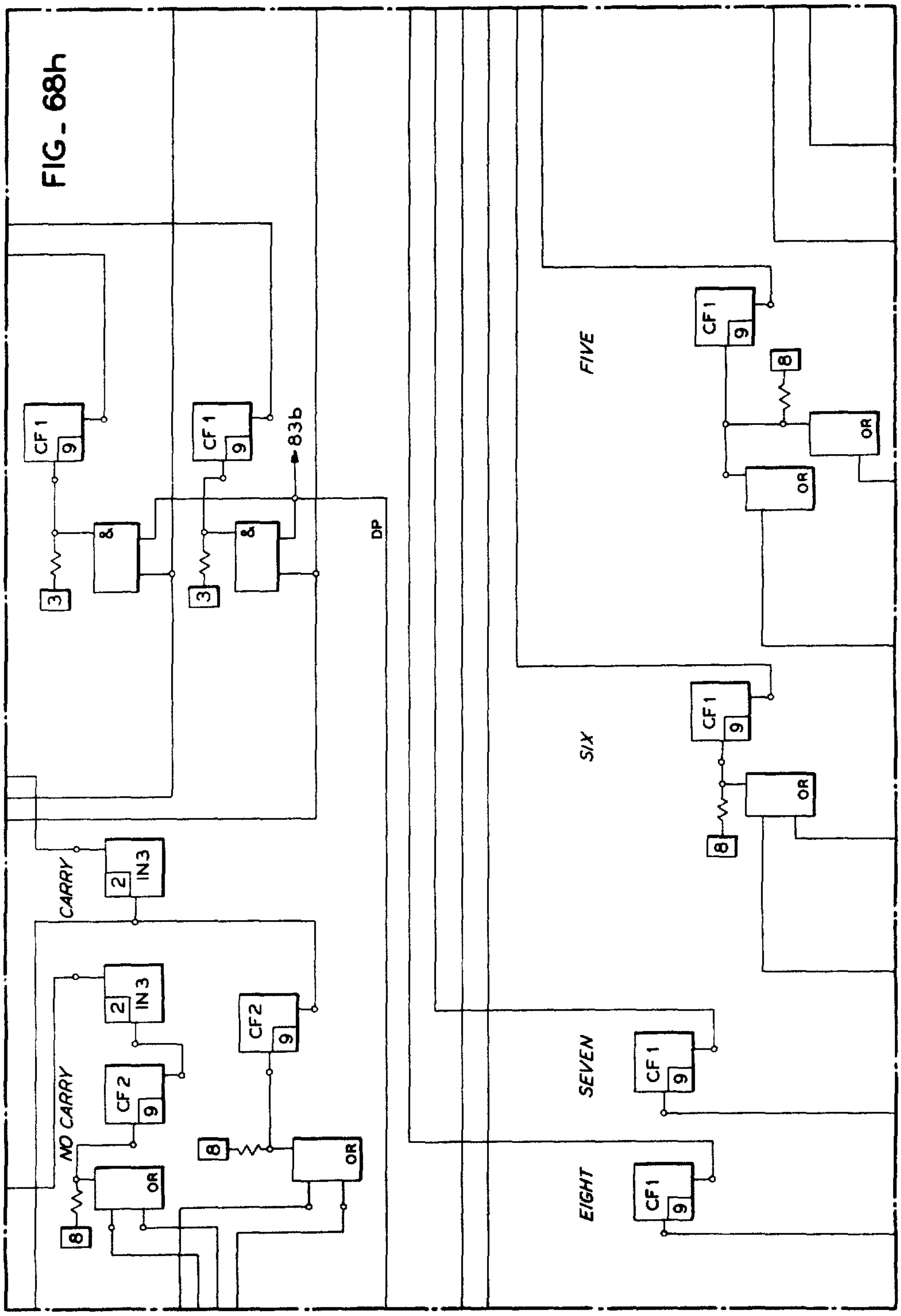
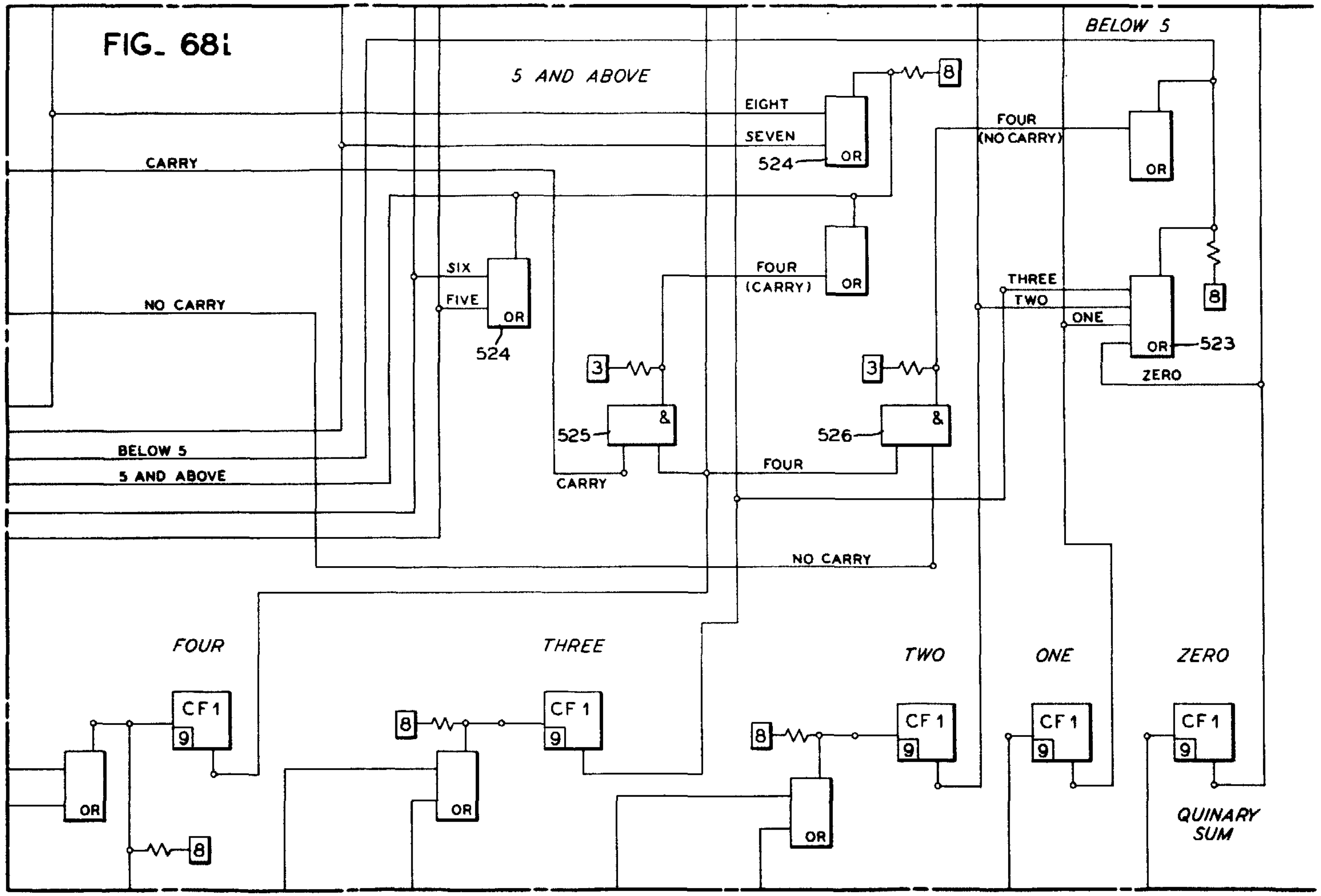


FIG_ 68i


5 AND ABOVE
BELOW 5
EIGHT
SEVEN
524
OR
8
FOUR
(NO CARRY)
OR
CARRY
SIX
FIVE
OR
524
FOUR
(CARRY)
OR
THREE
TWO
ONE
OR
523
8
ZERO
NO CARRY
3
&
525
3
&
526
BELOW 5
5 AND ABOVE
CARRY
FOUR
NO CARRY
FOUR
CF1
9
OR
8
THREE
8
CF1
9
OR
TWO
8
CF1
9
OR
ONE
CF1
9
ZERO
CF1
9
QUINARY SUM

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 115

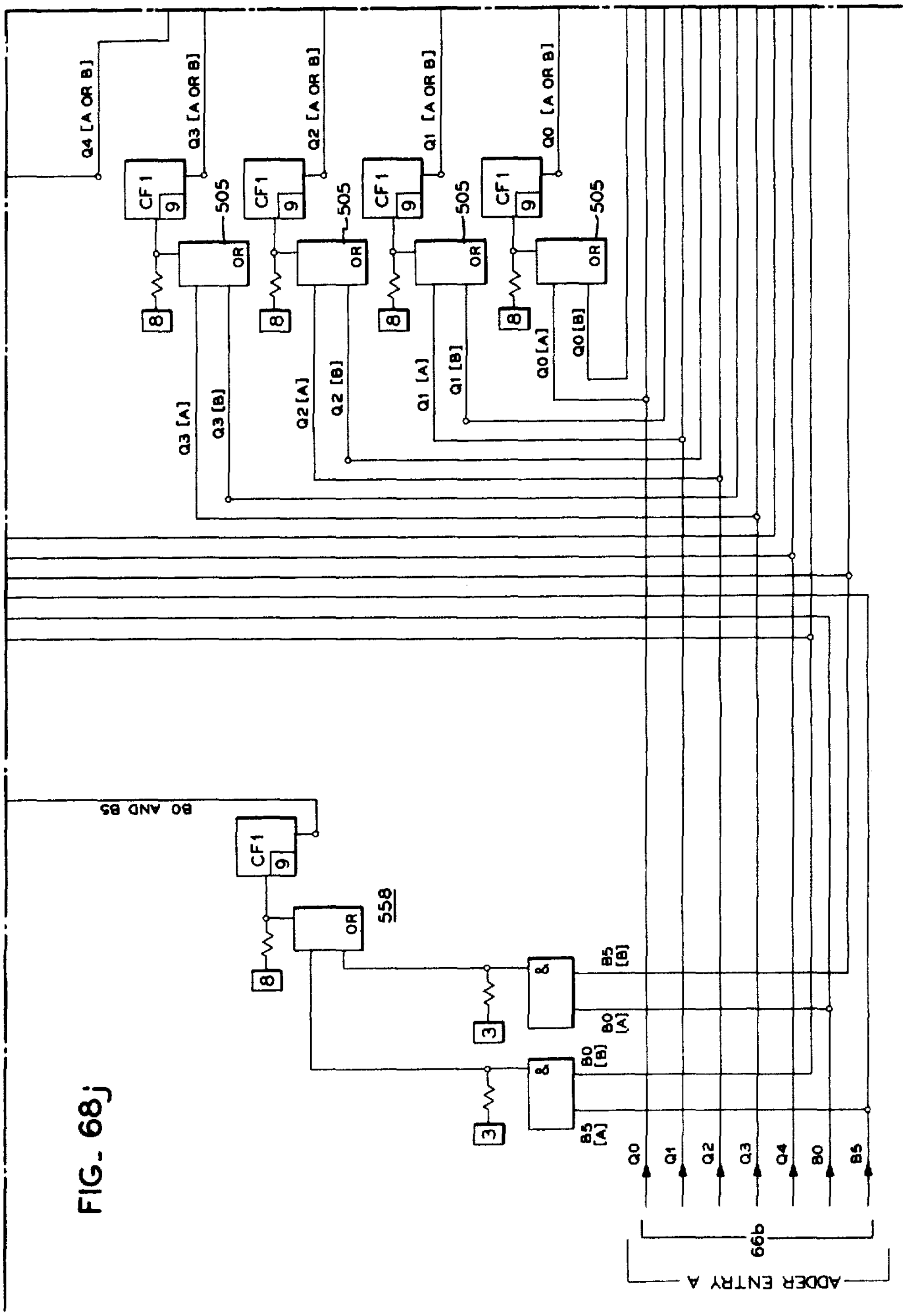


FIG. 68j

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

228 Sheets-Sheet 116

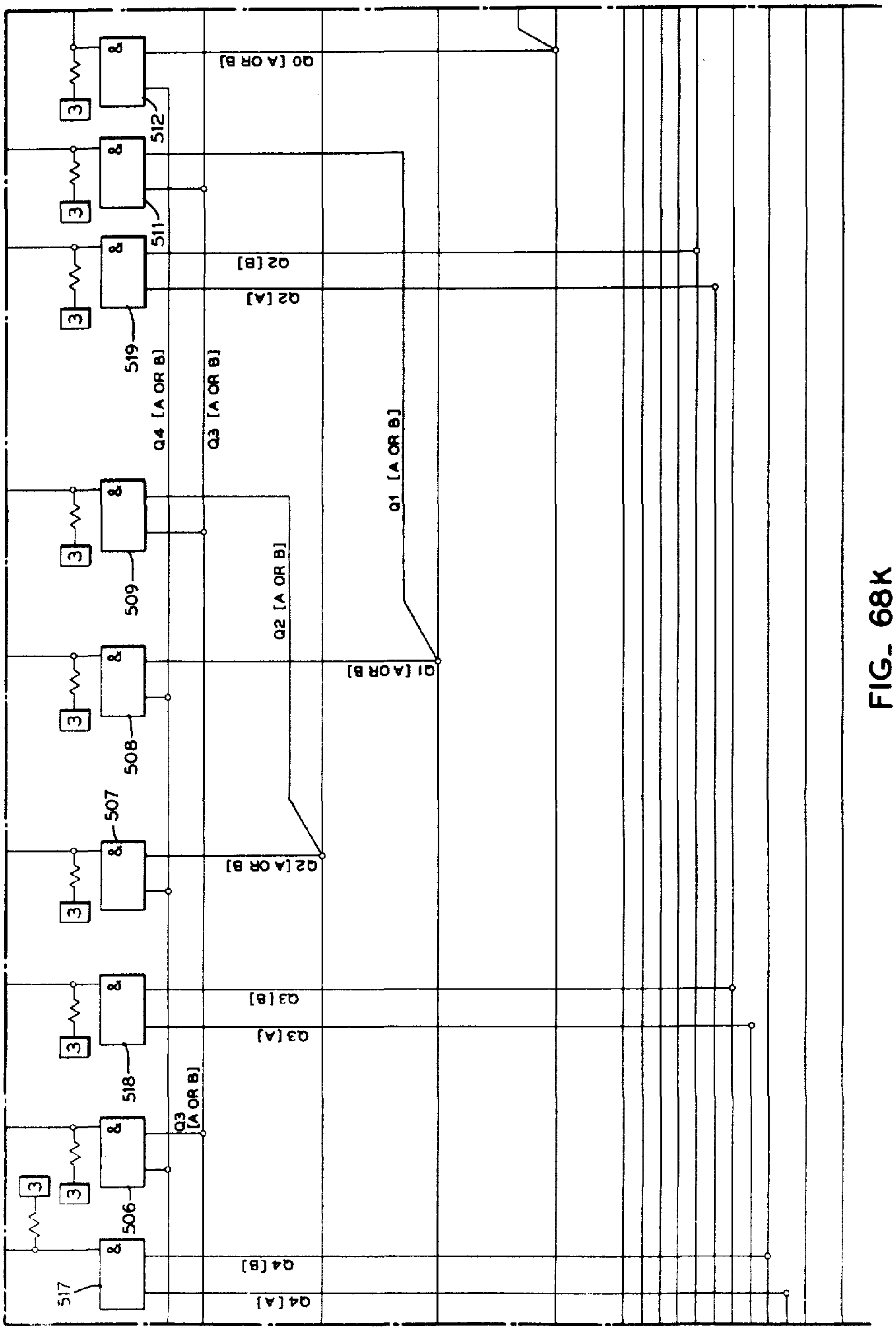


FIG_ 68K

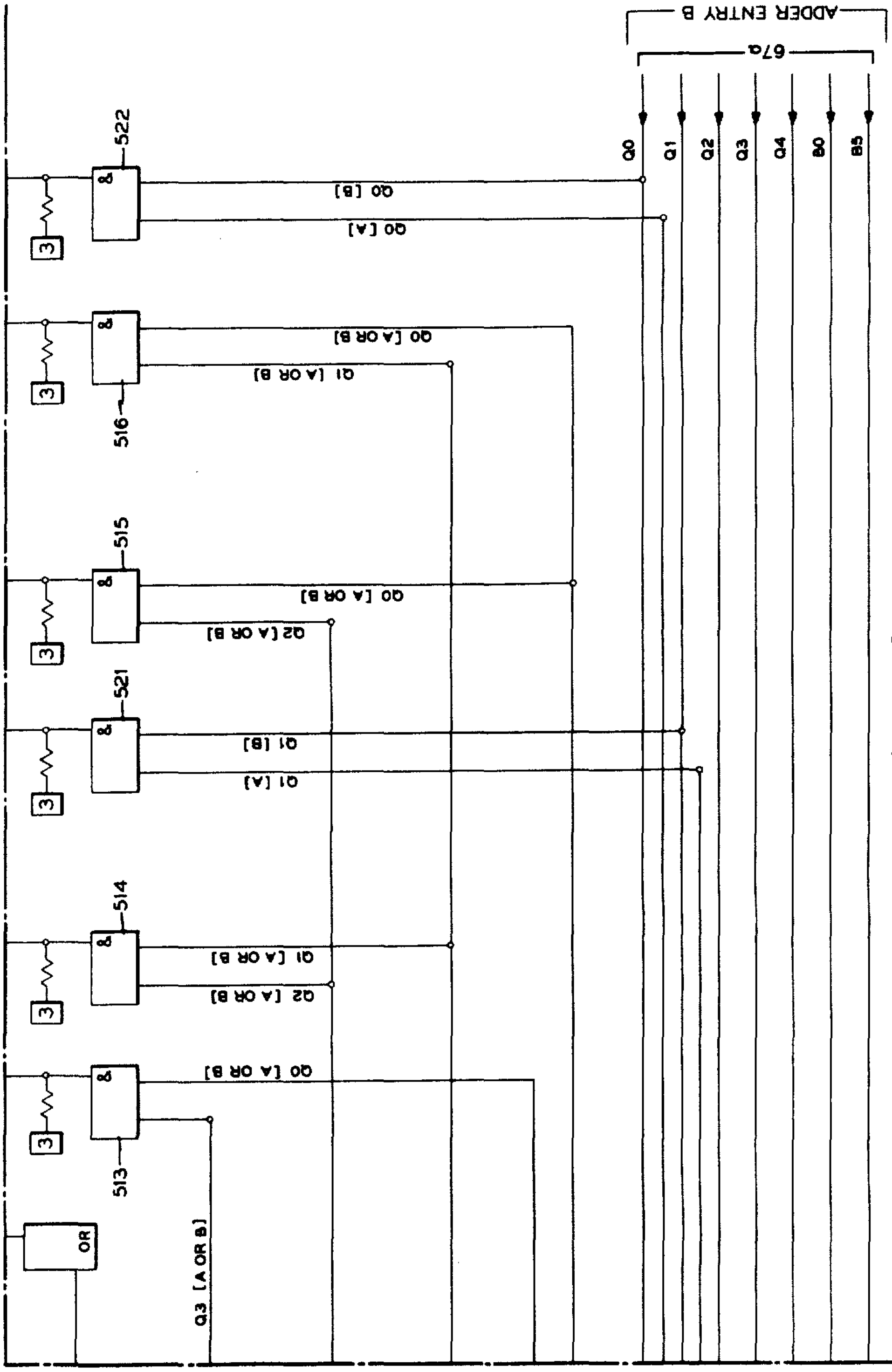

ADDER ENTRY B
67a
Q0
Q1
Q2
Q3
Q4
B0
B5
522
&
3
Q0 [B]
Q0 [A]
516
Q0 [A OR B]
Q1 [A OR B]
515
Q2 [A OR B]
521
Q1 [B]
Q1 [A]
514
513
Q3 [A OR B]
OR

FIG. 681

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 118

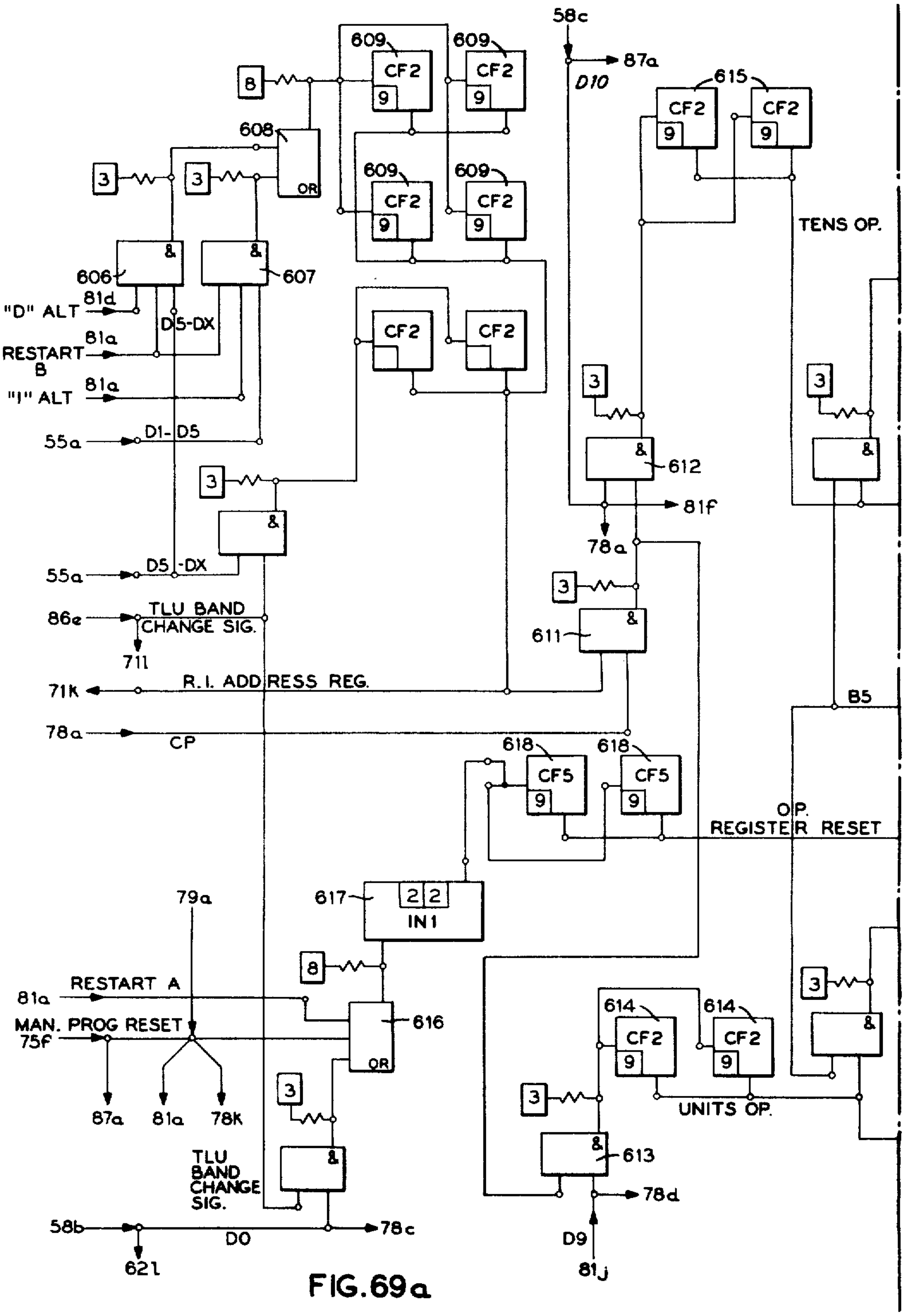


FIG.69a

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 119

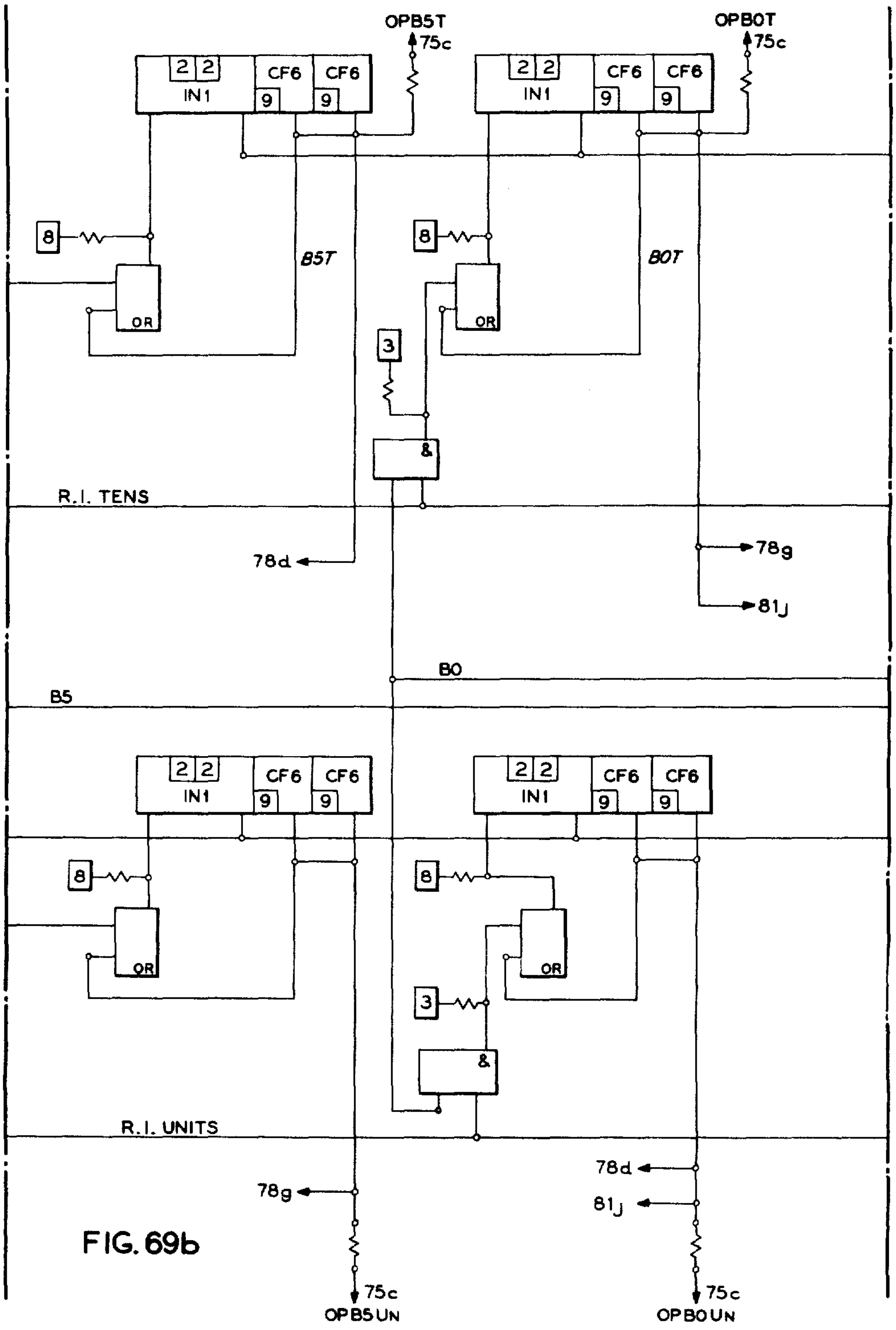


FIG. 69b

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 120

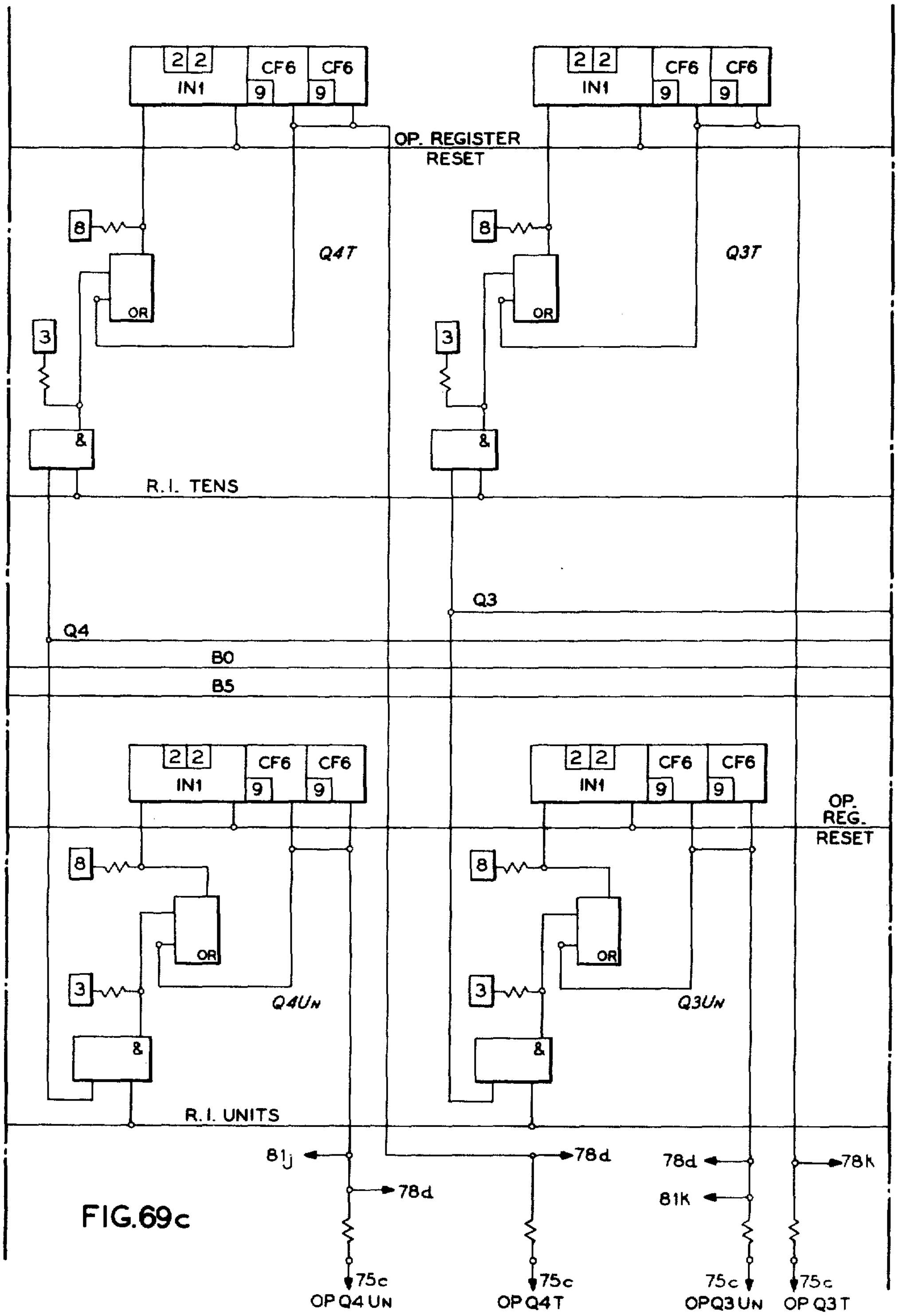


FIG.69c

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 121

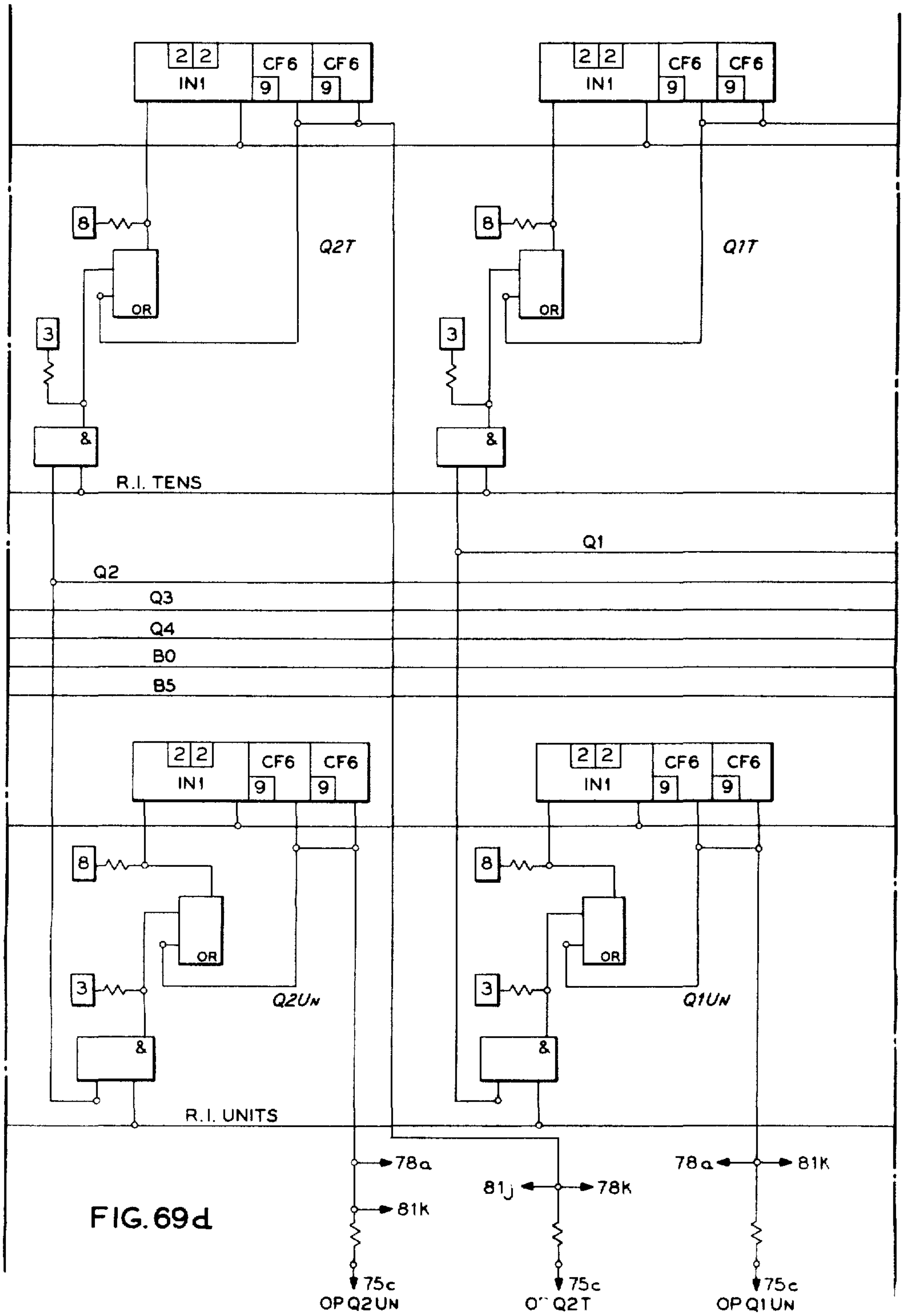


FIG. 69d

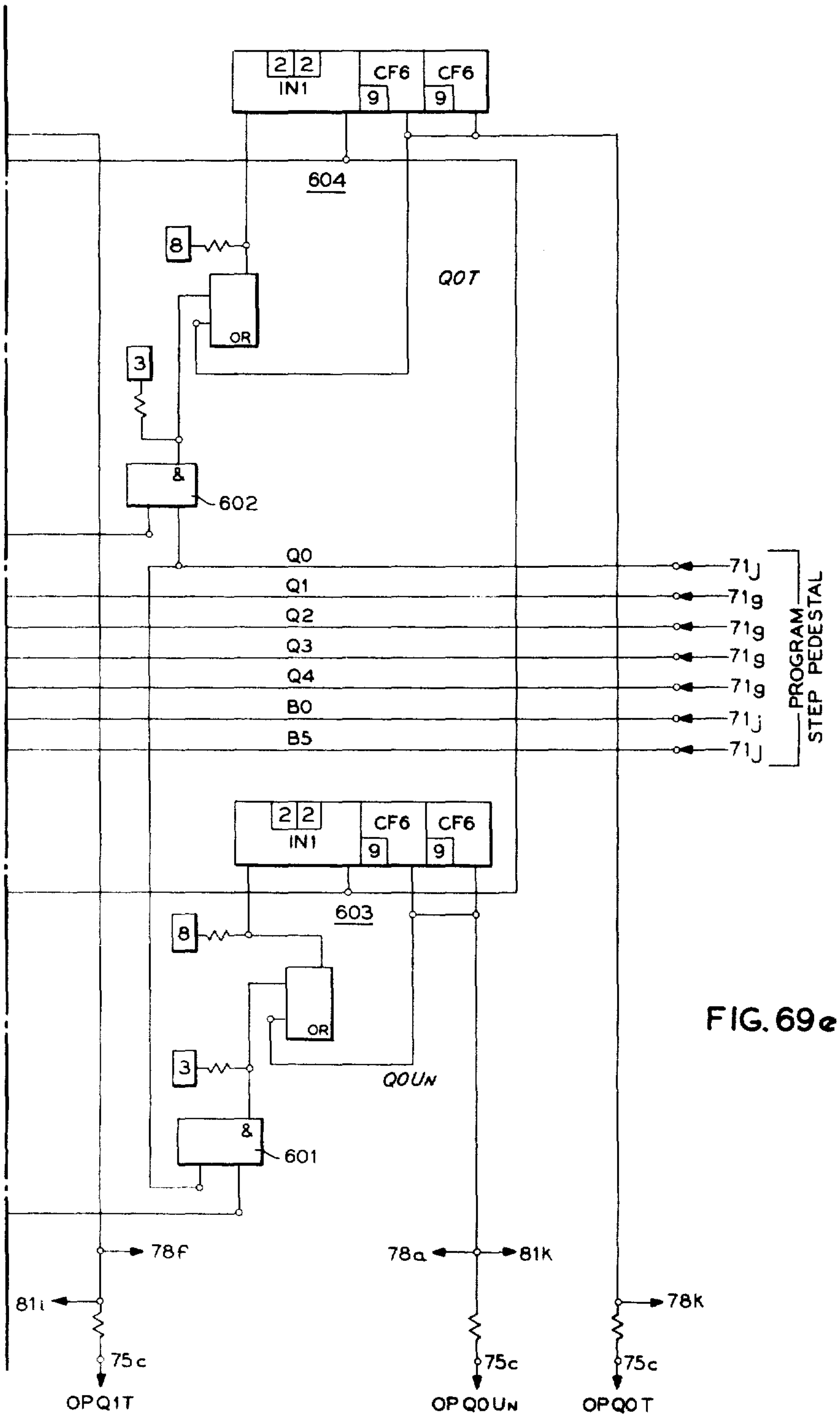

2 2
IN1
CF6
9
CF6
9
604
8
QOT
OR
3
&
602
Q0
Q1
Q2
Q3
Q4
B0
B5
71J
71g
71g
71g
71g
71j
71J
PROGRAM STEP PEDESTAL
2 2
IN1
CF6
9
CF6
9
603
8
OR
QOUN
3
&
601
78F
81i
75c
OPQ1T
78a
81K
75c
OPQOUN
78K
75c
OPQOT

FIG. 69e

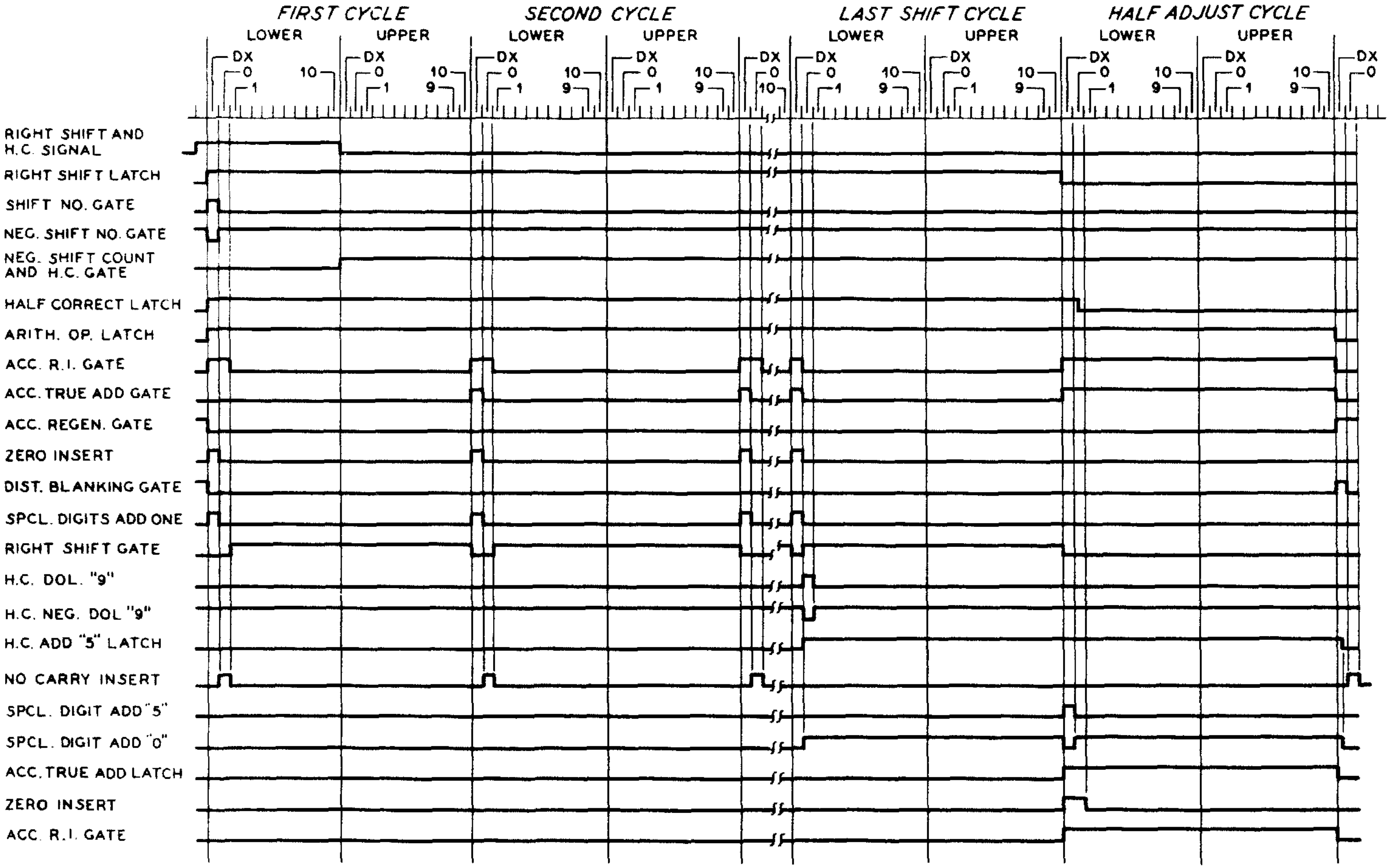

FIRST CYCLE
SECOND CYCLE
LAST SHIFT CYCLE
HALF ADJUST CYCLE
LOWER
UPPER
DX
0
1
10
9
RIGHT SHIFT AND H.C. SIGNAL
RIGHT SHIFT LATCH
SHIFT NO. GATE
NEG. SHIFT NO. GATE
NEG. SHIFT COUNT AND H.C. GATE
HALF CORRECT LATCH
ARITH. OP. LATCH
ACC. R.I. GATE
ACC. TRUE ADD GATE
ACC. REGEN. GATE
ZERO INSERT
DIST. BLANKING GATE
SPCL. DIGITS ADD ONE
RIGHT SHIFT GATE
H.C. DOL. "9"
H.C. NEG. DOL "9"
H.C. ADD "5" LATCH
NO CARRY INSERT
SPCL. DIGIT ADD "5"
SPCL. DIGIT ADD "0"
ACC. TRUE ADD LATCH
ZERO INSERT
ACC. R.I. GATE

FIG. 70

DATA STORAGE AND PROCESSING MACHINE

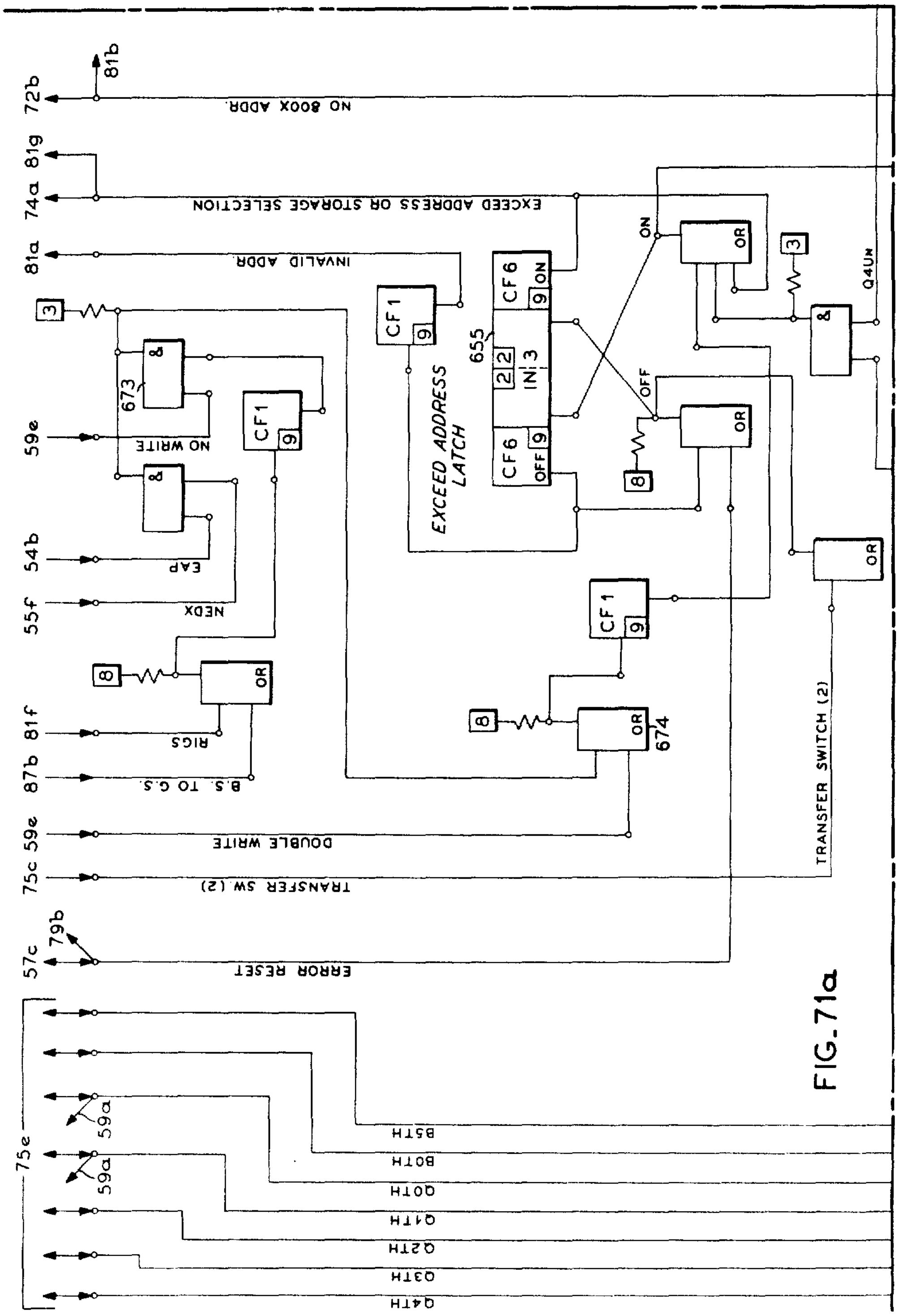


FIG. 71a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 125

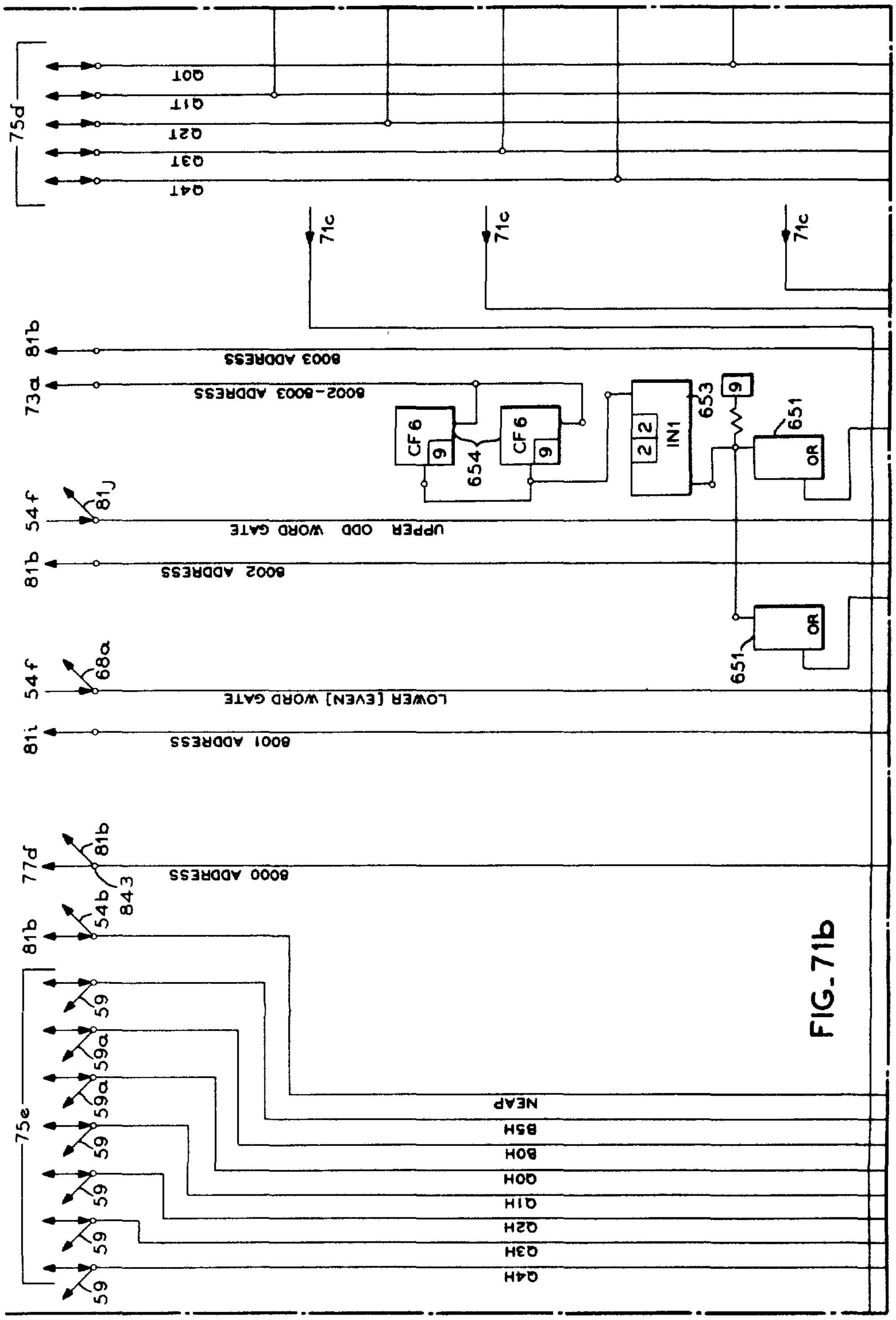


FIG. 71b

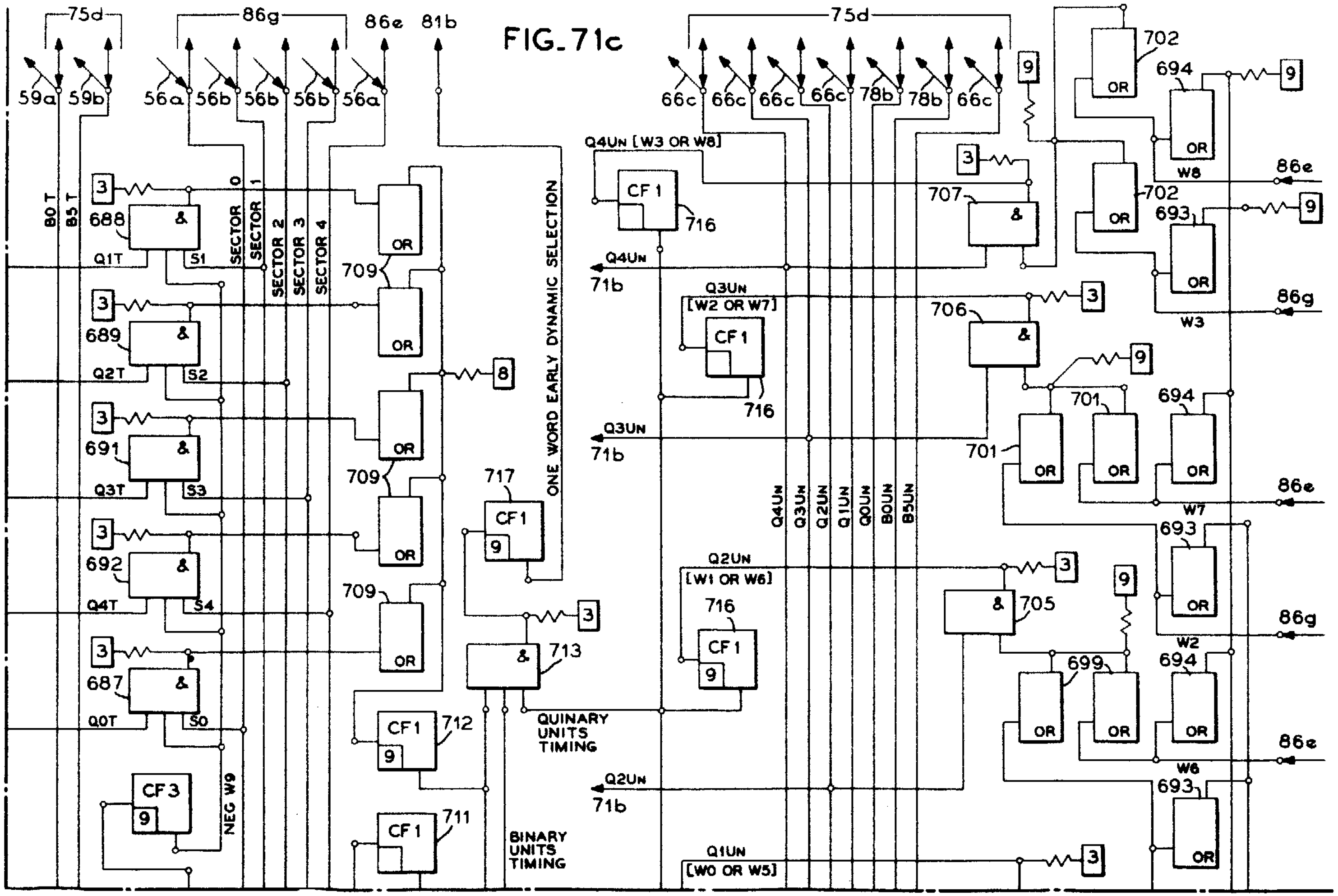
Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

Filed Nov. 2, 1955

DATA STORAGE AND PROCESSING MACHINE

228 Sheets-Sheet 126

FIG_71c



Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

Filed Nov. 2, 1955

DATA STORAGE AND PROCESSING MACHINE

228 Sheets-Sheet 127

FIG_71d

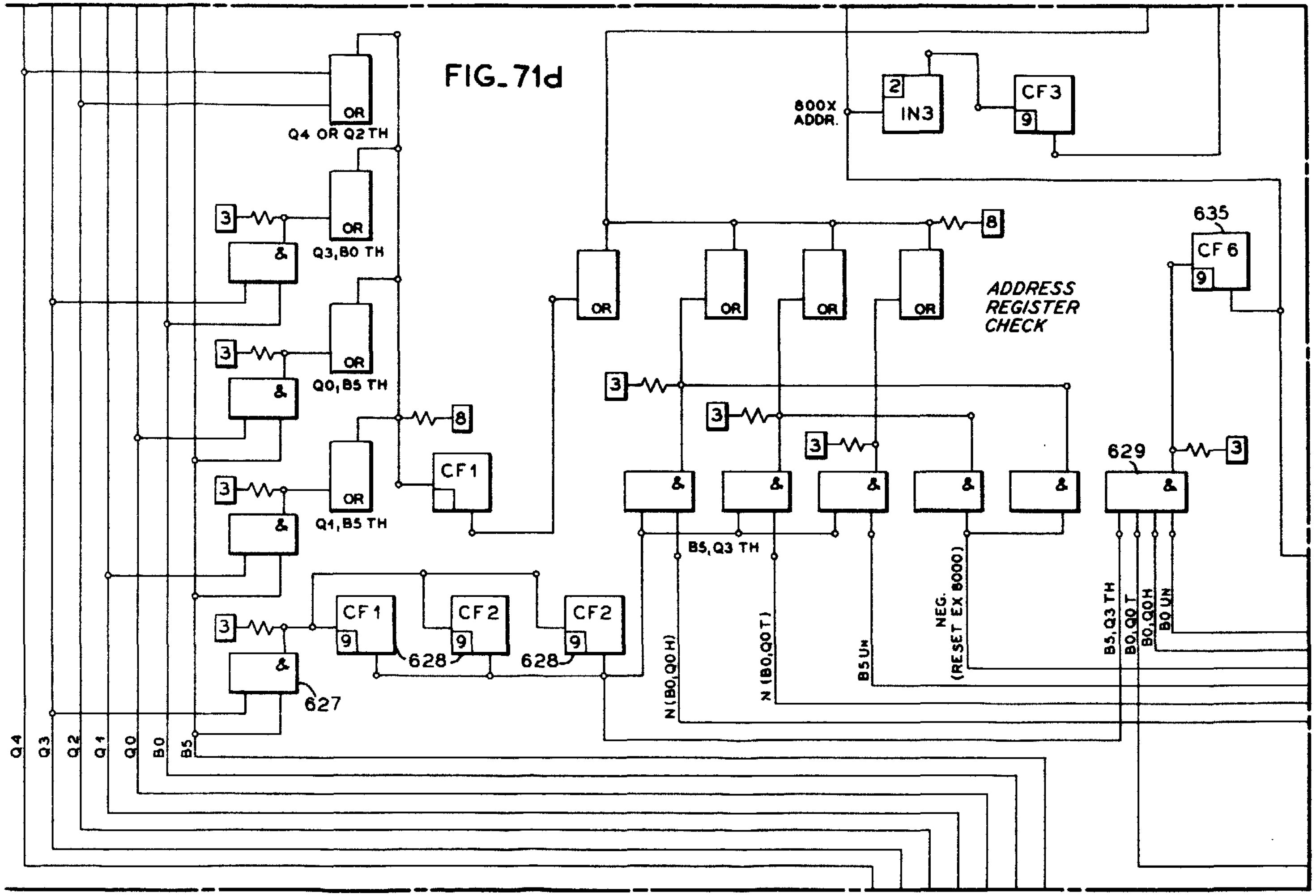
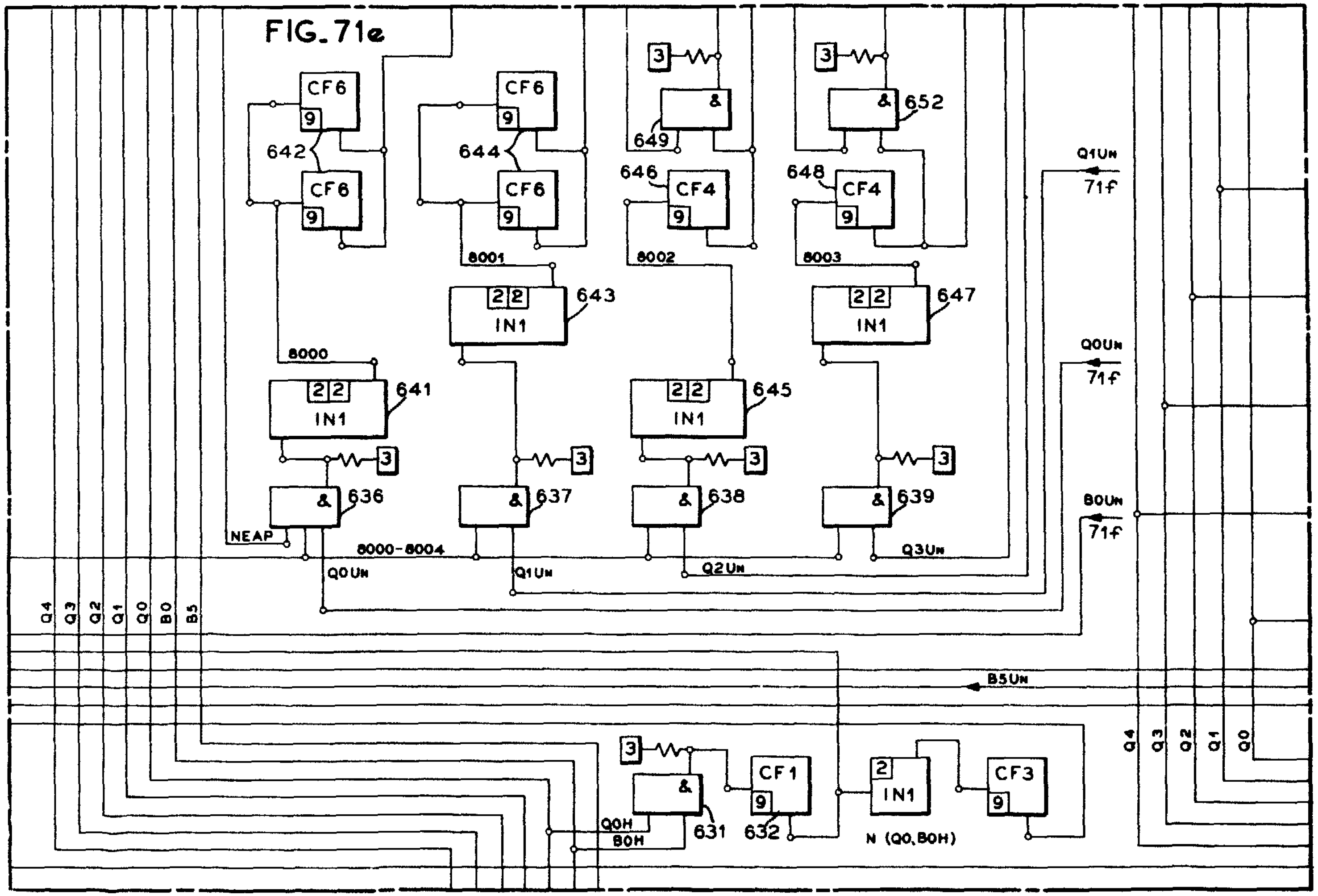


FIG. 71e


CF6
9
642
CF6
9
CF6
9
644
CF6
9
3
&
649
646
CF4
9
3
&
652
648
CF4
9
Q1UN
71f
8001
8002
8003
2 2
IN1
643
2 2
IN1
647
8000
2 2
IN1
641
2 2
IN1
645
Q0UN
71f
3
3
3
3
&
636
&
637
&
638
&
639
B0UN
71f
NEAP
8000-8004
Q0UN
Q1UN
Q2UN
Q3UN
Q4
Q3
Q2
Q1
Q0
B0
B5
B5UN
Q4
Q3
Q2
Q1
Q0
3
&
CF1
9
2
IN1
CF3
9
Q0H
B0H
631
632
N (Q0,B0H)

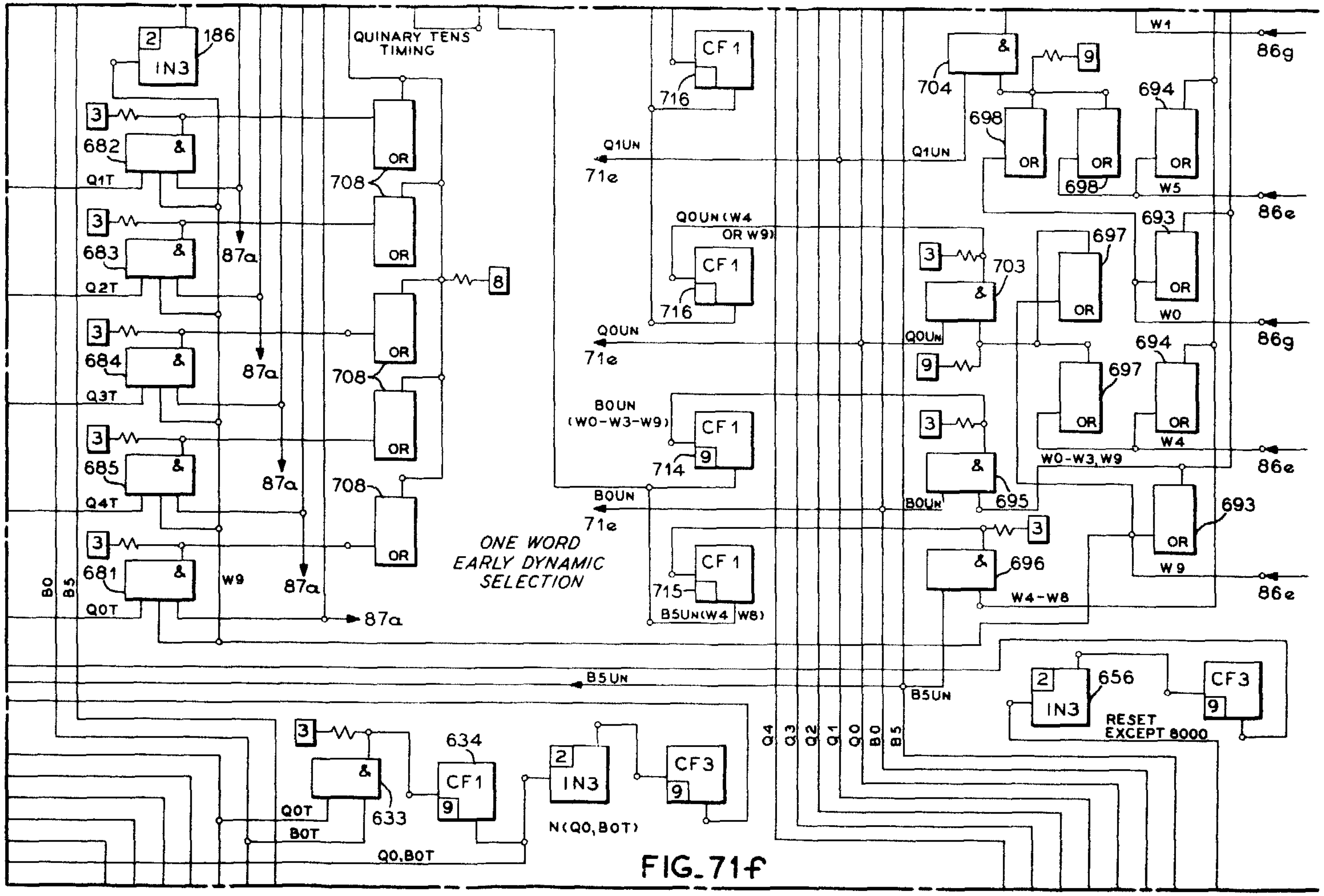
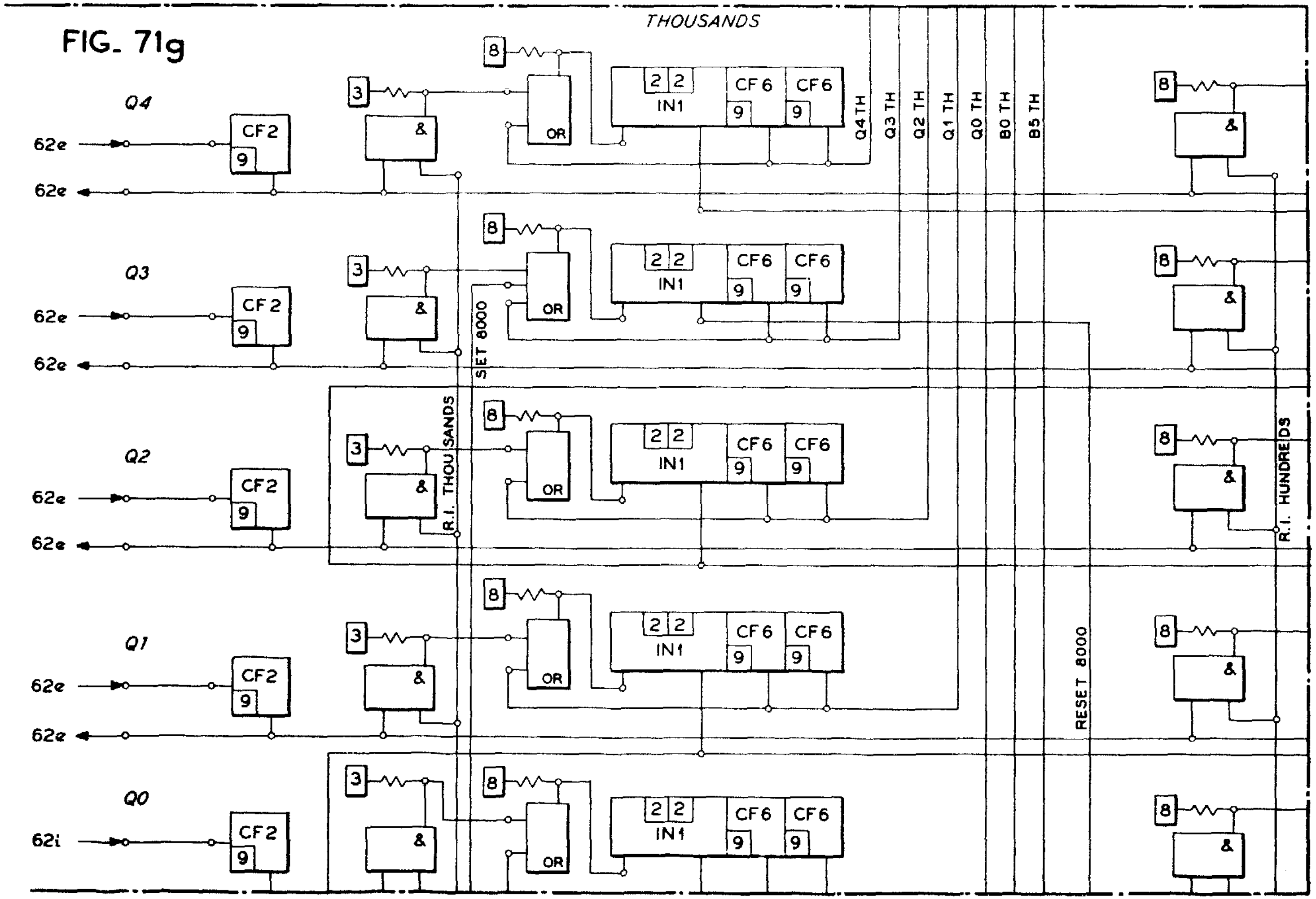

QUINARY TENS TIMING
186
IN3
2
3
682
683
684
685
681
Q1T
Q2T
Q3T
Q4T
Q0T
B0
B5
W9
87a
708
OR
8
ONE WORD EARLY DYNAMIC SELECTION
CF1
716
714
715
9
Q1UN
Q0UN
Q0UN(W4 OR W9)
B0UN
B0UN (W0-W3-W9)
B5UN(W4 W8)
71e
B5UN
704
698
694
693
697
703
695
696
W1
W5
W0
W4
W0-W3,W9
W4-W8
86g
86e
656
CF3
RESET EXCEPT 8000
Q4
Q3
Q2
Q1
Q0
634
633
B0T
Q0,B0T
N(Q0,B0T)
&

FIG_71f

FIG. 71g

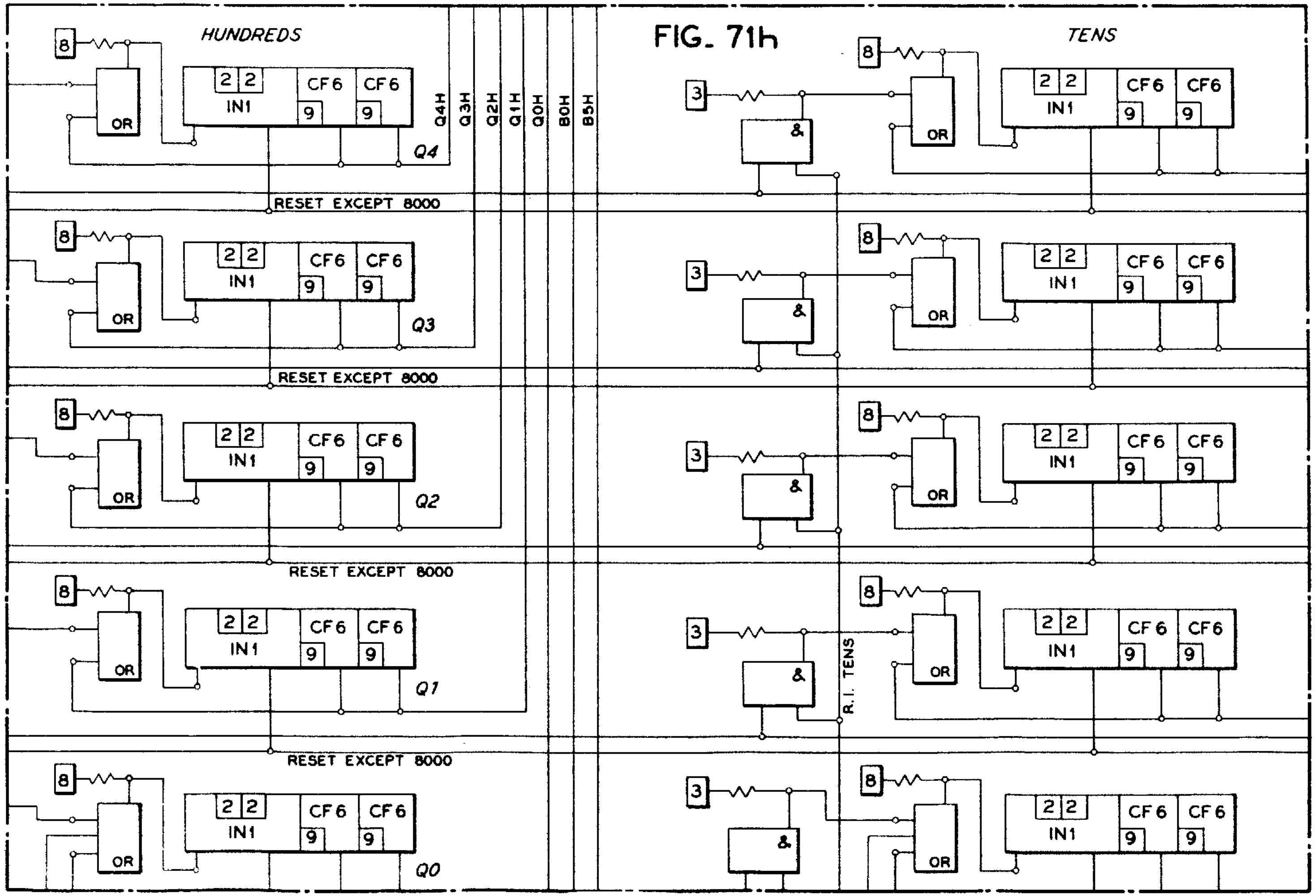

THOUSANDS
Q4
Q3
Q2
Q1
Q0
62e
62i
CF2
9
3
8
&
OR
2 2
IN1
CF6
Q4 TH
Q3 TH
Q2 TH
Q1 TH
Q0 TH
B0 TH
B5 TH
SET 8000
R.I. THOUSANDS
RESET 8000
R.I. HUNDREDS

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 131

FIG. 71h



DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

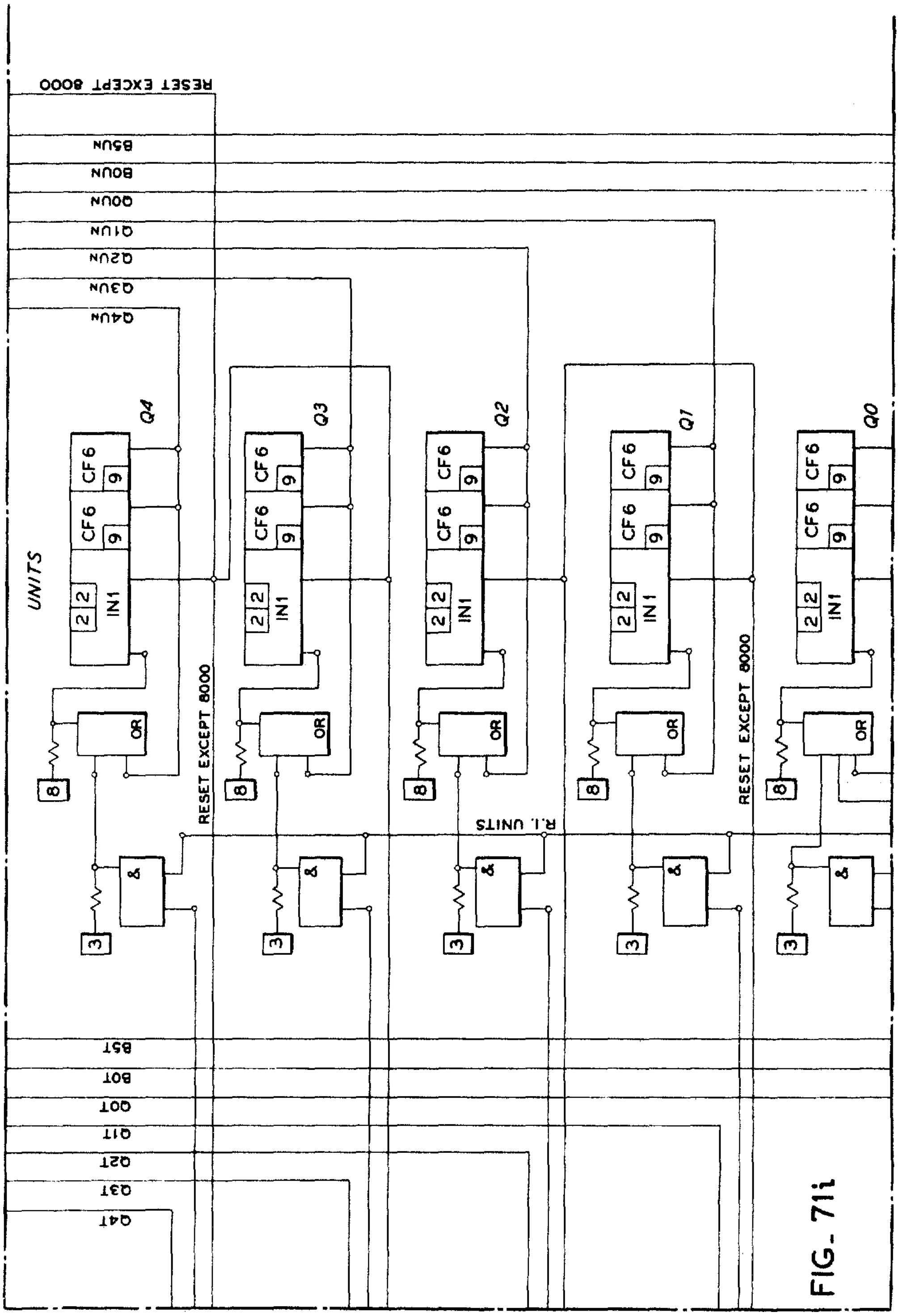


FIG. 71i

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 133

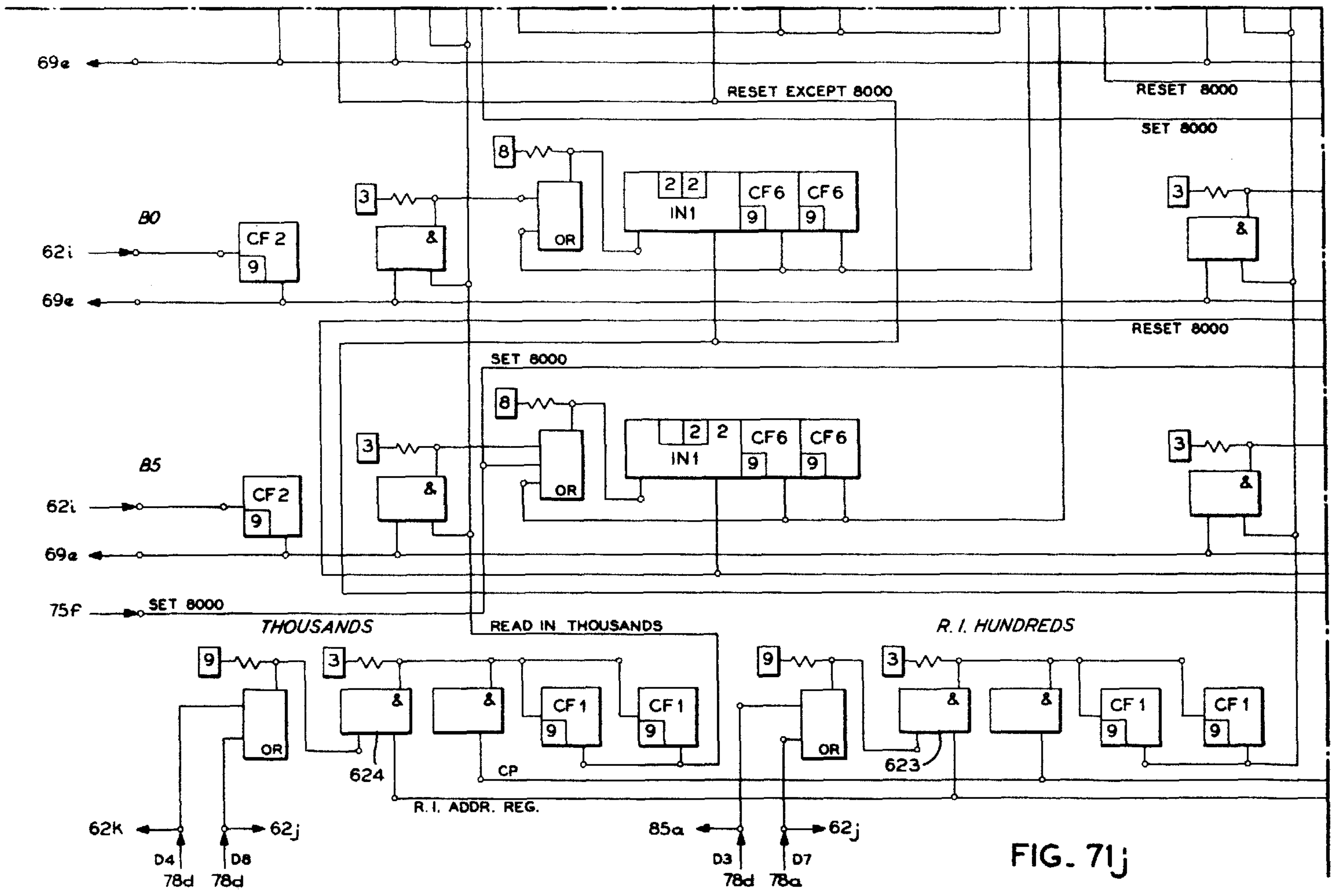


FIG. 71j

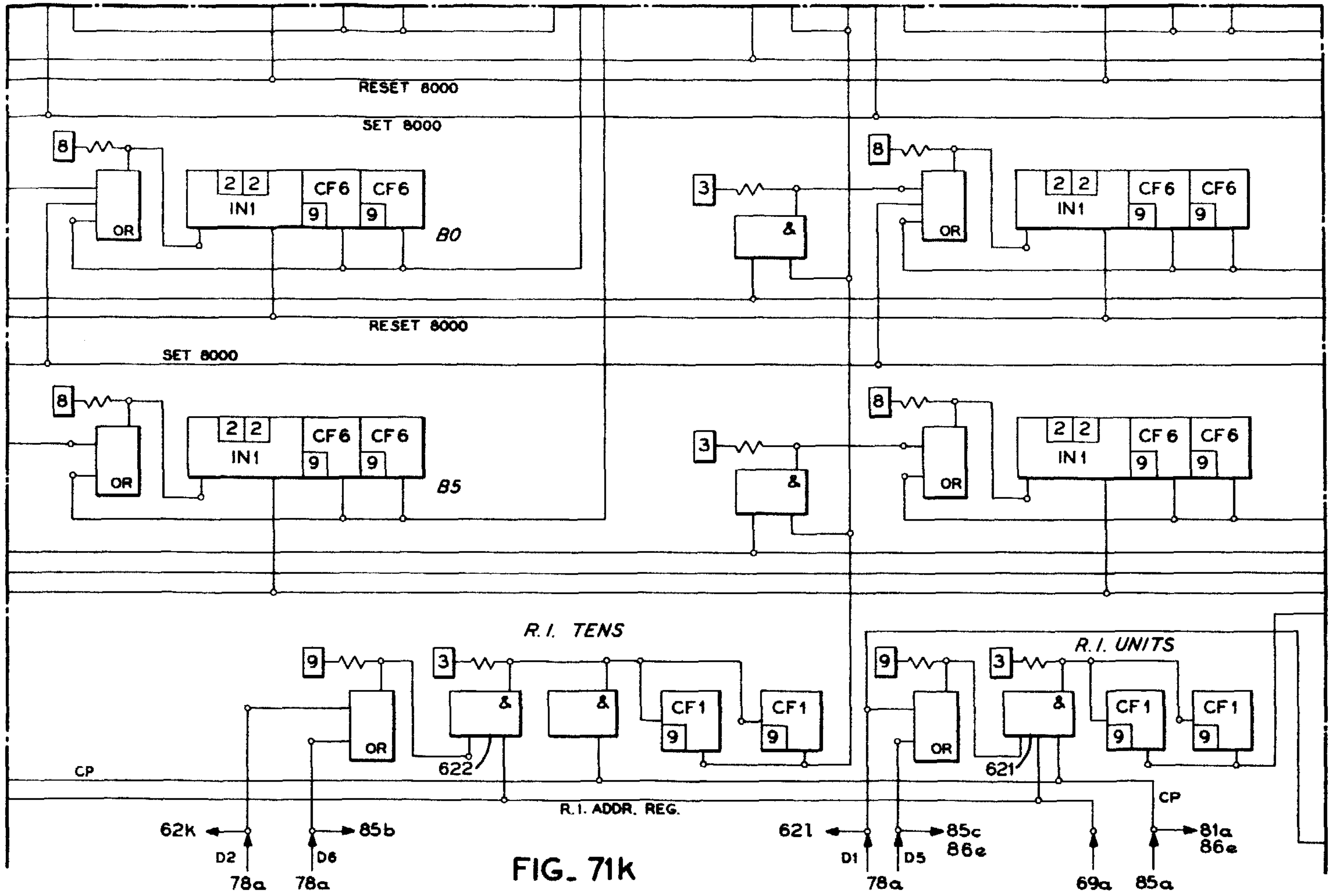

RESET 8000
SET 8000
8
OR
2 2
IN1
CF6
9
B0
3
&
RESET 8000
SET 8000
B5
R.I. TENS
R.I. UNITS
CF1
622
621
CP
R.I. ADDR. REG.
62K
D2
78a
D6
85b
621
D1
D5
85c
86e
69a
85a
81a
86e

FIG. 71k

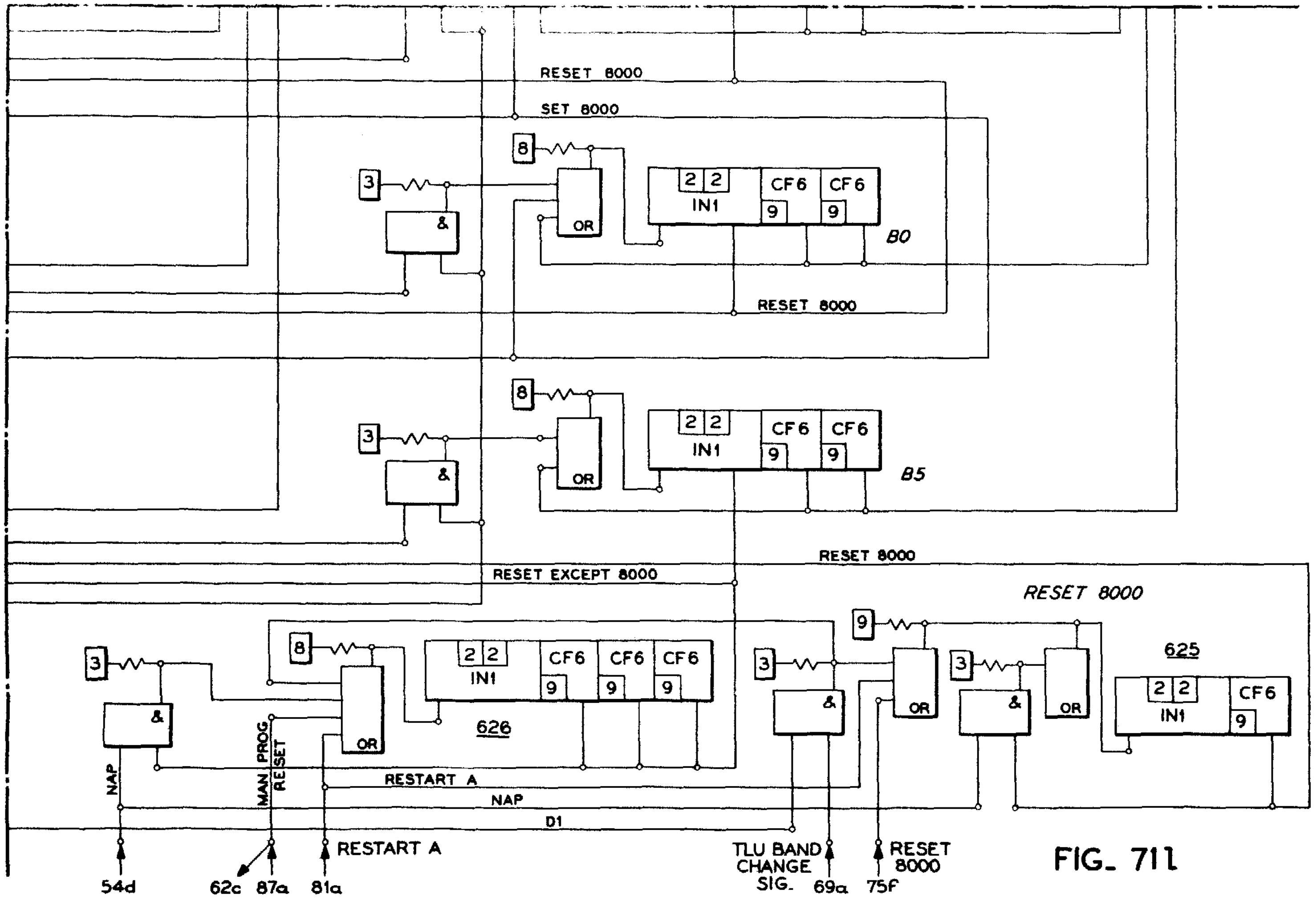

RESET 8000
SET 8000
8
3
&
OR
2 2
IN1
CF6
CF6
9
9
80
RESET 8000
8
3
&
OR
2 2
IN1
CF6
CF6
9
9
85
RESET 8000
RESET EXCEPT 8000
RESET 8000
3
&
8
OR
2 2
IN1
CF6
CF6
CF6
9
9
9
626
9
3
&
OR
3
&
OR
625
2 2
IN1
CF6
9
NAP
MAN PROG
RESET
RESTART A
NAP
D1
54d
62c
87a
81a
RESTART A
TLU BAND
CHANGE
SIG.
69a
RESET
8000
75f

FIG. 711

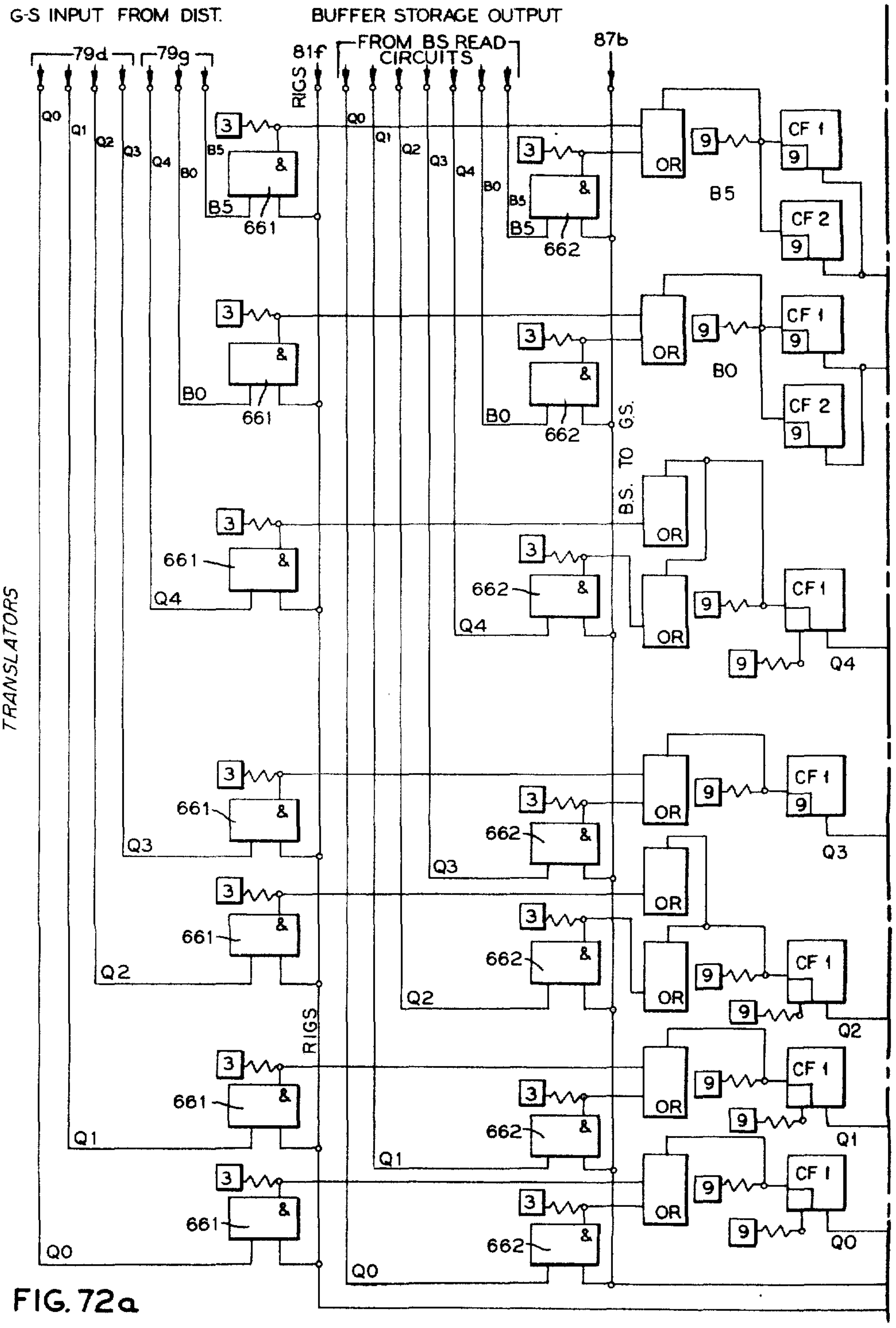

G-S INPUT FROM DIST.
BUFFER STORAGE OUTPUT
79d
79g
81f
FROM BS READ CIRCUITS
87b
RIGS
Q0
Q1
Q2
Q3
Q4
B0
B5
3
&
661
662
OR
9
CF 1
CF 2
B.S. TO G.S.
TRANSLATORS

FIG. 72a

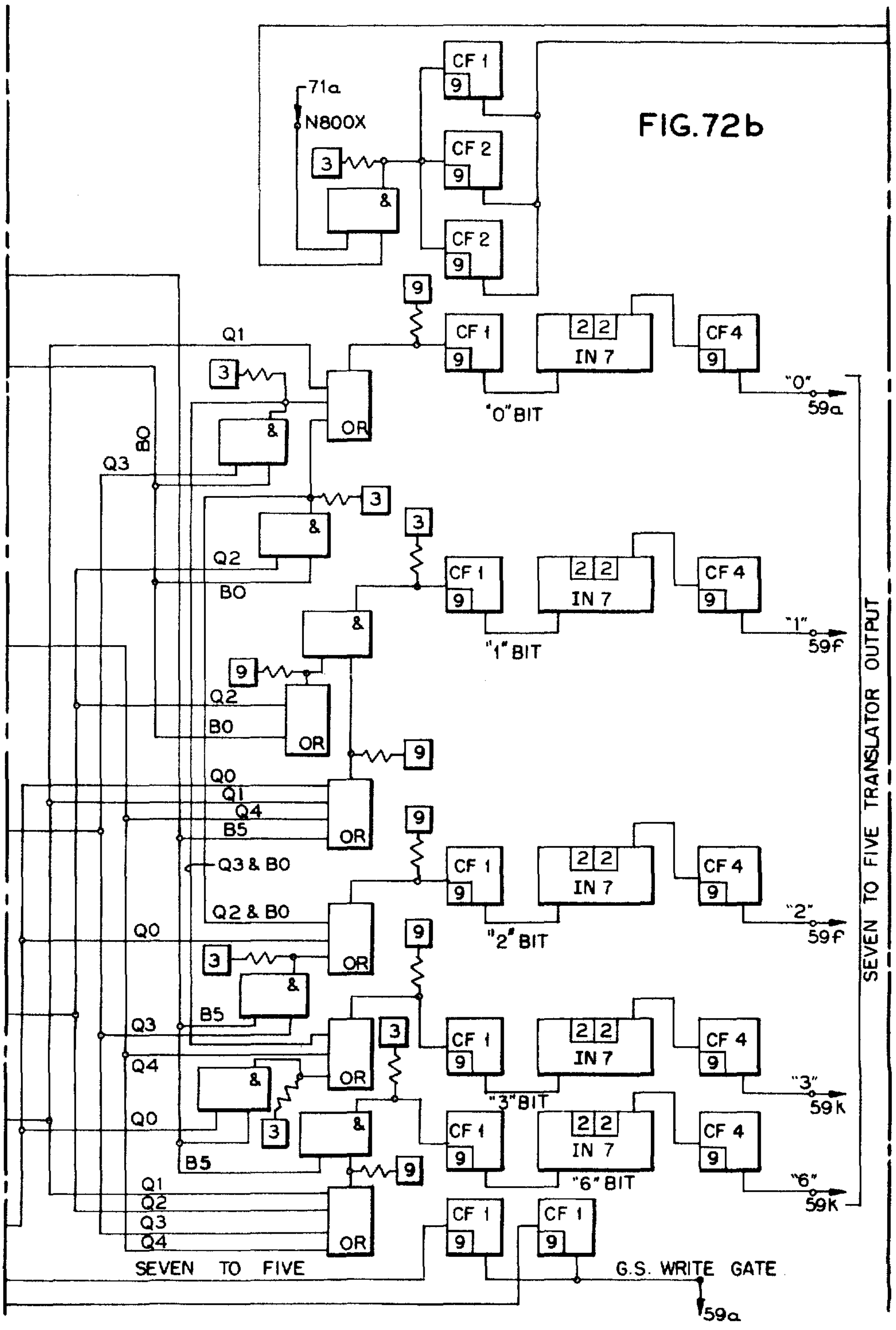


FIG.72b

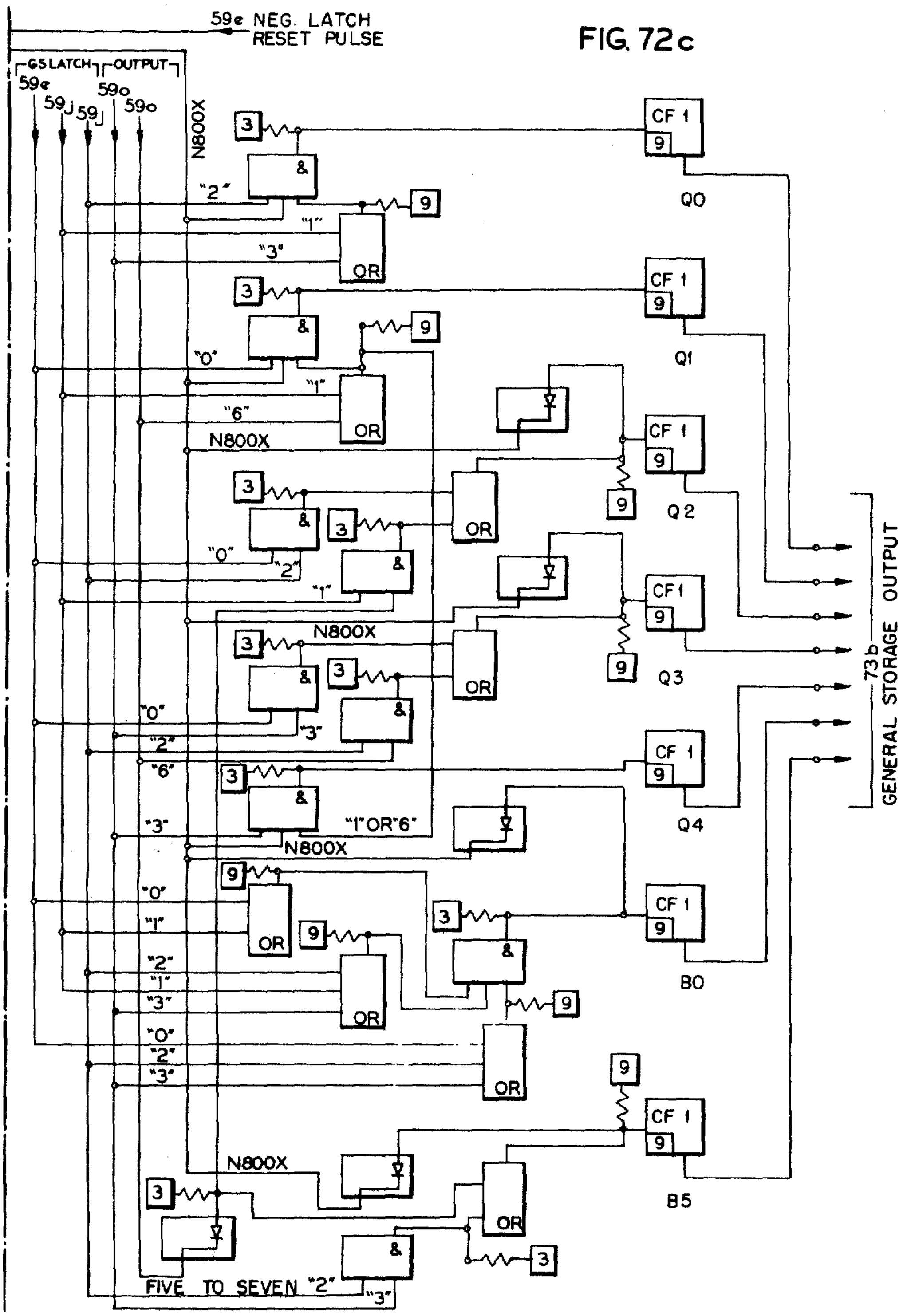
71a
N800X
3
&
CF 1
9
CF 2
9
CF 2
9
9
CF 1
9
22
IN 7
CF4
9
"0"
59a
"0" BIT
Q1
3
&
OR
BO
Q3
3
&
3
Q2
BO
CF 1
9
22
IN 7
CF 4
9
"1"
59f
&
"1" BIT
9
Q2
BO
OR
9
Q0
Q1
Q4
B5
OR
Q3 & BO
9
CF 1
9
22
IN 7
CF 4
9
"2"
59f
Q2 & BO
OR
"2" BIT
Q0
3
&
9
B5
Q3
OR
3
CF 1
9
22
IN 7
CF 4
9
"3"
59k
Q4
&
OR
"3" BIT
Q0
3
&
CF 1
9
22
IN 7
CF 4
9
"6"
59K
"6" BIT
B5
9
Q1
Q2
Q3
Q4
OR
CF 1
9
CF 1
9
SEVEN TO FIVE
G.S. WRITE GATE
59a
SEVEN TO FIVE TRANSLATOR OUTPUT

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 138

FIG. 72c



DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

228 Sheets-Sheet 139

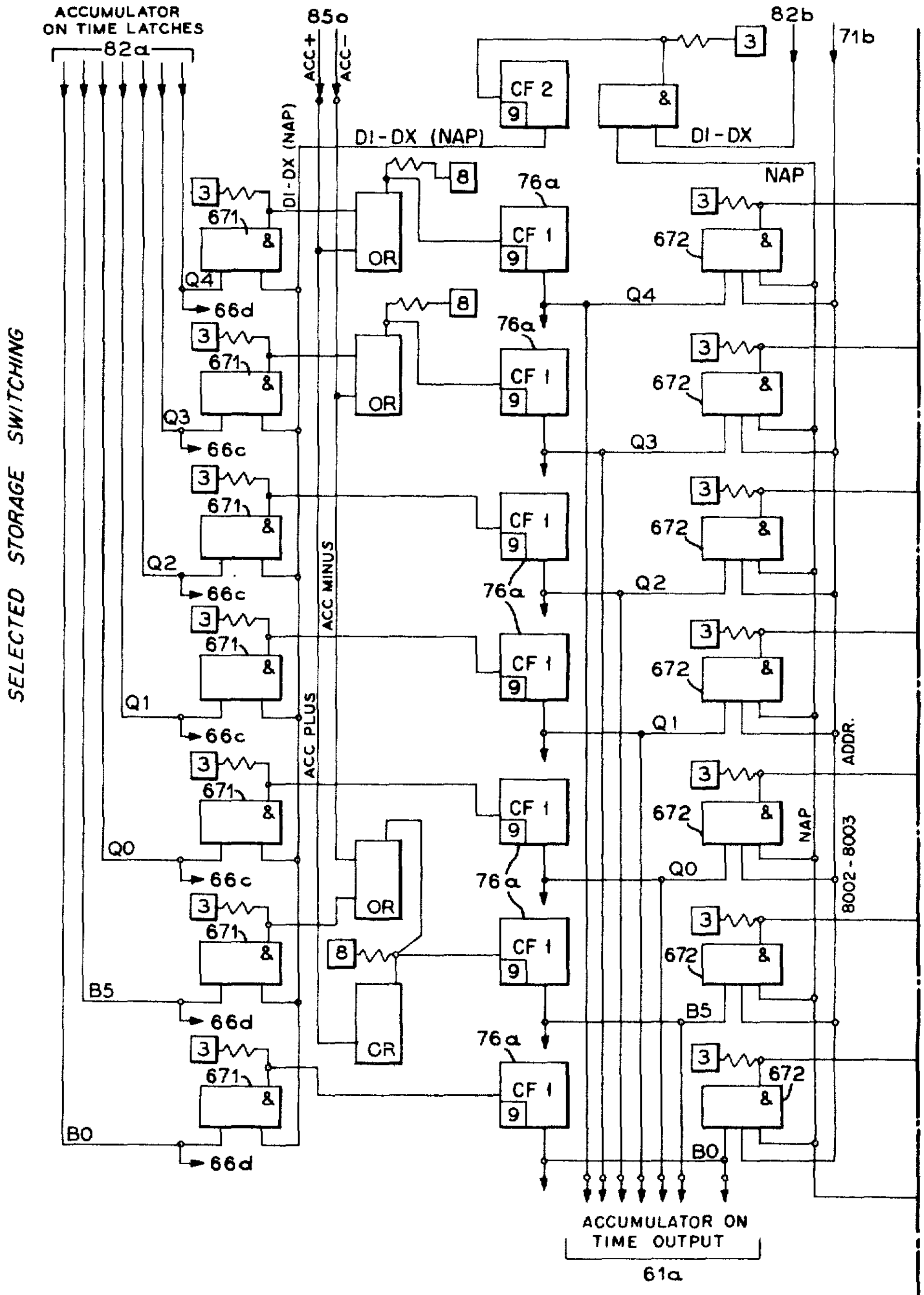


FIG. 73a

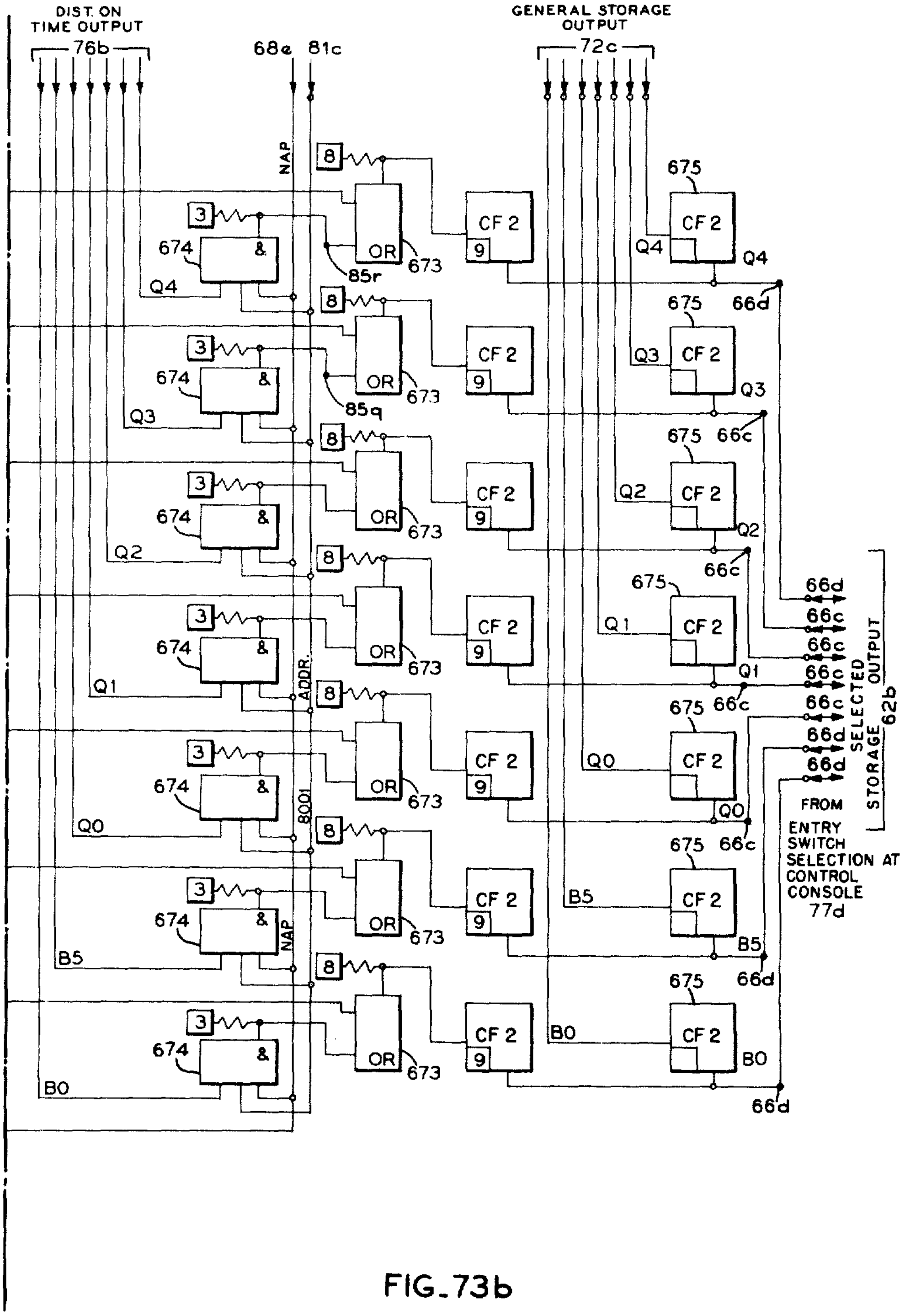

DIST. ON TIME OUTPUT
76b
68e
81c
GENERAL STORAGE OUTPUT
72c
NAP
8
3
674
&
OR
673
85r
85q
CF 2
9
675
Q4
Q3
Q2
Q1
Q0
B5
B0
66d
66c
ADDR.
8001
SELECTED STORAGE OUTPUT 62b
FROM ENTRY SWITCH SELECTION AT CONTROL CONSOLE 77d

FIG. 73b

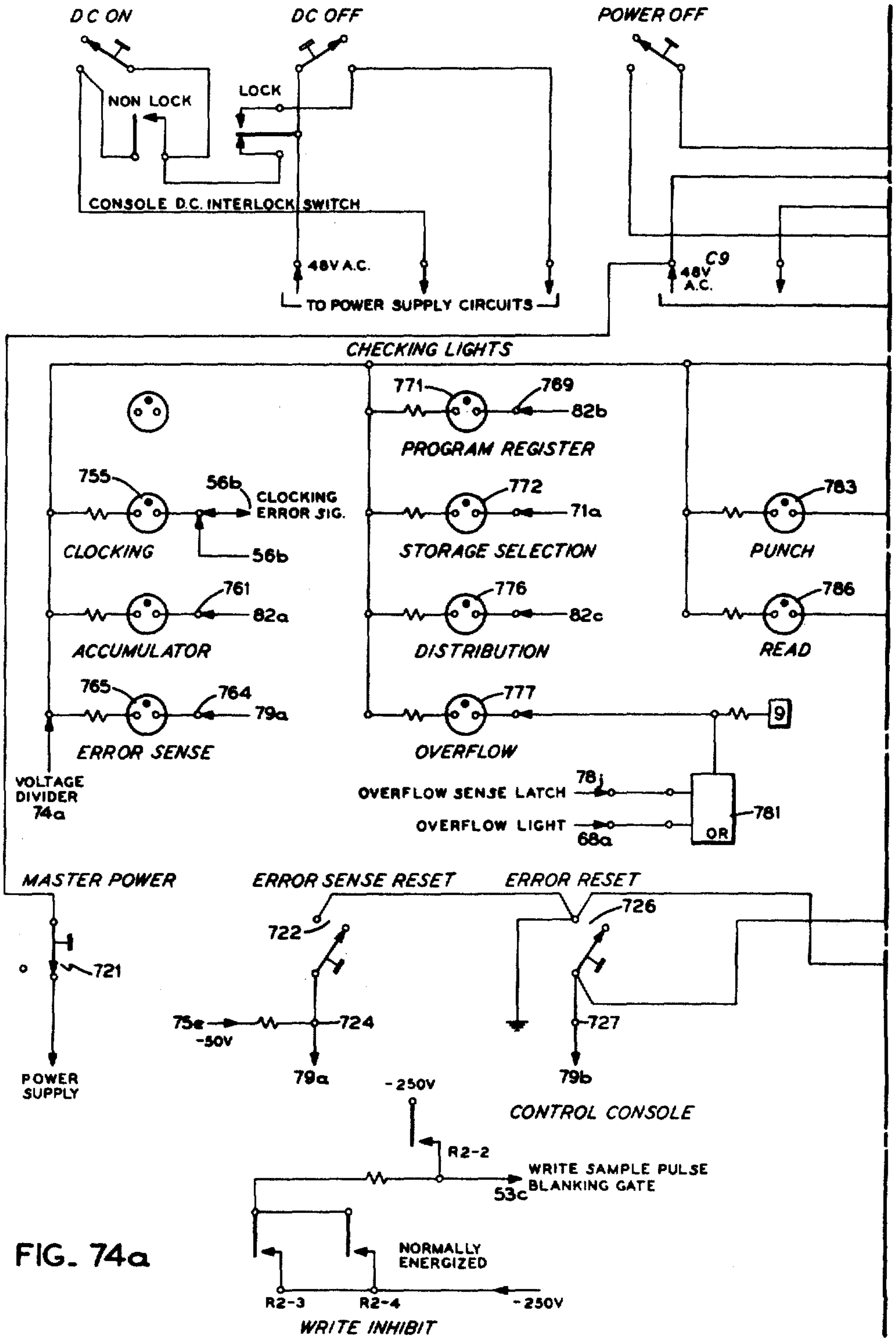

DC ON
DC OFF
POWER OFF
NON LOCK
LOCK
CONSOLE D.C. INTERLOCK SWITCH
48V A.C.
TO POWER SUPPLY CIRCUITS
C9
48V A.C.
CHECKING LIGHTS
771
769
82b
PROGRAM REGISTER
755
56b
CLOCKING ERROR SIG.
56b
CLOCKING
772
71a
STORAGE SELECTION
783
PUNCH
761
82a
ACCUMULATOR
776
82c
DISTRIBUTION
786
READ
765
764
79a
ERROR SENSE
777
OVERFLOW
9
VOLTAGE DIVIDER 74a
78j
OVERFLOW SENSE LATCH
OVERFLOW LIGHT
68a
OR
781
MASTER POWER
721
POWER SUPPLY
ERROR SENSE RESET
722
75e
-50V
724
79a
ERROR RESET
726
727
79b
CONTROL CONSOLE
-250V
R2-2
WRITE SAMPLE PULSE BLANKING GATE
53c
NORMALLY ENERGIZED
R2-3
R2-4
-250V
WRITE INHIBIT

FIG. 74a

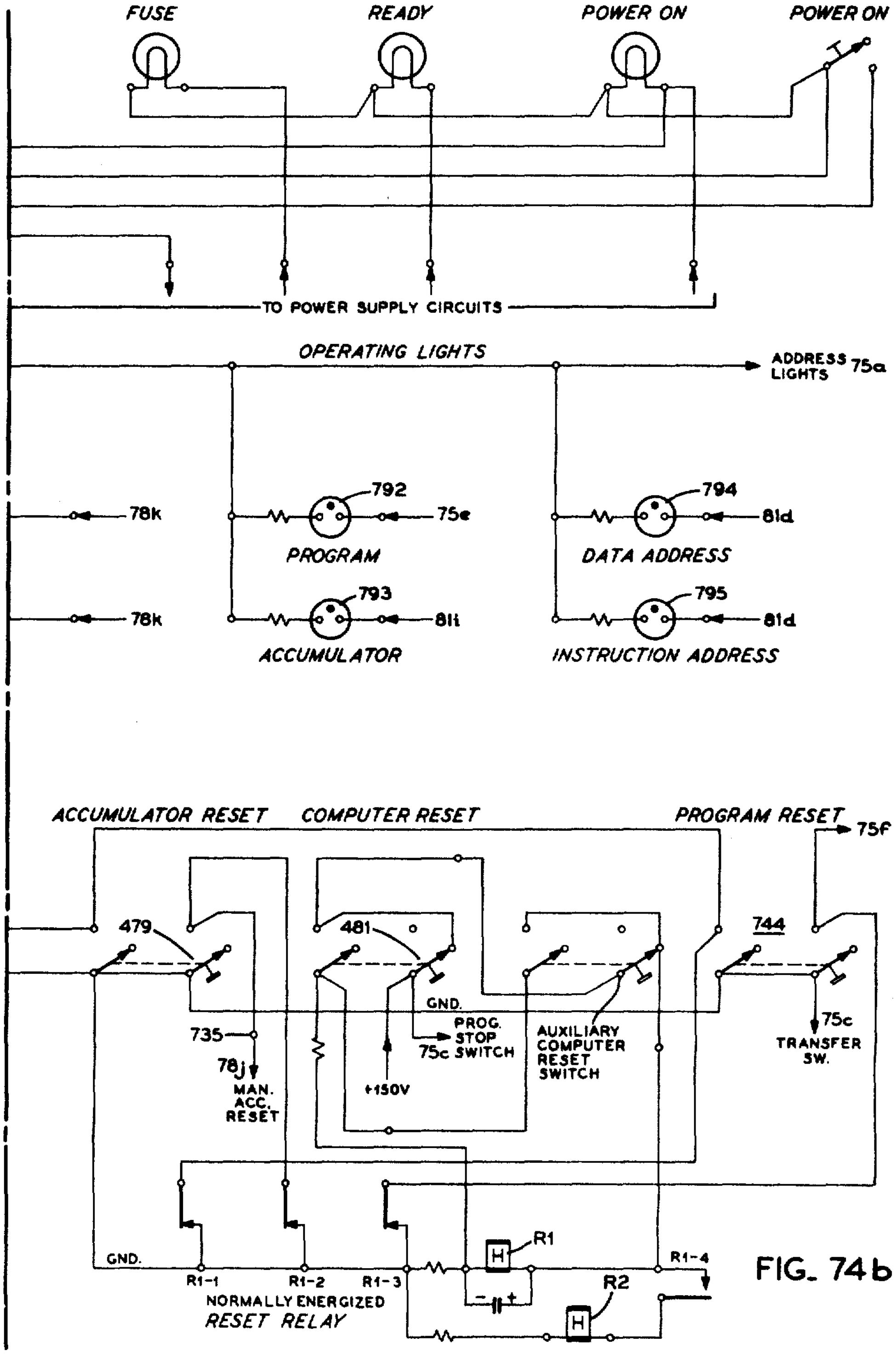

FUSE
READY
POWER ON
POWER ON
TO POWER SUPPLY CIRCUITS
OPERATING LIGHTS
ADDRESS LIGHTS 75a
78k
792
75e
PROGRAM
794
81d
DATA ADDRESS
78k
793
81i
ACCUMULATOR
795
81d
INSTRUCTION ADDRESS
ACCUMULATOR RESET
COMPUTER RESET
PROGRAM RESET
75f
479
481
744
GND.
735
78j
MAN. ACC. RESET
PROG. STOP 75c SWITCH
+150V
AUXILIARY COMPUTER RESET SWITCH
75c
TRANSFER SW.
GND.
R1-1
R1-2
R1-3
R1
R1-4
R2
NORMALLY ENERGIZED
RESET RELAY

FIG. 74b

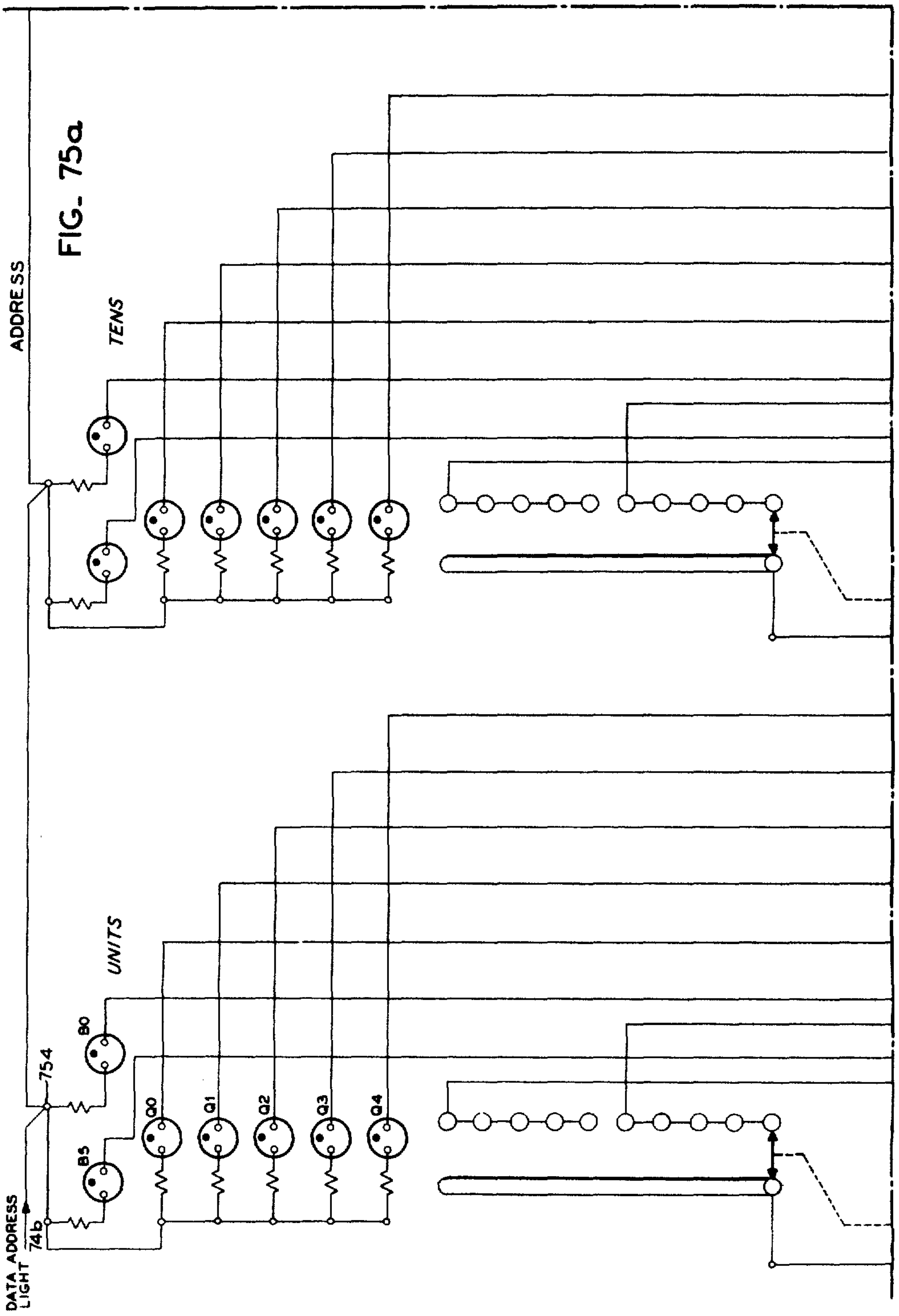


FIG. 75a
ADDRESS
TENS
UNITS
B0
B5
Q0
Q1
Q2
Q3
Q4
754
DATA ADDRESS LIGHT
74b

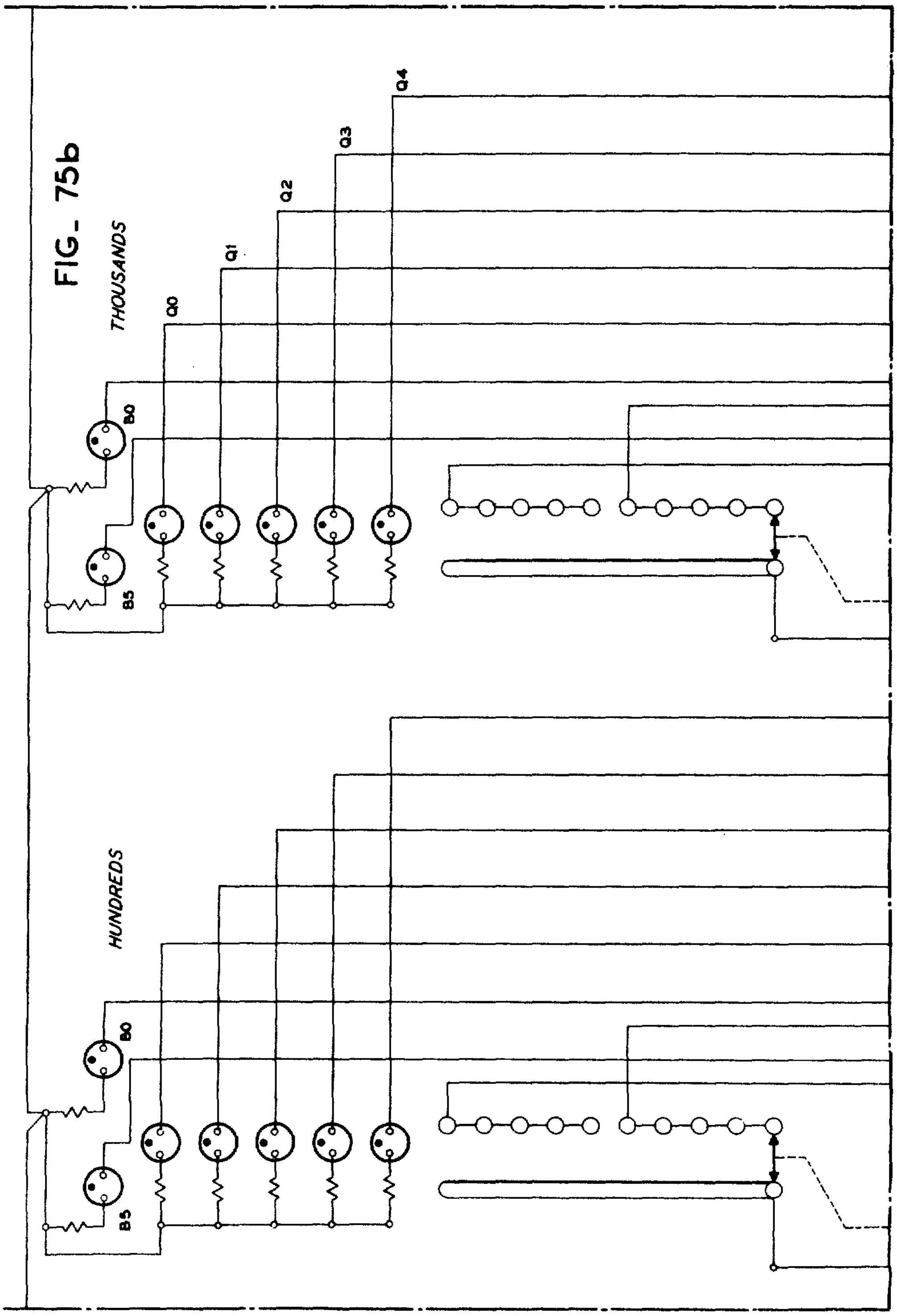
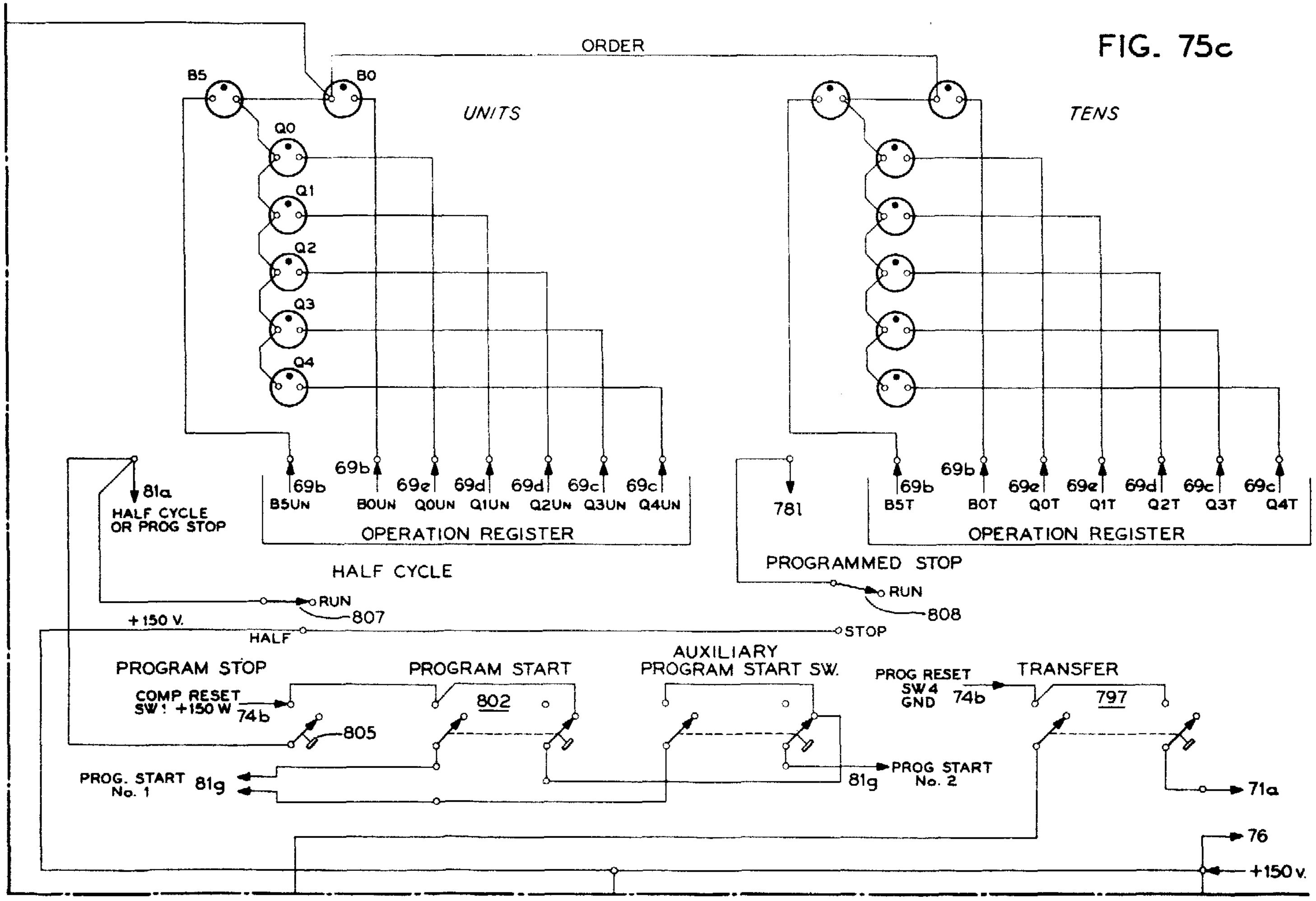


FIG. 75b

FIG. 75c


ORDER
B5
B0
UNITS
TENS
Q0
Q1
Q2
Q3
Q4
69b
69b
69e
69d
69d
69c
69c
B5UN
B0UN
Q0UN
Q1UN
Q2UN
Q3UN
Q4UN
OPERATION REGISTER
69b
69b
69e
69e
69d
69c
69c
B5T
B0T
Q0T
Q1T
Q2T
Q3T
Q4T
OPERATION REGISTER
81a
HALF CYCLE
OR PROG STOP
HALF CYCLE
RUN
807
+150 V.
HALF
STOP
781
PROGRAMMED STOP
RUN
808
PROGRAM STOP
COMP RESET
SW 1 +150 W
74b
805
PROGRAM START
802
AUXILIARY
PROGRAM START SW.
PROG RESET
SW 4
GND
74b
TRANSFER
797
PROG. START
No. 1
81g
81g
PROG START
No. 2
71a
76
+150 v.

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 146

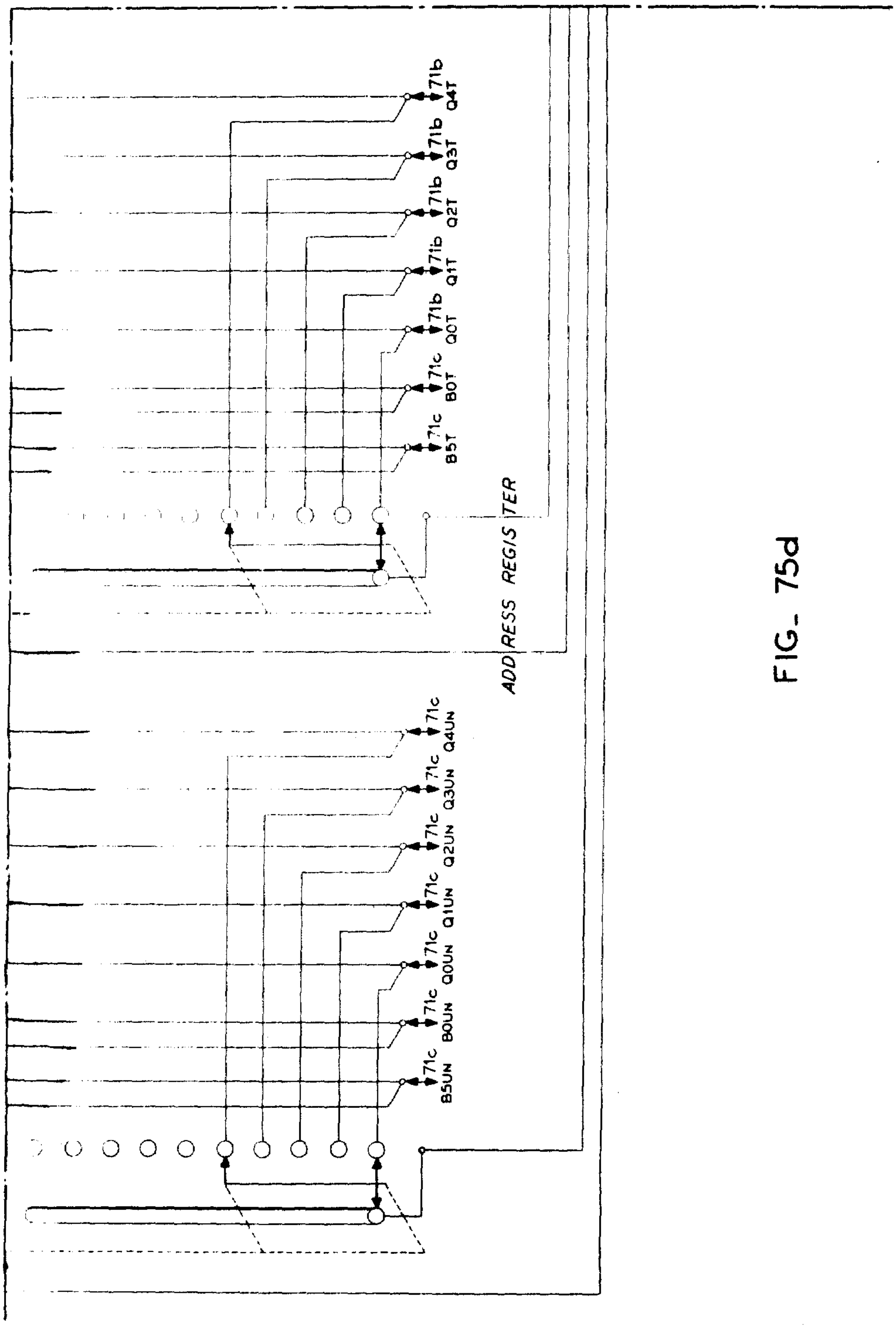


FIG. 75d

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 147

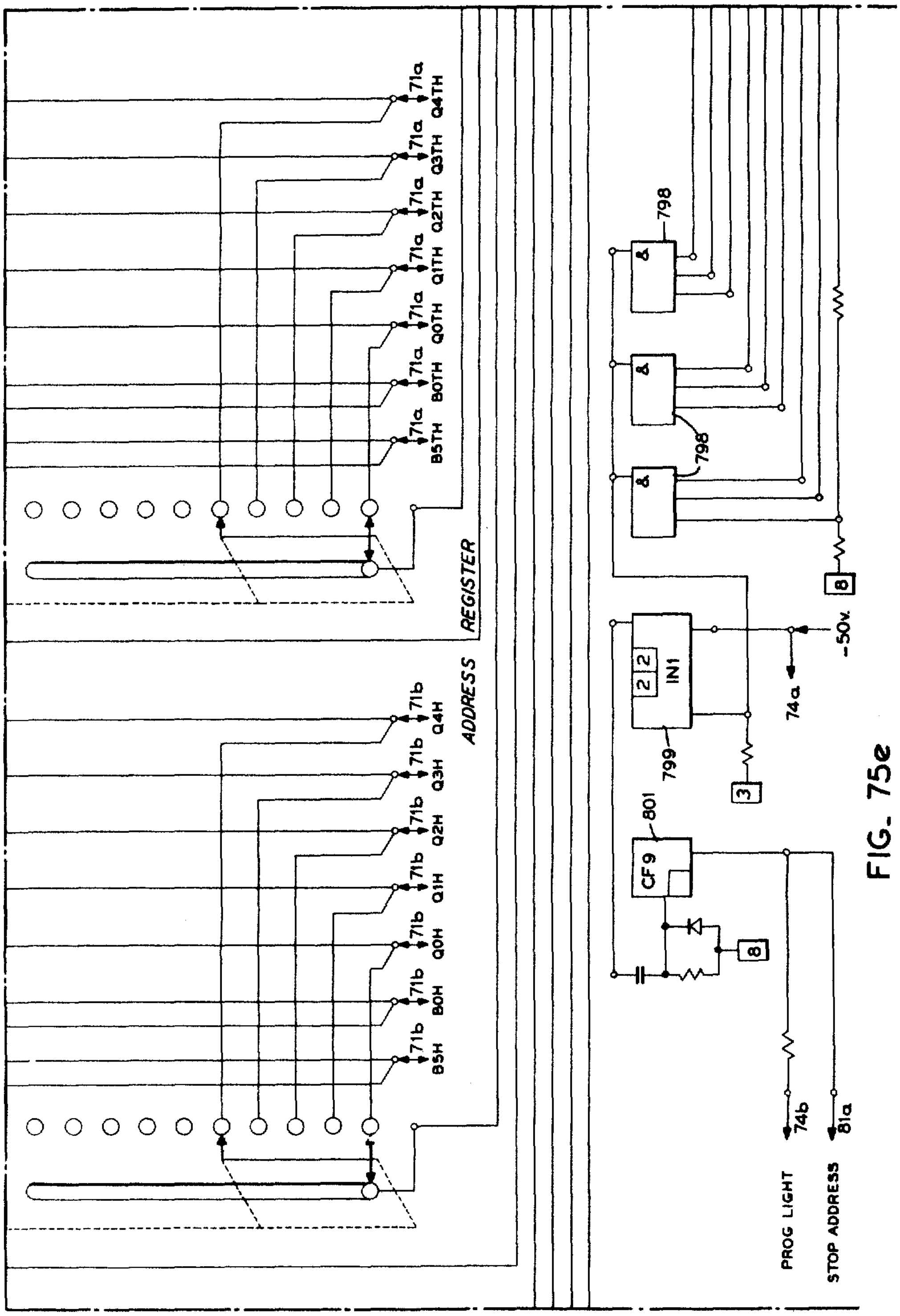


FIG. 75e

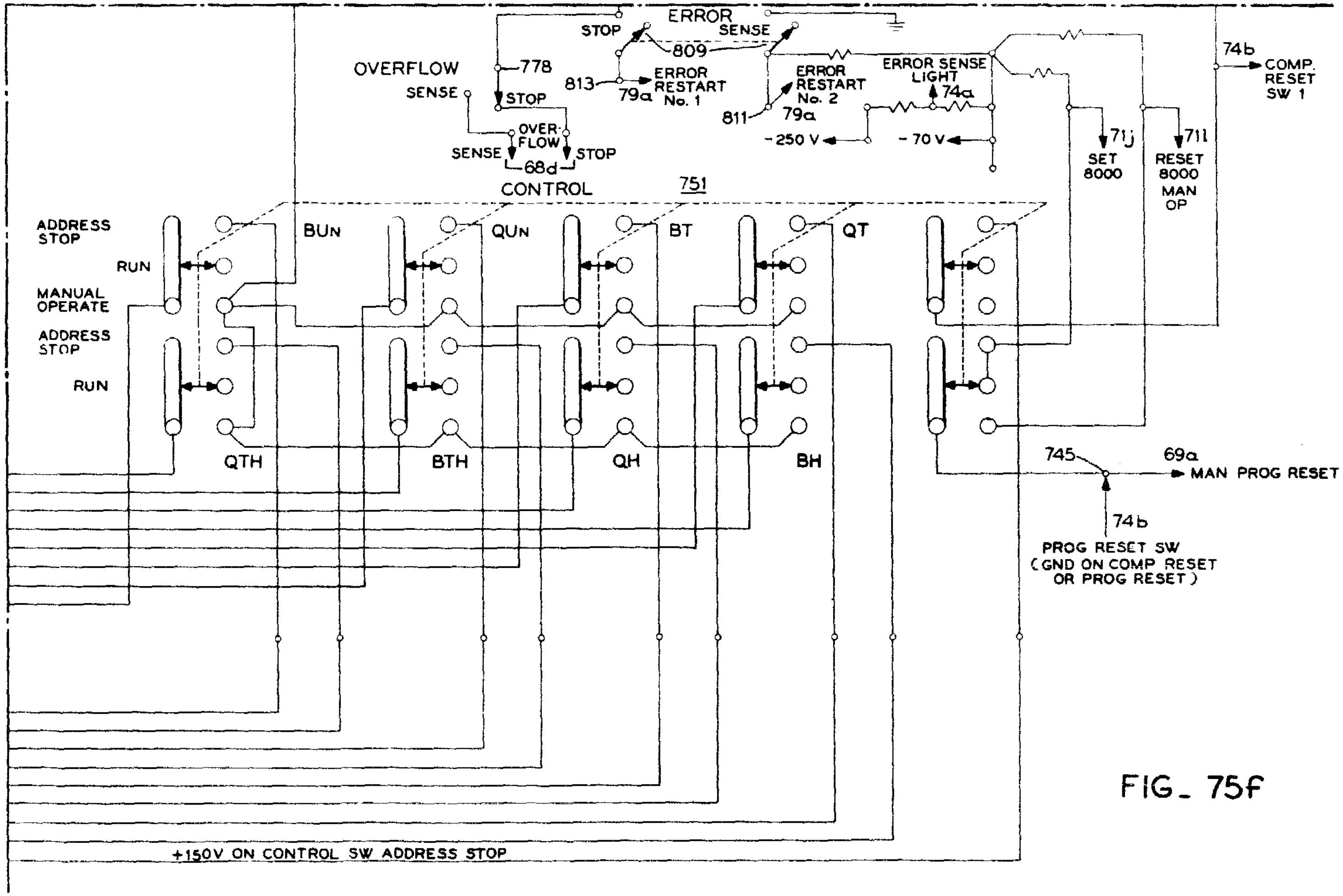

ERROR
STOP
SENSE
809
OVERFLOW
SENSE
778
STOP
813
ERROR RESTART No. 1
79a
ERROR RESTART No. 2
811
79a
ERROR SENSE LIGHT
74a
74b
COMP. RESET SW 1
OVER-FLOW
SENSE
STOP
68d
-250 V
-70 V
71j
SET 8000
711
RESET 8000 MAN OP
CONTROL
751
ADDRESS STOP
RUN
MANUAL OPERATE
ADDRESS STOP
RUN
BUN
QUN
BT
QT
QTH
BTH
QH
BH
745
69a
MAN PROG RESET
74b
PROG RESET SW (GND ON COMP RESET OR PROG RESET)
+150V ON CONTROL SW ADDRESS STOP

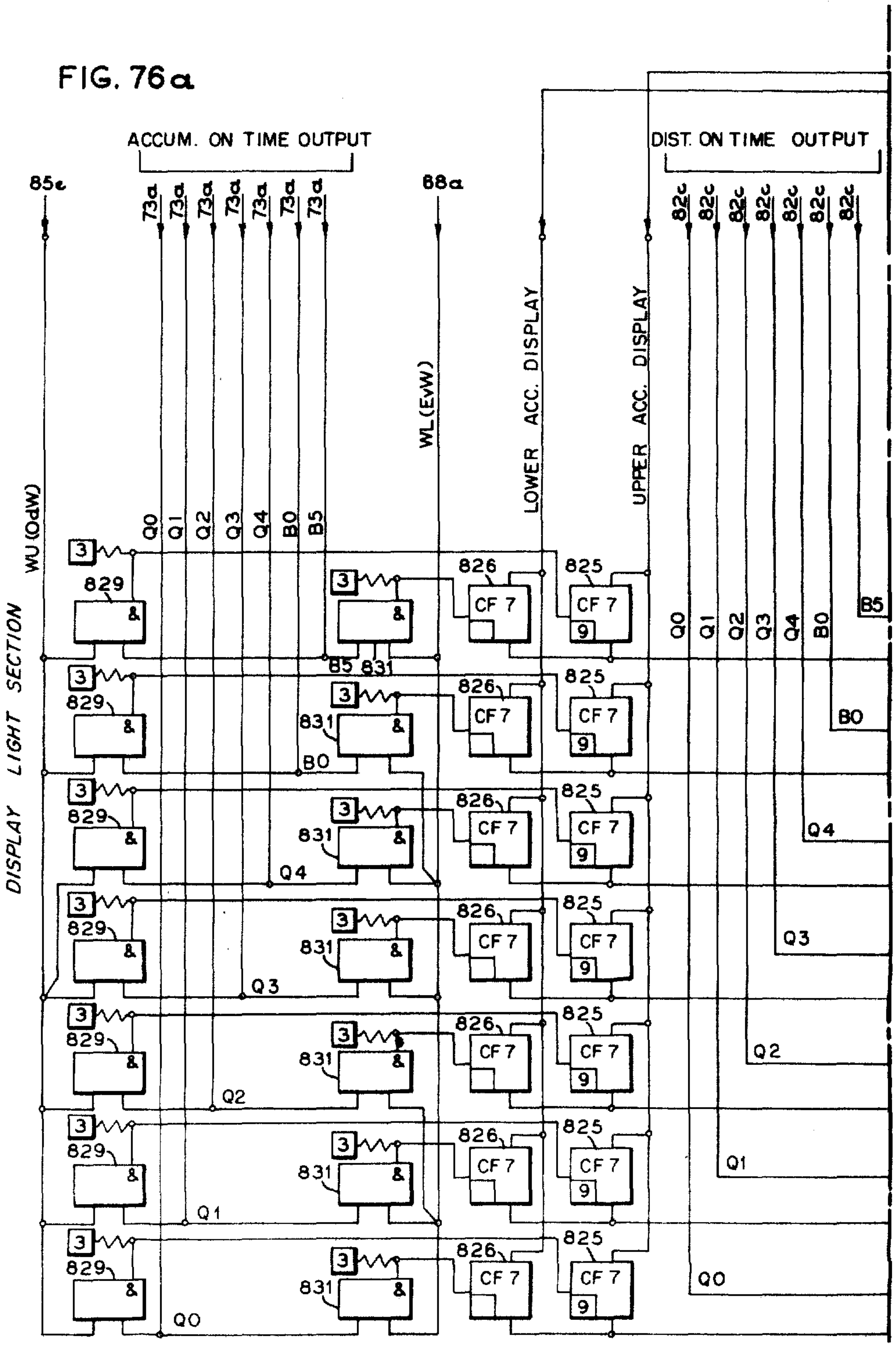
FIG_ 75F

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 149

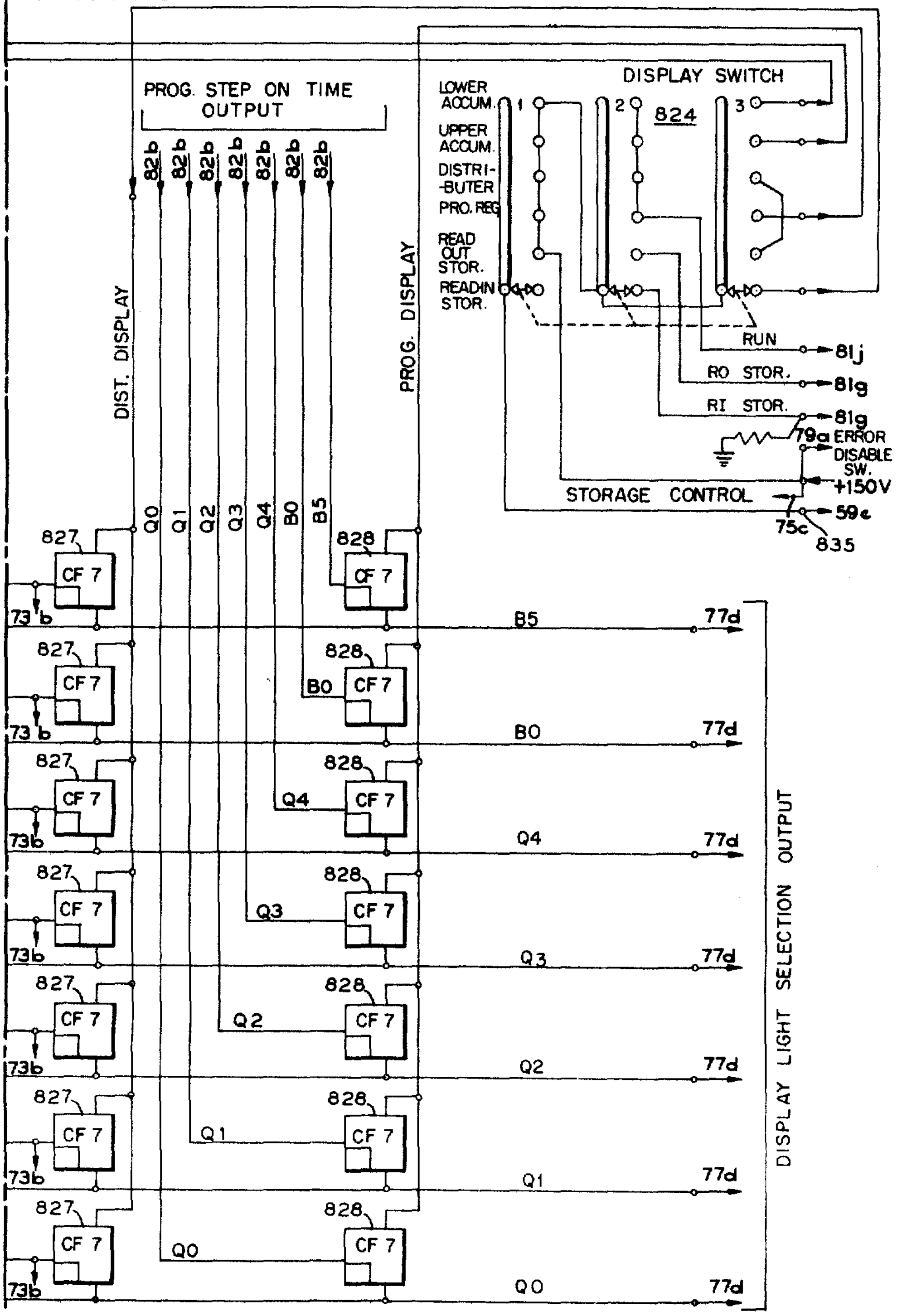
FIG. 76a



DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 150

FIG. 76 b



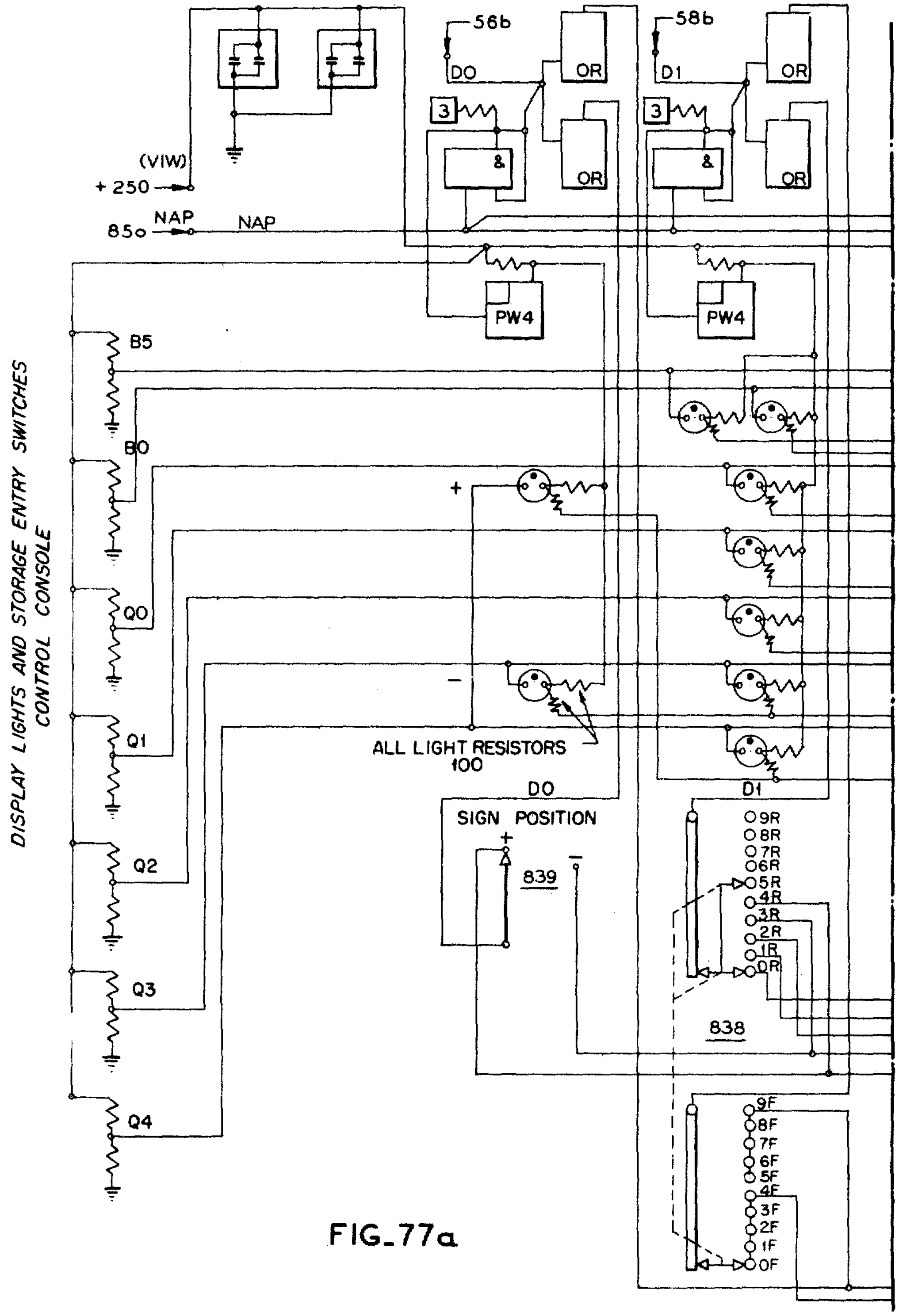

DISPLAY LIGHTS AND STORAGE ENTRY SWITCHES
CONTROL CONSOLE
(VIW)
+250
85o
NAP
NAP
56b
DO
OR
OR
3
&
PW4
58b
D1
OR
OR
3
&
PW4
B5
B0
Q0
Q1
Q2
Q3
Q4
+
−
ALL LIGHT RESISTORS
100
DO
D1
SIGN POSITION
+
−
839
9R
8R
7R
6R
5R
4R
3R
2R
1R
0R
838
9F
8F
7F
6F
5F
4F
3F
2F
1F
0F

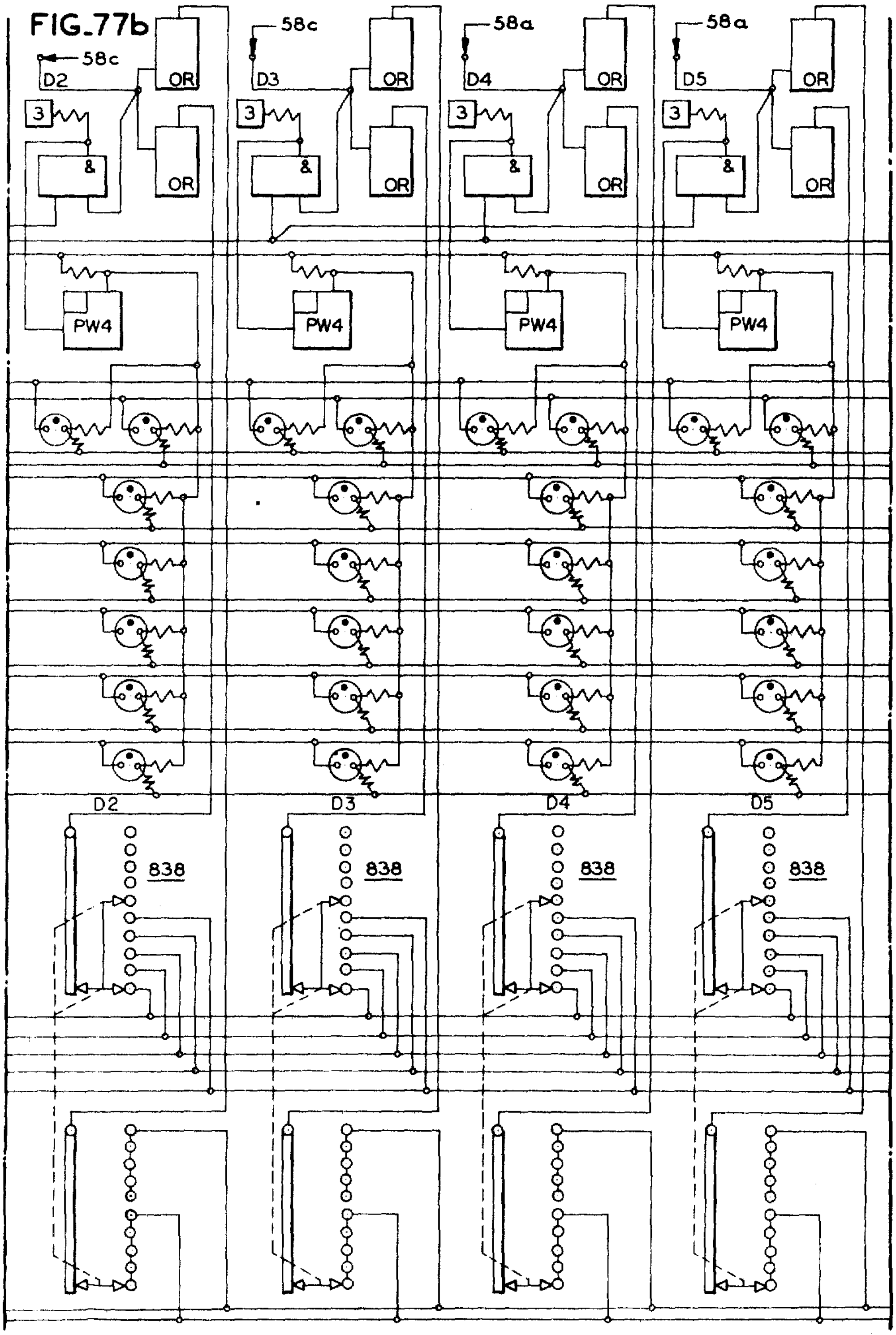
FIG_77a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

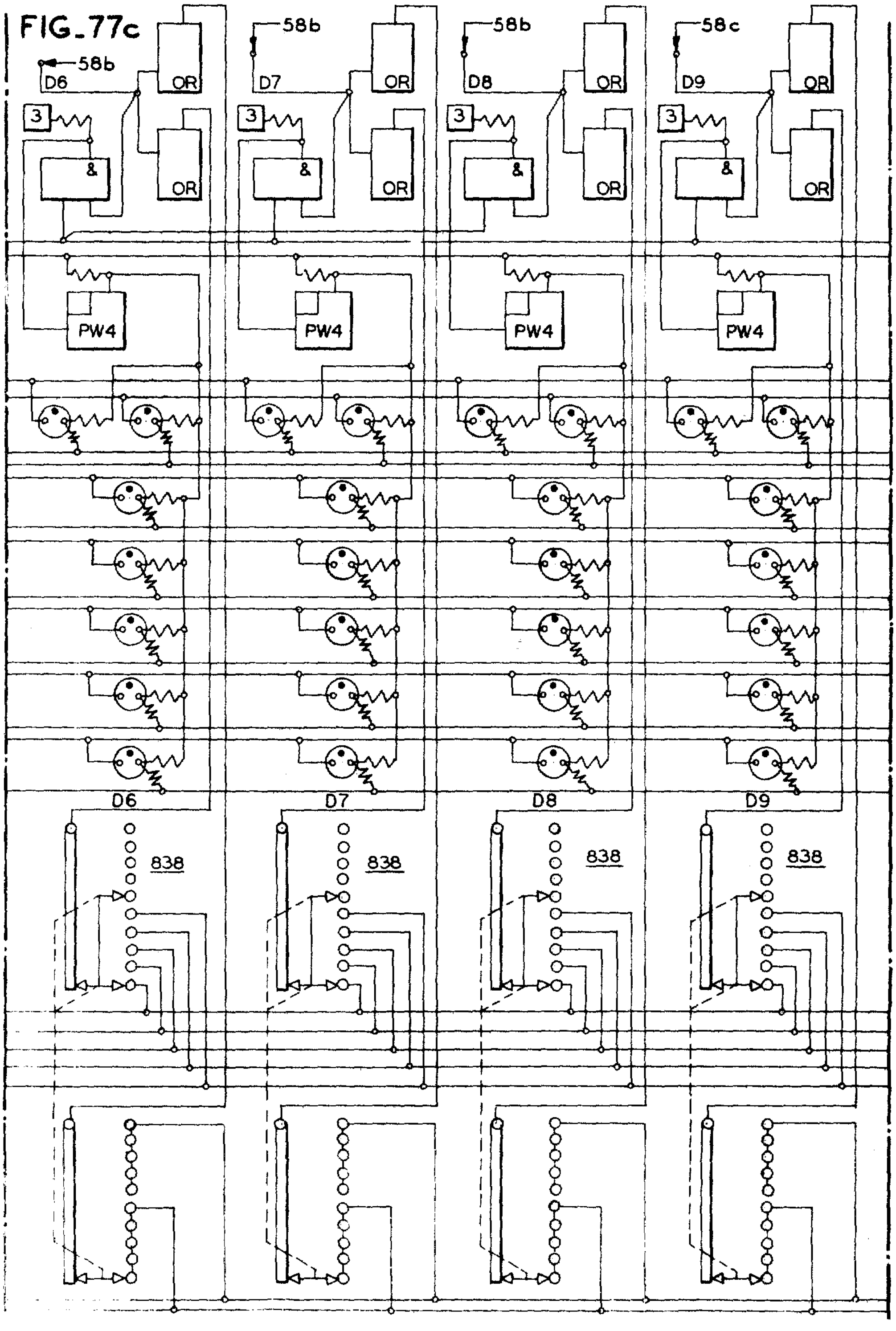
Filed Nov. 2, 1955 228 Sheets-Sheet 152

FIG. 77b



DATA STORAGE AND PROCESSING MACHINE

FIG_77c



Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 154

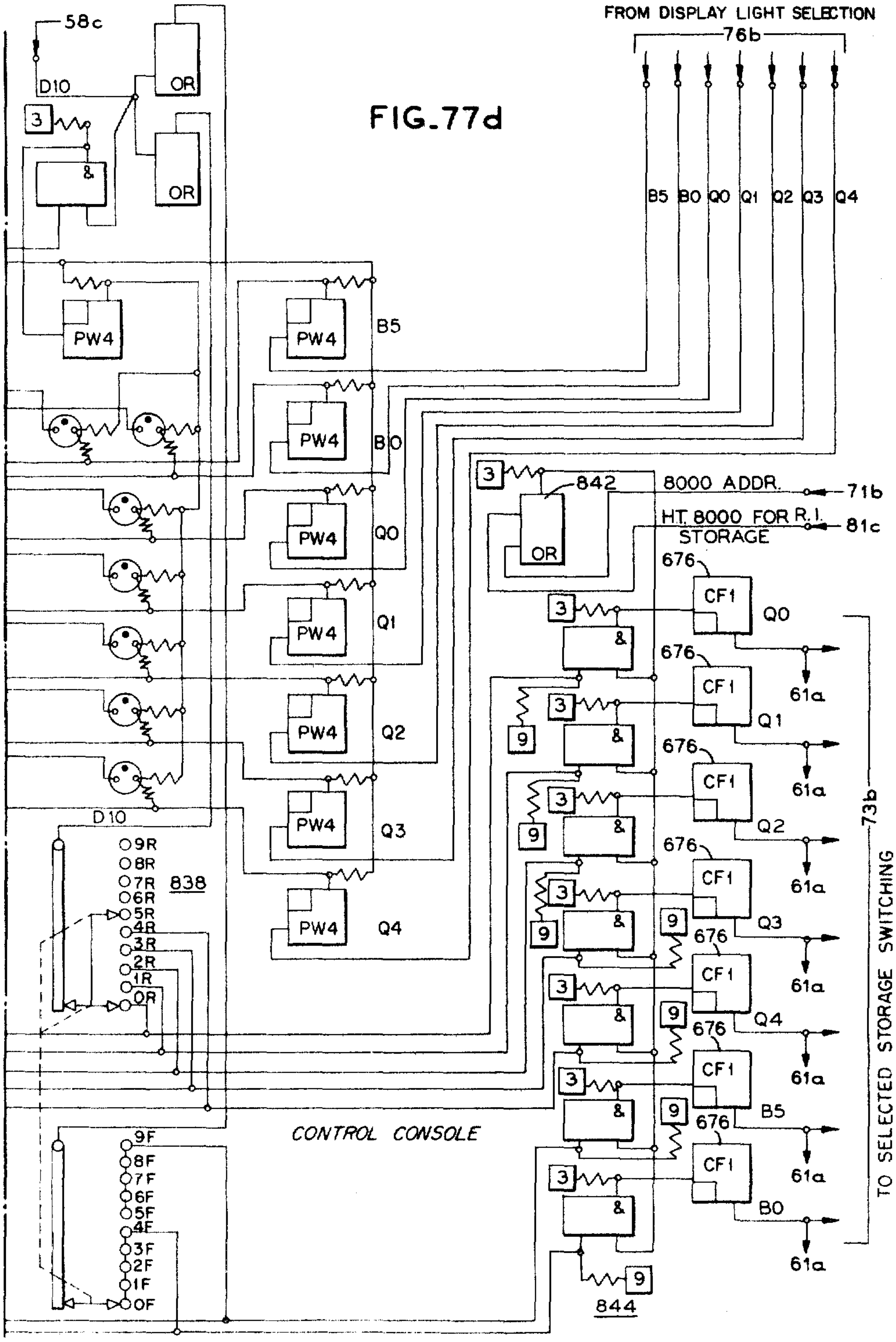
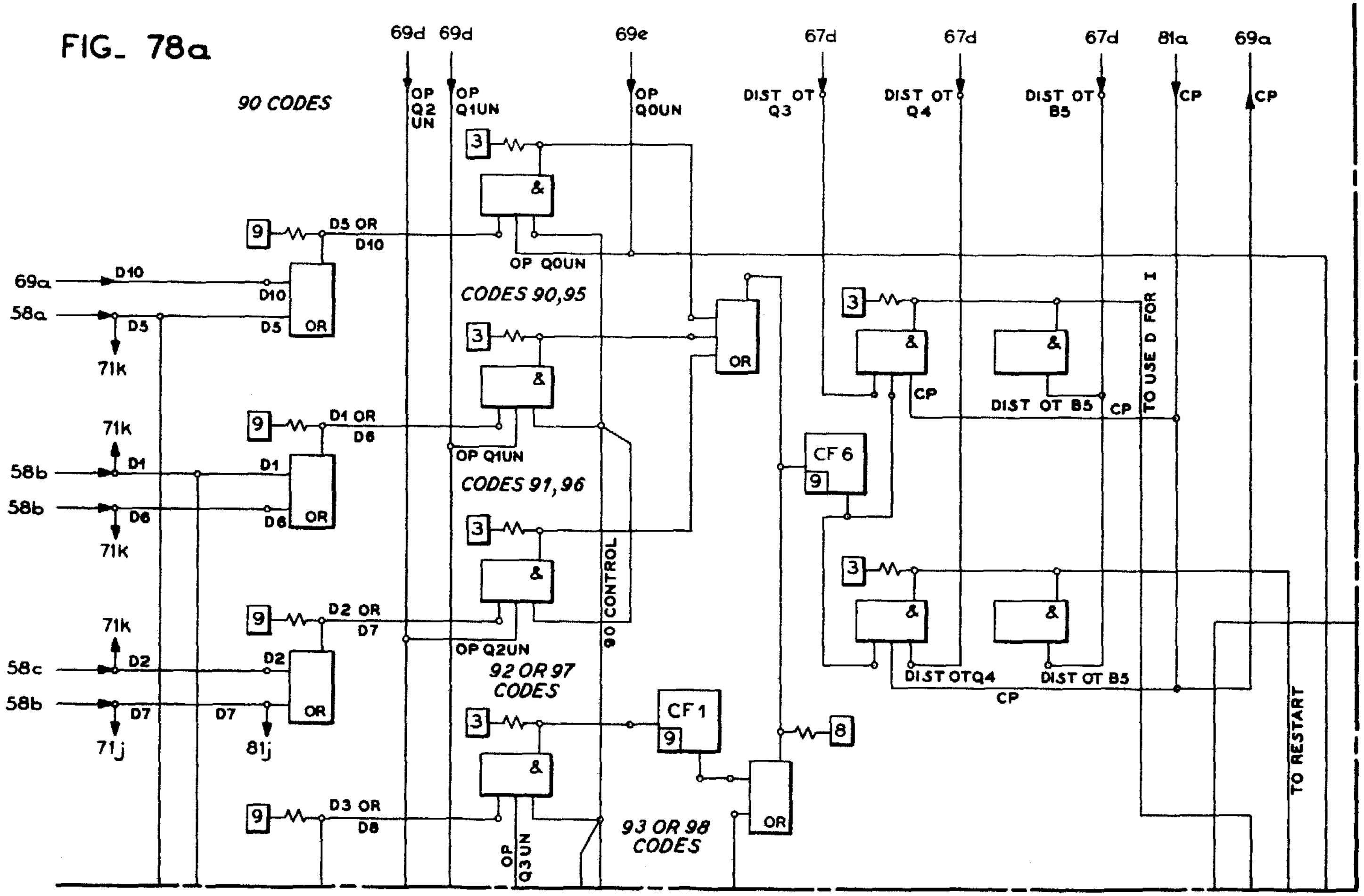


FIG. 77d

Filed Nov. 2, 1955

DATA STORAGE AND PROCESSING MACHINE

FIG. 78a



Nov. 8, 1960

F. E. HAMILTON ET AL

2,959,351

Filed Nov. 2, 1955

DATA STORAGE AND PROCESSING MACHINE

228 Sheets-Sheet 156

FIG. 78b

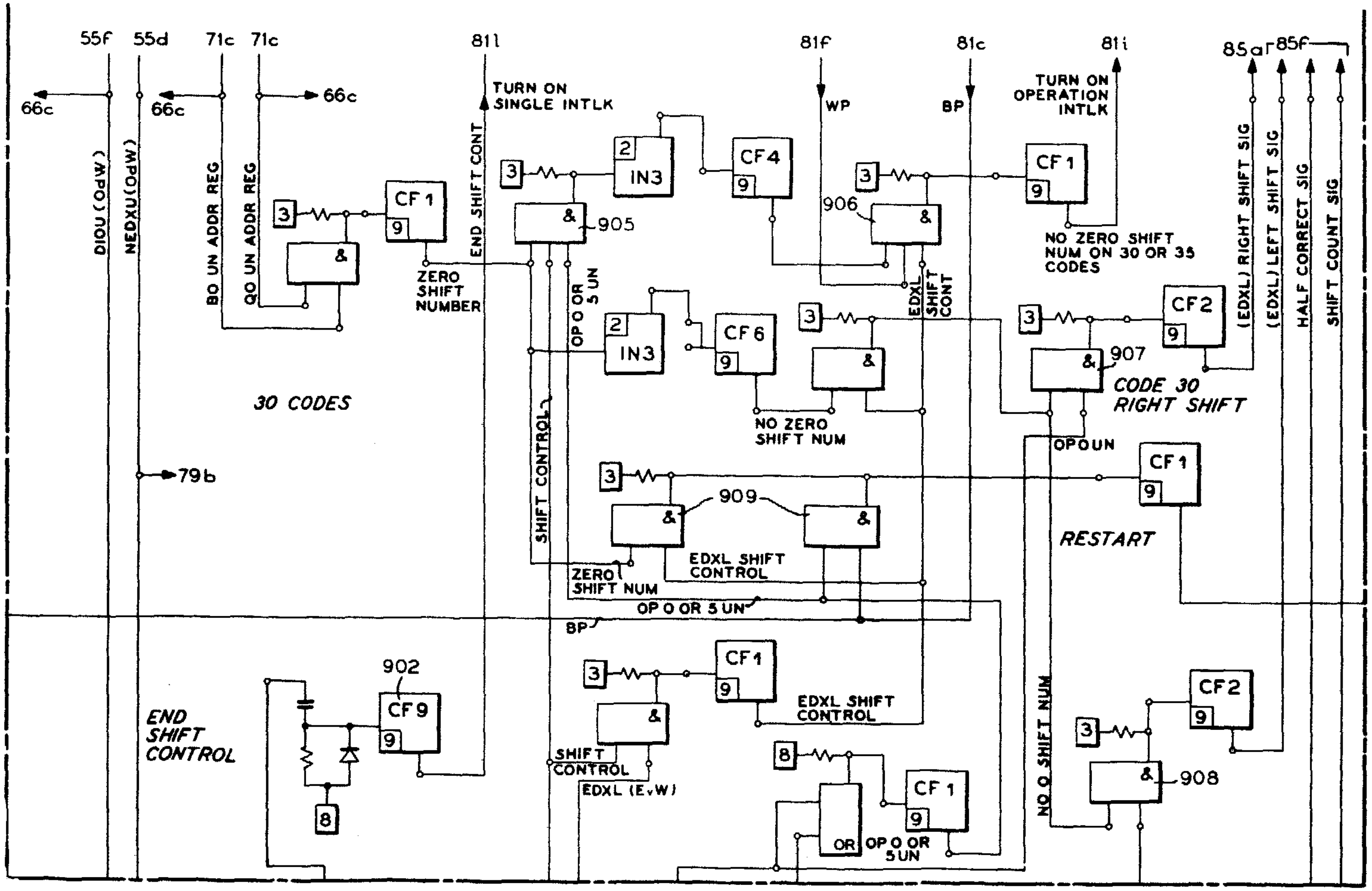
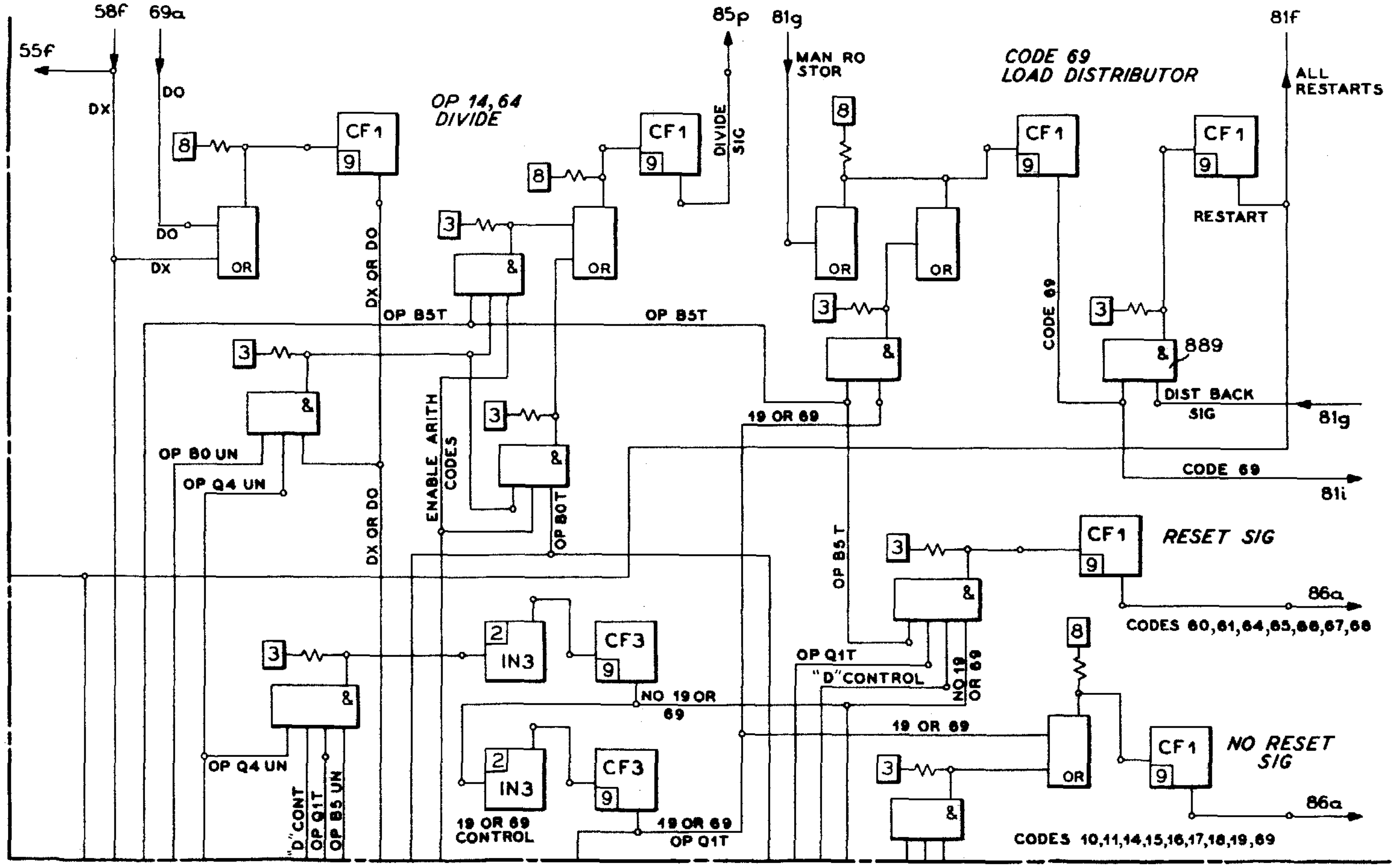


FIG. 78c


58f
69a
55f
DX
DO
8
CF1
9
DO
DX
OR
OP 14,64
DIVIDE
85p
DIVIDE SIG
8
CF1
9
3
&
OR
DX OR DO
OP B5T
3
&
OP 80 UN
OP Q4 UN
ENABLE ARITH CODES
3
&
OP B0T
DX OR DO
81g
MAN RO STOR
8
OR
OR
3
&
19 OR 69
OP B5T
CODE 69
LOAD DISTRIBUTOR
CF1
9
CODE 69
81f
ALL RESTARTS
CF1
9
RESTART
3
&
889
DIST BACK SIG
81g
CODE 69
81i
3
CF1
9
RESET SIG
&
86a
CODES 60,61,64,65,66,67,68
OP Q1T
"D" CONTROL
NO 19 OR 69
8
3
&
2
IN3
CF3
9
NO 19 OR 69
2
IN3
CF3
9
19 OR 69 CONTROL
19 OR 69
OP Q1T
OP Q4 UN
"D" CONT
OP Q1T
OP B5 UN
19 OR 69
OR
CF1
9
NO RESET SIG
3
&
86a
CODES 10,11,14,15,16,17,18,19,69

DATA STORAGE AND PROCESSING MACHINE

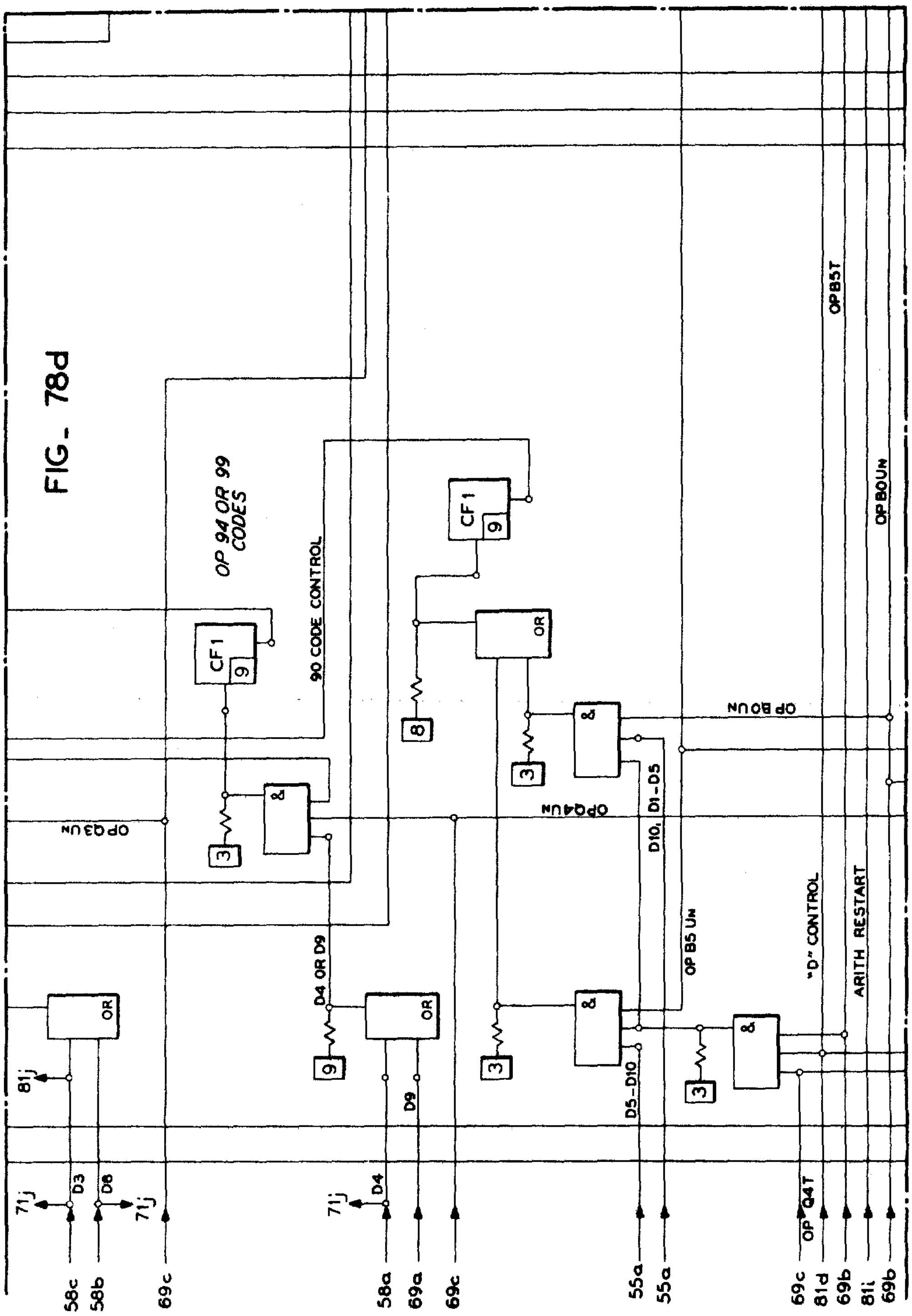


FIG. 78d

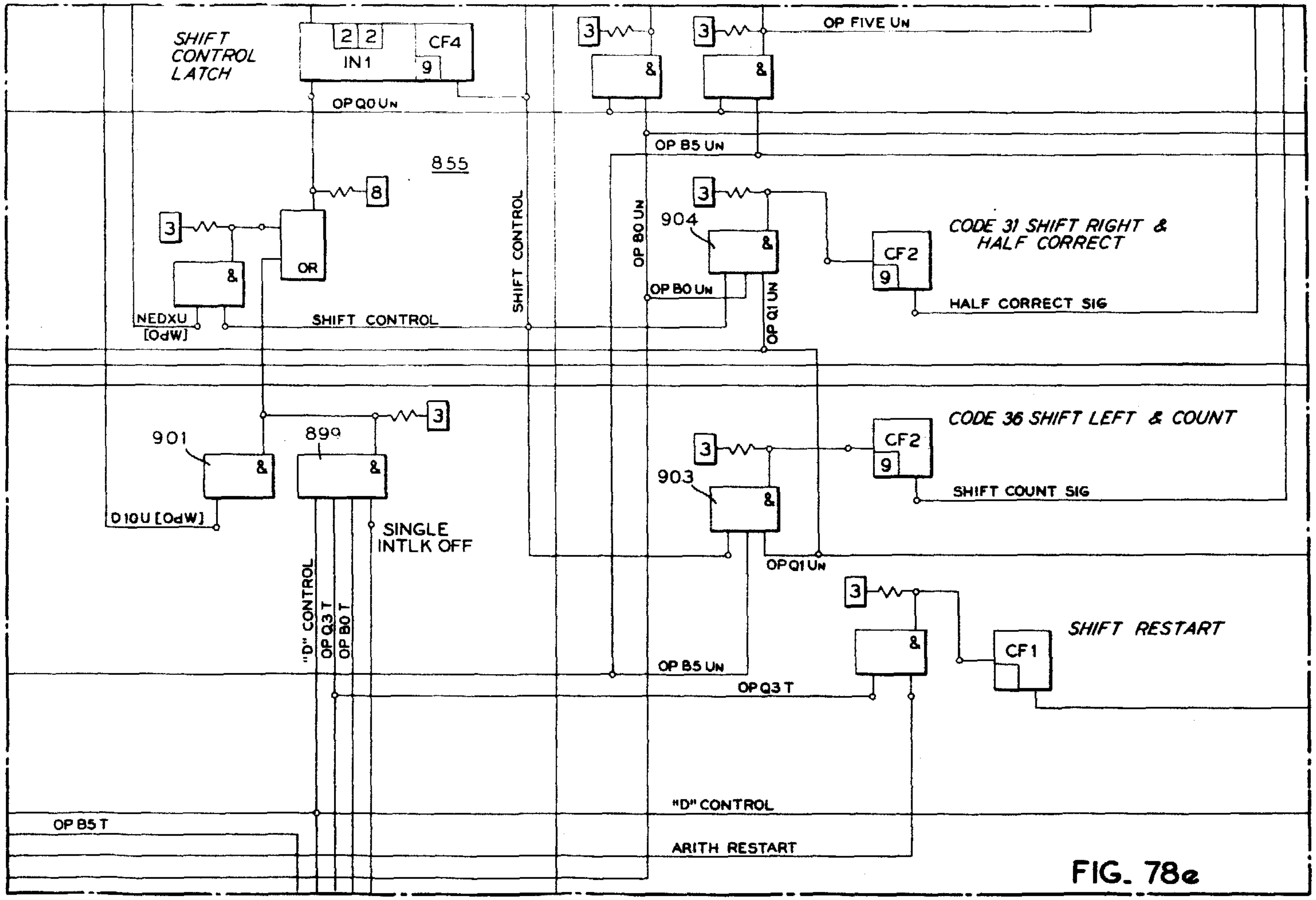
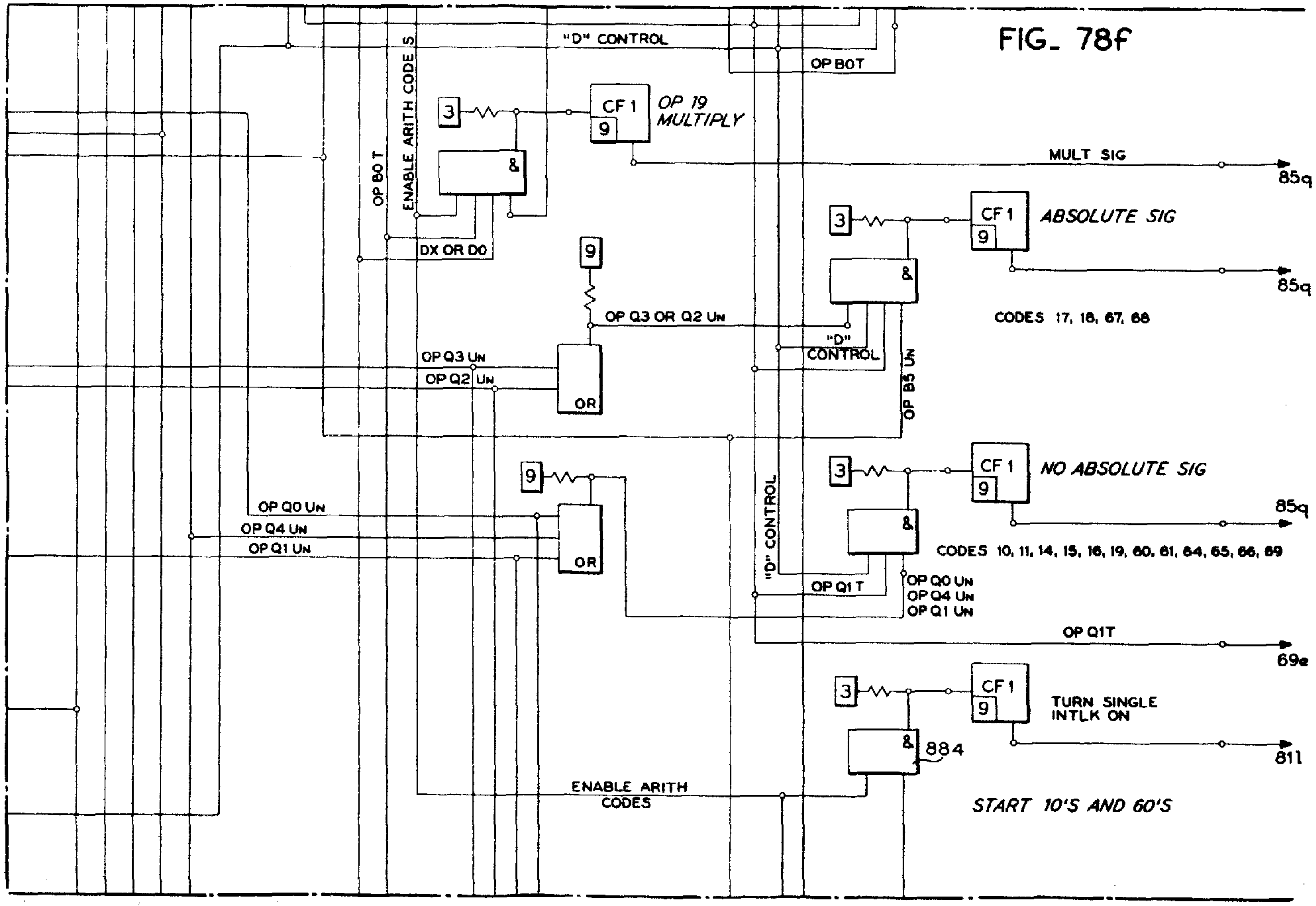

SHIFT CONTROL LATCH
2 2
IN1
CF4
9
OP Q0 UN
855
8
3
OR
&
NEDXU [OdW]
SHIFT CONTROL
SHIFT CONTROL
3
901
899
&
&
D10U [OdW]
SINGLE INTLK OFF
"D" CONTROL
OP Q3 T
OP B0 T
3
3
OP FIVE UN
OP B5 UN
OP B0 UN
904
3
CF2
9
CODE 31 SHIFT RIGHT & HALF CORRECT
OP B0 UN
OP Q1 UN
HALF CORRECT SIG
3
903
CF2
9
CODE 36 SHIFT LEFT & COUNT
SHIFT COUNT SIG
OP Q1 UN
3
&
CF1
SHIFT RESTART
OP B5 UN
OP Q3 T
OP B5 T
"D" CONTROL
ARITH RESTART

FIG. 78e

FIG_ 78f


"D" CONTROL
OP BOT
ENABLE ARITH CODES
OP BOT
3
CF 1
9
OP 19
MULTIPLY
&
DX OR DO
MULT SIG
85q
3
CF 1
9
ABSOLUTE SIG
&
85q
CODES 17, 18, 67, 68
9
OP Q3 OR Q2 UN
"D"
CONTROL
OP B5 UN
OP Q3 UN
OP Q2 UN
OR
9
3
CF 1
9
NO ABSOLUTE SIG
85q
&
CODES 10, 11, 14, 15, 16, 19, 60, 61, 64, 65, 66, 69
OP Q0 UN
OP Q4 UN
OP Q1 UN
OR
"D" CONTROL
OP Q1 T
OP Q0 UN
OP Q4 UN
OP Q1 UN
OP Q1T
69e
3
CF 1
9
TURN SINGLE
INTLK ON
&
884
811
ENABLE ARITH
CODES
START 10'S AND 60'S

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 161

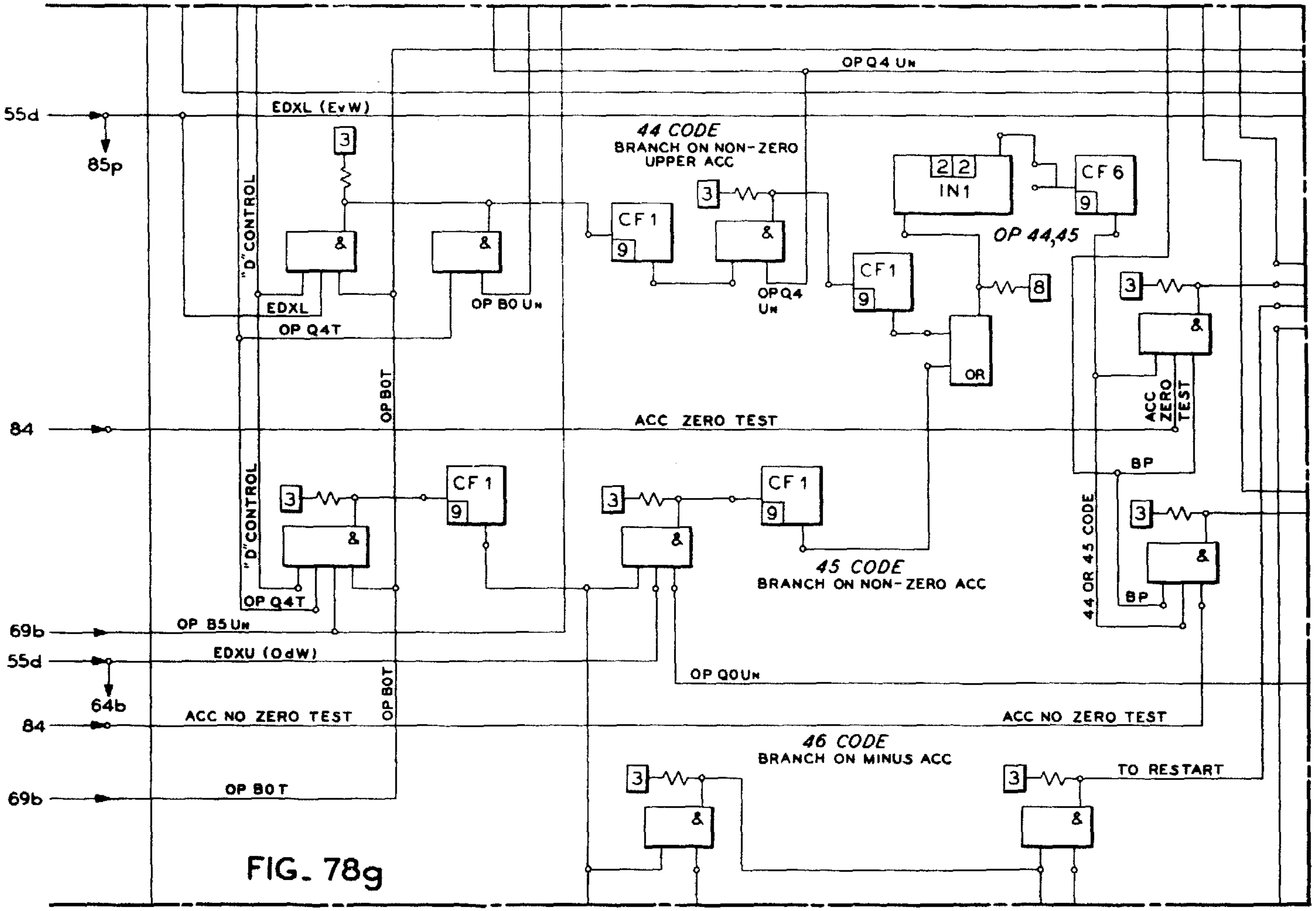


FIG. 78g

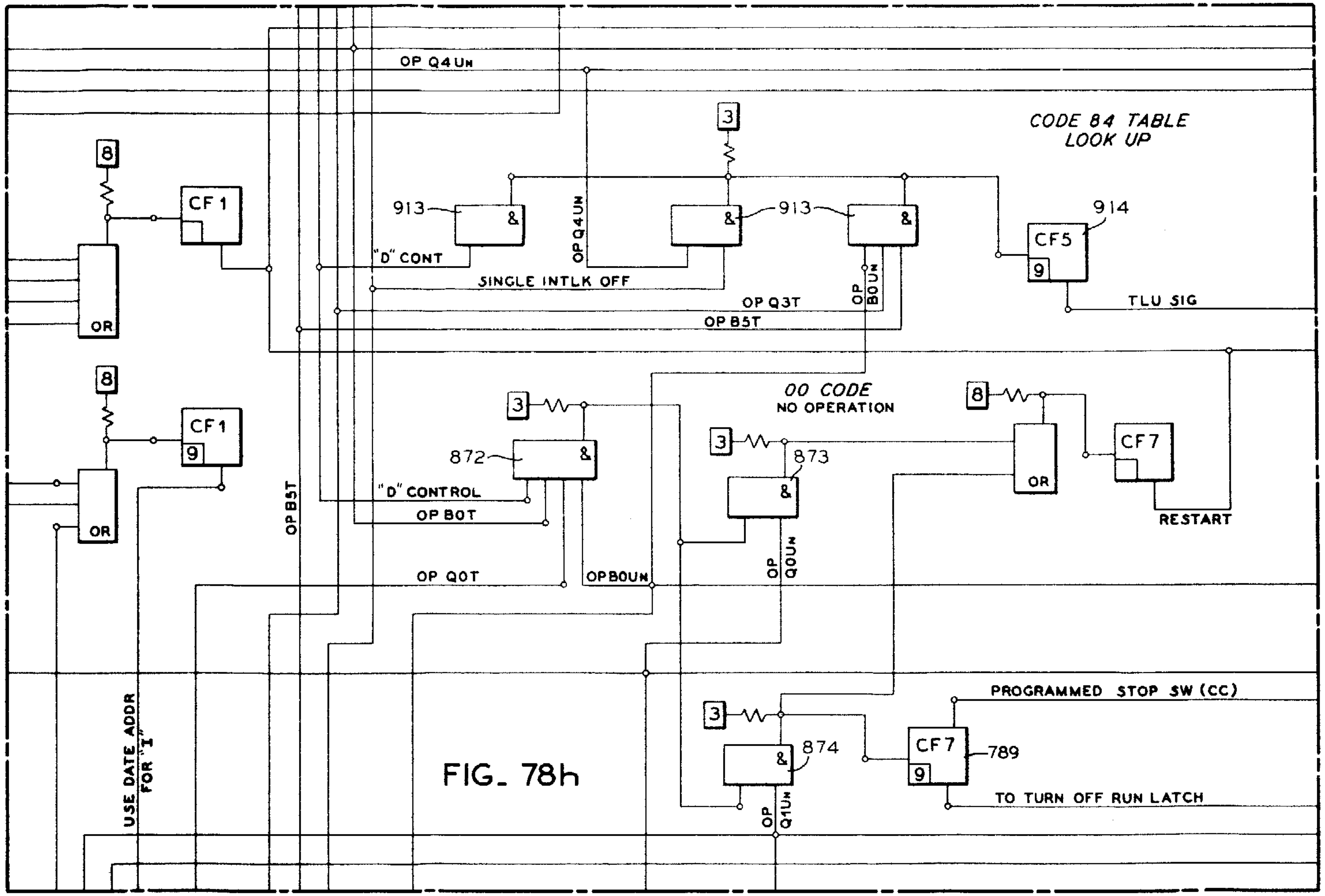

OP Q4 UN
CODE 84 TABLE
LOOK UP
8
CF 1
913
&
"D" CONT
OP Q4 UN
3
913
CF5
9
914
SINGLE INTLK OFF
OP Q3T
OP B0 UN
OP B5T
TLU SIG
OR
8
CF 1
9
OR
3
872
&
"D" CONTROL
OP B0T
OP B5T
00 CODE
NO OPERATION
3
873
&
8
OR
CF7
RESTART
OP Q0T
OPB0UN
OP Q0UN
PROGRAMMED STOP SW (CC)
3
874
&
CF7
9
789
TO TURN OFF RUN LATCH
USE DATE ADDR FOR "I"
OP Q1UN

FIG. 78h

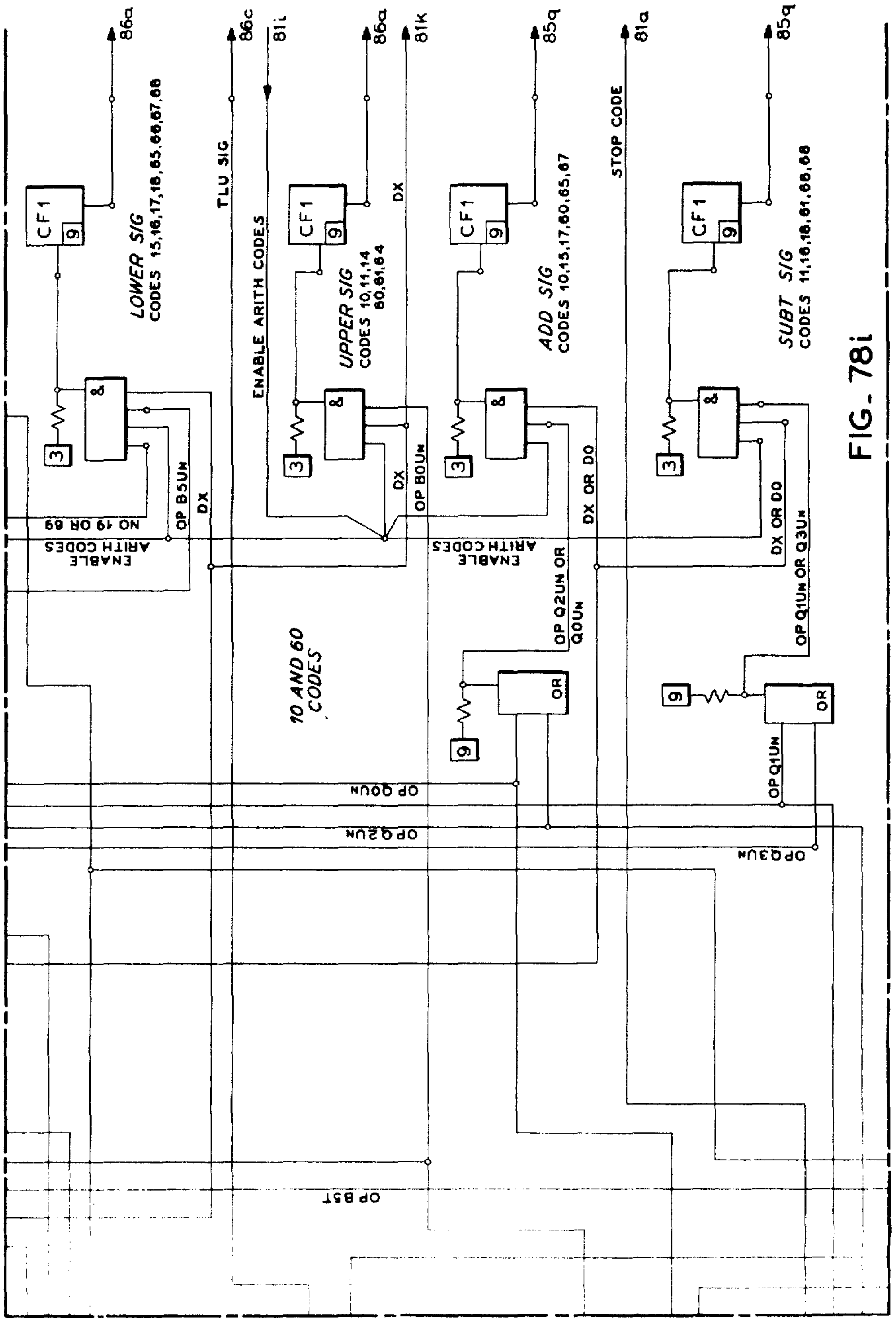

CF1
9
3
&
86a
LOWER SIG
CODES 15,16,17,18,65,66,67,68
ENABLE ARITH CODES
NO 19 OR 69
OP B5UN
DX
TLU SIG
86c
ENABLE ARITH CODES
81i
UPPER SIG
CODES 10,11,14 60,61,64
86a
DX
DX
81K
OP B0UN
10 AND 60 CODES
OP B5T
OP Q2UN
OP Q0UN
ENABLE ARITH CODES
ADD SIG
CODES 10,15,17,60,65,67
85q
OR
OP Q2UN OR Q0UN
DX OR D0
STOP CODE
81a
SUBT SIG
CODES 11,16,18,61,66,68
85q
OP Q3UN
OP Q1UN
DX OR D0
OP Q1UN OR Q3UN

FIG. 78i

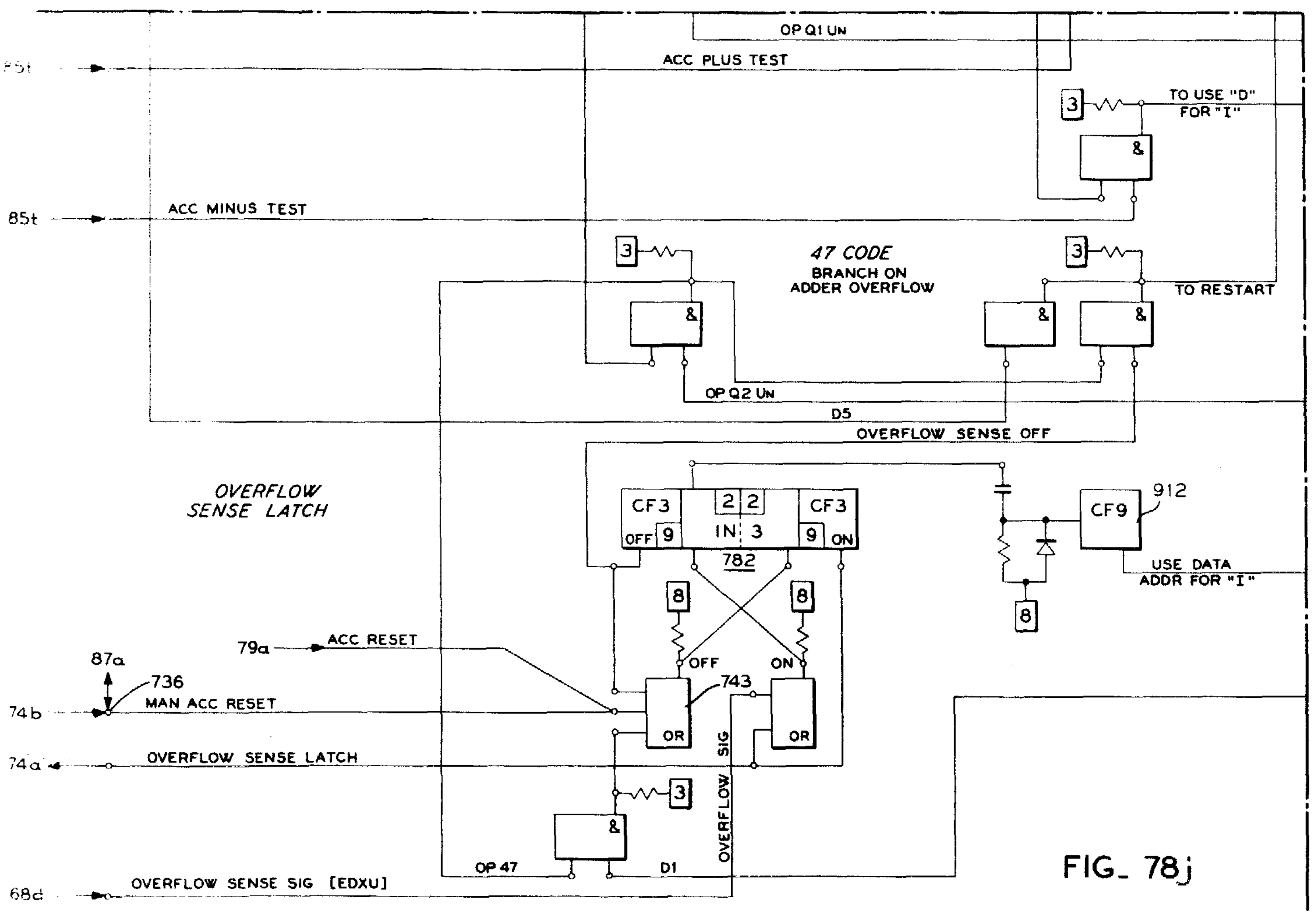

OP Q1 UN
85t
ACC PLUS TEST
3
TO USE "D" FOR "I"
&
85t
ACC MINUS TEST
3
47 CODE
BRANCH ON ADDER OVERFLOW
3
TO RESTART
&
&
&
OP Q2 UN
D5
OVERFLOW SENSE OFF
OVERFLOW SENSE LATCH
CF3
OFF
9
2
2
IN
3
782
CF3
9
ON
CF9
912
USE DATA ADDR FOR "I"
8
8
8
OFF
ON
79a
ACC RESET
87a
736
743
74b
MAN ACC RESET
OR
OR
74a
OVERFLOW SENSE LATCH
SIG
3
&
OVERFLOW
OP 47
D1
68d
OVERFLOW SENSE SIG [EDXU]

FIG_ 78j

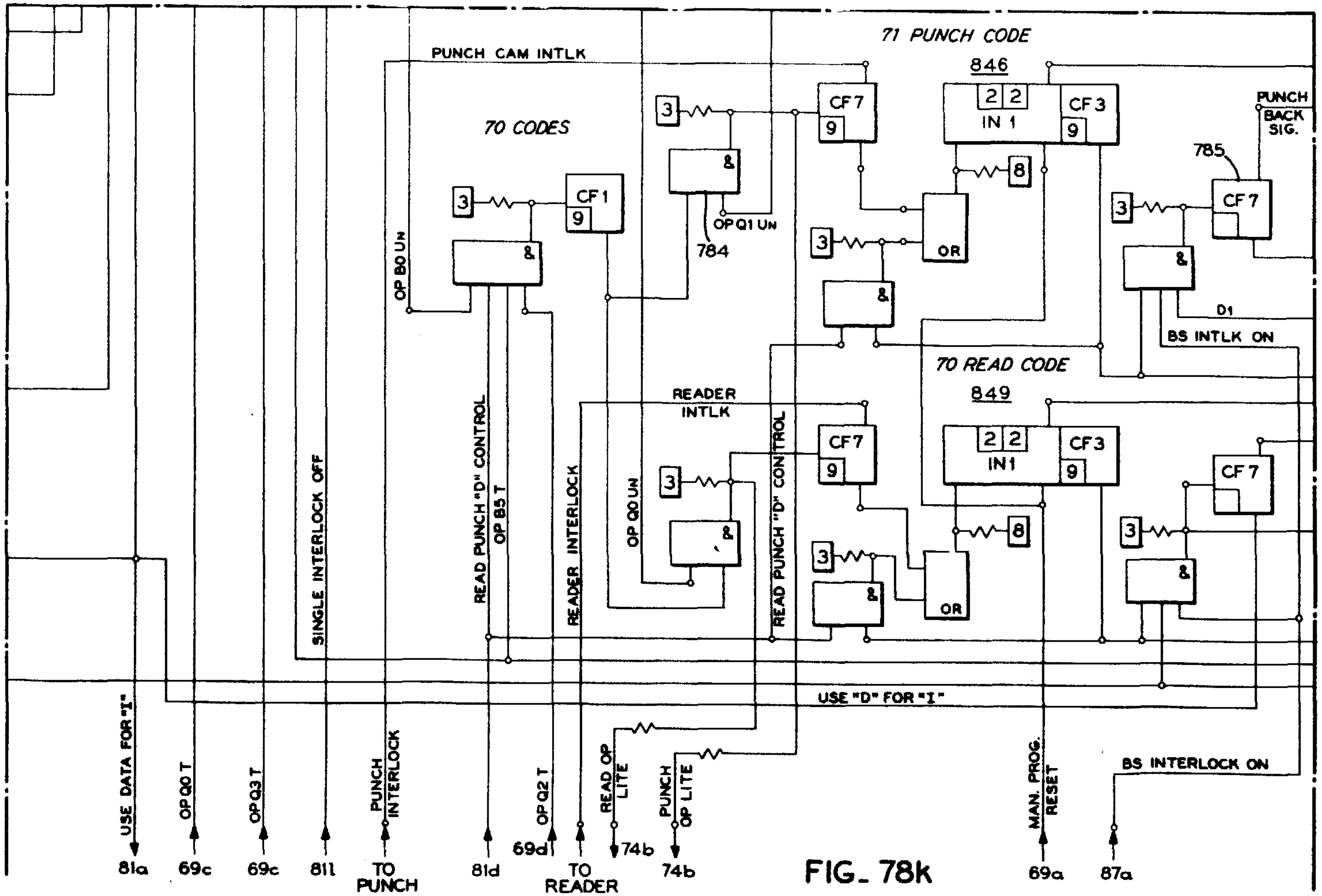

71 PUNCH CODE
846
PUNCH CAM INTLK
70 CODES
CF7
9
2 2
IN 1
CF3
9
PUNCH
BACK
SIG.
785
CF 1
9
3
&
8
OR
OP Q1 UN
784
OP B0 UN
D1
BS INTLK ON
70 READ CODE
849
READER
INTLK
CF7
CF 7
READ PUNCH "D" CONTROL
OP B5 T
READER INTERLOCK
OP Q0 UN
READ PUNCH "D" CONTROL
SINGLE INTERLOCK OFF
USE "D" FOR "I"
USE DATA FOR "I"
OP Q0 T
OP Q3 T
PUNCH
INTERLOCK
OP Q2 T
READ OP
LITE
PUNCH
OP LITE
MAN. PROG.
RESET
BS INTERLOCK ON
81a
69c
69c
81l
TO
PUNCH
81d
69d
TO
READER
74b
74b
69a
87a

FIG. 78k

Nov. 8, 1960

F. E. HAMILTON ET AL

2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

228 Sheets-Sheet 166

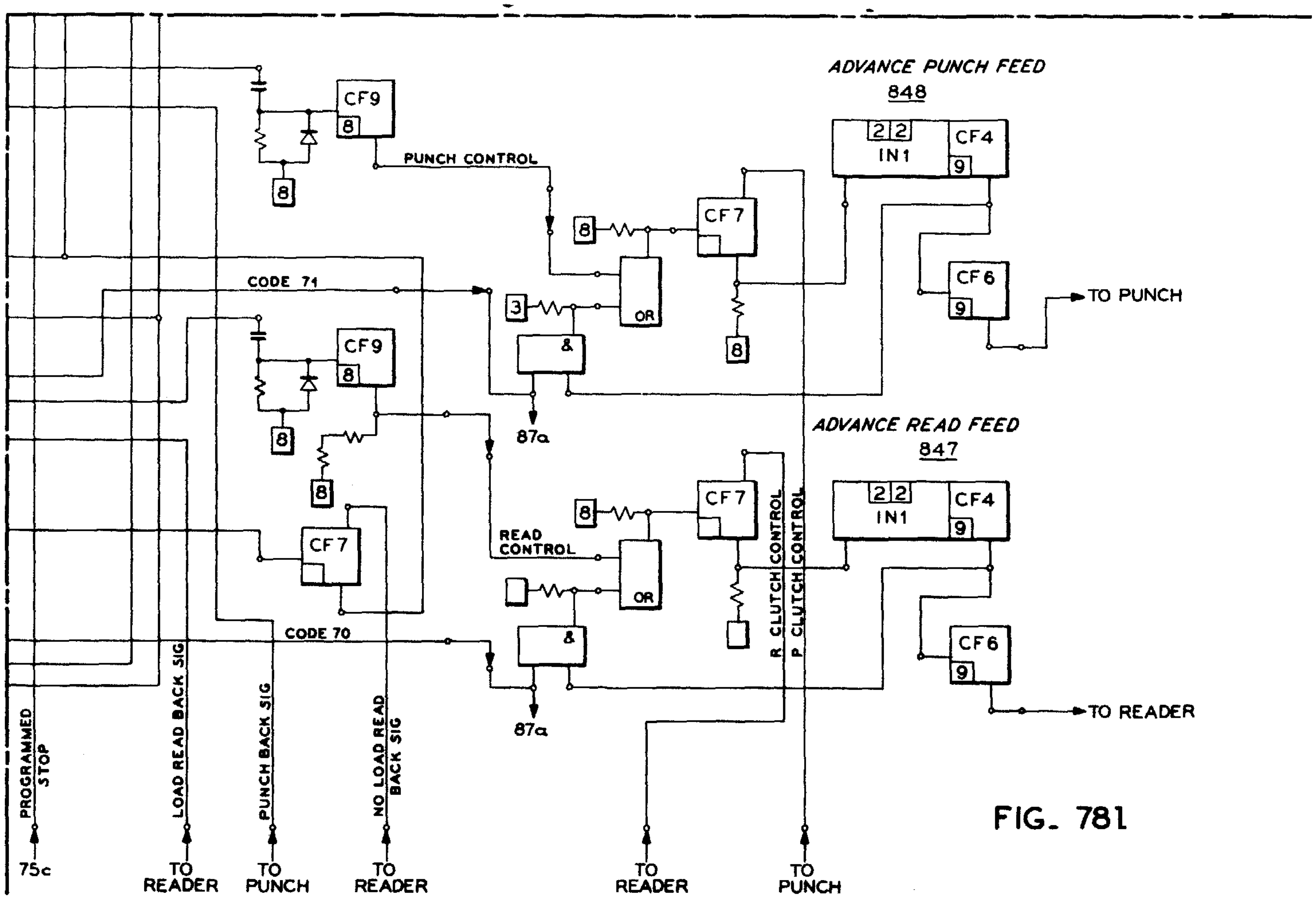
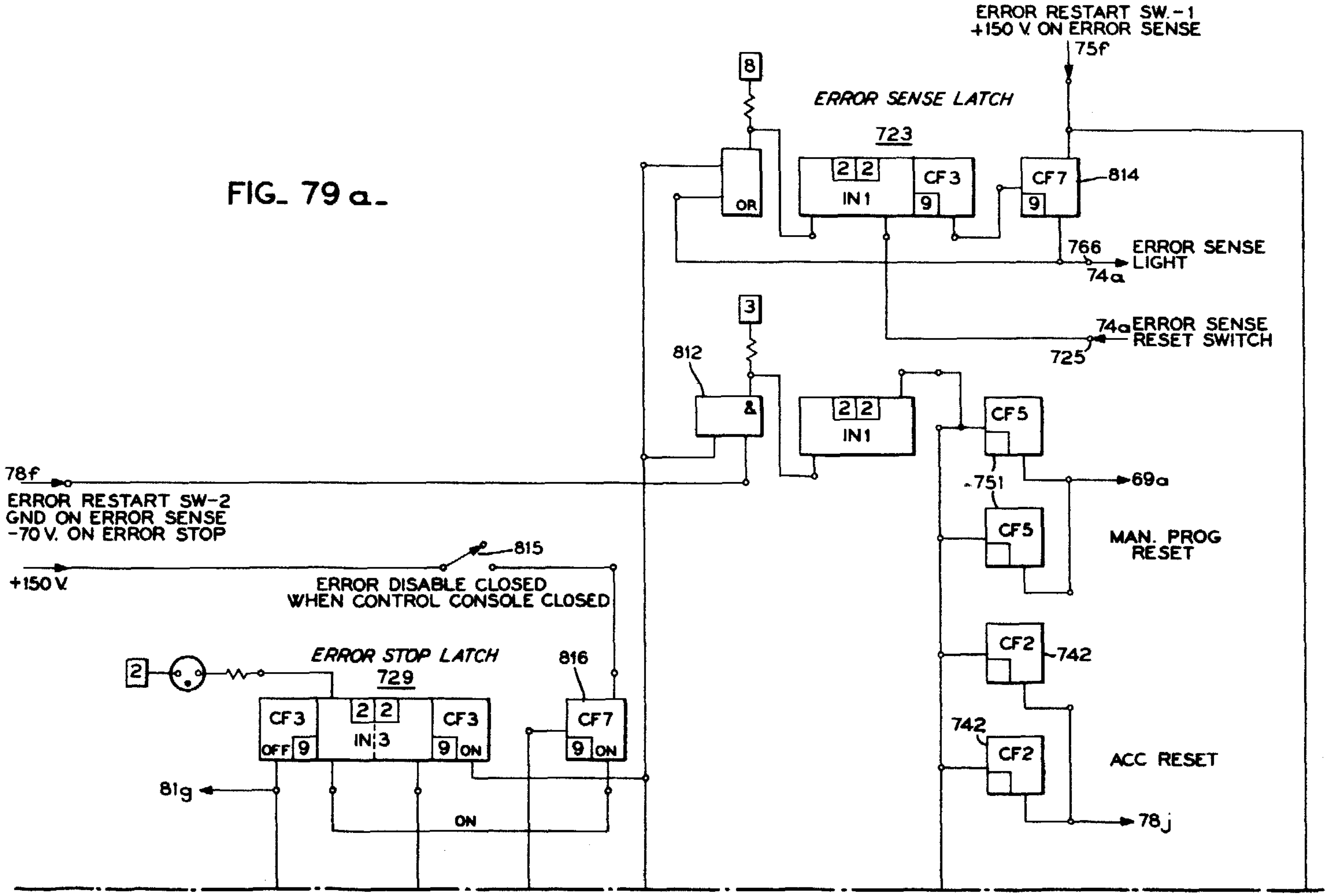


FIG. 781

FIG_ 79 a_


ERROR RESTART SW.-1
+150 V. ON ERROR SENSE
75f
8
ERROR SENSE LATCH
723
OR
2 2
IN 1
CF 3
9
CF 7
9
814
766
ERROR SENSE LIGHT
74a
3
74a ERROR SENSE RESET SWITCH
725
812
&
2 2
IN 1
CF 5
751
CF 5
69a
MAN. PROG RESET
CF 2
742
742
CF 2
ACC RESET
78j
78f
ERROR RESTART SW-2
GND ON ERROR SENSE
-70 V. ON ERROR STOP
815
+150 V.
ERROR DISABLE CLOSED
WHEN CONTROL CONSOLE CLOSED
ERROR STOP LATCH
729
816
2
CF 3
OFF 9
2 2
IN 3
CF 3
9 ON
CF 7
9 ON
81g
ON

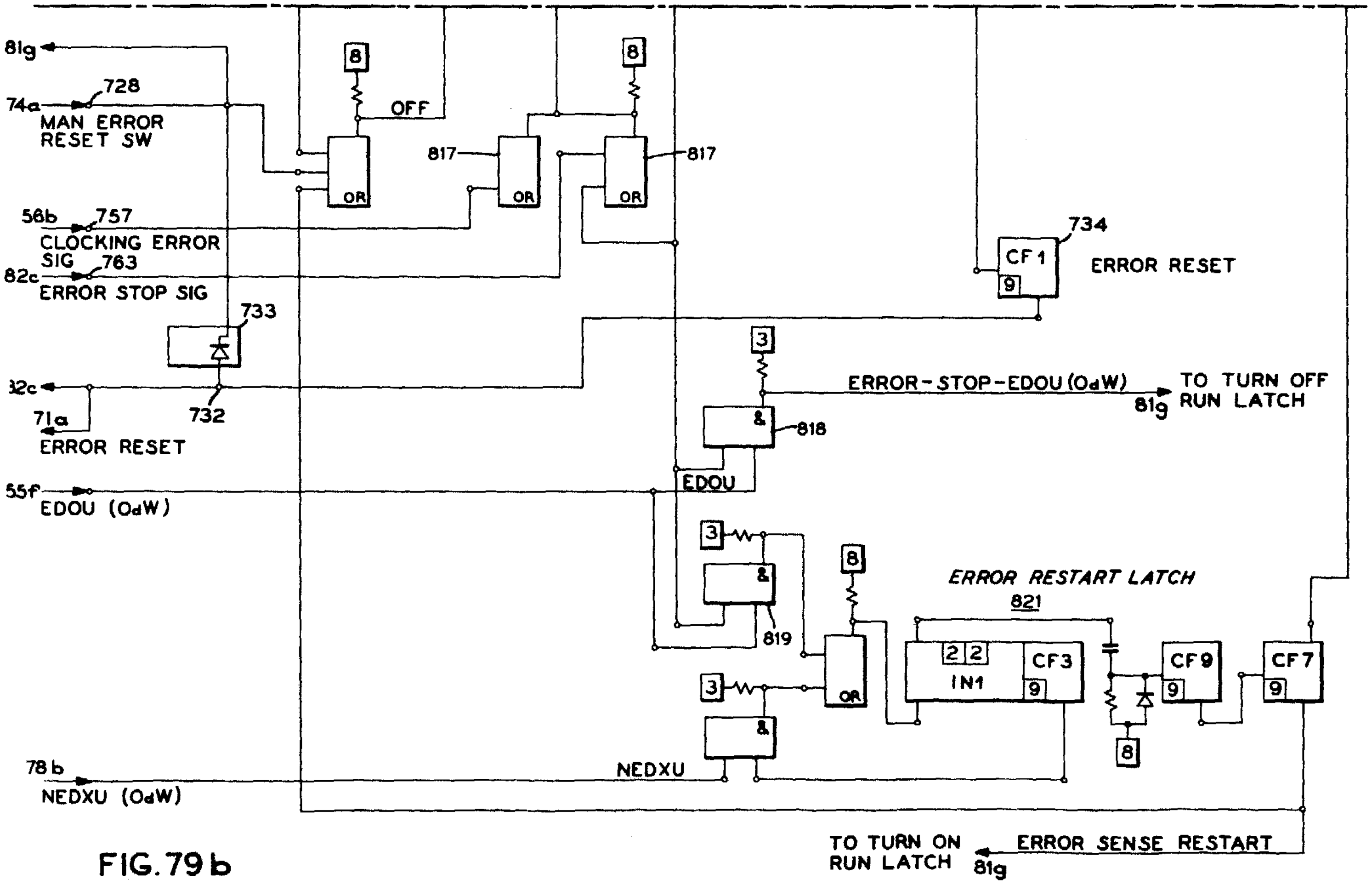

81g
728
74a
MAN ERROR
RESET SW
8
OFF
8
817
OR
OR
OR
817
58b
757
CLOCKING ERROR
SIG
763
82c
ERROR STOP SIG
734
CF1
9
ERROR RESET
733
3
32c
71a
732
ERROR RESET
ERROR-STOP-EDOU (OdW)
TO TURN OFF
81g
RUN LATCH
&
818
EDOU
55f
EDOU (OdW)
3
&
8
819
ERROR RESTART LATCH
821
2 2
IN1
CF3
9
CF9
9
CF7
9
3
OR
&
8
78b
NEDXU
NEDXU (OdW)
TO TURN ON
RUN LATCH
81g
ERROR SENSE RESTART

FIG. 79b

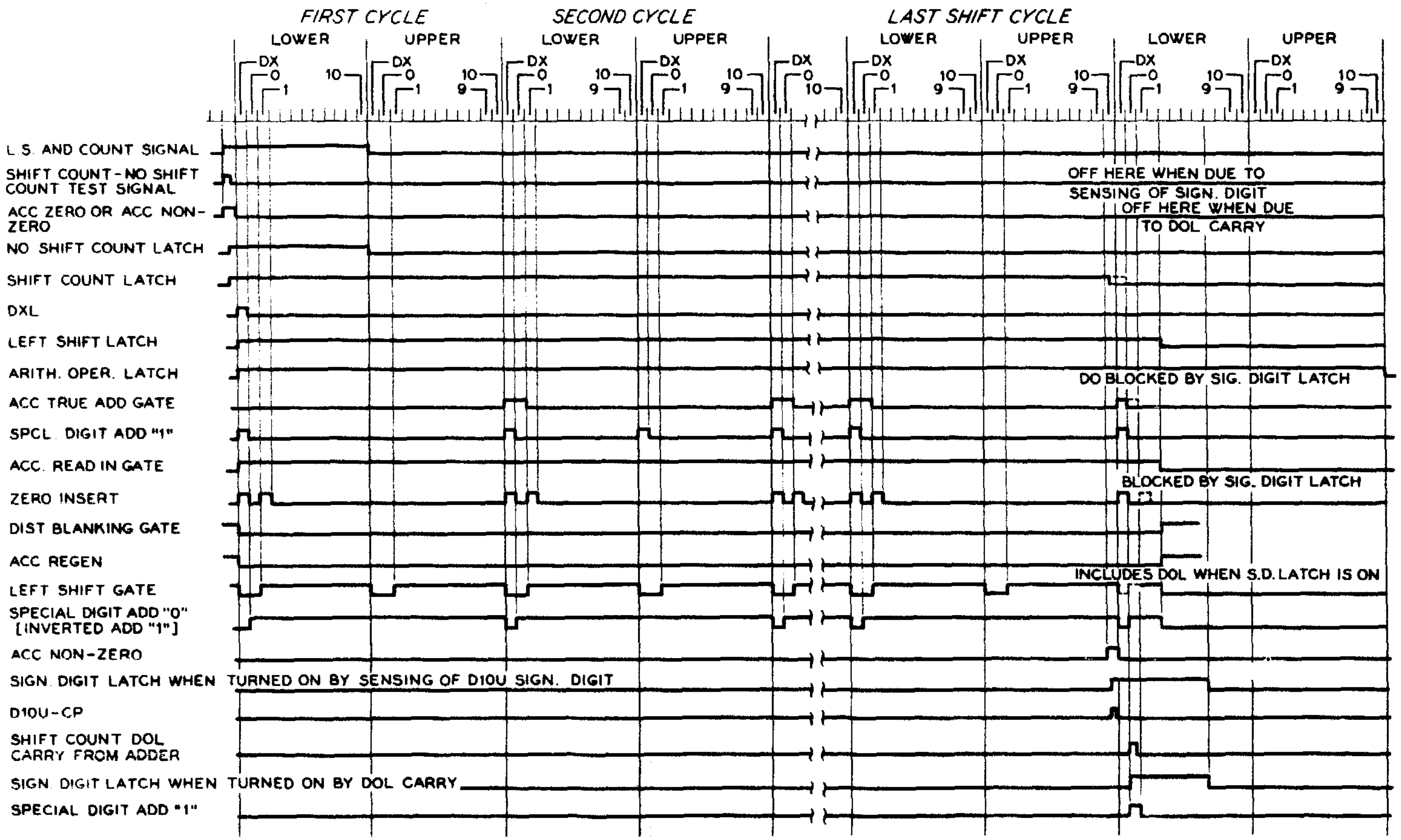

FIRST CYCLE
SECOND CYCLE
LAST SHIFT CYCLE
LOWER
UPPER
DX
0
1
10
9
L.S. AND COUNT SIGNAL
SHIFT COUNT-NO SHIFT COUNT TEST SIGNAL
ACC ZERO OR ACC NON-ZERO
NO SHIFT COUNT LATCH
SHIFT COUNT LATCH
DXL
LEFT SHIFT LATCH
ARITH. OPER. LATCH
ACC TRUE ADD GATE
SPCL. DIGIT ADD "1"
ACC. READ IN GATE
ZERO INSERT
DIST BLANKING GATE
ACC REGEN
LEFT SHIFT GATE
SPECIAL DIGIT ADD "0" [INVERTED ADD "1"]
ACC NON-ZERO
SIGN. DIGIT LATCH WHEN TURNED ON BY SENSING OF D10U SIGN. DIGIT
D10U-CP
SHIFT COUNT DOL CARRY FROM ADDER
SIGN. DIGIT LATCH WHEN TURNED ON BY DOL CARRY
SPECIAL DIGIT ADD "1"
OFF HERE WHEN DUE TO SENSING OF SIGN. DIGIT
OFF HERE WHEN DUE TO DOL CARRY
DO BLOCKED BY SIG. DIGIT LATCH
BLOCKED BY SIG. DIGIT LATCH
INCLUDES DOL WHEN S.D. LATCH IS ON

FIG_80

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 170

FIG. 81a

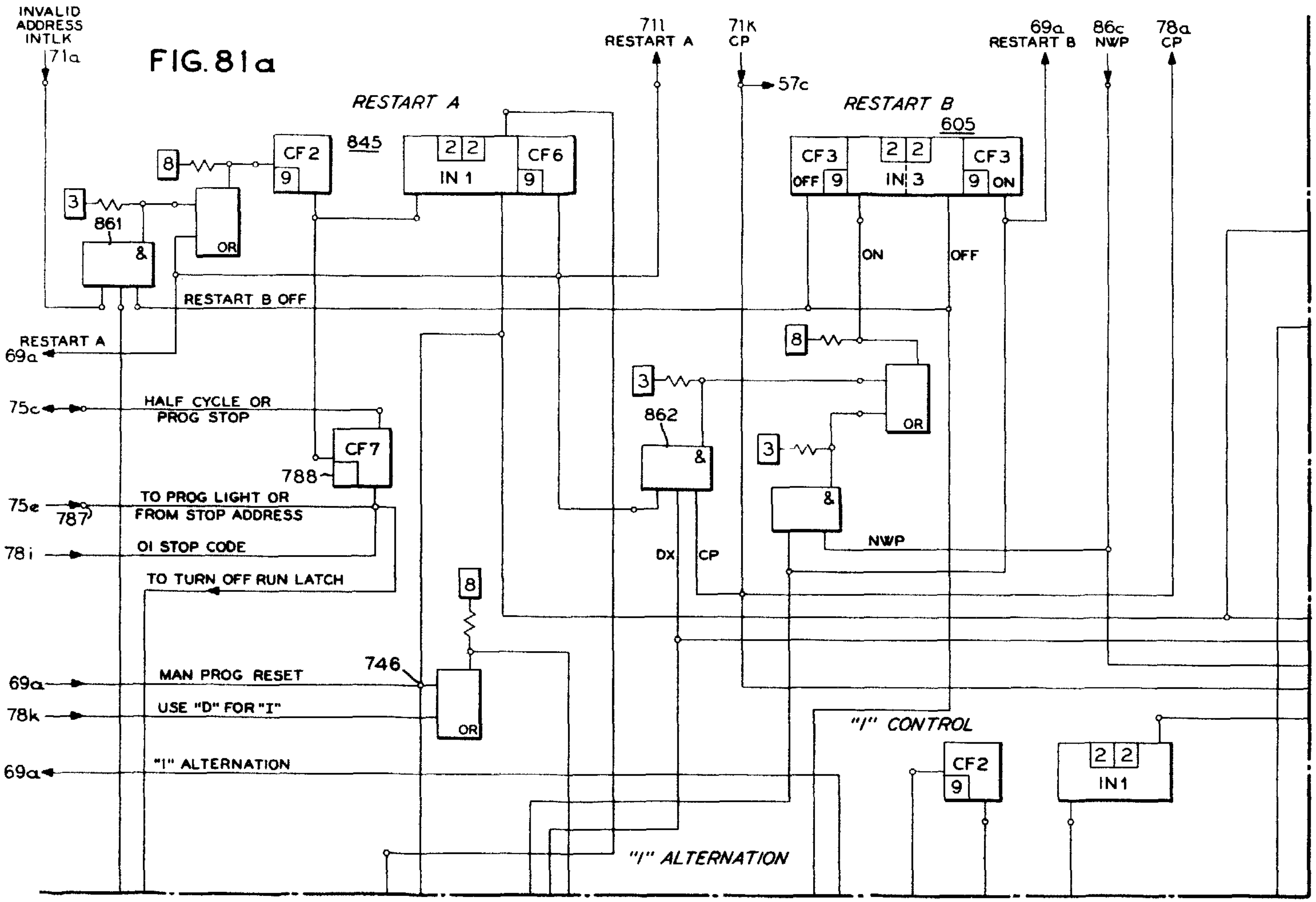


FIG. 81 b

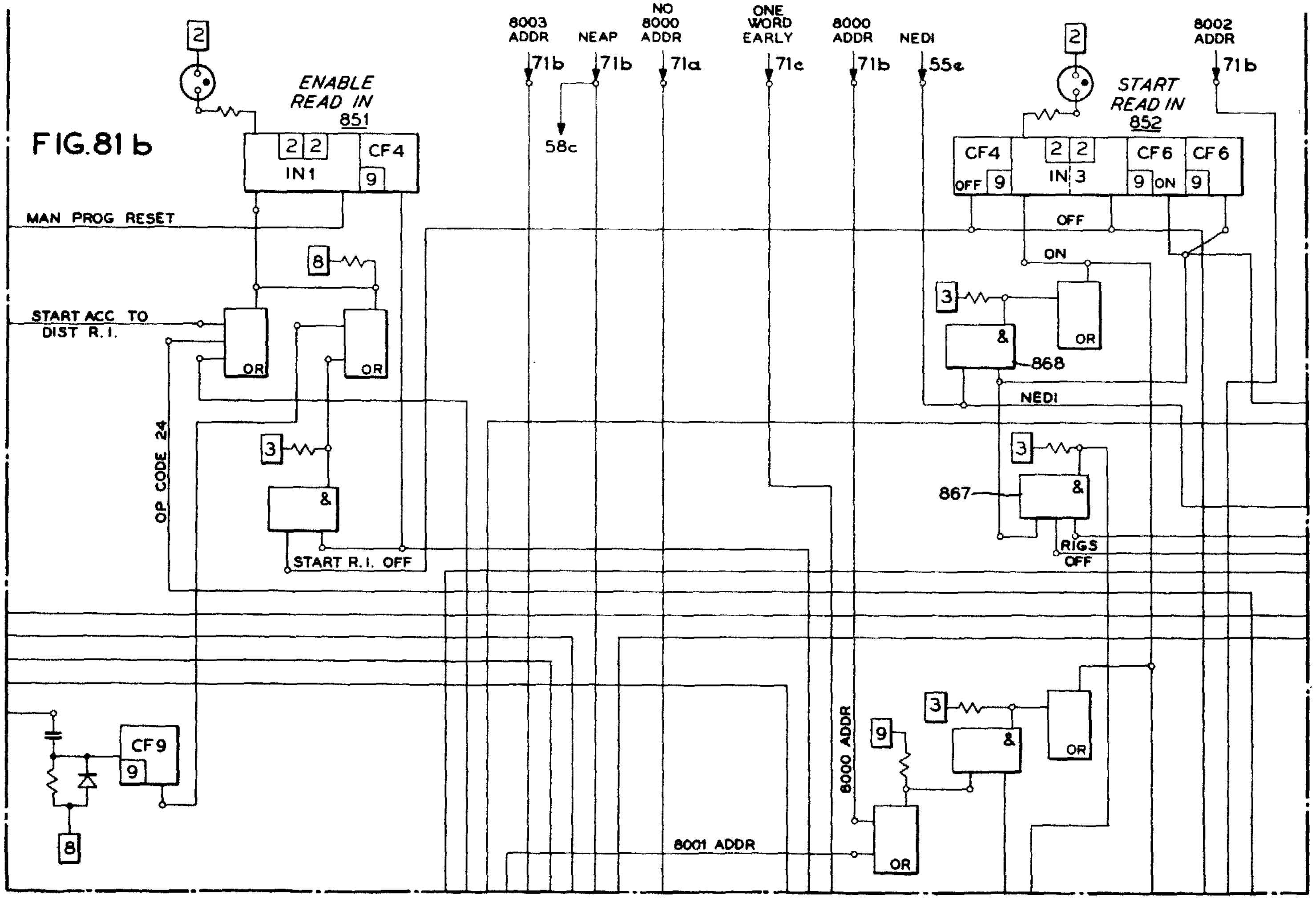
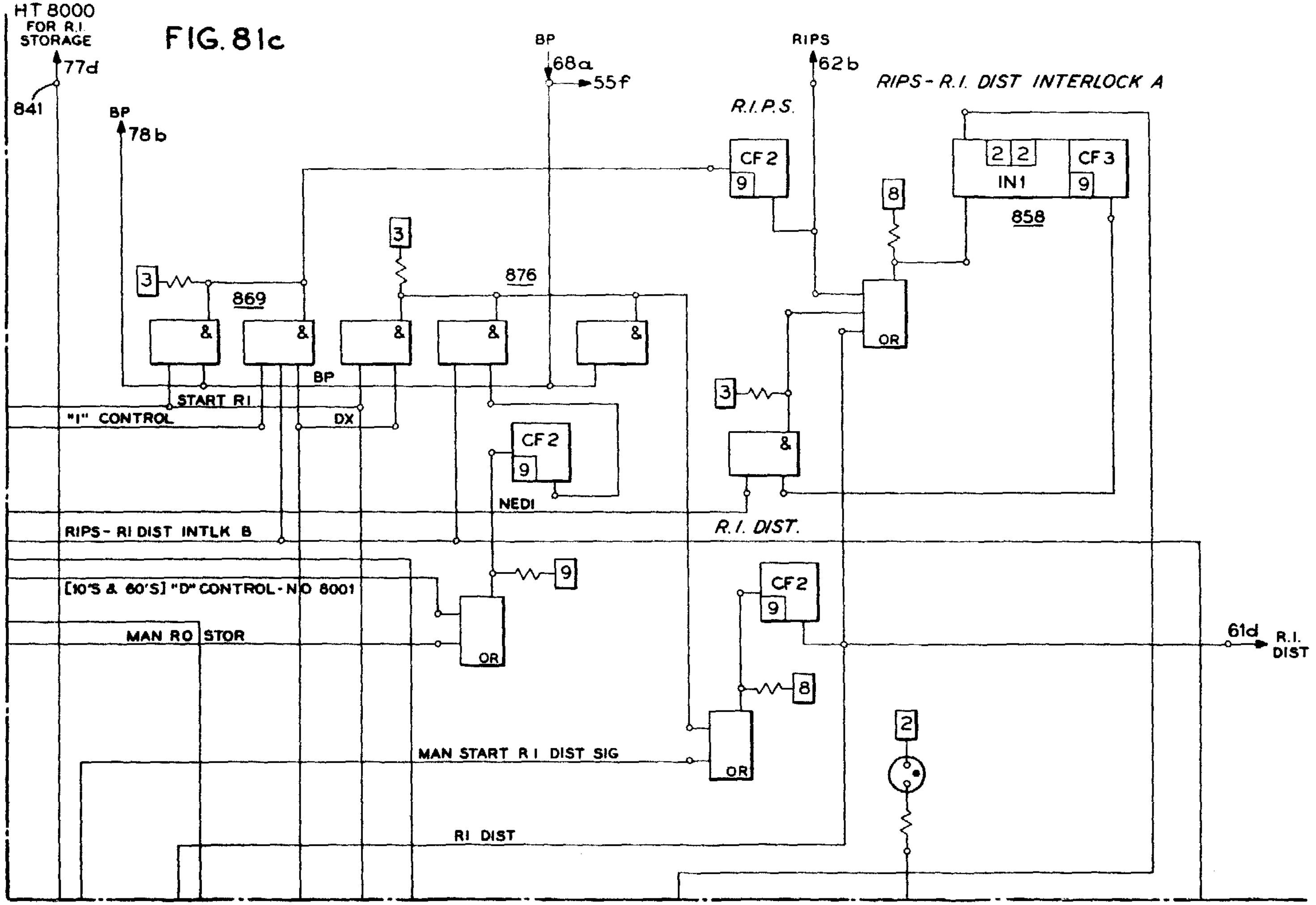

8003 ADDR
71b
NEAP
71b
58c
NO 8000 ADDR
71a
ONE WORD EARLY
71c
8000 ADDR
71b
NEDI
55e
8002 ADDR
71b
2
ENABLE READ IN
851
2 2
IN1
CF4
9
MAN PROG RESET
8
START ACC TO DIST R.I.
OR
OR
OP CODE 24
3
&
START R.I. OFF
CF9
9
8
2
START READ IN
852
CF4
OFF 9
2 2
IN 3
CF6
9 ON
CF6
9
OFF
ON
3
&
868
OR
NEDI
3
867
&
RIGS OFF
3
&
OR
9
8000 ADDR
OR
8001 ADDR

FIG. 81c


HT 8000
FOR R.I.
STORAGE
77d
841
BP
78b
BP
68a
55f
RIPS
62b
RIPS - R.I. DIST INTERLOCK A
R.I.P.S.
CF 2
9
2 2
IN1
CF 3
9
858
8
3
3
869
876
&
&
&
&
&
BP
START RI
"I" CONTROL
DX
CF 2
9
NEDI
OR
3
&
R.I. DIST.
RIPS - RI DIST INTLK B
9
[10'S & 60'S] "D" CONTROL - N/O 8001
OR
CF 2
9
MAN RO STOR
61d
R.I.
DIST
8
OR
MAN START R I DIST SIG
2
RI DIST

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 173

FIG. 81d

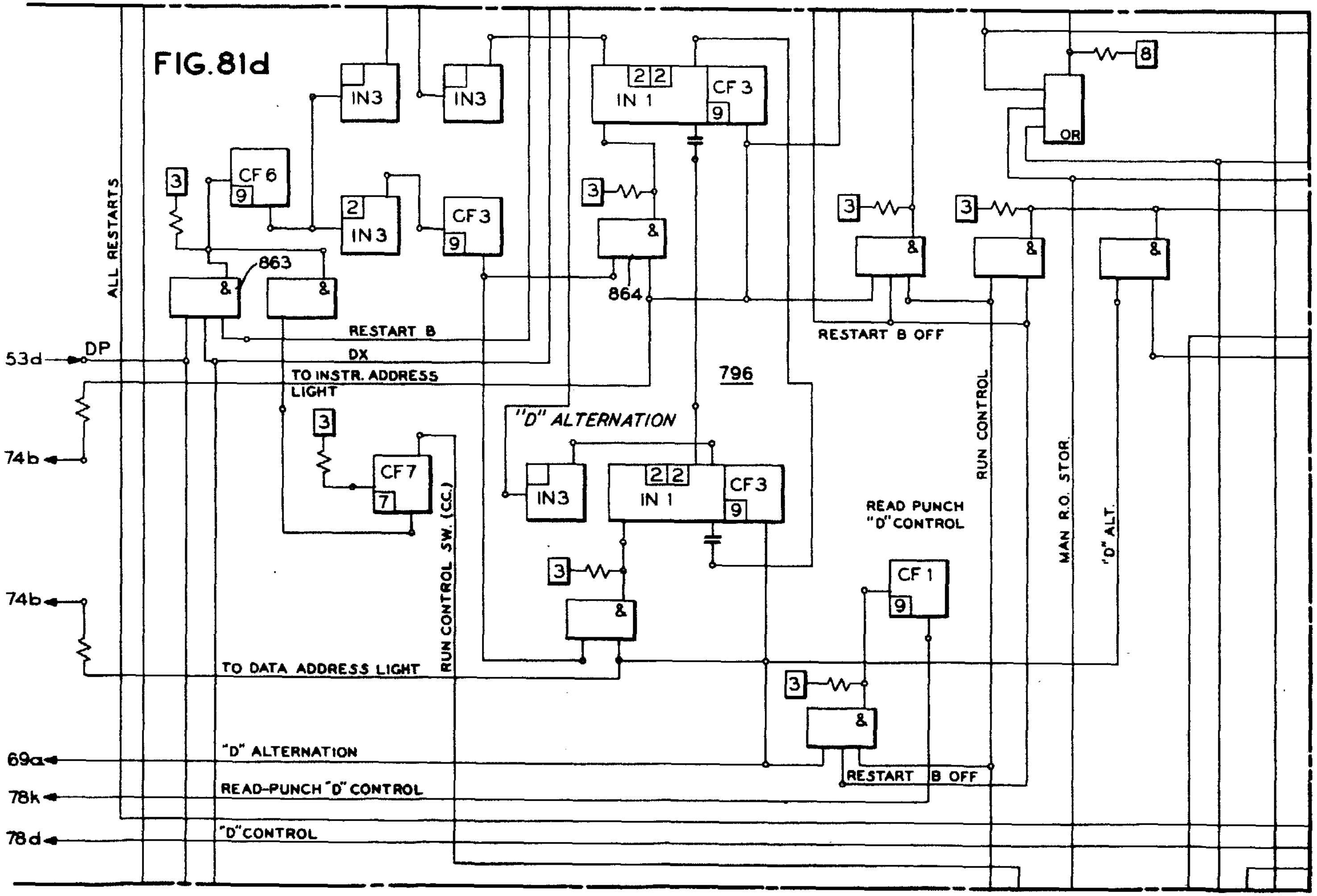


FIG. 81e

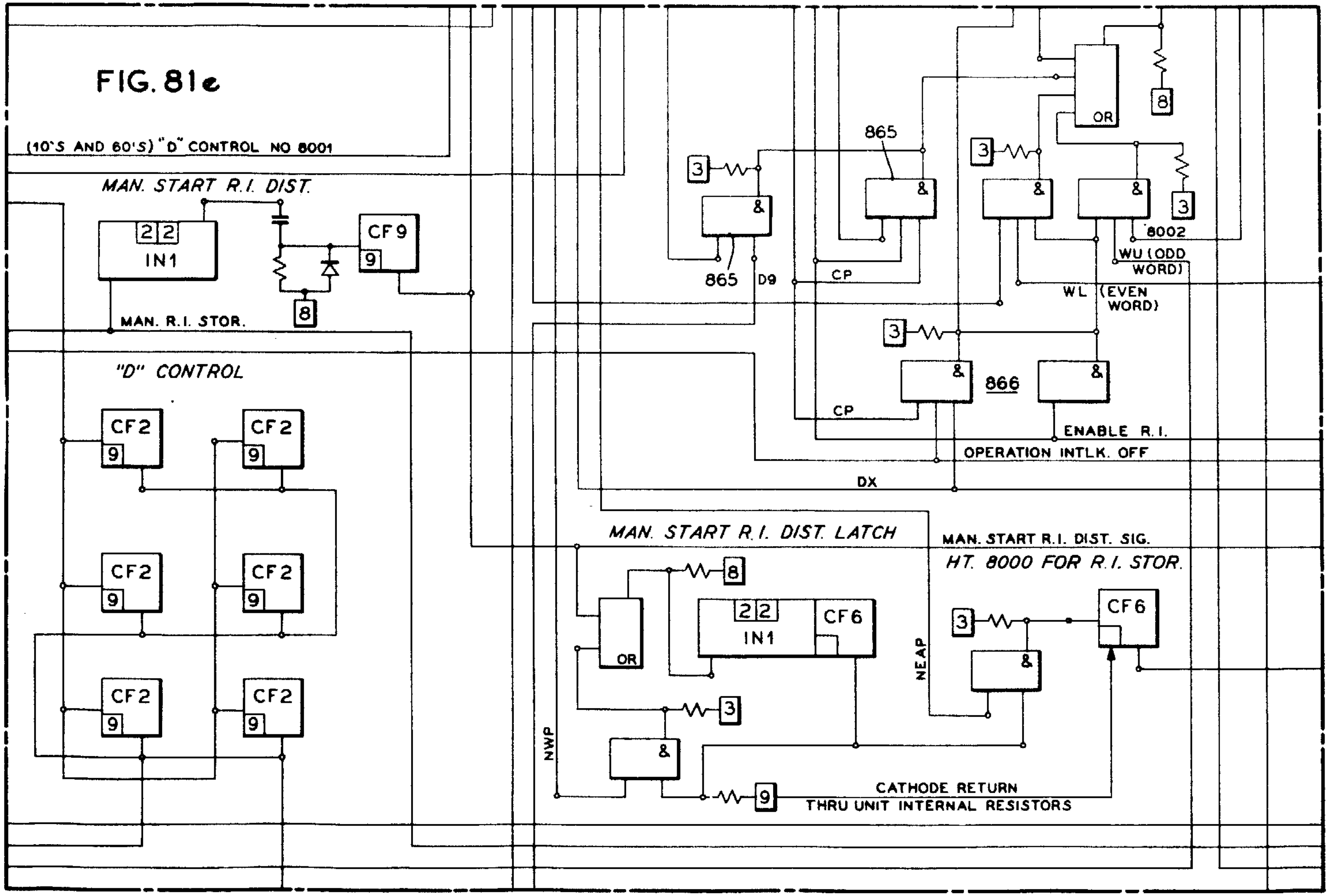
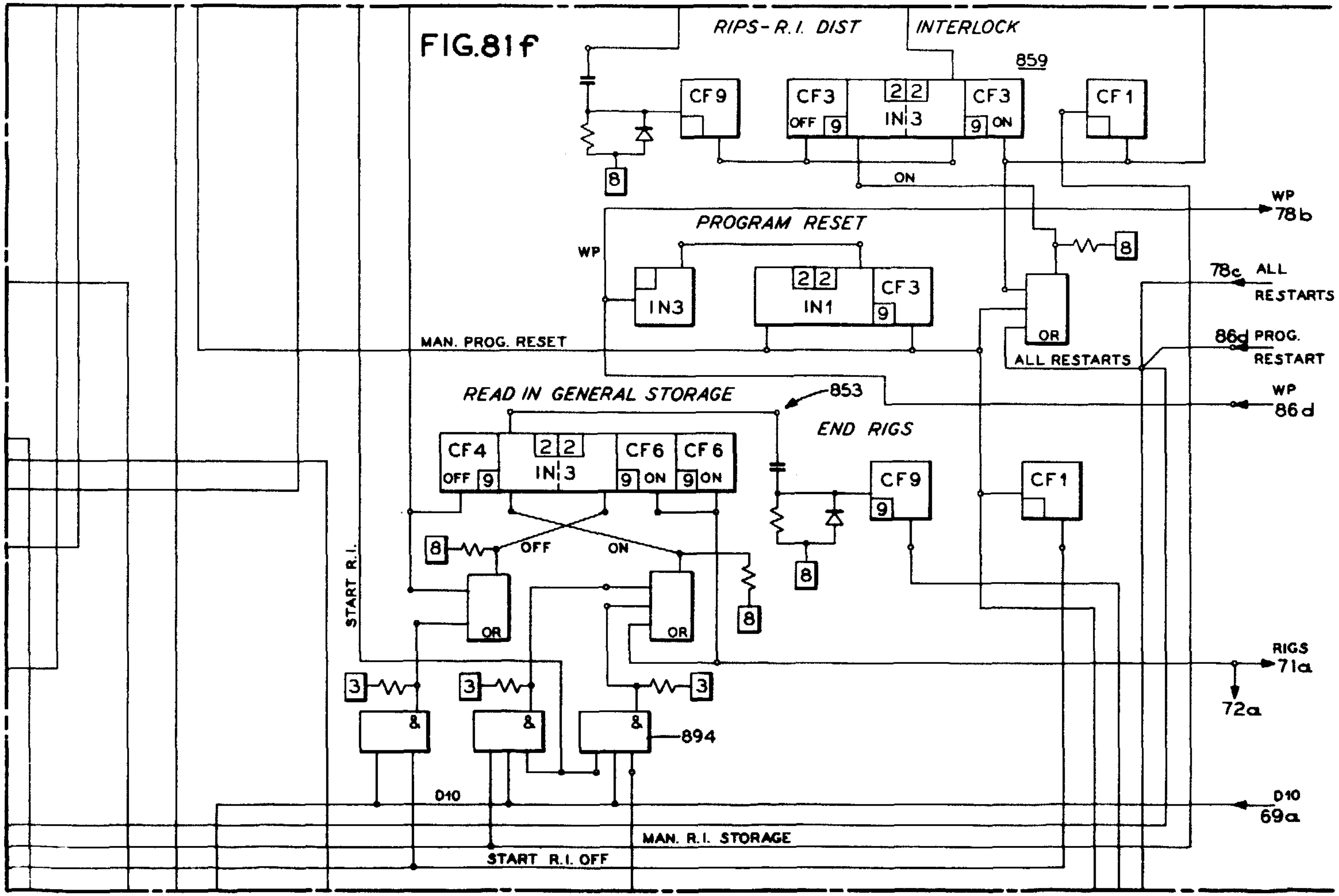

(10'S AND 60'S) "D" CONTROL NO 8001
MAN. START R.I. DIST.
2 2
IN1
CF9
9
8
MAN. R.I. STOR.
"D" CONTROL
CF2
9
CF2
9
CF2
9
CF2
9
CF2
9
CF2
9
3
&
865
D9
865
&
CP
3
OR
8
&
3
&
3
8002
WU (ODD WORD)
WL (EVEN WORD)
3
&
866
&
CP
ENABLE R.I.
OPERATION INTLK. OFF
DX
MAN. START R.I. DIST. LATCH
MAN. START R.I. DIST. SIG.
HT. 8000 FOR R.I. STOR.
8
OR
2 2
IN1
CF6
3
CF6
NEAP
&
3
&
NWP
9
CATHODE RETURN
THRU UNIT INTERNAL RESISTORS

FIG.81f


RIPS-R.I. DIST
INTERLOCK
859
CF9
CF3
OFF
9
2 2
IN 3
CF3
9
ON
CF1
8
ON
WP
78b
PROGRAM RESET
WP
IN3
2 2
IN1
CF3
9
8
OR
78c
ALL
RESTARTS
86d
PROG.
RESTART
MAN. PROG. RESET
ALL RESTARTS
WP
86d
READ IN GENERAL STORAGE
853
END RIGS
CF4
OFF
9
2 2
IN 3
CF6
9
ON
CF6
9
ON
CF9
9
CF1
8
OFF
ON
8
START R.I.
OR
OR
8
RIGS
71a
72a
3
3
3
&
&
&
894
D10
D10
69a
MAN. R.I. STORAGE
START R.I. OFF

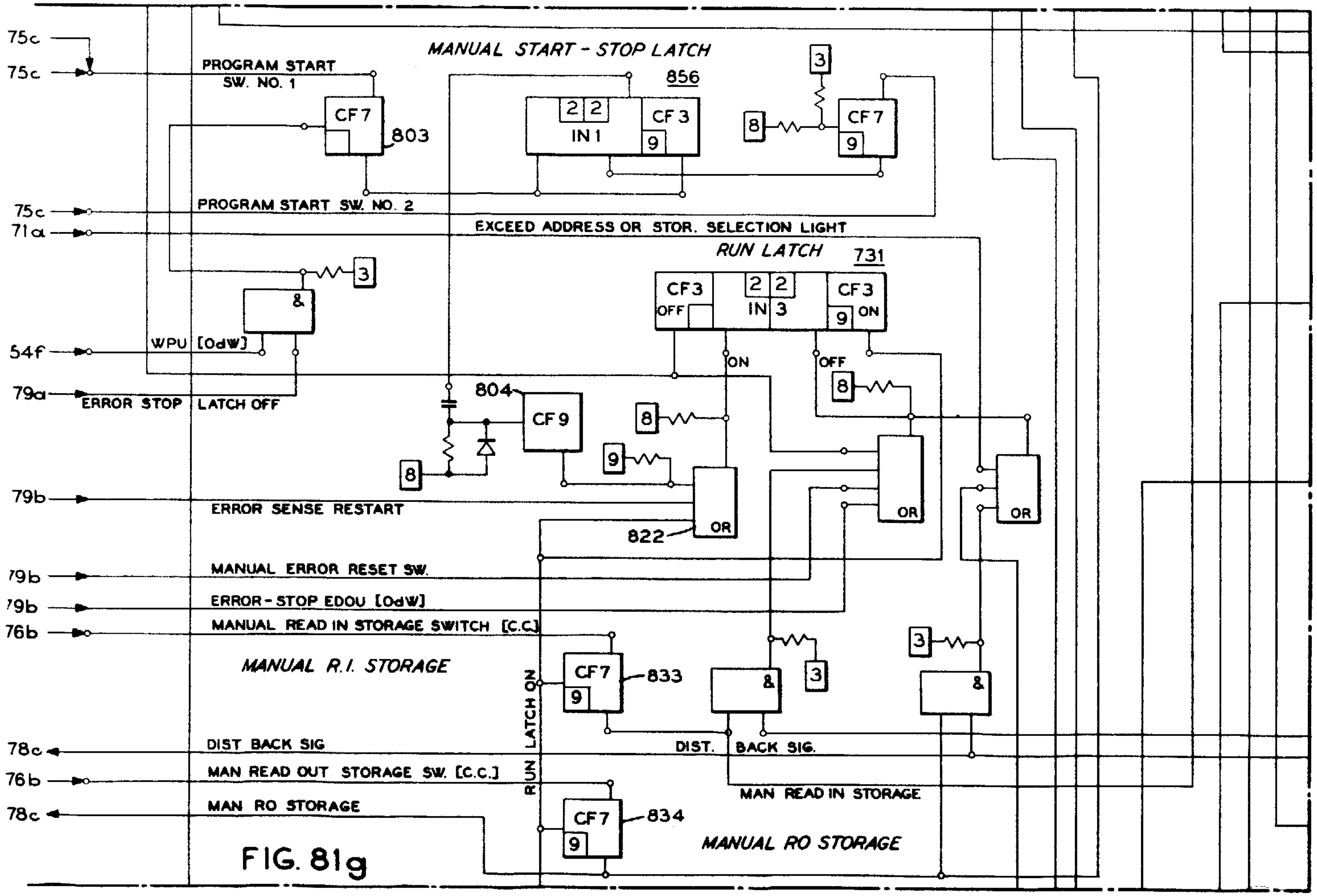

75c
75c
PROGRAM START SW. NO. 1
MANUAL START - STOP LATCH
856
3
CF7
803
2 2
IN 1
CF 3
9
8
CF7
9
75c
PROGRAM START SW. NO. 2
71a
EXCEED ADDRESS OR STOR. SELECTION LIGHT
RUN LATCH
731
3
&
CF3
OFF
2 2
IN 3
CF3
9
ON
54f
WPU [OdW]
ON
OFF
8
79a
ERROR STOP
LATCH OFF
804
CF9
8
8
9
OR
OR
OR
79b
ERROR SENSE RESTART
822
79b
MANUAL ERROR RESET SW.
79b
ERROR - STOP EDOU [OdW]
76b
MANUAL READ IN STORAGE SWITCH [C.C.]
MANUAL R.I. STORAGE
RUN LATCH ON
CF7
9
833
&
3
3
&
78c
DIST BACK SIG
DIST.
BACK SIG.
76b
MAN READ OUT STORAGE SW. [C.C.]
MAN READ IN STORAGE
78c
MAN RO STORAGE
CF7
9
834
MANUAL RO STORAGE

FIG. 81g

Nov. 8, 1960

F. E. HAMILTON ET AL

2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

228 Sheets-Sheet 177

FIG. 81 h

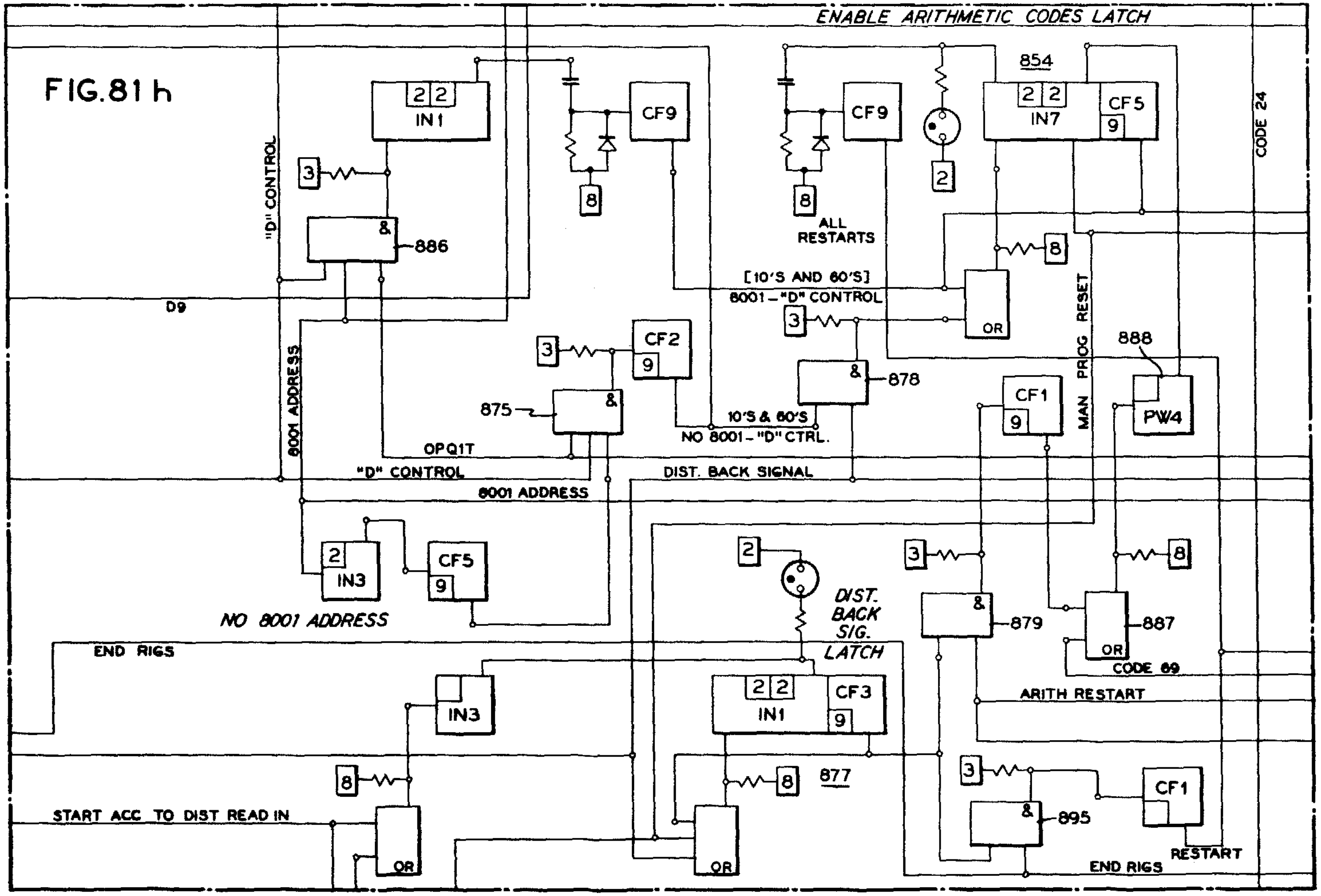
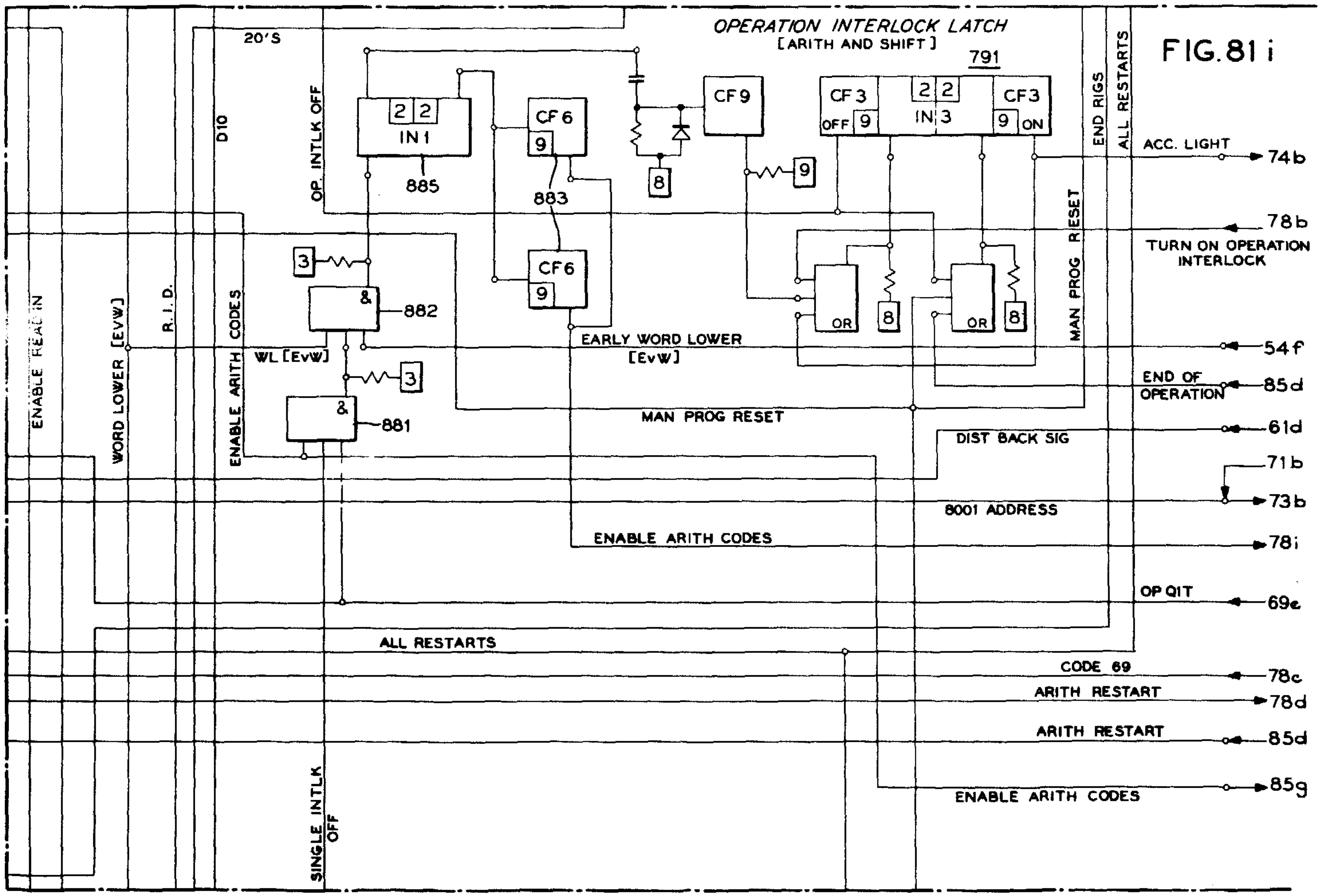


FIG. 81 i


OPERATION INTERLOCK LATCH
[ARITH AND SHIFT]
791
20'S
D10
OP. INTLK OFF
ENABLE READ IN
WORD LOWER [EvW]
R.I.D.
ENABLE ARITH CODES
2 2
IN 1
885
CF6
9
883
CF6
9
CF9
9
8
CF3
OFF 9
2 2
IN 3
CF3
9 ON
OR
OR
8
8
3
&
882
WL [EvW]
3
&
881
EARLY WORD LOWER
[EvW]
MAN PROG RESET
END RIGS
ALL RESTARTS
MAN PROG RESET
ACC. LIGHT
74b
78b
TURN ON OPERATION
INTERLOCK
54f
END OF
OPERATION
85d
DIST BACK SIG
61d
71b
73b
8001 ADDRESS
ENABLE ARITH CODES
78i
OP QIT
69e
ALL RESTARTS
CODE 69
78c
ARITH RESTART
78d
ARITH RESTART
85d
ENABLE ARITH CODES
85g
SINGLE INTLK
OFF

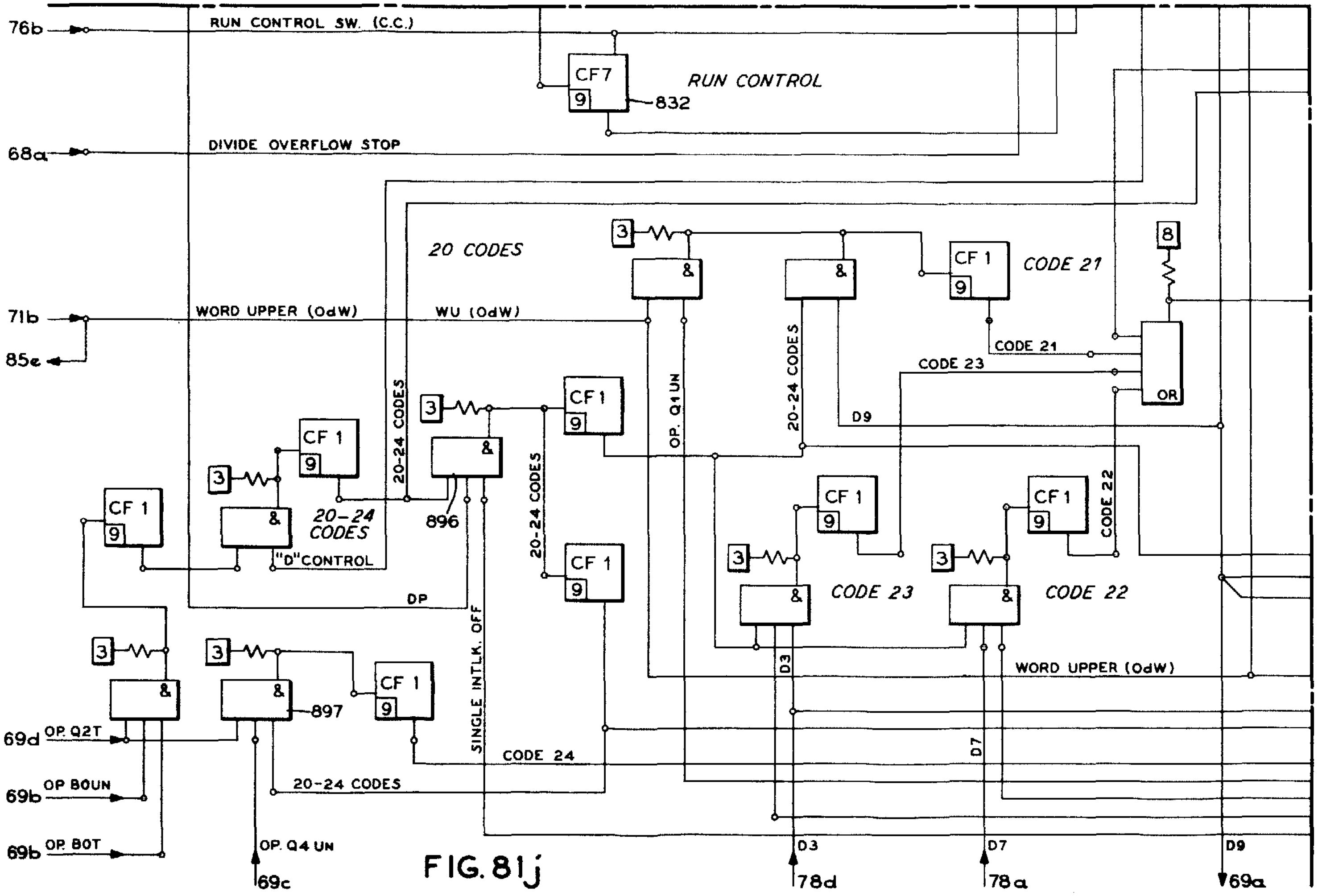

76b
RUN CONTROL SW. (C.C.)
CF7
9
832
RUN CONTROL
68a
DIVIDE OVERFLOW STOP
20 CODES
3
&
CF 1
9
CODE 21
8
71b
WORD UPPER (OdW)
WU (OdW)
85e
CODE 21
CODE 23
OR
20-24 CODES
OP. Q1 UN
20-24 CODES
D9
3
&
CF 1
9
CF 1
9
3
&
20-24 CODES
896
"D" CONTROL
CF 1
9
20-24 CODES
CF 1
9
CF 1
9
CODE 22
CODE 23
CODE 22
DP
SINGLE INTLK. OFF
3
&
3
&
897
CF 1
9
D3
WORD UPPER (OdW)
69d
OP. Q2T
69b
OP BOUN
69b
OP. BOT
CODE 24
20-24 CODES
D7
OP. Q4 UN
69c
D3
78d
D7
78a
D9
69a

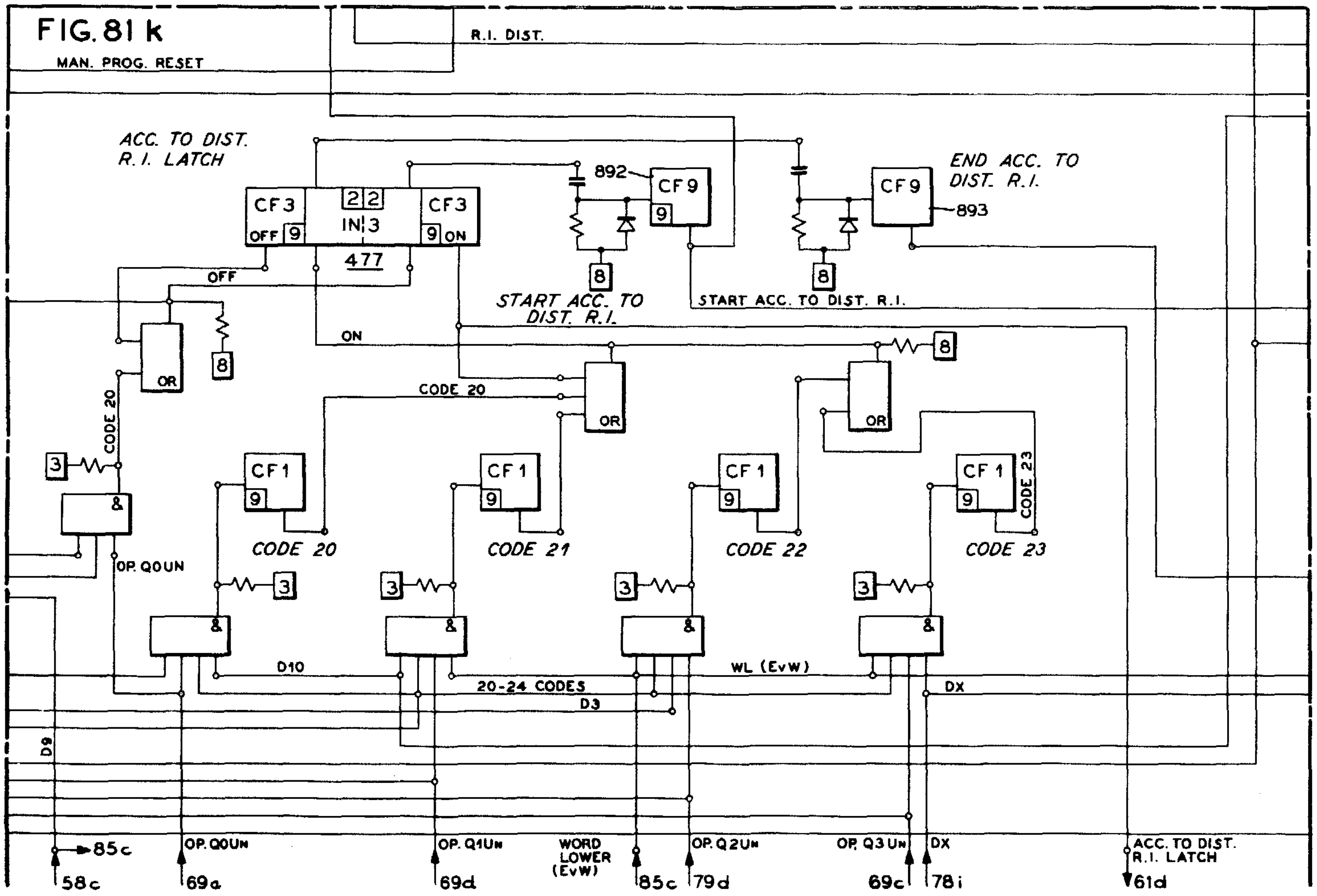
FIG. 81j

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 180

FIG. 81 k

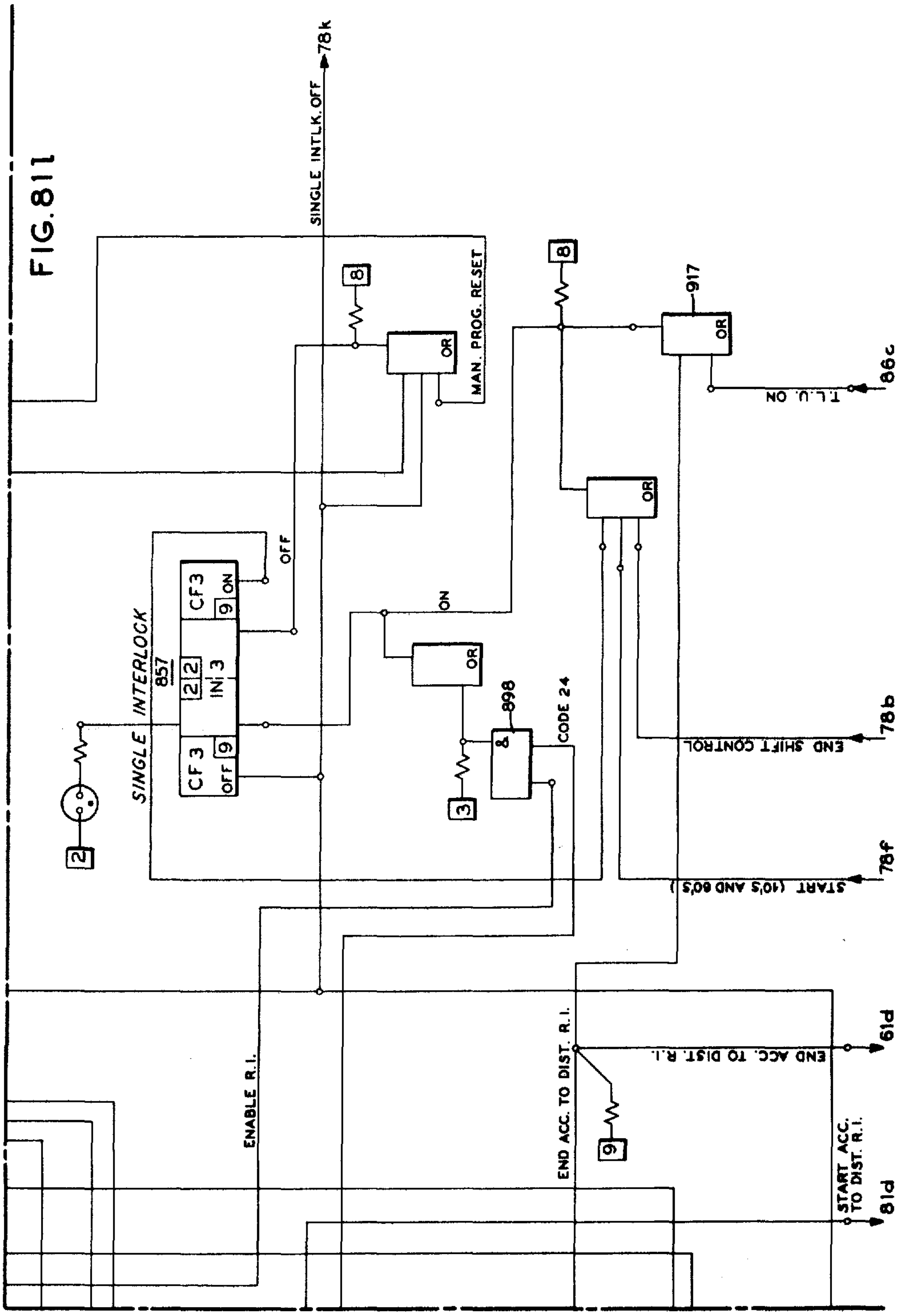


Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 181

FIG. 811



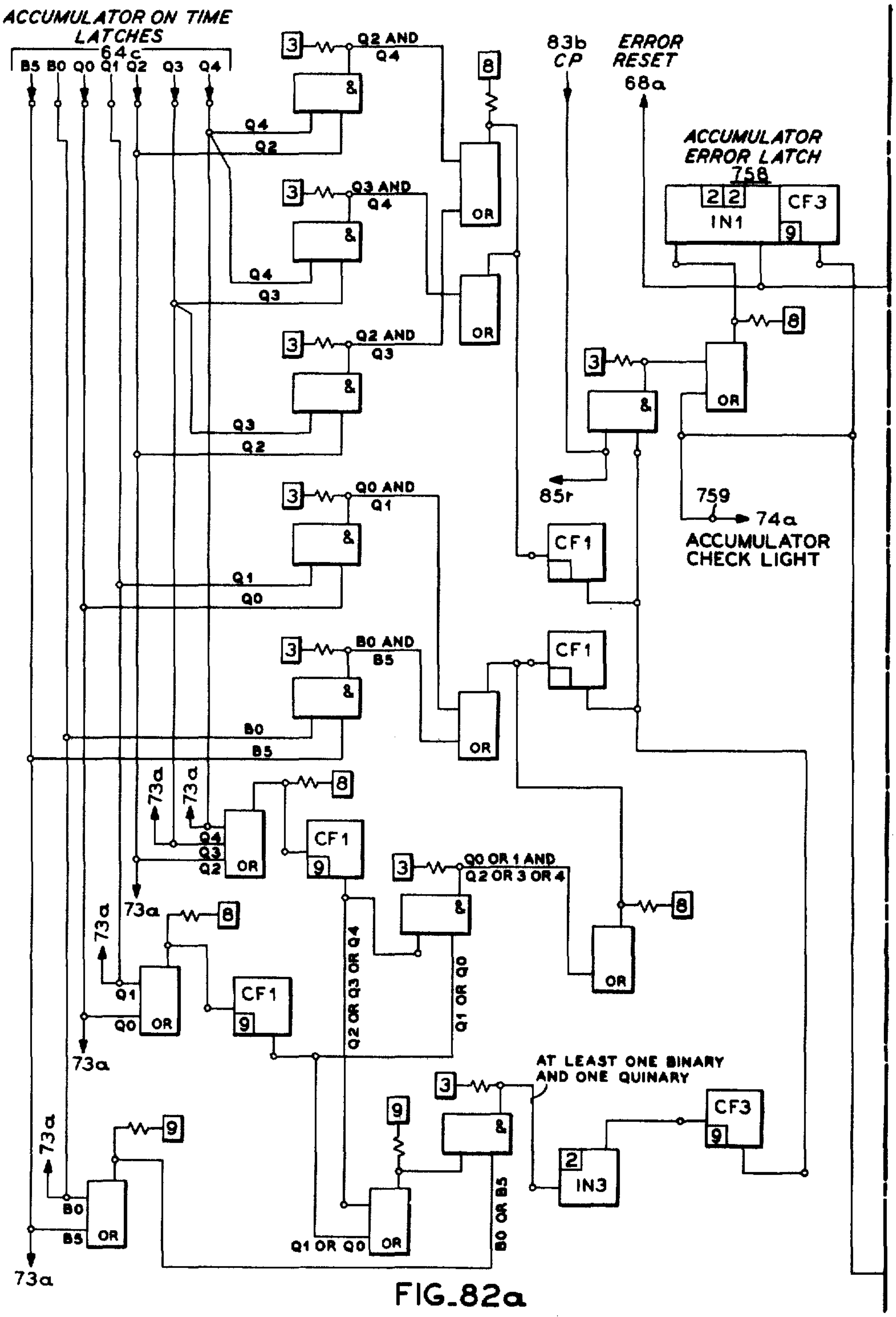

ACCUMULATOR ON TIME LATCHES
64c
B5 B0 Q0 Q1 Q2 Q3 Q4
Q2 AND Q4
Q3 AND Q4
Q2 AND Q3
Q0 AND Q1
B0 AND B5
83b CP
ERROR RESET
68a
ACCUMULATOR ERROR LATCH 758
IN1
CF3
85r
759
74a
ACCUMULATOR CHECK LIGHT
CF1
73a
Q0 OR 1 AND Q2 OR 3 OR 4
Q2 OR Q3 OR Q4
Q1 OR Q0
AT LEAST ONE BINARY AND ONE QUINARY
IN3
B0 OR B5
OR
&

FIG_82a

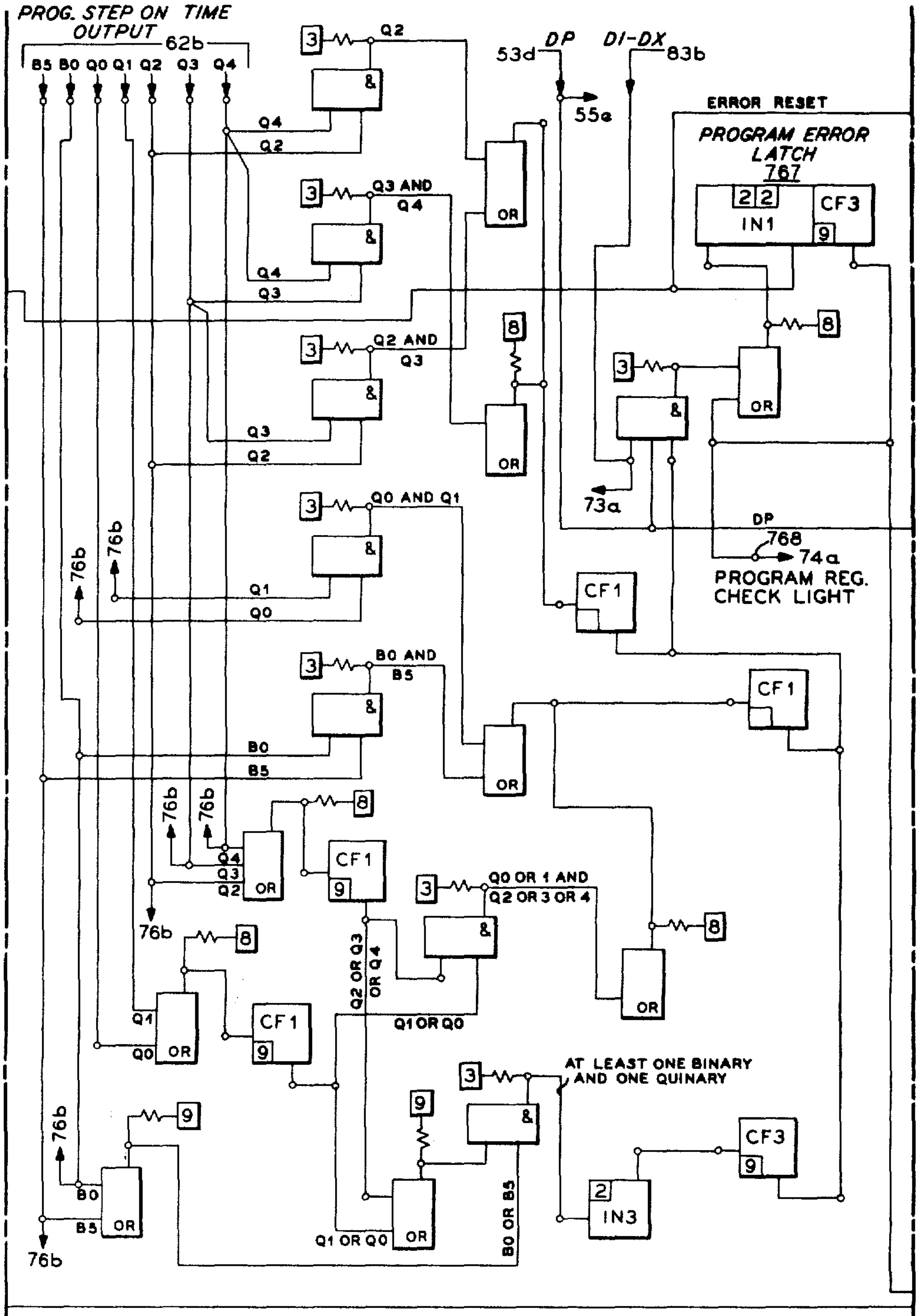

PROG. STEP ON TIME OUTPUT
62b
B5 B0 Q0 Q1 Q2 Q3 Q4
3
Q2
&
Q4
Q2
53d
DP
DI-DX
83b
55a
ERROR RESET
PROGRAM ERROR LATCH
767
2 2
IN1
CF3
9
OR
Q3 AND Q4
Q4
Q3
8
Q2 AND Q3
Q3
Q2
OR
73a
DP
768
74a
PROGRAM REG. CHECK LIGHT
Q0 AND Q1
Q1
Q0
76b
CF1
B0 AND B5
B0
B5
Q4
Q3
Q2
Q0 OR 1 AND Q2 OR 3 OR 4
Q2 OR Q3 OR Q4
Q1 OR Q0
Q1
Q0
AT LEAST ONE BINARY AND ONE QUINARY
9
IN3
B0 OR B5
Q1 OR Q0
B0
B5

FIG.82b

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 184

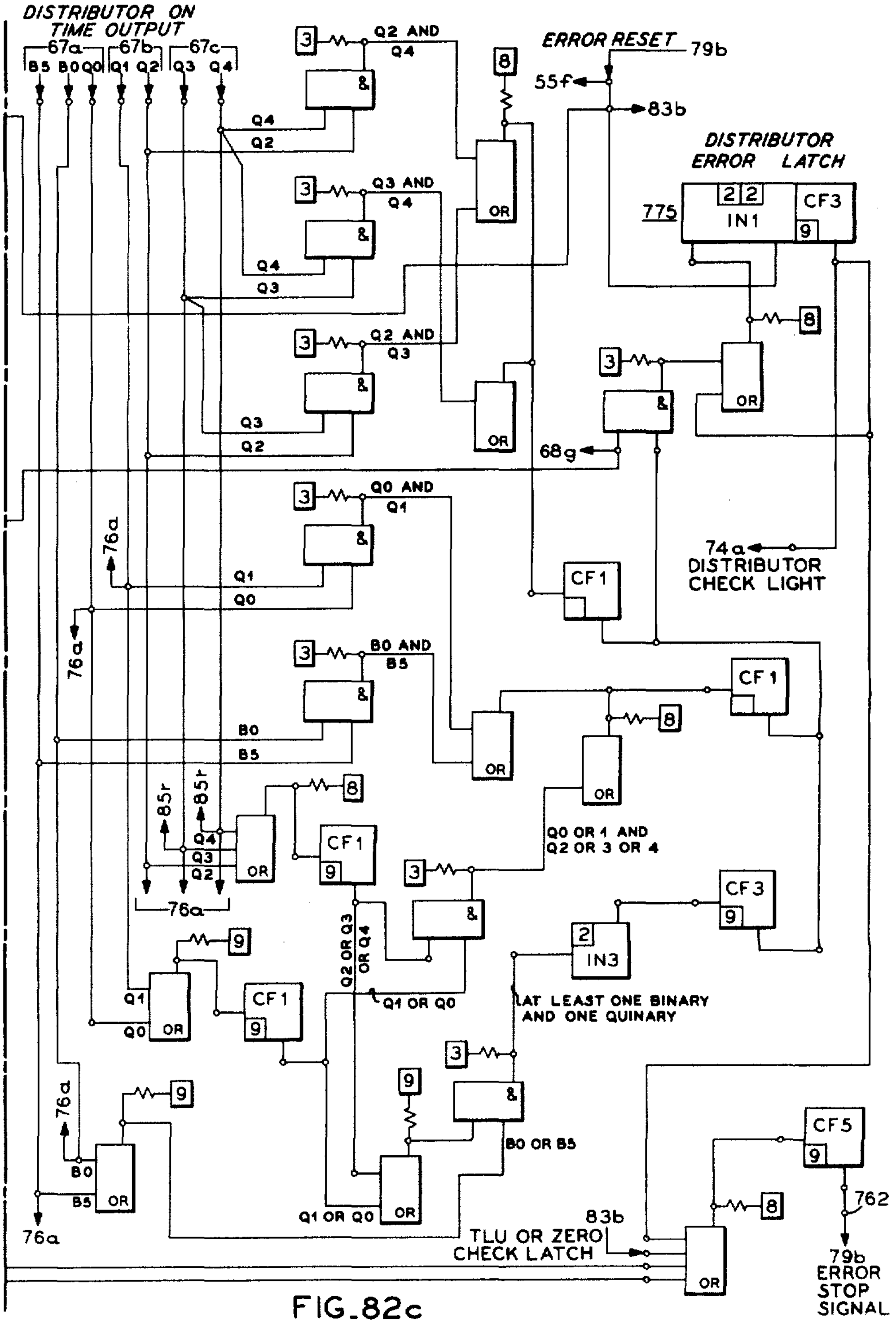


FIG. 82c

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 185

FIG_ 83a

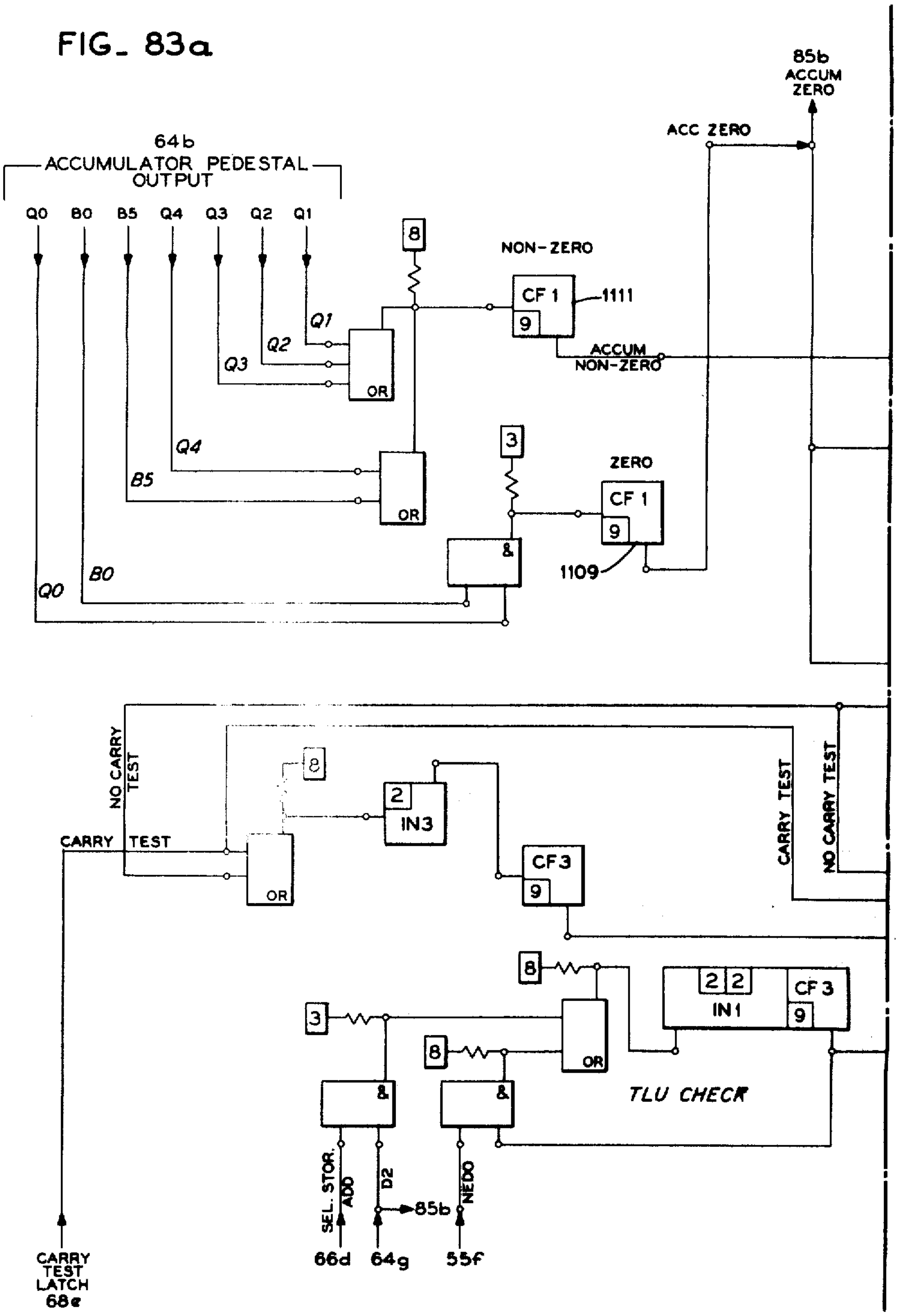
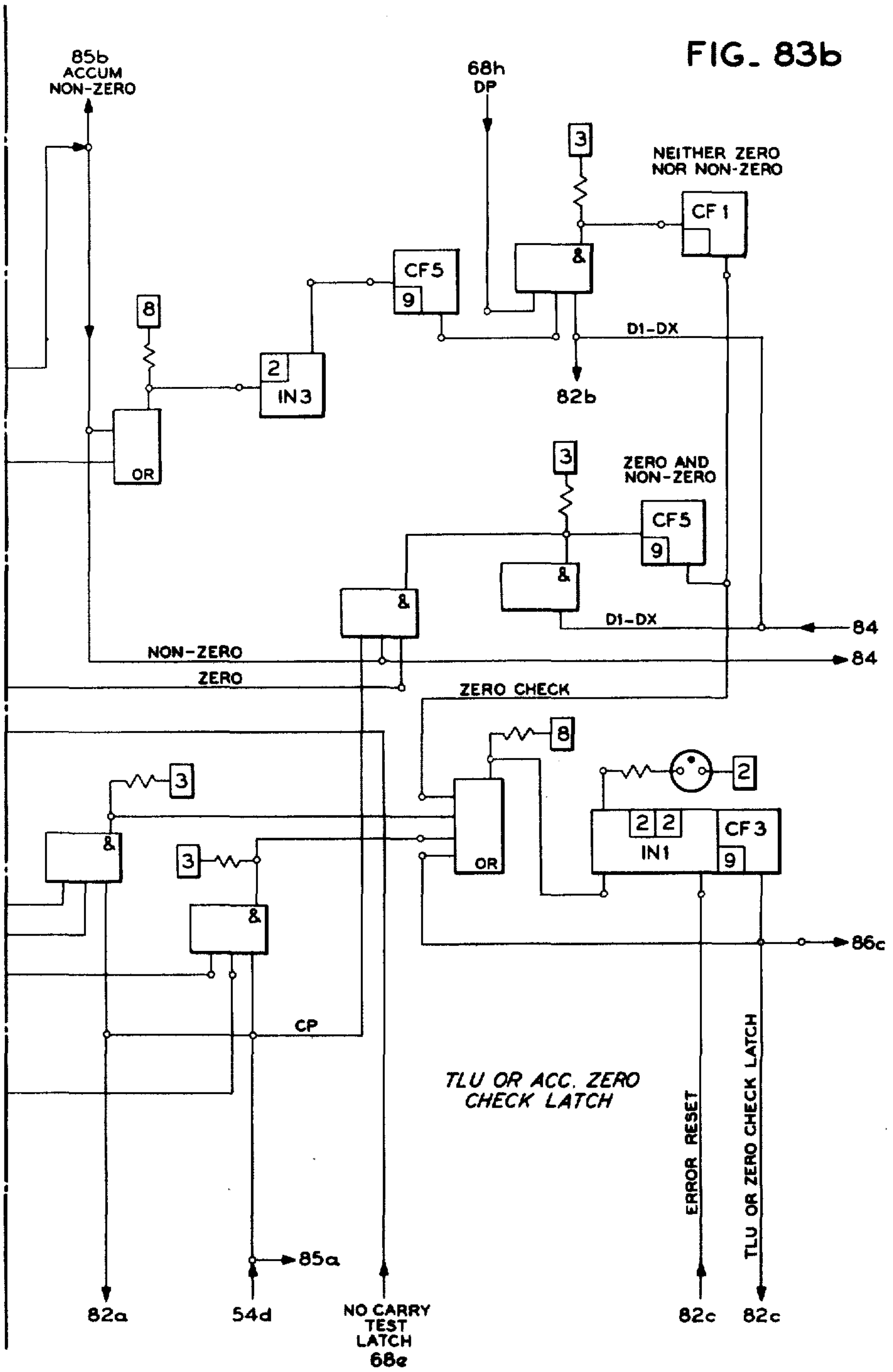


FIG. 83b


85b
ACCUM
NON-ZERO
68h
DP
3
NEITHER ZERO
NOR NON-ZERO
CF 1
&
CF5
9
8
2
IN3
D1-DX
82b
OR
3
ZERO AND
NON-ZERO
CF5
9
&
&
D1-DX
84
NON-ZERO
84
ZERO
ZERO CHECK
8
2
3
&
3
&
OR
2 2
IN1
CF3
9
86c
CP
TLU OR ACC. ZERO
CHECK LATCH
ERROR RESET
TLU OR ZERO CHECK LATCH
85a
82a
54d
NO CARRY
TEST
LATCH
68e
82c
82c

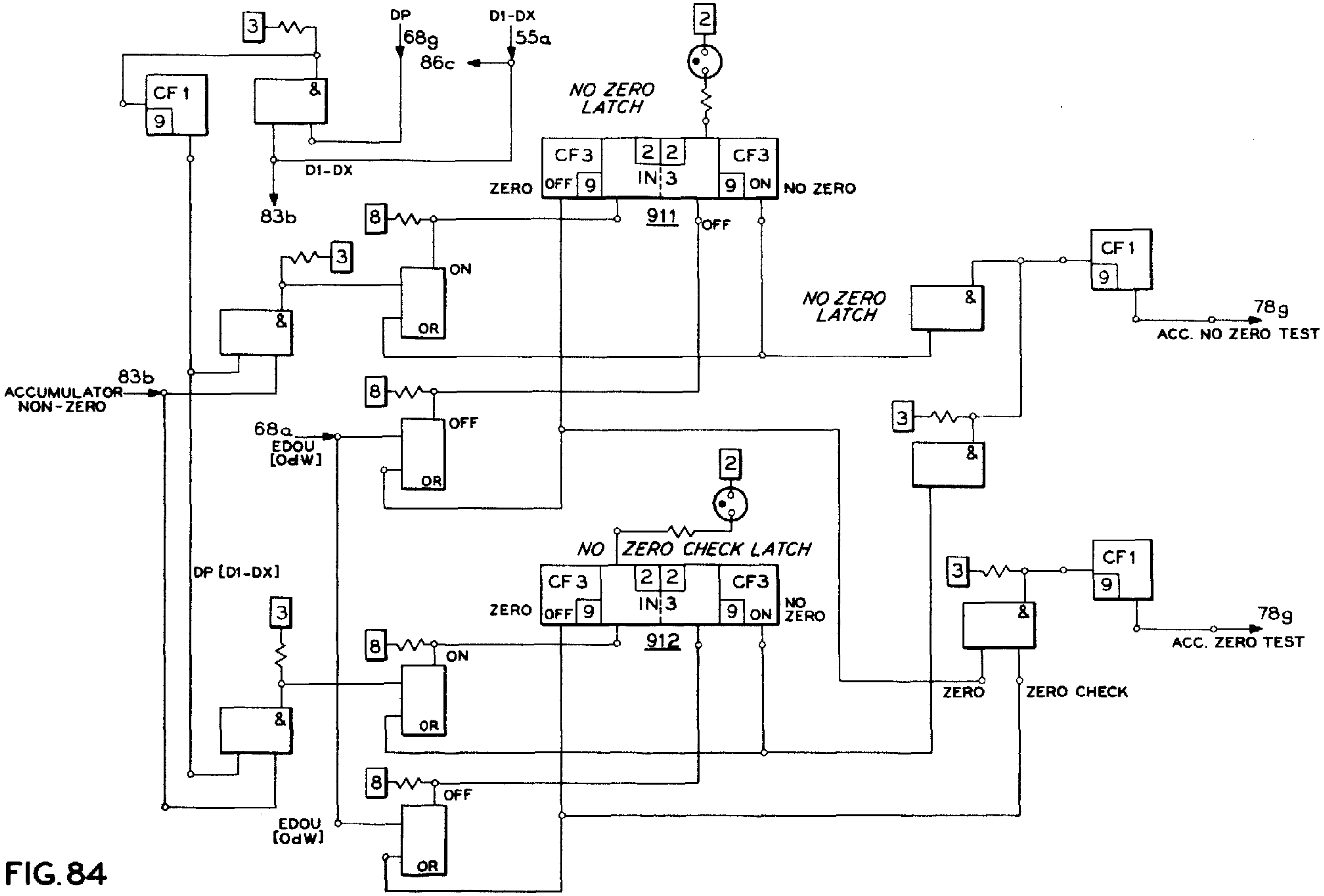

3
DP
68g
D1-DX
55a
86c
CF1
9
&
D1-DX
83b
2
NO ZERO
LATCH
CF3
2 2
IN 3
CF3
ZERO
OFF 9
9 ON
NO ZERO
911
OFF
8
3
ON
&
OR
NO ZERO
LATCH
&
CF1
9
78g
ACC. NO ZERO TEST
83b
ACCUMULATOR
NON-ZERO
8
OFF
68a
EDOU
[OdW]
OR
3
&
2
NO ZERO CHECK LATCH
DP [D1-DX]
CF3
2 2
IN 3
CF3
ZERO
OFF 9
9 ON
NO
ZERO
912
3
CF1
9
&
78g
ACC. ZERO TEST
3
8
ON
&
OR
ZERO
ZERO CHECK
8
OFF
EDOU
[OdW]
OR

FIG. 84

FIG. 85a

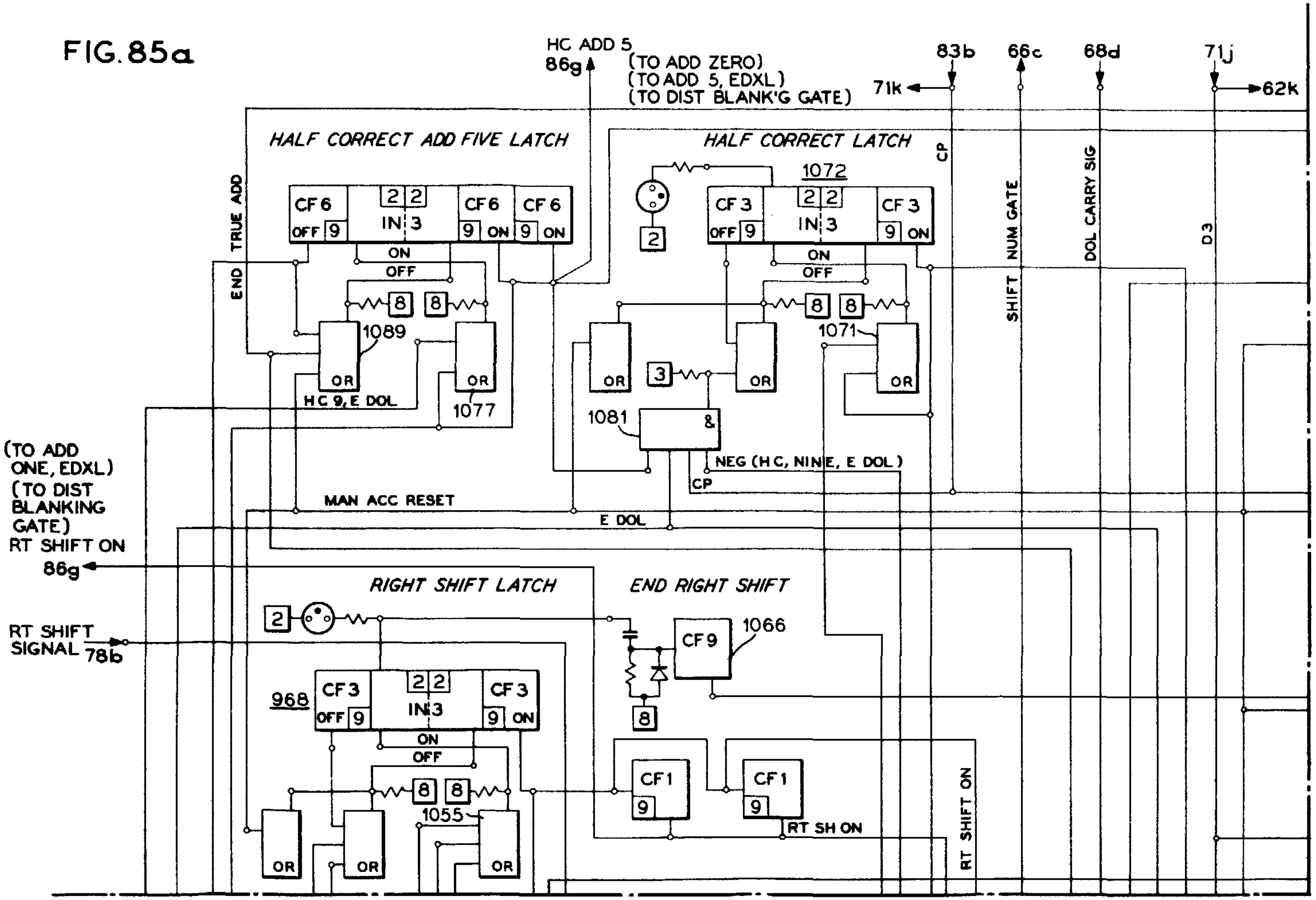

HC ADD 5
86g
(TO ADD ZERO)
(TO ADD 5, EDXL)
(TO DIST BLANK'G GATE)
71k
83b
66c
68d
71j
62k
HALF CORRECT ADD FIVE LATCH
HALF CORRECT LATCH
CP
SHIFT NUM GATE
DOL CARRY SIG
D3
END TRUE ADD
CF 6
OFF
9
2 2
IN 3
ON
OFF
8
1089
OR
1077
HC 9, E DOL
1072
CF 3
1071
1081
&
3
NEG (HC, NINE, E DOL)
CP
MAN ACC RESET
E DOL
(TO ADD ONE, EDXL)
(TO DIST BLANKING GATE)
RT SHIFT ON
86g
RIGHT SHIFT LATCH
END RIGHT SHIFT
RT SHIFT SIGNAL
78b
1066
CF 9
968
1055
CF 1
RT SH ON
RT SHIFT ON

FIG. 85b

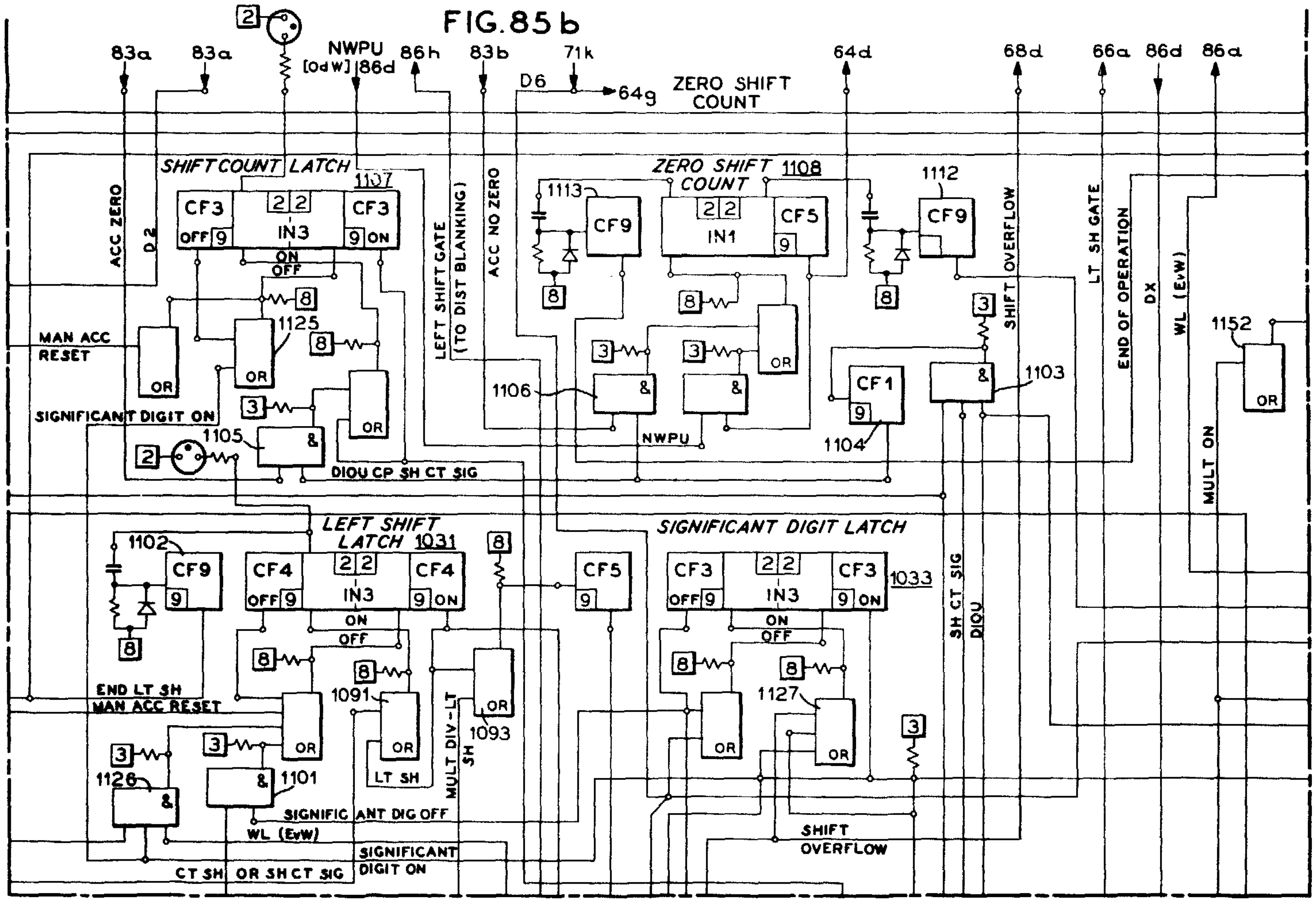

83a
83a
NWPU
[OdW] 86d
86h
83b
71k
D6
64g
ZERO SHIFT COUNT
64d
68d
66a
86d
86a
SHIFT COUNT LATCH 1107
CF3
OFF 9
2 2
IN3
CF3
9 ON
ON
OFF
ACC ZERO
D2
8
1125
OR
MAN ACC RESET
OR
3
OR
SIGNIFICANT DIGIT ON
2
1105
&
DIOU CP SH CT SIG
LEFT SHIFT GATE (TO DIST BLANKING)
ACC NO ZERO
ZERO SHIFT COUNT 1108
1113
CF9
2 2
IN1
CF5
9
1112
CF9
8
3
&
OR
1106
NWPU
SHIFT OVERFLOW
CF1
9
1104
&
1103
LT SH GATE
END OF OPERATION
DX
WL (EvW)
1152
OR
MULT ON
LEFT SHIFT LATCH 1031
1102
CF9
9
CF4
OFF 9
IN3
CF4
9 ON
ON
OFF
8
1091
OR
LT SH
END LT SH
MAN ACC RESET
3
OR
1126
&
1101
&
SIGNIFIC ANT DIG OFF
WL (EvW)
SIGNIFICANT DIGIT ON
CT SH OR SH CT SIG
MULT DIV-LT SH
1093
OR
CF5
9
SIGNIFICANT DIGIT LATCH
CF3
OFF 9
2 2
IN3
CF3
9 ON
1033
ON
OFF
8
1127
OR
OR
3
SHIFT OVERFLOW
SH CT SIG
DIOU

FIG. 85 c

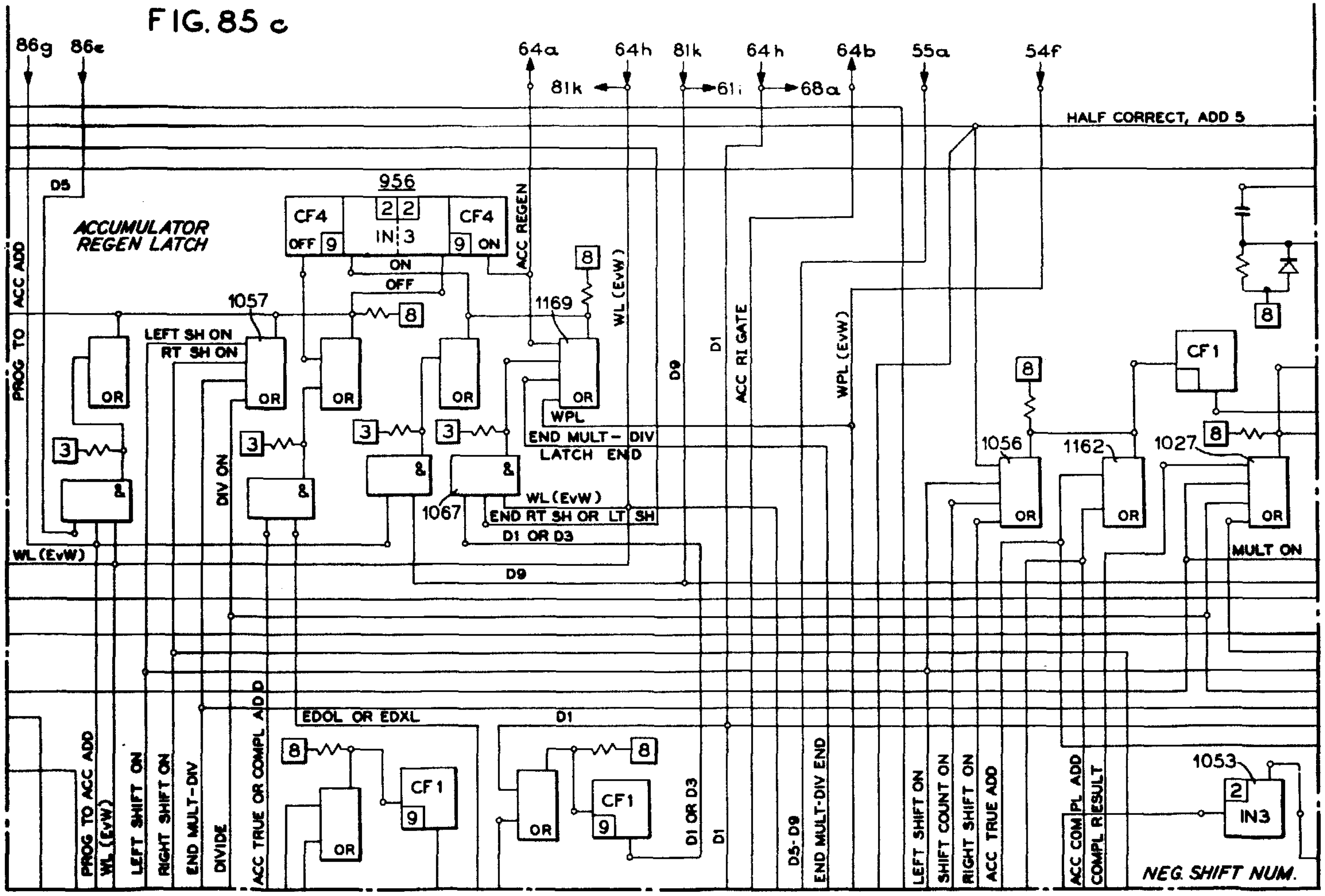
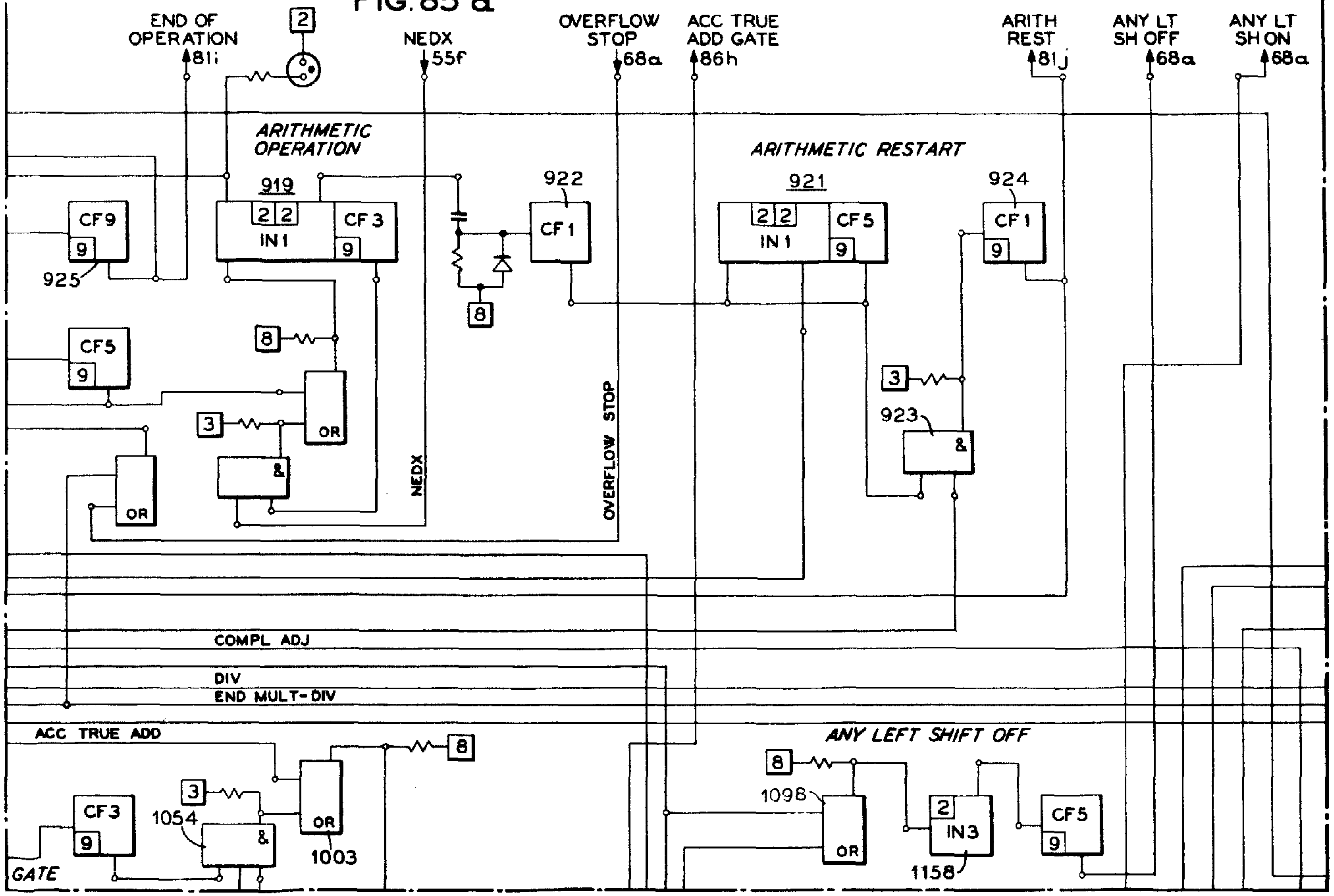

86g
86e
64a
64h
81k
64h
64b
55a
54f
81k
61i
68a
HALF CORRECT, ADD 5
D5
ACCUMULATOR REGEN LATCH
956
CF4
OFF
9
2
2
IN 3
CF4
9
ON
ON
OFF
ACC REGEN
PROG TO ACC ADD
1057
LEFT SH ON
RT SH ON
OR
OR
OR
OR
OR
8
8
1169
WL (EvW)
D9
D1
ACC RI GATE
WPL (EvW)
WPL
END MULT-DIV
LATCH END
3
3
3
3
&
&
&
&
DIV ON
1067
WL (EvW)
END RT SH OR LT SH
D1 OR D3
D9
WL (EvW)
8
8
8
CF1
1056
1162
1027
OR
OR
OR
MULT ON
EDOL OR EDXL
D1
8
8
CF1
9
CF1
9
OR
OR
PROG TO ACC ADD
WL (EvW)
LEFT SHIFT ON
RIGHT SHIFT ON
END MULT-DIV
DIVIDE
ACC TRUE OR COMPL ADD
D1 OR D3
D1
D5-D9
END MULT-DIV END
LEFT SHIFT ON
SHIFT COUNT ON
RIGHT SHIFT ON
ACC TRUE ADD
ACC COMPL ADD
COMPL RESULT
1053
2
IN3
NEG. SHIFT NUM.

FIG. 85 d

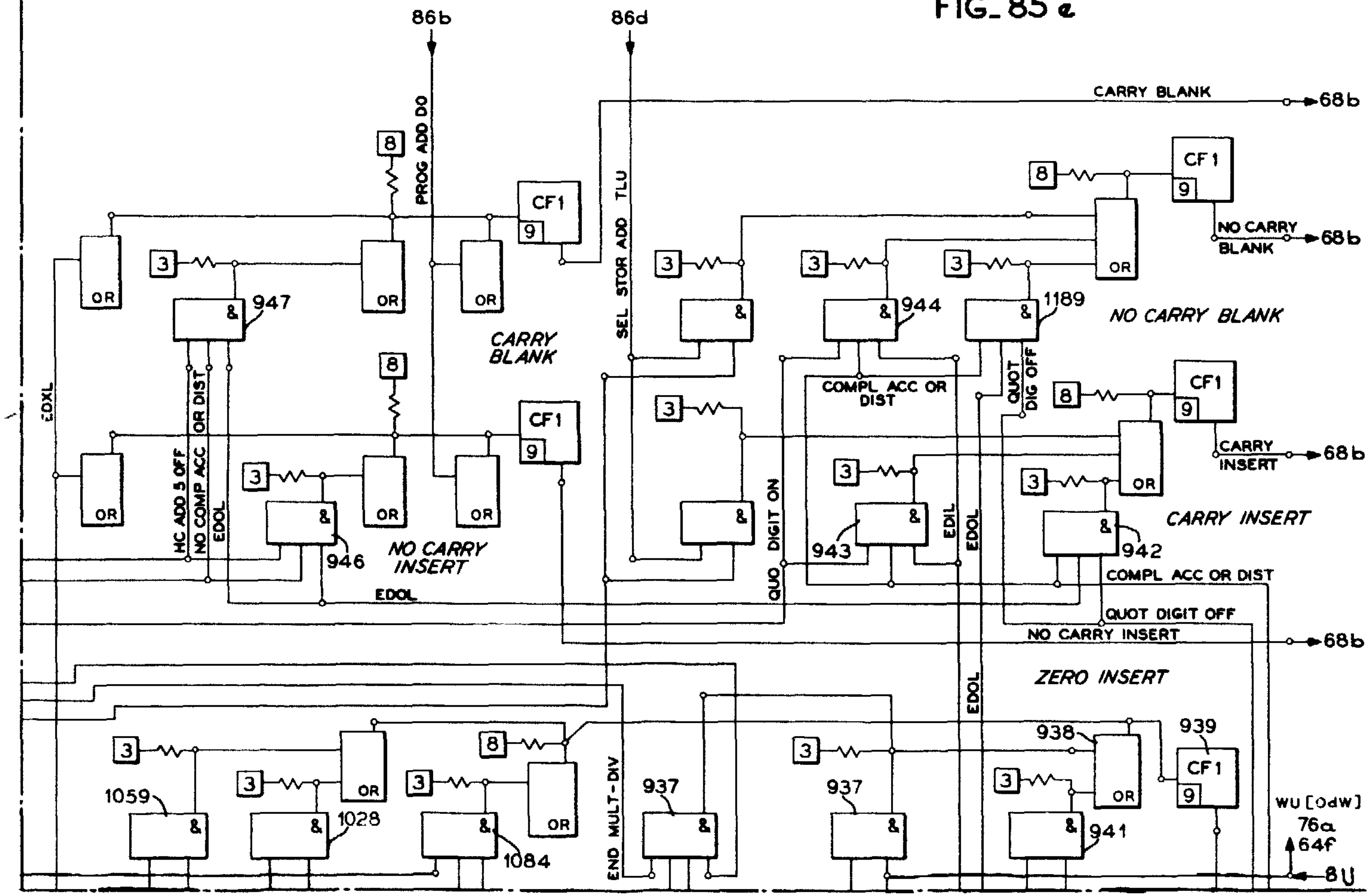

END OF OPERATION
81i
2
NEDX
55f
OVERFLOW STOP
68a
ACC TRUE ADD GATE
86h
ARITH REST
81j
ANY LT SH OFF
68a
ANY LT SH ON
68a
ARITHMETIC OPERATION
919
2 2
IN 1
CF 3
9
922
CF 1
8
ARITHMETIC RESTART
921
2 2
IN 1
CF 5
9
924
CF 1
9
CF 9
9
925
CF 5
9
8
3
OR
&
OR
NEDX
OVERFLOW STOP
3
923
&
COMPL ADJ
DIV
END MULT-DIV
ACC TRUE ADD
8
3
CF 3
9
1054
&
OR
1003
GATE
ANY LEFT SHIFT OFF
8
1098
OR
2
IN3
1158
CF5
9

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 192

FIG_85 e



Nov. 8, 1960

F. E. HAMILTON ET AL

2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955

228 Sheets-Sheet 193

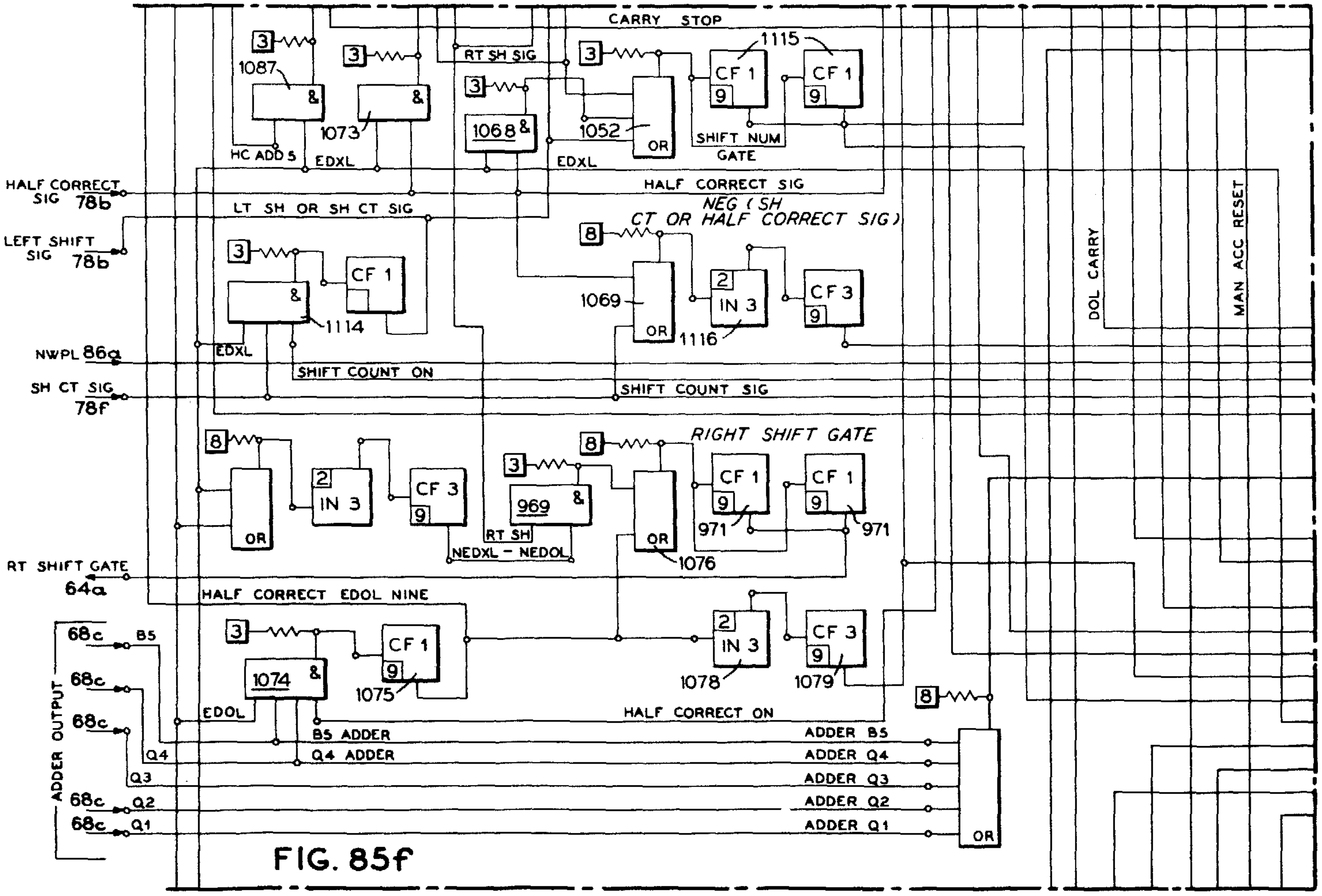
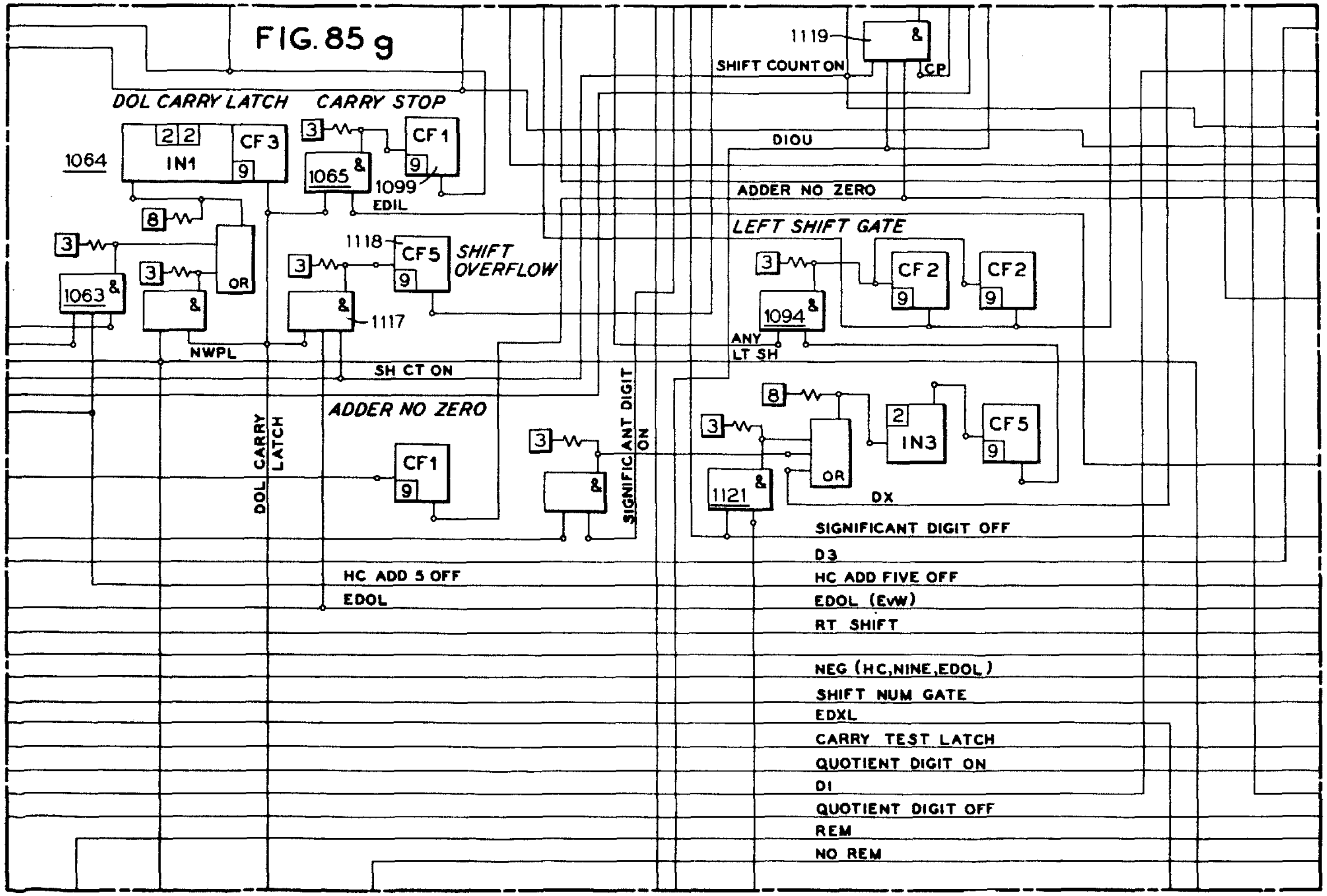


FIG. 85f

FIG. 85 g


DOL CARRY LATCH
CARRY STOP
1064
2 2
IN1
CF 3
9
3
CF 1
1065 &
9
1099
EDIL
8
3
3
OR
1063 &
&
1118
CF5
9
SHIFT OVERFLOW
3
&
1117
NWPL
SH CT ON
ADDER NO ZERO
DOL CARRY LATCH
CF1
9
3
&
SIGNIFICANT DIGIT ON
1119
&
SHIFT COUNT ON
CP
DIOU
ADDER NO ZERO
LEFT SHIFT GATE
3
CF 2
9
CF 2
9
1094 &
ANY LT SH
8
3
2
IN3
CF5
9
OR
1121 &
DX
SIGNIFICANT DIGIT OFF
D3
HC ADD 5 OFF
HC ADD FIVE OFF
EDOL
EDOL (EvW)
RT SHIFT
NEG (HC,NINE,EDOL)
SHIFT NUM GATE
EDXL
CARRY TEST LATCH
QUOTIENT DIGIT ON
DI
QUOTIENT DIGIT OFF
REM
NO REM

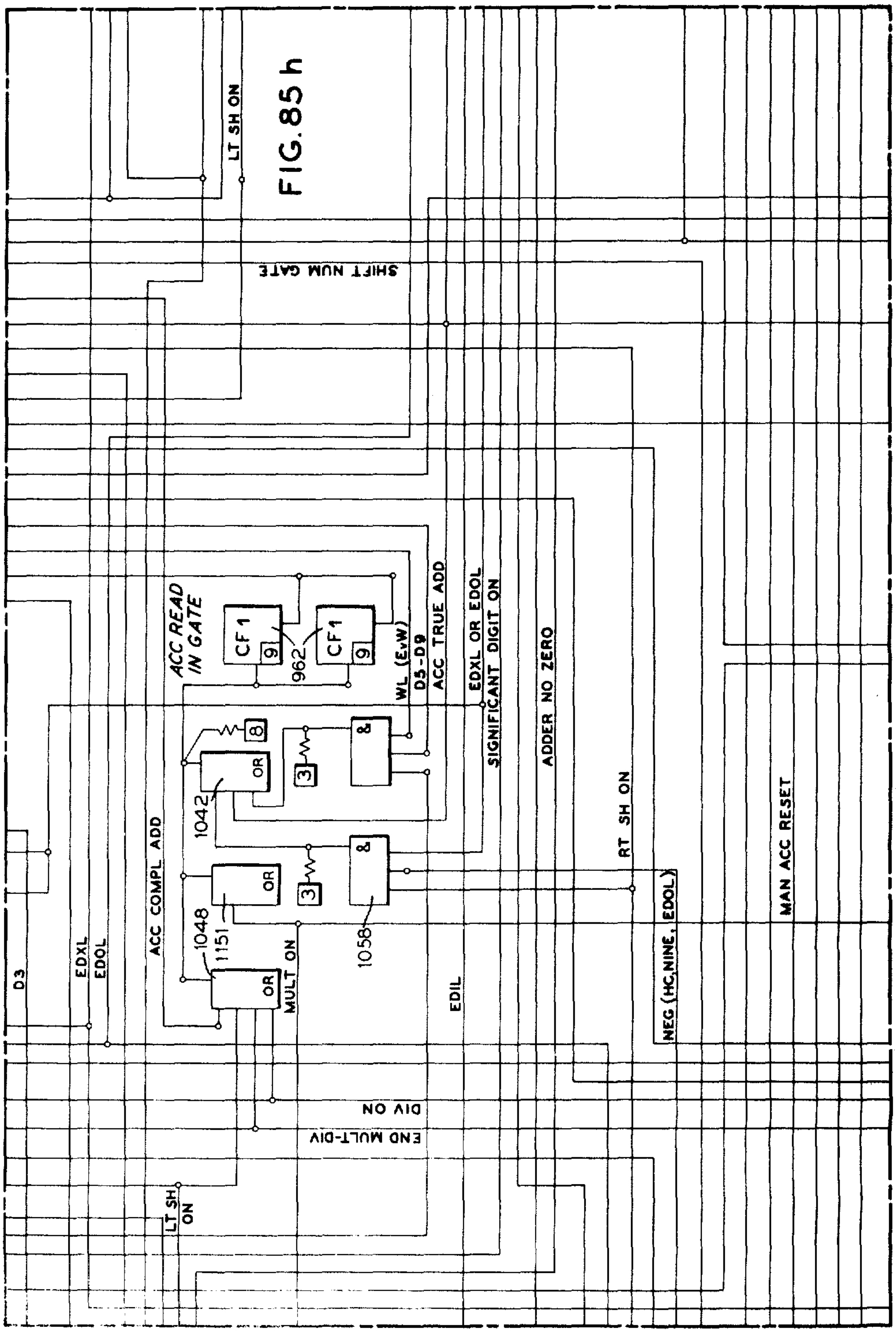
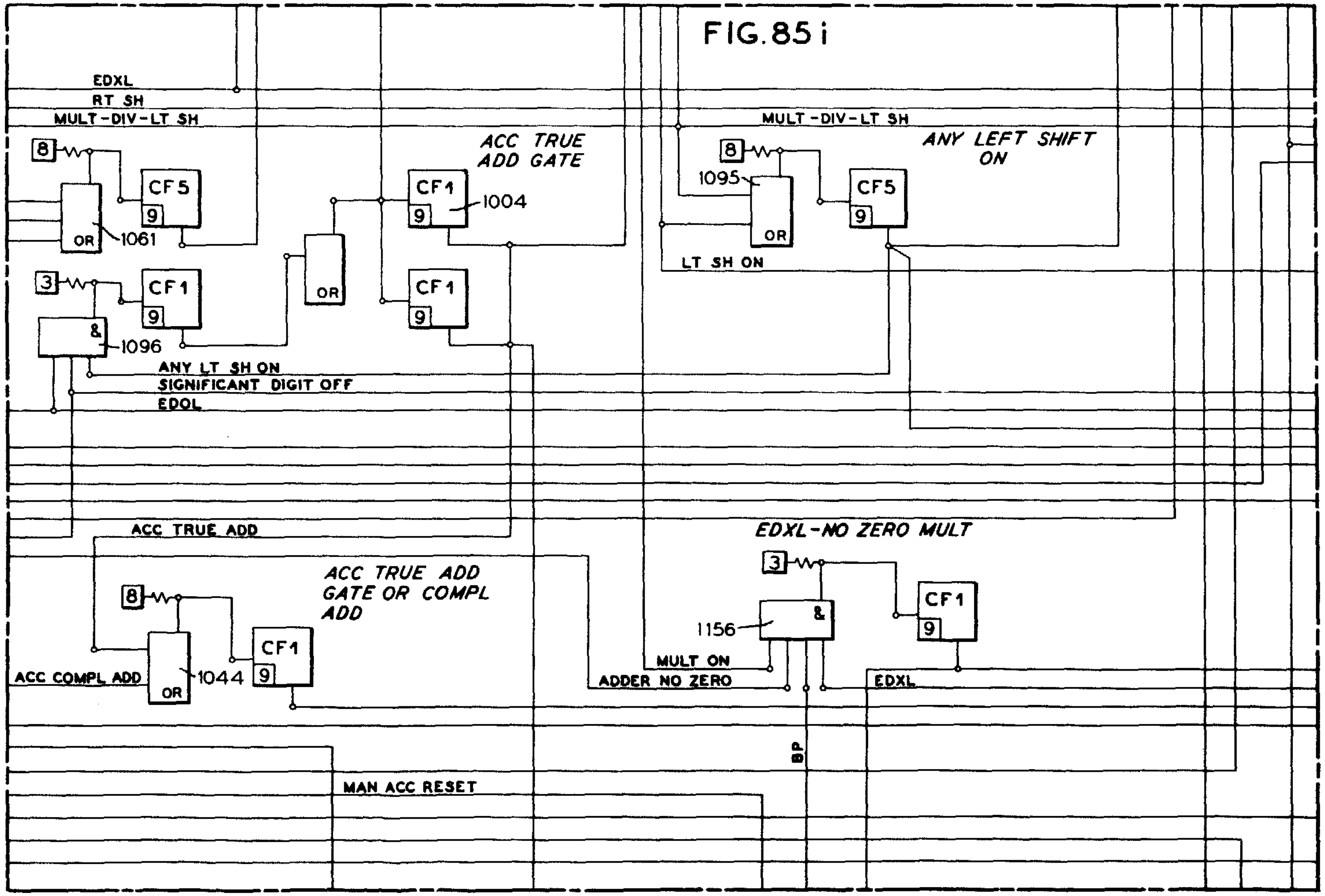


FIG. 85 h

LT SH ON
SHIFT NUM GATE
ACC READ IN GATE
CF1
9
CF1
9
962
WL (EvW)
D5-D9
ACC TRUE ADD
EDXL OR EDOL
SIGNIFICANT DIGIT ON
ADDER NO ZERO
8
OR
3
&
1042
ACC COMPL ADD
OR
3
&
1048
1151
1058
OR
MULT ON
D3
EDXL
EDOL
EDIL
RT SH ON
NEG (HC,NINE, EDOL)
MAN ACC RESET
DIV ON
END MULT-DIV
LT SH ON

FIG. 85i


EDXL
RT SH
MULT-DIV-LT SH
8
CF5
9
OR
1061
3
CF1
9
&
1096
ANY LT SH ON
SIGNIFICANT DIGIT OFF
EDOL
OR
ACC TRUE
ADD GATE
CF1
9
1004
CF1
9
MULT-DIV-LT SH
8
1095
OR
CF5
9
ANY LEFT SHIFT
ON
LT SH ON
ACC TRUE ADD
8
ACC COMPL ADD
OR
1044
CF1
9
ACC TRUE ADD
GATE OR COMPL
ADD
EDXL-NO ZERO MULT
3
1156
&
CF1
9
MULT ON
ADDER NO ZERO
EDXL
BP
MAN ACC RESET

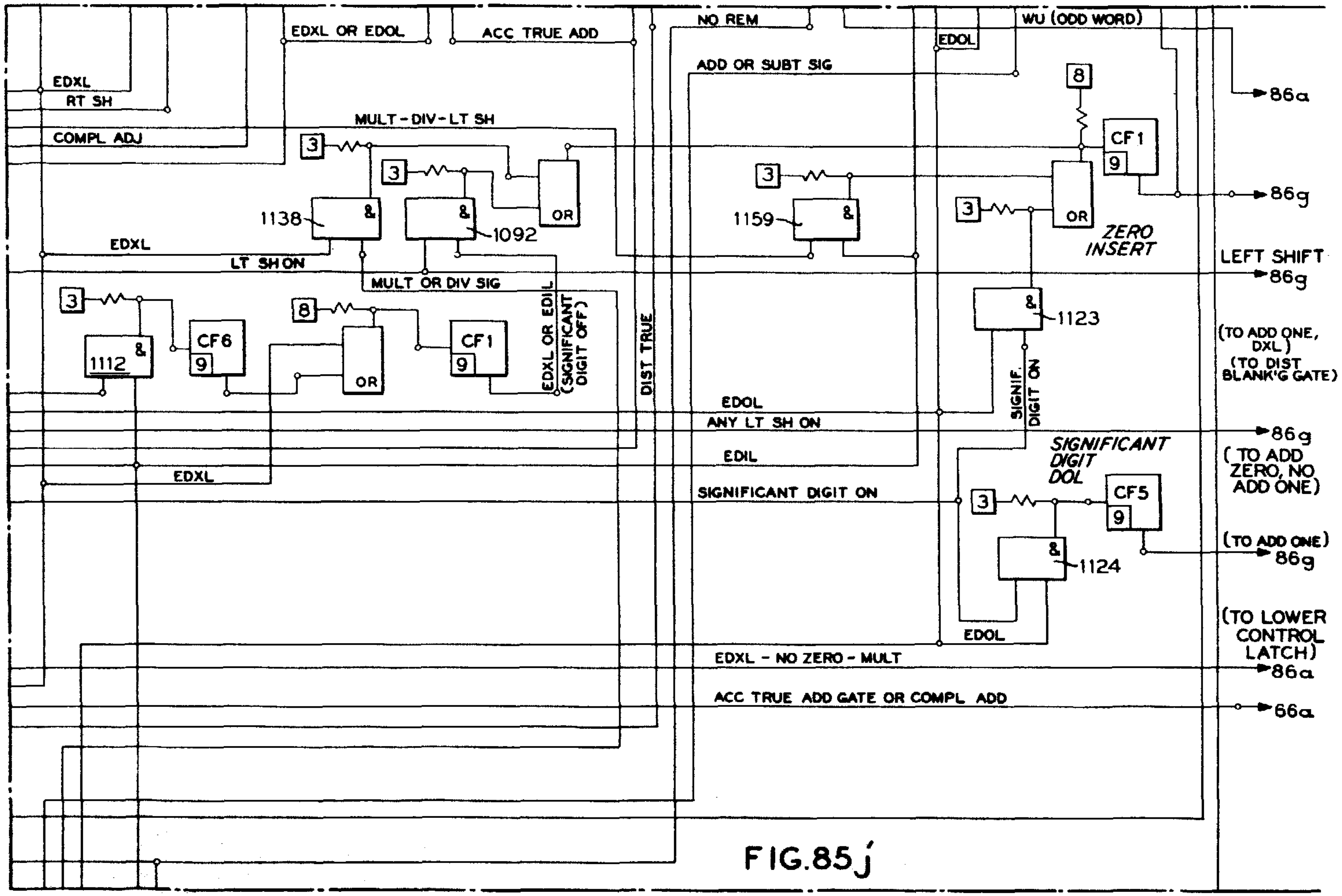

EDXL OR EDOL
ACC TRUE ADD
NO REM
EDOL
WU (ODD WORD)
ADD OR SUBT SIG
EDXL
RT SH
MULT-DIV-LT SH
COMPL ADJ
3
3
8
CF1
9
86a
OR
1138
&
&
1092
OR
3
1159
&
3
OR
86g
ZERO INSERT
EDXL
LT SH ON
MULT OR DIV SIG
LEFT SHIFT
86g
3
8
CF6
9
1112
&
OR
CF1
9
EDXL OR EDIL (SIGNIFICANT DIGIT OFF)
DIST TRUE
&
1123
(TO ADD ONE, DXL)
(TO DIST BLANK'G GATE)
EDOL
SIGNIF. DIGIT ON
ANY LT SH ON
86g
SIGNIFICANT DIGIT DOL
(TO ADD ZERO, NO ADD ONE)
EDIL
EDXL
SIGNIFICANT DIGIT ON
3
CF5
9
&
1124
(TO ADD ONE)
86g
EDOL
(TO LOWER CONTROL LATCH)
EDXL - NO ZERO - MULT
86a
ACC TRUE ADD GATE OR COMPL ADD
66a

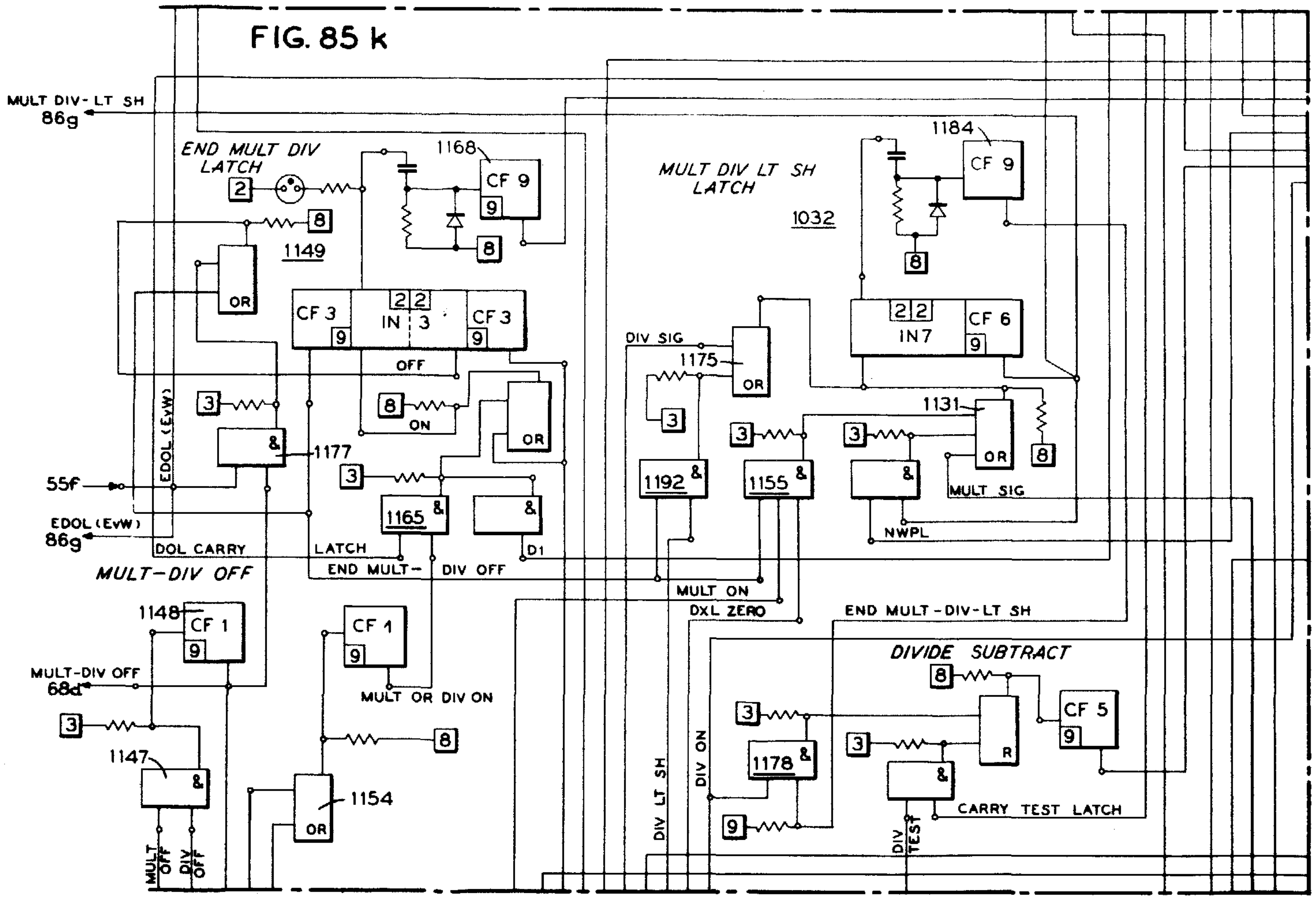
FIG.85j́

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 198

FIG. 85 k



Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 199

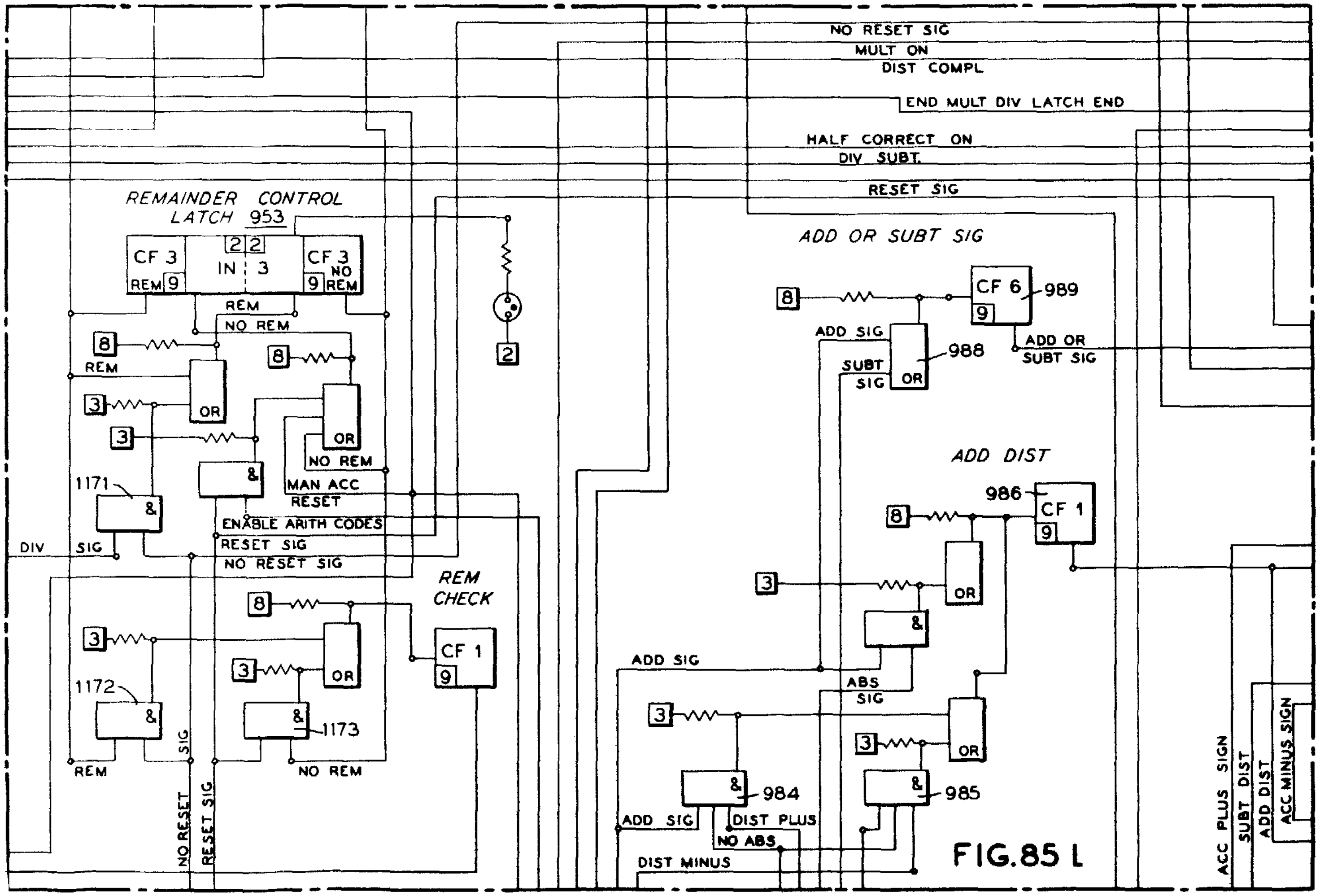


FIG. 85 L

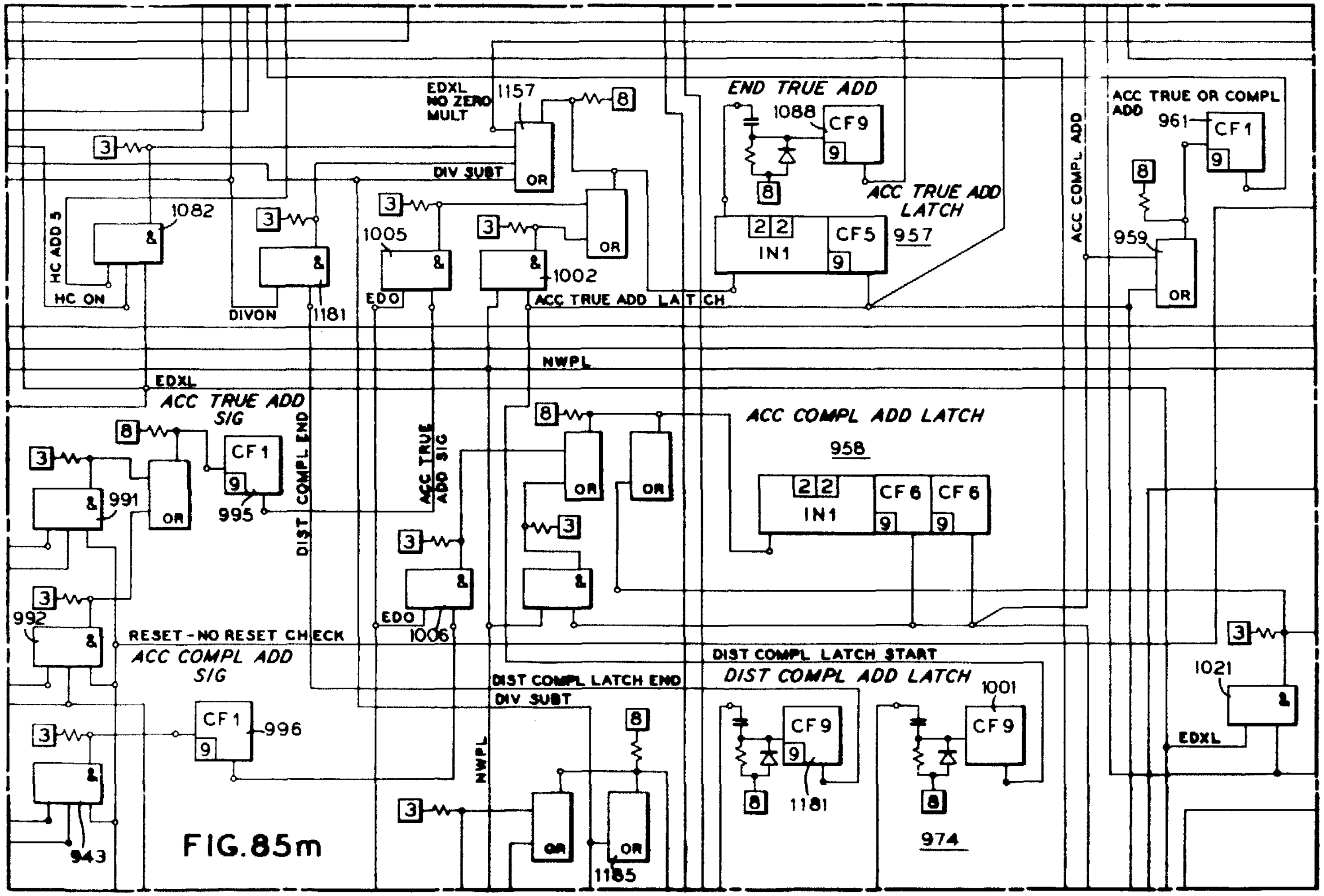

EDXL
NO ZERO
MULT
1157
8
DIV SUBT
OR
3
HC ADD 5
1082
&
HC ON
3
&
DIVON
1181
3
1005
&
EDO
3
&
1002
OR
ACC TRUE ADD LATCH
END TRUE ADD
1088
CF9
9
8
ACC TRUE ADD
LATCH
2 2
IN1
CF5
9
957
ACC COMPL ADD
ACC TRUE OR COMPL
ADD
961
CF1
9
8
959
OR
NWPL
EDXL
ACC TRUE ADD
SIG
8
3
CF1
9
OR
&
991
995
DIST COMPL END
ACC TRUE
ADD SIG
8
OR
OR
3
ACC COMPL ADD LATCH
958
2 2
IN1
CF6
9
CF6
9
3
&
3
&
3
992
&
RESET-NO RESET CHECK
ACC COMPL ADD
SIG
EDO
1006
DIST COMPL LATCH START
DIST COMPL ADD LATCH
1001
DIST COMPL LATCH END
DIV SUBT
3
1021
&
EDXL
3
CF1
9
996
&
943
NWPL
8
3
OR
OR
1185
CF9
9
8
1181
CF9
8
974

FIG.85m

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

Filed Nov. 2, 1955

DATA STORAGE AND PROCESSING MACHINE

228 Sheets-Sheet 201

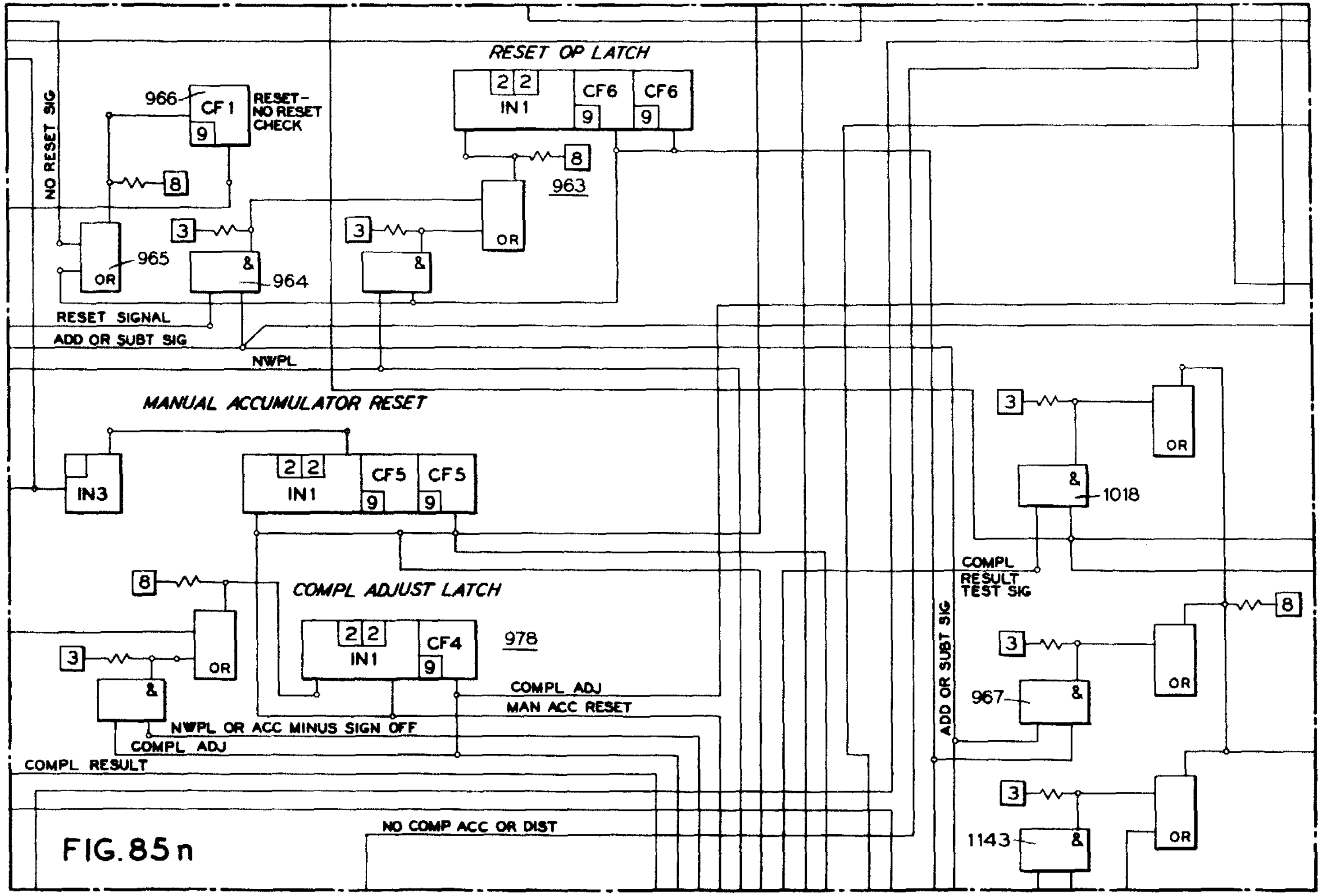
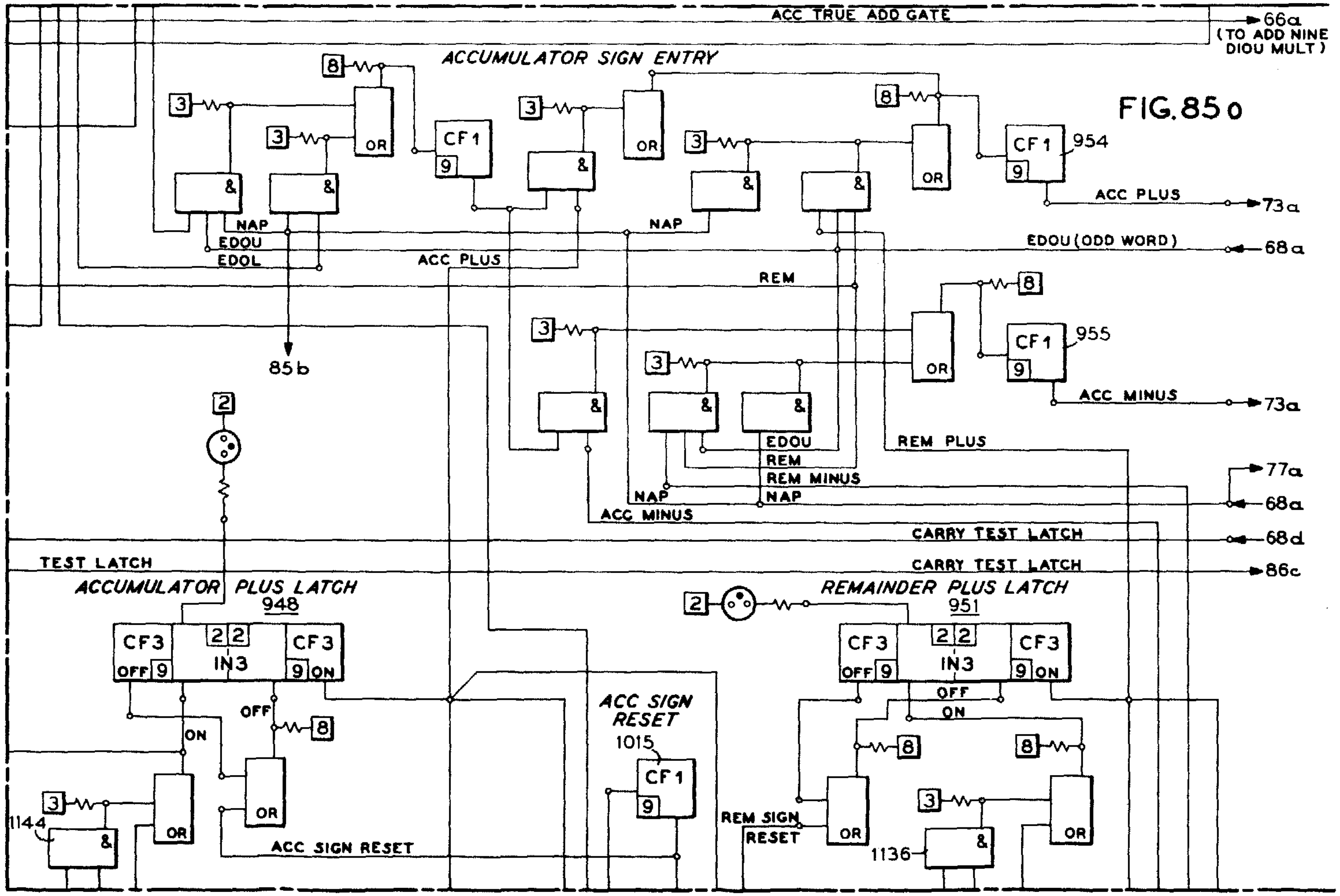


FIG. 85n

FIG. 85o


ACC TRUE ADD GATE
66a (TO ADD NINE DIOU MULT)
ACCUMULATOR SIGN ENTRY
8
3
3
OR
CF1
9
&
&
NAP
EDOU
EDOL
85b
ACC PLUS
3
OR
&
3
&
NAP
8
OR
&
CF1
954
9
ACC PLUS
73a
EDOU (ODD WORD)
68a
REM
8
CF1
955
9
3
OR
3
&
&
&
ACC MINUS
73a
EDOU
REM
REM MINUS
NAP
NAP
ACC MINUS
REM PLUS
77a
68a
CARRY TEST LATCH
68d
CARRY TEST LATCH
86c
2
TEST LATCH
ACCUMULATOR PLUS LATCH
948
CF3
OFF
9
2 2
IN3
CF3
9
ON
ON
OFF
8
3
1144
&
OR
OR
ACC SIGN RESET
ACC SIGN RESET
1015
CF1
9
2
REMAINDER PLUS LATCH
951
CF3
OFF
9
2 2
IN3
CF3
9
ON
OFF
ON
8
8
REM SIGN RESET
OR
3
1136
&
OR

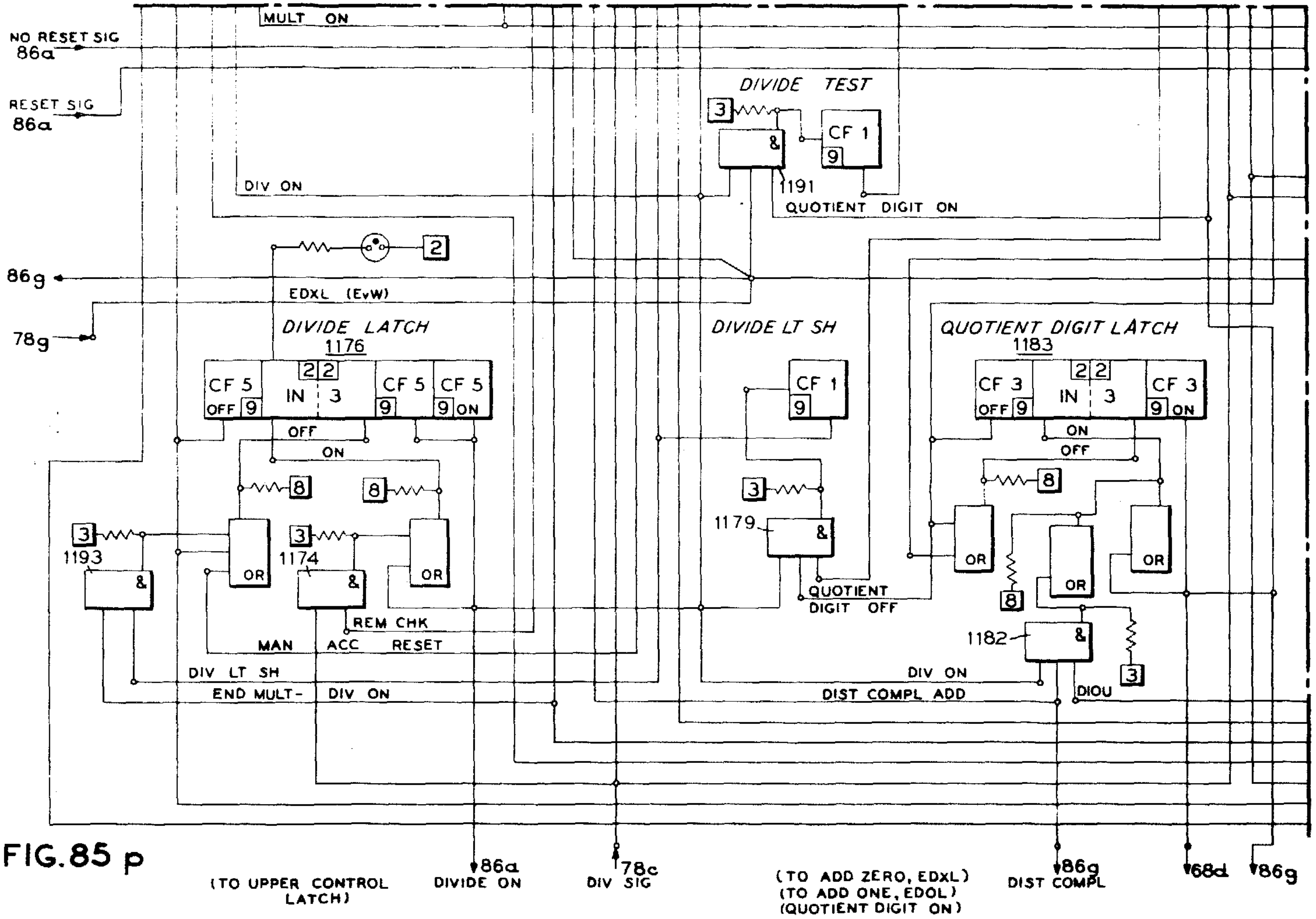

MULT ON
NO RESET SIG
86a
RESET SIG
86a
DIVIDE TEST
3
&
CF 1
9
1191
DIV ON
QUOTIENT DIGIT ON
2
86g
EDXL (EvW)
78g
DIVIDE LATCH
1176
CF 5
OFF
9
2 2
IN 3
CF 5
9
CF 5
9 ON
OFF
ON
8
8
3
1193
&
OR
3
1174
&
OR
REM CHK
MAN
ACC
RESET
DIV LT SH
END MULT-
DIV ON
DIVIDE LT SH
CF 1
9
3
1179
&
QUOTIENT
DIGIT OFF
QUOTIENT DIGIT LATCH
1183
CF 3
OFF 9
2 2
IN 3
CF 3
9 ON
ON
OFF
8
OR
8
OR
OR
1182
&
3
DIV ON
DIST COMPL ADD
DIOU
86a
DIVIDE ON
78c
DIV SIG
(TO UPPER CONTROL LATCH)
(TO ADD ZERO, EDXL)
(TO ADD ONE, EDOL)
(QUOTIENT DIGIT ON)
86g
DIST COMPL
68d
86g

FIG. 85 p

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 204

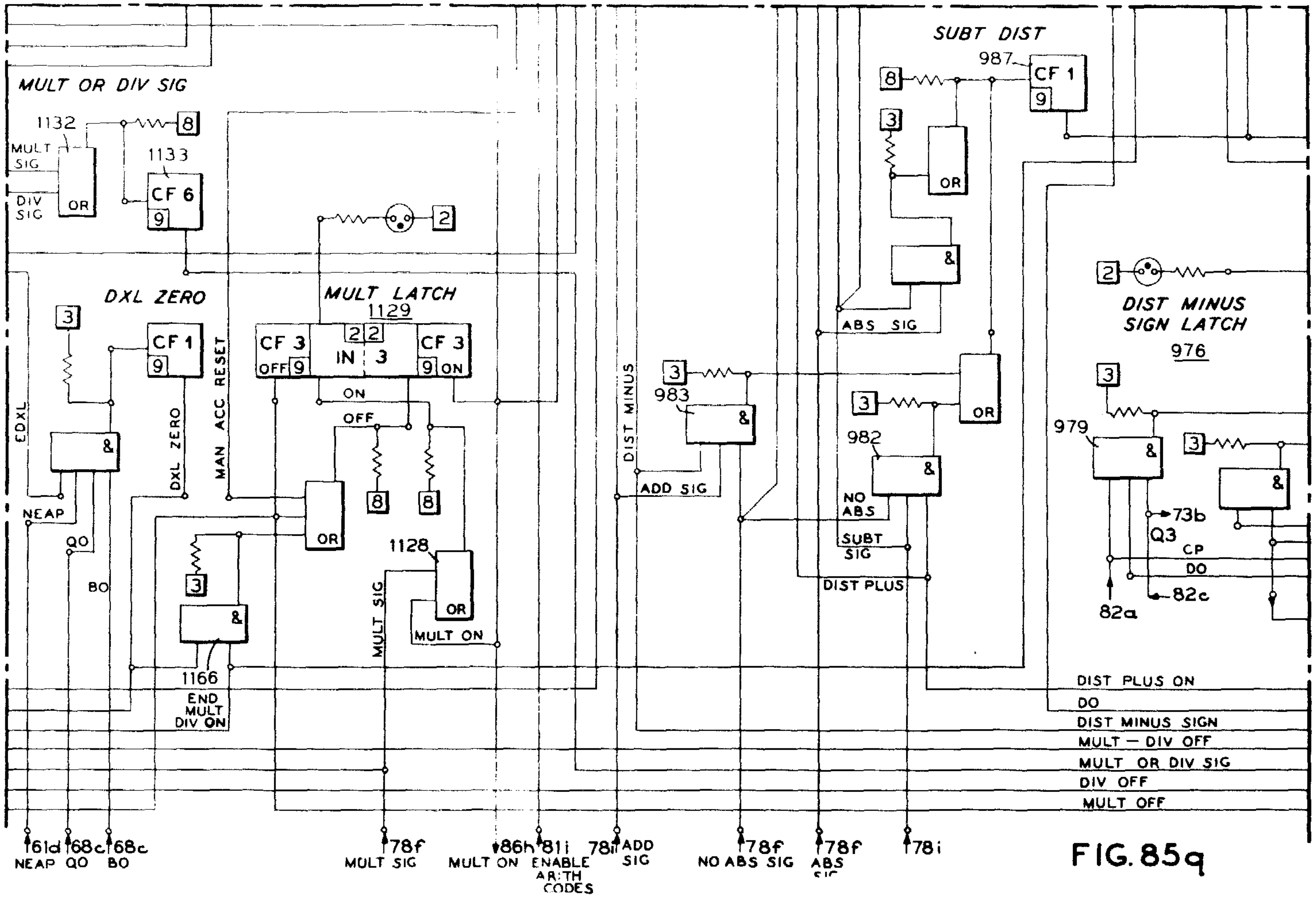


FIG. 85q

Nov. 8, 1960

F. E. HAMILTON ET AL

2,959,351

Filed Nov. 2, 1955

DATA STORAGE AND PROCESSING MACHINE

228 Sheets-Sheet 205

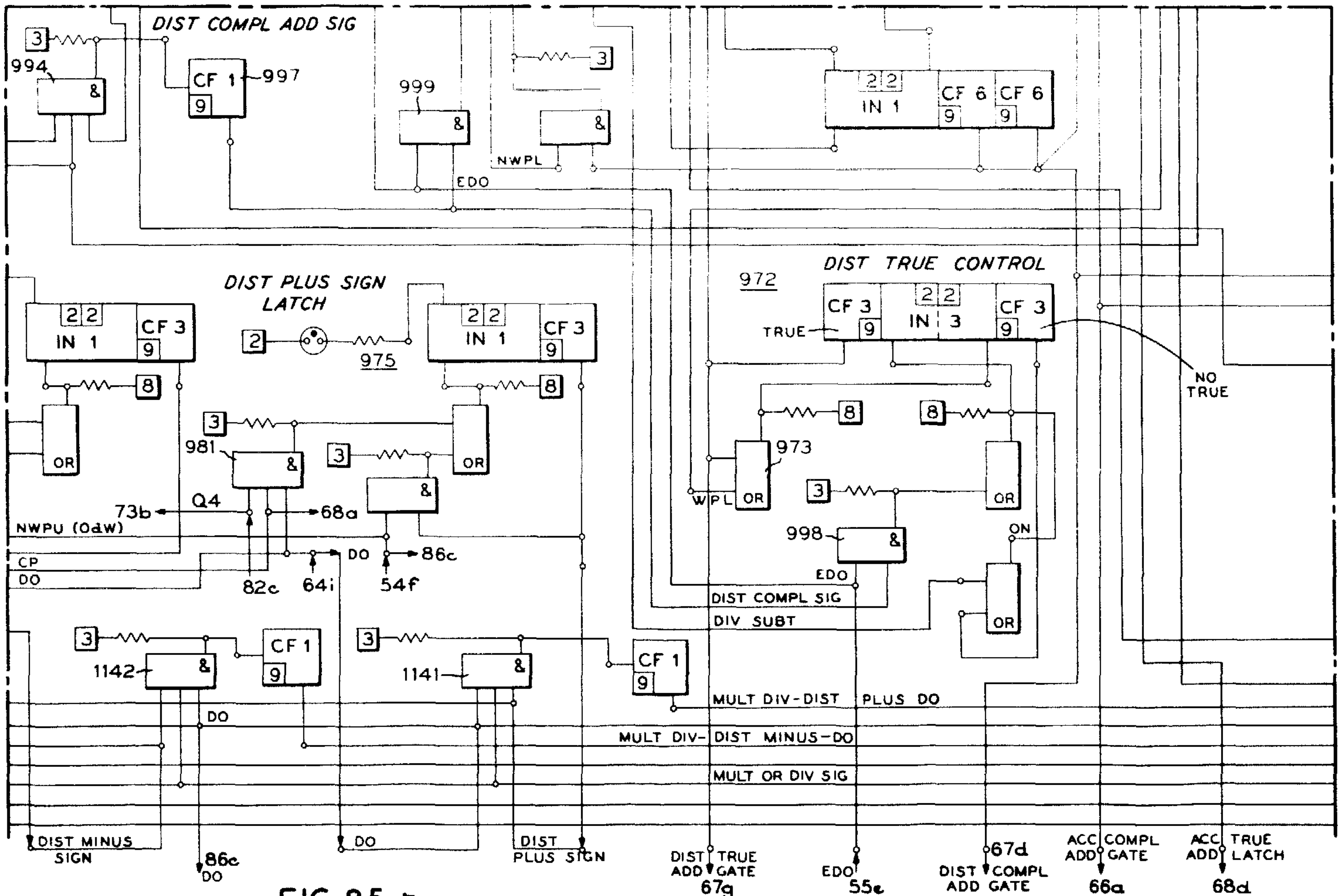


FIG. 85 r

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

Filed Nov. 2, 1955

DATA STORAGE AND PROCESSING MACHINE

228 Sheets-Sheet 206

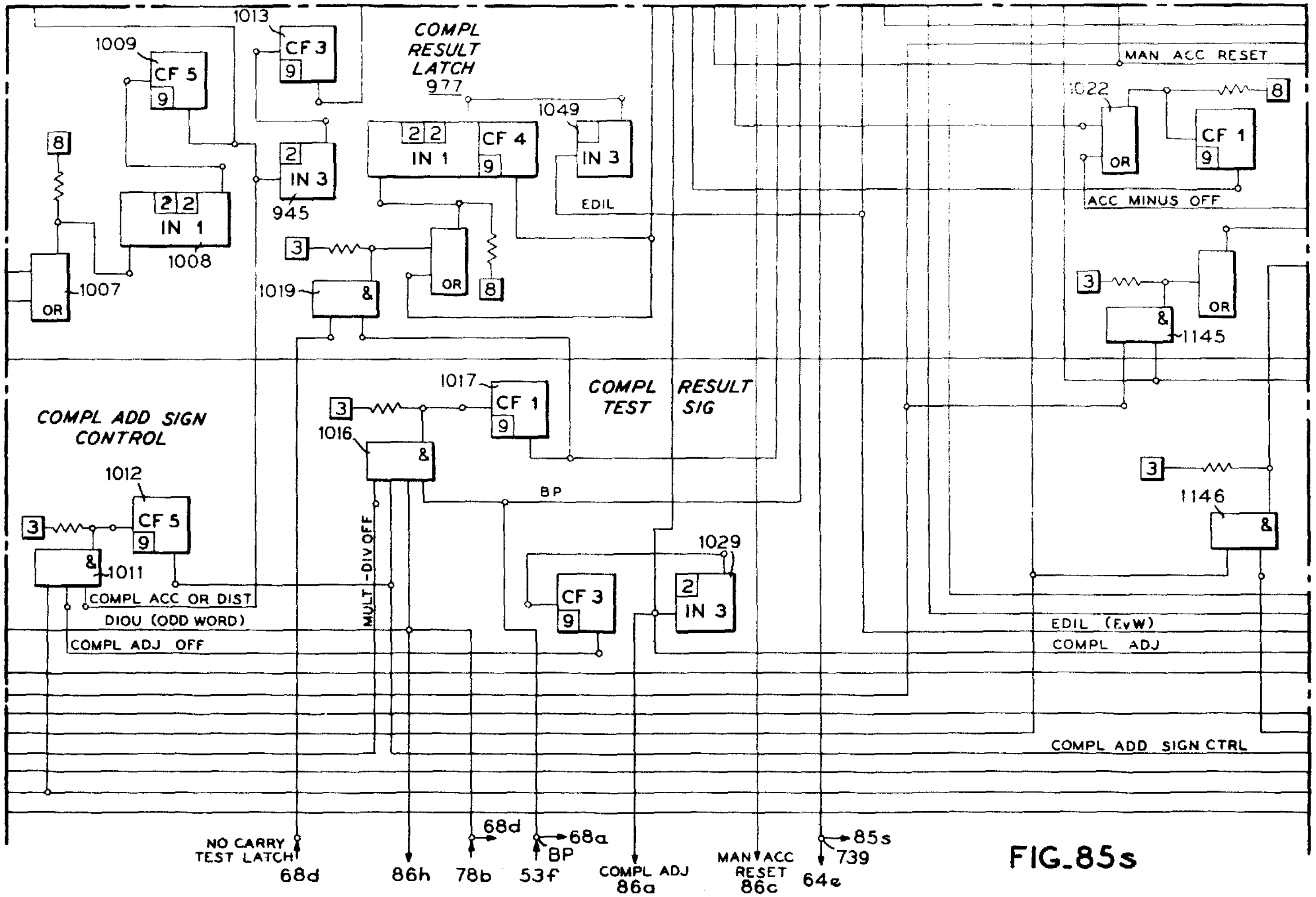


FIG. 85s

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 207

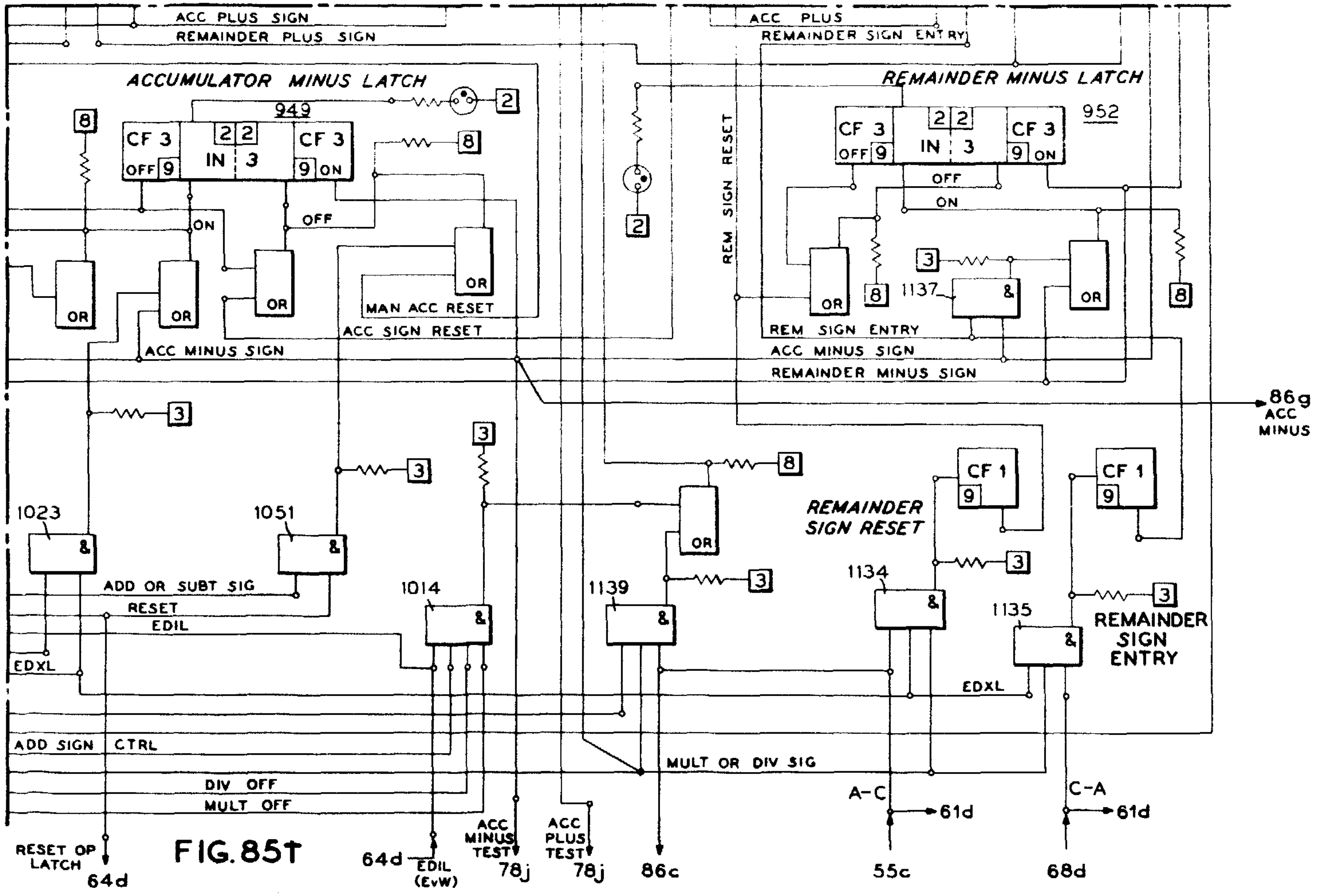


FIG. 85T

FIG. 86a

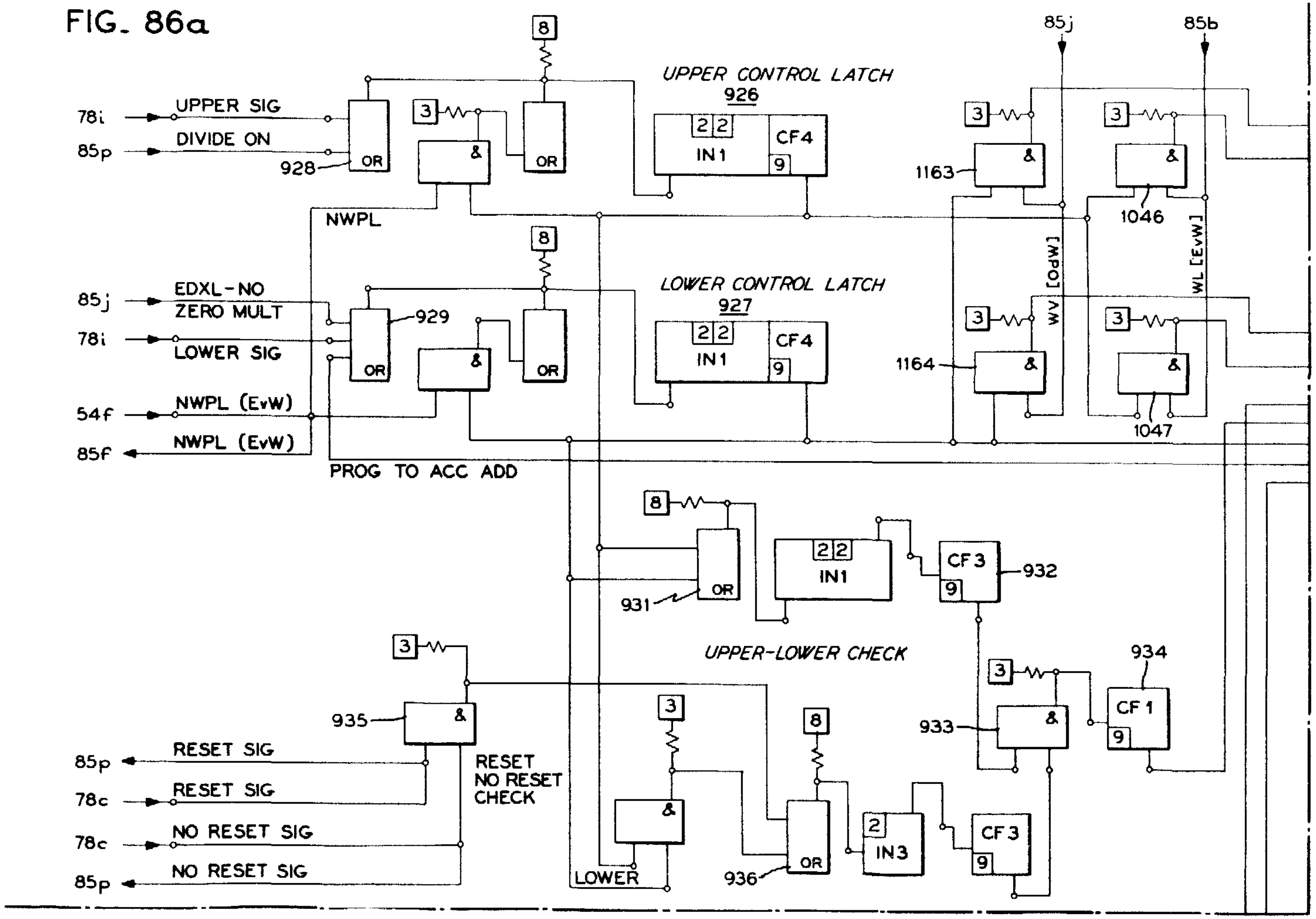
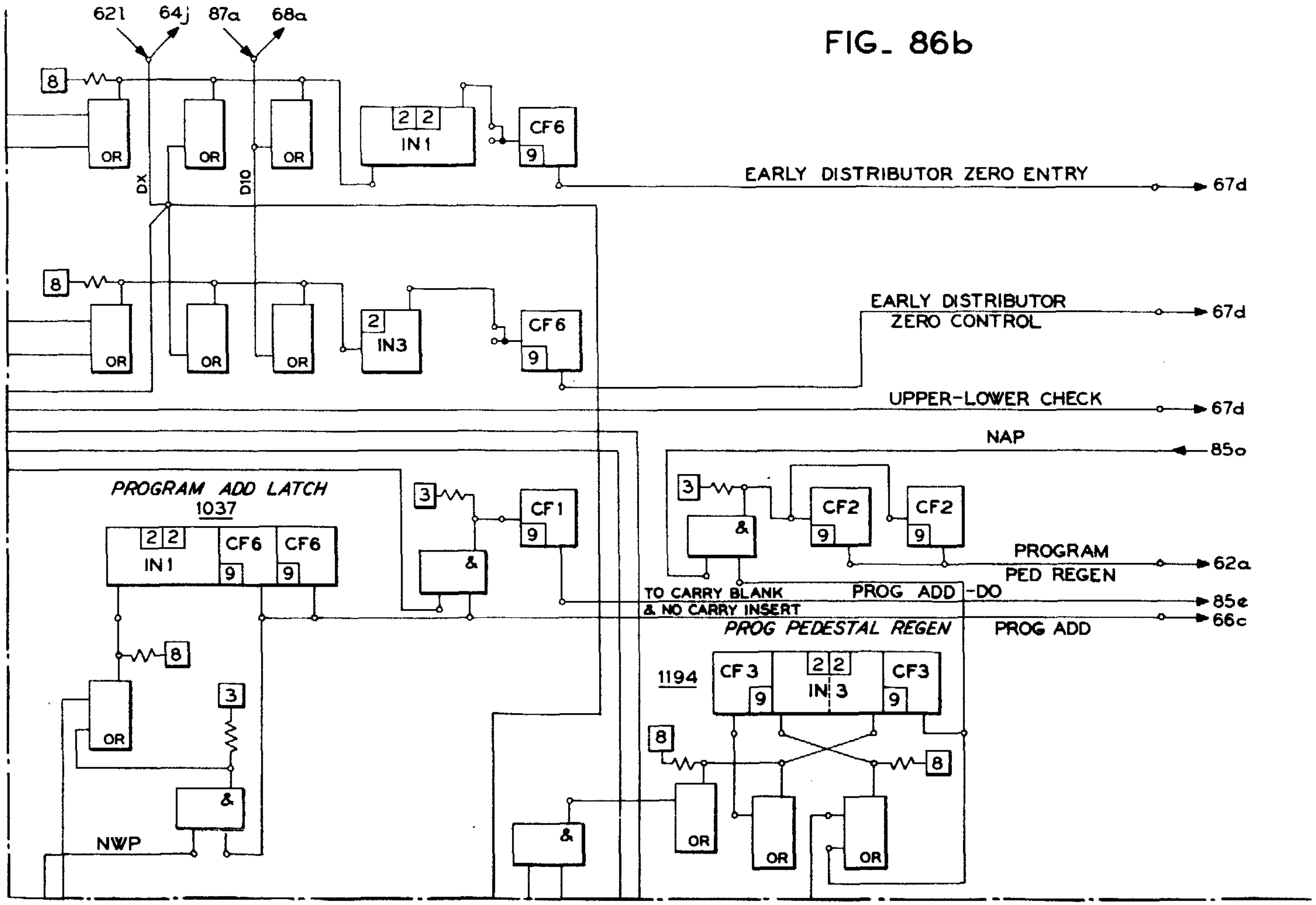

UPPER CONTROL LATCH
926
LOWER CONTROL LATCH
927
UPPER-LOWER CHECK
78i
UPPER SIG
85p
DIVIDE ON
928
OR
8
3
&
OR
2 2
IN1
CF4
9
NWPL
85j
EDXL-NO
ZERO MULT
78i
LOWER SIG
929
54f
NWPL (EvW)
85f
NWPL (EvW)
PROG TO ACC ADD
85j
85b
1163
1046
1164
1047
WV [OPW]
WL [EvW]
931
CF3
932
934
CF 1
933
935
RESET SIG
85p
78c
RESET SIG
78c
NO RESET SIG
85p
NO RESET SIG
RESET
NO RESET
CHECK
LOWER
936
IN3
2

FIG. 86b


62l
64j
87a
68a
8
OR
OR
OR
DX
D10
2 2
IN 1
CF6
9
EARLY DISTRIBUTOR ZERO ENTRY
67d
8
OR
OR
OR
2
IN3
CF6
9
EARLY DISTRIBUTOR
ZERO CONTROL
67d
UPPER-LOWER CHECK
67d
NAP
85o
PROGRAM ADD LATCH
1037
2 2
IN 1
CF6
9
CF6
9
3
CF1
9
&
3
&
CF2
9
CF2
9
PROGRAM
62a
PED REGEN
TO CARRY BLANK
& NO CARRY INSERT
PROG ADD -DO
85e
66c
PROG PEDESTAL REGEN
PROG ADD
1194
CF 3
9
2 2
IN 3
CF3
9
8
OR
3
&
NWP
8
8
&
OR
OR
OR

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

Filed Nov. 2, 1955

DATA STORAGE AND PROCESSING MACHINE

228 Sheets-Sheet 210

FIG. 86c

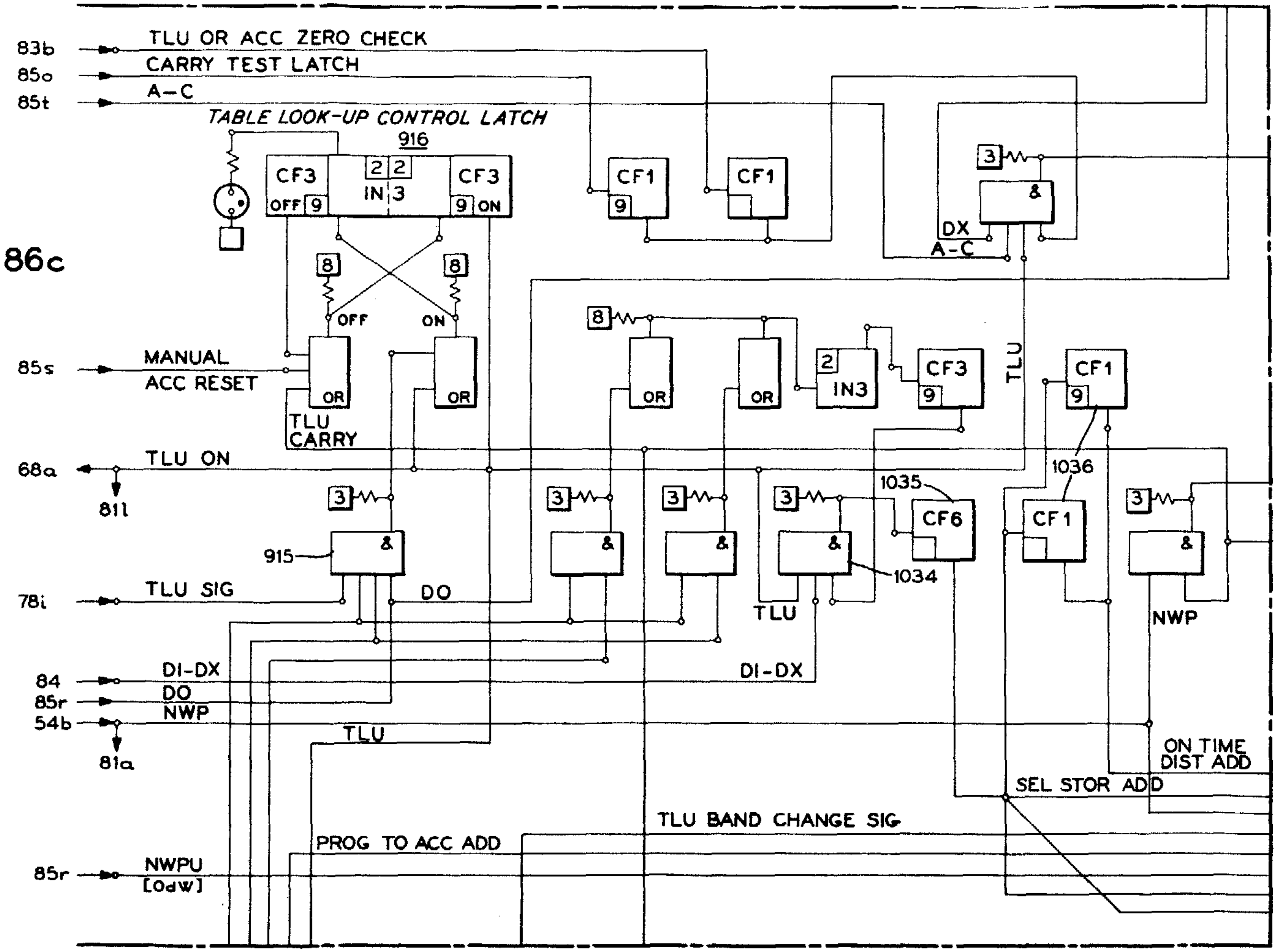


FIG. 86d

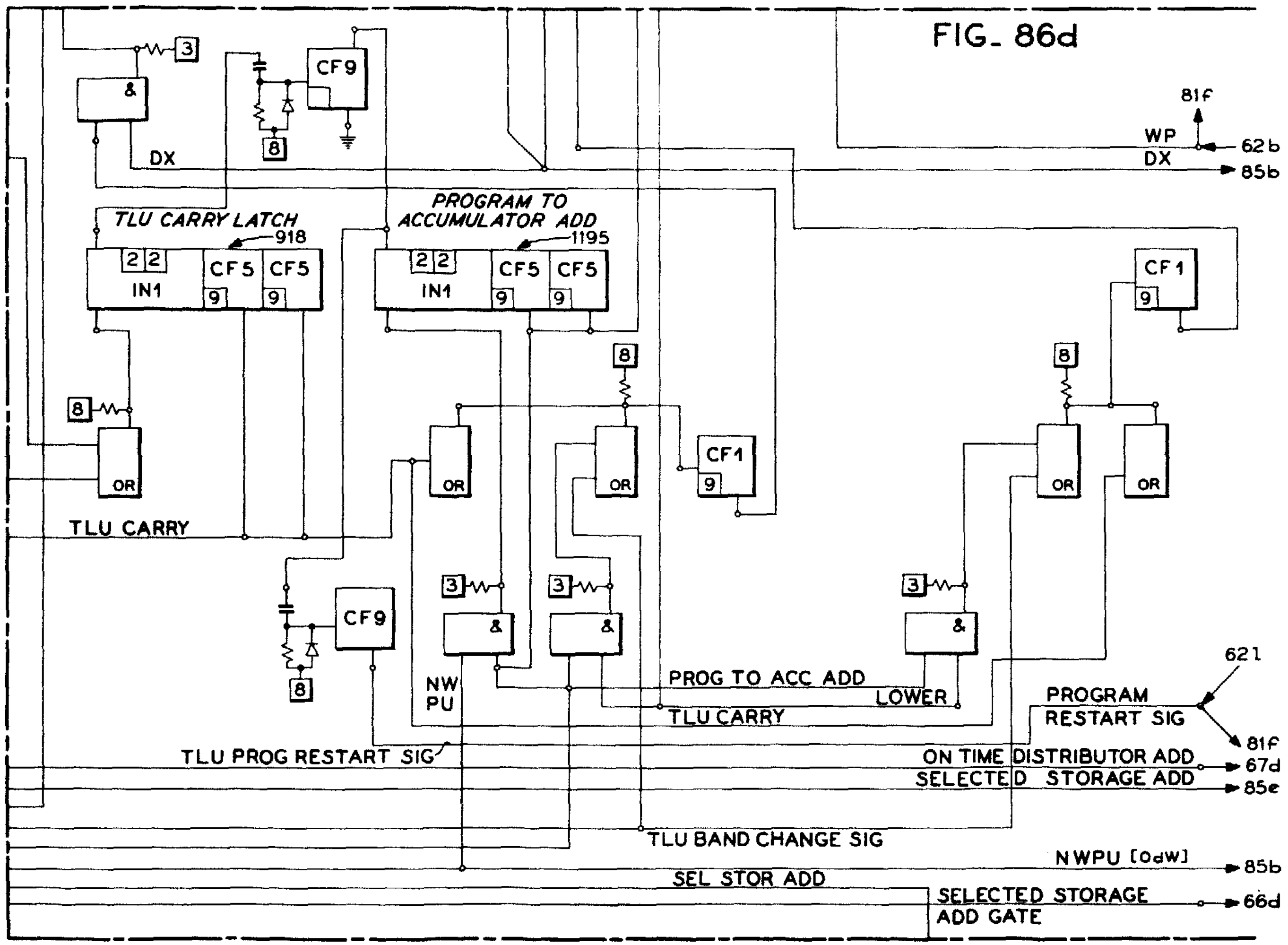
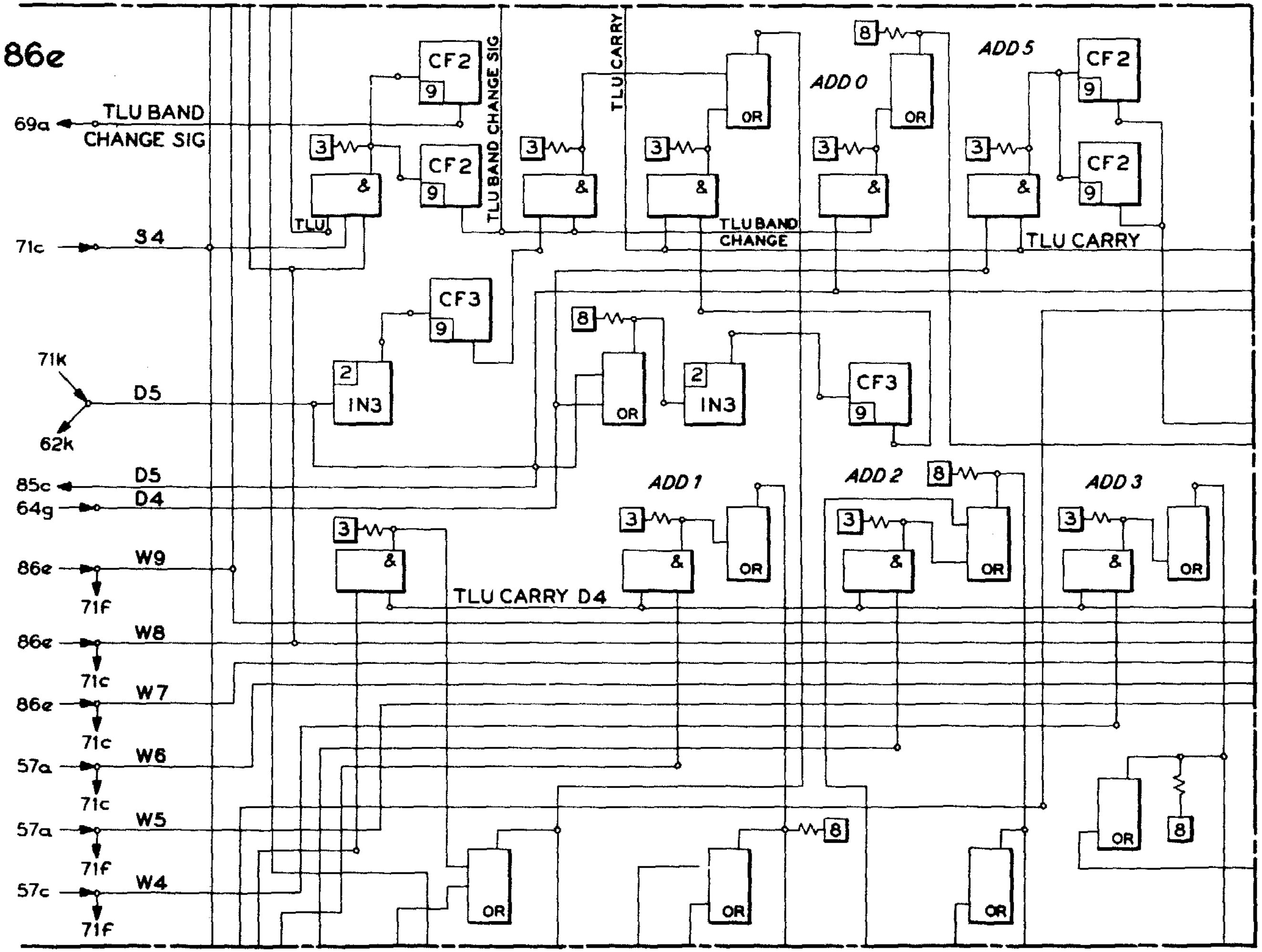

3
&
DX
CF9
8
81f
WP
DX
62b
85b
TLU CARRY LATCH
918
2 2
IN1
CF5
CF5
9
9
PROGRAM TO
ACCUMULATOR ADD
1195
2 2
IN1
CF5
CF5
9
9
CF1
9
8
8
8
OR
OR
OR
CF1
9
OR
OR
TLU CARRY
CF9
8
3
3
3
&
&
&
NW
PU
PROG TO ACC ADD
LOWER
TLU CARRY
621
PROGRAM
RESTART SIG
81f
TLU PROG RESTART SIG
ON TIME DISTRIBUTOR ADD
67d
SELECTED STORAGE ADD
85e
TLU BAND CHANGE SIG
NWPU [OdW]
85b
SEL STOR ADD
SELECTED STORAGE
ADD GATE
66d

FIG. 86e


69a
TLU BAND
CHANGE SIG
71c
S4
71k
D5
62k
85c
D5
64g
D4
86e
W9
71f
86e
W8
71c
86e
W7
71c
57a
W6
71c
57a
W5
71f
57c
W4
71f
CF2
9
3
&
TLU
CF2
9
TLU BAND CHANGE SIG
CF3
9
2
IN3
3
&
TLU CARRY D4
3
&
TLU CARRY
3
&
8
OR
OR
ADD 0
OR
3
&
TLU BAND
CHANGE
2
IN3
CF3
9
ADD 1
3
&
OR
8
OR
3
&
8
OR
ADD 2
8
3
&
OR
8
OR
ADD 5
CF2
9
CF2
9
3
&
TLU CARRY
ADD 3
3
&
OR
OR
8
OR
OR

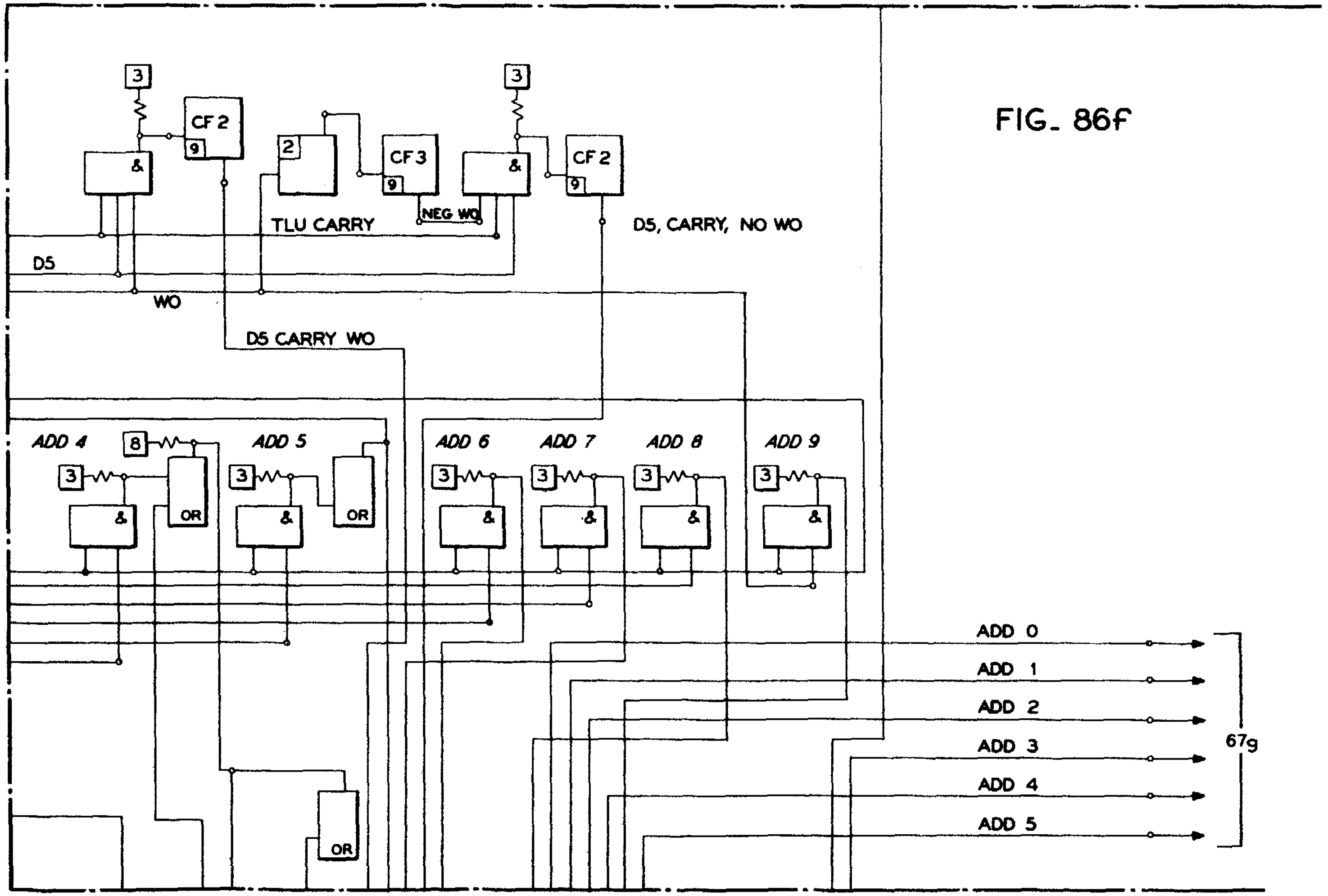

3
CF 2
9
&
2
CF 3
9
3
&
CF 2
9
TLU CARRY
NEG WO
D5, CARRY, NO WO
D5
WO
D5 CARRY WO
ADD 4
8
3
&
OR
ADD 5
3
&
OR
ADD 6
3
&
ADD 7
3
&
ADD 8
3
&
ADD 9
3
&
OR
ADD 0
ADD 1
ADD 2
ADD 3
ADD 4
ADD 5
67g

FIG. 86f

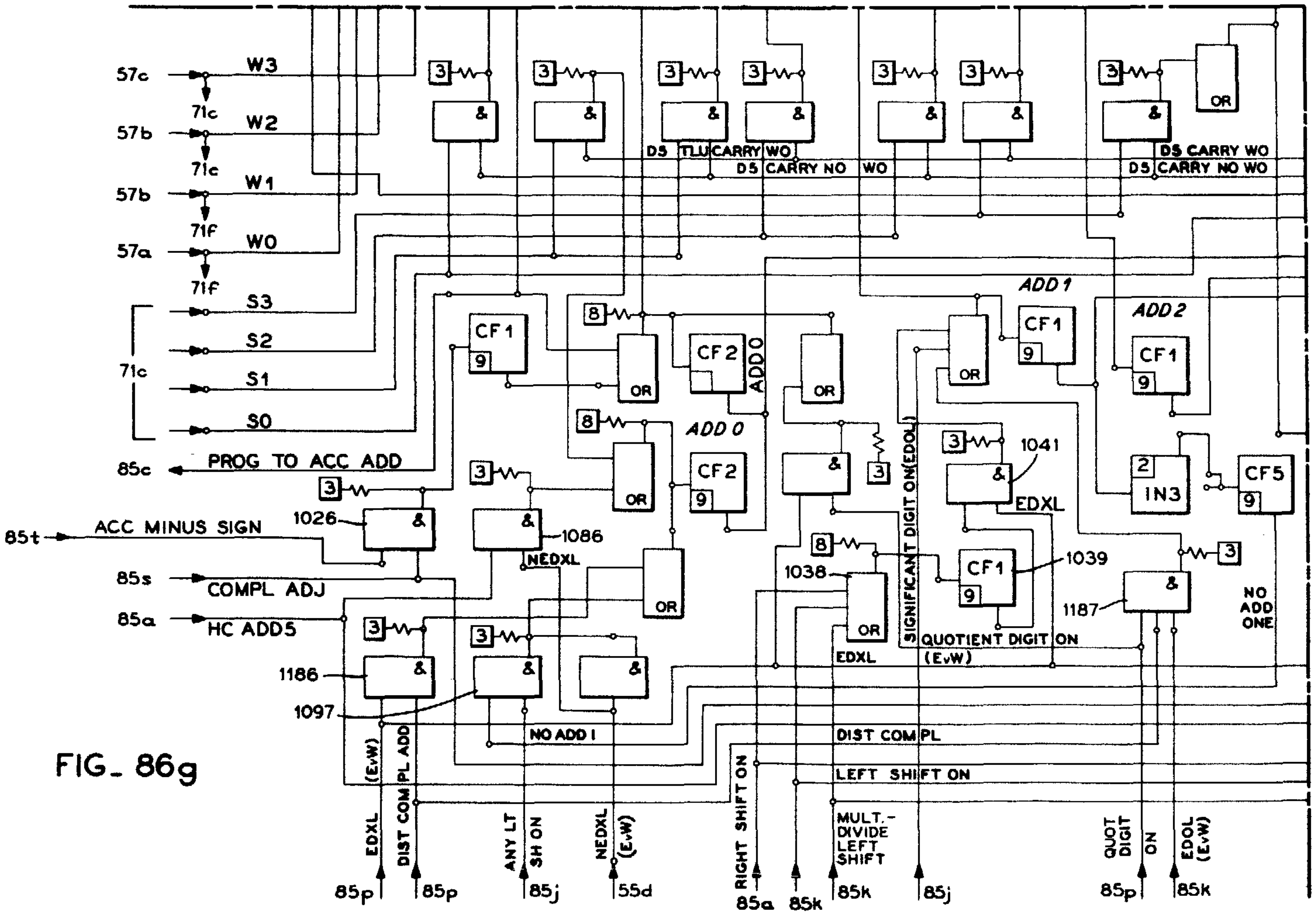

57c
57b
57b
57a
W3
W2
W1
W0
71c
71c
71f
71f
S3
S2
S1
S0
71c
85c
PROG TO ACC ADD
85t
ACC MINUS SIGN
85s
COMPL ADJ
85a
HC ADD5
1026
1086
NEDXL
1186
1097
NO ADD 1
CF 1
CF 2
CF 2
ADD 0
ADDO
OR
&
3
8
D5 TLU CARRY WO
D5 CARRY NO WO
D5 CARRY WO
D5 CARRY NO WO
ADD 1
ADD 2
CF 1
CF 1
IN3
CF5
1041
EDXL
1038
CF 1
1039
1187
NO ADD ONE
SIGNIFICANT DIGIT ON (EDOL)
QUOTIENT DIGIT ON
(EvW)
EDXL
DIST COMPL
LEFT SHIFT ON
(EvW)
DIST COMPL ADD
EDXL
ANY LT SH ON
NEDXL (EvW)
RIGHT SHIFT ON
MULT.-DIVIDE LEFT SHIFT
QUOT DIGIT ON
EDOL (EvW)
85p
85p
85j
55d
85a
85k
85k
85j
85p
85k

FIG_ 86g

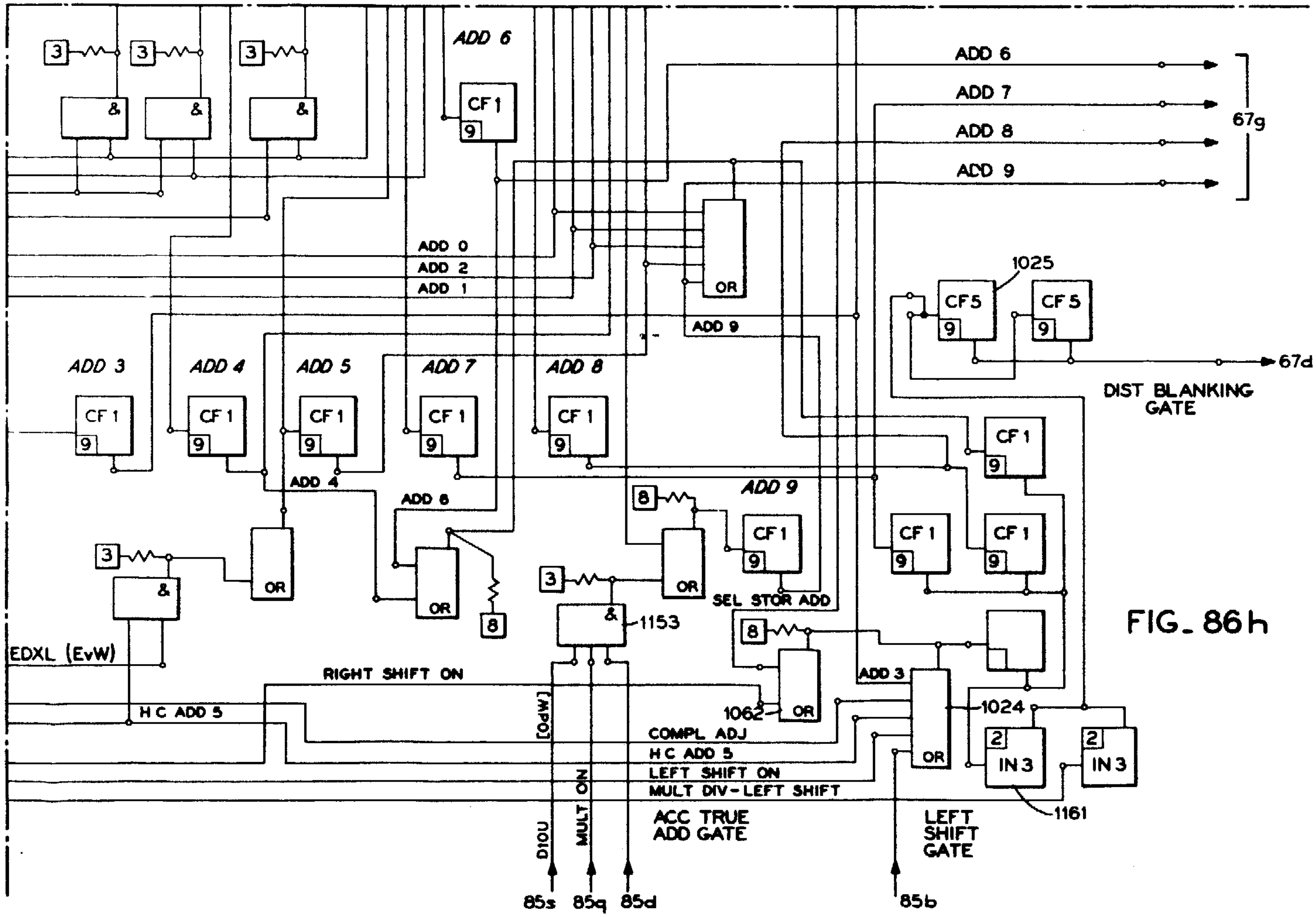

3
3
3
&
&
&
ADD 6
CF 1
9
ADD 6
ADD 7
ADD 8
ADD 9
67g
ADD 0
ADD 2
ADD 1
OR
ADD 9
1025
CF5
9
CF 5
9
67d
DIST BLANKING GATE
ADD 3
ADD 4
ADD 5
ADD 7
ADD 8
CF 1
9
ADD 4
ADD 6
ADD 9
8
3
OR
OR
8
3
&
1153
OR
SEL STOR ADD
8
1062
OR
ADD 3
OR
1024
2
IN 3
2
IN 3
1161
EDXL (EvW)
H C ADD 5
RIGHT SHIFT ON
[MPO]
COMPL ADJ
H C ADD 5
LEFT SHIFT ON
MULT DIV-LEFT SHIFT
DIOU
MULT ON
ACC TRUE ADD GATE
LEFT SHIFT GATE
85s
85q
85d
85b

FIG_ 86h

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 216

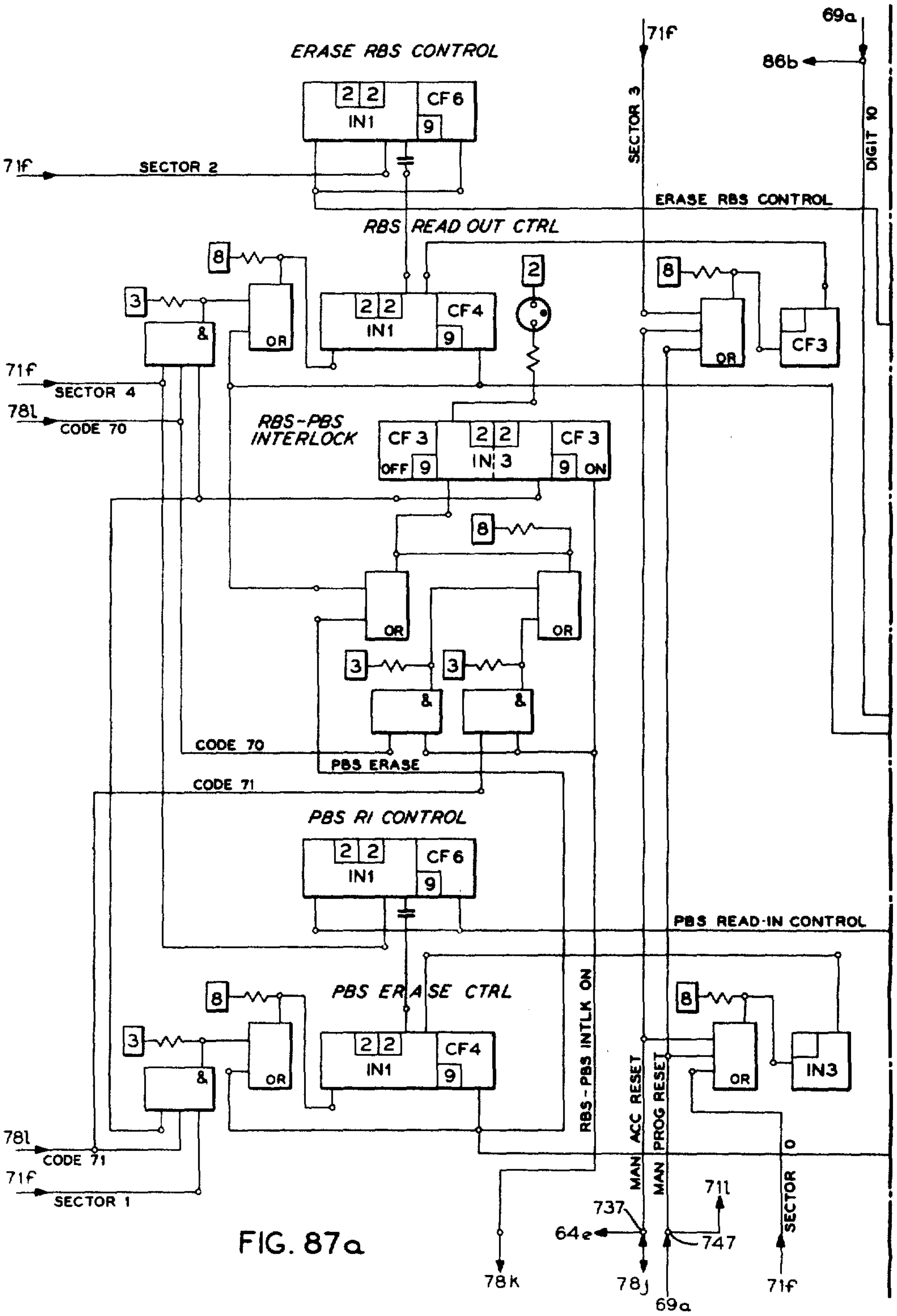


FIG. 87a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 217

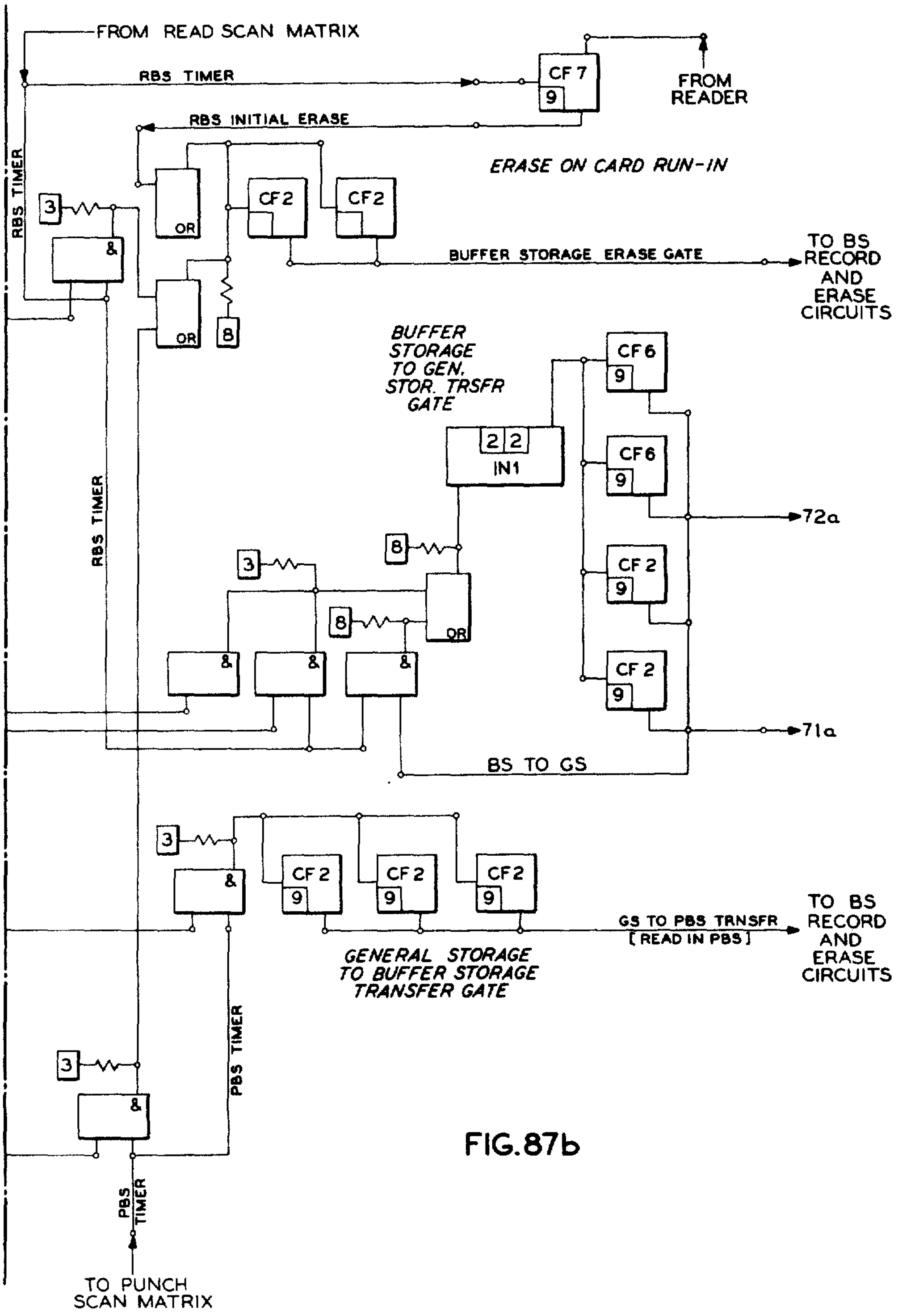


FIG. 87b

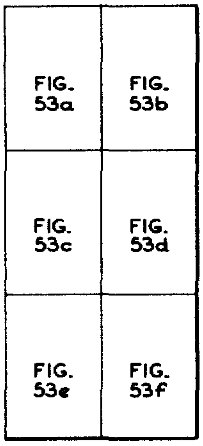


FIG. 88

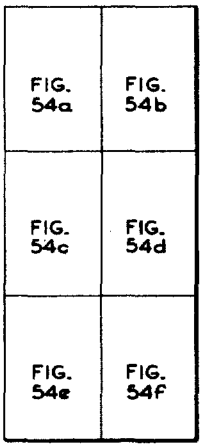


FIG. 89

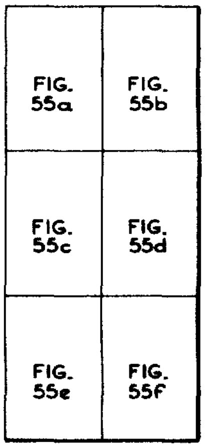


FIG. 90

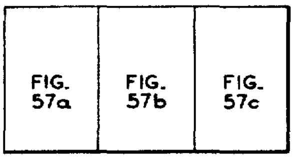


FIG. 92

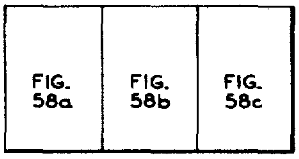


FIG. 93

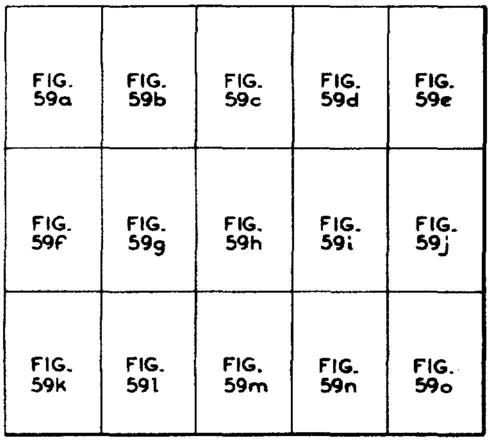


FIG. 94

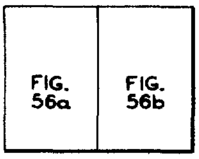


FIG. 91

FIG. 61a	FIG. 61b	FIG. 61c	FIG. 61d
FIG. 61e	FIG. 61f	FIG. 61g	FIG. 61h
FIG. 61i	FIG. 61j	FIG. 61k	FIG. 61l

FIG. 95

FIG. 62a	FIG. 62b	FIG. 62c	FIG. 62d
FIG. 62e	FIG. 62f	FIG. 62g	FIG. 62h
FIG. 62i	FIG. 62j	FIG. 62k	FIG. 62l

FIG. 96

FIG. 64a	FIG. 64b	FIG. 64c	FIG. 64d	FIG. 64e
FIG. 64f	FIG. 64g	FIG. 64h	FIG. 64i	FIG. 64j

FIG. 97

FIG. 66a	FIG. 66b
FIG. 66c	FIG. 66d

FIG. 98

FIG. 67a	FIG. 67b	FIG. 67c	FIG. 67d
	FIG. 67e	FIG. 67f	FIG. 67g

FIG. 99

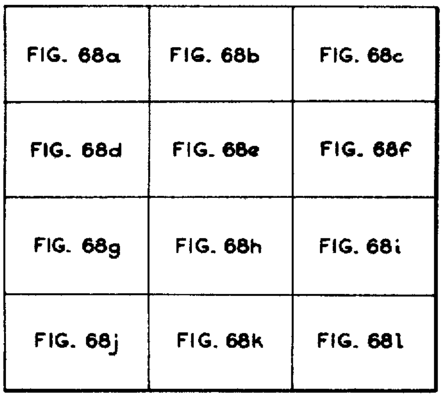


FIG. 68a	FIG. 68b	FIG. 68c
FIG. 68d	FIG. 68e	FIG. 68f
FIG. 68g	FIG. 68h	FIG. 68i
FIG. 68j	FIG. 68k	FIG. 68l

FIG. 100

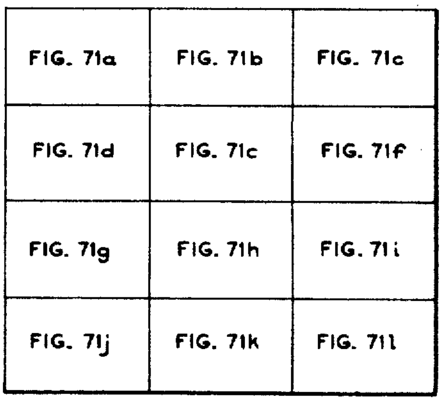


FIG. 71a	FIG. 71b	FIG. 71c
FIG. 71d	FIG. 71c	FIG. 71f
FIG. 71g	FIG. 71h	FIG. 71i
FIG. 71j	FIG. 71k	FIG. 71l

FIG. 102

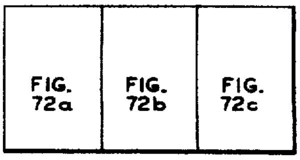


FIG. 72a	FIG. 72b	FIG. 72c

FIG. 103

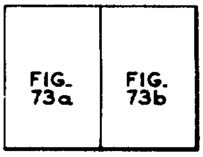


FIG. 73a	FIG. 73b

FIG. 104

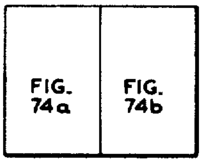


FIG. 74a	FIG. 74b

FIG. 105

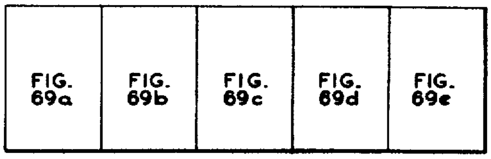


FIG. 69a	FIG. 69b	FIG. 69c	FIG. 69d	FIG. 69e

FIG. 101

FIG. 75a	FIG. 75b	FIG. 75c
FIG. 75d	FIG. 75e	FIG. 75f

FIG. 106

FIG. 76a	FIG. 76b

FIG. 107

FIG. 77a	FIG. 77b	FIG. 77c	FIG. 77d

FIG. 108

FIG. 78a	FIG. 78b	FIG. 78c
FIG. 78d	FIG. 78e	FIG. 78f
FIG. 78g	FIG. 78h	FIG. 78i
FIG. 78j	FIG. 78k	FIG. 78l

FIG. 109

FIG. 79a
FIG. 78b

FIG. 110

FIG. 81a	FIG. 81b	FIG. 81c
FIG. 81d	FIG. 81e	FIG. 81f
FIG. 81g	FIG. 81h	FIG. 81i
FIG. 81j	FIG. 81k	FIG. 81l

FIG. 111

FIG. 82a	FIG. 82b	FIG. 82c

FIG. 112

FIG. 83a	FIG. 83b

FIG. 113

FIG. 85a	FIG. 85b	FIG. 85c	FIG. 85d	FIG. 85e
FIG. 85f	FIG. 85g	FIG. 85h	FIG. 85I	FIG. 85j
FIG. 85k	FIG. 85L	FIG. 85m	FIG. 85n	FIG. 85o
FIG. 85p	FIG. 85q	FIG. 85r	FIG. 85s	FIG. 85t

FIG. 114

FIG. 86a	FIG. 86b
FIG. 86c	FIG. 86d
FIG. 86e	FIG. 86f
FIG. 86g	FIG. 86h

FIG. 115

FIG. 87a	FIG. 87b

FIG. 116

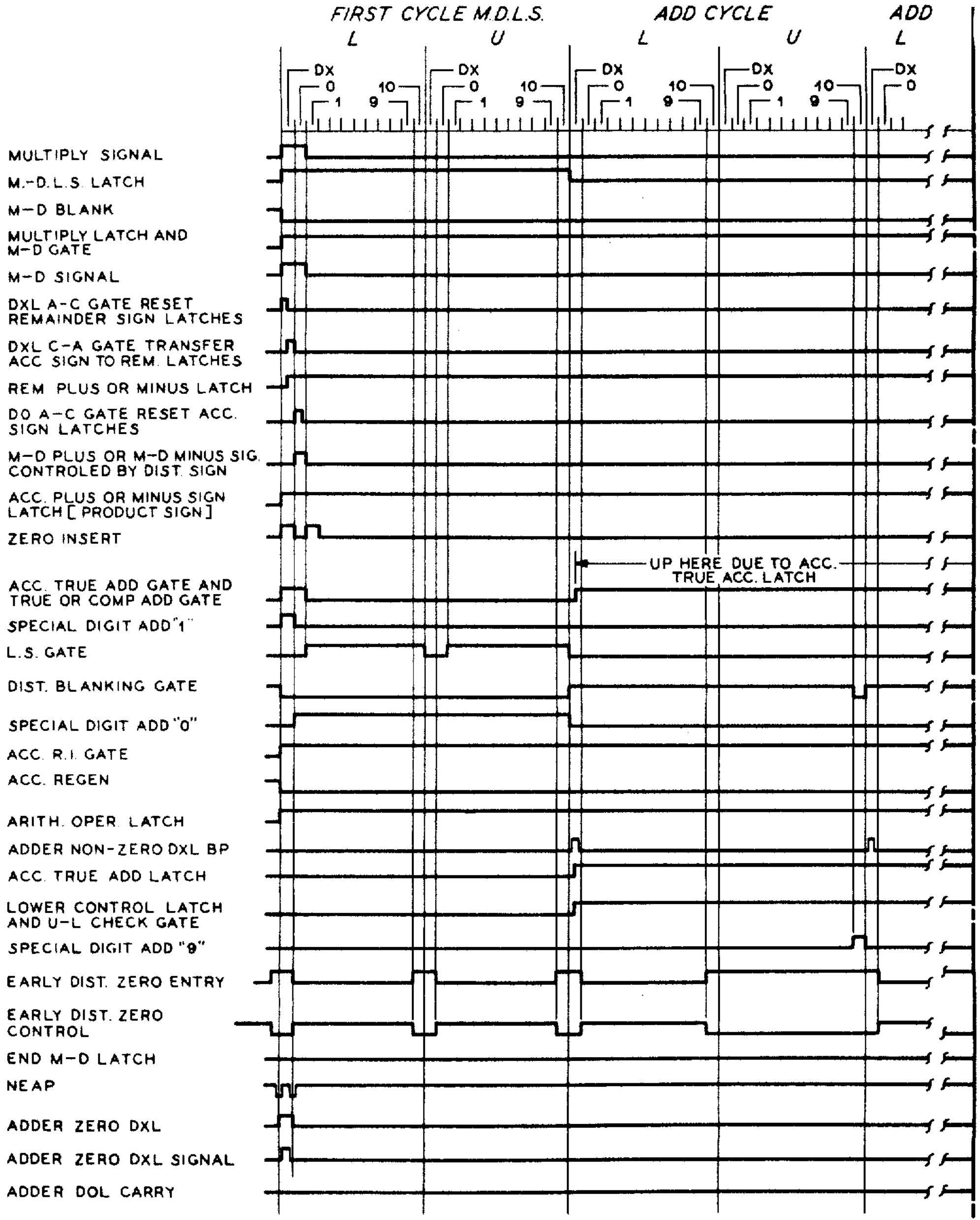

FIRST CYCLE M.D.L.S.
ADD CYCLE
ADD
L
U
L
U
L
DX
0
1
9
10
MULTIPLY SIGNAL
M.-D.L.S. LATCH
M-D BLANK
MULTIPLY LATCH AND M-D GATE
M-D SIGNAL
DXL A-C GATE RESET REMAINDER SIGN LATCHES
DXL C-A GATE TRANSFER ACC SIGN TO REM. LATCHES
REM PLUS OR MINUS LATCH
DO A-C GATE RESET ACC. SIGN LATCHES
M-D PLUS OR M-D MINUS SIG. CONTROLED BY DIST. SIGN
ACC. PLUS OR MINUS SIGN LATCH [PRODUCT SIGN]
ZERO INSERT
UP HERE DUE TO ACC. TRUE ACC. LATCH
ACC. TRUE ADD GATE AND TRUE OR COMP ADD GATE
SPECIAL DIGIT ADD "1"
L.S. GATE
DIST. BLANKING GATE
SPECIAL DIGIT ADD "0"
ACC. R.I. GATE
ACC. REGEN
ARITH. OPER. LATCH
ADDER NON-ZERO DXL BP
ACC. TRUE ADD LATCH
LOWER CONTROL LATCH AND U-L CHECK GATE
SPECIAL DIGIT ADD "9"
EARLY DIST. ZERO ENTRY
EARLY DIST. ZERO CONTROL
END M-D LATCH
NEAP
ADDER ZERO DXL
ADDER ZERO DXL SIGNAL
ADDER DOL CARRY

FIG_117a

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 224

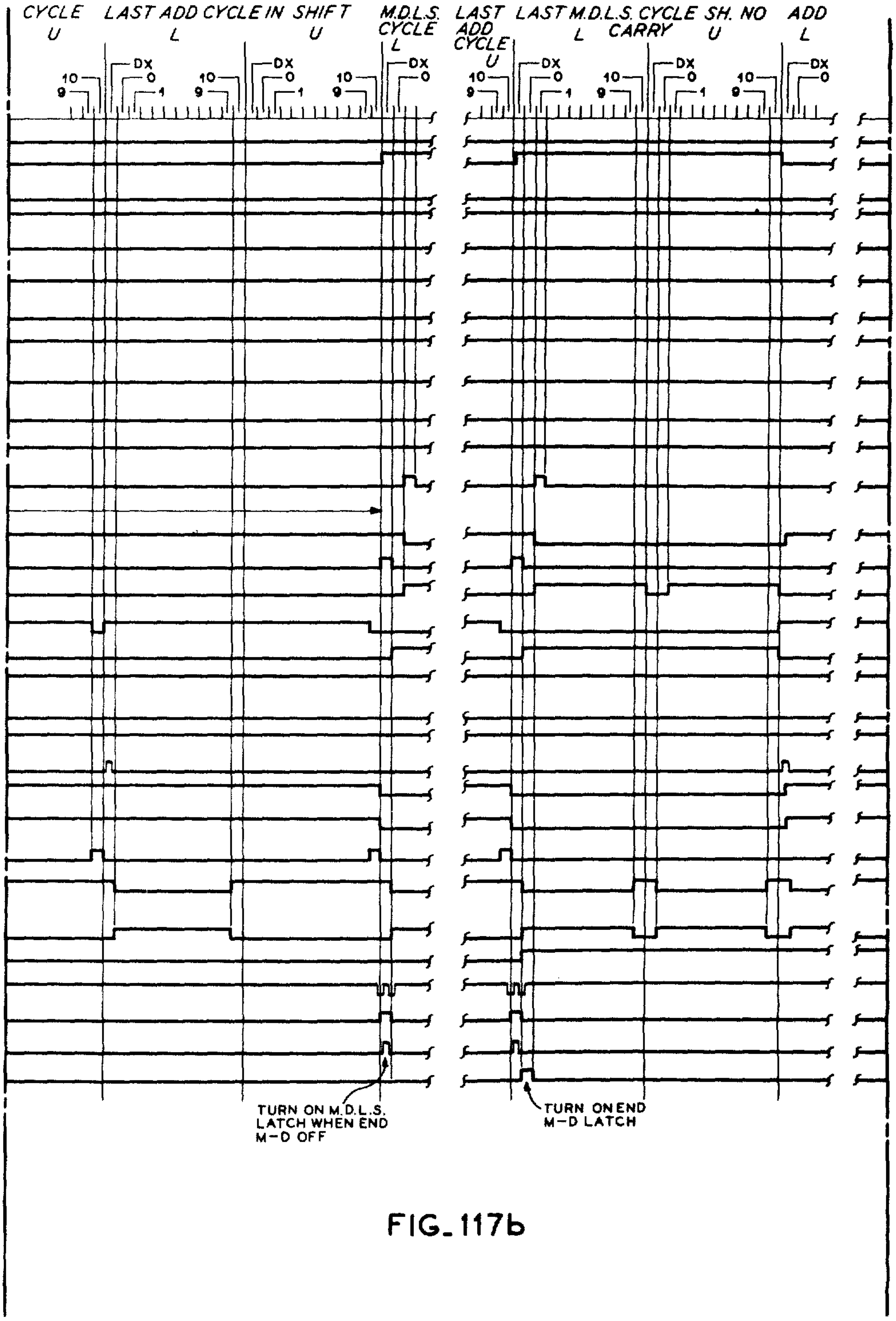


FIG_117b

Nov. 8, 1960 F. E. HAMILTON ET AL 2,959,351

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 225

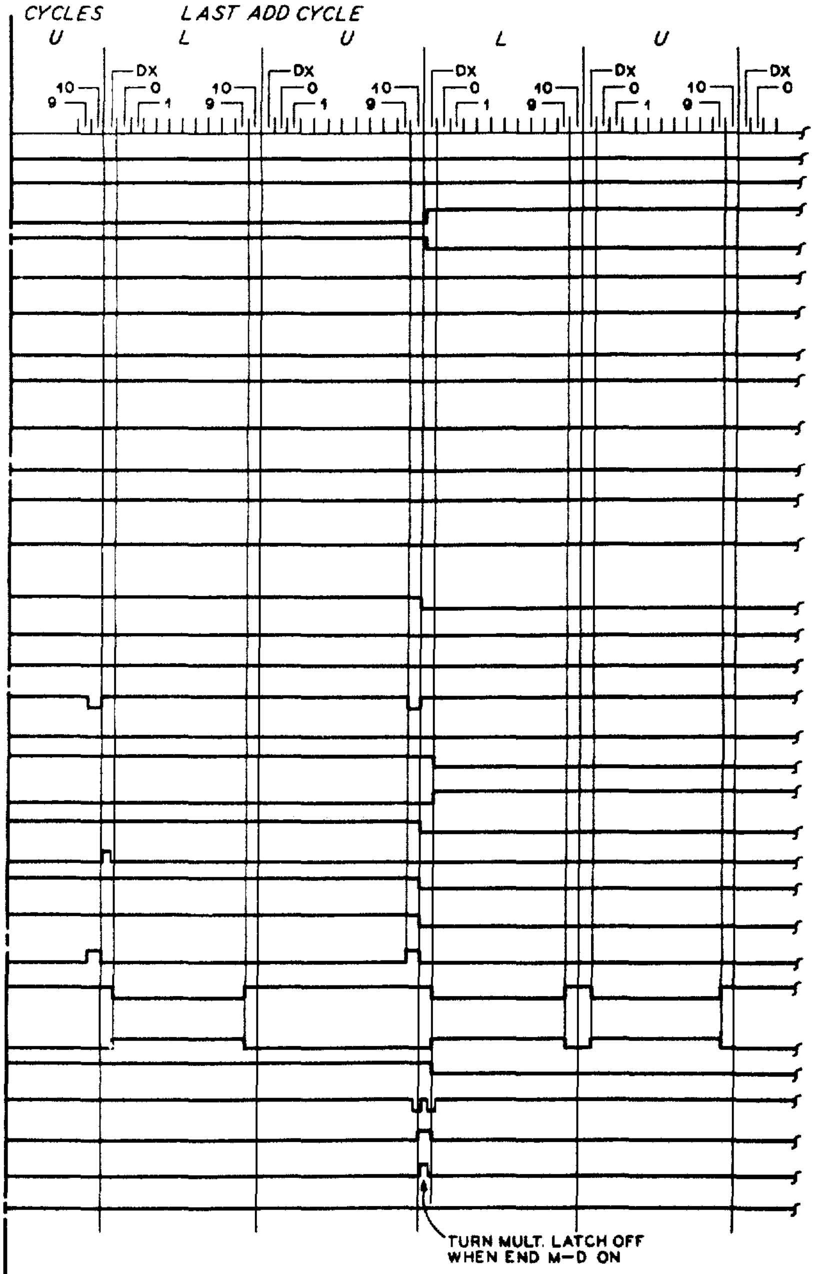


FIG_117c

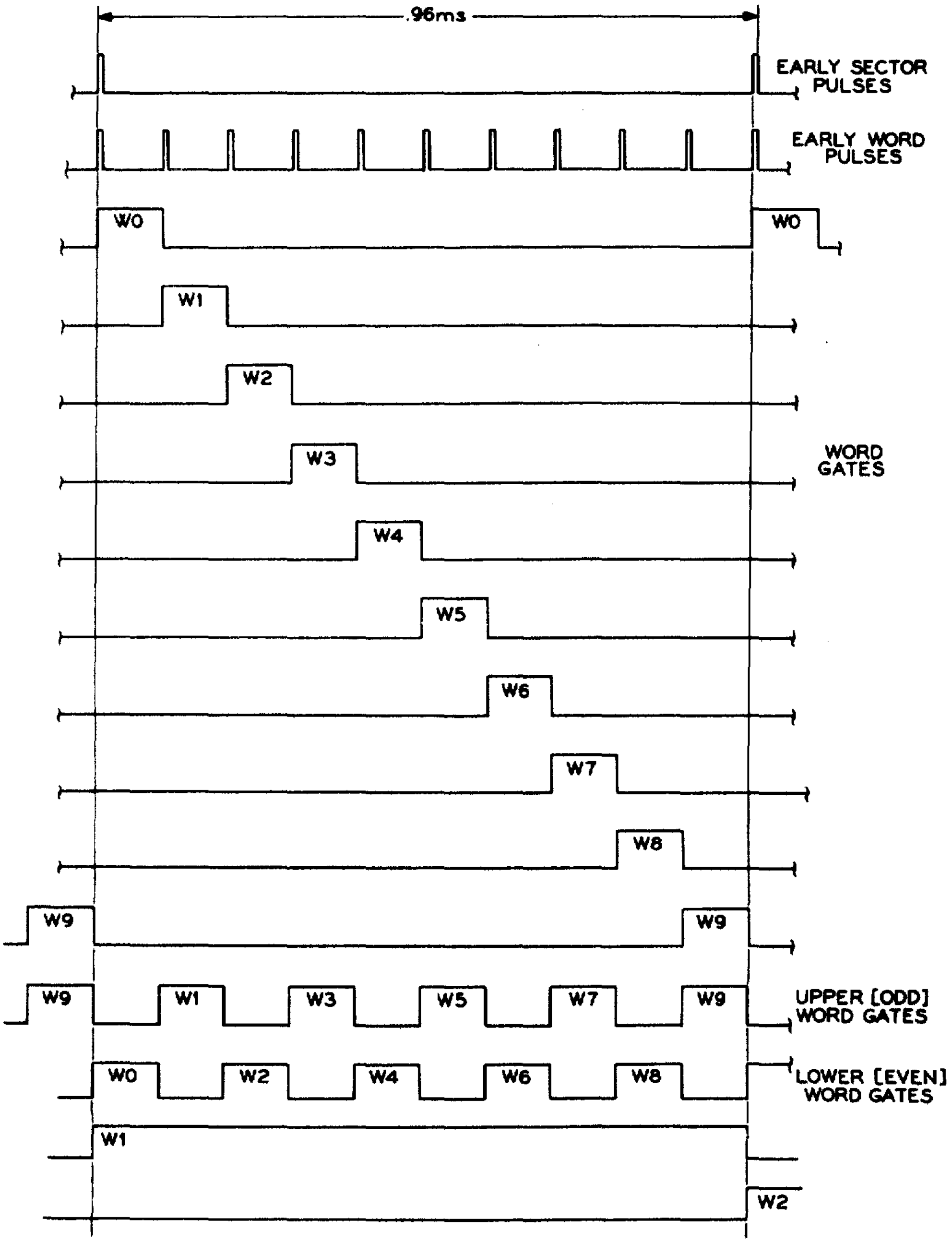

.96ms
EARLY SECTOR PULSES
EARLY WORD PULSES
W0
W1
W2
W3
W4
W5
W6
W7
W8
W9
W0
W9
WORD GATES
W9
W1
W3
W5
W7
W9
UPPER [ODD] WORD GATES
W0
W2
W4
W6
W8
LOWER [EVEN] WORD GATES
W1
W2

FIG. 118

DATA STORAGE AND PROCESSING MACHINE

Filed Nov. 2, 1955 228 Sheets-Sheet 227

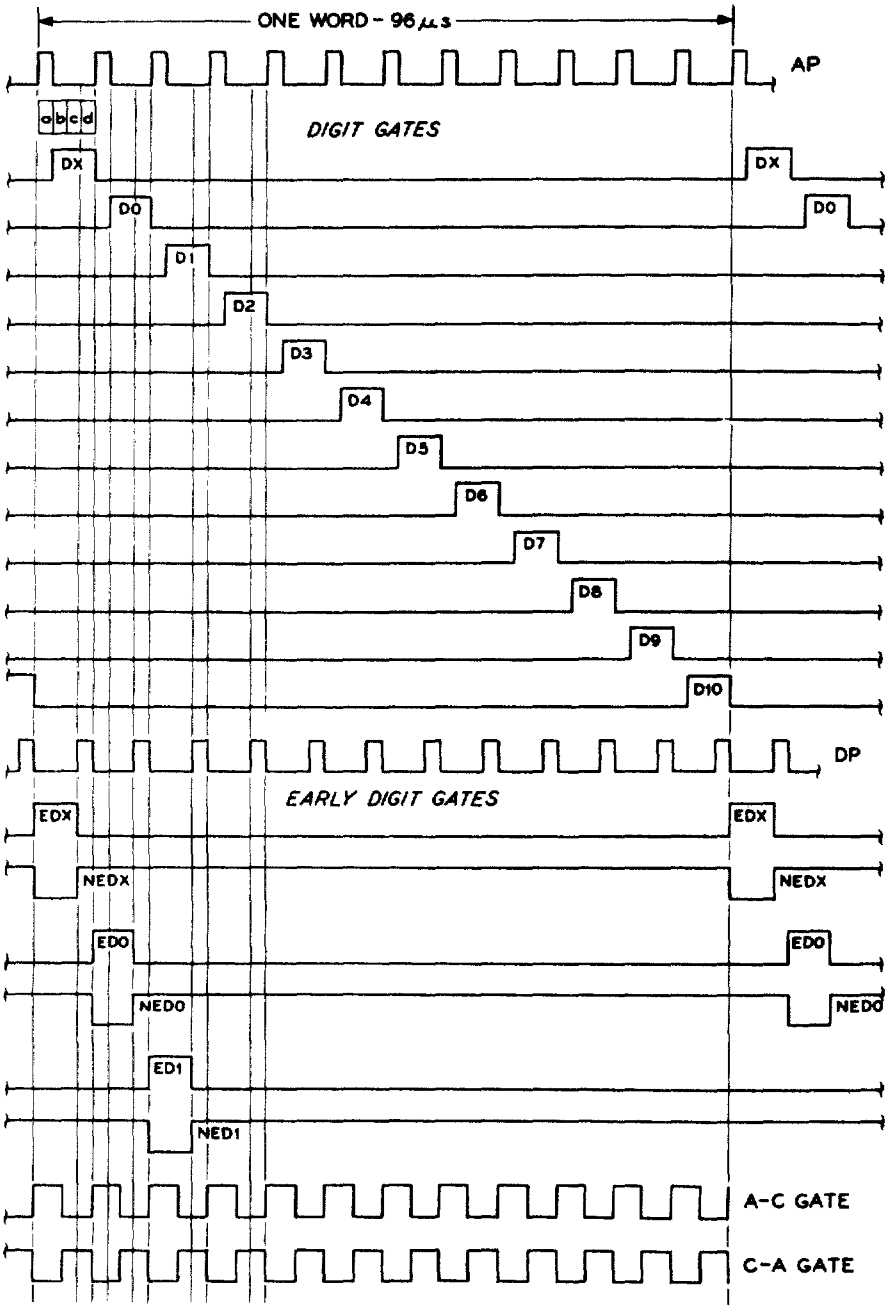


FIG. 119

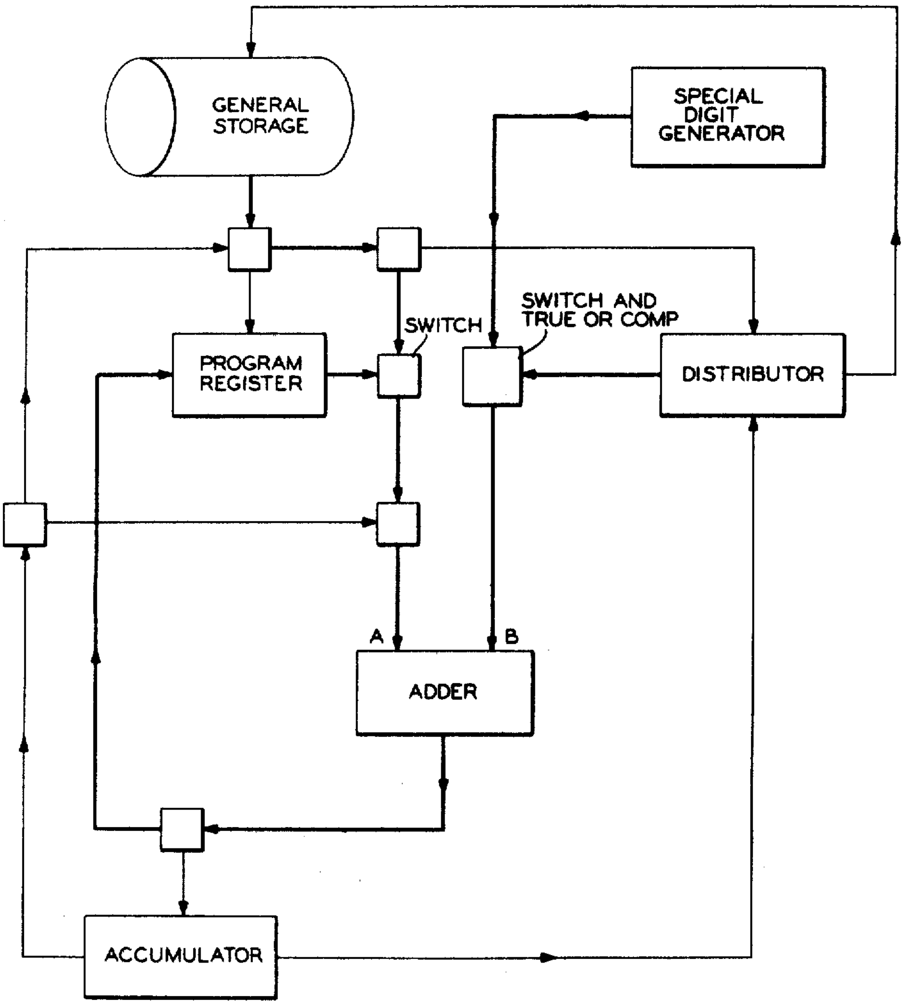

GENERAL STORAGE
SPECIAL DIGIT GENERATOR
SWITCH
SWITCH AND TRUE OR COMP
PROGRAM REGISTER
DISTRIBUTOR
A
B
ADDER
ACCUMULATOR

FIG. 120

United States Patent Office

2,959,351
Patented Nov. 8, 1960

2,959,351

DATA STORAGE AND PROCESSING MACHINE

Francis E. Hamilton, Binghamton, and Ernest S. Hughes, Jr., and Warren K. Lind, Vestal, N.Y., assignors to International Business Machines Corporation, New York, N.Y., a corporation of New York

Filed Nov. 2, 1955, Ser. No. 544,520

46 Claims. (Cl. 235—153)

This invention relates to high speed data storage and processing systems and particularly to stored program type data processing machines. This application is a continuation-in-part of our application Serial No. 398,922, filed December 18, 1953, now abandoned.

A frequently used system of classifying high speed digital data storage and processing machines is according to the type of programming used. A stored program type machine is one in which the sequence of steps of data processing or the sequences of functions of the machine is under the control of instructions contained in some type of storage device or devices within the machine. Heretofore machines of this type have been built having high programming capacity, high storage capacity, and a high degree of flexibility. These machines have been large in size and expensive in cost thus making them impractical for many applications.

The solution of many problems on smaller and less expensive machines has not been practical in the past because of the complexity of the problem. Some of these problems have been solved by multiple passes through the smaller machines, some by involved rearrangements of the data, and others by special setup of the machine requiring extensive time.

It is a prime object of this invention to provide a machine having large memory capacity, high reliability, ease of operation, and high programming capacity, while being compact in design and moderate in cost.

Another object of this invention is to provide new and improved methods of operation and apparatus for such a machine.

A further object is to provide an improved data processing machine adaptable to both scientific and accounting applications.

A still further object is to provide a data processing machine of improved efficiency in performing arithmetic and logical operations.

Another object is to provide an improved data processing machine capable of performing arithmetic operations on instructions as well as data.

Still another object is to provide an improved data processing machine capable of altering the sequence of operations as the program continues.

Still another object is to provide an improved stored program type data processing machine in which program data may be stored in any memory location.

Still another object is to provide a simplified stored program type data processing machine of high flexibility.

Still another object is to provide an improved method and apparatus for searching a table in a data processing machine.

Still another object is to provide a data processing machine of improved flexibility in handling tabular data.

Still another object is to provide a simplified stored program type data processing machine capable of handling a multiplicity of tables during the course of a calculation.

A further object is to provide an improved method and apparatus for magnetic drum track selection.

An additional object is to provide a data processing machine with an improved method and apparatus for address selection.

Still another object is to provide a data processing machine with improved flexibility in entering tabular information into a sequential calculation.

Another object is to provide an improved stored program type data processing machine wherein a program sequence may be altered by any of a number of conditions within the machine.

Another object is to provide an improved stored program type data processing machine wherein a bi-stable circuit alternately enables a pair of data transmission channels and signals from an arithmetic unit may control said bi-stable circuit.

Another object is to provide an improved stored program type data processing machine wherein a bi-stable circuit alternately enables a pair of data transmission channels and a test of a condition within said machine may control said bi-stable circuit.

Another object is to provide an improved stored program type data processing machine wherein programming is simplified.

Another object is to provide an improved stored program type data processing machine capable of detecting the omission of a program instruction.

Another object is to provide an improved stored program type data processing machine wherein the machine is stopped by the omission of a program instruction.

Another object is to provide an improved stored program type data processing machine wherein the machine may be stopped at any predetermined point in the program sequence.

Another object is to provide a data processing machine of improved ability to detect errors.

Another object is to provide an improved stored program type data processing machine wherein the detection of an error causes the program sequence to be interrupted.

Another object is to provide an improved stored program type data processing machine wherein the detection of an error causes a different program sequence to be initiated.

Another object is to provide an improved stored program type data processing machine wherein the detection of an error causes the machine to repeat a portion of the program sequence.

Another object is to provide a stored program type data processing machine of improved ability to check the validity of a transmitted code combination.

Another object is to provide an improved stored program type data processing machine wherein a "self-checking" code is employed to check the validity of another code combination.

Another object is to provide an improved stored program type data processing machine wherein a branching operation may be initiated upon the occurrence of a specified condition within the machine.

Another object is to provide an improved stored program type data processing machine wherein any of a number of conditions within the machine may stop the sequence of program steps.

Another object is to provide an improved stored program type data processing machine wherein any of a number of conditions within the machine may cause the machine to repeat a predetermined group of program steps.

Another object is to provide an improved stored program type data processing machine wherein a single group of program steps may be progressed through at each of a plurality of points in a sequence of other program steps.

Another object is to provide an improved stored program type data processing machine wherein a condition within the machine may alter a sequence of program steps to include a group of other program steps.

Another object is to provide an improved stored program type data processing machine wherein a test of the arithmetic unit determines a comparison between two numbers.

Another object is to provide an improved stored program type data processing machine wherein an adder serves conjointly as a component of the arithmetic unit and as a comparator.

Another object is to provide an improved stored program type data processing machine wherein the operation determined by one program step may be executed simultaneously with a portion of the operation determined by the preceding program step.

Another object is to provide an improved stored program type data processing machine wherein a plurality of operations may be simultaneously performed without the possibility of interference between the plurality of operations.

Another object is to provide an improved stored program type data processing machine under control of a program control register wherein the program register may serve as the source of one of two numbers to be added and as the sum register.

Another object is to provide an improved stored program type data processing machine capable of searching a stored table in which the arguments and functions are stored together.

Another object is to provide an improved stored program type data processing machine capable of searching a stored table of words of data in which a word is made up of an argument and its function or functions.

Another object is to provide an improved stored program type data processing machine wherein a first control register determines the function of a second control register.

Another object is to provide an improved stored program type data processing machine wherein an operation code control register determines whether a data code control register selects an address or controls an arithmetic operation.

Another object is to provide an improved stored program type data processing machine wherein an operation code control register determines whether a data code control register selects an address or controls the number of shifts to be performed.

Another object is to provide an improved stored program type data processing machine wherein data to be operated upon and program instructions are alternately selected.

Another object is to provide an improved stored program type data processing machine wherein a bi-stable circuit determines whether data to be operated upon or a program instruction is selected.

Another object is to provide an improved stored program type data processing machine wherein a bi-stable circuit controls the progression of data selection and signals from an arithmetic unit control said bi-stable circuit.

Another object is to provide an improved stored program type data processing machine wherein a bi-stable circuit alternately enables a pair of data transmission channels.

According to the embodiment of the invention disclosed herein a data processing machine is provided with a magnetic drum for storing a large quantity of data as magnetized spots on its surface. A program storage device is provided for storing a single program step or word. The program word is divided into three portions; an address portion for instructing the machine where data to be processed is located in storage, an operation portion for instructing the machine what operation or process the machine is to perform with the data found at the address of the address portion, and an instruction portion for instructing the machine where the next program step is located in storage. An address register and an operation register are provided for receiving the address portion and the operation portion respectively from the program storage device. Switching circuitry is provided under the control of the address register for selecting any storage position on the drum or any other storage device in the machine in accordance with the value stored in the address register. Switching circuitry is also provided under the control of the operation register for determining the operation the machine is to perform on the data found at a selected address position. After an address is selected and the data found at the address is operated upon by the machine the instruction portion of the program value is entered into the address register from program storage to replace the value previously in the register. A new program step located at the address in storage corresponding to the instruction portion of the program step in the address register is selected and transferred into the program storage device to replace the value previously stored therein. Alternatively a test may be made by the machine, and if a predetermined condition is found to exist, the address portion of a program step may be caused to remain in the address register and the next program step selected in accordance therewith and entered into the program storage device. Large numbers of program values and large amounts of other data may be stored on the magnetic drum. Thus, the sequence of the above-outlined procedure may automatically continue for a large number of program steps.

An accumulator, an adder, and a distributor are also provided in the machine as well as circuitry for introducing machine developed values into the adder.

The machine is capable of searching a table stored on the magnetic drum. This is accomplished by entering a program step having a table look-up code as the operation portion into the program storage device. The address portion of the program step is entered into the address register to select the address of the first argument of the table and the operation portion is entered into the operation register to instruct the machine to continue to search the table until an argument is found in the table equal to or higher than an argument stored in the distributor. Successive arguments of the table are selected and added to the complement of the argument in the distributor. If no argument equal to or higher than the argument in the distributor is found in the first track of the drum searched, the program step from the program storage device is entered into the adder and a machine developed value is added to the address portion to make the address portion correspond to the first address of the next track on the drum. The output of the adder is then entered into the program storage device and the new address portion is entered into the address register to replace the data previously stored therein. Successive arguments of the table continue to be added to the complement of the argument standing in the distributor until an equal or higher is found. Upon locating an equal or higher the program step from the program storage device may again be brought into the adder where the address portion has a value added thereto to make the address portion correspond to the address of the first argument found having a value equal to or higher than the value of the argument in the distributor. The new address portion is then entered into the accumulator where it is arranged with an instruction portion and an operation portion to give a new program step. This new program step may be transferred to any storage position in the machine. The functions of the arguments of the table may be stored on the drum at address positions removed from the address positions of the corresponding arguments by a constant number of address positions. This constant is added to the address portion of the new program step to give the address of the desired function of the argument.

Other objects of the invention will be pointed out in the following description and claims and illustrated in the accompanying drawings, which disclose, by way of examples, the principle of the invention and the best mode, which has been contemplated, of applying that principle.

In the drawings:

Fig. 1 is a general diagram of a stored program type data processing machine embodying the principles of the invention.

Fig. 2 shows how the ten decimal digits are represented in the bi-quinary code.

Fig. 3 shows how the ten decimal digits are represented in the two-out-of-five code.

Fig. 4 illustrates how the drum is arranged in sectors, words, digits and pulse times.

Fig. 5 illustrates the arrangement of timing tracks, buffer storages and address positions of general storage on the drum.

Figs. 6*a* and 6*b*, taken together as shown in Fig. 10, show the drum address arrangement.

Fig. 7 illustrates the components of a word of program instruction data.

Fig. 8 is a functional diagram of the control commutator.

Fig. 9 is a sequence chart for a shift right 3 operation.

Fig. 12 is a timing chart for adder operation left shift.

Figs. 13 through 52*a* show, schematically, the details of the basic circuit units of which the machine is comprised and the block symbols used to represent these basic units.

Figs. 53*a* through 53*f*, taken together as shown in Fig. 88, comprise the circuits for developing the drum pulses and the write sample pulse.

Figs. 54*a* through 54*f*, taken together as shown in Fig. 89, comprise the circuits for developing the developed pulses and special word gates.

Figs. 55*a* through 55*f*, taken together as shown in Fig. 90, comprise the circuits for developing the special digit gates.

Figs. 56*a* and 56*b*, taken together as shown in Fig. 91, comprise the circuits for developing the sector gates.

Figs. 57*a* through 57*c*, taken together as shown in Fig. 92, comprise the circuits for developing the word gates.

Figs. 58*a* through 58*c*, taken together as shown in Fig. 93, comprise the circuits for developing the digit gates.

Figs. 59*a* through 59*o*, taken together as shown in Fig. 94, comprise the circuits of the read-write matrix.

Fig. 60 is a timing chart for the distributor.

Figs. 61*a* through 61*l*, taken together as shown in Fig. 95, comprise the circuits of the distributor.

Figs. 62*a* through 62*l*, taken together as shown in Fig. 96, comprise the circuits of the program step register.

Fig. 63 is a timing chart for the program step register.

Figs. 64*a* through 64*j*, taken together as shown in Fig. 97, comprise the circuits of the accumulator.

Fig. 65 is a timing chart for the accumulator.

Figs. 66*a* through 66*d*, taken together as shown in Fig. 98, comprise the circuits of the adder entry A.

Figs. 67*a* through 67*g*, taken together as shown in Fig. 99, comprise the circuits of the adder entry B.

Figs. 68*a* through 68*l*, taken together as shown in Fig. 100, comprise the circuits of the adder.

Figs. 69*a* through 69*e*, taken together as shown in Fig. 101, comprise the circuits of the operation code register.

Fig. 70 is a timing chart for a right shift operation.

Figs. 71*a* through 71*l*, taken together as shown in Fig. 102, comprise the circuits of the address register and dynamic selection.

Figs. 72*a* through 72*c*, taken together as shown in Fig. 103, comprise the circuits of the translators.

Figs. 73*a* and 73*b*, taken together as shown in Fig. 104, comprise the circuits of the selected storage switching.

Figs. 74*a* and 74*b*, taken together as shown in Fig. 105, comprise the circuits of the control switches of the control console.

Figs. 75*a* through 75*f*, taken together as shown in Fig. 106, comprise the circuits of the address selection switches of the control console.

Figs. 76*a* and 76*b*, taken together as shown in Fig. 107, comprise the circuits of the display light selection.

Figs. 77*a* through 77*d*, taken together as shown in Fig. 108, comprise the circuits of the control console lights and storage entry switching.

Figs. 78*a* through 78*l*, taken together as shown in Fig. 109, comprise the circuits of the operation code switching.

Figs. 79*a* and 79*b*, taken together as shown in Fig. 110, comprise the circuits of the error sense and error stop circuits.

Fig. 80 is a timing chart for a left shift operation.

Figs. 81*a* through 81*l*, taken together as shown in Fig. 111, comprise the circuits of the control commutator.

Figs. 82*a* through 82*c*, taken together as shown in Fig. 112, comprise the circuits of the validity checks.

Figs. 83*a* and 83*b*, taken together as shown in Fig. 113, comprise the circuits of the table look-up carry and accumulator zero check.

Fig. 84 comprises the circuits of the accumulator zero-no zero control.

Figs. 85*a* through 85*t*, taken together as shown in Fig. 114, comprise the circuits of the arithmetic controls.

Figs. 86*a* through 86*h*, taken together as shown in Fig. 115, comprise the circuits of the table look-up, upper-lower and special digit controls.

Figs. 87*a* and 87*b*, taken together as shown in Fig. 116, comprise the circuits of the buffer storage control circuits.

Figs. 117*a* through 117*c*, taken together as shown in Fig. 11, are a timing chart for a multiply operation.

Fig. 118 shows various ones of the basic timing gates and pulses.

Fig. 119 shows various other basic timing gates and pulses.

Fig. 120 is a functional block diagram of the principal structures and the data flow paths therebetween for a Table Look-Up operation.

The numbers at the lines leaving and entering the wiring diagrams of the various components are the figure numbers from which the wire comes or to which the wire goes.

TABLE OF CONTENTS

TABLE OF CONTENTS—Continued

TABLE OF CONTENTS—Continued

GENERAL PRINCIPLES AND OVERALL MACHINE CONCEPT

General features

In Fig. 1 there is shown a simplified block diagram of a stored program type data processing machine. This machine is completely self-checking without requiring the use of recalculation or special checking programs and consequently provides a high degree of reliability. It has a large memory capacity in the form of magnetic drum storage and utilizes a system of stored programming which contributes to ease of operation and flexibility in application.

The machine is provided with a main storage capacity of 2000 "words." A word consists of up to ten digits and an algebraic sign. Magnetic drum buffer storage is provided between the main storage and the input and output units for increased speed of operation. Thus the machine can be calculating while both the input and output units are operating.

Words are stored serially on the drum. Further, the digits of a word are stored serially within each word interval. Digits are represented by parallel combinations of magnetically stored bits. Information is thus said to be stored serially by word and digit, parallel by bit.

Two systems of digit representation by stored bits are used in the machine. The buffer storages and the arithmetic units use a bi-quinary system, where the presence of two of seven possible parallel stored bits determine the digit's decimal value. All General Storage uses a "two out of five" bit system, where two of five possible parallel stored bits determine the decimal value of the digit.

Each of the word storage positions on the drum is located by a four digit code or "address."

In the stored programming system used by this machine, each instruction (program step) is stored in a word storage location as a ten digit word. The coded digits of an instruction word, when interpreted by the program control circuits, give information as to which operation is to be performed, in which storage location to find the data to be used in performing the operation and in which storage location the next ten digit instruction word is to be found. A stored sequence of such instruction words forms a program or "program routine."

Calculations are performed by electronic means. The calculator can add, subtract, multiply, divide and make logical tests such as plus, minus and zero accumulator balance. The program routine can be altered by any of these logical tests or by sensing a control punch in a card. All of these arithmetical and logical operations are built into the machine. They are activated by the operation code portion of the instruction word.

The calculator will accumulate 10 digit words to develop a 20 digit total, perform 10×10 multiplication to develop a 20 digit product and divide a 19 digit dividend by a 10 digit divisor to develop a 10 digit quotient and a 10 digit remainder. Sign control on all operations is automatic.

The arithmetic units of the machine are designed to handle numbers in a serial fashion. Thus during calculations the 10 digit words are processed by the arithmetic units on a digit by digit basis with machine time progressing from the unit's digit through the highest order digit of a word.

The basic cyclical timings of the machine are therefore related to digit position rather than digit value. In the arithmetic portion of the machine the value of a digit is determined by simultaneous combinations of bit pulses on two of the seven parallel information lines.

Number systems for digit representations

A position on a magnetic surface may be considered a binary storage device since it may conveniently have either of two magnetic conditions induced therein. One condition is taken to mean a binary "1" while the opposite direction means a binary "0." Some system must be used so that a combination of these binary devices can represent the decimal digits 0 through 9.

In this machine two such systems are used. The 2 of 7, bi-quinary coded decimal system, is used for Buffer Storages and for the arithmetic units. The 2 out of 5, 5 bit system is used for General Storage.

Fig. 2 shows how the bi-quinary system combines different binary "1" indications from two of seven parallel devices to represent each of the ten decimal digits. In the case of the Buffer Storages, the binary devices are recorded magnetic spots on seven parallel tracks. Successive "1" indications from the various combinations of two of the seven tracks during successive digit time intervals will indicate serially the decimal values of the digits of a word. In the case of the arithmetic units the paralleled binary devices are primarily capacitor storage units. Seven parallel capacitors are used to represent each of the ten digits of a word. Each row is tested at its digit time by pulses from a drum driven timing ring. Binary "1" indications from two of the seven capacitors will indicate the decimal value of the digit.

The bi-quinary system requires that there be a "1" indication from one and only one of the binary indicating devices in each of the two levels. It is this unique characteristic of the bi-quinary system which makes it easily adaptable to self-checking arrangements.

The binary level indication determines whether the quinary level indication is to be read as it is or have five added to it to determine the decimal value. This above or below five characteristic of the system makes it easily adaptable to calculating circuitry.

Fig. 3 shows how the "two of five" system combines different binary "1" indications from two of five parallel devices to represent each of the ten decimal digits. This system is used only for general storage largely because of the saving of heads and circuits which it affords. Use of this system requires that all information entering general storage pass through a 7 to 5 conversion matrix and that all information leaving general storage pass through a 5 to 7 conversion matrix.

Drum arrangement and timing

The calculator is built around the magnetic drum (Figs. 4 and 5). This drum may be a plated metallic cylinder approximately 4 inches in diameter and 16 inches long. Information is stored on its surface in the form of magnetic spots. In order to place information on the drum in an orderly fashion and to have the stored information available periodically the drum must revolve. In this machine the drum is driven at a speed of 12,500 r.p.m. Around the drum, attached to its end frames and parallel to its axis are a number of mounting bars on which are mounted the inductive heads which read and write information. Information stored on the drum is only erased by writing new information and thus may be read as often as needed without being destroyed.

The General Storage portion of the drum has 2000 addressable locations where ten digit words can be stored. Each of these word positions is located by first determining statically which one of 40, five track bands it is in across the drum and then dynamically which one of 50 angular drum positions it occupies around the periphery of the drum.

To accomplish head selection the 4 digit address portion of the instruction word is placed in the Address Register where it activates the head selection circuits. Both the static and the dynamic selection are determined by the numerical value of the 4 digit address in the Address Register.

For static selection purposes the General Storage portion of the drum is divided lengthwise into 40 five track bands, 1 head per track. Each of the 40 bands has a "6" bit track, a "3" bit track, a "2" bit track a "1" bit track and a "0" bit track in accordance with the "two of five" system of digit representation. Fifty 10 digit words are stored serially by digit and word in each of these 40 bands, ten words in each of five drum timing sectors. This drum division is illustrated in Figs. 4 and 5.

In actual practice the five heads of a band are not physically adjacent but are displaced in a spiral fashion around the drum in such a way that the "6" bit heads are adjacent as are the "3," "2," "1" and "0" heads each in its lateral row. This positioning is made necessary by the fact that the width of a head is somewhat greater than the width of a drum track. This spacing of the heads of a band in no way affects the machine operation since information placed on the drum by simultaneous impulsing of the heads of a band will later be available whenever the recorded spots pass under the same head positions during the same drum time interval.

The 4 digit address will appear in the Address Register in its bi-quinary form. The head selection circuits will select the proper 5 heads for reading or writing by interpreting the meaning of the 4 digit bi-quinary address. Fig. 6 shows how the 2000 positions of General Storage are arranged.

To explain this arrangement two specific addresses shown on Fig. 6 will be analyzed. The bi-quinary form of address 0225 is B0Q0, B0Q2, B0Q2, B5Q0. The hundreds position 02 indicates that the address is in the left half of the drum since the B0H addresses the left half of the drum and B5H addresses the right half of the drum. QH selects four of the twenty bands in the BH group, BT selects two of the four QH bands and QT*h* selects one of the pair. This static selection chooses one of the 40 five track bands by interpreting the QT*h*, BH, QH and BT values of the 4 digit address.

The dynamic selection of one of the 50 words in the band is accomplished by the QT, BU and QU values of the address. For dynamic selection and other timing purposes the drum is divided into 5 sectors (0–4) of 10 word intervals each (0–9). In the address 0225 the Q2T selects the third sector (sector 2) while B5U and Q0U select the 6th word interval in the sector.

Address 1683 in bi-quinary form is 01 51 53 03. B5H selects the right half of the drum. Q1H selects four of the twenty B5H bands. B5T selects two of the four Q1H bands and Q1T*h* selects the right-hand band of this pair.

Q3T selects sector 4 while B0U and Q3U select the 4th word position of sector 4.

All machine timings are related to the angular position of the drum. Figs. 4 and 5 show the various important timing intervals and how they relate to the drum. The drum is divided into 5 sectors. Each sector is further divided into 10 word intervals. Each word divides into 12 equal digit intervals, 10 digits, 1 sign position and a separation interval called Digit X (DX) between words. Each digit time is finally divided into four equal intervals A, B, C and D.

The 6 timing tracks contain permanently recorded spots which are used to establish reference positions of the drum. These signals drive timing rings and pulse developing circuits which produce all of the timing pulses and gates used throughout the machine. The six recorded signals are:

	Per revolution
(1) Home pulse	1
(2) Sector pulse	5
(3) Word pulse	50
(4) B pulse	600
(5) D pulse	600
(6) Read sample pulse	600

The drum may be, for example, 4 inches in diameter and rotates at a speed of 12,500 r.p.m. or 4.8 ms. or 4800 micro-seconds per revolution. Since there are 600 digit intervals around the drum then 1 digit interval$=\frac{1}{600}\times 4800$ micro-seconds$=8$ micro-seconds. Each A, B, C and D pulse then is ¼ of a digit time or 2 micro-seconds. The actual circumferential space occupied by a digit interval is $\frac{1}{600}\times 12.57$ inches or approximately .021 inch. Thus the "spot density" is approximately 50/inch.

Fig. 5 shows a 7 track storage band between the timing tracks and General Storage. This band is used for read-in and read-out buffer storage, called Read Buffer Storage and Punch Buffer Storage respectively. Both Read and Punch Buffer Storages use the same band and the same heads for reading and writing but occupy different sector positions on the drum and are thus written into and read from at different drum times. Digit values of words stored in Read and Punch Buffer Storage are represented in the bi-quinary system, thus 7 tracks are used. Both Read and Punch Buffer Storages have a capacity of 10 words. All card input data enters General Storage via RBS and all output data from General Storage is punched via Punch Buffer Storage.

On read-in operations, the parallel punched holes forming the digit notation of the words punched in the card are scanned by action of a cathode follower tube matrix and converted into serial forms for storage in the 10 serial word locations of RBS. At the same time the decimal punched hole notation is changed to the proper bi-quinary form of representation. This structure is described in more detail and claimed in the application of F. E. Hamilton et al., Serial No. 399,496, filed December 21, 1953, now Patent No. 2,877,450, and assigned to the present assignee.

On read-out operations the serial pulses from the PBS read circuits which form the bi-quinary digit notation of the 10 words in PBS are converted into single, punch cycle timed, pulses and distributed to the proper punch magnet for punching in decimal notation by action of a thyratron punch scanning matrix. This structure is described in more detail and claimed in the application of F. E. Hamilton et al., Serial No. 464,516, filed October 25, 1954, now Patent No. 2,919,429, and assigned to the present assignee.

Both reading and punching are initiated by an operation instruction interpreted by the program control circuits. On a read operation calculations are interrupted just long enough for the 10 words in RBS to transfer to 10 sequential General Storage locations the first of which is specified by the address portion of the read operation instruction word. The time required to complete this transfer will be from a minimum of 10 word times (960 micro-seconds) to a maximum of 1 drum revolution (4.8 ms.). The words so transferred, entered RBS from the card which passed the reading brushes on the previous read operation. Calculations then re-start and continue while the information from the next card is entering RBS at the slower card-reader speed. On a punch operation 10 sequential words from General Storage are transferred to the 10 word positions of PBS

from where they are scanned out to punch at punch speed. The General Storage location of the first of these 10 sequential words is specified by the address portion of the punch operation instruction word. The time required to complete this transfer from General Storage to PFS will be from 960 micro-seconds to 4.8 ms.

The "block" transfer of 10 words from RBS to General Storage is accomplished by switching together the RBS "read" output circuits and the General Storage "write" input circuits during the drum time that the words of RBS are passing their read heads. Similarly a transfer of 10 words from General Storage to PBS is accomplished by switching together the General Storage "read" output circuits and the PBS "write" input circuits during the drum time that the 10 General Storage words are passing their read heads. This means that those General Storage word addresses which occupy the same drum sector as the words of RBS receive information from RBS and those General Storage word addresses which occupy the same sector as the words of PBS, send information to PBS. Since there are 40 bands of General Storage, there are 40, ten word blocks or 400 word addresses which accept information from RBS and 40 other 10 word blocks which send information to PBS. These addresses will be specified in the section on Programming.

If it is desired to store original data or instructions in any General Storage location other than one of these 400, a transfer must be programmed. This programmed transfer is usually part of a pre-stored transfer routine which is used to distribute original information to the desired General Storage locations. Information to be punched must be placed by programming in one of the 40 General Storage 10 word blocks associated with PBS.

Word locations in RBS and PBS are not addressed but function automatically on programmed read and punch operations.

The area between the dotted lines of Fig. 5 shows the General Storage positions which can be entered from RBS.

Stored programming

The present machine uses a system of stored programming to provide the necessary sequence of operations for the solution of a problem, e.g., the machine refers to any of its own storage locations to obtain a previously stored or computed ten digit, coded, instruction word whose digit values can be interpreted by the machine to determine what its next operation should be.

Original data and instructions are normally stored in drum storage locations from punched cards during the loading process. Additional data and/or instructions may be inserted from cards during the solution of the problem. Each instruction (program step) is stored as a word. Since both data and instructions are stored in the same manner, an instruction word can be subjected to arithmetical operations and thus can be altered by programming. The meaning of any valid coded instruction is built into the machine. Any sequence of instructions is called a program or program routine.

All instructions are in the form of 10 digit words. The sign is carried along for checking purposes. It has no effect on the meaning of the instruction, but must be considered when the instruction word is altered arithmetically.

Referring to Fig. 7, it may be seen that the instruction word is divided into three sections. Digit positions 10 and 9 are the Operation Code which tells the machine which of its several operations to perform on this program step. Positions 8–5 are the data address and usually mean either the location of information to be used in the operation, or the location where the information is to be stored as a result of the operation. In certain operations the data address may also have one of the following meanings:

(1) The number of positions to be shifted either right or left.

(2) The General Storage band where the contents of the ten word positions of RBS are to be entered on a read operation. Any one of the 50 addresses of a band when used as the data address of a read instruction word will cause a transfer of the ten RBS words to the word 1 through 10 locations of the band.

(3) The General Storage band from which information is to be punched. Any one of the 50 addresses of a band, when used as the data address of a punch instruction word, will cause a transfer of the contents of word positions 27 through 36 of the band to the ten word positions of PBS. The transferred information will be punched from PBS.

(4) The location of an alternate instruction when selected by a branching operation. In any case the meaning of the data address depends on the associated operation code. Positions 4–1 are the instruction address which indicates the location in storage of the instruction word for the next program step, unless a branching operation has indicated the data address is to be used for this purpose.

A more complete summary of Addresses and Operation Codes will be found in a later section on Programming.

Logical organization, data flow and program control

The general logical arrangement of the machine is shown in Fig. 1. The principal functional units and the various paths of information flow are shown.

It should be remembered that, although the data flow paths in Fig. 1 are shown as a single line to simplify the illustration, each data flow path actually comprises seven parallel lines and that in accordance with the biquinary system, information pulses representing the numerical value of each digit will be present on two of these seven lines during each digit interval.

As explained in a preceding section on Drum Arrangement, original data and instructions enter the 10 word RBS section of the drum after passing through the converter circuits where they are changed from the parallel, decimal notation of the punched card to the serial, bi-quinary form of RBS. A programmed read operation then causes a block transfer of the 10 RBS word blocks which can accept information from RBS. These General Storage locations are marked "Card R.I." on Fig. 1.

A programmed punch operation causes a block transfer of 10 words from one of the 40 ten word General Storage blocks marked "Card Punch-Out" on Fig. 1, to the 10 word PBS section of the drum. These ten PBS words then pass through the converter circuits where they are changed from serial, bi-quinary form to punch unit timed pulses for punching into the card in parallel, decimal notation. Fig. 1 also shows that all information entering the General Storage section of the drum passes through a 7 to 5 conversion matrix where the 7 bit, bi-quinary digit representation is changed to the 5 bit code used for General Storage. Similarly, all information leaving General Storage passes through a 5 to 7 conversion matrix where the 5 bit representation is changed back to 7 bit, bi-quinary form. These 7 to 5 and 5 to 7 circuits are made up of combinations of diode coincidence switches and diode OR circuits.

The arithmetical and logical operations such as add, subtract, multiply, divide, transfer of words to and from storage, shifting and logical tests are carried out by the action of the Distributor, the Accumulator and the one digit Matrix Adder. The functionings of these units are controlled by the Program Control Circuits of which the Program Step storage unit and the Operation and Address Register are a part.

Each program step is performed in two parts or "half cycles." On the first part or "I" half cycle the operation and address registers are reset and the "I" address is placed in the address register where it is interpreted and used to select the location specified by the "I" address. The new instruction word is then read out of the selected "I" address location and into the program register. On the second part of the program step or "D" half cycle the OP and address registers are reset and the OP and Data parts of the new instruction word now in the program register are read out of the program register and into the OP and address registers. Here the data address is interpreted and used to select the "D" address. The data word is then read out of the "D" address location and into the Distributor. The OP code is interpreted and activates the operation which is performed using the data now in the Distributor. This completes this program step.

As soon as the operation is started program control causes a return to the "I" half cycle. The OP and address registers are reset and the "I" part of the instruction word in the program register is transferred to the address register, replacing the "D" address there, where it is interpreted and used to select the next "I" address. The next instruction word is then read out of the "I" address location and into the program register replacing the previous instruction word. The OP and address registers are again reset and the OP and Data parts of the new instruction word are read out of the program register and into the OP and address registers. At this point an interlock prevents further program advance except for read or punch operations, until the previous program step has finished using the arithmetic units. In this manner the machine advances through the steps of a stored program routine.

This half-cycle action, by which a program step is performed, is accomplished by a Program Control Commutator shown diagrammatically in Fig. 8, which controls the sequence of actions necessary to advance through any program step. This control commutator is a two branched ring with several positions in each branch. As it cycles, it alternately advances through each branch. The positions of one branch control the functions of the "I" half cycle, while the positions of the other branch control the functions of the "D" half cycle. Normally the ring must advance through both branches, first "I" then "D," to complete a program step. The outputs of the steps of the control commutator are used to control the various transfers of data required for the accomplishment of the program step.

Program Step Storage (Program Register) and the Distributor are capacitor storage units capable of storing serially, in bi-quinary form, the ten digits of a word and of making continuously available serially, a bi-quinary signal representing each of the ten digits in its proper sequence, once every word interval ($\frac{1}{50}$ drum revolution). Fig. 1 shows that both units receive information from a common storage exit channel but the switching into Program Step Storage is only open on an "I" half cycle while that controlling entry into the Distributor is only open on a "D" half cycle.

The main function of Program Step Storage is to hold the instruction word which it receives from the "I" address, and supply the proper part of this word, to the OP and address registers for interpretation, at the proper time.

The Distributor acts as a buffer between the accumulating components and addressable storage locations. Its main functions are to receive a word of data from a selected data address and make this word available to the accumulating circuits as required by the operation, or to receive a word of data from the accumulating circuits or the control console switches and make this word available for entry into a selected General Storage location. The Distributor is also an addressable storage location (8001) and can be used as a source of information, through "Selected Storage" switching, to supply an instruction word to the program register.

Capacitor storage units read their stored information out continuously during each drum word interval. This is done by driving the digit positions of the unit with drum timed digit pulses from a drum driven digit ring. Each capacitor storage unit has a "one digit early" output and an "on time" output. This is accomplished by driving each digit position of the unit with a drum time digit pulse one digit time in advance of the digit position of the storage unit. Thus the digit one position of the unit is read out at digit zero drum time, digit two of the unit is driven at digit one drum time, etc. Each early output is then delayed for one digit interval to provide an additional "on time" output. Either output may be used depending upon the operation.

Capacitor storage units are a "non-sustaining" type of storage, i.e., each position requires periodic "regeneration" of its stored information. Regeneration is accomplished within the unit by providing a data flow path between the outputs of the unit and its input circuits. Thus information which is available from the D1 position at digit zero drum time is read back into the D1 position at digit one drum time, etc., as long as the regeneration path is closed. Reading new information into a unit merely requires that the regeneration path be opened and that "on time" output signals from the desired source be substituted at the storage unit input circuits in place of the unit's own "on time" output signals. This is done whenever the input switching to the capacitor storage unit is activated.

Fig. 1 shows data flow paths from the outputs of all capacitor storage units. The information contained in any capacitor storage unit is continuously present serially, digit by digit, on these data flow paths. The data flow path from each of the capacitor storage units connects to the input switching circuitry of various other machine units, e.g., the Distributor output connects to the Adder "B" entry switching, Selected Storage switching, the Control Console Light entry switching and the General Storage entry switching. Reading information out of a capacitor storage unit merely requires that the entry switching for the desired receiving unit be opened for just the word time necessary to gate through the succession of output digit pulses representing the stored word.

The Operation Code register is a two position, bi-quinary, static storage unit. Each of its positions uses seven latches (electronic binary storage device) to indicate the OP code value in bi-quinary form. It can accept information from the 9th and 10th positions of the program register on a "D" half cycle. Once a code is entered, continuous, steady state, bi-quinary output is available from two of the seven latches until the register is reset. These outputs are used in coincidence switching arrangements to control the indicated operation.

The address register is a four position, bi-quinary, static storage unit similar to the OP code register. It can accept information from positions 5–8 of the program register on "D" half cycle or from positions 1–4 on an "I" half cycle. Its continuous steady state bi-quinary outputs are used by the General Storage Read-Write Matrix to select statically the proper storage band for reading out "I" and "D" addresses or for writing new information on programmed store operations and read operations. These outputs are also matched with the proper drum timing pulses by coincidence switching arrangements in the Dynamic Selection circuitry to select the proper word time for reading from or writing into a statically selected General Storage band.

OP code and address register outputs are also used as signals which help to control the advance of the Program Control commutator.

Fig. 1 shows that information from these registers is also instrumental in controlling Selected Storage switching in those cases where it is desired to use the Control Console switches, Upper and Lower Accumulator or the Distributor as a source of information, rather than a General Storage location.

The Accumulator is a capacitor storage unit similar to the program register and the Distributor, except that it has a two word capacity. The serial digit signals of each word are available at their proper "early" and "on time" drum digit times once every alternate word interval. The Accumulator is divided into two halves for programming and calculating purposes, the low order ten digits form the Lower Accumulator and the high ten digits the Upper Accumulator. Information in the Lower Accumulator is continuously available during each even numbered drum word interval while that in the Upper Accumulator is continuously available during each odd numbered drum word interval. Fig. 1 shows data flow paths from the accumulator output to its own input, to one of the added inputs, to Selected Storage switching, to the Distributor and to the Control Console lights. Transfer of information from either half of the accumulator to one of these destinations is accomplished by controlling the read-in switching of the receiving unit to be open for the proper even or odd word time, during which time the output of the desired half of the accumulator is gated through to the receiving unit.

The Upper Accumulator can also act as an overflow for the lower when a sum or product exceeds ten digits.

The main function of the Accumulator is to work with the Adder in Accumulating sums, products and quotients and in performing shift operations. It also supplied zero balance, plus balance, minus balance and overflow signals to the program control circuits for the logical test operations and can function as an addressable source of information (8002 or 8003) to supply an instruction word or a data word for programming purposes.

The Matrix Adder is a diode and tube unit capable of receiving two input digit values, analyzing them, and producing a digit output signal equal in value to the sum or difference of the two input values. The Distributor and the Accumulator feed into the Matrix Adder. The output from the Matrix Adder is stored back in the Accumulator. A delay of one digit time is incurred between input to and output from the Adder. An output will be available from the Matrix Adder only as a result of the entry of two input digits, one on each input line. Digits from both the Distributor and the Accumulator can enter the Adder, the Distributor digits on one input line and the Accumulator digits on the other. Provision is also made, in certain operations, for blanking the Distributor or Accumulator digit values and substituting zeros or constants for entry into the Matrix Adder. Most calculator operations are performed by merging Distributor and Accumulator digit values or their substitutes in the Matrix Adder and storing the result back in the Accumulator. In most arithmetical operations the Distributor and Accumulator "early" output is used to produce an "on time" Adder output for the Accumulator to store. In other operations such as shifting, the "on time" output may be used.

A summary of the various data flow paths follows:

Digits can enter the distributor from any general storage location, from the entry switches on the control console or from either half of the accumulator. The selected storage path is used between the accumulator and distributor on arithmetic operations when the data address is either half of the accumulator. The direct path is used on a "store accumulator" operation. In this case the data address is a general storage location meaning the destination of the word in the accumulator. The distributor can supply a word of information to the adder, to any general storage location, to the display lights on the control console (manual operation) or to the program register through selected storage switching.

The accumulator can be entered only from the output of the adder or from its own output. The accumulator can supply information to the adder, the distributor directly, the distributor or the program register through selected storage switching, or the display lights (manual operation).

The program register can be entered from the common storage exit line (either general storage output or selected storage) or from the adder on a table look-up operation. The program register can supply information to the OP and address registers, the display lights or to the adder on a table look-up operation.

The OP and address registers can be entered from the program register. The OP and "D" part of an instruction word is transferred on a "D" half cycle, the "I" part on an "I" half cycle. The OP and address register static outputs control general storage head selection, the machine "operation," selected storage switching and the control commutator.

The adder normally accepts information simultaneously from the accumulator and distributor and supplies information to the accumulator. It can also receive data from general storage or the program register and read out to the program register on table look-up operations.

General storage can be read into from either RBS or the distributor. General storage can read out either to PBS, the program register, the distributor, or to the adder on a table look-up operation.

The control console storage entry switches (8000) can read out to selected storage and thus to either the program register or the distributor.

Physical organization

The machine is comprised of three basic physical units interconnected by cables; a read-punch unit, a power unit and the calculator unit.

The read-punch unit is basically a high speed punch and a card reader. The card reader is used to read information into the calculator while the punch is used to record information taken from the calculator. The details of the read-punch unit are not shown in detail herein since any number of input and output devices could be used equally well with the calculator of the present invention.

The power unit includes most of the power supply components used to provide the various necessary voltages. The power unit is not shown in detail herein since the various voltages may be generated in any well known manner. Only the voltage values for the preferred embodiment are shown.

The calculator unit contains the magnetic drum, its associated read and write circuits and all calculating and program control circuits. In addition, it mounts the operators control console which contains switches and indicating lights for operation of the machine.

Self-checking

Accuracy of calculations is achieved by checking the transmission of all data and instructions to satisfy the validity requirement that there be one and only one bit in each level (binary and quinary) of a digit. When a validity check error occurs the machine may be stopped or the program routine may be branched to repeat a portion of the problem or to initiate a different problem.

Fig. 1 shows the location of these validity check circuits, at the output of each of the capacitor storage units. Since all information eventually passes through these units, any invalid combinations will be detected.

In addition to these validity checking features, the program control circuits and the arithmetic circuits are designed with inherent checking features and "fail safe" logical arrangements. These control checks are dependent upon the proper combination of signals within the machine, the proper sequence of signals, double circuitry and back signals which insure that the signal to perform some operation was received and that the operation was completed before proceeding to the next step.

A special check assures that information is entered into only one general storage location on any general storage "write" operation.

Meaningless address and operation codes are detected and cause an error indication.

The timing rings are continuously checked on each cycle of their operation. Any timing error is detected and causes a timing error indication. This structure is described and claimed in the copending application of F. E. Hamilton, Serial No. 408,702, filed February 8, 1954, now Patent No. 2,819,457, and assigned to the present assignee.

A continuous check is made of the accumulator for an accumulator overflow or a quotient of more than 10 digits. An overflow condition can cause a machine stop or can be used to branch the program routine.

Operating controls and indicators

Control switches and indicating lights for operating the machine and interpreting results are provided. The panel containing the operating switches and lights is called the control console.

The control console contains switches and lights through which the operator may observe and control the course of the problem being solved. Its principal uses are in checking out new program routines and in trouble analysis. A brief functional description of various ones of these switches and lights will be given presently with a more detailed description of the circuits and their operation following in a later section.

Power On Key.—When this key, Fig. 74*b*, is depressed the following sequence of events takes place:

(1) A.C. power is supplied to all parts of the calculator and the blowers and drum start.

(2) D.C. power automatically comes on after a three-minute delay.

Power On Light.—This light, Fig. 74*b*, comes on whenever the Power On Key is depressed. It remains on as long as A.C. power is supplied.

Ready Light.—This light, Fig. 74*b*, comes on when the D.C. power is turned on, either in the starting sequence after depressing the Power On Key or after depressing the the D.C. On Key. It remains on as long as D.C. power is supplied to the calculator.

Fuse Light.—This light, Fig. 74*b*, comes on whenever a fuse has blown.

Power Off Key.—When this key, Fig. 74*a*, is depressed the following sequence of events takes place:

(1) D.C. power is turned off.

(2) A.C. power is turned off and the drum motor is turned off.

(3) The blower motors are turned off after a five-minute delay.

D.C. Off Key.—When this key, Fig. 74*a*, is depressed, all D.C. power is turned off and the Ready Light goes off.

D.C. On Key.—When this key, Fig. 74*a*, is depressed, D.C. power is turned on and the Ready Light comes on. This key in inoperative until after the three-minute delay just mentioned.

The D.C. Off and D.C. On keys are provided so that the D.C. power may be turned off without turning the entire calculator off.

The remaining keys, switches and lights are used by the operator to observe and control the problem being solved. They are arranged so that those having similar functions are located close together.

Display Lights.—These are ten sets of seven lights and one set of two lights shown in Figs. 77*a*, *b*, *c* and *d*. Each set of seven lights has two binary lights arranged horizontally and five quinary lights arranged vertically. The digit values 0 and 5 are associated with their corresponding quinary lights; digit values 1 and 6 are associated with their corresponding quinary lights, and similarly 2 and 7, 3 and 8, and 4 and 9. The binary light indicates which of the two values indicated by the quinary light is represented. This arrangement makes it possible to tell at a glance what number is represented.

The sign lights, Fig. 77*a*, indicate the sign of the number represented by the display lights.

The display lights are used in conjunction with other keys and switches on the console to examine the contents of any location. Examples of how they are used are given in the section on Examples of Use of the Control Console.

Storage Entry Switches.—Any instruction or ten-digit signed number may be set on these switches, Fig. 77. They are used in conjunction with other keys and switches on the console to enter data or instructions into any storage location. They are usually used to enter corrections when correcting a program. They may be consulted by the program by using the address 8000.

Order Lights.—These two sets of lights shown in Fig. 75c indicate the contents of the operation register.

Address Lights.—These four sets of lights shown in Figs. 75*a* and 75*b* indicate the contents of the address register. Since this address may be either a data address or an instruction address, some means of distinguishing between these two cases must be provided. This is done by means of the data address light and the instruction address light of the operating lights, Fig. 74*b*. If the address in the address register is a data address, the data address light will be on, but if it is an instruction address, the instruction address light will be on.

Operating Lights.—These lights, Figs. 74*a* and 74*b*, indicate what is happening in the calculator at any time. As has been explained under Address Lights, the data address light and the instruction address light indicate the status of the control unit in that they indicate whether the address in the address register is a data address or an instruction address. The remaining four lights indicate the status of the read, punch, accumulator, and program units. The various functions of the operating lights are described as follows:

Program Light—If the calculator stops because of a programmed stop, address stop, or manual stop, the program light will turn on.

Accumulator Light—This light is on when the accumulator is being used.

Read Light—This light is on after a read instruction until the beginning of a card feed. If the calculator stops and this light remains on, it means that a card has failed to feed.

Punch Light—This light is on after a punch instruction until the beginning of a card feed. If the calculator stops and this light remains on, it means that a card has failed to feed.

Checking Lights.—These lights are used to indicate where an error has been detected. Their different functions are described in the following:

Program Register Light—This light comes on if an error has been detected in the program register.

Distributor Light—This light comes on if an error has been detected in the distributor.

Accumulator Light—This light comes on if an error has been detected in the accumulator.

Clocking Light—The clocking circuits consist of a series of rings driven by pulses recorded on the drum. These rings emit timed pulses with a duration equivalent to a digit value, a word, and one-fifth of a revolution. They are used to indicate the position of the drum at any instant. The failure of any ring to complete its cycle in this given time will illuminate the clocking light and stop the program at the end of that word.

Overflow Light—This light comes on if an overflow has been detected in the accumulator and the overflow switch, Fig. 75*f*, on the console is set on stop. If it is set to sense, the overflow circuit will be set. The overflow circuit may be interrogated by means of code 47 to determine automatically whether an overflow has occurred or not. The use of this code resets the overflow circuit and turns off the overflow light.

An exception to the above is quotient overflow. In this case the calculator will stop and the overflow light will come on, whether the overflow switch is set to stop or sense.

If the calculator has stopped because of an overflow, one of the reset buttons (except error sense) must be depressed before the program can proceed. Depressing this button will also turn off the overflow light.

Error Sense Light—If the error switch, Fig. 75*f*, is set to stop and an error is detected, the calculator will stop and the proper error indicating light will come on. One of the reset keys (except error sense) must be depressed before the calculator can be started again. Depressing one of these keys also turns off the error indicating light.

If the error switch is set to sense and an error is detected, the error sense light will come on, the program register, distributor and accumulator will be reset to zeros and the address register will be reset to 8000 which is the address of the storage entry switches. The next instruction will be taken from the storage entry switches. This feature is useful for self-correcting programs. The calculator will proceed automatically with the self-correcting program, and the error sense light will remain on to indicate to the operator that an error has occurred. It is turned off by depressing the error-sense reset button.

Programmed Stop Switch.—When this switch, Fig. 75*c*, is in the run position, the program will continue when a stop code (01) is reached. It will treat the stop code as a NO OP code (00).

When this switch is in the stop position the program will proceed until a stop code is reached. The calculator will stop, and the program light will be on.

Examples of the use of the programmed stop feature are given in the section on Examples of the Use of the Control Console.

Half-Cycle Switch.—When this switch, Fig. 75*c*, is in the run position, the program will proceed automatically once it is started. When it is in the half position, instructions will be executed one address at a time. That is, the first half of a cycle is that part of an instruction execution during which the data address of the instruction is in the address register and the second half is that part during which the instruction address is in the address register. Successively depressing the program start key will cause one half of a cycle to be executed. By this means, it is possible to cycle through portions of a new program when checking it out on the calculator.

Address Selection Switches.—These switches, Figs. 75*d* and 75*e*, are used to set up an address which may be used in either of the following ways:

(1) Address at which program is to be stopped.

(2) Address to be entered into address register.

Examples of their different uses are given in the section on Examples of Use of the Control Console.

Control Switch.—When this switch, Fig. 75*f*, is in the address stop position, the program proceeds until the address in the address register is the same as the address set on the address selection switches. When this occurs, the calculator stops.

When this switch is in the run position, the execution of the program proceeds automatically once the program has been started.

When this switch is in the manual position, the machine may be operated manually.

Examples of the various uses of this switch are given in the section on Examples of Use of the Control Console.

Display Switch.—When this switch, Fig. 76*b*, is in the lower accumulator, upper accumulator, program register, or distributor positions, the contents of the corresponding location will automatically be displayed on the display lights. The program must be stopped if the contents of the location are to be read.

The contents of any storage location may be examined by use of the read-out storage position of the switch. Any word may be entered into any storage location by use of the read-in storage position of the switch. Examples of this are given in the section on Examples of Use of the Control Console.

Transfer Key.—Depressing this key, Fig. 75*c*, causes the address set on the address selection switches to enter the address register. The control switch must be set to manual.

Program Start Key.—Depressing this key, Fig. 75*c*, starts the execution of the program. The number of program steps executed depends upon the setting of other switches on the console.

Program Stop Key.—Depressing this key, Fig. 75*c*, causes the program to stop at the end of the program step on which it is depressed.

Program Reset Key.—Depressing this key, Fig. 75*c*, resets the program register to zeros and resets any error circuits which may be on. If the control switch is in the manual position, the address register resets to blanks, but if it is in the run or address stop position, the address register resets to 8000.

Computer Reset Key.—Depressing this key, Fig. 74*b*, resets the program register, distributor, and accumulator to zeros and resets any error circuits which may be on. If the control switch is in the manual position, the address register resets to blanks, but if it is in the run or address stop position, the address register resets to 8000.

Accumulator Reset Key.—Depressing this key, Fig. 74*b*, will reset the distributor and accumulator to zeroes and reset any error circuits which may be on.

Error Reset Key.—Depressing this key, Fig. 74*a*, will reset the error circuits and turn out error indicating lights so that the program may continue when the program start key is depressed.

Examples of use of the control console

Programming Stop.—During the course of checking out a problem, it is often desirable to stop at some point to examine computed results in order that they may be compared with known results. This is usually accomplished by the insertion of a stop code in the program so that the calculator will automatically stop when it reaches that point. After the program has been checked out, it is no longer necessary to stop at these various checking points so that the stop code should be ignored. The programmed stop switch makes it possible to ignore stop codes which have been inserted in the program. If the programmed stop switch is set to stop, the computer will stop when a stop code is reached. If it is set to run, the computer will interpret the stop code as a NO OP code and proceed with the program.

Address Stop.—Another means of stopping the program automatically at any point is through the address stop feature. If it is desired to stop at instruction "I" the address of "I" is set in the address selection switches. The control switch is set to the address stop position and the program is started. The calculator will proceed until the address in the address register is equal to the address set on the address selection switches. When these two addresses are equal, it will stop.

Depressing the program start key will cause the calculator to restart.

By means of this feature it is possible to stop at any point in the program to examine results.

Overflow Sense.—Some operations may cause an overflow as previously described. It is frequently not necessary to stop the calculations when an overflow occurs, but merely to change the sequence of instructions to be executed in such a case. The overflow sense feature makes it possible to detect whether an overflow has occurred and to change the sequence of operations without stopping the calculator. If the overflow switch is set to sense and an overflow in the accumulator has been detected, the calculator will not stop but the overflow circuit will be energized. The BR OV operation code (47) tests whether the overflow circuit has been energized, thus making it possible to alter the sequence of operations automatically on the basis of this test. The overflow circuit can be reset either by use of this operation or by depressing one of the reset keys (except error-sense reset).

A quotient overflow in division always stops the calculator.

If the overflow switch is set to stop, the calculator will stop when any overflow in the accumulator has been detected.

Monitoring.—When a program is being checked out, it is often found desirable to examine the contents of a particular storage location to see if the proper instruction is in that location or to see if a computed result compares with a known result. The monitoring of storage locations may be accomplished as follows: (1) Stop the program. (2) Set the control switch to manual. (3) Set the display switch to read-out storage. (4) Set the address of the location to be examined on the address selection switches. (5) Depress the program reset key. (6) Depress the transfer key. (7) Depress the program start key.

The contents of the desired storage location will be indicated on the display lights. Manual read-out is through the distributor. Therefore, the previous contents of the distributor are destroyed.

When the display switch is in the distributor, program register, lower accumulator, or upper accumulator position, the contents of the corresponding location will be indicated on the display lights. Of course, the program must be stopped. If it is desired to make corrections to a program by insertion of new instructions or data, this can be accomplished by means of the control console. The steps required for manually inserting a word into some storage location are as follows: (1) Stop the program. (2) Set the control switch to manual operation. (3) Set the display switch to read-in storage. (4) Set the word to be entered on the storage entry switches. (5) Set the address of the desired location on the address selection switches. (6) Depress the program reset key. (7) Depress the transfer key. (8) Depress the program start key.

The word set on the storage entry switches will then enter the location set on the address selection switches. The entry can be checked by observing the display lights.

When checking a program it is sometimes necessary to examine the contents of the different storage locations on each program step or to see what happens in each half cycle of the program step. By setting the half cycle switch to Half, the calculator will execute instructions half of a cycle at a time. Thus, we can stop at a program step, set the half-cycle switch to Half Cycle, and examine the contents of the significant storage locations.

Successively depressing the program start key will cause the program to continue one half cycle at a time. On each half cycle the contents of relevant storage locations can be examined to see if the proper results are being produced.

Starting a Program.—There are many ways in which a program may be started. Two examples are as follows: (1) Set 000000 $X_1X_2X_3X_4$ on the storage entry switches where $X_1X_2X_3X_4$ is the address of the first instruction to be executed. (2) Depress the program or computer reset key. (3) Depress the program start key.

This procedure causes the calculator to get its first instruction from the storage entry switches. This instruction has a NO OP operation and its instruction address is the address of the first instruction of the program. The control switch must be on run or address stop positions.

The second example is as follows: (1) Set the control switch to manual. (2) Set the address of the first instruction to be executed on the address selection switches. (3) depress the program or computer reset key. (4)

Depress the transfer key. (5) Set the control switch to run. (6) Depress the program start key.

This procedure causes the calculator to obtain its first instruction from the location set on the address selection switches, this address being the address of the first instruction to be executed.

To start the calculator originally when no instructions are on the drum, and no loading routine is in the machine, a few instructions may be loaded through the control console. These few instructions will be used to enter a more complete loading routine. Once a loading routine is on the drum, it will remain there until replaced by something else. Thus, when the calculator is shut down, it is not necessary to insert a loading program through the console when it is restarted, since the loading routing already in storage may be used. An example of a self-loading routine is given later.

PROGRAMMING

To solve a problem, original data words and instruction words must first be loaded into storage locations, for example, from punched cards.

Storage address and operation codes

The addresses of the storage locations and the meaning of the Operation Codes follows:

Address:	Unit addressed
0000–1999	General Storage (A. G.S. Address can be either a source of information or a destination for information depending on the OP code meaning).
8000	Storage Entry Switches.
8001	Distributor.
8002	Lower Accumulator.
8003	Upper Accumulator.

The 8000 through 8003 addresses make it possible to obtain information from special sources other than drum storage. An 800X address is a source of information and will, therefore, be used either as an "I" address or as a "D" address in conjunction with an OP code which requires information from some source (10's and 60's) and not as the "D" address of a store code.

The two digit operation codes represent built-in operations which the machine can perform.

In the case of the arithmetic codes (10's and 60's) the data address of the instruction word specifies the location of the addend, subtrahend, multiplicand or divisor. In these operations the operand or operator is transferred from its storage location to the distributor. It is then added or subtracted into the accumulator as required by the operation. After the operation is complete the original operand or operator remains in the distributor. The next instruction is taken from the location specified by the "I" address.

In the case of stores codes (20's) the data address specifies the location where the number is to be stored. The information contained in a general storage location is automatically erased as the new number is entered. The stored number is also available in the distributor after completion of the store operation. The contents of the accumulator is not affected by a store operation.

A description of the operation codes follows:

NO OP, No Operation (00)—The data address of this instruction is ignored and no operation is performed. The next instruction is taken from the location specified by the instruction address.

STOP, Stop (01)—This code will stop the program. It may be made inoperative by means of the programmed stop switch on the console.

AU, Add to Upper Half of Accumulator (10)—The contents of the location specified by the data address are added to the contents of the upper half of the accumulator. If the sum should exceed the capacity of the accumulator, overflow digits will be lost but the overflow circuit will be set and will stop the calculator or may be interrogated by means of the branch on overflow code.

SU, Subtract from Upper Half of Accumulator (11)—The contents of the location specified by the data address are subtracted from the contents of the upper half of the accumulator. If the difference should exceed capacity of the accumulator, overflow digits will be lost but the overflow circuit will be set and will stop the calculator or may be interrogated by means of the branch on overflow code.

DIV, Divide (14)—The number contained in the accumulator is divided by the number contained in the location specified by the data address. The quotient will be developed in the lower half of the accumulator with its proper sign and the remainder will be in the upper half of the accumulator with its proper sign. This is the only case (except DIV RU) where the two halves of the accumulator have independent signs. Either half of the accumulator, with its corresponding sign, may be stored.

Division is accomplished by repeated shifting of the accumulator, subtracting the divisor which is in the distributor from the upper half of the accumulator, and counting in the low order position of the lower half of the accumulator. Ten shifts take place, the first shift taking place before subtraction is started. Thus, a zero must be in the high order position of the dividend so that the maximum dividend size is nineteen digits. The divisor and dividend must be lined up so that it is not possible to get more than ten digits in the quotient. Therefore, the dividend may have a maximum of nine digits more than the divisor.

AL, Add to Lower Half of Accumulator (15)—The contents of the location specified by the data address are added to the contents of the lower half of the accumulator. If the sum should exceed the capacity of the lower half of the accumulator, overflow digits are added into the low order position of the upper half of the accumulator.

SL, Subtract from Lower Half of Accumulator (16)—The contents of the location specified by the data address are subtracted from the contents of the lower half of the accumulator. If the difference should exceed capacity of the lower half of the accumulator, overflow digits are added into the low order position of the upper half of the accumulator.

AABL, Add Absolute to Lower Half of Accumulator (17)—The absolute value of the contents of the location specified by the data address is added to the contents of the lower half of the accumulator. If the sum should exceed the capacity of the lower half of the accumulator, overflow digits are added into the low order position of the upper half of the accumulator.

SABL, Subtract Absolute from Lower Half of Accumulator (18)—The absolute value of the contents of the location specified by the data address are subtracted from the contents of the lower half of the accumulator. If this difference should exceed the capacity of the lower half of the accumulator, overflow digits are added into the low order position of the upper half of the accumulator.

MULT, Multiply (19)—The contents of the location specified by the data address is multiplied by the contents of the upper half of the accumulator. The twenty digit product is developed in the accumulator and the multiplier is lost. The lower half of the accumulator must be clear for this operation.

Multiplication is performed as follows: (1) The multiplier is in the upper half of the accumulator as the result of a previous operation. (2) The multiplacand is put in the distributor from storage (part of mult. oper.). (3) The entire accumulator is shifted left one place. During this shift, the high order position of the accumulator, which is the multiplier digit, is read into a blind memory position. The multiplicand is added into the lower half of the accumulator the number of times indicated by this multiplier digit. (4) Step (3) is repeated ten times, once for each multiplier digit. (5) The twenty digit product is given the proper sign according to the signs of the factors.

If the lower half of the accumulator contains some number at the start of a multiplication, the final result will have the absolute value of that number added to the ten high order positions of the absolute value of the product and this result will be given the sign of the product. Thus, much care must be taken to assure that the lower half of the accumulator is clear before a multiplication.

ST L, Store Lower Half of Accumulator (20)—The contents of the lower half of the accumulator and its sign are stored in the location specified by the data address.

ST U, Store Upper Half of Accumulator (21)—The contents of the upper half of the accumulator and its sign are stored in the location specified by the data address.

ST DA, Store Data Address (22)—The digits in positions 8–5 of the distributor are replaced by the corresponding digits in the lower half of the accumulator. The modified number in the distributor is then stored in the location specified by the data address.

ST IA, Store Instruction Address (23)—The digits in position 4–1 of the distributor are replaced by the corresponding digits in the lower half of the accumulator. The modified number in the distributor is then stored in the location specified by the data address.

ST D, Store Distributor (24)—The contents of the distributor are stored in the location specified by the data address.

The units digit of the data address of any shift instruction specifies the magnitude of the shift. The calculator will ignore the higher order digits of the data address and will shift the contents of the entire accumulator only by the amount indicated by the units digit.

The next instruction is taken from the location specified by the instruction address. The contents of the distributor are not affected by these operations. Any digits shifted off are lost and no indication is given if this should occur.

SRT, Shift Right (30)—The contents of the accumulator are shifted to the right the number of places specified by the units digit of the data address. A "0" indicates a shift of 0, a "1" indicates a shift of 1, etc.

SRD, Shift and Round (31)—Digits 1 through 9 in the units position of the data address indicate that a 5 is added or subtracted into the corresponding position of the lower half of the accumulator and the entire accumulator is then shifted right by that amount. A "0" in the units position of the data address indicates that a 5 is added or subtracted into the 10th or high order position of the lower half of the accumulator and the entire accumulator is shifted right 10 places.

This operation effectively rounds the number in the accumulator in the position indicated by the units position of the data address and then shifts the accumulator so that the units digit of the rounded number is in the low order position of the lower half of the accumulator.

SLT, Shift Left (35)—The contents of the accumulator are shifted to the left the number of places specified by the units digit of the data address. A "0" indicates a shift of 0, a "1" indicates shift of 1, etc.

SCT, Shift and Count (36)—The data address of this instruction has no effect on the operation performed. The entire accumulator is shifted left until the most significant digit is in the high order position of the accumulator. The number of shifts necessary to do this is counted and this count is stored in the low order position of the accumulator. A count above 9 will set the overflow circuit and turn on the overflow light. The overflow circuit may then be interrogated by means of the BR OV code. A count above 9 will give 0 as the count number and only 9 shifts will be performed.

The shift and count operation is very helpful for floating point operations.

The branching operations do not affect the contents of the distributor or accumulator.

BRNZU, Branch on Non-Zero in Upper Half of Accumulator (44)—The contents of the upper half of the accumulator are examined. If it is zero, the next instruction is taken from the location specified by the instruction address. If it is not zero, the next instruction is taken from the location specified by the data address.

BRNZ, Branch on Non-Zero in Accumulator (45)—The contents of the entire accumulator are examined. If it is zero, the next instruction is taken from the location specified by the instruction address. If it is not zero, the next instruction is taken from the location specified by the data address.

BR MIN, Branch On Minus (46)—The sign of the accumulator is examined. If it is positive, the next instruction is taken from the location specified by the instruction address. If it is negative, the next instruction is taken from the location specified by the data address.

BR OV, Branch On Overflow (47)—Examine the condition of the overflow circuit. If it is not set, indicating that no overflow has occurred, then the next instruction is taken from the location specified by the instruction address. If it is set, indicating that an overflow has occurred, the next instruction is taken from the location specified by the data address.

RAU, Reset-Add to Upper Half of Accumulator (60)—Reset the entire accumulator to zero and add the contents of the location specified by the data address to the upper half of the accumulator.

RSU, Reset-Subtract from Upper Half of Accumulator (61)—Reset the entire accumulator to zero and subtract the contents of the location specified by the data address from the upper half of the accumulator.

DIV RU, Divide-Reset Upper Half of Accumulator (64)—This is the same as DIV (see OP 14) except that the upper half of the accumulator is reset at the end of the operation. This effectively clears the remainder.

RAL, Reset-Add to Lower Half of Accumulator (65)—Reset the entire accumulator to zero and add the contents of the location specified by the data address to the lower half of the accumulator.

RSL, Reset-Subtract from Lower Half of Accumulator (66)—Reset the entire accumulator to zero and subtract the contents of the location specified by the data address from the lower half of the accumulator.

RAABL, Reset-Add Absolute to Lower Half of Accumulator (67)—The entire accumulator is reset to zero and the absolute value of the contents of the location specified by the data address is added to the lower half of the accumulator.

RSABL, Reset-Subtract Absolute from Lower Half of Accumulator (68)—The entire accumulator is reset to zero and the absolute value of the contents of the location specified by the data address is subtracted from the lower half of the accumulator.

LD, Load Distributor (69)—The contents of the location specified by the data address are loaded into the distributor.

The store operations have the common characteristic that the number stored is also in the distributor after completion of the operation. The number contained in a memory location is automatically erased before new information is read into it. The contents of the accumulator is not affected by this operation. The data address specifies the location where the number is to be stored. The next instruction is taken from the location specified by the instruction address.

RD, Read One Card (70)—There are 40 groups of ten input registers in the main memory addressed 0000–0009, 0050–0059, 0100–0109, 0150–0159, etc. The data address of a read instruction refers to a group of ten registers by addressing the storage band of one of these groups of registers. The information in the ten input buffer storage registers is transferred to the group of ten input registers in the band addressed by the instruction. The next card is then read by the card reader and the data punched on that card is read into the input buffer storage.

Calculation may proceed as soon as the information is transferred from the input buffer storage to the input registers. Thus a read operation does not hold up calculations for the entire 300 milliseconds required to read a card but only for the small amount of time required for the above transfer of information.

If a second read operation is called for before the first has been completed, no errors are caused since the calculator simply holds up until the first read operation has been completed and then proceeds to read the second card.

PCH, Punch One Card (71)—There are 40 groups of ten punch registers in the main memory addressed 0026–0035, 0076–0085, 0126–0135, etc. The data address of a punch instruction refers to one of the groups of ten registers by addressing the storage band of the group. The information in the ten punch registers is transferred to the ten punch buffer storage registers. The contents of the punch buffer storage is then punched on a card.

Calculation may proceed as soon as the information is transferred from the punch registers to the punch buffer storage. Thus a punch operation does not hold up calculations for the 600 milliseconds required to punch a card but only for the small amount of time required for the above transfer of information.

If a second punch operation is called for before the first has been completed, no errors are caused since the calculator simply holds up until the first punch operation has been completed and then proceeds to punch the second card.

TLU, Table Look-Up (84)—Tables are stored in the memory in the form of an argument and functions of the argument. The table look-up operation will cause the calculator to scan the table to find the first argument that is greater or equal to a given argument.

The data address of the TLU instructions is the address of the first table argument. Completion of this operation will result in the address of the first table argument that is greater or equal to the given argument being in positions 8–5 of the lower half of the accumulator. This address may then be used to modify instructions which will pull the desired function of that argument from the table.

BR D 10, Branch on 8 in Tenth Position of Distributor (90)—Examine the tenth or high order position of the distributor. If this position contains a digit 9, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 8, the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

BR D 1, Branch on 8 in First Position of Distributor (91)—Examine the first or low order position of the distributor. If this position contains a digit 9, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 8, the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

BR D 2, Branch on 8 in Second Position of Distributor (92)—Examine the second position of the distributor. If this position contains a digit 9, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 8, the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

BR D 3, Branch on 8 in Third Position of Distributor (93)—Examine the third position of the distributor. If this position contains a digit 9, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 8, the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

BR D 4, Branch on 8 in Fourth Position of Distributor (94)—Examine the contents of the fourth position of the distributor. If this position contains a digit 8, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 9, the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

BR D 5, Branch on 8 in Fifth Position of Distributor (95)—Examine the contents of the fifth position of the distributor. If this position contains a digit 9, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 8 the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

BR D 6, Branch on 8 in Sixth Position of Distributor (96)—Examine the contents of the sixth position of the distributor. If this position contains a digit 9, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 8, the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

BR D 7, Branch on 8 in Seventh Position of Distributor (97)—Examine the contents of the seventh position of the distributor. If this position contains a digit 9, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 8, the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

BR D 8, Branch on 8 in Eighth Position of Distributor (98)—Examine the contents of the eighth position of the distributor. If this position contains a digit 9, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 8, the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

BR D 9, Branch on 8 in Ninth Position of Distributor (99)—Examine the contents of the ninth position of the distributor. If this position contains a digit 9, the next instruction is taken from the location specified by the instruction address. If this position contains a digit 8, the next instruction is taken from the location specified by the data address. If this position contains a digit other than an 8 or 9, it is an error and the calculator will stop.

Summary of operation codes

The following summary of OP codes is provided, without the detailed description, for easy reference.

As an aid in understanding the program control circuits which will be considered in a later section, notice that the operation codes can be grouped into 7 classifications according to their meaning. Also notice that each of the codes in one of these 7 classifications has a common ten's position numerical value. Within the machine, the codes and address appear in their bi-quinary form in the OP code and address registers where they are interpreted.

0's	NO OP and STOP.
10's and 60's	Arithmetic codes. The *binary* value of the ten's position determines the reset-no reset meaning of the codes. The quinary part of the ten's position and all of the units position are the same.
20's	Store codes.
30's	Shift codes.
40's and 90's	Test codes. Here the binary value of the ten's positions determines whether the accumulator or the distributor is to be tested.
70's	Read and punch codes.
84	Table look-up.

Code	Abbreviation	Operation
00	NO OP	No Operation.
01	STOP	Stop.
10	AU	Add to Upper.
11	SU	Subtract from Upper.
14	DIV	Divide.
15	AL	Add to Lower.
16	SL	Subtract from Lower.
17	AABL	Add Absolute Value to Lower.
18	SABL	Subtract Absolute Value from Lower.
19	MULT	Multiply.
20	STL	Store Lower.
21	STU	Store Upper.
22	STDA	Store Lower Data Address.
23	STIA	Store Lower Instruction Address.
24	STD	Store Distributor.
30	SRT	Shift Right.
31	SRD	Shift and Round.
35	SLT	Shift Left.
36	SCT	Shift Left and Count.
44	BRNZU	Branch on Non-Zero in Upper.
45	BRNZ	Branch on Non-Zero.
46	BR MIN	Branch on Minus.
47	BR OV	Branch on Overflow.
60	RAU	Reset-Add to Upper.
61	RSU	Reset-Subtract from Upper.
64	DIV RU	Divide-Reset Remainder.
65	RAL	Reset-Add to Lower.
66	RSL	Reset-Subtract from Lower.
67	RAABL	Reset-Add Absolute Value to Lower.
68	RSABL	Reset-Subtract Absolute Value from Lower.
69	LD	Load Distributor.
70	RD	Read One Card.
71	PCH	Punch One Card.
84	TLU	Table Look-Up.
90	BRD 10	Branch on 8 in 10th Position of Distributor.
91	BRD 1	Branch on 8 in 1st Position of Distributor.
92-99	BRD 2-9	Branch on 8 in 2nd through 9th Position of Distributor.

Examples of programming

To explain the use of the addresses and operation codes a simple problem is shown. The following chart shows the lines of coding which might be used to solve the problem

$$\frac{(A+B-C)D}{E}$$

The starting address of the program is 0056.

Loc.	Inst. or Data	Description
0051	0000000291	Data, Factor A
0052	0000000842	Data, Factor B
0053	0000000133	Data, Factor C
0054	0000000020	Data, Factor D
0055	0000000200	Data, Factor E
0056	6000510057	RAU. A in U (291)
0057	1000520058	AU. $A+B$ in U (10123)
0058	1100530101	SU. $A+B-C$ in U (10000)
0101	1900540102	MULT. $(A+B-C)D$ in L (200000)
0102	6400550103	DIV RU $\frac{(A+B-C)D}{E}$ in L (1000)
0103	2000270104	ST L (1000) in Card Punch Out G.S.
0104	7100270105	Punch Card (1000 W_1 Field)
0105	0100000000	Stop

In this example the values of the factors are:

$$A=291$$
$$B=842$$
$$C=133$$
$$D=20$$
$$E=200$$

Then

$$\frac{(A+B-C)D}{E}=\frac{(291+842-133)20}{200}=100$$

The first five locations contain the data to be used. (Factors A, B, C, D, E). The other locations contain the instructions to be performed. The sequence in which the instruction locations appear as "I" addresses determines the sequence of the program routine.

On a simple problem such as this one, loading could be done from the operator's console by setting the word values in the storage selection switches and the locations in the address switches and entering each word on the chart into its programmed location, line by line.

This would be a slow process, however, when the problem is long or when programming for many different problems must be loaded frequently. In practice the "words" to be stored may be punched in loading cards and placed in their proper locations by means of a pre-stored loading program or loading routine.

Subroutines

The basic arithmetic functions of Addition, Subtraction, Multiplication and Division are built into the machine and can be performed by the use of an OP code calling for the desired operation. Often, however, a problem requires that values be obtained for more complex functions, which are not built into the machine but, which must be arrived at by programming a sequence of basic operations in accordance with some procedure which will produce the desired function value. Examples of such functions are $\sqrt{N}$, trigonometric functions, logarithmic functions, etc.

When it is known that values for one of these types of functions will be required frequently in the solution of many different problems, the machine can be given the ability to compute the function value by means of a pre-stored subroutine. The program for the subroutine is developed and loaded. These storage locations then are never used for other information so that the subroutine is always available for use on any problem. This relieves the programmer of the need to plan these steps for each different problem. It is only necessary for him to branch into and out of the subroutine from and to the main program routine whenever he needs such a function value.

Such a subroutine is programmed and loaded in the same manner as any problem. Once planned and loaded it is only necessary for the programmer to know the starting address of the subroutine, the "I" address of its last line of coding and the address where the computed value is stored by the subroutine. With this information the problem can enter and leave the subroutine as often as necessary.

Only two extra "linkage" program steps are needed to enter and leave the subroutine from the main routine. These are coded as a part of the main routine whenever a point is reached in the programming where the subroutine is to be used. These steps transfer the instruction for the next step of the main program following the subroutine from its main program location to the "I" address of the last step in the subroutine and provide the means of getting back into the main program after the subroutine is used.

The calculation of $\sqrt{N}$ is used to illustrate a subroutine and as an example of how the machine may perform other than the four basic arithmetic functions. In this program, N is placed in the lower accumulator by one of the last steps in the main routine, for use by this routine.

The development of techniques for machine computation of complex functions is a subject in itself, in the broader field of numerical analysis. The development of a method for the calculation of $\sqrt{N}$ is presented here as a simple example of such a technique.

To obtain the $\sqrt{N}$, assume that X_0 approaches the value of $\sqrt{N}$.

Then

$$X_0 \cdot X_0 = X_0^2 \cong N$$

Let

$$\frac{N}{X_0} = X_a$$

Then

$$X_0 \cdot X_a = N$$

If N is represented as a point N units from 0, then $\sqrt{N}$ lies between N and 0. X_0, the approximation of the $\sqrt{N}$ lies between N and 0, although this not necessary since X_0 could be greater than N. Also in this case X_0 is assumed to be greater than $\sqrt{N}$ although it could just as well be less. Under these conditions X_a which is the quotient of $N \div X_0$ lies between $\sqrt{N}$ and 0. The object of this procedure is to make successive adjustments to the value represented by X_0, with consequent adjustments to the value represented by X_a, until the two values are equal, while subject to the restriction that their product must be equal to N. This is done by taking the average of the sum of the two values and repeating the above process, using the average value so obtained in place of the preceding approximation, until the two are equal.

Thus the first approximation is obtained by adding X_a+X_0 or

$$\left(\frac{N}{X_0}+X_0\right)$$

then dividing by two, so that X_1, the first approximation is

$$\frac{\frac{N}{X_0}+X_0}{2}$$

The process is repeated using X_1 in place of X_0 and obtaining X_b as the quotient of $N \div X_1$. Then X_2, the second approximation, is

$$\frac{X_b+X_1}{2}$$

or

$$\frac{\frac{N}{X_1}+X_1}{2}$$

Several repetitions will be required depending upon the accuracy of the initial approximation, X_0, and the decimal accuracy desired.

This procedure provides a way which can be used to program a machine which can basically only add, subtract, multiply and divide so that it can, by a proper sequence of these basic operations, provide square roots as needed.

Two linkage steps are used for entering and leaving this pre-stored subroutine for producing a square root from the main program routine. In this case location 0569 is, for example, where the next instruction of the main routine is stored. Location 0050 is, for example, the "I" address of the last instruction of the subroutine. The linkage steps serve to transfer the contents of 0569 to 0050, by way of the distributor. Then upon completion of the subroutine the next instruction, taken from 0050, will be the same as if it had been taken from 0569.

The $\sqrt{N}$ subroutine explained above may be used as a part of the programming for the solution of quadratic equations with real roots.

The procedure which the program sequence follows may be the well-known formula for the solution of quadratics, e.g.,

$$X = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

where

a=coefficient of the X^2 term of the equation
b=coefficient of the X term of the equation
c=constant term of the equation

Optimum programming

To obtain information from, or place information in, a storage location when a cyclical storage system is used, such as the magnetic drum of the present machine, it is necessary to wait for the time when the desired storage location passes under a reading or writing position. By analysis of the timing sequence of the program control circuits for the operations which this machine performs, procedure may be determined by which data and instruction may be located in such a way as to eliminate or minimize this waiting time. In this way the storage access time can be effectively reduced. The techniques for applying these procedures are called optimum programming.

With the present machine the programmer is not burdened with timing restrictions which he must always keep in mind, because interlocks make it impossible to violate timing conditions in such a way as to cause the machine to give erroneous results. If, however, proper recognition and analysis of the sequence of events occurring within the machine is made, the overall speed can be significantly increased by properly locating data and instructions.

The basic cycle of the machine is the "word time" or time interval required to read or write one word. As there are 50 locations around the drum where words can be written, each word time is equal to 1/50 of a drum revolution. With the drum revolving at 12,500 revolutions per minute, a word time is equal to .096 millisecond. Each of the machines coded operations can be analyzed in terms of word times. The object here is to analyze each operation to see how many word times are required in its interpretation and execution. In this manner the basic or fundamental word times are determined and from these a set of rules is derived through which optimum programming may be effected.

Angular drum locations are all that must be considered when optimum coding. The machine is completely synchronous so that the words located in equivalent angular positions are in phase and are equivalent from a timing viewpoint. Thus addresses 50 words apart are equivalent, so that, for example, addresses 0003, 0053, 0103 . . . , 1903, 1953 are in phase and therefore, equivalent so far as optimum programming is concerned.

The "D–I" address system used in the machine greatly facilitates optimum programming. Data and instructions may be in any locations on the drum and instructions may be taken in any desired sequence. As mentioned above, once the optimum location has been determined there are 40 different locations in which the word of data or instustions could be placed since the 40 bands are in phase. Also, an instruction location may be optimum if it is within a range of angular locations, for those operations where the search for the next instruction is made concurrently with the execution of the operation.

A number of functions in the machine may be performed simultaneously. Most operations branch into parallel execution shortly after the data is made available to the arithmetic unit. One branch is the arithmetic process and the other the obtaining of the next instruction.

Analyzing the steps carried out by the machine in the performance in any operation, it may be seen that, in every case, certain fundamental word times are required.

Begin consideration of each instruction at the time when the instruction has been located but has not entered the program register. Starting at this point, the first word time of every operation is used to transfer the instruction from its storage location to the program register. The next word time (restart position of control commutator) is used to initiate the interpretation of this instruction. This is done by resetting OP and address registers and transferring the data address to the address register and the operation code to the operation code register. The steps performed during the word times beyond this point will depend upon the particular instruction being executed.

Consider the shift right 3 operation. A sequence chart for this particular operation is shown in Fig. 9. Assume that the shift instruction is in location n and that n is an odd word. During word time n, the instruction is read into the program register. During word time $n+1$ the operation code (30) is transferred to the operation register and the data address (0003) is transferred to the address register. Word time $n+2$, Enable Shift Control, is used to set up the necessary control circuits to perform the shifting. Since $n+3$ is even, the next cycle, Wait for Even, is skipped and shifting begins immediately. At this point parallel operation begins and the actual shifting process on the right branch occurs simultaneously with the process of searching for and obtaining the next instruction indicated on the left branch.

The restart signal which occurs during word time $n+3$ indicates that the data address in the address register is no longer needed. During word time $n+4$ the instruction address is transferred to the address register. Word time $n+5$ is the Enable Program Register Read In Cycle which indicates that a new instruction is to be read into the program register. Thus, beginning with word time $n+6$ the machine starts searching for the next instruction. This search may require from 0 word times to as many as 49 word times depending on the location of the next instruction.

Examining the arithmetic branch, it is seen that during word times $n+3$ through $n+8$ the actual shifting takes place. The interlock at point A is removed during word time $n+9$. The function of this interlock will be explained later.

It is seen that although the arithmetic portion of the operation is not completed until word time $n+9$, the program register is able to accept the next instruction at word time $n+6$. If the next instruction were placed in a location corresponding to word times $n+6$, $n+7$, or $n+8$, it would be read into the program register before completion of the arithmetic portion of the shift operation. In addition to this, interpretation of this instruction would begin in that the operation code would be transferred to the operation register and the data address transferred to the address register. Thus, the first two cycles of the interpretation of the next instruction occur in parallel with the execution of the shift instruction. At this point, interpretation must cease since further steps may make use of control circuits or portions of the arithmetic unit which are already in use (unless it is a read or punch instruction). The interlock at A is provided to insure that interpretation of the next instruction does not proceed past this point.

If the next instruction were placed in a location corresponding to word time $n+9$, the instruction would be read into the program register during the completion of the arithmetic portion of the shift operation. In this case only the first cycle of the interpretation of the next instruction occurs in parallel with the execution of the shift instruction.

If the next instruction were placed in a location corresponding to word time $n+10$, it would be read into the program register immediately upon completion of the shift operation.

Each of the locations corresponding to word times $n+6$ through $n+10$ effectively reduces the search time to 0 since there is no waiting time between completion of the shifting operation and the obtaining of the next instruction. However, if the next instruction were placed in a location corresponding to word times $n+6$ through $n+9$, the interpretation of the next instruction would begin in that the instruction would be read into the program register. If it were restricted to a location corresponding to word times $n+6$ through $n+8$, two cycles of interpretation of the next instruction would take place.

Therefore, not only can the search for the next instruction take place during completion of the arithmetic portion of an operation, but also interpretation of this next instruction can begin. If the next instruction calls for a read or punch operation, execution of either of these operations can proceed since they do not require use of the arithmetic unit. In this case the interlock at A does not stop interpretation of the next instruction after the first 2 cycles, since read and punch instructions by-pass this interlock.

There are three classes of operations. They are read, punch and calculate operations. Correspondingly, there are three interlocks. The read interlock stops execution of the following read instructions, until completion of the previous read instruction. If the following instructions are punch or calculate operations, however, they may proceed without delay. The punch interlock stops execution of the following punch instructions, until completion of the previous punch instruction. If the following instructions are read or calculate operations however, they may proceed without delay. The calculate interlock at A in Fig. 9 stops execution of the following calculation instructions until completion of the previous arithmetic operation. If the following instructions are read or punch instructions however, they may proceed without delay. Thus simultaneous reading, punching, and calculating is possible.

Since read or punch instructions are not held up by the interlock at A, they should be placed in a location corresponding to word time $n+6$. If the next instruction is another arithmetic operation, there is no advantage to $n+6$ or $n+7$ over $n+8$, since the interlock at A causes the machine to wait for completion of the shifting process.

Since the storage entry switches (8000), distributor (8001), lower accumulator (8002) and upper accumulator (8003) are addressable, special consideration must be given to the cases where one of these locations is addressed. The storage entry switches and distributor are always immediately accessible. However, the lower accumulator can be read into or out of only during an even word time and the upper accumulator can be read into or out of only during an odd word time. Thus, for purposes of optimum coding the storage entry switches and the distributor may be treated as being equivalent to any address, the lower accumulator as equivalent to an even address and the upper accumulator as equivalent to an odd address.

If, for example, a reset add upper operation (or any add type operation) has a data address of 8000 or 8001, these addresses can be treated as being equivalent to $n+3$ and the instruction address of that instruction can be determined accordingly. If a data address of 8002 were used, and the instruction were in an even location, the optimum location of data for that instruction would be $n+3$, which is odd in this case. Therefore, an extra cycle must be taken to wait for an even location so that the lower accumulator may be read out. Since the effective data address is then even, the rule $i=d+5$ must be used to determine the location of the next instruction. Similar analyses may be made for each of the other cases.

Since the accumulator may be in use when an "I" address of 8002 or 8003 is given, it would be possible to take the next instruction from the accumulator before the arithmetic operation was completed. For example, if an 8002 "I" address were used on a multiply operation, one of the partial products might be taken as the next instruction rather than the final product. For this reason, the interlock at point A is effective in these cases to prevent the next instruction from being taken from an 800X address until completion of the arithmetic portion of the operation.

BASIC ELECTRONIC CIRCUITS

The basic functional electronic circuits used in the machine are relatively few and simple. The computing circuits are designed to employ twin triode tubes and germanium crystal diodes. In the computing circuits tubes are used functionally as inverters, double inverters and cathode followers. Elsewhere in the machine triodes are used as voltage amplifiers, shaping amplifiers, power units and single shot multivibrators. Thyratrons are employed in the read-out circuitry. Inverters and cathode followers are connected by external circuitry to form a "latch" circuit. Latches are used rather than triggers where a momentary signal must be stored for later use and in timing ring circuits.

The logical circuits of the machine make extensive use of germanium crystal diodes arranged to act as positive signal AND circuits and positive signal OR circuits. Hereafter, to avoid confusion due to multiple usage of the terms AND and OR, AND circuits will be referred to as *switches,* and OR circuits will be called *mixes.*

Crystal diodes are also used occasionally to clamp tube grid voltages at a fixed level.

Double diode vacuum tubes are used in some circuits where applied voltages are too severe for crystal diodes or where the back resistance of a crystal diode is too low for proper back circuit elimination.

Capacitor storage is used for accumulator storage, distributor and program register where rapid access to the stored information is essential. In the capacitor storage unit a capacitor and two associated vacuum diodes form a basic binary storage device, combinations of which are used to build up the one or two word storage unit.

All tubes are part of pluggable units. Units are named according to function. Inverter units are identified with an IN prefix such as IN1, IN3, etc. Cathode followers, power units, voltage amplifiers, shaping amplifiers, diode units and capacitor storage units are similarly identified with the prefixes CF, PW, VA, SA, DD and CS, respectively.

Several types of CF, IN and PW units are provided. Use of a particular type depends on the circuit requirements.

Several types of cathode follower units provide different load resistance values, various types of grid divider and input arrangements, etc. Many cathode follower units, such as the CF1 and the CF2 are duplicates except for the use of the 6350 tube instead of the 5965 to provide more drive.

Germanium crystal diode switch and mix circuits are also built up in pluggable units. The associated resistors used in the switch or mix circuits are connected externally.

To simplify the logical representation of circuits, a system of block diagrams is used. Each block symbol represents a basic function. An effort has been made to identify the functional classification of a unit by the configuration of its block symbol. The unit type identification as well as the characteristic block symbol is shown for cross referencing to the detailed unit drawings.

The most common tube functions are single inverters and cathode followers, both single triode functions. These are shown as a square block with grid input on the left and anode or cathode output from top or bottom as the case may be. In each case these are half of a pluggable unit. When both halves of a unit are used for a single function, as in double inverters, the block is a rectangle with twice the area of a half unit. Double diode pluggable units contain two, double vacuum diode tubes or four diodes. Thus a DD functional block is a rectangle, one half the area of a half unit block.

A single latch is made up of a double inverter and a cathode follower. Connections between the inverter output anode and the cathode follower grid is not included every time a latch symbol appears. Thus a latch symbol is an IN rectangle and a CF square joined as shown in Figs. 49*a* and 50*a*. Similarly, a double latch is an inverter plus two CF's, one on each side. Where latch input circuitry utilizes diode switches and/or mixes, these will be shown as separate diode blocks. These distinctive latch configurations will help to locate latches on the wiring diagrams. Since latches are usually key components of a logical circuit, this should aid in following the wiring diagram.

On latch symbols grid inputs are shown entering the block from the bottom (left or right) instead of the left side as on other tube symbols in order to make room for the adjoining cathode followers.

In all other cases grid inputs are straight in from the left if there is no grid voltage divider network built into the unit. When a divider is a part of the unit, the grid input terminal is shown above the grid level as in the CF3 unit.

When a unit has a built-in load resistor this is indicated by a small box in the lower left corner of the block if it is a CF or the upper left corner if it is an IN. If the load resistor is used, as is usually the case, the number in the box is the code number for the service voltage to which the load resistor is connected. These voltage codes are listed at the end of this section.

Special input circuitry such as capacitors, and diode clamps are shown as in the CF9, 13 and 16. Where a cathode load resistor for a preceding stage, which contains no built-in resistor of its own, is in the input of the next stage, it is shown as in the CF12.

In some cases where it is helpful to know that a tube element connects to an unusual voltage, this is shown. An example is the cathode of an IN5 which connects to —20 rather than ground.

In general, power units are shown as a block representative of their basic function as either an inverter or cathode follower. Special power units like the PW2 and 3 have their own special block representation.

Voltage amplifiers and shaping amplifiers are special types of double inverters.

A crystal diode pluggable unit may comprise several logical switch and/or mix functions. Each switch or mix function contained in a unit is shown as a separate switch or mix block.

To distinguish crystal diode functions from tubes, the shape and size of the blocks have been made distinctive. Further, to distinguish between a diode switch function and a diode mix function, the blocks are shown in distinctive positions. The switch is shown as a long, low block with inputs to the bottom and output from the top. The mix is shown as a narrow, high block with inputs to the left side and output from the top.

Voltage codes:

1–W	250 for Magnetic Storage Write Circuits.
1–R	250 for Magnetic Storage Read Circuits.
2	150.
3	70.
4	—20.
5	—35 (Tap on —50 supply).
6	50.
7	—50 Unregulated protective bias supply.
8	—50.
9	—70.
10	—250.

DIODE SWITCH AND MIX CIRCUITS

A diode switch is a circuit wherein the coincidence of two or more signals is required to produce an output. Fig. 13 shows a pair of diodes connected to form part of a switch for operating with positive going signals. As used here and throughout this disclosure, the triangular portion of the diode symbol represents the anode while the flat portion of the symbol represents the cathode. If terminal **201** is connected through a resistor to a positive source of potential, a complete switch is formed. This resistor is not shown as a part of this functional unit since frequently a switch is comprised of more than two diodes to operate on the coincidence of more than two signals. Assume that signal A at terminal **202** is −35 volts and signal B at terminal **203** is +10 volts. The source impedance of signal A in series with the forward resistance of the diode is sufficiently low in comparison to the resistor to hold terminal **201** at a minus potential. The back resistance of the other diode is high enough to prevent signal B from raising the potential at terminal **201**. Thus terminal **201** will rise to a positive potential only when both signals A and B are at positive potentials.

The block symbol used in the wiring diagrams to represent the circuit of Fig. 13 is shown in Fig. 13*a* with corresponding terminals bearing the same reference numerals.

A diode mix is a circuit wherein the appearance of a signal on any one or more inputs will produce an output. Fig. 14 shows a pair of diodes connected to form part of a mix circuit for operating with positive going signals. If terminal **204** is connected through a resistor to a negative source of potential, a complete mix circuit is formed. Here again the resistor is not shown as a part of the functional unit since frequently a mix circuit is comprised of more than two diodes.

In the mix circuit a positive signal A applied to terminal **205** will raise terminal **204** to approximately the signal potential (10 volts) since the combined diode forward resistance and the signal A source impedance is small in comparison to the resistor. The negative signal B applied to terminal **206** has no effect on the potential at terminal **204** since its diode back resistance is high and provides a means of dropping the potential difference which exists between the two signals. Under these conditions a positive signal on either the terminal **205** or **206** will raise the potential of terminal **204**.

The block symbol used to represent the circuit of Fig. 14 is shown in Fig. 14*a* with corresponding terminals bearing the same reference numerals.

In either the switch or the mix circuits the potential of the output terminal is determined by that branch of the switch or mix which has the larger potential across it when both are not the same.

Fig. 15 shows the diode configuration of a combination switch and mix and Fig. 15*a* shows the block symbols used to represent this configuration with corresponding reference numerals appearing on corresponding terminals.

Fig. 16 shows the symbol used for a crystal diode employed as a clamp.

Vacuum diodes are used in several places in the machine where loads would be too severe for crystal diodes or where higher back resistances are needed for better isolation. The 6AL5, double diode type, may be used for all vacuum diode applications.

Fig. 17 shows one half of a 6AL5 tube, and Fig. 17*a* shows the block symbol used to represent it with corresponding terminals bearing corresponding reference numerals.

Capacitor storage

Capacitor storage units are used in the machine for "working" storage, where the information must be available frequently and rapidly, for calculating and control purposes. These capacitor storage units (accumulator, distributor and program step) are built up from basic capacitor storage "binary cells," each of which is capable of storing one binary bit by virtue of its two possible states, namely charged and discharged.

One such capacitor storage device is shown in Fig. 18. The basic cell consists of a pair of diodes **214** and **215** and the storage capacitor **216**. The block symbol used to represent the circuit of Fig. 18 is shown at Fig. 18*a* with corresponding terminals bearing corresponding reference numerals.

Cathode followers

A cathode follower is a tube, usually a triode, with circuitry arranged so that the output signal is taken from the cathode end of a cathode load resistor. As its name suggests, the output signal follows or is in phase with the grid input signal. A cathode follower is inherently a negative feed-back device, i.e., for values of grid voltage between cut-off and saturation, as the grid voltage is increased the cathode voltage also increases due to the increased IR drop across the cathode load resistor. The net result is a self-biasing action which limits the amplitude of the output signal to a value very nearly equal to the amplitude of the input signal. The amount of bias developed depends upon the tube characteristics and the load resistor value.

In any case the cathode reaches an equilibrium potential more positive than the grid. Thus a signal will gradually have its levels raised as it passes through a series of cathode followers. Cathode follower load resistors need not be as large as those used with inverters, since the value of the load resistor has less effect on the amplitude of the output signal. Thus a cathode follower can be a relatively low impedance device whose output signal is less susceptible to distortion by associated circuit capacitance. In this machine the voltage levels between which a cathode follower output signal swings are commonly of the order of −35 to +10.

An important point in connection with cathode followers as used in this machine is that, unlike inverters, the tube is always conducting whether the grid is at the no-signal level (−35) or the signal level (+10), since the cathode resistor is returned to a voltage value more negative than the no-signal level.

Cathode followers are frequently used with double inverters for D.C. signal level restoration. They are also often used to give a signal, whose levels are still usable, more drive after it has been attenuated by passage through a series of diodes.

Two cathode followers may share a common load resistor and thus form a logical OR circuit.

Fig. 19 shows a simple type of cathode follower wherein a signal is applied to the grid of one half of a type 5965 tube through terminal **221** and the output is taken from the cathode through terminal **222**. Fig. 19*a* shows the block symbol used to represent the cathode follower of Fig. 19 with corresponding reference numerals denoting corresponding terminals. Thus when the block symbol of Fig. 19*a* is seen in the wiring diagram the details of the circuit may be ascertained by reference to Fig. 19.

Fig. 20 shows a cathode follower like that of Fig. 19 with the exception of the tube type and resistance values. Fig. 20*a* shows the block symbol for the cathode follower of Fig. 20.

Fig. 21 shows a cathode follower having a voltage level establishing network in its input circuit. Fig. 21*a* shows the block symbol for the cathode follower of Fig. 21.

Fig. 22 shows a cathode follower similar to that of Fig. 21. Fig. 22*a* shows the block symbol for the cathode follower of Fig. 22.

Fig. 23 shows two cathode followers with slightly different input arrangements sharing a single twin triode tube. Figs. 23*a* and 23*b* show the block symbols for the cathode followers of Fig. 23 where corresponding reference numerals refer to corresponding terminals.

Fig. 24 shows two cathode followers similar to those of Fig. 23. Figs. 24*a* and 24*b* show the block symbols for the cathode followers of Fig. 24.

Fig. 25 shows another type of cathode follower. Fig. 25*a* shows the block symbol for the cathode follower of Fig. 25.

Fig. 26 shows two other types of cathode followers sharing a single twin triode tube. Figs. 26*a* and 26*b* show the block symbols for the cathode followers of Fig. 26.

Fig. 27 shows a pair of cathode followers sharing a common cathode resistor. Fig. 27*a* shows the block symbol for the cathode followers of Fig. 27.

Fig. 28 shows a cathode follower wherein the cathode resistor is not included with the unit. Fig. 28*a* shows the block symbol for the cathode follower of Fig. 28.

Fig. 29 shows still another type of cathode follower, and Fig. 29*a* shows the block symbol for the cathode follower of Fig. 29.

Figs. 30 and 31 show two more types of cathode followers. Figs. 30*a* and 31*a* respectively show the block symbols for the cathode followers of Figs. 30 and 31.

Fig. 32 shows two other types of cathode followers sharing a single twin triode vacuum tube. Figs. 32*a* and 32*b* show the block symbols for the cathode followers of Fig. 32.

Fig. 33 shows another type of cathode follower with a neon glow tube in its input circuit. Fig. 33*a* shows the block symbol for the cathode follower of Fig. 33.

Inverters and double inverters

The basic characteristic of an inverter is implied in its name. A signal applied to its grid is reversed in polarity or inverted at the plate end of its load resistor. The signal is usually also amplified in the process because of the tube's characteristics. The voltage levels, between which the inverter's output signal swings, are relatively high. (Typical values are +150 to +50 although lower amplitudes are often obtained by tapping the plate load resistor.) The load resistance, across which the output signal is developed, is relatively high (typical value, 20K). Thus the inverter is a high impedance device whose output signal is rather easily distorted by associated circuit capacitance.

A double inverter is merely two coupled inverter stages where the plate output signal of the first drives the grid of the second. In a double inverter the signal is inverted twice so that the final output is of the same phase as the input signal. The coupling between stages may be direct or capacitive. If direct the output signal has the same duration as the input. If capacitive a pulse output signal is obtained even though the input may be of relatively long duration.

In this machine inverters are largely used to restore the level of signals which have been altered by passage through diodes and cathode followers. The output signal of a double inverter is usually used to operate the grid of a cathode follower, whose output is a low level, low impedance, restored signal.

Fig. 34 shows a double inverter wherein a signal applied to terminal **267** in inverted by the first half of the twin triode tube and applied through a voltage level establishing network to the second half of the tube. Outputs may be taken from terminals **268**, **269** or **271**. Fig. 34*a* shows the block symbol for the double inverter of Fig. 34 where corresponding terminals bear corresponding reference numerals.

Fig. 35 shows a single inverter, and Fig. 35*a* shows the block symbol for the inverter of Fig. 35.

Fig. 36 shows another type of double inverter, and Fig. 36*a* shows the block symbol for the inverter of Fig. 36.

Fig. 37 shows another type of single inverter, and Fig. 37*a* shows the block symbol for the inverter of Fig. 37.

Fig. 38 shows still another type of double inverter, and Fig. 38*a* shows the block symbol for the inverter of Fig. 38.

Fig. 39 shows another type of single inverter, and Fig. 39*a* shows the block symbol for the inverter of Fig. 39.

Power units

A number of units are used in this machine which are called power units. In general a power unit is basically either a cathode follower or an inverter using a tube type designed to deliver more power to the circuits which it drives.

One rather special type of power unit (PW2) is found in the magnetic storage writing circuits. It is a two stage unit consisting of an inverter, with a plate peaking inductor and a capacitor coupled output cathode follower. The purpose of this unit is to develop an extremely sharp, short duration signal for controlling the tube which pulses the write coils so that a short, sharp pulse of write current will be forced through the coil whenever a magnetic spot is to be placed on the drum surface.

Fig. 40 shows a power unit of the inverter type. Fig. 40*a* shows the block symbol for the power unit of Fig. 40.

Fig. 41 shows the details of the power unit (PW2) mentioned above. Fig. 41*a* shows the block symbol for the power unit of Fig. 41.

Fig. 42 shows another type of power unit of the cathode follower type having a common plate resistor for two cathode follower units. Fig. 42*a* shows the block symbol for the power unit of Fig. 42.

Fig. 43 shows a pair of similar power units of the inverter type, one of which has provision for capacitive input. Figs. 43*a* and 43*b* show the block symbols for the power units of Fig. 43.

Figs. 44 and 45 show two additional types of power units, and Figs. 44*a* and 45*a* respectively show the block symbols for the power units of Figs. 44 and 45.

Shaping amplifier

In the magnetic storage read circuitry, voltage amplifier outputs are applied to a special type of circuit called a shaping amplifier. The main function of the shaping amplifier is to alter the shape of the voltage amplified, e.g., a sinusoidal head signal, so that it more nearly approaches a square pulse and is thus better suited for use as an input to a diode switch.

Fig. 46 shows a shaping amplifier circuit. A sinusoidal head signal of approximately 20 volts peak to peak is applied to terminal **317**. The diode in the grid circuit prevents the amplifier from responding to a negative swing of the signal. On positive excursions of the head signal the grid voltage swings positive and would increase by the full 20 volts of the applied signal except that it is clipped to a 10 volt value by the flow of grid current. The output of the first stage is capacitor coupled to the second stage the output of which is taken from terminal **318**.

Thus whenever the head signal swings in a positive direction the shaping amplifier responds with a reasonably square pulse which is in phase with the positive signal swing. Fig. 46*a* shows the block symbol for the shaping amplifier of Fig. 46.

Voltage amplifiers

In this machine class A voltage amplifier circuits are used to amplify the sinusoidal head signals before they are used. Fig. 47 shows a typical voltage amplifier circuit used in connection with the buffer storage read circuits. The circuit is a straightforward, two stage, class A amplifier with direct coupling to the first stage and capacitor coupling to the second. The output is an amplified replica of the input signal. Fig. 47*a* shows the block symbol for the amplifier of Fig. 47.

Fig. 48 shows another type of voltage amplifier similar to that of Fig. 47 with a clamped input. Fig. 48*a* shows the block symbol for the amplifier of Fig. 48.

Latch circuits

In most electronic computers the timing, counting, computing, conversion and input-output circuits utilize some type of flip-flop device having two stable states. In the past this device has often been a multivibrator type of unit commonly called a trigger. In the present machine a device called a latch circuit is used. This circuit has two primary advantages over the trigger. First, it is not a balanced circuit and as a result tube balance and matched components are not of prime importance. The latch will continue to function properly under conditions which would cause erratic operation of a trigger. Second, a latch is made up of a double inverter and cathode follower, usually with a diode switch or mix input to the latch circuits. This permits standardizing on fewer types of units. When triggers are used, several different types are required. A latch is put together from the previously described inverter, cathode follower and diode units.

Fig. 49 shows a portion of a latch circuit. A pulse on terminal **324** will raise the grid of the first triode inverter above cut-off causing the triode to conduct. This in turn lowers the grid potential of the second inverter below its cut-off point. The plate voltage rise of the second inverter in the non-conducting state causes the grid of the cathode follower to rise to a conducting potential. As a result the cathode follower output rises to approximately a plus 10 volt potential. This output is available for use, to act on some further circuit, and may be fed back to the grid of the first inverter through a mixing diode to hold it at a conducting potential to form the latch circuit. This positive feed back is sufficient to hold the circuit in the "latched" condition. This "on" condition will sustain itself until some external action is applied to the circuit to render the output cathode follower non-conductive. This latch can be reset by means of a reset pulse applied to terminal **327**. A pulse applied to terminal **327** raises the grid of the second inverter to conducting potential. With the second inverter conducting the latch cathode follower grid potential is lowered and its output returns to a negative value. The positive feed back is removed and the first inverter becomes non-conductive. This holds the grid of the second inverter positive and the latch has been restored. The latch may also be restored by clamping the plate of the second inverter through a diode with the same resultant effect. Fig. 49*a* shows the block symbol for the latch of Fig. 49. When terminal **327** is not used as a reset terminal it is returned to −50 volts and is not designated on the latch symbol.

Fig. 50 is a portion of a latch similar to that of Fig. 49 except that a second cathode follower has been placed in parallel with the first cathode follower to give the output more power. Fig. 50*a* shows the block symbol for the portion of the latch of Fig. 50.

Fig. 51 shows a portion of a double latch circuit. This circuit is frequently used where the extra safety and checking action of a positive "off" signal is desirable. This circuit is made up of two single inverters and two cathode followers with divider grid inputs. In order to complete the latch circuit a first mix circuit is connected between terminals **334** and **335**, and terminal **337** is connected through a second mix to terminal **336**. Input variations can be obtained by using switch-mix diode circuitry instead of simple mixes.

Initial circuit conditions, assuming the latch to be "off" are as follows: (1) Terminal **338** at low potential. (2) Terminal **337** at no-signal level (−35). (3) Terminal **336** below cut-off potential through a mix diode. (4) Terminal **339** at high potential. (5) Terminal **334** at signal level (+10). (6) Terminal **335** at conducting potential through a mix diode.

When an "on" input signal is applied to terminal **336** through a mix diode the circuit is transferred to an opposite set of conditions. Terminal **338** rises and the following cathode follower conducts raising terminal **337** which holds terminal **336** up through a mix diode. The latch has been turned on and remains in this condition until an "off" input signal restores it to the initial state. Fig. 51*a* shows the block symbol for the portion of the latch shown in Fig. 51.

Fig. 52 shows a portion of a double latch similar to that of Fig. 51 except parallel cathode followers are provided. Fig. 52*a* shows the block symbol for the portion of the latch shown in Fig. 52.

TIMING PULSES AND GATES

Information is recorded in the buffer storages and general storage by switching information gates with timing pulses to produce a specifically timed information pulse for energizing the writing heads.

Information is read from magnetic storage by switching a shaped, read head signal with a timing pulse to produce a specifically timed information pulse which can be used to turn on a latch whose output will be an information gate similar in duration and timing to the one originally used in recording the spot.

A capacitor storage unit (distributor), whose output pulses are in the form of information gates, is frequently the source of information thus recorded in general storage. Also, capacitor storage units (distributor or program step) whose input circuits will respond to information gates, are frequently the destination of such information read from general storage.

Timing pulses occur at about the mid-point of the information gate interval when writing and at the middle of the shaper output interval when reading.

Thus timing pulses determine the position of a spot when it is placed on the drum and also insure that information read from the drum is available in the form of a properly timed information gate. In a similar way, other necessary drum timing divisions are established by timing pulses such that the exact position of the drum is known by the computing and control circuits at all times.

Fig. 4 shows the principal drum timing divisions. All drum times are with relation to "Home" position which is the beginning of sector 0. Other drum divisions are sectors (5 per drum revolution), words (50 per drum revolution), digits (600 per revolution), and A, B, C and D pulses, one of each per digit or 600 per drum revolution.

The basic timing interval is the 8 microsecond digit interval of which there are 600 around the circumference of the drum. Each digit interval is divided into four equal pulse intervals, A, B, C and D. The beginning of a digit timing interval is marked by the leading edge of its A pulse. The B, C and D pulses of a digit follow at 2 microsecond intervals. Twelve digit intervals, each with its A, B, C and D pulses are included in a word interval. The twelve digits of each word are successively DX, D0, D1, D2, D3, D4, D5, D6, D7, D8, D9, D10. DX is used as a switching interval between successive words. D0 is the sign storage time and D1–D10 represent the serial time-space intervals for the storage of the first through tenth position digits of a 10 digit number. Ten word intervals are included in each sector, and five sectors make up the entire cycle of drum timing.

Just as the beginning of a digit interval is marked by its A pulse, pulses must be available to mark the beginning of each word, sector and home interval. These basic timing pulses are obtained by reading permanently recorded spots on the drum's timing tracks.

There are 6 timing tracks for the 6 basic, recorded timing signals.

They are: (1) Home pulse, (2) Sector pulses, (3) Word pulses, (4) B pulses, (5) D pulses, (6) RSP—Read Sample Pulses.

The other necessary timing pulses and gates are obtained from multivibrator or latch circuits actuated by these basic pulses and/or other "manufactured" pulses.

The following list defines terms used to describe timing pulses and gates and describes the pulses and gates used:

Bit.—Smallest unit of information. A bit usually represents any one element of the seven element bi-quinary code or the five element general storage code.

Digit (*D*).—Elements of bi-quinary or five element code representing one decimal number such as six (6). Bits representing a digit are present simultaneously within a time interval of eight microseconds. A valid digit code designation contains only two positively identified code elements. One of twelve intervals of a word.

Home (*H*).—A recorded spot on the drum which is a reference point for all timing signals.

Microsecond.—One millionth of a second.

Sector (*S*).—A designated area representing one fifth the circumference of the drum.

A sector contains ten words per band and represents a time interval of .96 millisecond.

Word (*W*).—A combination of twelve digit intervals representing the smallest *addressable* storage unit of information.

A word contains ten digits (D1–D10) representing numerical information, plus one digit (D0) indicating sign, plus one digit (DX) allowed for a switching interval between words.

Pulse (*P*).—Basic timing signal usually about two microseconds' duration.

Most pulses are obtained from permanently recorded signals on the drum which are amplified and shaped. Some pulses are generated from other pulses.

Gate.—Any signal not labeled pulse is assumed to be a gate and has no basic abbreviation.

Any created timing signal usually over two microseconds in duration which represents an exact timing signal between two pulses.

Gates are usually created by switching certain timing signals or by timed control of latch circuitry.

"A" Pulse (*AP*).—A positive pulse, two microseconds' duration, occurring during the first quarter of a digit interval. "A" pulses are produced from a "one shot" multivibrator which is controlled by the leading edge of a delayed Early "A" pulse and the leading edge of a negative "B" pulse.

"B" Pulse (*BP*).—A positive pulse, two microseconds' duration, occurring during the second quarter of a digit interval. "B" pulses are obtained from permanently recorded signals on the magnetic drum.

"C" Pulse (*CP*).—A positive pulse, two microseconds in duration, occurring during the third quarter of a digit interval. "C" pulses are produced from a "one shot" multivibrator controlled by the trailing edge of a negative "B" pulse and the leading edge of a negative "D" pulse.

"D" Pulse (*DP*).—A positive pulse, two microseconds' duration, occurring during the fourth quarter of a digit interval. "D" pulses are obtained from permanently recorded signals on the magnetic drum.

Pulses that occur just prior (less than two microseconds) to a designated pulse time are signified with a prefix "E" meaning "early." Early pulses are often used to compensate for time lost between stages of electronic circuitry which require an output at a certain designated pulse time.

Early "A" Pulse (*EAP*).—A positive pulse, two microseconds' duration, designed to occur approximately ½ microsecond prior to an "A" pulse. Early "A" pulses are obtained by delaying "D" pulses. Six hundred pulses per drum revolution, eight microseconds apart.

Early Word Pulse (*EWP*).—A positive pulse, two microseconds' duration, which occurs at Early "A" pulse time every twelfth digit. Early Word Pulses are obtained from permanently recorded signals on the magnetic drum and are associated with Digit X, the first digit time of every word. Fifty pulses per drum revolution, 96 microseconds apart.

Early Sector Pulse (*ESP*).—A positive pulse, two microseconds' duration, which occurs at Early "A" pulse time every one hundred twentieth digit (or every tenth word). Early Sector Pulses are obtained from permanently recorded signals on the magnetic drum and are associated with Digit X, Word 0, the first digit of the first word of every sector. Five pulses per drum revolution, 960 microseconds apart.

Early Home Pulse (*EHP*).—A positive pulse, two microseconds' duration, which occurs at Early "A" pulse time once every drum revolution. Early Home Pulse is obtained from a permanently recorded signal on the magnetic drum. Early Home Pulse is associated with Digit X, Word 0, Sector 0, the first digit of the first word of the first sector. One pulse per drum revolution, 4.8 milliseconds apart.

Home Pulse (*HP*), *Word Pulse* (*WP*).—These pulses, two microseconds in duration, are obtained by delaying their respective "early" pulse approximately ½ microsecond so that these pulses occur at "A" pulse time.

Word Pulse (*Even Word*) (*WPL* (*EvW*)).—A positive word pulse, two microseconds' duration, which occurs for only every even word (192 microseconds apart). Words 0, 2, 4, 6, 8 are considered even words and are associated with lower accumulator read-out time.

Word Pulse (*Odd Word*) (*WPu* (*OdW*)).—A positive word pulse, two microseconds' duration, which occurs for only every odd word (192 microseconds apart). Words 1, 3, 5, 7, 9 are considered odd words and are associated with upper accumulator read-out time.

Negative timing pulses are signified with a prefix "N." Negative pulses are obtained by inverting normal pulses. A negative pulse therefore has a positive voltage level except during the designated pulse time.

NAP.—Negative "A" Pulse.

NBP.—Negative "B" Pulse.

NCP.—Negative "C" Pulse.

NDP.—Negative "D" Pulse.

NEAP.—Negative Early "A" Pulse.

NEHP.—Negative Early Home Pulse.

NHP.—Negative Home Pulse.

NESP.—Negative Early Sector Pulse.

NEWP.—Negative Early Word Pulse.

NWP.—Negative Word Pulse.

NWPL (*EvW*).—Negative Word Pulses occurring only for every even word.

NWPU (*OdW*).—Negative Word Pulses occurring only for every odd word.

Negative Latch Reset Pulse (*N. Latch Res. P*).—A negative pulse, approximately one microsecond duration, which occurs at the beginning of "*a*" time. This pulse is obtained from a "one shot" multivibrator triggered by an "A" pulse. This pulse is used to control the reset of some of the latches used in Magnetic Storage.

Read Sample Pulse (*RSP*).—A positive pulse, about one microsecond duration, which occurs near "*b*" time. Read Sample Pulses are used to sample information gates from the magnetic storage read-out circuitry. Read Sample Pulses are obtained from permanently recorded signals on the mangetic drum. Six hundred pulses per drum revolution, eight microseconds apart.

Write Sample Pulse (*WSP*).—A positive pulse, two microseconds in duration, which is similar to a "*c*" pulse. Write Sample Pulses are obtained from a "one shot" multivibrator controlled by the trailing edge of a negative "B" pulse and the leading edge of a negative "D" pulse. These pulses are used for sampling information gates which feed magnetic storage write amplifying circuitry. Six hundred pulses per drum revolution, eight microseconds apart.

Digit Gates are initiated at "*a*" time but are considered on only from "*b*" time of the designated digit to "*a*" time of the following digit.

DX—Switching Digit Gate.—One digit time allowed for a switching interval between words.

D0—Sign Digit Gate.—One digit time allowed for sign indication.

D1, D2, D3, D4, D5, D6, D7, D8, D9, D10—Digit Gates 1–10.—Digits designated for numerical information. Digit 1 represents units position, digit 10 high order position.

D1–D5.—A gate providing Digit Gates 1–4 only.

D5–DX.—A gate providing Digit Gates 5–10 only.

D5–D10.—A gate providing Digit Gates 5–9 only.

D1–DX.—A gate providing all Digit Gates except D0 and DX.

D5–D9.—A gate providing Digit Gates 5–8 only.

D10, D1–D5.—A gate providing Digit Gates 10 and 1–4 only.

D10U (OdW).—Digit Ten Gate on only for every odd word (Upper accumulator read-out time).

(Od) D.—Digit Gates occurring only for every odd digit (X, 1, 3, 5, 7, 9).

(Ev) D.—Digit Gates occurring only for every even digit (0, 2, 4, 6, 8, 10).

DXUc (OdW)–D1Uc (OdW).—A gate on from "c" time of Switching Digit X to "c" time of Digit One occurring only for every odd word. Odd Word Gates are associated with upper accumulator read-out time.

D10Lc (EvW)–D0Uc (OdW).—A gate on from "c" time of Digit Ten of an even word to "c" time of Sign Digit Zero of the following odd word.

N(D10Lc–D0Uc).—A gate on from "c" time of Digit Zero of an odd word to "c" time of Digit Ten of an even word.

The following gates have a prefix "E" which indicate early digit gates which are initiated at "d" time of the previous digit but are considered on only from "a" time to "d" time of the designated digit:

EDX.—Switching Digit Gate.

NEDX.—A gate positive for all digits except EDX.

EDXL (EvW).—Switching Digit Gate occurring only for every even word (Lower accumulator read-out time).

EDXU (OdW).—Switching Digit Gate occurring only for every odd word (Upper accumulator read-out time).

NEDXL (EvW).—A gate positive at all times except EDX of even words (Lower accumulator read-out time).

NEDXU (OdW).—A gate positive at all times except EDX of odd words (Upper accumulator read-out time).

ED0.—Sign Digit Gate.

NED0.—A gate positive for all digits except ED0.

ED0L (EvW).—Sign Digit Gate occurring only for every even word (Lower accumulator read-out time).

ED0U (OdW).—Sign Digit Gate occurring only for every odd word (Upper accumulator read-out time).

ED1.—Digit One Gate.

NED1.—A gate positive for all digit times except ED1.

ED1L (EvW).—Digit One Gate occurring only for every even word (Lower accumulator read-out time).

ED1U (OdW).—Digit One Gate occurring only for every odd word (Upper accumulator read-out time).

ED2.—Digit Two Gate.

NED2.—A gate positive for all digits except ED2.

ED2L (EvW).—Digit Two Gate occurring only for every even word (Lower accumulator read-out time).

Sector Gates are considered on from "b" time of Digit X, Word 0 to "a" time of the following Digit X, Word 0 (ten words).

S0, S1, S2, S3, S4—Sector Gates 0–4.—Positive signals from a ring lasting approximately 960 microseconds between two sector pulses, distinguishing one of five drum sectors.

Word Gates are considered on from "b" time of Digit X to "a" time of the following Digit X (twelve digits).

W0, W1, W2, W3, W4, W5, W6, W7, W8, W9—Word Gates 0–9.—Positive signals lasting 96 microseconds between two word pulses establishing position of word in a sector.

WL (EvW).—Word Gates occurring only for every even word (0, 2, 4, 6, 8), Lower accumulator read-out time.

WU (OdW).—Word Gates occurring only for every odd word (1, 3, 5, 7, 9), Upper accumulator read-out time.

a–c.—Timing Gate on from "a" to "c" time within each digit (equivalent of $AP+BP$).

c–a.—Timing Gate on from "c" to "a" time within each digit (equivalent of $CP+DP$).

Figs. 53*a* through 58*c* show the details of the development of all timing pulses and gates.

MAGNETIC DRUM STORAGE

In the present machine a revolving magnetic drum is used for storing information in the form of small magnetized spots on the drum's surface. This system of storage offers a large storage capacity in small space together with reasonably fast access.

It is well known that a current through a coil around a bar of magnetic material will induce the bar to become a magnet having a north and south pole at either end. Which end becomes the north pole is dependent upon the direction of the current in the coil. This magnet has associated with it a magnetic field, which field exerts a magnetizing force on materials placed in it.

If a sheet of suitable magnetic material is placed near the electromagnet, some of the field of the magnet will pass through the sheet. In fact there will be a greater concentration of field in the sheet than in the surrounding air because the magnetic material of the sheet presents less reluctance than air. During the time that current is flowing in the coil, that part of the sheet through which the field passes will itself become a magnet. If the proper material is used, some of this induced magnetism will remain in the sheet after current ceases to flow. The fact that current was made to flow in the coil has been recorded in the sheet and this fact could be detected by passing a compass needle over the sheet. The polarity of the recorded spot would indicate the direction of flow of the current through the coil. This is the principle of magnetic storage.

In the machine the electromagnet has become a recording head and the sheet of magnetic material has been wrapped around a drum. The head is placed so that the air gap is adjacent to the drum surface. If the drum is made to revolve, bits of information in the form of current pulses can be recorded as small magnets or magnetic spots on the surface of the drum. With this arrangement the recorded spots can be packed very close together. The spots are recorded serially in tracks, imaginary strips of surface the width of a head core which extend around the drum. The area for the recording of a spot is often called a cell.

When writing there is sufficient fringing of the field across the air gap between the poles to link with the surface of the drum and induce a magnetic spot.

With the drum revolving under the head, a ready means of getting the information back into a voltage pulse form is provided. The reading out of information is dependent on the fact that a change in magnetic field intensity in the air gap will induce a voltage across the winding. Since the spot previously written on the drum surface is itself a small magnet, there is a field associated with it. As this field passes under the head some of it fills the air gap and induces the core to become a magnet with the result that a voltage proportioned to the rate of change of flux in the core is induced across the winding. The voltage induced across the winding will be sinusoidal in form reaching maximum and minimum values when the north and south poles pass under the air gap, i.e. when the flux in the core is changing most rapidly, and falling to zero between the poles when the flux is constant. The shape of the output signal is established by the polarity of the magnetic spot, which in turn was governed by the direction of the writing current through the winding when the spot was placed on the drum. The term "writing" is taken to mean the action of placing a spot of either polarity on the drum. "Recording" and "erasing" are the two forms of writing and are used to describe the two possible polarities. A recorded spot reads back as a "down-up-down" sinusoidal signal while an erased spot reads back as an "up-down-up" signal. In the parallel bit storage systems used in this machine (bi-quinary or two-of-five) the recorded spot indicates a binary "1" while the erased spot indicates a binary "0." In the bi-quinary system the simultaneous reading of two recorded spots and five erased spots within a digit position interval determines the decimal value of a digit. In the two-of-five system the simultaneous reading of two recorded spots and three erased spots within a digit interval determines the digit's decimal value. The magnetic storage read circuits are arranged to detect the difference in the two types of signals.

The organization of the drum used in this machine is illustrated in Fig. 5. The entire drum and drive motor form a complete assembly. Cables leading to the heads terminate in sockets into which the individual heads are plugged.

The head used for reading and writing may be of the type shown and described in the copending application of Edgar Alan Brown, Serial No. 409,101, filed February 9, 1954, now Patent No. 2,915,593, and assigned to the present assignee. The assembly consists of a pivoted, laminated core with a hollow plastic bobbin around one side, on which are two 80 turn windings, one of which is center tapped. A compression spring holds the core in position with its stud against the adjusting screw. The head is held to two mounting bars by the mounting screws. As the mounting screws are tightened the tension locking clips flatten out and hold the position. This is necessary since the mounting screws are also used for adjusting purposes. The position of the head along its track can be varied by more than a full digit interval by use of these screws.

Head adjustment is accomplished by turning in on the mounting screws until maximum read signal amplitude from a previously written spot is observed on a scope. This occurs as the head contacts the drum. Further turning of the screws only results in forcing the core to pivot against the force of the compression spring with no increase and eventual decrease of signal amplitude. The screws are then backed off to the point of maximum amplitude. The adjusting screw is then used to back the core out of contact with the drum and is adjusted for a signal amplitude of approximately 75%–80% of the maximum. This results in a head to drum clearance of the order of .0005". Thus the head is adjusted for proper signal amplitude not for drum clearance. Any resulting clearance is satisfactory since even an occasional contact condition is not serious with this type of head.

The principles of reading and writing will be explained with reference to general storage. Fig. 59*a* through 59*o* show how the heads are connected for general storage. Here the two 80 turn coils are connected and a center tap connection is made at their junction. One coil is then used to record and the other to erase. The same coil which is used to record is also used for reading.

When placing information in general storage the object is to place the spot on the drum in such a position that when it is later read out it will cause the read circuits to produce an "on time" B–A information gate, i.e., if the spot is to represent D1, it should, when read out, cause a B–A information gate to be developed during the D1 timing interval. It should be remembered that all information is transmitted throughout the machine in the form of B–A gates. Reference to Fig 11 will show that information can be sent to general storage from either the distributor or from RBS. Thus the output from these units must be in the form of B–A information gates, which, when allowed to reach the general storage input, will actuate the write circuits.

Because time is required to write and to read, the general storage write circuits must be actuated in advance of the drum time of the digit to be stored, if the digit is later to be available from the general storage read circuits as on time B–A gates. In this machine the magnetic storage circuits have been designed so that one whole digit time is used in the process of writing and reading. Thus the B–A gates from either distributor or RBS which actuate the general storage write circuits must be one digit time early with respect to the drum time of the digit which they represent. The spots will then be placed on the drums in such position as to cause the read circuits to put out on time B–A gates when they are read out. In other words general storage receives information in the form of one digit early B–A gates and puts out information as on time B–A gates. It will be remembered that the distributor, a capacitor storage unit, has both a one digit early output and an on time output. The one digit early output is used when information is sent from the distributor to general storage. It will be seen also that information is written in RBS at such a time that it will be available from the buffer storage read circuits one digit early for proper entry into general storage.

The operation of the read-write matrix will be explained with reference to Figs. 59*a* through 59*o*. The outputs from the address register will have activated the selection circuits so that the potential of the center tap of a pair of coils, **401**, Fig. 59*b* for example, will have been raised from −70 to +100 volts due to conduction of power unit **402** (PW5), thus readying the selected head for use. The output from the address register, B0H, O0H and B0T, are switched at switch **403** and fed to inverter **404** (IN8), from inverter **404** through power unit **405** (PW1), through cathode follower **406** (CF19) and through cathode follower **407** (CF12) to power unit **402**. Any other head would be selected in a similar manner. A one digit early, B–A gate present on one of the bit input terminals ("0" Bit) will activate the record circuit. If the value of the digit is such that no information gate is present this "lack of information" signal level is inverted at inverter **408** (IN3) and the erase circuit will be activated instead of the record circuit.

The record circuit is activated as follows: At "*c*" time of the digit interval a write sample pulse is switched at switch **409** with the output of the Q0TH position of the address register and with a general storage write gate. The output of switch **409** is fed through a pair of parallel connected power units, **411** and **412**, (PW5's) to switch **413** where it is switched with a B–A gate on the "0" Bit line. The output of switch **413** is fed through power unit **414** (PW2) and capacitively coupled to power unit **415** (PW2). A sharp, short voltage pulse is produced at the output of power unit **415** by the trailing edge of the WSP. This pulse is fed to power unit **416** (PW7) to cause power unit **416** to conduct heavily for the short time of the pulse thus energizing the record coil through diode unit **417** (DD1) to place a spot on the drum. The complete circuit to energize the head is from ground, cathode of power unit **416**, anode of power unit **416**, diode **417**, the coil, cathode of power unit **402**, anode of power unit **402** and to the 250 volt write circuit supply.

The erase circuit is activated as follows: The outputs of the above-mentioned power units **411** and **412** are switched at switch **418** with a signal produced by a "no B–A gate" passing through inverter **408** and cathode follower **419** (CF3). The output of switch **418** is fed through power units **421**, **422**, **423** and diode **424**, in the same manner as the output of switch **413** activated the record circuit, to energize the erase coil.

The power units **416** and **423** are caused to conduct for about 1½ microseconds by a pulse on their grids and write current flows through one of the coils during this time.

When it is desired to read information from general storage, it is necessary that the read circuits respond to recorded head signals by producing B–A gates during the digit timing interval following the one which caused the spot to be recorded. It is also necessary that the read circuits ignore signals due to erased spots. Thus the read circuits of general storage will put out two simultaneous B–A gates, from two of the five parallel read circuits, during each digit interval whenever the address register outputs activate the selection circuits. This is in accord with the two of five system used for general storage.

For reading a recorded spot the address register outputs will again select the head by causing power unit **402** to conduct, thus raising the potential of the center tap as in writing. In this case no B–A information gates are allowed to reach the write circuit input terminals and power units **416** and **423** remain cut off. Thus when the selection is made and power unit **402** conducts, parallel circuits are completed from ground, through the resistors tied to the anodes of power units **416** and **423**, vacuum diodes, read and erase windings, power unit **402** to +250 volts. Any induced voltages which tend to build up across the windings cancel out and the net effect of the selection is to raise the D.C. level of the lines to which the anodes of power units **416** and **423** (Fig. 59*b*) are tied from 0 to approximately +100. This raises the D.C. level of the grid of cathode follower **425** (CF11, Fig. 59*e*) associated with the read line to +100. Since the cathode is returned to +90, cathode follower **425** is conditioned for conduction. Cathode follower **425**, whose anode is normally blocked by action of power unit **426** (PW4) used as a blocking inverter, will be "unblocked" by the Q0TH selection from the address register. Thus, cathode follower **425** will conduct and drive voltage amplifier **427** (VA2). After the selection transient has died out, the induced A.C. head signal present across the read winding and superimposed on the 100 volt selection level, continues to drive the "unblocked" cathode follower **425** and thus becomes the input signal to voltage amplifier **427**. The output of voltage amplifier **427** is then an amplified replica of this A.C. head signal. The D.C. component of the voltage amplifier **427** output is removed by the input capacitor of the shaping amplifier **428** (SA1). The output from a recorded signal of the shaping amplifier **428** which responds only to positive changes in the input signal, overlaps the time of the "read sample pulse" (RSP) while that from an erased signal does not overlap or coincide. Thus there is an RSP output from the diode switch **429** on recorded signals and no output on erased signals. This pulse turns the output latch **836** on at "*b*" time. The latch is reset every "*a*" time by a "negative latch reset pulse." The output of the latch is the necessary "on time" B–A information gate. These B–A gates pass through the 5 to 7 conversion circuits where they are converted to the 2 of 7 bi-quinary system of digit value representation for transmission to distributor, program register or adder.

This action takes place continuously as long as a valid general storage address is in the address register. Information gates from a whole selected band of general storage are therefore available from the output latches. The desired series of B–A gates is selected from this "train" of information by the read-in switching circuitry of the unit which is to receive the information as directed by the program control circuits.

By examining Figs. 59*a* through 59*o* it may be seen that a combination of circuitry like that just described accomplishes all writing into and reading from general storage.

Buffer storage circuitry used to record, erase and read differs primarily in the way the heads are connected. Since no head selection is necessary, less circuitry is needed and the head circuits are completed to ground.

Information gates from the seven parallel buffer storage read circuits are continuously available from the output latches. On a read operation a 10 word control gate coinciding in time with RBS, allows these B–A information gates to get through switches to operate the general storage write circuits thus transferring the 10 RBS words to the selected general storage band. As soon as this transfer is complete a similarly timed control gate (RBS erase gate) is applied to gate through a WSP for each digit of each of the RBS words to operate the erase circuits and erase RBS to blanks. The card information gates from the scanning matrix are then gated through to operate the record windings of RBS. This information will be transferred to a selected general storage band on the next read operation. The above control gates are developed by the buffer storage transfer and erase controls which are activated by read or punch operation codes.

On a punch operation a PBS erase gate which coincides in time with PBS, gates through a WSP for each digit of each PBS word to operate the erase circuits and erase PBS to blanks. As soon as this erasure is completed, on the next drum revolution, a similarly timed control gate (PBS read-in) is applied to diode switching to allow the 5 to 7 converted B–A output gates of this portion of a selected general storage band to operate the buffer storage record circuits thus transferring the information to be punched to PBS. The information in PBS is scanned out to punch by repeated usage of a PBS readout gate on several successive drum revolutions.

CAPACITOR STORAGE UNITS

In this machine, three major functional units are built up from the basic capacitor storage "cells": the distributor, the program register or program step storage unit and the accumulator storage unit. Each of these units are much the same in principle, although slight differences exist in methods of read-in and regeneration control. The principle of operation of this capacitor storage will be presently explained by reference to Figs. 18, 18*a* and 61*a* through 61*l*.

With reference to Fig. 61*j*, assume the initial circuit conditions are as follows:

Condenser **216** (Fig. 18) of condenser storage unit **441** (Fig. 61*j*) (CS1) is discharged. The ROD8 line is at −35 volts. The RID8 line is at +150 volts. Cathode follower **442** (CF16, Fig. 61*i*) is conducting at no-signal level, thus its output, the Q3 Even line is at −35 and, since there is no charge on the condenser, the junction between diodes **214** and **215** (Fig. 18) is also at −35.

The RID8 and ROD8 lines are pulsed periodically by the outputs of inverters **443** (IN6) and cathode followers **444** (CF15, Fig. 62*j*). Timing pulses from timing rings drive these inverters and cathode followers.

The pulsing of the ROD8 line provides a positive readout gate and is followed by the pulsing of the RID8 which produces a negative read-in gate. Cathode follower **442** is pulsed under control of the capacitor storage read-in circuitry and is active only when it is desired to "read in" to the capacitor.

The capacitor is considered to be storing a binary "1" when it is discharged and a binary "0" when charged. Thus reading into a capacitor storage cell requires that the capacitor be left discharged upon termination of the read-in gate.

When the read-out (R.O.) gate is applied, the ROD8 line rises to +10 causing the diode **214** to conduct. This connects the junction between diodes **214** and **215** to the ROD8 line and pulls the junction between diodes **214** and **215** up to +10 volts. This upward shift raises the Q3 Even line momentarily until the capacitor can charge and allow the Q3 Even line to return to −35 volts along the capacitor charge curve. This sharp pulse at the Q3 Even line is obtained whenever the R.O. gate tests a discharged capacitor. Had the capacitor been charged, no output pulse would appear. Termination of the R.O.

gate returns the ROD8 line at −35 volts and effectively disconnects the junction of diodes **214** and **215** and the ROD8 line leaving the ROD8 line at +10 volts. When the read-in (R.I.) gate is applied the RID8 line is lowered to +10 volts, but with the junction between diodes **214** and **215** at +10 there is no potential difference across diode **215** and no conduction occurs, thus no path is available to discharge the condenser. Under these conditions the condenser remains charged upon termination of the R.I. gate. A "1" has been read out and a "0" has been stored.

Cycle 2 also assumes that the condenser **216** is discharged when tested by the R.O. gate and an output pulse is obtained. Here, however, it is desired to store a "1" and the capacitor must be left discharged. To accomplish this the read-in control circuits cause the pedestal cathode follower **442** to conduct simultaneously with the R.I. gate. This causes the Q3 Even line to be raised by an amount equal to the pedestal pulse from the output of cathode follower **442**. This positive shift on the Q3 Even line is reflected momentarily through the condenser to the junction between diodes **214** and **215**. Thus when the RID8 line is lowered to +10 volts by the R.I. gate the junction between diodes **214** and **215**, which was at +10 volts is simultaneously raised to +55 volts by the pedestal pulse. Under these conditions diode **215** conducts and the condenser discharges through diode **215** and the cathode resistor of cathode follower **442** while sitting on top of the pedestal. At the termination of the pedestal pulse, the Q3 Even line and the junction between diodes **214** and **215** are lowered to −35 volts. The condenser has been discharged during the R.I. interval and a "1" is stored.

Because a capacitor will not maintain a charge indefinitely, capacitor storage is a "non-sustaining" type of storage requiring periodic regeneration. In this machine each capacitor has a R.O. gate and a R.I. gate applied to it once every word interval (96 microseconds) in the case of the distributor and program step storages and once every two word intervals (192 microseconds) in the case of the accumulator storage. When the capacitor storage unit is required to read out and retain its information for several word intervals each cell must be regenerated once each word. This is accomplished by causing the output pulse obtained when a "1" is read out, and which is available several microseconds before R.I. time, to actuate the read-in control circuits so that a pedestal pulse is obtained simultaneously with the R.I. gate. Thus, when regenerating, any time a "1" is read out a pedestal is obtained and a "1" is read back into the same capacitor. If a "0" is read out, no output pulse is available to actuate the read-in control circuits, no pedestal is obtained and a "0" is read back in.

When it is desired to read a new word into a capacitor storage unit the path between the output pulse and the read-in control circuits is blocked, and the read-in controls and consequently the pedestal cathode followers are actuated by information pulses from the read-in source.

The principles of this capacitor storage are described more fully and claimed in the copending application of Roy L. Haug and Charles W. Allen, Serial No. 445,221, filed July 24, 1954, and assigned to the present assignee.

The following sections will describe how several capacitor storage cells are combined and controlled to form a bi-quinary, serial storage unit capable of storing the digits of one or two words.

The distributor

Figs. 61*a* through 61*l* show the circuits comprising the distributor and its regeneration, read-in and reset controls. The unit consists of a capacitor storage matrix, even and odd digit amplifiers and blocking inverters, early latches, on time latches and pedestal cathode followers together with entry switching and read-in and regeneration controls.

The matrix is made up of eleven columns of seven capacitor storage cells. There is one column for each digit storage position and one cell in each column for each bi-quinary bit. The distributor is required to store eleven digits, D0 being the sign storage position.

The seven cells of each column have their R.O. terminals commoned and driven by a "1 digit early" gate and their R.I. terminals commoned and driven by an "on time" digit gate.

The capacitor terminal of each of the even digit cells in a bi-quinary row connects to a common pedestal line controlled by a pedestal cathode follower. The same is true of the odd digit cells.

For purposes of explanation, assume that the distributor has been reset and is storing plus zeros. In this case the D0 position contains a 9 (only the B5 and Q4 capacitors are discharged). All other digit positions contain zeros (B0 and Q0 capacitors discharged).

At DX time a B–A, R.O. gate is applied to the D0 position. The B5 and Q4 capacitors charge, developing capacitor charge "spikes" across the resistor in the corresponding pedestal cathode followers **445** (Fig. 61*e*) and **446** (Fig. 61*i*). These "spikes" drive their corresponding even digit amplifiers, double inverters **447** and **448** (IN5's), Figs. 61*h* and 61*l*, which are free to operate since their blocking inverters **449** and **451** (IN3's) normally only conduct during even digit time intervals. (DX is an odd time interval.) These amplifier outputs through cathode followers **452** and **453** (CF4's) turn on the corresponding B5 and Q4 early latches **454** and **455**, respectively, Fig. 61*c*. These latches stay on for the remainder of the DX interval and are reset by an inverted AP through the associated reset inverters **456** and **457** (IN3's), Fig. 61*d*. The early latches while on, furnish a distributor early output. When the early latches go off at "*a*" time, the corresponding on time latches **458** and **459**, Fig. 61*b*, are pulled on by a pulse from the left anode of the early latches through capacitor coupled inverters **461** and **462** (IN9's). These on time latches are held on until the next "*a*" time through a latch back circuit through switches **463** and **464** controlled by a NAP, and while on furnish an on time output during D0 time. The on time latch outputs also switch at switches **465** and **466** (Figs. 61*e* and 61*i*) with NAP's and (in this case) an even digit gate to raise the B5 and Q4, even digit pedestal lines during D0 time. This raising of the pedestal lines coincides with the D0 R.I. gate on the D0 position and allows the B5 and Q4 capacitors to discharge on top of the pedestal as explained above. The even digit amplifier blocking inverters **449** and **451** are conducting during D0 time while the even digit lines are being pedestalled, thus preventing the amplifiers from responding to the pedestal pulse and incorrectly turning on the early latches.

At the same time the D1 position (an odd position) is being read out through the odd digit amplifiers and the above process is repeated with read-out occurring at D0 and read-in at D1 time. The B0 and Q0 latches will be active since the B0 and Q0 capacitors were discharged.

This process of reading out one digit early, reading in on time and pedestalling two of the seven lines during read-in time from the on time latches is repeated for each digit position. Each column of capacitors is tested once each word interval and thus is required to maintain its state of charge or discharge for 96 microseconds.

In this way a number stored in the unit is continuously available from the early and on time outputs and is continually regenerated.

Reading a new number into the unit merely requires that the circulating regeneration path be opened and that two of the seven pedestal lines be operated each digit time by on time pulses from the source of the information instead of pulses from the distributors' own output.

This is accomplished by the distributor read-in and regeneration controls, Fig. 61*d*. In the case of the distributor, regeneration is interrupted by causing the output of the regeneration control latch **467** to energize all of the even and odd digit amplifier blocking inverters through mixes **468** and **469** and through cathode followers **471** and **472** so that the outputs from the capacitor matrix cannot turn on the early latches. With the early latches blocked, the on time latches cannot be turned on by outputs from the distributor and regeneration is prevented. Instead, an output from either of the distributor read-in control latches **473** and **474** (R.I. Dist. from Stor. or R.I. Dist. from ACC) will open up read-in switching Fig. 61*a*, allowing the on time latches to be turned on by on time output pulses from either selected storage or the accumulator. When regeneration is interrupted in this way, by blocking the turning on of the early latches, it must be done by a gate which is one digit early with respect to the gate which opens the read-in switching.

The distributor can be read into from two sources:

(1) Selected storage which includes any general storage location, the storage entry switches (8000) and upper or lower accumulator when selected by an 8002 or 8003 address.

(2) Directly from the accumulator on time output.

The selected storage source is used to enter information into the distributor on all arithmetic operations (10's and 60's codes). The first step of any arithmetic operation is the entering of the operator or operand into the distributor from a storage location specified by the data address. The operator or operand is then available from the distributor outputs for use in accomplishing the operation.

The direct, accumulator on time, output source is used to enter information into the distributor from the accumulator on store accumulator operations (codes 20, 21, 22, 23).

On 10's and 60's operations, a signal is sent from the program control commutator to operate the distributor read-in and regeneration controls. This signal is sent at the beginning of the word time when the storage location containing the data to be entered is in position to be read out and occurs at DXB time.

Fig. 60 shows a timing chart of this action.

To enter this selected word of information into the distributor, the distributors' selected storage entry switching must be open for D0 through D10 of the selected word interval and its regeneration path must be blocked from D0 through D9 time. Figs. 60 and 61*d* show how the storage to distributor read-in signal (DXB) turns on the storage to distributor read-in latch **473** and in turn the distributor regeneration control latch **467** to provide the necessary read-in and regeneration blocking gates.

The read-in gate is actually open for most of the DX interval although not needed until D0 time. No output is available from any general storage location or the switches at DX time, and any DX output from the accumulator is a zero. A zero at DX time would only turn on the B0 and Q0 on time distributor latches which are forced on at DX time by the distributor reset circuit which includes inverters **475** and **476**, Fig. 61*c*, to provide an on time distributor output at DX to satisfy the distributor validity check circuits.

Regeneration is not blocked for DX time when the distributors' sign position is being read out because the signal from program cannot be developed in time to accomplish this. Instead a DX reset circuit holds the Q3 and Q4 early latches off for every DX interval. Thus the corresponding on time latches will not be affected by the early latches at D0 and will respond only to the Q3 or Q4 pulse from the sign position of the selected storage unit. The B5 early latch is allowed to turn on at DX and in going off will turn on the on time B5 latch. This causes no failure since the B5 on time latch would be turned on by the B5 pulse from the D0 position of the selected storage unit.

The storage to distributor readi-n latch and the regeneration control latch are turned off with a D10DP. Fig. 60 shows them dropping at D10D time. Actually they taper off, since time is required for a latch to transfer, and cannot be considered fully down until the end of D10 time. Thus the read-in gate is up through D10 as required. The regeneration blocking gate is also held through D10.

On store operations (20 codes) a gate developed by the program control commutator activates the distributor read-in and regeneration controls for the particular code.

On a store lower operation (code 20) it is necessary for the read-in switching to be open from D0L through D10L and for regeneration to be blocked from D0L through D9L. It is desirable that regeneration also be blocked for DXL although not necessary because of the reset circuit mentioned above.

Since, in the case of the 20 codes, the gate from program is available in time to do it, the regeneration latch is turned on at DXL, in time to blank regeneration for DX as well as the necessary D0 and D9. Read-in switching is not opened until D0 time thus there are no accumulator on time DX pulses available to turn on the distributor B0 and Q0 on time latches for a DX on time zero output. Examination of the distributor reset controls shows that the B0 and Q0 on time latches are forced on at every DX, in order to have an on time zero output available to satisfy the validity check circuits.

On a store upper operation (code 21) the timings of the gates are the same as for code 20, except that they must include an odd word interval rather than an even. Thus the read-in switching will be open from D0U through D10U and regeneration will be blocked from DXU through D9U.

On a store D operation (code 22) the read-in switching must be open for D5 through D8 and regeneration blocked for D4 through D7.

On a store I operation (code 23) the read-in switching must be open for D1 through D4 and regeneration blocked for D0 through D3.

Separate distributor read-in and regeneration controls are provided for the store codes. An accumulator to distributor control gate is developed by program control whose timing depends on the Op. code and is different for each code as shown in the timing chart of Fig. 60. This gate is the output of the accumulator to distributor control double latch **477** (Fig. 81*k*). For 20 and 21 codes it is turned on with a D10DP and off with a D9DP. For code 22 it is turned on by a D3DP and off with a D7DP. For code 23, on with a DXDP and off with a D3DP. Because time is required for the latch to transfer its output in each case is available to the distributor slightly before "*a*" time. Capacitor coupled cathode follower pulses from the inverter anodes of this double latch supply the start accumulator to distributor R.I. and end accumulator to distributor R.I. signals which are used to operate the distributor regeneration control latch **467** (Fig. 61*d*).

The accumulator to distributor R.I. control gate, in each case, is developed approximately one digit early by program control. This is done to eliminate the effect of any time shift in the control gate as it is transmitted from program control to the distributor. This early control gate is then delayed by the action of the accumulator to distributor R.I. delay latch **478** and the accumulator to distributor R.I. latch **474** to provide the exact read-in gate needed. This delay action is shown in the timing chart of Fig. 60.

Whenever a manual accumulator reset gate is developed by depressing either the accumulator reset key **479** (Fig. 74*b*) or the computer reset key **481** (Fig. 74*b*) on the control console, the distributor is reset to plus zeros. This is done by holding the Q1, Q2, Q3 on time latches off for all digits during manual accumulator reset; the B5, Q4 latches on and the B0, Q0 latches off for D0 during manual accumulator reset; the B0, Q0 latches on and the B5, Q4 latches off for all digits except D0 during manual accumulator reset. Fig. 61*c* shows the switching of manual accumulator reset with D0 and NED0 at switches **482** and **483** to accomplish this resetting.

The numbers at the lines leaving or entering the distributor drawings (Figs. 61*a* through 61*l*) are the Fig. numbers to which or from which the lines go or come.

The program register

The program register or program step storage shown at Figs. 62*a* through 62*l*, is a 10 digit capacitor storage unit similar to the distributor. It consists of a 10 digit capacitor storage matrix with positions for storage of digits 1 through 10 (no sign is stored), even and odd digit amplifiers and blocking inverters, early latches, on time latches, pedestal cathode followers, entry switching and read-in, regeneration and reset controls.

As in the distributor, one digit early capacitor readout spikes from two of the seven capacitors in a digit column drive the even or odd digit amplifiers causing the corresponding early latches to be turned on. These latches are normally turned off at the end of the early digit time by an inverter AP, and while on provide the program step early output. The early latches in going off cause the corresponding on time latches to be turned on through capacitor coupled inverters. The on time latches remain on until the next "*a*" time through a latch back circuit controlled by a NAP, and while on provide the program step on time output. The outputs of these latches also switch at switches **484** (Fig. 62*a*) with a program pedestal regeneration gate (normally open except when closed during a TLU operation) to drive amplifiers **485** (Figs. 62*e* and 62*i*) whose outputs switch with NAP and even or odd digit gates as the case may be, to operate the pedestal cathode followers. The output of these on time amplifiers **485** is also the input to the address register. Thus as long as the even and odd digit amplifier blocking inverters are not held in conduction and the TLU program regeneration gate is open, a circulating regeneration path is possible whereby the stored information is continuously regenerated and made available from the early and on time outputs.

Note that the switch, cathode follower, inverter driving circuits of Figs. 62*i* through 62*l* which provide the R.O. and R.I. gates also drive the distributor.

The primary function of the program register is to receive and store an instruction word from a selected storage location and make the OP–D and I portions of this word quickly available to the operation and address registers for interpretation. During TLU operations the D portion of the instruction word in the program register is altered by merging the contents of the program register with special digits, in the adder, and storing the altered word back in the program register.

Thus the program register can be read into from two sources:

(1) Selected storage on a control commutator I half cycle.

(2) The adder during table look-up.

Two separate sets of program register read-in and regeneration controls are provided for these two conditions.

On a program register read-in from selected storage (RIPS), program register read-in switching and regeneration blocking are accomplished in the same manner as for the distributor. A Read In Program Step (RIPS) latch **486** (Fig. 62*b*) is turned on by a signal from program control at the beginning of the word time when the selected storage location is in position to read out. The output of this RIPS latch opens the read-in switching to the on time latches and turns on a program regeneration control latch **487** (Fig. 62*l*) whose output blocks regeneration by holding the even and odd digit amplifier blocking inverters in conduction. Both the RIPS and program regeneration control latches are turned off with the next WP, through reset inverters **488** and **489**.

During table look-up operations regeneration cannot be blocked between the matrix and the early latches because a program register early output is needed for entry into the adder. Instead the path between the on time latch outputs and the pedestal cathode followers is opened and on time information from the adder output is substituted. This is accomplished by the TLU program regeneration and TLU program add gates and the associated switches **484** and **491** and mixes. These gates are the output of separate TLU program regeneration and TLU program add latches, part of the arithmetic control circuits.

The timing of the program register read-in and regeneration controls is shown in Fig. 63.

All digit positions of the program register are reset to zeros when the manual program reset gate is developed. This gate is developed whenever either the program reset key or the computer reset key on the control console is depressed. This resetting is accomplished by the reset inverters **492** (Fig. 62*c*) associated with the B5, Q1, Q2, Q3 and Q4 on time latches and the extra mix diodes in the input circuits of the B0 and Q0 latches. This circuitry holds the B0 and Q0 latches on and the others off during the manual accumulator reset gate duration. Regeneration is not interrupted during reset so the B0 and Q0 lines are pedestalled for each digit interval resulting in zeros being placed in each digit position.

The accumulator

The accumulator, Figs. 64*a* through 64*j*, is a capacitor storage unit similar in principle and circuitry to the distributor and the program register except that it has capacity for two words of storage. The lower accumulator has twelve storage positions, DXL, D0L, and D1L through D10L positions. The upper accumulator has 10 positions, D1U through D10U. There is a two digit time gap between the D10L and D1U positions since there are no DXU or D0U storage positions. The timing chart for the accumulator is shown at Fig. 65.

The accumulator consists of a 22 digit capacitor storage matrix, even and odd digit amplifiers and blocking inverters, early latches, on time latches, read-in and regeneration switching, pedestal cathode followers and reset controls. Stored information is continuously regenerated as long as the even and odd blocking inverters are not held in conduction and the accumulator regeneration gate is up. As with the distributor and program register, regeneration is accomplished by allowing the on time latch outputs to control the pedestal cathode followers.

All read-in to the accumulator is to the pedestal cathode followers, controlled in the same manner as the TLU program register read-in operations, i.e., accumulator regeneration is turned off, opening the path between on time latches and pedestal cathode followers, and accumulator read-in is brought up, closing the path between adder output and pedestal cathode followers. The adder output is substituted for the on time latch output and read-in from the adder takes place.

It is also possible to control the pedestal cathode followers from the accumulator early latch outputs. This is done when the accumulator regeneration gate is turned off and the right shift gate is developed and is used to accomplish a right shift operation.

The accumulator regeneration, accumulator read-in and right shift latches which develop the read-in and regeneration control gates are part of the arithmetic control circuits and thus are not shown with the accumulator.

The accumulator is read into during the execution of the following codes:

10, 11, 14, 15, 16, 17, 18, 60, 61, 64, 65, 66, 67, 68
(Add, Subtract, Multiply and Divide operations)
30, 31, 35, 36 (Shift operations)
84 (TLU)

Arithmetic controls for each of these operations will therefore operate the accumulator regeneration latch and accumulator read-in latch. Codes 30 and 31 will also use the right shift latch.

The operation of these control latches will be discussed later in connection with the description of the specific operation.

Several special circuits, not found in the other capacitor storage units are associated with the accumulator to handle conditions arising because of its design and use. These circuits will be discussed in the following paragraphs.

Because there are no DXU and D0U accumulator positions, the D1U position is read out at D10L time (one digit position early). D1U cannot be read-in until D1U time. The early latches, therefore, must not be reset during DXU and D0U time. To accomplish this the AP's which control the early latch reset inverters **493** (Fig. 64*d*) are switched with a N(D-10*c*L–D0*c*U) gate. (Negative digit 10 lower "*c*" to digit 0 upper "*c*".) This is a gate which is up for "*a*" time of all digits except DXU and D0U. Thus AP's are supplied to reset the early latches except at DXU and D0U times. On all operations requiring an accumulator entry to the adder, except the left shift codes (35 and 36), the accumulator early output is used. On these operations therefore, the D1U information contained in the early latches is presented to the adder for three successive digit time intervals, D10L, DXU and D0U. It will be seen later, when the adder circuits are described, that the adder does not accept this information for analysis until D0 time. Reset of the adder carry latches is prevented during DXU and D0U time so that their carry or no carry information set up from the D10L digit is still available at D0U time for use with the D1U information to provide the correct result.

Because of the lack of DXU and D0U positions and of the above behavior of the early latches during DXU and D0U time, there is no normal input to the on time latches at DXU and D0U times. To satisfy the validity check circuits an on time DXU and D0U output of zero is created by forcing the B0 and Q0 on time latches on with DXU and D0U gates. This is accomplished through the mix circuit inputs **494** and **495** (Fig. 64*c*) to these two latches.

On reset add or subtract operations it is necessary to disregard the number which may be in the accumulator and to merge, in the adder, the contents of the distributor with zeros substituted in place of the contents of the accumulator. This result is then stored back in the accumulator. The number in the distributor has been placed in the accumulator and the effect of an add or subtract with reset has been accomplished. This action is controlled by the reset gate which accompanies every "with reset" operation. This gate does three things:

(1) Switches with NAP's at switch **496** (Fig. 64*d*) to effectively produce a succession of B–A digit gates which through cathode follower **497** turn the early B0 and Q0 latches on and force zeros on the accumulator early output lines (for entry into the adder).

(2) Prevents the matrix output from turning on the early latches by causing all even and odd digit amplifier blocking inverters **498** (Figs. 64*e* and 64*j*) to conduct.

(3) Mixes at mix **499** (Fig. 64*d*) to become one of the conditions which operate the double inverter **501** (Fig. 64*e*) and cathode followers **502** whose purpose is to force zeros into the on time latches to satisfy the validity check circuits.

On a shift and count operation the high order position of the accumulator is checked for a zero condition. If zero, the shift count latch is turned on and the operation proceeds. If not zero, no shifting is required to bring the highest significant digit into the high order position and the shift count, which is stored in the D1L and D2L positions of the accumulator, is 00. A signal from the no shift count latch is switched with D1L and D2L gates at switch **503** and mixed at mix **499** (Fig. 64*d*) to operate the double inverter **501** and cathode followers **502** (Fig. 64*l*) for these two digit times, placing zeros in the on time latches for regeneration into the D1L and D2L positions.

On manual accumulator reset operations, a manual accumulator reset gate, which is produced by depression of either accumulator reset or computer reset keys on the control console, is mixed at mix **499** to operate the double inverter **501** and cathode followers **502**. This forces zeros into the on time latches for each digit during the full two accumulator word times. These on time latch zeros are regenerated into the matrix thus resetting the accumulator.

The D0L position of the accumulator is not used for storage of the sign. Each time a number is entered into the accumulator its sign is determined and held in the accumulator sign latches, described in a later section. The output of these latches determines in part whether the next accumulator entry to the adder will be true or complement. The output of these latches also causes either a bi-quinary 8 or 9 to be inserted into the accumulator on time output lines at 00 time for transfer to the distributor.

The D0L position is used on multiply, divide and shifting operations to hold the shift count. The DXL position is used on multiply and divide to keep track of the number of multiplicand additions or divisor subtractions.

THE ADDER

The adder is used in the accomplishment of all arithmetic operations (10's and 60's except 69), the shift operations (30's) and the table look-up operation (84). There are two 7 bit, bi-quinary input lines, and one 7 bit, bi-quinary output line associated with the adder. Bi-quinary digit information is presented to the adder simultaneously on each of its two inputs. The bi-quinary sum or difference of the digits presented, is available from the output. The information entering the adder is presented digit by digit, from any of the several serial storage sources which have data flow paths connecting their outputs to an adder input. The selection of the source of information to be presented and the time of presentation to the adder are the functions of the A and B input switching circuits. These circuits are opened and closed, thus gating information to the adder, by control gates developed by the program control and arithmetic control circuits. A delay of one digit time is incurred in the adder, i.e., the sum of two digits is available from the output one digit time after they have entered the inputs. The design of the Adder is such that information must be presented simultaneously at the two inputs to obtain an output signal.

Bi-quinary arithmetic

The adder, Figs. 68*a* through 68*i*, uses only the basic arithmetic operation of addition to accomplish all calculations. Subtraction is achieved by entering one of the two numbers as a complement. Multiplication is done by over and over addition of the multiplicand. Division requires repeated subtraction of the divisor from the dividend. Thus the serial, one digit adder need only have the ability to receive two simultaneous, bi-quinary indications of digit values; combine them with a stored "carry" or "no-carry" indication from the preceding digit analysis, and produce a bi-quinary indication of this result. To understand how this is accomplished the following review of bi-quinary addition with decimal system comparisons is included.

Addition:

1 2 3 4 5+6 7 8 9 0=8 0 2 3 5

Decimal					Bi-Quinary				
D5	D4	D3	D2	D1	D5	D4	D3	D2	D1
1 1 6	1 2 7	1 3 8	4 9	5 0	1 01 51	51 02 52	51 03 53	5 04 54	50 00
8	0	2	3	5	53	00	02	03	50

Subtraction:

8 0 2 3 5−6 7 8 9 0=1 2 3 4 5

Decimal							Bi-Quinary						
D7	D6	D5	D4	D3	D2	D1	D7	D6	D5	D4	D3	D2	D1
1 0 9	1 0 9	8 3	0 2	2 1	1 3 0	1 5 9	51 00 54	51 00 54	5 53 03	00 02	02 01	1 03 00	51 50 54
0	0	1	2	3	4	5	00	00	01	02	03	04	50

Note that on all additions involving a complement a "Carry" is inserted in the units position.

The digits to be added are presented in bi-quinary form. Each decimal digit is represented by simultaneous B–A information gates on two of the seven bi-quinary lines, one gate on one of the lines of the binary level and one on one of the lines of the quinary level. The quinary levels are added. In the quinary (base 5 level) the only values possible are 0, 1, 2, 3 and 4. Whenever a quinary sum exceeds 4, a carry into the binary level occurs, the quinary returns to 0 and again begins to advance toward 4.

Each of the two positions in the binary level represents a count of five units. That is, the two binary indications alternate to show whether the value represented by the quinary level is below five (0, 1, 2, 3 or 4) or five and above (5, 6, 7, 8, 9). Because the binary level counts in units of five, any carry from the quinary level to the binary will be a five unit carry. Similarly, because the quinary level counts units of one, any carry from a binary level to the quinary of the next digit will be a one unit carry.

Subtraction is accomplished by adding the complement of the number to be subtracted. The table below shows the bi-quinary complements of the ten decimal digit values.

Decimal Value	Bi-Quinary Form		Bi-Quinary Complement	
0	0	0	5	4
1	0	1	5	3
2	0	2	5	2
3	0	3	5	1
4	0	4	5	0
5	5	0	0	4
6	5	1	0	3
7	5	2	0	2
8	5	3	0	1
9	5	4	0	0

Notice that in the binary level B0 and B5 are complementary while in the quinary level Q0 and Q4 are complementary as are Q1 and Q3. Q2 is its own complement. The fact that the two levels are diagonally complementary makes complement entries to the adder quite simple. It is only necessary to allow the information gate from the source to enter the adder on the diagonally opposite adder input lines. The result is the entry of a 9's complement. However, as indicated in the preceding subtraction example, a carry indication is inserted into the adder so that its analysis of the unit's digit is the same as if a carry had been left over from a preceding digit. This is done on any complement entry and the effect is the same as if a 10's complement had been entered.

Adder A and B entry switching

Figs. 66*a* through 66*d* show that adder entry switching A provides six different data flow paths, namely selected storage output, accumulator on time output, units position of address register (comp.), program step early output, and accumulator early output both true and complement, to the adder A input lines. Figs. 67*a* through 67*d* show that adder entry switching B provides four different data flow paths, namely the special integer entry distributor on time output, and the distributor early output both true and complement, to the adder B input lines. The control gates for controlling these entries are developed by the arithmetic control circuits in accordance with the operation being executed. The development of these control gates will be discussed in a later section on arithmetic controls. Thus for any given arithmetic operation the control gates will select the data flow paths and will open these A and B switching circuits at the proper times and for the necessary intervals to supply information to the adder as required by the operation.

Adder circuits

The one digit adder, shown at Figs 68*a* through 68*i*, consists of diode switch and mix circuitry, carry and no-carry latches, binary and quinary output latches, reset controls, carry and no-carry insert controls and adder output latch zero insert controls.

In effect the bi-quinary addition tables are built into the adder. For a particular pair of bi-quinary inputs only one bi-quinary output, which is the sum of the inputs, is obtained.

During a digit interval when information is presented to the adder two of the seven bi-quinary input lines, at both A and B inputs, are raised. Either the carry or the no-carry latch will be on as a result of the preceding digit analysis or as a result of a carry or no-carry insert. Quinary inputs are analyzed in the quinary matrix. Similar A and B quinary inputs are mixed at mixes **505** (Figs. 68*g* and 68*j*) to give A or B Q0, A or B Q1, A or B Q2, A or B Q3, or A or B Q4 signals from the cathode followers (Figs. 68*g* and 68*j*) whose outputs are so labelled. When the A and B quinary inputs are not the same, two of these cathode followers will be energized. All combinations of two or these five cathode followers are switched at switches **506, 507, 508, 509, 511, 512, 513, 514, 515** and **516** (Figs. 68*k* and 68*l*) to operate the quinary sum cathode followers (Figs. 68*h* and 68*i*) and indicate a quinary sum of 1, 2, 3, 4, 5, 6 or 7. When the A and B quinary inputs are similar they are also switched at switches **517, 518, 519, 521** and **522**, and mixed where necessary, to operate the quinary sum cathode followers and indicate a quinary sum of 0, 2, 4, 6 or 8. The 0, 1, 2, 3 and 5, 6, 7, 8 quinary sums are separately mixed at mixes **523** and **524** (Fig. 68*i*) to give "below 5" or "5 and above" inputs to the binary matrix. The Q4 sum is switched at switches **525** and **526** with carry and no-carry signals. These switch outputs are separately mixed into the "below 5" and "5 and above" cathode followers. Thus a Q4 sum and no-carry is "below 5" while a Q4 sum and a carry is "5 and above." The quinary sum signals 0 and 5, 1 and 6, 2 and 7 and 3 and 8 are mixed at mixes **527, 528, 529** and **531** to provide with the Q4 sum the five signals for operation of the quinary output latches. These signals are all separately switched at switches **532** through **539** and **541** and **542** with carry or no-carry D pulse signals to turn on the quinary latches, e.g., 0 or 5 and no-carry turns on Q0 while 0 or 5 and carry turns on Q1, etc.

Effectively then, the three pieces of information necessary for turning on a quinary latch (quinary A input, quinary B input and carry or no-carry) are available during the adder entry digit interval and are sampled at the end of this interval with a DP. Thus during "*d*" time of the adder entry digit interval, coincidence of signals exists at one of the quinary latch input switches. This positive DP output from the switch is inverted by inverters **543** through **547** to a negative DP and applied to the grid of a capacitor coupled inverter **548** which is normally cut off. A negative output is required from this inverter to pull the latch on. Thus the latch **549, 551, 552, 553** or **554** is turned on, not during signal coincidence, but when coincidence is removed. This happens with the beginning of the next digit interval as the DP and the input and carry or no-carry signals terminate. It is the fact signal coincidence was established and then is removed from one of the latch input switches, which causes the latch to turn on.

Any adder latches which are on (binary, quinary and carry or no-carry) are reset at "*a*" time of every digit interval as long as the adder reset control latch **555** (Fig. 68*a*) is not on. This reset is done by removing coincidence from the latch back switches with NAP's. Whenever it is desired to prevent resetting these latches the adder reset control latch is turned on. Its output mixes to shunt the NAP's and hold the reset line up.

The time constant of the capacitor coupled inverter grid circuit is sufficient to supply a latch "pull on" pulse of approximately 4 microseconds' duration. As explained above, this pulse is acting to turn the latch on during the time that the NAP is resetting the latches. However, since the inverter pulse is of longer duration than the NAP it over-rides it, holding the latch on during the NAP reset time and allowing it to latch on once the latch back circuit is again opened at the termination of the NAP.

In this way, information which enters the adder during a given digit time is analyzed, combined with a carry or no-carry indication held over from the preceding digit, and transferred to the output latches at the beginning of the next digit interval. This information in the adder latches is available during the remainder of this "next digit interval," for transfer to accumulator, etc.

Notice that the information contained in the carry and no-carry latches **556** and **557** (Fig. 68*e*), regarding the carry or no-carry status of the preceding digit, is used to set up the quinary latch for the new digit, at the time when the sampling DP terminates. The carry, no-carry latches are therefore free at this time, to be reset and set up with new information regarding the carry status of the new digit.

Returning to the adder entry terminals, at the same time that the quinary matrix is being energized by the quinary A and B inputs, the binary matrix is also energized by simultaneous binary A and B inputs. Again, three pieces of information must be analyzed to properly determine the binary sum and the carry, no-carry condition to be stored for the next digit. If one A or B input is B0 and the other B5, the input sum is 5. This "B0 and B5" signal is produced by the switch-mix-cathode follower circuitry **558** (Fig. 68*j*) associated with the B0 and B5, A and B input lines. This signal is switched at switches **559** and **591** (Fig. 68*g*) with a "below five" signal from the quinary matrix and a sampling DP to yield a "B5 and No-Carry" sum. It is also switched at switches **562** and **563** with an "above 5" indication from the quinary matrix (this is the five unit carry from quinary to binary) and a DP to yield a "B0 and Carry." If both A and B inputs are B0, this 0 input sum is switched at switches **564** and **565** with "below 5" and DP to give "B0 and No-Carry" and at switches **566** and **567** with "above 5" and DP to give "B5 and No-Carry." The other possibility is that both A and B inputs may be B5. In this case this B0 sum (B5+B5=B0 and carry) is switched at switches **568** and **569** with "below 5" and DP to give "B0 and Carry" and at switches **571** and **572** with "above 5" and DP to give "B5 and Carry." These four possible binary matrix output signals (0 and NC, 0 and C, 5 and NC, 5 and C) are available as DP's from the four cathode followers **573** through **576**. These cathode followers energize the input circuits to the binary latches **577** and **578**, (Fig. 68*b*) and the carry and no-carry latches **556** and **557** in the same way that the quinary latches were energized, i.e., the termination of signal coincidence produces a latch "pull on" signal which overrides the NAP reset.

In this way, as with the quinary latches, the new digit is set up in the binary latches, and the new carry or no-carry indication in the carry and no-carry latches during the beginning of the next digit interval following adder entry. This new information is available from these latches during the remainder of this "next digit interval."

As explained above, three input signals are required to energize the binary matrix; a binary A input, a binary B input and an above 5 or below 5 input from the quinary matrix. If any one of these three inputs is blocked, there will be no output from the binary matrix to alter the status of the binary latches or the carry and no-carry latches.

If the blocking inverters **579** and **581** (Fig. 68*g*) associated with the above 5 and below 5 amplifiers are caused to conduct, they will block entry of the above 5 or below signals to the binary matrix and thus prevent an output signal. This will prevent the status of the carry and no-carry latches from being altered.

This blocking action is used on those occasions where it is desired to prevent the normal resetting of the adder. When the adder reset control latch **555** is turned on, not only is the NAP reset blocked as previously explained, but also the above and below 5 blocking inverters are energized by the output of this latch. The carry and no-carry latches are thus protected so that they will still contain the carry-no-carry indication of the last digit analyzed, even though several digit intervals pass before analysis of the next digit.

The switch circuitry controlling the adder reset control latch is arranged to turn the latch on through switch **585** from D10LB to D0UB on any operation other than left shift or table look-up. On a left shift operation the latch is turned on through switch **582** (Fig. 68*a*) from DXUB to D1UB, and on a TLU operation it is prevented from turning on at all by a signal through mix **583** and inverter **584**.

The need for these adder reset controls is brought about by the two digit gap in the middle of the accumulator at DXU and D0U. On all arithmetic operations except TLU, the accumulator must supply its two words of information to the adder. On all of these operations except left shift, the accumulator "early" output is used. On left shift the "on time" output is used.

When the accumulator early output is used to supply the adder, the information to be analyzed must be presented to the adder one digit time early, so that an "on time" output will be available from the adder for storage back in the accumulator. It will be recalled from the discussion of the accumulator that the D1U position is read out at D10L time and this information stored in the accumulator early latches which are not reset until D1UA time. Thus the D1U information is actually read out three digit times early but is still available for presentation to the adder at the normal "one digit early" time, D0U. When this D1U information is accepted by the adder at D0U time, the carry and no-carry information from D10L must still be available for use by the adder in properly analyzing the D1U information.

This information will still be available in the adder carry, no-carry latches if they have not been reset and have not received any new signal. This blocking of any possible alteration of the carry status is accomplished by the "not left shift" control of the adder reset control latch.

On the left shift operation the accumulator "on time" output enters the adder. It is merged with zeros from the B input and is available from the adder output one digit time later, to be stored in the accumulator one digit position to the left of its original location. In this case the D10L position of the accumulator is read out at D9L time but held in the early latches during D9L time and is presented to the adder from the on time latches during D10L time. For proper left shift operation the D10L information must be available from the adder output at D1U time, for storage in the D1U position of the accumulator. The D10L information cannot be held in the "on time" latches since they are reset at DXUA time. Therefore the D10L information must be transferred to the adder latches by normal adder action at DXUT time and held there through D1U time. The status of the adder latches must not now be allowed to change until after D1U time, when their information is read into the accumulator. The D1U position is read out to the early latches at D10L time as usual and stored there until transferred to the on time latches when the early latches are allowed to reset at D1UA time. This D1U information is presented to the adder from the on time latches during D1U time and the carry or no-carry indication from D10L must still be available at this time for proper analysis of the D1U information. This carry, no-carry indication will still be available in the carry, no-carry latches if they are not allowed to change until after D1U time.

To accomplish this, no new information must be presented to the adder latches during DXU and D0U time, and none of the adder latches must be reset at D0UA or D1UA times.

The left shift gate is closed at DX and D0 time, insuring that no new information will enter the A input during these times, thus blocking any new input to the adder latches. The adder reset control latch is controlled to block latch resets at D0UA and D1UA times.

On TLU operation the NAP adder reset must not be blocked at all. Information enters the adder from general storage or program register and the need for special treatment of DXU and D0U times is not present. The TLU "on" gate from the TLU control latch in the arithmetic control circuits holds the adder reset latch off through the blocking inverter **584**.

The carry insert—no-carry blank and no-carry insert—carry blank circuits are used to force carry or no-carry conditions into the adder for any given digit time. For example, a carry is inserted at D0L time for analysis of the digit to be stored in the D1L position on any complement add operation. The carry insert and no-carry blank signals are always developed together as are the no-carry insert and the carry blank signals. They act on the carry and no-carry latches **556** and **557** to turn one "off" through a blocking inverter and the other "on" through an input mix diode, in accordance with the developed signals.

The zero insert circuit, when energized for a digit time or a series of digit intervals, forces the binary and quinary adder output latches into a zero status for those intervals, regardless of the conditions within the adder. This is done by holding the input grids of B0 and Q0 latches on through an additional mix diode **586** and the B5, Q1, Q2, Q3, Q4 latches off through blocking inverters **587** whenever the zero insert signal is developed. This control is used when a zero output from the adder must be available for storage in the accumulator but no normal zero output would be expected. For example, on add and subtract operations the true or complement add gates which allow accumulator digits to enter the adder "A" input are not opened until D0L time. No normal output can be expected from the adder at D0L time since no entry was made at DXL time. It is necessary, however, that a zero signal be available from the adder output for storage in the D0L position of the accumulator. The zero insert circuit, energized at D0L time, will supply this signal.

OPERATIONAL AND ADDRESS REGISTERS

The operation and address registers are static storage registers with two digit and four digit capacities respectively. Their function is to receive the operation (Op.) code and address portions of the instruction word and to make this information available to the program control, arithmetic control and head selection circuits.

It will be recalled from a previous description of how the machine cycles through a program step, that the new instruction word is read into the program register during an "I" half cycle of the control commutator. A back signal from the program register then advances the control commutator, Figs. 81*a* through 81*l*, to its "Restart" position by causing the restart B latch **605** (Fig. 81*a*) to be turned on. This latch will remain on for one word interval. It is during this interval and under control of this latch that the Op. and D or the I positions of the instruction word are transferred to the Op. and address registers. As the restart latch is turned on, the alternation control flip-flop latch, Fig. 81*d*, is transferred to indicate whether this is the start of a "D" or an "I" half cycle. A restart caused by a program register back signal will always be a restart to "D." During this restart word interval, following a restart to "D," the Op. code must be read into the Op. code register and the "D" address into the address register.

During a word interval following a restart to "I" the "I" address must be read into the address register.

During a TLU operation the "D" address must be altered and stored in the address register.

Figs. 69*a* through 69*e* and 71*a* through 71*l* show a data flow path from the program register to the Op. and address register and read-in switching for the Op. and address registers. On a "D" half cycle it is necessary that this entry switching be open for D5 through D10, the times at which the "D" address and Op. code digits are available from the program register. On an "I" half cycle the entry switching must be open for D1 through D4 when the "I" address digits are available from the program register. During TLU the entry switching must be open for D5 through D10, when the general storage search is changed from one band to the next.

A "Read-In Op. and Address Register" signal is developed whenever an entry is to be made. This signal occurs during D5 through D10 of the restart word on a "D" alternation, D1 through D4 of the restart word on an "I" alternation or D5 through D10 on a TLU band change signal and conditions the input switches of the Op. and address register latches.

The operation code register

The operation code register is a two digit, bi-quinary storage register as shown in Figs. 69*a* through 69*e*. Each digit position has seven latches, one for each bi-quinary bit. Input to the register is from the seven pedestal amplifiers of the program register (Figs. 62*e* and 62*i*). Two of seven, bi-quinary, B–A gates, representative of the digit value in one position of the program register are present on these seven lines each digit time. The Q0 line feeds the input switches **601** and **602** (Fig. 69*e*) of the units and tens Q0 latches **603** and **604**. The Q1 line feeds the Q1 latches, etc. The Op. and address register read-in signal, referred to above, is further switched at the entry to the Op. code register with a D10 and a CP to condition entry to the tens position latches and with D9 and a CP for entry into the units position latches. The restart B signal is switched at switch **606** (Fig. 69*a*) with the output of the "D" alternation latch and a D5–DX gate and at switch **607** with the output of the "I" alternation latch and a D1–D5 gate. The outputs of switches **606** and **607** are mixed at mix **608** and fed to cathode followers **609** whose outputs are fed to switch **611** where a switch occurs with a CP. The output of switch **611** is switched at switch **612** with D10. The output of switch **611** is switched at switch **613** with D9. The switches **613** and **612** drive cathode followers **614** and **615** respectively, which in turn drive the input switches to the units and tens Op. code register latches. Thus the units position will receive only the D9 output from the program register while the tens position will receive only D10.

Assume an Op. code of 71. During D9 time B–A gates will be present on the B0 and Q1 input lines. This can only result in turning on the B0 and Q1 latches of the units position. During D10 time, B5 and Q2 gates will be able to turn on the B5 and Q2 latches of the tens position.

Once on, any latch has a latch back circuit through a mix diode and will remain on until reset by a reset pulse.

The Op. register reset signal is taken from mix **616** through double inverter **617** and cathode followers **618** to the common line that resets all the Op. register latches. This line is pulsed to reset the Op. code register at the beginning of each restart, with a program reset signal from the control console or at D0 of a TLU band change. Mix **616** is fed by a restart A signal, a manual program reset signal or D0 and a TLU band change signal.

The address register

The address register, shown at Figs. 71*a* through 71*l*, consists of four digit positions of bi-quinary latch storage. The "Read-In Op. and Address Register" signal (the output of cathode followers **609**, Fig. 69*a*) which occurs for D5–D10 of a "D" alternation and D1–D4 of an "I" alternation, is further switched at switches **621** through **624** at the entry to each position of the address register. This switching provides a D1 or D5 CP to the entry switches of the units position latches, D2 or D6 CP to the tens position, D3 or D7 to the hundreds, and D4 or D8 to the thousands.

The selection of which of the two possible digit times each position will accept is, of course, dependent upon the "D" or "I" alternation latch.

As with the Op. code register, input to the address register is from the pedestal amplifiers of the program register. These seven bi-quinary input lines are shown in Figs. 71*g* and 71*j*. Each line connects to its associated bi-quinary latch of each position of the register. Thus a "D" address of 1 7 4 6 would be available as gates on the B5–Q1 lines at D5 time, B0–Q4 lines at D6 time, B5–Q2 lines at D7 time and B0–Q1 lines at D8 time. These gates would cause the B5–Q1 units position latches to turn on, the B0–Q4 tens latches, the B5–Q2 hundreds latches and the B0–Q1 thousands latches. An "I" address of 1 7 4 6 would be available as gates on the B5–Q1 lines at D1 time, B0–Q4 lines at D2 time, B5–Q2 lines at D3 time and B0–Q1 lines at D4 time. These gates would cause the same latches to be turned on.

It will be recalled from the section describing the control console and its uses that the address register may be reset to two different values. When the control switch is set to Manual, depression of either the Computer or the Program reset key will reset the address register to blanks. If, however, the Control Switch is in the Run or Address Stop position, the address register will reset to 8000. This is accomplished by dividing the address register latches into two groups, those which are "on" to represent an 8000 address and those which are not. The resets of these two groups are separately controlled. A "set 8000" signal, which turns the latches in the 8000 group on, is provided whenever the Control Switch is in the Run or Stop Address position. The two reset circuits and their latches **625** and **626** are shown at Fig. 71*l*. The "Reset 8000" latch **625** is turned on by any restart, by a TLU, D1, or by either reset key when the Control Switch is in Manual, and when on resets the 8000 latches. The "Reset Except 8000" latch **626** is turned on by any restart, by a TLU, D1, or by either reset key regardless of the setting of the Control Switch, and when on resets all latches except those which represent 8000. In those cases where both reset latches are turned on the register is reset to blanks. When the Control Switch is not in the Manual position, the "Manual Operate–reset" signal is not developed and the "Reset 8000" latch is not turned on. Instead the "Set 8000" signal is developed and the 8000 latches are turned on.

The following summarizes the address register resets:

Control Switch Manual reset address register to blanks.—Control Switch Run or Address Stop resets address register to 8000. Reset 8000 signal resets B5 and Q3 thousands position, B0 and Q0 hundreds, B0 and Q0 tens and B0 and Q0 units. These are all the latches used to represent 8000. The reset 8000 signal is caused by any restart, control switch Manual and computer or program reset keys or TLU, D1.

The reset except 8000 signal resets B0, Q0, Q1, Q2 and Q4 thousands; B5, Q1, Q2, Q3 and Q4 hundreds; B5, Q1, Q2, Q3 and Q4 tens; and B5, Q1, Q2, Q3 and Q4 units. These are all latches except those used to represent 8000. The reset except 8000 signal is caused by any restart, computer or program reset keys regardless of control switch position, or TLU, D1.

The set 8000 signal turns on B5 and Q3 thousands, B0 and Q0 hundreds, B0 and Q0 tens and B0 and Q0 units. These are all latches used to represent 8000. The set 8000 signal is caused by control switch in run or stop address positions and computer or program reset keys.

Some special switching circuits (Figs. 71*b*, 71*d* and 71*e*) provide single gates which indicate when the contends of the register is 8000, 8001, 8002 or 8003 for use by the program control circuits. The outputs of the B5 and Q3 throusands position latches are switched at switch **627** and fed through cathode followers **628** to switch **629**. The outputs of the B0 and Q0 hundreds latches are switched at switch **631** and fed through cathode follower **632** to switch **629**. The outputs of the B0 and Q0 tens latches are switched at switch **633** and fed through cathode follower **634** to switch **629**, and the outputs of the B0 units latch is fed to switch **629**. The output of switch **629** is fed through cathode follower **635** to switches **636**, **637**, **638** and **639**. The output of cathode follower **635** is switched at switch **636** with a NEAP and the Q0 units latch output and fed through double inverter **641** and cathode followers **642** to produce the 8000 address signal. The output of cathode follower **635** is switched at switch **637** with the Q1 units latch output and fed through double inverter **643** and cathode followers **644** to produce the 8001 address signal. The output of cathode follower **635** is switched at switch **638** with the Q2 units latch output and fed through double inverter **645** and cathode follower **646** to produce the 8002 address signal. The output of cathode follower **635** is switched at switch **639** with the Q3 units latch output and fed through double inverter **647** and cathode follower **648** to produce the 8003 address signal. The output of cathode follower **646** is switched at switch **649** with a lower even word gate and fed to mix **651**. The output of cathode follower **648** is switched at switch **652** with an upper odd word gate and to mix **651**. The output of mix **652** is fed through double inverter **653** and cathode followers **654** to produce the 8002–8003 address signal.

Check circuits associated with the address register are shown at Figs. 71*a* and 71*d*. These are provided to check the validity of the address in the register. They check the thousands position for presence of any digit value other than 0, 1 or 8. If any other value is found the "exceed address" double latch **655** is turned on. This turns on the exceed address light on the control console and lowers the "Invalid address" interlock line, removing one of the conditions necessary for a restart of the control commutator and thus stopping calculation. The "exceed address" latch is only reset by a depression of the "Error Reset" key.

When the thousands position is 8, a further check of the hundreds, tens and units position is made to insure that they are 0, 0 and below 5, respectively.

These circuits are straightforward and will not be explained in further detail except to point out the reason for the "Neg. Reset Except 8000" gate. When a reset to 8000 operation occurs, the B5–Q3 thousands, B0–Q0 hundreds and B0–Q0 tens latches are turned on. Variations in the times required for the 8 thousands, Neg. 0 hundreds, and Neg. 0 tens signals to appear at the check switches (8–2D and 8–3D) could allow a voltage spike through to turn on the check latch. The negative reset except 8000 gate prevents the test from being made until the latches have had time to turn on. Inverter **656**, Fig. 71*f*, inverts the reset except 8000 signal to produce the negative reset except 8000 gate.

SEVEN-TO-FIVE AND FIVE-TO-SEVEN CONVERSION CIRCUITS

Information from the distributor (on a store operation) or from read buffer storage (on a read operation), must be changed from its two-of-seven bi-quinary form to the two-of-five system used by general storage. Also, two-of-five information from the general storage readout latches must be changed back to the two-of-seven bi-quinary form before it is sent to the selected storage line or to punch buffer storage.

These conversions are done by the 7 to 5 and 5 to 7 circuits shown at Fig. 72*a* through 72*c*.

Bi-quinary output lines from the distributor and from buffer storage enter the 7 to 5 circuits at Fig. 72*a*. Information on either of these inputs is gated through to the 7 to 5 matrix by the presence of either a RIGS gate (developed by program control on 20 codes) at switches **661** or a BS to GS gate at switches **662** (developed by program control on a 70 code, Figs. 87*a* and 87*b*).

Information gates on the B0 and Q0 input lines must cause similar gates to be present on the "1" and "2" output lines. B0 and Q1 gates must cause "0" and "1" outputs, etc. The circuits that accomplish this are the switch and mix circuits of Fig. 72*b* which switch and mix the several inputs as indicated, and drive the output amplifiers.

The two-of-five output lines from the general storage read-out latches bring information to the 5 to 7 circuits at Fig. 72*c*.

The 5 to 7 switch and mix circuits of Fig. 72*c* accomplish the conversion by switching and mixing the several inputs as shown and drive the output cathode followers. A "No 800X address" gate from the address register is used as an additional condition on each of the 5 to 7 switches. This insures that outputs will occur only for general storage addresses and that there will be no conflict between possible general storage latch outputs and outputs from an 800X location on the selected storage line.

A "negative latch reset pulse" is switched with the "No 800X" signal to cut off the trailing edge of the 5 to 7 output gates sharply at "*a*" time.

SELECTED STORAGE SWITCHING

The selected storage, bi-quinary, data flow lines which take information to distributor, program register and adder can be supplied from any general storage band, upper or lower accumulator, distributor or storage entry switches. It is the function of selected storage switching to gate information through to the selected storage lines from accumulator, distributor or switches in accordance with the particular 800X address.

The details of these circuits are shown at Figs. 73*a* and 73*b*. Here it is seen that the outputs of the accumulator on time latches are fed through switches **671** where the outputs are switched with D1–DX (NAP) to switches **672**. At switches **672** these outputs are switched with the 8002–8003 address signal and NAP and fed to mixes **673** the outputs of which pass through cathode followers to the selected storage output, Fig. 73*b*. The distributor on time outputs are fed through switches **674** where they are switched with the 8001 address signal and NAP to mixes **673**. General storage outputs are fed through cathode followers **675** to the selected storage output and the storage entry switches feed through cathode followers **676**, Fig. 77*d*, to the selected storage output.

PROGRAM CONTROLS

The Program Controls consist of the following:

(1) Address register
(2) Dynamic selection circuits
(3) Address selection switches, control switches and keys, on control console
(4) Operation code register
(5) Operation code analysis circuits
(6) Control commutator

Address register

The address register circuits are shown at Figs. 71*a* through 71*l* and were described above. The outputs from the thousands, hundreds and binary tens latches are taken to the general storage read-write matrix (Figs. 59*a* through 59*o*) where they control the static selection of one of the 40 bands as previously explained, for reading or writing. The outputs from the quinary tens, and both binary and quinary units are used by the dynamic selection circuits (Figs. 71*c* and 71*f*) to select the proper word interval within the statically selected band.

Dynamic selection circuits

The quinary portion of the tens position of an address and both portions of the units position, indicate the word position within a band. It is the function of the dynamic selection circuits to match the outputs from the Q tens and B and Q units latches of the address register with timing pulses to produce a one word gate, the presence of which indicates that the word position specified by the address is about to pass under the heads. This gate will be used to time the opening of the general storage read-in switching (RIGS gate) on all operations requiring read-in to general storage from distributor (20 codes) and to time the opening of distributor or program register read-in switching on operations where they receive a word of information from general storage. This includes all "I" half cycles where the "I" address is a general storage location, and "D" half cycles on 10's and 60's codes where the operand or operator must be transferred from a general storage location to the distributor.

Because of necessary variations in the timings of the general storage, program register and distributor read-in gates and to insure that at least one full word interval will elapse before the general storage read-write circuits are used, after they are energized by the address register outputs, the dynamic selection gate is developed "one word early." That is, the timing gates with which the address register outputs are switched are chosen so that the dynamic selection gate will occur for the word interval preceding the one during which the read-in gate will actually be open. For this reason the dynamic selection circuits produce a gate which is called "One Word Early Dynamic Selection."

This "One Word Early" gate is then switched with timing pulses, in the control commutator, to turn on latches whose outputs are the exactly timed general storage, distributor or program register read-in gates.

The following switches and mixes are made (Figs. 71*c* and 71*f*) to produce the one word early gate: W9, Q0T and S4 at switch **681**; W9, Q1T and S0 at switch **682**; W9, Q2T and S1 at switch **683**; W9, Q3T and S2 at switch **684**; W9, Q4T and S3 at switch **685**. The W9 is inverted by inverter **686** to produce a negative W9 (NW9). This NW9 is switched at switch **687** with Q0T and S0; at switch **688** with Q1T and S1; at switch **689** with Q2T and S2; at switch **691** with Q3T and S3; and at switch **692** with Q4T and S4. W9, W0, W1, W2 and W3 are switched at switch **693** and W4, W5, W6, W7 and W8 are switched at switch **694**. The output of switch **693** is switched with B0Un at switch **695** and the output of switch **694** is switched with B5Un at switch **696**. W9 and W4 are switched at switch **697**, W0 and W5 at switch **698**, W1 and W6 at switch **699**, W2 and W7 at switch **701** and W3 and W8 at switch **702**. The output of switch **697** is switched with Q0U at switch **703**, the output of **698** with Q1U at switch **704**, the output of **699** with Q2U at switch **705**, the output of **701** with Q3U at switch **706**, and the output of **702** with Q4U at switch **707**. The outputs of switches **681**, **682**, **683**, **684** and **685** are switched at switch **708** and the outputs of switches **697**, **688**, **689**, **691** and **692** are switched at switch **709**. The outputs of switches **708** and **709** are mixed by cathode followers **711** and **712** and are fed to switch **713** The outputs of switches **695** and **696** are mixed by cathode followers **714** and **715** and fed to switch **713**. The outputs of switches **703**, **704**, **705**, **706** and **707** are mixed by cathode followers **716** and fed to switch **713**. The output of switch **713** is fed through cathode follower **717** as the one word early dynamic selection signal.

Control console switches, lights and keys

The functions of most of the switches, keys and lights on the control console are such that they may be logically grouped with program controls. These are shown at Figs. 74*a* and 74*b*, 75*a* through 75*f*, 76*a* and 76*b*, and 77*a* through 77*d*.

Master power cutoff switch.—This is a recessed toggle switch **721** (Fig. 74*a*) whose operation immediately cuts off all power to the machine. It is provided for use only as a safety measure and is not used for normal machine operation since it disregards the normal power on and power off operating sequences.

Error sense reset key.—The error sense, error stop and error reset circuits, shown in Figs. 79*a* and 79*b*, will be discussed in a later section. The error sense reset key **722** (Fig. 74*a*) resets the error sense latch, **723** (Fig. 79*a*). When this key is depressed control console terminal **724**, normally at —50 volts, is raised to ground. This raises terminal **725** (Fig. 79*a*) to ground and resets the error sense latch.

Error reset key.—When this key **726** is depressed, terminal **727** is raised to ground. This raises terminal **728** (Fig. 79*b*) to ground, turns off the error stop latch **729** and holds the run latch **731** (Fig. 81*g*) off. Also terminal **732** (Fig. 79*b*) is raised through diode **733**, thus resetting any error detection circuits which may be on. The diode **733** acts as a filter. It allows both the error reset key and the signal from the error reset cathode follower **734** to operate the reset line but only allows the error reset key to turn off the error stop latch. The action of cathode follower **734** will be discussed later in connection with "Error Sense."

Accumulator reset key.—When this key **479** (Fig. 74*b*) is depressed, one of its sections parallels the error reset key and energizes the error reset circuits as above. The other section raises terminal **735** to ground. Terminal **735** connects to the following commoned terminals: **736**, Fig. 78*j*, to reset the overflow sense latch; **737**, Fig. 87*a*, to reset RBS and PBS transfer control latches; **738**, Fig. 64*e*, to reset accumulator to zeros; **739**, Figs. 85*s*, to turn on accumulator reset latch and reset all necessary arithmetic control latches; and **741**, Fig. 61*c*, to reset distributor to zeroes. The action of the accumulator reset key is paralleled by an accumulator reset signal from cathode followers **742** (Fig. 79*a*) on an error sense operation. When this accumulator reset signal is developed it feeds mix **743** and terminal **736** and thus all of the above commoned terminals.

Program reset key.—When this key **744** (Fig. 74*b*) is depressed, one of its sections parallels the error reset key and energizes the error reset circuits. The other section raises terminal **745** (Fig. 75*f*) to ground.

Terminal **745** connects to one segment of the control switch **751** (Fig. 75*f*). If this switch is in the "Run" or "Address Stop" position, a "Set 8000" signal is sent to the address register; if it is on the "Manual" position a "Reset 8000" signal is sent.

Terminal **745** also connects to several commoned terminals as follows: **746**, Fig. 81*a*, to turn on the program reset latch and reset all necessary program control latches; **747**, Fig. 87*a*, to reset the RBS and PBS transfer control latches; **748**, Fig. 71*l*, to turn on the reset except 8000 latch; **749**, Fig. 62*c*, to reset the program register to zeros. The action of the program reset key is paralleled by a program reset signal from the error sense circuit. This signal comes from cathode followers **752** (Fig. 79*a*) on an error sense operation, and connects to the above commoned terminals.

Computer reset key.—When D.C. power comes on, the normally closed computer reset key **481** applies +150 to relays R1 and R2. These relays are energized continuously except when the computer reset key is depressed. When this key is used, R1 and R2 drop. R1–1 parallels the error reset key and energizes the error reset circuits. R1–2 parallels the accumulator reset key and energizes the accumulator reset circuits as above. R1–3 parallels the program reset key and energizes the program reset circuits as above. Thus, computer reset is equivalent to both accumulator and program reset.

Write inhibit circuit.—Before D.C. power comes on, and while computer reset is in process, R–2 points are open. At this time a blanking gate prevents the generation of WSP so that no writing can be done. This WSP blanking gate controls multivibrator **753** (Fig. 53*c*) which develops the WSP.

Checking lights.—The checking lights are provided to give an indication as to where an error has occurred, when one is detected by any of the checking circuits. One side of all the lights is common to terminal **754** (Fig. 75*a*) which is the tap on a voltage divider between —250 and —70. The resistor ratio is such that terminal **754** is at approximately —80. The other side of each light connects to an error detection latch. When the latch is off, its output level is about —35. This 45 volt drop across the neon bulb is not sufficient to fire it. When the latch turns on, its output level rises to about +10 volts. This 90 volts across the bulb causes it to fire.

Clocking light.—This light **755** (Fig. 74*a*) indicates an error in one of the ring circuits. The sector ring error detection circuit output is available at terminal **756** (Fig. 56*b*). The digit and word rings error detection outputs are also sent to terminal **756** and thus the clocking light comes on if an error is detected in any of the ring circuits.

Terminal **756** also connects to terminal **757** (Fig. 79*b*) to turn on the error stop latch.

Accumulator light.—This light indicates that a non-valid bi-quinary digit representation has been detected by the accumulator validity check circuits of Fig. 82*a*. Any extra or missing bi-quinary bits, as indicated by the status of the accumulator on time latches will be detected by the switch-mix validity check circuitry and will turn on the accumulator error latch **758**. The output of this latch raises terminal **759** which connects to terminal **761** (Fig. 74*a*) and lights the accumulator error light.

Two other checking circuits also turn on the accumulator error latch and light. These are the TLU carry or no-carry check and the accumulator zero or non-zero check, shown in Figs. 83*a* and 83*b*.

The output of the accumulator error latch is also one of three conditions which provide an error stop signal at terminal **762** (Fig. 82*c*), to terminal **763** (Fig. 79*b*) to turn on the error stop latch.

Error sense light.—Terminal **764** connects to terminal **766** (Fig. 79*a*). When this terminal is raised by turning on the error sense latch, the error sense light **765** goes on. The circuits to turn on the error sense latch will be described later.

Program register light.—The program register on time latch outputs are checked by validity checking switch-mix circuitry at Fig. 82*b*. Any non-valid digit value will turn on the program register error latch **767**. The output of this latch raises terminal **768** which connects to terminal **769** (Fig. 74*a*) and turns on the program register light **771**.

The program register error latch output is also one of the three conditions which develop an error stop signal at terminal **762** to turn on the error stop latch.

Storage selection light.—This light **772** is turned on when the exceed address latch **655** (Fig. 71*a*) is turned on. This latch is turned on whenever an invalid address appears in the address register or in the event of a failure to write on any general storage read-in operation or a double write situation. The "no write"–"double write" detection circuit is shown as part of the read-write circuitry of Fig. 59*e*. No write is switched with a RIGS gate and a BS to GS gate at switch **673** (Fig. 71*a*) and mixed at mix **674** with double write to provide a no write-double write signal to turn on the exceed address latch **655**.

Distributor light.—The distributor on time latch outputs are checked by validity checking switch-mix circuitry at Fig. 82*c*. Any non-valid digit value will turn on the distributor error latch **775**. The output of this latch turns on the distributor light **776** (Fig. 74*a*).

The distributor error latch output is also one of the three conditions which develop an error stop signal at terminal **762** to turn on the error stop latch.

Overflow light.—The overflow light **777** indicates when an accumulator overflow condition has occurred.

Several different accumulator overflow conditions are detected and used to control machine functioning. When two true members are added, an overflow from the D10U position of the accumulator indicates that its capacity has been exceeded. On a shift count operation an overflow from the D0L position is an indication that the maximum number of shifts have been taken except during the first word time of the operation when the shift number is being transferred to the D0L position of the accumulator.

An overflow is detected by testing for a carry from the adder, during the desired digit interval. If a carry is detected during either of the above situations, the overflow light should be turned on. In addition, the machine should either be stopped or the overflow condition stored for later interrogation by a code 47, depending upon the setting of the overflow sense stop switch **778** (Fig. 75*c*).

The circuits which detect an overflow are shown in Figs. 68*a* and 68*d*. If the switch is set to stop, the overflow stop latch **779** (Fig. 68*a*) will be turned on. The output of this latch is fed through a mix **781** (Fig. 74*a*) to turn on the overflow light. This latch also holds the arithmetic operation latch (Fig. 85*d*) on. With this latch held on, no end of operation signal is developed so the operation interlock latch is never turned off. Thus the control commutator is interlocked and cannot advance into another "D" half cycle to start the next operation. If the switch is set to sense, an overflow sense signal turns on the overflow sense latch **782** (Fig. 78j) instead of the overflow stop latch.

The output of this latch is fed through mix **781** (Fig. 74*a*) and turns on the overflow light. It also is available for later use by the code 47 circuits which will be explained later.

A carry from D1L during a divide operation, when the quotient digit latch is on, indicates a quotient overflow. In this case the machine must be stopped. This signal turns on the overflow stop latch directly, regardless of the setting of the sense-stop switch.

Operating lights.—The operating lights are provided to give an indication of what units in the calculator are in operation.

The punch light **783** comes on at the beginning of a control commutator "D" half cycle on a punch operation (code 70). It remains on until a restart signal is provided by a punch cam which closes as the punch starts its feed cycle. These circuits are shown at Figs. 78*k* and 78*l*. When the read-punch "D" control signal of the control commutator coincides with a 71 from the Op. code register, the output of switch **784** (Fig. 78*k*) turns on the punch light. This signal is available until the read-punch "D" control signal is removed with the next restart. This restart signal is developed at cathode follower **785** with the next D1 after the punch cam back signal applies plate voltage to cathode follower **785**.

The operation of the read light **786** is similar to that of the punch light, except that it occurs on a 70 code.

If the machine stops and either of these lights remain on, it indicates a failure of the read or punch feeds.

The program light indicates that the calculator has stopped because of a condition which has turned the run latch off, thus preventing the control commutator from advancing. These conditions are: (1) Programmed stop, (2) Address stop, (3) Manual program stop, (4) Half cycle.

The run latch **731** (Fig. 81*g*) can be turned off by a stop address signal at terminal **787** (Fig. 81*a*) or by a signal from cathode follower **788**, if the half cycle switch or the program stop key have supplied plate voltage to cathode follower **788**, or by a signal from cathode follower **789** (Fig. 78*h*) when a stop code (01) is sensed and plate voltage has been supplied to cathode follower **789** by the programmed stop switch. When the run latch is off, its off output is available from terminal **787** which is tied to the program light **792**.

The accumulator light **793** will be on all during arithmetic and shift operations. These are the operations in which the accumulator is used. The arithmetic operation interlock latch **791** (Fig. 81*i*), in the control commutator, will be on at these times. Its output turns the accumulator light on.

The data and instruction address lights **794** and **795** indicate whether the control commutator is in an "I" or "D" half cycle, and thus whether the address in the register is a data or instruction address. The alternation control latches **796** (Fig. 81*d*) in the control commutator are connected to form a flip-flop device which alternates on each restart or use D for I signal. The "I" latch output turns the "I" light on. The "D" latch output turns the "D" light on.

Address selection switches.—The address selection switches, Figs. 75*a*, 75*b*, 75*d* and 75*e*, have two functions:

(1) To set up a four digit address which it is desired to manually transfer to the address register.

(2) To set up a four digit address which is to be compared with the contents of the address register to develop an address stop signal when the two addresses are the same, if the control switch is in the address stop position.

The outputs of the 28 address register latches (Figs. 71*a* through 71*l*) connect to the 28 terminals at Figs. 75*d* and 75*e* as shown. These terminals are wired to the binary and quinary sections of each address selection switch as shown. When a decimal value is set in a switch, the wiring is such that a circuit is complete between the binary and quinary common switch segments and those address register latches which represent the decimal value.

To transfer a number from the switches to the address register latches the address is set in the switches. The control switch is set to manual and the transfer key **797** depressed. The following circuit is completed to turn the proper latches on: ground, program reset switch (Fig. 74*b*), transfer key **797** (Fig. 75*c*) to manual terminal of each section of control switch, to binary and quinary common segments of control switch, wiring to binary and quinary common segments of address selection switches, through switches set for a specific address, to above-mentioned terminals to those address register latch output lines (Figs. 71*a*, 71*b* and 71*c*) corresponding to the switch settings to turn the latches on through the latch-back diodes.

With a specific address set in the address selection switches, all 8 lines from the common segments will be raised only when the corresponding address register latches are on. When this occurs the 8 binary and quinary common segments of the control switch will also be raised. If the control switch is in the address stop position, the 8 lines from its address stop terminals will also be raised. This supplies 8 of the 9 necessary inputs to the coincidence switching circuit at Fig. 75*e*. The ninth input is available from the fifth segment of the control switch and is only a check on the position of the switch. The coincidence of these 9 signals at switch **798** indicates that the control switch is in address stop and that the address in the register matches that set in the switches. The output from the diode switches, through double inverter **799** and cathode follower **801**, produces the address stop signal which turns off the run latch, turns on the program light and stops the machine.

Operation and address register lights.—These lights indicate the bi-quinary value of the number in the operation code and address registers.

One side of all lights are commoned to the same —80 voltage divider tap (**754**) as the checking and operating lights. The other side of each light connects to its corresponding address or operation register output latch, via the terminals shown at Figs. 75*a* through 75*f*. Thus the lights are a direct indication as to which latches are on.

Control switch.—The functions of this switch have been covered in connection with the address register and the address selection switches.

In address stop it allows an address stop signal to be developed when the address selection switch setting is matched by the address register latches and causes any reset of the address register to be to 8000 rather than blanks.

In run, it also causes any address register reset to be to 8000.

In manual, it allows the transfer key to transfer a number from the address selection switches to the address register and causes any reset of the address register to be to blanks rather than 8000.

Transfer key.—The function of this key was explained under address transfer.

Program start key.—This key **802** provides a way to manually turn on the run latch and allow the control commutator to advance. It has two sections. In its normal position the section 2 brings ground potential to the program start No. 2 line. This holds the manual start-stop latch off. When the program start key is depressed this circuit is opened. The manual start-stop latch remains off because of the latch back wire, but is now free to be turned on by an input signal. The section 1 of the program start key supplies plate voltage to cathode follower **803** (Fig. 81*g*) which supplies a signal to turn the manual start-stop latch on with the arrival of the next WPU. When the start-stop latch goes on the capacitor coupled cathode follower **804** (Fig. 81*g*) is pulsed. This output turns on the run latch and starts the control commutator. In this way it is insured that the run latch will always turn on at the beginning of an upper word interval.

Program stop key.—This key **805** (Fig. 75*c*) turns the run latch off so that the control commutator will stop at the end of the next restart word interval. This allows the operation in process to be completed but does not allow the next one to start.

When this key is depressed the plate of cathode follower **788** (Fig. 81*a*) is raised to +150 volts. The next restart signal will pulse this cathode follower whose output turns the run latch (Fig. 81*g*) off.

Half cycle switch.—The action of this switch **807** (Fig. 75*c*) parallels that of the program stop key. Thus when it is in the half cycle position the control commutator is stopped at the end of each restart word interval. Each depression of the program start key will allow it to advance until the next restart. Thus the machine may be manually half cycled through program steps.

Programmed stop switch.—When this switch **808** (Fig. 75*c*) is in the stop position and an 01 operation code is sensed the run latch is turned off, stopping the control commutator. This action was described under "program light."

Error stop-sense switch.—This switch **809** (Fig. 75*f*) is provided to allow the use of self-correcting program routines. The machine can be made to respond to a detected error in either of two ways, depending upon the setting of this switch.

(1) Error stop: With the switch in this position an error will cause the run latch to be turned off and the control commutator to be stopped at the end of the operation on which the error was detected.

(2) Error sense: With the switch in this position an error will turn the run latch off momentarily. Automatic program, accumulator and error reset signals will be developed which will reset all units just as if the program, accumulator and error reset keys had been used. The address register will be reset to 8000 (since the control switch is in the run or address stop position) and the run latch will be automatically turned back on. The next instruction will be taken from the switches (8000) instead of the intended "I" address and the machine branches into the self-correcting routine.

Because the action of the error stop-sense switch is tied to the error stop-sense circuits (Figs. 79*a* and 79*b*) these will be described at this time.

The switch **809** has two sections. Terminal **811** connects to —70 volts through a 3.6K resistor so that the input to switch **812** (Fig. 79*a*) is normally at —70 volts. When the switch **809** is closed the input to switch **812** is raised to ground. The other section of the switch **809** brings +150 volts to switch terminal **813** (Fig. 75*f*), raising the anode potential of cathode follower **814** (Fig. 79*a*).

The error disable switch **815** is a control console cover interlock switch which opens when the control console panel is swung out. It provides anode voltage for cathode follower **816** and allows the output of the error stop latch to be used.

Any error detected by the accumulator, program register or distributor validity check circuits or by the TLU carry-no carry check or by the accumulator zero-no zero check produces an error stop signal. This error stop signal is mixed at mix **817** to turn on the error stop latch. Also, any error detected in the operation of the timing rings produces a clocking error signal. This clocking error signal is mixed at mix **817** to turn on the error stop latch. Thus any detected error (except those which operate the exceed address latch) will cause the error stop latch to turn on.

With switch **809** (Fig. 75*f*) in the error sense position the circuits act as follows, after the error stop latch turns on: (1) The error sense latch **723** (Fig. 79*a*) is turned on and lights the error sense light (Fig. 74*a*). The latch **723** is only reset by a subsequent depression of the error sense reset key **722** (Fig. 74*a*). (2) Program, accumulator and error reset signals are developed as a result of the signal coincidence at switch **812** (Fig.

79*a*). These signals parallel the similar signals developed when the accumulator, program and error reset keys are used except that the error reset signal from cathode follower **734** (Fig. 79*b*) does not reset the error stop latch because of the blocking diode **733**. (3) The run latch is turned off with the next EDOU gate by switch **818** and the error restart latch **821** is turned on by switch **819**. The error restart latch remains on through its coincident latch back circuit until the next NEDXU. (4) The error restart latch output connects to mix **822** (Fig. 81*g*) to turn the run latch back on and to mix **823** (Fig. 79*b*) to turn the error stop latch off. The error sense light remains on until the error sense reset key is depressed, as a reminder that an error has occurred.

With switch **809** in the error stop position the output of the error stop latch is only able to turn the run latch off. Switch **812** is blocked as are cathode followers **814** and **816**.

Overflow stop-sense switch.—The functions of this switch were described under overflow light.

Display switch.—The main function of the display switch **824** (Fig. 76*b*) is to select the accumulator (upper or lower), the distributor or the program register for display by the display lights. It also allows the control commutator to cycle continuously if it is in the upper accumulator, lower accumulator, distributor or program register positions but only allows it to cycle once in either of its manual operation positions (R.O. storage or R.I. storage). This switch also removes plate voltage from the general storage read-out latches in its R.I. storage position to prevent a conflict of information on the selected storage line on a R.I. storage operation.

The switch is a 6 position switch with three sections. The third section supplies anode voltage to the upper accumulator, lower accumulator, distributor or program register light selection cathode followers **825**, **826**, **827** and **828**, respectively, depending upon its setting. Outputs from the accumulator on time latches are switched at switches **828** and **831** (Fig. 76*a*) with upper and lower word gates to provide inputs to the upper and lower light selection cathode followers. On time outputs from the distributor and program register drive the distributor and program register light selection cathode followers **827** and **828** (Fig. 76*b*). All these cathode follower outputs are commoned and feed the bi-quinary, light selection output lines. The information on these lines then consists of on time B–A information gates from the selected capacitor storage unit. These lines connect to the information input to the display lights (Fig. 77*d*).

Notice that anode voltage is supplied to the distributor light selection cathode followers by either the distributor, R.O. storage or R.I. storage positions of the switch. On a R.O. storage operation the contents of a storage location, specified by the number in the address register are transferred to the distributor and displayed from there. On a R.I. storage operation the information set in the storage entry switches is transferred to the distributor and then to general storage.

In order for the control commutator to advance normally a run signal must be available when the run latch is on, provided the display switch is in any position other than the two manual operation positions. (R.O. storage and R.I. storage are both manual operations.) Fig. 81*g* shows that the "on" output of the run latch raises the grids of three cathode followers, **832** (Fig. 81*j*), **833** and **834** (Fig. 81*g*). When the second section of the display switch is in the lower accumulator, upper accumulator, distributor or program register positions, +150 volts is applied to the anode of cathode follower **832** and provides a run signal for the control commutator. In the R.O. storage position +150 volts is applied to cathode follower **834**. Thus when the run latch is on it provides the R.O. storage signal instead of the run signal. In the R.I. storage position +150 volts is applied to cathode follower **833**. In this case the run latch, when on, provides a R.I. storage signal instead of the run signal.

On a R.I. storage operation an 8000 control signal is forced by the control commutator due to the man. R.I. storage signal from cathode follower **833**. This opens the path between the storage entry switches and the selected storage output line, allowing the switch information to enter the distributor. (The control commutator also provides a distributor R.I. gate.) Meanwhile, the address register output has energized the general storage read-write matrix. Thus when a RIGS gate is provided, timed by the "one word early" selection circuits, the switch information (now in the distributor) is read into the selected general storage location. Reference to the general storage read-write matrix circuits (Figs. 59*a* through 59*o*) will show that information is available from the selected band as long as a valid general storage address is in the address register. Normally this information operates the five general storage read-out latches, supplying a stream of B–A information gates to the selected storage output line. If this is allowed to happen, there will be a conflict of information from entry switches and the general storage band on the selected storage line. To prevent this the general storage information is blocked on a R.I. storage operation. When the first section of the switch is on the R.I. storage position, +150 volts is removed from terminal **835**, and thus from the plates of the double inverters of the general storage read-out latches **836** (Figs. 59*e*, 59*j* and 59*o*).

Display lights.—There are 10 display light positions (Figs. 77*a* through 77*d*) one for each of the digit positions of the capacitor storage unit to be displayed, arranged in accordance with the bi-quinary code system. There is also a sign position (Fig. 77*a*) having two bulbs, plus and minus, for use when the distributor or accumulator is being displayed. The neon lights respond to coincidence of an information gate and a digit timing gate. This coincidence will occur once every word interval for distributor and program register and once every two word intervals for the accumulator. When the control commutator is stopped so that the information in the capacitor storage units is not changing, the same information will be supplied periodically to the same bulbs. The bulbs so impulsed will appear to glow continuously.

The potential of one terminal of one bulb in each position is determined by the center tap of a voltage divider consisting of two 100K resistors. One end of this divider receives an inverted information gate, the other end an inverted timing pulse. Thus a bulb receives all information gates from its bi-quinary input lines but responds only to those which occur during its digit timing interval. For example, the B5 bulb of D1 receives all B5 gates but responds only to those which occur at D1 time. The other bulb terminal is commoned to a similar terminal of the corresponding bi-quinary bulbs of each of the other digit positions. This common line returns to the plate of a power amplifier whose output is at +230 volts except at "*a*" time.

Normally, the plates of the power amplifier timing gate inverters and the power amplifier information gate inverters are at +250 and therefore the center taps of the bulb dividers are at +250. This 20 volt difference across the bulb is not sufficient to fire it.

The plate of the power amplifier timing gate inverter drops to +80 when the timing gate causes it to conduct, pulling one end of the bulb divider down to +80. If no inverted information gate is present, the other end of the divider remains at +250. In this case the center tap drops to +165. The difference across the bulb is 65 volts, still not sufficient to fire the bulb. If an inverted information gate coincides with the inverted timing gate, both ends of the divider, and the center tap, drop to +80.

This difference of 150 volts is more than sufficient to fire the bulb.

There are two lights in the sign position (D0). The plus light fires when a Q4 information gate coincides with a D0 timing gate. The minus light fires when a Q3 information gate coincides with a D0 timing gate. This follows from the fact that a plus sign is stored in the distributor as a 9, while a minus is stored as an 8. Q4 and Q3 information gates will be available at D0 time only from the distributor or the accumulator sign latches since there is no D0 position in the program register.

Storage entry switches.—These switches (Figs. 77*a* through 77*d*) provide a way to manually insert a word of information into any storage location. They are addressable (8000) and therefore may be used as a source of information by the program routine. As such they may be used as the alternate source of an instruction word on self-correcting error routines.

There are eleven switches, one for each significant digit of a word **838** and one for the sign **839**.

Each switch has two sections, one for the binary level and one for the quinary. Digit timing gates corresponding to the digit position represented are fed to the common segments of each switch. For a given decimal setting of a switch, digit gates will be distributed to the two bi-quinary lines which represent the digit. In the case of the sign position a plus setting distributes a D0 gate to the B5 and Q4 lines, while a minus setting distributes a D0 gate to the B5 and Q3 lines.

These storage entry switch outputs are used either on a manual R.I. storage operation or when the output of the address register is 8000. In the first case a forced 8000 signal (Heat 8000) is developed at terminal **841** (Fig. **81***c*) by action of the control commutator and is applied to mix **842** (Fig. 77*d*). In the second case an 8000 address signal is developed at terminal **843** (Fig. 71*b*) from the address register output and is applied to mix **842**. The output of mix **842** conditions switches **844** (Fig. 77*d*). The storage entry switch outputs are thus gated through switches **844** to the selected storage output lines, by either an 8000 address or a heat 8000 signal, from where they may enter the distributor or program register as the case may be.

Operation code register

The operation code register was described under Basic Principles. The operation code is read into its latches from the D9 and D10 positions of the program register during the restart word interval of a "D" half cycle. Bi-quinary indications of the Op. code value are available from its latch outputs until it is reset by the next restart, a program reset or a TLU band change.

Its output signals are used throughout the machine, wherever energization of a circuit must be restricted to a specific Op. code. Its outputs are primarily used to energize the operation code analysis circuits of the control commutator.

Operation code analysis circuits

The operation code analysis circuits (Figs. 78*a* through 78*l*) consist primarily of diode switching circuits which match combinations of Op. register outputs with timing pulses and gates and signals from the control commutator, to provide signals which allow the machine to function in accordance with the meaning of the Op. code. The outputs of these switching circuits are used to control advancing of the control commutator and by the arithmetic control circuits.

Because these circuits are so closely related to and interlocked with the control commutator, no effort will be made here to describe their detailed functions. The function of each will be discussed in connection with the control commutator.

Control commutator

The control commutator (Figs. 81*a* through 81*l*) is the machine's calculating cycle control device. It consists of a series of latches and diode switch-mix circuitry. In general the latches are turned on and off by combinations of signals from the Op. code analysis circuits, timing pulses and gates, address register signals, signals from the arithmetic control circuits and signals from other units within the commutator. "On" outputs, "Off" outputs and pulses developed when the status of a latch is changed are used as control signals to develop R.I. control gates and to energize the arithmetic controls.

The control commutator can be thought of as advancing through its cycle in "word interval" time units. The number of word times required to complete a commutator cycle will depend on the operation to be performed and often on the location of a word in storage.

In concept the control commutator can be visualized as two branched rings. An "I" half cycle, including restart to "I," selection of "I" address and read-in program step, places the instruction word in the program register. A "D" half cycle, including restart to "D," and subsequent performance of the operation completes the commutator cycle and the program step.

Latches and Signals

It was noted earlier that the operations break down into seven groups. Each group is identified by its Q tens value.

(00 and 01)	No Op and Stop Codes.
(10's and 60's)	Arithmetic Codes.
(20's)	Store Codes.
(30's)	Shift Codes.
(40's and 90's)	Test Codes.
(70's)	Read and Punch Codes.
(84)	TLU.

The action of the control commutator is similar for each of these groups. It will be described by following its action through for typical codes in each of these groups.

Before the control commutator action is explained, the key units and signals and their functions will be listed. These are:

Restart A latch (**845**, Fig. 81*a*): On with any restart signal. Stores A restart signal which may occur at any digit time and allows restart B latch to be turned on at start of next word. Off when restart B turns on.

Restart B latch (**605**, Fig. 81*a*): On at start of next word after restart A turns on. Provides the control commutator restart word time when on. Flips alternation control latch to indicate "D" or "I" half cycle. Conditions Op. and address register entry switching. Off at end of word. Off output conditions "I" control, "D" control, or R–P control signal.

Alternation control latches (**796**, Fig. 81*d*): Flip to opposite status when restart B goes on. Change status with every restart. Also can be flipped from "D" to "I" directly, without a restart, by a "Use D" signal. Conditions Op. and address register entry switching and "I" control, "D" control or R–P control signals.

"I" control signal: Requires coincidence of alt. ctrl. "I," run restart B off. Provides the "I" half cycle control signal.

"D" control signal: Requires coincidence of alt. ctrl. "D," run, restart B off, and operation interlock off. Conditions all Op. codes except 70 and 71 and thus controls the "D" half cycle on all except read and punch operations.

R–P control signal: Requires coincidence of alt. ctrl. "D," run and restart B off. Conditions 70 and 71 Op. codes.

Read code latch (**849**, Fig. 78*k*): On with coincidence of R–P control signal, Op. code 70 and reader intlk. (cam). Conditions RBS transfer and erase control circuits. Conditions cam back signal which provides restart or use "D." Off when restart B turns on and removes "R–P ctrl." on restart. Off when alt. ctrl. flips and removes R–P control signal on "use D."

Punch code latch (846, Fig. 78*k*): On with coincidence of R–P control signal Op. code 71 and punch intlk. (cam). Conditions PBS erase and transfer control circuits. Conditions cam back signal which provides restart. Off when restart B turns on and removes "R–P control."

Advance read feed latch (847, Fig. 78*l*): On with coincidence of read clutch control cam and pulse when read code latch goes on. Provides signal to fire read clutch thyratron. Off when read code latch goes off with restart or use "D."

Advance punch feed latch (848, Fig. 78*l*): On with coincidence of punch clutch control cam and pulse when punch code latch goes on. Provides signal to fire punch clutch thyratron. Off when punch code latch goes off with restart.

Enable read-in latch (851, Fig. 81*b*): Used on all operations requiring a read-in program step, read-in distributor or read-in general storage. When on, it signifies that the Op. code has been analyzed and the read-in operation can start. Insures a delay of at least one word, when a general storage selection is involved, to allow time for general storage selection circuits to settle down. Conditions turning on of start R.I. latch. Off when start R.I. latch goes on.

Start read-in latch (852, Fig. 81*b*): Controls the read-in switching to program register, distributor and general storage. On with coincidence of enable R.I., one word early dynamic selection and timing pulses, when a general storage location is involved. On with coincidence of enable R.I., upper or lower word gates and timing pulse when 8002 or 8003 address is used. On with coincidence of enable R.I. and timing pulse on 8000 or 8001 address. Conditions R.I. program step and storage to distributor R.I. signals and turning on of RIGS latch. Off next ED1 after RIPS. R1D interlock B latch or RIGS latch goes on.

Accumulator to distributor latch (477, Fig. 81*k*): Used only on codes 20, 21, 22, 23 where the word to be stored must be transferred from the accumulator to the distributor from where it can be sent to general storage. Its output controls the distributor R.I. switching at the proper time for the specific operation. Its output also turns on the enable R.I. latch to start development of the RIGS gate in accordance with the one word early signal.

RIPS signal.: Developed by start R.I. and "I" ctrl. Turns program step regeneration off and program step R.I. on.

Storage to distributor R.I. signal: Developed by start R.I. on any operation requiring a read-in to distributor from a storage location. Turns distributor regeneration off and distributor R.I. on.

RIGS latch (853, 81*f*): Turned on by start R.I. on any operation requiring a read-in to general storage from the distributor. Provides RIGS gate to control general storage read-write circuits.

Enable arithmetic codes latch (854, Fig. 81*h*): All arithmetic operations (10's and 60's) require a transfer of an operand or operator from the "D" address to the distributor unless the "D" address is 8001. A distributor backsignal indicates that this transfer has been completed. Coincidence of a distributor back signal and a 10's or 60's Op. code with no 8001 address turns on the enable arithmetic codes latch. The output of this latch conditions all arithmetic codes and allows their signals to be sent to the arithmetic controls. It is turned off with an arithmetic restart signal.

Shift control latch (855, Fig. 78*e*): Turns on for all shift codes. Times the conditioning of shift codes and thus the signals to arithmetic controls.

Overflow sense latch (782, Fig. 78*j*): Turns on when overflow is sensed by test of adder carry. Its status is

tested by an Op. code 47 to provide either a restart or a use "D" signal.

Manual start-stop latch (856, Fig. 81*g*): Held on by N/C program start key. Turns off with next WPU after program start key is depressed. Output turns run latch on. Insures that run latch will come on at beginning of an upper word.

Run latch (731, Fig. 81*g*): Provides the run control signal necessary for the commutator to advance beyond the end of the restart position if the display switch is in accumulator, distributor or program step positions. Provides the manual R.I. and R.O. storage signals when the display switch is set to either of these positions.

Single interlock latch (857, Fig. 81*l*): This latch when off, conditions the development of arithmetic code signals, store code signals, shift code signals and TLU signals. When any of these signals are developed they cause the single interlock latch to be turned on thus preventing further development of the same signals during successive word intervals of the same commutator half cycle. The latch is turned off by any restart or by program reset.

RIPS–RID interlock A and B latches (858 and 859, Figs. 81*c* and 81*f*): These latches provide an interlock to insure that only one RIPS or storage to distributor R.I. signal will be developed on any commutator half cycle. Interlock B must be off to obtain either RIPS or storage to distributor R.I. Either signal when developed turns on interlock A which turns off with the next D1 and turns interlock B on. With interlock B on, one of the conditions necessary for RIPS or storage to distributor R.I. is removed. Interlock B is reset by any restart, program reset or manual R.I. storage.

Distributor back signal latch (478, Fig. 61*d*): This latch turns on with any distributor back signal. It insures that the distributor read-in has been accomplished before allowing a restart signal to be developed, on any operation requiring a distributor read-in. Its output switches with ends RIGS to provide the restart signal on a store operation and with arithmetic restart to provide the restart signal on an arithmetic operation. It is reset by the next storage to distributor R.I. signal or accumulator to distributor R.I. signal.

Operation interlock latch (791, Fig. 81*i*): The off output of this latch is one of the conditions needed for the development of the "D" control signal and for the turning of the start R.I. latch on an 800X address. The arithmetic and shift operations make use of the accumulator and distributor and may require several word times for their completion. The restart signal on these operations is developed during the first operation word interval. Thus the commutator can advance concurrently with the performance of the operation and allow the next instruction to be located, read into program step storage, restart to "D" and transfer the new Op. code and "D" address to the registers. At this point further advance of the commutator cannot be allowed until the preceding operation is completed. This interlocking is done by allowing arithmetic and shift signals to turn on the operation interlock latch, removing one of the necessary conditions for "D" control. When an arithmetic or shift instruction has an 800X "I" address, the location of the next instruction must be prevented until the preceding operation is complete, since the instruction itself is in the process of being developed by the operation. This is accomplished by the Op. interlock off condition necessary for turning on the start R.I.

The action of the program control commutator and Op. code analysis circuits will be described for each of the following conditions:

A. *"I" half cycle,* starting with a restart signal; "I" address portion of instruction word in program register will be used to locate the new instruction.

B. *"D" half cycle,* starting with a restart signal, operations in each of the following code groups will be described.

(1) 00, 01—No Op. and Stop
(2) 10's and 60's—Arith. operations
(3) 20's—Store operations
(4) 30's—Shift operations
(5) 40's and 90's—Test operations
(6) 70 and 71—Read and Punch
(7) 84—Table Look-up

C. Manual R.O. Storage Operation.

D. Manual R.I. Storage Operation.

"I" half cycle—Objectives:

(1) Reset Op. and address registers.

(2) Transfer D1–D4 ("I" address) from program register to address register.

(3) Allow one word early signal to develop RIPS R.I. and regeneration control gate and read-in program step from selected storage, or if "I" address is 800X develop RIPS control gate immediately without waiting for one word early.

Circuit operation:

(1) Restart signal from "D" half cycle switches at switch **861** (Fig. 81*a*) with exceed address interlock and restart B off, to turn on restart A.

(2) Restart A stays on until next DXDP after restart B comes on, if display switch is in one of the run positions or until program reset.

(3) Restart A output resets Op. and address registers and switches with DXCP at switch **862** (Fig. 81*a*) to turn restart B on.

(4) Restart B stays on for one word time, until next NWP, through its latch back circuit.

(5) Restart B output switches with DXDP at switch **863** (Fig. 81*d*) to flip alternation control latches to "I."

(6) Alt. ctrl. "I" output switches with D1–D5 timing gates at switch **607** (Fig. 69*a*) to provide addr. reg. R.I. control.

(7) Alt. ctrl. "I" output turns on "I" Op. light **795** (Fig. 74*b*).

(8) Alt. ctrl. "I" output switches at switch **864** (Fig. 81*d*) with run control signal and restart B off to provide "I" control. This signal becomes available when restart B goes off.

(9) "I" control turns on enable R.I. and conditions RIPS.

(10) Enable R.I. stays on through a latch back circuit until start R.I. comes on.

(11) Enable R.I. output switches at switch **865** (Fig. 81*e*) with one word early dynamic selection gate, D9CP, and No. 800X address to turn on start R.I., if "I" address is in general storage. Here, several word times may pass after enable R.I. turns on before one word early is available, depending upon the location of the instruction.

(12) Enable R.I. output switches at switch **866** with DXCP, operation interlock off, the appropriate upper or lower word gate where necessary, and a specific 800X address, to turn on start R.I., if the "I" address is not in general storage. Here, the only waiting necessary is for the proper upper or lower word time, if the "I" address is upper or lower accumulator.

(13) Start R.I. on output switches at switch **867** (Fig. 81*b*) with RIPS–RID interlock B off and RIGS off to provide a latch back circuit which holds start R.I. on. Its on output also switches with a NEDI at switch **868** in a parallel latch back circuit which attempts to release the latch at every D1 time. The latch thus stays on until the next D1 after either RIPS–RID interlock B or RIGS goes on.

(14) Start R.I. on output switches at switch **869** (Fig. 81*c*) with "I" control, RIPS–RID interlock B off and DXBP to develop the RIPS signal.

(15) RIPS (DXBP) goes to program step capacitor storage (Fig. 62*b*) to operate the read-in and regeneration controls and turns on RIPS–RID interlock A.

(16) RIPS–RID interlock A latches back until the next NEDI. When it unlatches a capacitor coupled cathode follower pulse turns on RIPS–RID interlock B, allowing start R.I. to unlatch as in (13) above and preventing a second development of RIPS signal.

(17) At the end of the program step read-in word time, during which the information from the storage location selected by the "I" address has entered program step storage, the program step R.I. and regeneration control latches are turned off (Figs. 62*h* and 62*l*). When the regeneration latch **871** (Fig. 62*l*) turns off a capacitor coupled cathode follower pulse (program step back signal) is developed and sent to the common restart line to cause a commutator restart.

"D" half cycle: The restart part of a "D" half cycle is similar to that of the "I" half cycle except that the alternation control latches are flipped to "D." This provides D5–D10 time control for the address and Op. register read-in switching resulting in the transfer of digits 5–10 ("D" address and Op. code) from the program register to the Op. and address registers.

When restart B goes off, at the end of the restart word interval, it switches with run control and alt. control "D" to develop read-punch control. This signal is further switched with operation interlock off, to develop "D" control.

From this point on the circuit operation depends on the Op. code. R–P control and "D" control condition the Op. code analysis circuits to allow the operation to start.

No Op. and stop codes (00 *and* 01)—Objectives: On a 00 operation it is only necessary to develop a restart signal and proceed with the next program step.

On a 01 operation a restart must be developed but the run latch must also be turned off so that calculation will stop at the end of the next restart word with the "I" address in the address register and the control commutator ready to continue on the interrupted "I" half cycle.

Circuit operation:

(1) "D" control is switched at switch **872** (Fig. 78*h*) with B0 Tens, Q0 Tens and B0 Units from the Op. code register to develop 0's Op. signal. This signal is separately switched at switches **873** and **874** with Q0 Units and Q1 Units to provide either a No Op. or stop signal.

(2) If No Op. is developed it is used as a restart signal.

(3) If Stop is developed it mixes to provide the same restart signal and if the programmed stop switch is closed also develops a programmed stop signal to turn the run latch off.

Arithmetic codes (10*'s and* 60*'s*)—Objectives: The performance of all 10's and 60's codes requires the transfer of an operator or operand from a storage location specified by the data address to the distributor, unless the "D" address is 8001, in which case the data is already in the distributor and no transfer is needed. This preliminary transfer is common to all 10's and 60's and must be accomplished before those Op. code analysis circuits which send the signals to the arithmetic controls are energized.

To accomplish this a part of the Op. code analysis circuits are tested by "D" control to determine which of the seven general Op. code groups is present. The presence of an Op. Q1 tens indicates that a 10's or 60's is to be performed. This signal must then cause the commutator to advance through an enable R.I., start R.I. sequence as on the "I" half cycle, except that with "D" control present instead of "I" control, a storage to distributor R.I. signal must be developed rather than RIPS. A distributor back signal will then indicate completion of the data transfer and will energize the arithmetic codes of the Op. code circuits.

"If the "D" address is 8001, no transfer is necessary. The test for a 10's and 60's condition develops either Q1T and 8001 or Q1T and no 8001. If the former signal is developed no enable R.I.-start R.I. sequence is necessary. Instead, the arithmetic codes are immediately energized.

If the Op. code is 69 (load distributor) only the enable R.I.-start R.I. sequence is needed. Energization of the arithmetic codes must be blocked and the distributor back signal must instead cause a restart.

Circuit operation ("D" Address not 8001):

(1) "D" control switches at switch **875** (Fig. 81*h*) with Q1T and No 800 to develop 10's and 60's N8001.

(2) 10's and 60's N8001 turns on enable R.I. which stays on until start R.I. comes on.

(3) Enable R.I. switches with one word early, D9CP and No 800X at switch **865** or DXCP, Op. interlock off a specific 800X signal and the appropriate upper or lower word gate where necessary at switch **866**, to turn start R.I. on. Start R.I. stays on until the next D1 time after RIPS–RID interlock B comes on.

(4) Start R.I. switches at switch **876** (Fig. 81*c*) with 10's and 60's N8001, RIPS–RID interlock B off and DXBP to develop storage to distributor R.I.

(5) Storage to distributor R.I. controls distributor R.I. and regeneration latches and turns distributor back signal latch **877** (Fig. 81*h*) off. It also turns on RIPS–RID interlock A which goes off at the next D1 time and turns on interlock B. Interlock B prevents a second development of storage to distributor R.I.

(6) Distributor back signal, a capacitor coupled cathode follower pulse when distributor regeneration latch goes off, switches with 10's and 60's N8001 at switch **878** to turn on enable arithmetic codes latch. It also turns on the distributor back signal latch. (This gate actually lasts from DXL to D1L. The timing overlap of EWL and WL provides a DXL to D5L gate but single interlock will go on at D1L causing the enable arithmetic codes gate to terminate at D1L.)

(7) Enable arithmetic codes stays on until turned off later by the arithmetic restart signal switched at switch **879** with distributor back signal latch output.

(8) Enable arithmetic codes switches at switches **881** and **882** with single interlock off, EWL (D5U to D5L), WL and Op. Q1 tens to provide an enable arithmetic codes gate from cathode follower **883**.

(9) Enable arithmetic codes gate conditions that part of the Op. code analysis circuits (Fig. 78*i*) which make the detailed arithmetic Op. code analysis and send the signals to the arithmetic control circuits.

(10) Enable arithmetic codes gate switches with D1 at switch **884** (Fig. 78*f*) to develop a start 10's or 60's signal which turns the single interlock latch **857** (Fig. 81*l*) on. This terminates the enable arithmetic codes gate.

(11) Fig. 81*i* shows that the output of the switch **882** which develops the enable arithmetic codes gate is taken through a double inverter **885** and cathode followers **883**. When the gate terminates as in (10) the positive shift at the first anode of the double inverter turns the operation interlock latch **791** on through a capacitor coupled cathode follower.

(12) Depending upon the value of the digits in the Op. code register, the Op. code analysis circuits now develop either add, subtract, multiply or divide signals and upper or lower signals. Note that the timing pulses are such that the add, subtract, multiply and divide signals last for DXL and D0L and the upper and lower signals for DXL.

(13) These signals are sent to arithmetic control where they turn on a control latch to initiate the specific operation. The output of any arithmetic control latch mixes to turn on the arithmetic operation latch.

(14) Arithmetic operation turns on arithmetic restart which is reset with the next D9.

(15) Arithmetic restart switches with D6 to develop an arithmetic restart signal. This signal switches with distributor back signal latch output at switch **879** to turn off enable arithmetic codes latch.

(16) When enable arithmetic codes latch **854** goes off a capacitor coupled cathode follower pulse from its inverters first anode is the restart signal. Note that this signal is sent at D6 time of the first word interval following energization of the arithmetic control circuits.

Circuit operation ("D" Address 8001):

(1) "D" control switches at switch **886** with Op. Q1 tens and an 8001 signal from the address register to drive capacitor coupled cathode follower which turns on enable arithmetic codes latch. No enable R.I.-start R.I. sequence is initiated.

Circuit operation (Op. code 69):

(1) An Op. code of 69 would probably always be accompanied by a N8001 address.

(2) "D" control switches with Op. Q1 tens and N8001 to develop 10's and 60's N8001 and initiate an enable R.I.-start R.I. sequence to read-in the distributor.

(3) Code 69 from Op. code circuits (Fig. 78*c*) is mixed at mix **887** (Fig. 81*h*) and inverted by power amplifier **888** to hold enable arithmetic codes latch off. (This same inverter was used to reset the latch on the previous operation.)

(4) Distributor back signal switches at switch **878** with 10's and 60's N8001 and attempts to turn on enable arithmetic codes latch, but fails to do so because of (3) above.

(5) Distributor back signal switches with code 69 at switch **889** (Fig. 78*c*) to provide a restart signal.

Store codes (20's)—Objectives: The store codes provide a means of transferring information from arithmetic storage (accumulator or distributor) to general storage. A store Op. code is always accompanied by a general storage data address.

A transfer from accumulator to general storage is accomplished by first transferring the data from accumulator to distributor and then from distributor to general storage. On these operations the direct data flow path from accumulator on time output to distributor R.I. switching is used.

To do this the control commutator must, upon sensing a store accumulator code, develop an accumulator to distributor R.I. control gate for the proper upper word, lower word or portion of a lower word. Concurrently with the development of this gate it must also initiate an enable R.I.-start R.I. sequence which will allow the one word early signal to time the development of a RIGS gate. This RIGS gate will then allow the transferred information to enter general storage from the distributor. It will be seen that an enable R.I.-start R.I. sequence in combination with a 20's code, will result in turning on the RIGS latch instead of the development of an RIPS or storage to distributor R.I. signal as in the previous cases.

If a store distributor code is sensed, the accumulator to distributor control is not developed. Instead the enable R.I.-start R.I. sequence is started immediately resulting in the development of a RIGS gate and the entry into general storage of the data already in the distributor.

Circuit operation (Store Acc. codes 20, 21, 22, 23):

(1) "D" control switches with Op. B0 tens, Q2 tens and B0 units to develop a store code signal.

(2) This signal is switched at switch **891** (Fig. 81*j*) with single interlock off and DP and further separately switched with the Q units Op. value and timing gates to turn on accumulator to distributor R.I. latch **477** (Fig. 81*k*). The combinations of Q units values and timing pulses are such that the latch is turned on at D10U for code 20, D10L for code 21, D3L for code 22 and DXL for code 23. In each case the latch is turned on two digit times in advance of the time when the distributor R.I. gate must be opened.

(3) The accumulator to distributor R.I. latch is turned off by the next D9 on codes 20 and 21, the next D7 on a code 22 and the next D3 on a code 23.

(4) Accumulator to distributor R.I. output goes to the distributor to initiate the development of the distributor R.I. gate. A capacitor coupled cathode follower

pulse through cathode follower **892** (Fig. 81*k*) is the start accumulator to distributor R.I. signal which controls the distributor regeneration latch and turns off the distributor back signal latch.

The use of these gates and pulses to control distributor R.I. and regeneration was described in connection with the distributor and is shown in the timing chart of Fig. 60.

(5) When accumulator to distributor R.I. is turned off, a capacitor coupled cathode follower pulse through cathode follower **893** is the end accumulator to distributor R.I. signal which controls distributor regeneration and turns the single latch **857** (Fig. 81*l*) on.

(6) The same capacitor coupled cathode follower pulse which interrupted distributor regeneration when the accumulator to distributor R.I. latch turns on is also used to turn on enable R.I. (Fig. 81*b*).

(7) Enable R.I. switches with one word early, D9CP and No 800X at switch **865** (Fig. 81*e*) to turn on start R.I. (Fig. 81*b*).

(8) Start R.I. switches at switch **894** (Fig. 81*f*) with the 20's signal and D10 to turn on RIGS and provide the necessary one digit early, general storage write gate which controls the read-write matrix.

(9) RIGS latch output mixes with RIPS–RID interlock B output and turns start R.I. off on all 20's codes.

(10) RIGS on output is switched with start R.I. off and D10 to turn RIGS off with the next D10 after it is turned on.

(11) A capacitor coupled cathode follower pulse when RIGS goes off is the end RIGS signal which switches with distributor back signal latch on at switch **895** (Fig. 81*h*) to provide a restart.

Circuit operation: (Store Dist. code 24):

(1) "D" control switches with Op. B0 tens, Q2 tens and B0 units to develop the 20's signal.

(2) This signal is switched with single interlock off and DP at switch **896** (Fig. 81*j*) and further switched with Q4 units at switch **897** to turn on enable R.I.

(3) The enable R.I.-start R.I. sequence develops RIGS as in the preceding circuit.

(4) Enable R.I. switches with the 24 code DP signal at switch **898** (Fig. 81*l*) to turn on the single interlock latch.

Shift codes (30'*s*)—Objectives: The four shift operations, shift right (code 30), shift right and half correct (code 31), shift left (code 35) and shift left and count (code 36) are all performed by repositioning the digits stored in the accumulator. No transfer of data between storage locations is required. The program control circuits therefor have little to do except to sense the shift code and send a properly timed signal indicative of the shift operation to the arithmetic controls.

When the shift code is 30 or 35 the program controls must also determine whether or not the shift number (units position of address register) is zero. If it is zero, no signal is sent to arithmetic controls and a restart is developed.

Restart from a shift operation will be similar to an arithmetic operation and operation interlock is required since the performance of a shift operation may require several word times.

Proper timing of the shift signals is accomplished by the shift control latch (Fig. 78*e*). It must be remembered that the "D" control signal, developed when restart B goes off, starts at the beginning of a word interval but the word could be either upper or lower. All shift operations start at lower word time. Shift control is timed to turn on just prior to the beginning of a lower word and insures that the signals will be developed during a lower word interval.

Circuit operation:

(1) "D" control switches with single interlock off Op. B0 tens and Q3 tens at switch **899** (Fig. 78*e*) to develop a shift signal.

(2) This signal is switched with D10U at switch **901** to turn on shift control latch **855**.

(3) Shift control stays on until the next EDXU and thus is on for an entire lower word.

(4) A capacitor coupled pulse through cathode follower **902** (Fig. 78b) when shift control goes off, turns on the single interlock latch.

(5) Shift control switches with Op. B5U and Op. Q1U at switch **903** (Fig. 78*e*) to develop a left shift and count gate for use by the arithmetic controls. This gate is from D10U through D10L.

(6) Shift control switches with Op. B0U and Op. Q1U at switch **904** to develop a shift right and half correct gate for use by the arithmetic controls. This gate is D10U through D10L.

(7) Shift control switches with EDXL and Address 000X to provide a timed signal to be used in testing for the presence of a code 30 or code 35 (switch **905**, Fig. 78*b*).

(8) This shift control, EDXL test signal switches at switch **906** with a blanking gate which is normally up, to turn on the operation interlock latch (Fig. 81*i*). Fig. 78*b* shows how this blanking gate is developed by switching Op. code 30 or 35 with shift control and address 0000 and inverting this signal to form the blanking gate. Whenever all three conditions coincide the gate is down and blanks the turning on of the Op. interlock latch.

(9) This test signal is separately switched with Op. B0U and Op. Q0U at switch **907** to provide a right shift signal for the arithmetic controls and with Op. B5U and Op. Q0U at switch **908** to provide a left shift signal for arithmetic controls. Either of these signals are EDXL gates.

(10) The shift control, EDXL test signal is also switched with Address 0000 and then with either Op. code 30 or 35 and a BP at switch **909** to provide a restart signal on a zero shift number condition. These are the same conditions which develop the blanking gate as in (7) above. Therefore the operation interlock latch is not turned on when the shift number is zero.

(11) When any of the above shift signals is developed it is sent to arithmetic controls where it turns on the shift count, shift right and half correct, left shift or right shift latch and initiates the operation.

(12) The outputs of these arithmetic control latches mix to turn on the arithmetic operation latch.

(13) Arithmetic operation turns on arithmetic restart which stays on until D9 and while on switches with D6 and Op. Q3 tens to provide the restart signal for all shift operations.

(14) The arithmetic operation latch is allowed to reset when the specific arithmetic control latch turns off at the end of the operation.

(15) A capacitor coupled cathode follower pulse when the arithmetic operation latch goes off is the end of operation signal which resets the operation interlock latch **791** (Fig. 81*i*) and allows the commutator to advance.

Test operations (*40's and 90's*)*:* The 40's codes allow tests to be made of accumulator plus, minus, zero, nonzero and overflow no-overflow conditions and provide a means of branching from the main routine if the results of the test indicate minus, non-zero or overflow. The 90's codes allow a test to be made of any distributor position, 1 through 10, for the presence of an 8 or 9 and provide a means of branching from the main routine whenever the position tested contains an 8.

Code 44 tests for an on or off output from an accumulator non-zero double latch **911** (Fig. 84). This latch is turned off every D0U time and turns on whenever an on time output from a D1 through D10 accumulator position is other than B0 and Q0. The zero test of the upper accumulator must therefore be made at DXL time, after each position of the upper accumulator has had a chance to turn the latch on. Code 45 tests the same non-zero double latch except that the test is not made until DXU, after each position of both upper and lower accumulator have had a chance to turn the latch on. In these cases then, the test is made the next EDXL or EDXU after the commutator provides the "D" control signal to energize the Op. code analysis circuits. When the test finds the non-zero latch off, a zero condition is indicated and a restart signal must be developed. If the test finds the latch on, a use "D" signal must be produced.

Fig. 81*d* shows how a use "D" signal is inverted to pull the "D" alt. latch off and turn the "I" alt. latch on. Thus a use "D" effectively replaces "D" control with "I" control in the middle of a commutator cycle. Further advance of the control commutator is in the "I" branch. An enable R.I.-start R.I. sequence occurs and program step storage is read into from the "D" address in the address register.

Code 46 tests to discover whether the accumulator plus sign or accumulator minus sign latch is on. These latches were set up when the last accumulator entry was made. No waiting for a special test time is required and the test is timed by the appearance of "D" control. If the plus latch is found on, a restart is developed. If the minus latch is on a use "D" is developed.

Code 47 tests for an accumulator overflow condition as indicated by the status of the overflow sense latch **782** (Fig. 78*j*). If this latch is off, a restart is signalled. If it is on, the next D1 turns it off and a capacitor coupled cathode follower pulse through cathode follower **912** when it goes off is the use "D" signal.

The 90's codes test the distributor. During the word interval following the appearance of "D" control the Op. code value is switched with digit timing pulses to produce a test gate. The digit interval during which the test gate is developed depends on the value of the Op. code such that:

Code 90 produces a D10, code 91 produces a D1, code 92 produces a D2, code 93 produces a D3, code 94 produces a D4, code 95 produces a D5, code 96 produces a D6, code 97 produces a D7, code 98 produces a D8, code 99 produces a D9. This test gate is then separately switched with the distributor B5–Q3 on time outputs and the B5–Q4 on time outputs to develop either a use "D" or a restart.

The accumulator non-zero latch circuit is shown in Fig. 84. A non-zero condition in any digit position of the accumulator is detected by the circuit of Figs. 83*a* and 83*b*. This non-zero digit gate (Fig. 83*b*) is taken to turn on the accumulator non-zero latch and the accumulator non-zero check latch **912** (Fig. 84). These latches are reset each D0U. When both are on a non-zero test signal is available. When both are off, a zero test signal is available. These are the signals which are tested at either DXL or DXU by codes 44 and 45. The non-zero check latch merely provides duplicate circuitry to insure more reliable circuit operation.

The operation of the accumulator plus and minus sign latches, which are tested by code 46, will be explained under arithmetic controls.

The overflow sense latch whose status is tested by code 47 was explained under overflow light.

Read and punch operations (70*'s*)*:* A read code (70) provides an automatic reader cycle. A reader cycle includes energization of the read clutch, transfer of the information already in RBS to 10 Address selected general storage locations, erasure of RBS to blanks and entry of the new card data into RBS.

A punch code (71) provides an automatic punch cycle. A punch cycle includes energization of the punch clutch, erasure of PBS to blanks, transfer of the information in 10 address selected general storage locations to PBS and punching from PBS of this transferred information.

Table Look-Up (84)—Objectives: The Table Look-Up operation provides a means whereby a stored table can be consulted and the storage address of the argument of the desired function, obtained. Most of the work in performing table look-up operations is done by the arithmetic control circuits. It is only necessary for the program control circuits to sense the TLU code, turn on the TLU control latch **916** (Fig. 86*c*) at the proper time, and receive a TLU end signal from arithmetic controls to provide a restart signal.

The TLU operation allows storage of table arguments in the first 48 word positions of any general storage band. The last two word times of a band (Words 8 and 9 of Sector 4) are used for switching and transfer time by the TLU circuits. In order to be ready to start a search at the beginning of a band, the TLU control latch should turn on during W9 of S4.

Circuit operation:

(1) "D" control switches with B5T, Q3T, B0U, Q4U and Single Interlock off at switch **913** (Fig. 78*h*) to provide a TLU signal from cathode follower **914**.

(2) This signal switches with S4, W9 and D0 timing gates at switch **915** (Fig. 86*c*) to turn the TLU control latch **916** on.

(3) The output of this latch through mix **917** (Fig. 81*l*) turns single interlock latch **857**. The output of the TLU control latch is also sent to arithmetic controls where it energizes the TLU circuits.

(4) Later it will be seen that a carry from the adder on a TLU operation (from TLU carry latch **918** Fig. 86*d*) is a signal that the desired argument has been found. This carry signal turns off the TLU control latch.

(5) A capacitor coupled cathode follower pulse when the TLU carry latch turns off is the restart signal.

ARITHMETIC CONTROLS

The arithmetic operations, add, subtract, multiply, divide, shift and table look-up (10's and 60's, 30's, 84) are accomplished in general by merging, in the adder, distributor and accumulator outputs, or their substitutes, or general storage and distributor outputs. These outputs are merged in sequences as specified by the operation and the result is stored back in the accumulator.

The Accumulator Entry control gates are: (1) Accumulator regeneration. (2) Accumulator read-in. (3) Accumulator reset. (4) Right shift.

The Adder Entry control gates are: Entry A: (1) Accumulator true add. (2) Accumulator complement add. (3) Accumulator true-complement add. (4) Left shift. (5) Table look-up selected storage add. (6) Table look-up program add. (7) Shift number gate. Entry B: (1) Distributor true add. (2) Distributor complement add. (3) Distributor blanking gate. (4) Special digit entry. (5) Early distributor zero control and zero entry. (6) Table look-up on time distributor add.

The way in which the accumulator and the adder respond to these control gates has been discussed in the basic principles sections on the accumulator capacitor storage unit and the one digit adder unit.

It is the function of the arithmetic control circuits, upon receipt of a signal from program indicative of the operation, to interpret this signal and develop the above gates as necessary for accomplishment of the operation. The arithmetic controls must also send "restart" and "end of operation" signals back to program at the proper time to allow further advance of the control commutator and removal of the operation interlock.

All other coded operations (0's, 20's, 40's and 90's, 70's) are accomplished directly by signals developed by the program control circuits.

Several special control circuits which are not associated with any particular operation or group of operations are energized by the arithmetic controls.

These are: (1) Arithmetic operation and arithmetic restart. (2) Upper-lower check (adder control gate). (3) Adder output zero insert control. (4) No carry insert-carry blank and carry insert-no carry blank. (5) Accumulator sign read-out.

The arithmetic control circuits which develop these accumulator, adder and special control gates will be generally covered in the following paragraphs. In a later section specific examples of add, subtract, multiply, divide, shift and table look-up operations will be described with reference to the wiring diagram, timing charts and function charts.

In most cases the accumulator and adder control gates are used by several operations. There are therefore several conditions under which the gate is developed and signals from several different arithmetic operation control circuits are mixed, any of which will produce the gate. In the paragraphs immediately following the description will be confined to the gate and the mix circuits which control it. The multiple mix inputs are labeled so that each operation which develops the gate is seen. The development of the actual operation control signals which feed the mix will be described later under specific examples of arithmetic operations.

Special controls

Arithmetic operation and arithmetic restart.—On arithmetic operations the restart signal is sent back to program control shortly after the operation signal is received by arithmetic control, before the operation is completed. This allows the control commutator to advance on its "I" half cycle, find the next instruction and begin its interpretation, concurrently with the performance of the operation by arithmetic control. The control commutator's operation interlock will prevent its advance beyond the point where there would be conflict between the arithmetic operation in process and an operation called for by the new instruction. At the end of the arithmetic operation in process, an end of operation signal developed by arithmetic control releases the operation interlock and allows the control commutator to advance.

These circuits are shown in Figs. 85*a* through 85*t*.

Key units are the arithmetic operation latch **919** (Fig. 85*d*) and the arithmetic restart latch **921** (Fig. 85*d*).

Circuit operation:

(1) Elsewhere in the arithmetic control circuits an operation set-up latch will have been turned on by a signal from program control.

(2) The outputs from any of these several latches (accumulator true add, accumulator complement add, left shift, right shift, half correct add 5, shift count, divide, multiply, complement result, complement adjust, and overflow stop) mix to turn on the arithmetic operation latch.

(3) The set-up latches stay on, until turned off by the arithmetic control circuit at the end of the particular operation and, while on, hold the arithmetic operation latch on.

(4) The arithmetic operation latch attempts to turn off with each NEDX gate and will finally do so after the particular set-up latch, which is holding it on, turns off.

(5) When the arithmetic operation latch goes on, a capacitor coupled cathode follower pulse, through cathode follower **922**, turns on the arithmetic restart latch. The capacitor coupling insures that arithmetic restart will go on only once for each operation.

(6) Arithmetic restart output is switched with a D6 gate at switch **923** and fed through cathode follower **924** to provide the arithmetic restart signal which is sent back to program control to restart the control commutator. The arithmetic restart latch is reset at D9.

(7) When the arithmetic operation latch goes off with the next NEDX gate after the completion of the operation, a capacitor coupled cathode follower signal through cathode follower **925** is the end of operatoin signal sent back to program control to turn off the operation interlock latch.

(8) Parallel arithmetic restart and end of operation signals are provided by another circuit on a shift count

operation when the D10U position of the accumulator is not zero and therefore the shift count is zero.

Upper-lower check—adder control gate.—It will be recalled from the description of the one digit adder in the section on basic principles that an upper-lower check gate was required as one of the conditions necessary to allow the distributor early outputs through to the adder, in either true or complement form. The purpose of this gate is to insure that either an upper or a lower signal and not both has been sensed and that both reset and no reset are not present before allowing distributor values to enter the adder.

The circuits which develop this gate are shown in Fig. 86*a*.

Key units are the upper control latch **926** (Fig. 86*a*) and the lower control latch **927** (Fig. 86*a*).

Circuit operation:

(1) On all add and subtract upper codes a DXL upper signal is developed in program control. On divide codes, where the divisor is subtracted and added to the upper accumulator, a DXL–D0L gate is developed in program control. These signals mix at mix **928** to turn on the upper control latch. This latch stays on through its latch back circuit until the next NWPL.

(2) On all add and subtract lower codes a DXL signal is developed in program. During TLU operation, a program add signal is developed and, during multiplication, when the multiplicand is to be added in the lower accumulator a DXL signal is developed. These signals mix at mix **929** to turn on the lower control latch. This latch stays on until the next NWPL through its latch back circuit.

(3) The outputs of the upper control and lower control latches mix at mix **931** to give an upper or lower gate from cathode follower **932.**

(4) This upper or lower gate switches with an inverted upper and lower gate at switch **933** to give the U–L check gate from cathode follower **934.**

The upper and lower gate is not normally present except for an error.

(5) Reset and no reset gates, from program, are switched at switch **935** and mixed with upper and lower at mix **936** to provide also an inverted reset and no reset gate as an additional check on the proper sensing of reset signals.

Adder output zero insert control gate.—It will be recalled from the basic principles discussion of the adder that the adder output, zero insert circuit forces zeros on the adder output lines by holding the B0 and Q0 latches on and all others off.

This insertion of zeros into the accumulator is used in several of the arithmetic operations. The circuits which develop the zero insert gate are shown at Fig. 85*e*.

As an example, on the last part of a divide with reset operation, zeros must be entered into the upper accumulator to replace the remainder. The switching at switch **937**, mixed at mix **938** operates the cathode follower **939** to accomplish this.

On add or subtract operations a zero must be inserted at D0L time since regeneration is interrupted for D0L and no other output is available from the adder for entry in D0L position of the accumulator. This is accomplished by the switching and mixing at switch **941** and mix **938.**

The other conditions which require zero insertion will be described later in connection with specific operations.

No carry insert-carry blank, carry insert-no carry blank.—Either a carry or a no-carry condition can be forced into the adder carry, no-carry latches by action of the carry insert-no-carry blank and no-carry insert blank signals. The circuits which provide these gates are shown in Fig. 85*e*.

On add and subtract operations, when the sign analysis indicates that a complement adder entry is to be made from either accumulator or distributor, a carry condition must be forced at D0L time. This causes D1 to be analyzed by the adder just as if a carry had been left from a preceding digit and provides the means of injecting the "elusive one" into the first significant position for correct 10's complement arithmetic.

If no complement entry is indicated a no-carry condition must be set up at D0L time for proper addition of the two true numbers.

When a complement entry is to be made the complement accumulator or distributor signal is developed and energizes the carry insert-no carry blank circuit including switches **942**, **943** and **944**, to insert a carry at D0L, except when the Q0Un digit latch is on during a divide operation.

When a true entry is to be made the lack of a complement accumulator or distributor signal is inverted at inverter **945** (Fig. 85*s*) to a no complement signal to energize the no carry insert-carry blank circuit including switches **946** and **947** (Fig. 85*e*), except when the "H.C. add 5" latch is on during a shift right and half correct operation.

Other conditions requiring a no carry insert-carry blank are: (1) Table look-up program add, D0. (2) Every DXL.

Other conditions requiring a carry insert-no carry blank are: (1) TLU selected storage add, D1. (2) On a divide operation, D1L when the quotient digit latch is on. This injects the "elusive one" into the D2 position when the divisor is being subtracted. The quotient is developed by entries into the D1 position.

The above is accomplished by switching and mixing the various signals as shown at Fig. 85*e*.

Accumulator sign read-out.—The sign of the accumulator is not held in capacitor storage but is stored in the accumulator sign latches **948** and **949** (Figs. 85*o* and 85*t*). These latches are set up when an accumulator entry is made.

The accumulator on time outputs are available for display or for entry to the distributor from D1 through D10 time. At D0 time either a B5 and Q4 or a B5 and Q3 must be supplied, depending on whether the accumulator plus or minus sign latch is on. Normally the sign of the accumulator applies to both the lower and upper. Therefore these B5–Q4 or B5–Q3 indications must be supplied at both D0L and D0U times so that a sign indication will be available for display or distributor entry regardless of which half of the accumulator is being read out.

On a divide operation the sign of the remainder will be the same as the sign of the dividend. At the beginning of each divide operation the accumulator sign is transferred from the accumulator sign latches to the remainder sign latches **951** and **952** (Figs. 85*o* and 85*t*). These latches are not reset until the next mult. or div. operation. The remainder sign is therefore available from the remainder latches after the division is completed.

When a divide without reset is programmed the remainder will be left in the upper accumulator at the end of the operation and the remainder control latch **953** (Fig. 85*l*) will be on. Under these conditions the sign indicated by the remainder sign latches will be supplied at D0U time instead of the sign indicated by the accumulator sign latches. The accumulator sign latches will, however, still control the sign supplied at D0L time.

The above objectives are accomplished by the switch-mix circuitry of Fig. 85*o* which develops D0L and D0U gates on either the accumulator plus R.O. or the accumulator minus R.O. line through cathode followers **954** and **955** by switching and mixing the various signals as shown. The D0L and D0U gates are sent to selected storage switching at Fig. 73*a* where an accumulator plus R.O. raises the B5–Q4 lines and an accumulator minus R.O. raises the B5–Q3 lines.

Accumulator entry controls

Accumulator regeneration control.—The action of the accumulator regeneration gate has been explained in the basic principles section on the accumulator. The circuits which control this gate are shown at Fig. 85*c*. The gate is produced by the on output of the accumulator regeneration latch **956** (Fig. 85*c*). Any operation requiring an accumulator R.I. must cause this latch to turn off at the beginning of the read-in interval and back on at the end of the interval. For example, on any add or subtract operation an accumulator entry is indicated by the turning on of either the accumulator true add latch **957** (Fig. 85*m*) or the accumulator complement add latch **958** (Fig. 85*m*). When either of these latches goes on a mixed signal from mix **959** and cathode follower **961** is developed which turns accumulator regeneration off at ED0L. On these operations it will remain off for two word intervals and will turn back on with the next WPL.

Other operations which require accumulator entries and therefore control the accumulator regeneration latch are multiply, divide, right shift, left shift and TLU.

The accumulator regeneration controls will be described later, in connection with the specific operations.

Accumulator read-in controls.—An accumulator R.I. gate which is up for the R.I. interval must be developed whenever the accumulator is to receive information from the adder. It is developed by the switch-mix circuitry by switching and mixing the various signals as shown at Fig. 85*h* and, from cathode followers **962**, complements the accumulator regeneration gate on all accumulator entries from the adder.

Accumulator reset controls.—On add with reset and subtract with reset operations an accumulator reset gate is developed. The effect of this gate on the accumulator was covered in basic principles under accumulator capacitor storage. This gate is developed by the reset latch **963** (Fig. 85*n*). The latch turns on at DXL time with coincidence at switch **964** of reset and add or subtract signals from the Op. code analysis circuits. It stays on for two-word intervals and turns off with the next NWPL. Its on output is the reset gate which connects to the accumulator at Fig. 64*d* and controls accumulator reset. Its output also mixes at mix **965** (Fig. 85*n*) with a no reset signal from the Op. code analysis circuits to provide a reset-no reset check signal at the output of cathode follower **966** which is one of the conditions necessary to the development of an accumulator true add, accumulator complement add or distributor complement add signal.

The reset latch output also switches with the DXL–D0L add or subtract signal at switch **967** to turn the accumulator plus sign latch on and again to turn the accumulator minus sign latch off.

Accumulator right shift control.—As explained under accumulator capacitor storage, a right shift operation is accomplished by allowing the accumulator early latch outputs to control the pedestal lines. A right shift gate controls the switching which connects the early outputs to the pedestal lines.

Figs. 85*a* and 85*f* show how the output of the right shift latch **968** (Fig. 85*a*) is switched with a negative DXL and D0L gate at switch **969** (Fig. 85*f*) to produce a right shift gate from the outputs of cathode follower **971** which is up for D1L through D10U. This gate does not include DXL and D0L. Entry to these positions is from the adder; a zero insert at DXL time and the shift number at D0L time.

Adder entry controls

Accumulator true add, accumulator complement add, accumulator true-complement add, distributor true add and distributor complement add are included in the adder entry controls.

Add substract controls

Add and subtract operations are accomplished by merging accumulator early outputs and distributor early outputs in the adder. An accumulator true add gate allows accumulator early outputs to enter the adder in true form; an accumulator complement add gate allows accumulator early outputs to enter in complement form, and an accumulator true-complement add gate controls the Q2 entry line, which is the same for both true and complement entries. A distributor true add gate is necessary to allow distributor early outputs to enter the adder in true form, and a distributor complement add gate is necessary to allow distributor early outputs to enter the adder in complement form. The Q2 line here is only controlled by the distributor blanking gate, which is always up for any distributor entry, so that the Q2 line receives the distributor Q2 early output on either a true or complement entry.

On some operations the entire two words of the accumulator early output must be allowed to enter the adder in true form. In these cases the accumulator true add gate must be available for the two accumulator word times and is developed by turning on the accumulator true add latch **957** (Fig. 85*m*). Examples of these operations are: add, subtract, multiply, divide and right shift and half correct.

On other operations a true adder entry from the accumulator early output is only required at DXL or D0L time. In these cases the accumulator true add gate is developed for the necessary digit interval by switch-mix circuitry which parallels and mixes with the accumulator true add latch output.

A complement adder entry of accumulator early outputs is required for some add and subtract operations. This requires an accumulator complement add gate which is developed for the two accumulator word times when the accumulator complement add latch **958** is turned on.

The distributor true add gate, which allows distributor early outputs to enter the adder in true form, is continuously available from the off side of the distributor true control double latch **972** (Fig. 85*r*). Notice that this latch is turned off by every WPL through mix **973**. Any operation which requires a complement distributor entry must therefore turn this latch on. This latch will be turned on for some add and subtract operations and during division when the divisor is subtracted. These same operations must also cause the distributor complement add latch **974** (Figs. 85*m* and 85*r*) to turn on, to provide the distributor complement add gate.

On add and subtract operations the algebraic sign of the operation code, the number in the distributor and the number in the accumulator must be analyzed to determine the proper combination of true or complement entry to the adder which will accomplish the desired result. The switch-mix circuitry which makes this sign analysis and controls the accumulator and distributor true and complement latches is shown at Figs. 85*l* through 85*o* and 85*q* through 85*t*. The key latches are: Distributor plus sign **975** (Fig. 85*r*), distributor minus sign **976** (Figs. 85*q* and 85*r*), accumulator plus sign **948** (Fig. 85*o*), accumulator minus sign **949** (Fig. 85*t*), distributor true control **972** (Fig. 85*r*), distributor complement add **974** (Figs. 85*m* and 85*r*), accumulator true add **957** (Fig. 85*m*), accumulator complement add **958**, complement result **977** (Fig. 85*s*) and complement adjust **978** (Fig. 85*n*).

The sign of the number in the distributor is determined by the value of the D0 position, plus if 9, minus if 8. This sign in transferred to the distributor plus and minus latches at each D0 time, by switching a Q3 output from the distributor with D0 and a CP at switch **979** (Fig. 85*q*) to turn on the minus latch or a Q4 with D0 and a CP at switch **981** (Fig. 85*r*) to turn on the plus latch. Both latches are reset at DXU time with a NWPU. This distributor plus or minus indication is switched with add or subtract and absolute or no absolute Op. code signals at switches **982** and **983** (Fig. 85*q*) and switches **984** and **985** (Fig. 85*l*) to provide a distributor add or distributor subtract signal at the outputs of cathode followers **986** (Fig. 85*l*) and **987** (Fig. 85*q*). These add or subtract signals are from the Op. code analysis circuits and last from DXL through D0L, a two digit gate. The add or subtract Op. code signals are also mixed at mix **988** to provide an add or subtract signal at the output of cathode follower **989**. This signal is used to turn on the reset latch if the Op. code is a reset code and to reset the accumulator plus and minus sign latches to plus, on a reset code. The add or subtract signal is also used to operate the adder output zero insert circuit (Fig. 85*e*) at D0L. This is necessary so that a zero will be placed in the D0L position of the accumulator to satisfy the validity check circuits since there is no normal output from the adder at this time and regeneration is interrupted starting with D0.

The distributor add or distributor subtract signals from cathode followers **986** and **987** (DXL–D0L gates) are further switched at switches **991**, **992**, **993** and **994** (Figs. 85*m* and 85*r*) with accumulator plus or minus indications from the accumulator sign latches to produce either an accumulator true, an accumulator complement or a distributor complement signal from cathode follower **995**, **996** or **997**. The switches **991**, **992**, **993** and **994** are further conditioned by a reset or no-reset check signal from cathode follower **966** (Fig. 85*n*) to insure that either a reset or a no-reset condition has been sensed.

These accumulator true, accumulator complement or distributor complement signals are used to control the distributor true control, distributor complement, accumulator true and accumulator complement latches to provide the gates which allow entry to the adder.

The distributor true control double latch **972** at Fig. 85*r* is turned off with every WP. Thus it will normally be off until a distributor complement signal is developed, and in this off state provides a distributor true gate for true entry of distributor digits to the adder. Where a distributor complement signal is developed, this latch is turned on at ED0 by switch **998**. The distributor complement latch **974** at Figs. 85*m* and 85*r* is also turned on at ED0 by a switch **999**. Both latches stay on until the next WPL. The output of the distributor complement latch is the distributor complement gate which causes the distributor output to enter the adder in complement form. An analysis of the circuit objectives indicates that a distributor complement entry should always be accompanied by an accumulator true entry. This is accomplished by the capacitor coupled cathode follower **1001** (Fig. 85*m*) from the distributor complement latch. When this latch goes on, the accumulator true add latch is turned on by this cathode follower circuit through its latch back diode at switch **1002**. Thus the distributor complement signal provides a distributor complement add gate and an accumulator true add gate. Once on, the accumulator true add latch stays on until the next NWPL and its output mixes at mix **1003** (Fig. 85*d*) to provide the accumulator true gate at the output of cathode follower **1004** (Fig. 85*i*) for true entry of accumulator digits into the adder. Also, whenever this latch is on, it provides an accumulator R.I. gate and a signal to turn on the arithmetic operation latch as described previously under arithmetic restart. The output of the accumulator true add latch is also used as one of the conditions for turning on the overflow stop latch **779** (Fig. 68*a*), when two true numbers are combined and a carry occurs from D10U indicating that the capacity of the accumulator has been exceeded. The accumulator true latch **957** (Fig. 85*m*) when on, mixes at mix **959** to turn off the accumulator regeneration latch and interrupt accumulator regeneration.

When the sign analysis results in an accumulator true signal, this signal is switched with an ED0 at switch **1005** to turn on the accumulator true latch, whose output is used as above. No signal is developed which can affect the distributor true control latch so it is left off and thus provides the distributor true gate.

An accumulator true entry to the adder may be accompanied by either a distributor true or complement entry. When the sign analysis indicates an accumulator true, distributor complement combination the adder controlling gates are developed as above under control of a distributor complement signal. When an accumulator true, distributor true combination is indicated the controlling gates are developed as a result of the accumulator true signal.

When the sign analysis results in an accumulator complement signal it is necessary to develop an accumulator complement gate and a distributor true gate to control adder entry. The accumulator complement signal is switched with ED0 at switch **1006** to turn on the accumulator complement latch. The output of this latch is the accumulator complement gate. It also provides an accumulator R.I. gate from cathode followers **962** (Fig. 85*h*) and controls the accumulator regeneration latch. No signal is developed to affect the distributor true latch so it is left off to develop the distributor true gate.

Notice that when either an accumulator true or an accumulator complement add gate is developed, mix **959** provides the accumulator true or complement gate which controls the Q2 adder entry A line.

There are twelve possible combinations of Op. code, distributor and accumulator signs on add and subtract operations. On six of these twelve combinations it is known that the result sign will be the same as that of the original accumulator number. No change in the status of the accumulator sign latches is required. These are the six combinations where both entries are true. If either entry is to be complement, as indicated by either a distributor complement or accumulator complement signal, the sign of the result will not be known until the operation is complete. The result must then be analyzed for a true or complement condition. This is done by checking for a carry or no-carry from D10U. A carry means the result is a true number and therefore plus. No-carry means the result is a complement and therefore minus. When any complement entry is indicated, both accumulator sign latches will be turned off. A carry from D10U (plus result) will then turn the accumulator plus latch on. No-carry from D10U (minus result) will indirectly cause the accumulator minus latch to be turned on. When the result in the accumulator is complement, as indicated by no-carry, an additional complement adjust cycle must be taken during which the complement number in the accumulator is run through the adder as a complement entry and merged with zeros substituted for the distributor valve. This converted number is read into the accumulator where it is available for further use. A no-carry condition turns on the complement result latch **977** (Fig. 85*s*), to initiate a complement adjust cycle.

If either the complement distributor or the complement accumulator latch has been turned on (at ED0) as a result of the sign analysis, a complement distributor or accumulator gate is developed by mix **1007** (Fig. 85*s*), the double inverter **1008** and the cathode follower **1009**. This gate is switched at switch **1011** with a no complement adjust gate and a no divide gate to provide a complement add sign control gate at the output of cathode follower **1012**, and is inverted at inverter **945** to provide the no complement gate from the output of cathode follower **1013** used by the carry insert circuits. The complement add sign control gate is switched at switch **1014** (Fig. 85*t*) with no multiply, no divide, and D1L to provide an accumulator sign reset signal from cathode follower **1015** (Fig. 85*o*) to turn off both sign latches, and is switched at switch **1016** (Fig. 85*s*) with multiply-divide blank and D10UBP to provide a complement result test signal from cathode follower **1017**. This signal is switched with a carry signal at switch **1018** (Fig. 85*n*) to turn on the accumulator plus sign latch and with a no-carry signal at switch **1019** (Fig. 85*s*) to turn on the complement result latch which is reset at D1L. While the complement result latch is on its output holds the arithmetic operation latch on and switches with EDXL at switch **1021** (Fig. 85*m*) to turn on the complement adjust latch. This latch stays on until the next NWPL through its latch back circuit. The output signal from the off side of the accumulator minus sign latch **949** (Fig. 85*t*) is used at mix **1022** (Fig. 85*s*) as a check to insure that the minus sign latch has been turned on before the complement adjust latch is allowed to turn off. The complement adjust latch output provides the necessary controls for the complement adjust cycle. It is switched with EDXL at switch **1023** (Fig. 85*t*) to turn on the accumulator minus sign latch. It mixes at mix **1024** (Fig. 86*h*) to provide a distributor blanking gate from cathode follower **1025**. It is switched at switch **1026** (Fig. 86*g*) with an accumulator minus sign gate to operate the special digit "add zeros" circuit of Fig. 86*g* for substitution of zeros in place of distributor values for entry into the adder. It mixes at mix **1027** (Fig. 85*c*) to hold the arithmetic operation latch on. It switches at switch **1028** (Fig. 85*e*) with EDXL or ED0L to operate the adder output zero insert circuit. It is inverted by inverter **1029** (Fig. 85*s*) to blank a possible second reset of the accumulator sign latches and a second result test.

On all add or subtract operations, if no complement entry is indicated, the lack of a complement distributor or accumulator gate at the output of cathode follower **1009** (Fig. 85*s*) is inverted to a positive no complement distributor or accumulator gate at the output of cathode follower **1013**. This gate switches at switch **947** (Fig. 85*e*) with D0L and HC add 5 off to provide carry blank and no carry insert signals for setting the adder carry latches to a no-carry status at D0L time. If a complement entry is indicated the presence of the complement distributor or accumulator gate, inverted to negative at inverter **945** (Fig. 85*s*) causes the switches **947** and **946** (Fig. 85*e*) to be blocked so that the carry blank and no carry insert signals are not provided. Instead, the complement distributor and accumulator gate switches at switches **942** and **944** with D0L and quo. digit off, to provide carry insert and no carry blank signals for setting the adder carry latches to a carry status at D0L time. In this way the "elusive one" is injected into the first significant position for correct 10's complement arithmetic.

Left shift gate

A left shift gate allows the accumulator on time outputs to enter the adder. This left shift gate is needed on programmed left shift operations and on multiply and divide operations. On programmed left shifts it is developed by the left shift latch **1031** (Fig. 85*b*). On multiply and divide operations it is developed by the M–D left shift latch **1032** (Fig. 85*k*). In either case the latch output is switched with a negative DX and negative D0 significant digit off. Thus when the significant digit latch **1033** (Fig. 85*b*) is off, the left shift gate is up for D1L through D10L, down for DXU and D0U, up for D1U through D10U and down for DXL and D0L. When the significant digit latch is on, D0 is included in the left shift gate.

Table look-up selected storage add gate and table look-up on time distributor add gate

On a table look-up operation (Fig. 120), the contents of the first 48 storage locations of a general storage band are successively compared with the contents of the distributor. When a number in a general storage location equals or exceeds the searching argument in the distributor, the address of this location is placed in the "D" address positions of the lower accumulator. The comparison is made by merging, in the adder, the complement of the distributor on time outputs with the successive general storage outputs and checking for a carry from the D10U position (at DXL time). A TLU selected storage add gate and a TLU distributor add gate allow these adder entries to be made.

These control gates are developed when the TLU latch **916** (Fig. 86*c*) is on. The latch output is switched at switch **1034** with a D1 through D10 gate and a negative S4, W8 and 9 gate to provide the TLU selected storage add gate from the output of cathode follower **1035** and TLU on time distributor add gate from cathode follower **1036** for D1 through D10 of each word of the band except words 48 and 49.

Table-look-up program add gate

When no equal or higher argument is found in the successive comparison of the first 48 words of a band (0 through 47) the next band must be selected and a similar comparison made. To effect this selection the number in the address register must be increased by 50. This is accomplished during word intervals 48 and 49 by running the program register through the adder and merging it with zeros and special digits substituted for distributor outputs so that 50 is added to its data address. This modified word is then stored back in the program register and the modified data address read into the address register. This action is called program add and is accomplished by the program add gate developed when the program add latch **1037** (Fig. 86*b*) is turned on. The program add latch is turned on by a TLU band change signal which is developed when S4, W8 is reached and no carry has been sensed.

Shift number gate

On shift operations a shift number (units position of address register) determines how many positions the accumulator is to be shifted. The complement of the number in the units position of the address register enters the adder at DXL time where it is merged with a special digit "1" to form a 10's complement. This complement shift number is stored in the D0L position of the accumulator where it is available for entry to the adder to be increased by the "1" at each DXL time at the beginning of each shift "cycle." Shifting then continues until the shift number goes to zero as indicated by a carry from D0L.

The number in the units position of the address register is allowed to enter the adder whenever a DXL, shift number gate is developed. This gate is developed for DXL time of the first word of any left shift, right shift, shift right and half correct or shift left and count operation.

Distributor blanking gate

The distributor blanking gate controls the distributor true and distributor complement gates to allow early distributor outputs or substituted zeros through, to the adder B entry lines. This distributor blanking gate is up for all operations where the distributor early outputs or substituted zeros are used and is down for all operations where special digit values are substituted in place of distributor outputs. It is necessary to prevent a conflict of information from the two sources on the adder input lines.

The gate, which is normally up, is lowered by the inverted switch and mix circuitry outputs shown at Fig. 86*h*. It is lowered by all special digit gates, by the right shift gate, left shift gate, left shift latch, complement adjust gate, TLU selected storage add gate and the M–D left shift latch.

Special digit gates

The special digit circuits provide a means of supplying specific digit values to adder entry B. They are used to change the value contained in an accumulator position as necessary to accomplish the operation. The special digit circuits are used primarily in the shifting and TLU operations.

Each special digit gate raises two adder B entry, bi-quinary lines to provide an adder B entry representative of the particular digit gate. For example, an add zero gate raises the B0–Q0 lines, etc.

The gates are produced by switch-mix circuitry shown at Figs. 86*e* through 86*h*. Each operation requiring a special digit entry actuates the necessary special digit circuit for that time interval during which the accumulator digit which is to be changed is presented to the adder A entry. For example, on left shift, right shift and M–D left shift, a "one" must be entered at DXL to combine with the shift count number stored in the D0L accumulator position, which is presented to adder A entry from the accumulator early output at DXL time. This increases the shift number by one. This is accomplished by mixing the outputs of the right shift latch, the left shift latch and the M–D left shift latch at mix **1038** (Fig. 86*g*) and switching this signal from cathode follower **1039** with a DXL timing gate at switch **1041** to raise the add one line for DXL time.

Distributor zero entry and control gates

On operations such as add or subtract lower, add or subtract upper, multiply, divide, etc., the entire two words of the accumulator enter the adder via adder entry A. The contents of the distributor enters adder entry B to merge with either the upper or lower accumulator word as the case may be. On an add lower operation, zeros must be entered at entry B, in place of the distributor early outputs, during the time that the upper word is entering the adder. On an add lower operation, zeros must be substituted for the distributor values during upper word time.

This is accomplished by the early distributor zero control gate and the early distributor zero entry gate. The zero control gate blocks the early distributor outputs and the zero entry gate raises the B0–Q0 lines to allow a true or complement zero entry to adder entry B.

These gates are developed by switch-mix circuitry under control of the upper and lower word control latches **926** and **927** (Fig. 86*a*). These latches are turned on at the beginning of a lower word interval by an upper, lower, divide or multiply signal from the Op. code analysis circuits or by a TLU signal. They remain on until the next DXL. While on, their outputs switch with upper word or lower word timing gates as shown in Figs. **86*a***, 86*b*, 86*c* and 86*d* to provide the zero control and zero entry gates.

A parallel circuit develops these gates for each D10 interval. This supplies a zero to fill the gap created by the missing DX position of the distributor (if there were a DX position it would be read out at D10 time).

Another parallel circuit develops these gates for each DX interval to substitute a zero early output in place of the sign indication (8 or 9) contained in the D0 position for entry to the adder. The sign is only used when the distributor word is sent to general storage or displayed.

Circuit operation for add and subtract

An add or subtract operation is accomplished by merging the two accumulator words in the adder with distributor outputs and zeros. On an add or subtract upper operation the lower accumulator word will merge with zeros and the upper with distributor values. On an add or subtract lower, zeros will be merged with the upper word and the lower will merge with the distributor values. A sign analysis will indicate whether the accumulator and distributor entries are to be true or complement and will set the accumulator sign latches to properly indicate the sign of the sum.

It will be recalled from program control, that each 10's and 60's operation is preceded by a control commutator enable R.I.-start R.I. sequence which places the operator in the distributor. A distributor back signal at the completion of this distributor read-in turns on the enable arithmetic codes latch whose output switches with single interlock off, EWL and WL to provide a DXL through D4L gate which which energizes the arithmetic Op. code analysis circuits. Each arithmetic operation will be considered as starting at the beginning of this lower word interval, when the DXL–D0L add, subtract, multiply or divide signals and the DXL upper or lower signals energize the arithmetic controls. Of course other signals such as absolute, no absolute, reset and no reset will be available continuously as long as "D" control is up.

Code 10—Add Upper.

Conditions: Op. code plus, distributor sign plus, accumulator sign plus.

Signals from Program: No Reset, No Absolute, "D" Control, Upper (DXL), and Add (DXL and D0L).

Objectives:

(1) Analyze signs and develop accumulator true add gate and accumulator true-complement add gate for D0L through D10U.

(2) Turn on arithmetic operation latch, restart.

(3) Leave accumulator plus sign latch on since sum will be plus.

(4) Develop accumulator R.I. gate for D0L through D10U.

(5) Remove accumulator regeneration gate for D0L through D10U.

(6) Turn upper control latch on and develop distributor zero control and entry gates for the lower word time (DXL through D10L); develop U–L check gate.

(7) Operate adder output zero insert circuit for D0L time to put zero in D0L position of accumulator since regeneration gate is down and no other output is available from adder.

(8) Because there is no complement entry, insert a no-carry at D0L.

(9) Allow arithmetic operation latch to turn off at completion of operation. This will develop the end of operation signal to turn off operation interlock and allow the commutator to advance.

Circuit operation:

(1) Add signal from program (DXL and D0L) (Fig. 78*i*) switches with no absolute from program (Fig. 78*f*) and distributor plus latch **975** (Fig. 85*r*) output at switch **984** (Fig. 85*l*) to give an add distributor signal (DXL and D0L) from cathode follower **986.**

(2) Add signal from program also mixes at mix **988** to give add or subtract signal which operates adder output zero insert circuit at D0L.

(3) No reset signal from program (Fig. 78*c*) mixes at mix **965** (Fig. 85*n*) with reset latch output (not on for this operation) to give reset-no reset check gate from cathode follower **966.**

(4) Add distributor signal switches with reset-no reset check and accumulator plus latch output at switch **991** (Fig. 85*m*) to give a DXL and D0L accumulator true add signal.

(5) Accumulator true add signal switches with ED0 at switch **1005** to turn on the accumulator true add latch **957**, which latches back until the next NWPL.

(6) Accumulator true add latch output turns on arithmetic operation latch **919** (Fig. 85*d*) which provides operation interlock through cathode follower **725** to operation interlock latch **791** (Fig. 81*i*) and arithmetic restart through cathode follower **922**; mixes at mix **1042** (Fig. 85*h*) to become accumulator R.I. gate; switches at switch **1043** (Fig. 68*d*) with D10U, carry pulse from adder, distributor true and distributor blanking gate to indicate an overflow if accumulator capacity is exceeded when two true numbers are added; mixes at mix **1003** (Fig. 85*i*) to develop the accumulator true add gate; mixes at mix **959** to control accumulator regeneration.

(7) Accumulator true add gate mixes at mix **1044** (Fig. 85*i*) to develop accumulator true-complement add gate.

(8) No signal is developed to turn the distributor true control latch **972** (Fig. 85*r*) on; it stays off and continues to develop the distributor true control gate. This gate switches at switch **1045** (Fig. 67*g*) with distributor blanking gate and U–L check to develop the distributor true add gate. It also switches at switch **1043** (Fig. 86*d*) to detect an accumulator overflow as in (6) above.

(9) No signal is developed to change the status of the accumulator sign latches (plus on, minus off).

(10) Upper control latch turns on (DXL) and switches with WL at switches **1046** and **1047** (Fig. 86*a*) to develop distributor zero control and entry gates. It also mixes at mix **931** to develop the U–L check gate.

(11) Accumulator lower word enters adder, merged with zeros from distributor zero entry. Adder output enters lower accumulator positions.

(12) Accumulator upper word merges with distributor early outputs. Adder output enters accumulator upper word positions.

(13) When accumulator true add latch **957** (Fig. 85*m*) goes off with NWPL, no parallel signal has been provided to hold arithmetic operation latch **919** (Fig. 85*d*) on any longer. It turns off with NEDX and provides end of operation signal to turn operation interlock **791** (Fig. 81*i*) off and allow commutator to advance.

Code 10—Add Upper.

Conditions: Op. code plus, distributor sign plus, accumulator sign minus.

Signals from Program: No Reset, No Absolute, "D" Control, Upper (DX), and Add (DXL and D0L).

Objectives:

(1) Analyze signs and develop accumulator complement add gate and accumulator true-complement add gate for D0L through D10U. Retain distributor true add gate.

(2) Turn on arithmetic operation latch, restart.

(3) Because a complement entry is indicated, develop complement add sign control gate to reset accumulator sign latches.

(4) Develop accumulator R.I. gate for D0L through D10U.

(5) Remove accumulator regeneration gate for D0L through D10U.

(6) Turn upper control latch on to develop distributor zero control and entry gates for the lower word time (DXL through D10L) and to develop U–L check gate.

(7) Operate adder output zero insert circuit for D0L time to provide a zero to store in accumulator D0L position.

(8) Because a complement entry is indicated, block the no-carry insert at D0L and provide a carry insert.

(9) Wait for carry or no-carry at D10U to indicate whether sum is plus or minus. If sum is plus (D10U Carry): (*a*) Turn on accumulator plus sign latch. (*b*) Allow accumulator regeneration to turn on, remove accumulator R.I. gate. (*c*) Allow arithmetic operation latch to turn off; turn off Op. interlock and advance commutator.

If sum is minus (No Carry D10U), number in accumulator is complement and must be converted. Minus sign latch must be turned on: (*a*) Turn on complement result latch instead of plus sign latch. Turn on complement adjust latch. (*b*) Hold arithmetic operation latch on; hold accumulator R.I. gate on and regeneration off; turn on accumulator minus sign and provide necessary gates to enter accumulator complement and merge with zeros.

Circuit operation:

(1) Add signal from program switches with no absolute and distributor plus to give add distributor signal as in (1) above under circuit operation.

(2) Add or subtract signal is also developed as in the preceding circuit, to insert zero at D0L.

(3) No reset signal develops reset-no-reset check gate as before.

(4) Add distributor signal switches with reset-no reset check and accumulator minus latch output at switch **993** (Fig. 85*m*) to give a DXL and D0L accumulator complement add signal.

(5) Accumulator complement add signal switches with ED0 at switch **1006** to turn on accumulator complement add latch **958** until an NWPL time, when no input signal is available to override the unlatching action.

(6) Accumulator complement add latch output turns on arithmetic operation latch, provides operation interlock and arithmetic restart as above; mixes at mix **1048** to provide accumulator R.I. gate; provides accumulator complement add gate; mixes at mix **1044** (Fig. 85*i*) to give accumulator true-complement add gate and mixes at mix **959** to turn off accumulator regeneration.

(7) Accumulator complement add latch output also mixes at mix **1007** (Fig. 85*s*) to provide accumulator or distributor complement gate.

(8) No signal is developed to turn the distributor true control latch on; it stays off and continues to develop the distributor true control gate which with distributor blanking and U–L check give distributor true add.

(9) Accumulator or distributor complement gate is inverted at inverter **945** to block no carry insert at D0L and also switches at switch **942** (Fig. 85*e*) to provide a carry insert at D0L for correct 10's complement arithmetic.

(10) Accumulator or distributor complement gate switches at switch **1011** (Fig. 85*s*) with divide off and no complement adjust to give a complement add sign control gate which is up until complement adjust comes on.

(11) Complement add sign control switches at switch **1014** (Fig. 85*t*) with multiply off, divide off and D1L to reset the accumulator sign latches. This is at the beginning of the two accumulator entry word intervals.

(12) Upper control latch turns on (DXL) and switches with WL at switches **1046** and **1047** (Fig. 86*a*) to develop distributor zero control and entry gates; it also mixes at mix **931** to develop the U–L check gate.

(13) Accumulator lower word enters adder, merges with zeros supplied by zero control and zero entry circuits. Adder output enters lower accumulator positions controlled by accumulator R.I. gate.

(14) Accumulator upper word merges with distributor early outputs. Adder output enters upper accumulator positions. D10U positions of accumulator and distributor enter adder at D9U time. Carry or no carry pulse, as a result of analysis of D10U position, is available from adder during D10 time.

(15) Complement add sign control gate used in (11) above is still up. This gate switches with D10U, BP and M–D blank at switch **1016** (Fig. 85*s*) to give a D10U, complement result test gate.

(16) Complement result test switches with carry pulse from adder at switch **1018** (Fig. 85*n*) to turn on accumulator plus latch if result of operation is plus (true). In this case accumulator complement add latch turns off with next NWPL, arithmetic operation latch is allowed to unlatch with NEDX, operation interlock turns off and commutator advances.

(17) Complement result test switches with no carry from adder at switch **1019** (Fig. 85*s*) to turn on complement result latch **977** if result of operation is minus (complement). This latch resets with the next D1L through its reset inverter **1049**.

(18) Complement result latch output mixes at mix **1027** (Fig. 85*c*) to help hold arithmetic operation on through its unlatching time, and switches with EDXL at switch **1021** (Fig. 85*m*) to turn on complement adjust latch and to hold accumulator complement add on through its unlatching time.

(19) Complement adjust latch output also mixes to help hold arithmetic operation on; switches at switch **1023** (Fig. 85*t*) to turn accumulator minus sign on; mixes at mix **1024** (Fig. 86*h*) to lower distributor blanking gate for DXL through D10U; switches at switch **1026** (Fig. 86*g*) with accumulator minus to operate the special digits add zeros circuit to supply zeros to adder entry B for the conversion word intervals (DXL through D10U); switches with DXL and D0L at switch **1028** (Fig. 85*e*) to operate the adder output zero insert circuit for the conversion, just as the add-subtract signal did for the add cycle (zero insertion is needed for DXL as well as D0L here because accumulator regeneration is not functioning) is inverted at inverter **1029** to remove the complement add sign control gate and prevent a second complement result test.

The complement adjust latch stays on through its latch back circuit until the next NWPL after the accumulator minus sign latch has been turned on.

(20) With the above gates developed, the contents of the accumulator will again be run through the adder in complement form, merged with zeros supplied by the special digits add zero circuit and entered back in the accumulator.

(21) Complement adjust turns off with NWPL after conversion.

(22) Accumulator complement add also turns off with this NWPL.

(23) Accumulator true or complement add gate is removed allowing accumulator regeneration to turn on with WPL.

(24) Arithmetic operation turns off with NEDX, operation interlock turn off and commutator advances.

Code10—Add Upper.

Conditions: Op. code plus, distributor sign minus, accumulator sign plus.

Signals from Program: No Reset, No Absolute, "D" Control, Upper (DXL) and Add (DXL and D0L).

Objectives:

(1) Analyze signs and develop distributor complement add gate for D0L through D10U, accumulator true add gate and accumulator true-complement add gate for D0L through D10U.

(2) Turn on arithmetic operation latch, restart.

(3) Develop complement add sign control gate to reset accumulator sign latches.

(4) Develop accumulator R.I. gate for D0L through D10U.

(5) Remove accumulator regeneration gate for D0L through D10U.

(6) Turn upper control latch on to develop distributor zero control and entry gates for the lower word (DXL through D10L) and to develop U–L check gate.

(7) Operate adder output zero insert circuit for D0L time.

(8) Block no carry insert, provide carry insert.

(9) Wait for carry or no carry at D10U to indicate whether sum is plus or minus.

If plus (D10U carry): (*a*) Turn on accumulator plus sign latch. (*b*) Allow accumulator regeneration to turn off; remove accumulator R.I. gate. (*c*) Allow arithmetic operation to turn off; turn off operation interlock and advance commutator.

If sum is minus (No carry D10U): (*a*) Turn on complement result instead of plus sign. (*b*) Turn on complement adjust. (*c*) Hold arithmetic operation on; hold accumulator R.I. gate on and accumulator regeneration off; turn on accumulator minus sign and provide necessary gates to enter accumulator in complement form and merge with zeros.

Circuit operation:

(1) Add signal from program switches at switch **983** (Fig. 85*q*) with no absolute and distributor minus to give a DXL and D0L, subtract distributor signal at the output of cathode follower **987**.

(2) Add or subtract signal is developed at cathode follower **989** (Fig. 85*l*) to give zero insert at D0L.

(3) No reset signal develops reset-no reset check gate as before.

(4) Subtract distributor signal switches with reset-no reset check gate and accumulator plus at switch **994** (Fig. 85*r*) to give a DXL and D0L distributor complement add signal at cathode follower **997**.

(5) Distributor complement add signal switches with ED0 at switch **998** to turn distributor true control on and remove the distributor true add gate at adder B entry.

(6) Complement distributor signal also switches with ED0 at switch **999** to turn on the complement add distributor latch which stays on for two words through its latch back circuit until the next NWPL.

(7) As the complement add distributor latch goes on a plus pulse from its right hand inverter anode feeds a capacitor coupled cathode follower **1001** (Fig. 85*m*) to turn on the accumulator true add latch through its latch back switch-mix circuit.

(8) The distributor complement add latch provides the distributor complement add gate at adder entry B and mixes at mix **1007** (Fig. 85*s*) to give the accumulator or distributor complement add gate at the output of cathode follower **1009**.

(9) The accumulator true add latch provides the accumulator R.I. gate, the accumulator true add gate and turns accumulator regeneration off. Accumulator true add gate mixes at mix **959** (Fig. 85*m*) to give accumulator true-complement add gate.

(10) From this point on the circuit operates as in the preceding circuit description. A carry from D10U turns on the accumulator plus latch and allows the commutator to advance.

No carry turns on complement result, complement add accumulator and complement adjust and a conversion sequence results.

Code 10—Add Upper.
Conditions: Op. code plus, distributor sign minus, accumulator sign minus.
Signals from Program: No Reset, No Absolute, "D" Control, Upper (DXL), and Add (DXL and D0L).

Objectives:

(1) Analyze signs and develop distributor true add gate, accumulator true add gate and accumulator true-complement add gate, for D0L through D10U.

(2) Turn on arithmetic operation latch, restart.

(3) Leave accumulator minus sign latch on since the sum will be minus.

(4) Develop accumulator R.I. gate, D0L through D10U.

(5) Turn off accumulator regeneration gate, D0L through D10U.

(6) Turn on upper control latch and develop distributor zero control and entry for the lower word time (DXL through D10L); develop U–L check gate.

(7) Adder output zero insert at D0L.

(8) Insert n-carry at D0L.

(9) Allow arithmetic operation to turn off at completion of operation and advance commutator.

Circuit operator: The circuit operation except for the sign analysis is identical to that of accumulator plus, distributor plus.

Code 11—Subtract Upper: The subtract upper code (11) can be accompanied by the same accumulator and distributor sign combinations as explained in connection with code 10. The sign analysis circuits function to develop the necessary gates in a way similar to that above. The circuit operation then is the same as one of the four described under code 10.

Codes 15 and 16—Add or Subtract Lower: The circuit operation for these codes is similar to that for code 10. The difference is that the distributor early outputs must merge with the accumulator lower word and the distributor zero control and entry circuit must function in conjunction with the accumulator upper word. This difference is controlled by the turning on of the lower control latch **927** (Fig. 86*a*) by a lower signal from program instead of the upper control latch as in code 10. This causes the distributor zero control and entry circuit to be operated for an upper word time and also develops the U–L check gate needed for development of the distributor true add gate.

Codes 17 and 18—Add or Subtract Absolute Lower: In these codes the sign of the distributor is ignored. In add or subtract distributor signal (DXL and D0L) is produced solely on the basis of the operation sign and is switched with the accumulator sign indication to provide the necessary true or complement gates. Reference to the Op. code analysis circuits of Figs. 78*a* through 78*l* will show that the absolute codes always cause a lower signal to be developed. This causes the lower control latch **927** (Fig. 86*a*) to be turned on and results in merging the accumulator lower word with distributor early outputs and the upper word with zeros from the zero entry circuit. Circuit operation is the same as explained under code 10.

Add or subtract with reset

Codes 60, 61, 65, 66, 67, 68: These are the reset codes. The circuit operation is the same as described under codes 10, 11, 15, 16, 17 and 18 except that a reset signal from program replaces the no-reset signal. This causes the reset latch **963** (Fig. 85*n*) to be turned on when the add-subtract DXL and D0L signal is developed. The output of the reset latch provides the reset-no-reset gate at cathode follower **966**; is the reset gate sent to the accumulator at Fig. 64*d* which resets the accumulator to zeros and forces zeros from the accumulator early outputs as explained under accumulator capacitor storage; switches with the DXL and D0L add or subtract signal at switches **967** (Fig. 85*n*) and **1051** (Fig. 85*t*) to reset the accumulator plus latch on and the accumulator minus latch off so that an accumulator plus condition will be available for the sign analysis.

Circuit operation for shifting

There are four shift operations: (1) Code 30—shift right. (2) Code 31—shift right and half correct. (3) Code 35—shift left. (4) Code 36—shift left and count.

These operations provide a means of repositioning the number in the accumulator. Right shifting is accomplished by allowing each accumulator early digit output to control the pedestal lines. Thus D2L is read out at D1L time, controls the pedestals and reads back into the D1L position at D1L time. Each digit successively follows this procedure for the two accumulator word times, with the result that the accumulator number is shifted one position to the right. The number of shifts taken is controlled by the value of the units position of the data address (in the address register) which accompanies the shift Op. code. Each shift requires two word times and in the following descriptions, each of these "two word intervals" will frequently be called a "shift cycle."

On the first cycle of each shift operation the units position of the address register enters the adder at DXL time in complement form and merges with a "one" supplied by energizing the special digits add one circuit. This sum, the 10's complement of the shift number, is stored in the D0L position of the accumulator. On subsequent shift cycles the D0L position of the accumulator enters the adder at DXL time, merges with a special digit one and stores back in the D0L accumulator position. In this way the shift number is advanced by one at the beginning of each shift cycle. A carry from the adder at D0L time stops the shifting.

Right shift and half correct (code 31) is accomplished in much the same manner as right shift except that a shift-count of "9" is detected at D0L time of the last shift cycle. This indication sets up circuits which allow the D1 position to shift into D0 position on the last shift cycle and which cause an extra cycle to be taken. This extra cycle is a half correction cycle during which "5" is added to the contents of D0L and any carry is reflected up through the other accumulator positions during the rest of the cycle. The operation ends at the end of the half correction cycle. As in right shift the units position of the address register controls the number of shifts taken except that a shift number of zero ("D" address 0000) causes 10 shifts to be taken, with the result that the half corrected contents of the upper accumulator are left in the lower accumulator.

Left shifting (code 35) is accomplished by allowing the accumulator D1L through D10L and D1U through D10U positions to enter the adder from the on time outputs, merge with zeros and the resulting adder outputs to store back in the accumulator, one digit time later. Thus D1L on time output is available from the adder at D2L time for storage in the accumulator D2L position. Each accumulator digit successively follows this procedure for the two accumulator word intervals with the result that the accumulator number is shifted one position to the left. As with right shift, the number of positions shifted is determined by the shift number in the units position of the address register. The shift count is kept in the D0L accumulator position as in right shift.

A left shift and count operation is accomplished much the same as left shift except that the appearance of a significant digit in the accumulator D10U position is detected to stop the operation. The shift number in the address register units position is usually zero, so the shift count in the accumulator D0L position, when a significant digit is detected, is an indication of the number of shifts which were used. If a D0L carry is detected before a significant digit appears in D10U, it is an indication that the maximum allowable number of shifts have been taken. The overflow circuits will be set up and a shift count of 10 left in the accumulator D2L and D1L positions.

When a shift number of zero accompanies this operation, the maximum allowable number of shifts is ten. If it is desired, for any reason, to limit the number of allowable shifts to less than ten, the number of allowable shifts desired can be used as a shift number in the units position of the accompanying data address. In this way the number of shifts needed to cause a D0L carry is reduced. If, in this case, a significant digit is sensed before a D0L carry occurs, the shift count number left in the D0L accumulator position is the tens complement of the unused allowable shifts.

The shift signals are developed by the shift control latch. These signals from program are D10U to DXU for codes 31 and 36 and DXL for codes 30 and 35.

The shift operations will be considered as starting upon receipt of one of these signals.

Right Shift—Code 30.

Signals from Program: Right Shift (EDXL).

Key Units: Right shift latch; on, EDXL right shift signal. Off, next D1L after D0L carry. D0L carry latch; on, D0L carry. Off, NWPL.

Circuit operation:

(1) Right shift signal (DXL)—

(*a*) Mixes at mix **1052** (Fig. 85*f*) to give DXL shift number gate for entry of address register units position to adder, in complement form, on the first right shift cycle. Shift number gate also inverted at inverter **1053** (Fig. 85*c*) to give negative shift number gate which blocks development of accumulator true add gate and true-complement add gate for DXL of the first right shift cycle. The blocking is accomplished at switch **1054** (Fig. 85*d*).

(*b*) Mixes at mix **1055** (Fig. 85*a*) to turn on right shift double latch **968**.

(2) Right shift double latch—

(*a*) Mixes at mix **1056** (Fig. 85*c*) to turn arithmetic operation latch on and hold it on through the operation.

(*b*) Mixes at mix **1057** (Fig. 85*c*) to turn accumulator regeneration off and hold it off during each WPL for the entire R.S. operation.

(*c*) Switches at switch **1058** (Fig. 85*h*) with DXL and D0L to give an accumulator R.I. gate for DXL and D0L on each R.S. cycle. (Negative D0L on half correction is continually up on a R.S. operation.) This allows the adder output to enter the accumulator DXL and D0L positions on each R.S. cycle.

(*d*) Switches with DXL at switch **1059** (Fig. 85*e*) to operate the adder output-zero insert circuit for DXL time of each R.S. cycle. This supplies a zero for entry into the accumulator DXL position on each R.S. cycle. This zero is later read out of DXL at each D10U time and shifted into D10U position on each R.S. cycle, thus filling the place vacated by shifting D10U to D9U.

(*e*) Mixes at mix **1061** (Fig. 85*i*) to develop accumulator true add gate and true-complement add gate for DXL on each R.S. cycle except the first. This allows the shift count number, in the accumulator D0L position, to enter the adder from the accumulator early output at DXL time, on each R.S. cycle, except the first.

(*f*) Mixes at mix **1038** (Fig. 86*g*) to operate the special digit add "1" circuit for DXL time of each R.S. cycle. This supplies a "1" to merge with the complement shift number from the address register units position on the first cycle, to form a tens complement shift count for storage in accumulator D0L position. It also supplies the "1" to merge with the shift count from the accumulator D0L position at DXL time on subsequent R.S. cycles, to advance the shift count.

(*g*) Switches at switch **969** (Fig. 85*f*) with negative DXL and negative D0L to give a right shift gate which is up for D1L through D10U of each R.S. cycle. This allows the accumulator D2L through DXL early latch outputs to control the pedestal lines for D1L through D10U times, resulting in a shift to the right of each accumulator position. Note that the output of the D1L position, which is read out at D0L time, is not used. Thus the contents of D1L are lost on each R.S. cycle.

(*h*) Mixes at mix **1062** (Fig. 86*h*) to lower the distributor blanking gate for the entire R.S. operation. This prevents any distributor early outputs or substituted zeros from entering the adder during a R.S. operation.

(3) With these gates developed, shifting continues until an adder carry at D0L time indicates that the shift count has reached "10." This carry occurs at D0L time at the beginning of the lower word interval following the last shift cycle.

(4) D0L carry (A–C of D0L), from adder D0L carry detection circuit—(*a*) Switches at switch **1063** (Fig. 85*g*) to turn D0L carry latch **1064** on (negative shift count and half correct signal and half correct add 5 are up during R.S. operations). This latch stays on through the two words following the last shift cycle, until the next NWPL.

(5) D0L carry latch—(*a*) Switches at switch **1065** (Fig. 85*g*) with D1L to turn right shift latch **968** (Fig. 85*a*) off.

(6) Right shift latch off—

(*a*) Removes all accumulator and adder R.S. entry controls.

(*b*) Allows arithmetic operation latch to turn off with next NEDX and provide end of operation signal.

(*c*) Capacitor coupled cathode follower pulse at cathode **1066** as right shift latch goes off, switches with D1 and WL (EvW) at switch **1067** (Fig. 85*c*) to turn accumulator regeneration latch back on at D1L. This is necessary since the normal time for turning the latch on (WPL) has passed. Accumulator regeneration takes over starting with the D1L position, which now contains the units position of the shifted number, and continuing until another accumulator operation is programmed. A zero has been placed in the DXL position by zero insertion and in the D0L position because of accumulator R.I. of the shift number advanced to zero. These zeros will be regenerated as the accumulator regeneration process continues.

(7) Note that the no carry insert-carry blank circuit is active for each DXL and D0L time during a right shift operation.

Right shift and half correct—Code 31.

Signals from Program: Right shift and half correct (D10U through D10L).

Key units: Half correct latch; on D10U, off D0L CP of half correction cycle. Right shift latch; on DXL, off DXL at start of half correction cycle. Half correct add "5" latch, on when D0L "9" is sensed in adder output, off when accumulator true add goes off. Accumulator true add latch; on DXL at start of half correction cycle, off next NWPL. Timing chart of Fig. 70.

Circuit operation:

(1) Right shift and half correct signal (D10U through D10L)—

(*a*) Switches with DXL at switch **1068** (Fig. 85*f*) to develop DXL shift number gate and negative shift number gate. This allows units position of address register to enter adder in complement form, on first cycle. Blocks development of accumulator true add gate for DXL of first cycle.

(*b*) Mixes at mix **1069** (Fig. 85*f*) to develop D10U through D10L, negative shift count and half correct signal. This signal, at switch **1063** (Fig. 85*g*) blocks a possible shift stop due to turning on the D0L carry latch on the first cycle of a right shift and half correct operation. A carry from D0L can occur on the first cycle when the shift number is zero. This negative shift right and half correct signal is also sent to the adder and used to block a set-up of the overflow circuits on a D0L carry on the first cycle of a shift count operation.

(*c*) Mixes at mix **1071** (Fig. 85*a*) to turn on the half correct latch **1072.**

(*d*) Switches with DXL at switch **1073** (Fig. 85*f*) to turn on the right shift latch.

(2) Right shift latch—(*a*) Develops the same gates as for right shift (code 30) to cause right shifting of the contents of the accumulator.

(3) Shifting continues as in a right shift operation until a "9" is sensed at the adder output at D0L time of the last shift cycle. This "9" is sensed at switch **1074** (Fig. 85*f*) by the coincidence of D0L, adder output B5 and Q4 and half correct latch on.

(4) Half correct D0L "9" from cathode follower **1075** (Fig. 85*f*)—

(*a*) Mixes at mix **1076** (Fig. 85*f*) to extend the right shift gate from cathode folowers **971** to include D0L for this last shift cycle. This allows the contents of the D1L accumulator position to enter the D0L position on this last shift cycle in preparation for half correction on the next cycle.

(*b*) Mixes at mix **1077** (Fig. 85*a*) to turn on the half correct add "5" latch at D0L time of the last shift cycle.

(*c*) Is inverted at inverter **1078** (Fig. 85*f*) to give a negative half correct D0L "9" at cathode follower **1079.**

(5) Negative half correct D0L "9" from cathode follower **1079**—

(*a*) Blocks turning off of the half correct latch at D0L CP time of the last shift cycle, until D0L CP of the half correct cycle at switch **1081** (Fig. 85*a*).

(*b*) Blanks the D0L part of the accumulator R.I. gate during D0L time of the last shift cycle, so that the contents of the accumulator D1L position can enter the D0L position at this time, via the right shift path as in (4*a*) above, without conflict with the "9" coming from the adder output.

(6) Shifting continues through the last shift cycle. The D0L accumulator position now contains the digit which is to be half adjusted and the half correct add "5" latch is on.

(7) Half correct add "5" latch—

(*a*) Removal of off output blocks a possible turning on of the D0L carry latch at D0L time of the half correct cycle, in the event that the half correction causes a carry.

(*b*) Removal of off output also prevents the usual true add, D0L no-carry insert-carry blank action during D0L time of the half correct cycle. This allows the D0L carry or no-carry status of the adder latches to be determined by the carry or no-carry result of the half correction. This result will be used during the adder's analysis of the D1L accumulator position.

(*c*) On output switches with half correct latch output and DXL at switch **1082** (Fig. 85*m*) to turn on the accumulator true add latch for the half correct cycle. The true add latch output provides an accumulator true add gate, an accumulator true-complement add gate and an accumulator R.I. gate for the half correct cycle. This allows the accumulator early outputs for D0L through DXL positions, to enter the adder on the half correct cycle, and the adder outputs for D0L through D10U to enter the accumulator.

The accumulator true add latch output also switches with half correct add "5" output and DXL and D0L timing gates at switch **1084** (Fig. 85*e*) to insert zeros in place of the adder output for entry into the accumulator DXL and D0L positions on the half correct cycle, thus leaving zeros in these accumulator positions at the end of the operation.

The accumulator true add latch output also holds the arithmetic operation latch off for the half correct cycle.

The accumulator true add latch output also mixes at mix **959** to hold accumulator regeneration off for the half correct cycle.

(*d*) On output switches with DXL at switch **1085** (Fig. 86*h*) to give special digit add "5" for DXL of half correct cycle. This "5" merges with the digit to be corrected, from the accumulator D0L early output at DXL time and sets up a carry or no-carry condition in the adder for use in the analysis of the next digit.

(*e*) On output switches with D0L CP and the lack of a negative half correct D0L "9," at switch **1081** (Fig. 85*a*), to turn off the half correct latch.

(*f*) On output switches with NDXL at switch **1086** (Fig. 86*g*) to operate special digit add zeros circuit for each digit interval during the half correct cycle except DXL. This provides zeros to merge with the contents of the accumulator D1L through D10U positions and allows any carry which may occur during half correction to be reflected up through the other accumulator positions.

(*g*) On output switches with accumulator true add latch output and DXL and D0L to give zero insert as in (*c*) above.

(*h*) On output switches with DXL at switch **1087** (Fig. 85*f*) to turn off the right shift latch at the start of the half correct cycle.

(*i*) On output mixes at mix **1024** (Fig. 86h) to extend the distributor blanking gate for the half correct cycle, after the right shift latch turns off.

(8) The accumulator true add latch turns off with the NWPL at the beginning of the next word after the half correct cycle.

As this latch goes off, a capacitor coupled cathode follower **1088** pulse from the first anode of the latch **957** (Fig. 85*m*) mixes at mix **1089** (Fig. 85*a*) to turn the half correct add "5" latch off.

(9) As these latches go off, all right shift and half correct control gates are removed and accumulator regeneration is restored. The arithmetic operation latch turns off and the end of operation signal is developed allowing the commutator to advance on the next program step.

Left shift—Code 35.

Signals from Program: Left shift (EDXL).

Key Units: Left shift latch, on EDXL left shift signal, off next D1L after D0L carry; D0L carry latch, on D0L carry, off NWPL.

Circuit operation:

(1) Left shift signal (DXL)—

(*a*) Mixes at mix **1052** (Fig. 85*f*) to give DXL shift number gate and negative shift number gate to allow complement adder entry of shift number from units position of address register, at DXL of first left shift cycle and to block development of accumulator true add at this time.

(*b*) Mixes at mix **1091** (Fig. 85*b*) to turn on left shift latch **1031**.

(2) Left shift latch—

(*a*) Mixes at mix **1056** (Fig. 85*c*) to turn arithmetic operation latch on and hold it on through the operation.

(*b*) Mixes at mix **1057** to turn accumulator regeneration off and hold it off during each WPL for the entire left shift operation.

(*c*) Mixes at mix **1048** (Fig. 85*h*) to provide an accumulator R.I. gate for the entire left shift operation. This allows the adder output to enter the accumulator all during the operation.

(*d*) Switches with DXL and D1L at switch **1092** (Fig. 85*j*) to operate adder output zero insert circuit for DXL and D1L times of each left shift cycle. This supplies a zero to enter the accumulator DXL position and also the D1L position to fill the vacancy left by the left shift action. (Significant digit latch stays off on a left shift operation.)

(*e*) Mixes at mix **1061** (Fig. 85i) to develop accumulator true add gate and accumulator true or complement add gate for DXL of each left shift cycle except the first. This allows the shift count number, in the accumulator D0L position, to enter the adder from the accumulator early output at DXL time, on each left shift cycle except the first.

(*f*) Mixes at mix **1038** (Fig. 86*g*) to operate the special digit add "1" circuit for DXL time of each left shift cycle. This supplies a "1" to merge with the complement shift number from the address register units position on the first cycle, to form a tens complement shift count for storage in the accumulator D0L position. It also supplies the "1" to merge with the shift count from the D0L position at DXL time, on subsequent left shift cycles, to advance the shift count.

(*g*) Mixes at mix **1024** (Fig. 86*h*) to provide a distributor blanking gate for the entire left shift operation.

(*h*) Mixes at mix **1093** (Fig. 85*b*) to give an "any left shift gate" which then further switches at switch **1094** (Fig. 85*g*) with negative DXL and negative D0L to give a left shift gate which is up for D1L through D10L and D1U through D10U. (Significant digit latch remains off all during a left shift operation.) This left shift gate allows the D1L through D10L and D1U through D10U "on time" accumulator outputs to enter the adder during each left shift cycle, where they will be merged with zeros and stored back in the accumulator one digit later and thus one digit position to the left.

The left shift gate also mixes at mix **1024** (Fig. 86*h*) to provide a distributor blanking gate. This duplicates the action of the left shift latch on a left shift operation and that of the M–D left shift latch on multiply and divide operations.

(*i*) Mixes at mix **1095** (Fig. 85*i*) to give an "any left shift gate" which: (1) Switches with D0L at switch **1096** to give accumulator true add gate for D0L of each left shift cycle. (Significant digit latch off for all left shift operations.) This allows the contents of accumulator D1L position to enter the adder at D0L time and merge with zero. This adder output is not used because of the zero insert of (*d*), but the entry keeps the adder no-carry circuit set up for the next digit analysis and eliminates the need for a D1L no-carry insert. (2) Switches at switch **1097** (Fig. 86*g*) with an inverted special digit add "1" to operate the special digit add zero circuit. This supplies zeros to the adder B side for each digit interval of each left shift cycle except DXL, when a "1" is being entered as in (*f*). (3) Is sent to the adder to give left shift "on" control for adder reset.

(*j*) Mixes at mix **1098** (Fig. 85*d*) to give an inverted any left shift gate for left shift "off" control of adder reset.

(3) Note that a no carry insert-carry blank is provided for each DXL and D0L interval during this operation.

(4) With these gates developed, shifting continues with each accumulator position being shifted one position to the left and the shift count being advanced by one on each shift cycle, until an adder carry occurs at D0L time indicating that the shift count has reached "10."

(5) Adder D0L carry (A.-C. pulse) switches at switch **1063** (Fig. 85*g*) to turn on the D0L carry latch at the beginning of the two-word interval following the last shift cycle.

This latch stays on through this two-word interval, until the next NWPL.

(6) D0L carry latch—(*a*) Switches with next digit (D1L) at switch **1065** to give a carry stop signal at the output of cathode follower **1099** which switches at switch **1101** (Fig. 85*b*) with significant digit off, to turn off the left shift latch.

(7) When left shift latch goes off at D1L time, all left shift control gates are removed. Arithmetic operation latch will turn off with the next NEDX and develop the end of operation signal to turn off operation interlock and allow the commutator to advance.

Also, as the left shift latch goes off, a capacitor coupled cathode follower pulse at cathode follower **1102** (Fig. 85*b*), from the left anode of the latch double inverter, switches at switch **1067** (Fig. 85*c*) with D1 and WL to turn accumulator regeneration on. This allows regeneration to start with D1L.

Left shift and count—Code 36.

Signals from program: Left shift and count signal (D10U–DXU).

Key units: Shift count latch; on D10U, CP, if accumulator high order position is zero. Off, when significant digit latch comes on. No shift count latch; on, D10U, CP, if accumulator high order position is non-zero. Off, NWPU. Left shift latch; on, DXL on a shift and count operation if the shift count latch is turned on. Off, next D3L after significant digit latch comes on. Significant digit latch; on, D10U, CP, when D10U adder output is non-zero and shift count latch is on, indicating that a high order significant digit has been detected. Off, next D6. Timing chart of Fig. 80.

Circuit operation:

(1) Left shift and count signal (D10U through D10L)—

(*a*) Switches with D10U, CP, at switch **1103** (Fig. 85*b*) to develop a shift count-no shift count test pulse at cathode follower **1104**.

This test pulse switches with accumulator zero at switch **1105** or accumulator non-zero at switch **1106** to turn on either the shift count latch **1107** or the zero shift count latch **1108**. The accumulator zero and non-zero indications are from cathode followers **1109** and **1111** (Fig. 83*a*) which in turn are from the switching and mixing of the accumulator pedestal output as shown in Fig. 83*a*.

If the accumulator D10U position is non-zero and the zero shift count latch is turned on, a capacitor coupled cathode follower pulse at cathode follower **1112** (Fig. 85*b*) from the right anode of the latch double inverter, parallels the normal arithmetic restart circuit to supply the restart signal. A capacitor coupled cathode follower pulse from cathode follower **1113** from the left anode when the latch goes off at WPU time, parallels the end of operation signal to turn the operation interlock latch off.

The zero shift count latch output is taken to switch **503** (Fig. 64*d*) to reset the D1 and D2 positions of the accumulator to zeros, indicating a zero shift count.

(*b*) Switches at switch **1114** (Fig. 85*f*) with DXL and the shift count latch output (if accumulator high order position is zero) to turn on the left shift latch **1031** (Fig. 85*b*) and to develop the shift number gate and negative shift number gate from cathode followers **1115** and **1053** (Figs. 85*f* and 85*c*). These gates allow entry of the complement shift number from the address register units position and block development of an accumulator true add gate, for DXL of the first shift cycle.

(*c*) Mixes at mix **1069** (Fig. 85*f*) and inverted at inverter **1116** to give a negative shift count and half correct signal (D10U through D10L). This signal blocks turning on of the D0L carry latch at switch **1063** (Fig. 85*g*) and thus prevents the carry stop action of a possible D0L carry on the first cycle, as the tens complement shift number is developed in the adder.

(2) Shift count latch—

(*a*) Gives shift count gate which is taken to adder via switch **1117** and cathode follower **1118** (Fig. 85*g*) and used as a control on the overflow detection circuit.

(*b*) Switches with DXL and shift count signal to turn on left shift latch and develop shift number gate as in (1*b*) above.

(*c*) Mixes at mix **1056** (Fig. 85*c*) to turn on arithmetic operation latch. This action parallels a similar action by the left shift latch.

(*d*) Switches at switch **1119** (Fig. 85*g*) to allow the significant digit latch to be turned on when a D10U, non-zero condition is detected.

(3) Left shift latch—(*a*) The same gates are developed to cause left shifting and advancing of the shift count, as described in paragraph 2 of left shift circuit operation.

(4) Shifting continues until, either a non-zero is detected from the adder output at D10U time, or a D0L carry is detected when the tens complement shift count number reaches "10," before a significant digit is detected at D10U time.

(5) When an adder D10U non-zero is detected, shifting must be stopped and the tens complement shift count number left in the D2L and D1L accumulator positions.

(*a*) Adder non-zero indication switches with D10U, CP and shift count latch on output at switch **1119** to turn on the significant digit latch which stays on until the next D6.

(*b*) The shift count number enters the adder and merges with a special digit add "1" at DXL time as in the preceding cycles. The adder output is not used at D0L time, however, since zero insert is energized for D0L when significant digit latch is on. A D0L carry could occur (if shift count has reached 9) but must not be allowed to set up an overflow since a significant digit has been detected.

(*c*) Significant digit latch: (1) Removal of off output blocks D0 at switch **1121** thus extending the left shift gate to include D0L for the word following the last shift cycle; blocks accumulator true add gate for D0L at switch **1096** (Fig. 85*i*); blocks zero insert for D1L time at switch **1122** (Fig. 85*j*) and blocks D1L carry stop signal from turning left shift latch off at switch **1101** (Fig. 85*b*) in the event of a D0L carry after the detection of a significant digit. (2) On output switches at switch **1123** (Fig. 85*j*) with D0L to give zero insert for D0L time; switches with D0L at switch **1124** to operate special digit add "1" for D0L time.

These gates, developed under control of the significant digit latch, allow the shift count number to enter the adder a second time, this time from the accumulator D0L "on time" output via the left shift data flow path, merge with "1" and store in the accumulator D1L position, meanwhile inserting a zero in the D0L accumulator position. Assuming that the shift number was zero to start, this leaves a true shift count in the D1L position and no-carry set-up in adder.

(*d*) The zero in the accumulator D1L position (previously inserted) enters the adder from the accumulator "on time" output at D1L time via the left shift path, merges with special digit add zero and adder no-carry as on preceding cycles. The adder output (zero) stores in accumulator D2L position.

(*e*) The significant digit latch on output also: (1) Mixes at mix **1125** (Fig. 85*b*) to turn the shift count latch off. This removes one of the conditions necessary for setting up an overflow, if a D0L carry occurs after a significant digit is detected as in (*b*) above. (2) Switches with the next D3L at switch **1126** to turn off the left shift latch which will end the operation and allow the commutator to advance.

(*f*) A capacitor coupled cathode follower pulse when the left shift latch goes off from cathode follower **1102** (Fig. 85*b*) switches with D3L at switch **1067** (Fig. 85*c*) to turn accumulator regeneration on and start regeneration with D3L.

(6) When a D0L carry is detected before a D10U significant digit appears, the overflow circuit must be set up; the shift count number in accumulator D0L position (now advanced to 9) must be advanced by "1" and the result ("10") stored in the accumulator D2L and D1L positions; shifting must be stopped, the operation ended and regeneration resumed at D3L time.

(7) Last shift cycle just completed, with a shift count of 9 in D0L position. No significant digit was detected at D10U.

(*a*) At DXL the shift count number (9) enters the adder and merges with special digit "1" as on preceding cycles. The adder sum is zero and carry.

(*b*) The shift count D0L carry (shift overflow) mixes at mix **1127** (Fig. 85*b*) to turn on the significant digit latch and sets up the adder overflow circuit. The significant digit latch stays on until the next D6L.

A D0L carry is developed in the adder and switched at switch **1063** (Fig. 85*g*) to turn the D0L carry latch on. However, the D1L carry stop signal developed at switch **1065** and cathode follower **1099** is prevented from turning the left shift latch off at switch **1101** (Fig. 85*b*) by the fact that the significant digit latch has turned on.

(*c*) Significant digit latch: (1) Removal of off output blocks D0 at switch **1121** (Fig. 85*g*), thus enabling the left shift gate to include D0L; blocks accumulator true add gate for D0L at switch **1096** (Fig. 85*i*); blocks zero insert for D1L time at switch **1112** (Fig. 85*j*) and blocks D1L carry stop signal from turning left shift latch off at switch **1101** (Fig. 85*b*). (2) On output switches at switch **1123** (Fig. 85*j*) with D0L to give zero insert for D0L time; switches with D0L at switch **1124** to operate special digit add "1" for D0L time.

These gates, developed under control of the significant digit latch, allow the shift count number ("9") in the D0L accumulator position, to enter the adder a second time, this time from the accumulator D0L "on time" output via the left shift path, merge with "1" and store in accumulator D1L position. This leaves zero in D1L and carry set-up in adder.

(*d*) The zero in accumulator D1L position enters the adder from accumulator "on time" output at D1L time via the left shift path, merges with zero and adder carry. Adder "1" sum stores back in accumulator D2L position. This leaves the shift count "10" in the two low order accumulator positions.

(*e*) The significant digit latch on output also: (1) Mixes at mix **1125** (Fig. 85*b*) to turn the shift count latch off, but not before an overflow has been set up.

(2) Switches with the next D3L at switch **1126** to turn off the left shift latch which will end the operation and allow the commutator to advance.

(*f*) A capacitor coupled cathode follower pulse from cathode follower **1102** when the left shift latch goes off, switches at switch **1067** (Fig. 85*c*) to turn accumulator regeneration on and start regeneration with D3L.

Circuit operation for multiplication

Multiplication is accomplished by over and over addition of the multiplicand. The number of additions in each of ten shifts is controlled by the value of a multiplier digit. The multiplier must be in the upper accumulator as a result of a previous operation. The multiplicand is placed in the distributor from the data address location by the enable R.I-start R.I. sequence of the control commutator, which is a part of each 10's and 60's operation. From the distributor the multiplicand is repeatedly added to the lower accumulator as specified by the multiplier digit, to develop the partial product. This process is repeated for ten shifts. On each shift cycle a new multiplier digit is moved into position to control the number of add cycles to be taken and the partial product is shifted up toward the high end of the accumulator. At the end of the operation the product is in the accumulator (lower and upper) and the multiplier has been shifted out of the upper accumulator and lost.

Each multiply operation consists of a series of ten M–D left shift cycles. Each M–D left shift cycle is followed by from zero to nine add cycles depending upon the value of the controlling multiplier digit.

On each M–D left shift cycle the high order multiplier digit is shifted from the D10U into the DXL accumulator position and the shift count number in the D0L accumulator position is advanced by "1." Because the D0L accumulator position contains a zero at the start of a multiply operation, a D0L carry will not occur until the tenth M–D left shift cycle. As the high order multiplier digit enters the DXL accumulator position on each cycle during multiplication it is checked for a zero or non-zero condition. If it is zero, another M–D left shift cycle is taken. If non-zero, an add cycle is set up, during which the multiplicand is added to the lower accumulator and a "9" is added to the controlling multiplier digit in the DXL position, thus effectively reducing it by "1." Adding cycles continue until a DXL zero indicates that the controlling multiplier digit has been reduced to zero and the required number of add cycles taken, *in that shift*. Add cycles are stopped and another M–D left shift cycle set up. This process is repeated until a D0L carry on the tenth M–D left shift cycle indicates that the last shift is in process. A DXL zero on the last add cycle of this last (10th) shift stops the multiply operation.

Figs. 117*a*, 117*b* and 117*c* show the timing chart for a multiply operation.

Multiply—Code 19.

Signals from Program: Multiply (DXL and D0L), No reset ("D" control) not used, No absolute ("D" control) not used.

Key units:

Multiply latch. On, DXL multiply signal. Off, next DXL adder zero, after end M–D latch comes on.

M–D left shift latch. On, DXL multiply signal or DXL, adder zero, when end M–D latch is off. Off, NWPL when no DXL adder zero occurs to shunt the NWPL reset.

Accumulator true add latch. On, DXL, adder non-zero, when multiplying. Off, NWPL when DXL adder output is zero.

Lower control latch. On, DXL, adder non-zero, when multiplying. Off, NWPL when DXL adder output is zero.

D0L carry latch. On, D0L carry. Off, NWPL.

End M–D latch. On, when D0L carry latch goes on, on multiply or divide operations. Off, next ED0L after multiply latch turns off.

General circuit action: The first cycle of a multiply operation is a M–D left shift cycle. A M–D left shift cycle, controlled by the multiply latch and the M–D left shift latch, results in shifting the multiplier one position to the left in the accumulator which places the high order digit in the DXL position. Any partial product which may be in the lower accumulator is, of course, also shifted left one position, since any shift involves the whole accumulator. Also, on a M–D left shift cycle the shift count number in the accumulator D0L position is advanced by "1." This number is zero at the start of the operation, so it becomes "1" during the first part of the first multiply cycle.

At DXL time of each multiply cycle the adder output is tested for zero or non-zero. When this test is made at DXL of the cycle following a M–D left shift cycle and a non-zero condition is detected the accumulator true add latch and the lower control latch are turned on. If a zero is detected the M–D left shift latch is held on for another M–D left shift cycle. If the accumulator true add and lower control latches are turned on (and with distributor true control off), gates are developed which allow the distributor contents to be added to the lower accumulator on a true and cycle, just as in add and subtract operations. On a true add cycle, during multiplication a "9" is added to the multiplier high order digit (in the DXL position) as it is processed by the adder at D10U time. This effectively reduces its value by "1." The reduced multiplier digit is stored back in the DXL position. As this reduced multiplier digit comes from the adder at DXL time its zero or non-zero condition is tested. Non-zero holds the accumulator true add and lower control latches on for another add cycle. Zero allows them to turn off, but turns M–D left shift back on for another M–D left shift cycle, to shift the multiplier and partial product one position to the left and advance the shift count number.

This shifting and adding continues until a D0L carry is detected as the shift count is advanced to ten on the last (10th) M–D left shift cycle. This D0L carry turns on the end M–D latch. Another series of add cycles occurs, until DXL is reduced to zero. This adder DXL zero, the first to be detected after the end M–D latch is turned on, turns off the multiply latch to end the operation instead of turning on the M–D left shift latch.

The sign of the product, developed by analyzing the sign of the accumulator (multiplier) and the sign of the distributor (multiplicand) is placed in the accumulator sign latches during the first part of the first M–D left shift cycle. It is entered under control of the multiply signal which only occurs once at the beginning of the operation.

Circuit operation:

(1) Multiply signal, DXL and D0L—

(*a*) Mixes at mix **1128** (Fig. 85*g*) to turn on the multiply latch **1129**.

(*b*) Mixes at mix **1131** (Fig. 85*k*) to turn on the M–D left shift latch **1032**.

(*c*) Mixes at mix **1332** (Fig. 85*q*) to give an M–D signal at the cathode follower **1133** which: (1) Switches at switch **1134** (Fig. 85*t*) with DXL, A–C gate to reset the remainder sign latches. (2) Switches at switch **1135** with DXL, C–A gate to give a test gate, which in turn switches at switches **1136** and **1137** (Fig. 58*o*) with an accumulator plus or an accumulator minus indication from the accumulator sign latches, to turn on either the remainder plus or remainder minus latch (**951** Fig. 85*o* or **952** Fig. 85*t*). This transfers the multiplier sign to the remainder sign latches. (3) Switches with DXL at switch **1138** (Fig. 85*j*) to give zero insert for first cycle. (4) Switches at switch **1139** (Fig. 85*t*) with D0, A–C gate to reset the accumulator sign latches, after the multiplier sign has been transferred to the remainder sign latches. (5) Switches at switches **1141** and **1142** (Fig. 85*r*) with either a distributor plus or a distributor minus sign and a D0L timing gate to develop either a multiplicand-divisor plus signal or a multiplicand-divisor minus signal.

These signals switch at switches **1143** and **1144** (Fig. 85*n*) and **1145** and **1146** (Figs. 85*s*) with the multiplier sign from the remainder plus or remainder minus latch to turn on either the accumulator plus or accumulator minus sign latch. This places the sign of the product, which is to be developed, in the accumulator sign latches.

(2) Multiply latch—

(*a*) Removal of off output lowers the M–D blanking gate by action of switch **1147** (Fig. 85*k*). This gate from cathode follower **1148** prevents the end M–D latch **1149** from being turned off until after the mutliply latch goes off.

(*b*) Removal of off output blocks switch **1014** (Fig. 85*t*) and prevents any D1L reset of accumulator sign latches.

(*c*) On output mixes at mix **1151** (Fig. 85*h*) to provide an accumulator R.I. gate for the multiply operation.

(*d*) On output mixes at mix **1152** (Fig. 85*b*) to turn accumulator regeneration off and hold it off for the multiply operation.

(*e*) On output mixes at mix **2027** (Fig. 85*c*) to turn arithmetic operation latch on.

(*f*) On output switches at switch **1153** (Fig. 86*h*) with D10U and accumulator true add gate to operate special digit add "9" circuit. This allows "9" to be added to high order multiplier digit on add cycles, when multiplying.

(*g*) On output mixes at mix **1154** (Fig. 85*k*) to give M–D gate. This allows the output of the D0L carry latch to turn on the end M–D latch, on multiplication and division operations.

(*h*) On output switches at switch **1155** with adder DXL zero and end M–D off, to allow adder DXL zero to turn on M–D left shift latch on multiply operations **1032**.

(*i*) On output switches at switch **1156** (Fig. 85*i*) with adder non-zero and DXL, BP to provide an adder DXL non-zero pulse which mixes at mix **1157** (Fig. 85*m*) to turn on the accumulator true add latch and at mix **929** (Fig. 86*a*) to turn on the lower control latch. This allows the detection of an adder DXL non-zero to set up an add lower condition.

(3) M–D left shift latch—

(*a*) Mixes at mix **1061** (Fig. 85*i*) to give accumulator true add gate for DXL to allow the shift count from accumulator D0L position to enter the adder one digit early on each M–D left shift cycle.

(*b*) Mixes at mix **1038** (Fig. 86*g*) to give special digit add "1" for DXL of each M–D left shift cycle. This advances the shift count number by "1" on each M–D left shift cycle. Advanced shift count stores back in D0L accumulator position.

(*c*) Mixes at mix **1095** (Fig. 85*i*) to develop a left shift or M–D left shift gate which: (1) Switches at switch **1096** with D0L and significant digit latch off to give accumulator true add gate for D0L of M–D left shift cycle. This allows the accumulator D1L position to enter the adder and merge with a zero to keep the adder no-carry circuits set up for the next digit. Adder output not used. (2) Switches at switch **1097** (Fig. 86*g*) to "add zero" when not adding "1" on any left shift operation. (3) Controls adder reset action on left shift.

(*d*) Also mixes at mix **1098** (Fig. 85*d*) and inverted at inverter **1158** to give left shift off control of adder.

(*e*) Mixes at mix **1093** (Fig. 85*b*) to develop left shift gate for D1L through D10L and D1U through D10U of each M–D left shift cycle. This allows the accumulator on time outputs to enter the adder, merge with zeros and store back in accumulator one position later, as in any left storage operation.

(*f*) Switches at switch **1159** (Fig. 85*j*) with D1L to give zero insert at D1L time on a M–D left shift cycle. This provides a zero to enter the accumulator D1L position on each left shift to fill the vacancy left when the contents of D1L shift to D2L.

(*g*) Mixes at mix **1024** (Fig. 86*h*) and is inverted at inverter **1161** to give distributor blanking gate for each M–D left shift cycle.

(4) Accumulator true add latch—

(*a*) Cannot operate overflow circuit during a multiply operation because of M–D blanking gate to adder.

(*b*) Mixes at mix **1162** (Fig. 85*c*) to turn on arithmetic operation latch (parallels multiply latch action).

(*c*) Mixes at mix **1042** (Fig. 85*h*) for accumulator R.I. gate (parallels multiply latch action).

(*d*) Mixes at mix **959** to turn accumulator regeneration off (parallels multiply latch action).

(*e*) Mixes at mix **1003** (Fig. 85*d*) to give accumulator true add gate for DXLB through D10U and accumulator true-complement add gate. These gates allow entry to the adder of the accumulator early outputs on add cycles.

(*f*) Accumulator true add gate switches at switch **1153** (Fig. 86*h*) with multiply latch on and D10U to operate special digit add "9" for D10U time of a multiply add cycle. This "9" merges with the accumulator DXL early output and results in a reduced high order multiplier digit value, to store back in the accumulator DXL position.

(5) Lower control latch—

(*a*) Switches with WU(OdW) at switches **1163** and **1164** (Fig. 86*a*) to develop distributor zero control and entry gates for DXU through D10U. These gates are also developed for every D10 and DX. This supplies zeros, in place of distributor early outputs, to merge with upper accumulator early outputs on multiply add cycles.

(*b*) Mixes at mix **931** and switches at switch **936** to develop U–L check gate for adder entry B operation.

(6) D0L carry latch—

(*a*) On with D0L carry by action of switch **1063** (Fig. 85*g*).

(*b*) On output switches at switch **1165** (Fig. 85*k*) with M–D gate to turn on end M–D latch.

(7) End M–D latch—

(*a*) Removal of off output at switch **1155** blocks any further turning on of the M–D left shift latch on this multiply operation.

(*b*) On output conditions switch **1166** (Fig. 85*q*) so that the next adder DXL zero will turn off the multiply latch.

(*c*) On output mixes at mix **1057** (Fig. 85*c*) to hold accumulator regeneration off and at mix **1048** (Fig. 85*h*) to extend accumulator R.I. for DXL of the cycle following the last add cycle. This allows the zero adder output to enter accumulator DXL position.

(*d*) End M–D turns off with next ED0L by action of switch **1167** (Fig. 85*k*) after multiply latch goes off and restores M–D blanking gate.

(*e*) A capacitor coupled cathode follower pulse from cathode follower **1168** as end M–D latch goes off, mixes at mix **1169** (Fig. 85*c*) to turn accumulator regeneration on and start regeneration with D0L.

(8) Removal of Multiply latch on output lowers multiply control gates and allows arithmetic operation latch to turn off. This develops end of operation signal to turn off Op. interlock and allow commutator to advance.

Circuit operation for division

Division is accomplished by repeatedly subtracting the divisor from the high order digits of the dividend, until a sign change indicates that it has been subtracted once too often. Each time a subtraction is made, without causing a sign change, "1" is added to the quotient. When a sign change occurs the divisor is added once to the reduced dividend, to correct for the overdraw. The entire reduce dividend and partial quotient are then shifted left one position and the process repeated, for ten shifts in all.

The dividend must be in the accumulator as the result of a previous operation. The divisor is placed in the distributor from the data address location by the enable R.I.-start R.I. sequence of the control commutator, which is a part of each 10's and 60's operation. From the distributor, the divisor is repeatedly subtracted from the upper accumulator until the sign change occurs and is then added once to the upper accumulator. A quotient digit entry of "1" is made in the lower accumulator D1 position each time that a subtraction does not result in a sign change. Thus at the end of a divide operation the quotient is in the lower accumulator, the remainder is in the upper accumulator, the dividend has been destroyed and the divisor is still in the distributor.

The following chart, illustrates the concept of division, using an abbreviated accumulator with 8 positions for simplicity. The problem 1440÷12=120 is used. The first cycle of a divide operation is a M-D left shift cycle. This clears the D1L accumulator position for possible quotient digit entries during the first shift. In the case of the 8 position accumulator shown, only four shifts are possible. With a 20 digit accumulator ten shifts for each division would be used. The number of shifts is counted, as in other shifting operations and the count retained in the accumulator D0L position. This shift count action is not shown in the chart.

	Upper				Lower			
	D4	D3	D2	D1	D4	D3	D2	D1
Dividend	0	0	0	0	1	4	4	0
Shift	0	0	0	1	4	4	0	0
Subtract Divide (Upper)	−0	0	1	2	0	0	0	0
Sign Change	9	9	8	9	4	4	0	0
Correct	+0	0	1	2	0	0	0	0
	0	0	0	1	4	4	0	0
Shift	0	0	1	4	4	0	0	0
Subtract Divide (Upper)	−0	0	1	2	0	0	0	0
No Sign Change	0	0	0	2	4	0	0	0
Subtr. Div. (Upper) Enter "1" (Lwr)	−0	0	1	2	0	0	0	1
Sign Change	9	9	9	0	4	0	0	1
Correct	+0	0	1	2	0	0	0	0
	0	0	0	2	4	0	0	1
Shift	0	0	2	4	0	0	1	0
Subtract Divide (Upper)	−0	0	1	2	0	0	0	0
No Sign Change	0	0	1	2	0	0	1	0
Subtr. Div. (Upper) Enter "1" (Lwr)	−0	0	1	2	0	0	0	1
No Sign Change	0	0	0	0	0	0	1	1
Subtr. Div. (Upper) Enter "1" (Lwr)	−0	0	1	2	0	0	0	1
Sign Change	9	9	8	8	0	0	1	2
Correct	+0	0	1	2	0	0	0	0
	0	0	0	0	0	0	1	2
Last Shift	0	0	0	0	0	1	2	0
Subtract Divide (Upper)	−0	0	1	2	0	0	0	0
Sign Change	9	9	8	8	0	1	2	0
Correct	+0	0	1	2	0	0	0	0
Stop	0	0	0	0	0	1	2	0

On any division the number of digits in the quotient is determined by the number of digits in the dividend and divisor. In the present machine the maximum number of quotient digits which can be developed is ten and is governed by the capacity of the accumulator and distributor. Thus on any divide operation, the highest significant dividend digit must be positioned in the accumulator, taking into account the number of digits in the divisor, so that no more than nine subtractions will be required on the first shift to produce a sign change. Considering the number of dividend digits to be the number of accumulator digit positions up to and including the position of the highest significant dividend digit, and remembering the initial left shift on a divide operation, the rule for positioning the dividend in the accumulator is as follows:

The total number of digits in the dividend must not exceed the number of digits in the divisor, plus 9; unless the absolute value of the divisor digits is greater than the absolute value of the corresponding number of high order dividend digits; in which case the total number of dividend digits can equal the number of divisor digits plus 10.

When the dividend is not properly positioned and more than 9 subtractions would be needed to cause a sign change, more than 9 quotient digit "1" entries will be entered at D1L. A carry from D1L detects this condition and sets up an overflow stop.

Each divide operation consists of a series of ten M-D left shift cycles. Each M-D left shift cycle is followed by the number of subtract cycles necessary to cause a sign change and an add cycle to correct for the overdraw. On a M-D left shift cycle the high order dividend digit is shifted left from the D10U accumulator position into the DXL accumulator position. From here it enters the adder at D10U time on a divide subtract cycle, from the accumulator early output. The distributor contents are subtracted from the upper accumulator by the usual complement add entry. A zero is supplied by the distributor zero entry circuit as a DXL distributor early output, to merge with the high order dividend digit. This zero enters as a complement on a subtract (reduction) cycle and in true form on an add (correction) cycle. When a complement number is added to a true number a carry from the high order position indicates that the result is still a true number and the sign has not changed. No carry indicates that the result is complement and the sign has changed.

This principle is illustrated by the following simplified accumulator chart, using the previous example of 1440÷12=120.

	Upper					Lower			
	DXL	D4	D3	D2	D1	D4	D3	D2	D1
Dividend	0	0	0	0	0	1	4	4	0
Shift	0	0	0	0	1	4	4	0	0
Subtract Divide	9	9	9	8	7	9	9	9	9^1
No Carry DXL, Sign Change	9	9	9	8	9	4	4	0	0
Correct	0	0	0	1	2	0	0	0	0
	0	0	0	0	1	4	4	0	0
Shift	0	0	0	1	4	4	0	0	0
Subtract Divide (Upper)	9	9	9	8	7	9	9	9	9^1
Carry DXL, No Sign Change	0	0	0	0	2	4	0	0	0
Sub. Div. (U) Enter "1" (Lwr)	9	9	9	8	7	9	9	9^1	1
No Carry DXL, Sign Change	9	9	9	9	0	4	0	0	1
Correct	0	0	0	1	2	0	0	0	0
	0	0	0	0	2	4	0	0	1
Shift	0	0	0	2	4	0	0	1	0
Subtract Divide (Upper)	9	9	9	8	7	9	9	9	9^1
Carry DXL, No Sign Change	0	0	0	1	2	0	0	1	0
Sub. Div. (U) Enter "1" (Lwr)	9	9	9	8	7	9	9	9^1	1
Carry DXL, No Sign Change	0	0	0	0	0	0	0	1	1
Sub. Div. (U) Enter "1" (Lwr)	9	9	9	8	7	9	9	9^1	1
No Carry DXL, Sign Change	9	9	9	8	8	0	0	1	2
Correct	0	0	0	1	2	0	0	0	0
	0	0	0	0	0	0	0	1	2
Shift	0	0	0	0	0	0	1	2	0
Subtract Divide (Upper)	9	9	9	8	7	9	9	9	9^1
No Carry DXL, Sign Change	9	9	9	8	8	0	1	2	0
Correct	0	0	0	1	2	0	0	0	0
Stop	0	0	0	0	0	0	1	2	0

This DXL carry or no-carry on a divide subtract cycle is used to determine whether the following cycle will be add or subtract. When an add (correction) cycle is finally set up, its completion signals another M–D left shift cycle.

On each M–D left shift cycle the shift count number in the D0L accumulator position is advanced by "1." This number is zero at the start of the divide operation, so it becomes "0" and a D0L carry occurs on the tenth shift. After this D0L carry occurs, the completion of the next add (correction) cycle stops the operation instead of signalling a M–D left shift.

Division can be programmed to retain the remainder (code 14) or to reset the remainder (code 64). In the latter case zeros are inserted during the upper word time of the last correction cycle. When the remainder is retained its sign, held in the remainder sign latches, is read out in place of the accumulator sign at D0U time, as explained under accumulator sign read-out.

The sign of the quotient, developed by analzying the sign of the accumulator (dividend) and the sign of the distributor (divisor) is placed in the accumulator sign latches during the first part of the first M–D left shift cycle.

Divide—Code 14 or 64.

Signals from Program: Divide (DXL and D0L), No reset or reset ("D" control), No absolute ("D" control) not used.

Key units:

Divide latch. On DXL divide signal. Off with divide left shift signal when end M–D latch is on.

M–D left shift latch. On DXL divide signal or DXL divide left shift signal at end of an add (correction) cycle, when end M–D latch is off. Off NWPL.

Upper control latch. On DXL when divide latch comes on. Off next NWPL after divide latch goes off.

Distributor complement add latch. On when M–D left shift goes off or DXL carry when quotient digit latch is on. (These are both divide subtract set-up signals.) Off NWPL, if no DXL carry occurs to shunt the NWPL.

Distributor true control latch. On with divide subtract signal. Off NWPL, if no divide substract signal.

Accumulator true add latch. On with divide subtract signal. Off next NWPL after distributor complement add latch goes off. (This is at the end of a correction cycle.)

Quotient digit latch. On next D10U after complement add distributor goes on, when dividing. (This is at the end of a reduction cycle.) Off next D6. While on, it controls the quotient digit "1" entry and the development of a DXL test pulse which tests for a DXL carry or no-carry.

D0L carry latch. On D0L carry. Off next NWPL.

End M–D latch. On when D0L carry latch goes on, on multiply or divide. Off next ED0L after divide latch goes off.

General circuit action: The first cycle of a divide operation is a M–D left shift cycle. On a M–D left shift cycle the dividend is shifted one position to the left, placing the high order position in the DXL accumulator position. Any partial quotient which may be in the lower accumulator is also shifted left one position. Also, on a M–D left shift cycle the shift count number in the accumulator D0L position is advanced by "1." This number is zero at the start of the divide operation, so it is advanced to "1" on the first M–D left shift cycle.

When the M–D left shift latch turns off at the beginning of the next cycle a capacitor coupled cathode follower pulse switches with the divide latch output to develop a divide subtract signal which turns on the distributor complement add latch, the distributor true control latch and the accumulator true add latch. With these latches on, and with the upper control latch on (turned on by the divide latch) the necessary conditions for a subtract upper cycle are set up. Distributor zero entry is active for the lower word time and for all D10 and DX intervals. The divisor (in the distributor) is subtracted from the upper accumulator. On this cycle the high order dividend digit in the DXL accumulator position, enters the adder at D10U time, merges with a complement zero from the distributor zero entry circuit and comes from the adder at DXL time to enter the DXL accumulator position. It is the carry or no-carry pulse signal from the adder, during this DXL time, which is tested to determine whether or not the sign of the reduced dividend has changed. The quotient digit latch is turned on at D10U time of each reduction cycle by testing with a D10U timing gate to discover whether or not the complement distributor add latch is on. The quotient digit latch output switches with a DXL timing gate to develop a divide test signal, which switches with a DXL carry to test for a sign change. If a carry is present, a divide subtract signal is developed, which shunts the NWPL reset of the distributor complement add, distributor true control and accumulator true add latches, thus holding them on for another reduction cycle. If the complement distributor add latch stays on, the quotient digit latch output switches with complement distributor add and D0L to operate special digit add "1" to enter a quotient digit of "1" during this next reduction cycle.

The quotient digit latch output also controls the carry insert circuit so that a carry is inserted at D1L on all divide complement add cycles except the first one after a shift, instead of at D0L as on other complement add cycles. This inserts the necessary "elusive one" into the D2L position when dividing, on all cycles when a quotient entry is made in the D1L position; but allows its insertion into the D1L position on the first reduction cycle after a shift, when no quotient entry is made.

This is illustrated in the preceding simplified chart of division.

Reduction cycles continue until no DXL carry occurs. No divide subtract signal is developed to shunt the NWPL reset of the distributor complement add latch and the distributor true control latch. These latches turn off, and remove the gates necessary for subtracting the distributor. As the distributor complement add latch goes off, on a divide operation, a capacitor coupled cathode follower pulse holds the accumulator true add latch on. With accumulator true add and distributor true control on, the necessary gates are developed to true add the distributor to the upper accumulator, thus providing the correction cycle.

At the end of the correction cycle the accumulator true add latch turns off. A capacitor coupled pulse, when it goes off, switches with divide latch on and quotient digit latch off to develop a divide left shift signal which turns the M–D left shift latch on, if the end M–D latch is still off.

This process of shifting, reducing and correcting and the consequent entry of quotient digits continues until the shift count number in the accumulator D0L position is advanced to 10, as indicated by a D0L carry. This carry turns on the M–D left shift latch. With this latch on the divide left shift signal at the end of the last correction cycle, is prevented from turning on the M–D left shift latch and instead turns off the divide latch to stop the operation.

Circuit operation:

(1) Divide signal DXL and D0L—

(*a*) Mixes at mix **1132** (Fig. 85*q*) to give an M–D signal which: (1) Switches at switch **1134** (Fig. 85*t*) with DXL, A–C gate to reset the remainder sign latches. (2) Switches at switch **1135** with DXL, C–A gate to give a test gate which in turn switches at switches **1136** and

1137 (Fig. 85*o*) with an accumulator plus or accumulator minus indication from the accumulator sign latches, to turn on either the remainder plus or remainder minus sign latch. This transfers the dividend sign to the remainder sign latches and establishes the sign of the remainder. (3) Switches with DXL at switch **1138** (Fig. 85*j*) to give zero insert for the first cycle. (4) Switches at switch **1139** (Fig. 85*t*) with D0, A–C gate to reset the accumulator sign latches, after the dividend sign has been transferred to the remainder sign latches. (5) Switches at switches **1141** and **1142** (Fig. 85*r*) with either a distributor plus or a distributor minus sign and a D0L timing gate, to develop either a multiplicand-divisor plus signal or a multiplicand-divisor minus signal. These signals switch at switches **1143** and **1144** (Fig. 85*n*) and **1145** and **1146** (Fig. 85*s*) with the dividend sign from the remainder plus or remainder minus latch, to turn on either the accumulator plus or accumulator minus latch. This establishes the sign of the quotient, in the accumulator sign latches.

(*b*) Switches at switch **1171** (Fig. 85*l*) with no reset signal to turn remainder control latch to the remainder side.

The remainder control latch will have been turned to the no remainder side by coincidence of a reset signal and D10U if a divide reset (code 64) was programmed.

The remainder control latch on or off outputs switch at switches **1172** and **1173** with either no reset or reset to develop the remainder check gate needed to turn the divide latch on. This is a check to insure that the remainder control latch is properly set in accord with the programmed reset or no reset before proceeding with the division. The remainder control latch outputs are also switched and mixed as shown at Fig. 85*o* to control the accumulator sign R.O.

(*c*) Switches at switch **1174** (Fig. 85*p*) with remainder check gate to turn on the divide latch **1176**.

(*d*) Mixes at mix **1175** (Fig. 85*k*) to turn on the M–D left shift latch.

(2) Divide latch—

(*a*) Removal of off output: (1) Lowers M–D blank at output of cathode follower **1148** to block D10U overflow circuit; prevents turning off end M–D latch until division latch goes off, by action of switch **1177**; blocks switch **1016** (Fig. 85*s*) to prevent accumulator sign reset on divide complement add cycles. (2) Blocks switches **1011** (Fig. 85*s*) and **1014** (Fig. 85*t*) to prevent reset of accumulator sign latches on divide complement add cycles.

(*b*) On output: (1) Mixes at mix **1048** (Fig. 85*h*) to give accmulator R.I. gate for divide. (2) Mixes at mix **1057** (Fig. 85*c*) to turn accumulator regeneration off for division. (3) Mixes at mix **928** (Fig. 86*a*) to turn upper control latch on for division. (4) Mixes at mix **1027** (Fig. 85*c*) to turn arithmetic operation latch on for division. (5) Switches at switch **1178** (Fig. 85*k*) with end M–D left shift pulse to give divide subtract signal. (6) Switches at switch **1179** (Fig. 85*p*) with quotient digit off and accumulator true add off pulse at end of correction cycle to give divide left shift signal. (7) Switches at switch **1181** (Fig. 85*m*) with distributor complement add off pulse, to hold accumulator true add latch on for correction cycle. (8) Switches at switch **1182** (Fig. 85*p*) with distributor complement add on and D10U, to turn on quotient digit latch **1183**. (9) Switches at switch **937** (Fig. 85*e*) as one condition necessary to zero insert for the upper word portion of the last correction cycle, to reset the remainder, on divide with reset.

(3) M–D left shift latch—

(*a*) Develops the same left shift control gates listed under M–D left shift latch, on multiply.

(*b*) Provides distributor complement add gate used by coupled cathode follower pulse at cathode follower **1184** (Fig. 85*k*) provides the end left shift signal which switches at switch **1178** with divide latch on, to give a divide subtract signal.

(4) Distributor complement add latch—

(*a*) Turns on with a divide subtract signal through mix **1185** (Fig. 85*m*).

(*b*) Provides distributor complement add gate used by adder entry B for a complement distributor entry.

(*c*) Mixes at mix **1007** to give complement accumulator or distributor gate. This gate blocks no-carry insert at D0L time and gives carry insert at D0L when quotient digit latch is off or D1L when quotient digit latch is on.

(*d*) Switches at switch **1182** (Fig. 85*p*) with divide latch on and D10U to turn quotient digit latch on at D10U time of each reduction cycle.

(*e*) Switches at switch **1186** (Fig. 86*g*) with DXL to give special digit add zero at DXL of each distributor complement add cycle. This supplies a true zero at DXL and blanks the complement zero from the distributor zero entry circuit. This true zero is needed to merge in the adder with the shift count number so that it will not be altered on divide subtract cycles.

(*f*) Switches at switch **1187** (Fig. 86*g*) with quotient digit latch on and D0L to give special digit add "1" for quotient entry. This "1" merges with contents of accumulator D1L from early output and stores back in D1L position.

(*g*) As the distributor complement add latch turns off, at the end of the last reduction cycle of the shift, a capacitor coupled cathode follower pulse at cathode follower **1188** (Fig. 85*m*) switches at switch **1181** with divide latch on, to hold accumulator true add latch on for the correction cycle.

(5) Distributor true control latch—

(*a*) Turns on with divide subtract signal.

(*b*) Removal of off output removes distributor true add gate from adder entry B switching and blocks any possible upper word zero insert at switch **937** (Fig. 85*e*) until last correction cycle on divide with reset.

(6) Accumulator true add latch—

(*a*) Cannot operate overflow circuit during a divide operation because of M–D blanking gate.

(*b*) Mixes at mix **1162** (Fig. 85*c*) to turn on arithmetic operation latch (parallels divide latch action).

(*c*) Mixes at mix **1042** (Fig. 85*h*) for accumulator R.I. gate (parallels divide latch action).

(*d*) Mixes at mix **959** (Fig. 85*m*) to turn accumulator regeneration latch off (parallels divide latch action).

(*e*) Mixes at mix **1003** (Fig. 85*d*) to give accumulator true add gate and accumulator true-complement add gate for DXL through D10U. These gates allow the accumulator early outputs to enter the adder on reduction and correction cycles.

(*f*) Switches at switch **937** (Fig. 85*e*) as one condition necessary to give zero insert during upper word of last correction cycle, on divide with reset.

(7) Quotient digit latch—

(*a*) On D10U when distributor complement add latch is on by action of switch **1182** (Fig. 85*p*).

(*b*) Removal of off output: (1) Blocks switch **1179** (Fig. 85*p*) to prevent development of divide left shift signal except when quotient digit latch is off. (2) Blocks switches **1189** and **942** (Fig. 85*e*) to prevent D0L carry insert on complement add cycle when quotient digit latch is on. (All reduction cycles except the first of any shift.)

(*c*) On output: (1) Switches at switches **943** and **944** (Fig. 85*e*) to give carry insert at D1L on all reduction cycles except the first of a shift. (2) Gives quotient digit gate to control overflow circuit. (3) Switches at switch **1187** (Fig. 86*g*) with D0L and distributor complement add on, to give special digit add "1" for quotient entry to the accumulator D1L position. (4) Switches at switch **1191** (Fig. 85*p*) with DXL and divide latch on to develop a divide test signal which tests for the presence of a DXL carry on reduction cycles.

(8) D0L carry latch—

(*a*) On with D0L carry, when shift count is advanced to 10 on an M–D left shift cycle, by action of switch **1063** (Fig. 85*g*).

(*b*) On output switches at switch **1165** (Fig. 85*k*) with M–D gate, to turn end M–D latch on.

(9) End M–D latch—

(*a*) Removal of off output blocks switch **1192** (Fig. 85*k*) to prevent divide left shift signal from turning on the M–D left shift latch.

(*b*) On output conditions switch **1193** (Fig. 85*p*) to allow divide left shift signal, at end of last correction cycle, to turn off divide latch.

(*c*) On output mixes at mix **1057** (Fig. 85*c*) to hold accumulator regeneration off and at mix **1048** (Fig. 85*h*) to extend accumulator R.I. for DXL of the cycle following the last correction cycle. This allows the DXL adder output to enter the accumulator DXL position on the last correction cycle.

(*d*) End M–D latch turns off, with next ED0L by action of switch **1177** (Fig. 85*k*) after the divide latch goes off and restores the M–D blanking gate.

(*e*) A capacitor coupled cathode follower pulse at cathode follower **1168** when end M–D latch goes off, mixes at mix **1169** (Fig. 85*c*) to turn accumulator regeneration on and start regeneration with D0L.

(10) Removal of divide latch on output lowers divide control gates and allows arithmetic operation latch to turn off. This develops end of operation signal to turn off operation interlock latch and allow commutator to advance.

(11) If remainder control latch is on no remainder, due to use of code 64, the remainder is cleared from the upper accumulator during the upper word interval of the last correction cycle. This is done by the switch **937** (Fig. 85*e*). A coincidence of divide latch on, end M–D on, distributor true on, accumulator true add on, remainder control latch on and an upper word timing gate operates the zero insert circuit to enter zeros in the upper accumulator during the last correction cycle.

Circuit operation for table look-up

The table look-up operation is provided to allow information to be obtained which has been pre-stored in the form of a table (Fig. 120).

A table consists of a series of arguments (reference facts) arranged in a sequence of ascending absolute values and an associated series of functions (result values). The stored arguments are searched until one is found which compares with, or is next higher than, the searching argument. The stored function value, associated with the selected stored argument, is the desired result.

The arguments of a table are stored in ascending sequence. The first argument of a table is in the first word location of a band (addresses ending in —00 or —50). Each argument after the first is placed in successive locations, except that the last two word locations of a band cannot be used for the storage of arguments (address ending in —48, —49, —98, —99).

The associated function values are usually stored in an area of the drum other than that containing the arguments. When this is done, each function value is a fixed number of storage locations away from its associated argument, so that once the address location of the argument is known, the address location of the function value can be determined by the addition of a constant to the address of the argument.

The function value associated with each argument may be stored in the same storage location as the argument, provided the total number of positions does not exceed ten.

A table look-up operation is accomplished by a code 84 on one program step. The known argument (searching argument) must have been placed in the distributor by a preceding operation. The table search starts at the location specified by the data address which accompanies the 84 code. This address, except under special conditions, will be one ending in —00 or —50, the first word of a "table" band.

During the table look-up operation the searching argument in the distributor is successively compared with each of the stored arguments. When the equal argument, or next higher when no equal is present is located, the address of this selected argument is placed in the data address positions (8–5) of the lower accumulator.

On the next program step a stored constant can be added to the lower accumulator to alter the "D" address to that of the location of the associated function value. At the same time the appropriate Op. code and "I" address can be added to the other lower accumulator positions. The result is an instruction word, located in the lower accumulator, whose "D" address is the location of the desired function value. This word can be used by using an 8002 "I" address or can be stored in general storage for later use.

On a TLU operation the contents of the word locations of the address selected "table" band are successively compared with the contents of the distributor. This is accomplished by allowing the general storage outputs to enter the adder via entry A and the "on time" distributor outputs to enter the adder, in 10's complement form, via entry B. An adder carry indicates that an equal or next higher argument has been found. If no adder carry is detected, the selecting address in the data address positions (5–8) of the program register and in the address register must be increased by 50. This is accomplished by running the contents of the program register through the adder during the 49th word interval of the band, adding a 5 to its D6 position and storing the result back in the program register. The modified D5 through D8 positions also enter the address register from the program register on time lines, in the normal manner. The new general storage selection is available from the address register during D5 through D8 of the 49th word interval. The 50th word interval allows time for the general storage selection circuits to recover, before being used.

The search continues as before until an adder carry is detected. When a carry is sensed, a number equal to the drum word interval during which the carry was sensed, is added to the modified address in the program register. (This address in the program register is the address of the first word of the band.) Thus the general storage address of the equal or next higher argument is found. This addition is also done on a program register add cycle, during the word interval following the one on which the carry is detected. The value of the number which is added is determined by the sector and word timing interval during which the carry is detected.

After the desired address is in the program register an additional cycle is taken, during which the program register contents enter the adder, merge with zeroes and the adder output for D5 through D8 allowed to enter the lower accumulator.

Table look-up—Code 84.
Signal from Program: TLU ("D" control).
Key units:

TLU control latch **916** (Fig. 86*c*). On, TLU signal, D0, S4, W9. Off when TLU carry latch comes on. Sets up TLU operation. TLU program add latch **1037** (Fig. 86*b*). On, DX and TLU band change signal (S4, W8), or DX and TLU carry latch on, or DX and coincidence of program to accumulator latch on and lower control latch on. Off next NWP. Develops gates which allow program register early outputs to enter adder and adder outputs to control program register pedestals. Also control no-carry insert on program add. TLU program regeneration control latch **1194** (Fig. 86*b*). Off with same conditions which turn TLU program add latch on. On with next

WP. When off, interrupts program register regeneration by blocking the path between program on time latch outputs and pedestal lines.

TLU carry latch **918** (Fig. 86*d*). On, DX, A-C gate and adder carry. Off next NWP. Controls addition of proper number to program register D5 and D6 position, depending on which word time it is turned on.

Program to accumulator control latch **1195** (Fig. 86*d*). On when TLU carry latch goes off at end of address adjustment cycle. Off next NWPU. When on, causes entry of the program register contents to adder A during a lower word interval; the entry of special digit zeros to adder B to merge with the program register values and the development of a distributor blanking gate; the entry of the D5 through D8 adder outputs into the corresponding lower accumulator positions and the entry of all adder outputs back into the program register.

Lower control latch **927** (Fig. 86*a*). On, WP (DX) when program to accumulator control latch goes on. Off next NWPL after program to accumulator control latch goes off. (Note: This latch action not needed for proper logical operation of the circuit.)

Accumulator regeneration latch **956** (Fig. 85*c*). On next D5L after program to accumulator control latch comes on. Off next D9. Interrupts accumulator regeneration for D5 through D8 of the next lower word after program to accumulator control comes on. Timing chart of Fig. 63.

Circuit action: Any table search starts with W0 of a band. Program control sends a signal when an Op. code of 84 is present, which lasts from the time "D" control appears until S4, W9, when the TLU control latch turns on and turns single interlock **857** (Fig. 81*l*) on. This signal turns on the TLU control latch which provides the necessary adder A and B entry gates and carry insert controls to allow the general storage outputs to enter the adder and merge with distributor complement "on time" outputs.

If an adder DX carry is not detected by S4, W8 time, a TLU band change signal is developed. This signal resets the address register, develops an address register read-in gate for D5 through D8 of the next word interval, operates add zeros and add 5 circuits, turns on the program add latch and turns off the TLU program regeneration control latch. This action develops those adder control gates, address register controls and program regeneration controls necessary to enter the early outputs of the program register into the adder, add 5 to the D6 position, store the result back in the program register and enter the modified data address in the address register. This is done during W8 of S4. W9 time is allowed for the general storage head selection circuits to settle, after being activated by the new address register output. During this S4, W8 time the gates which allow general storage and distributor outputs to enter the adder, under control of the TLU control latch, are blocked. At the end of S4, W9 time these gates are restored and the newly selected general storage band outputs compared with the contents of the distributor.

This process continues until an adder DX carry is detected. A carry at any digit time will be reflected up through the successive digit intervals and is detected at DX time to turn the TLU carry latch on. In this way the TLU carry latch is made to turn on at the beginning of a word interval on any carry condition. The TLU carry latch stays on for one word, until the next NWP.

At this point it is necessary to set up another program add cycle, during which a number, equal to the timing word interval during which the carry was found, is added to the D5 and D6 positions of the address in the register. This addition will occur during the next word interval after the carry was detected and the value of the number to be added will be one less than the word time during which the addition takes place. For example, an addition occurring during word 20 time (S2, W0) was signalled by a carry detected during word 19 time (S1, W9) and must result in the addition of 19 to the program register.

The TLU carry latch output energizes coincident circuits which energize the proper special digit circuits to accomplish this selection of the correct number to be added. At this point the address of the selected argument is in the D5 through D8 positions of the program register.

When the TLU carry latch goes off, a capacitor coupled cathode follower pulse turns the program to accumulator control latch on. This latch may come on at the start of either an upper or lower word. Once on, it stays on until the next WPU. While on it turns on the program add latch and holds it on through a lower word, if necessary; turns off the TLU program regeneration control latch and holds it off through a lower word, if necessary; operates the add zeros and distributor blanking circuits; turns accumulator regeneration off and develops accumulator R.I. for D5 through D8 of the lower word interval during which it is on. These actions develop those adder, program register, and accumulator control gates necessary to enter the program register contents into the adder, merge it with zeros, store the result back in the program register and enter D5 through D8 adder outputs into the D5 through D8 positions of the lower accumulator. Note that if the program to accumulator control latch comes on at the start of an upper word, the program register contents will be passed through the adder twice, once during the upper word time and again during the following lower word interval. The entry to the accumulator can be made only during the lower word interval.

When the program to accumulator control latch goes off at NWPU, a capacitor coupled cathode follower pulse is the restart signal which advances the control commutator to start an "I" half cycle.

Summary of table look-up

Table arguments are stored on the drum in ascending sequence by their absolute values (argument signs are ignored). Therefore, it will be necessary to treat some tables containing arguments with different signs as two different tables. In some cases, only one set of arguments would be stored and two different locations referred to according to the sign of the search argument.

Arguments are stored 48 to a band except for the last band of a table which may contain less. The last two memory locations in each band (0048, 0049, 0098, 0099, etc.) cannot be used to store table arguments. They may be used to store functions, instructions, etc., however. Table arguments should be stored in successive drum locations starting with the first word in a band (0000, 0050, 0100, etc.).

A table does not need to contain all of the possible search arguments which will be encountered. A table look-up operation will cause a search until an equal or next higher condition occurs. Thus, when a search argument is not actually in the table, the search will stop on the next higher table argument. Interpolation can be accomplished by programming, if necessary.

The fact that the search stops on a next higher as well as an equal condition makes possible, in many cases, the location of both the table argument and the associated function in the same word. The argument must be stored in the most significant positions of the word.

Function values also may be stored a fixed number of locations from their arguments. Thus, having found the location N of the argument, the function is located at

N+C where C is the fixed separation of the functions from the arguments.

The "D" address of the table look-up instruction is the address of the location of the first argument in the table. Should the "D" address be other than the first address of a band, the search will begin at the first address of the band and add the address of the equal (or next higher) argument to the "D" address. This may be useful for short tables having less than one band of arguments as it saves modification to locate the function address. For example: the instruction 84 0020 xxxx is given for a 20 argument table whose arguments are located 0000–0019. The argument at 0015 equals the given argument xx 0035 xxxx will appear in the lower accumulator. If the function is stored 20 locations from the argument, the lower accumulator contains the address of the function with no further modification.

It is possible to store several short tables in one band and use the same "D" address. For example: assume 3 tables 15 arguments in length each having 5 digit arguments: Searching will begin at 0000. Proper placement of the argument will cause the search to be effective only in the applicable section of the band. When this method is used, it is necessary for the first argument of each table to be greater in value than the last argument of the previous table (. . . 9999, 10000 . . ., 99999, 100000 . . .).

The known argument must be placed in the distributor prior to the search operation. Throughout the table look-up operation this argument is compared against the table arguments stored on the drum. When an equal argument or next higher (if no equal exists) is located, the address is placed in positions 8–5 of the lower half of the accumulator. As mentioned above, the absolute value (sign ignored) of the distributor argument is compared with the absolute value of the table arguments. When the search is complete, the known argument will be unaltered in the distributor. Positions 10–9 and 4–1 of the lower half of the accumulator will also be unaltered.

It will be necessary to modify the argument address in positions 8–5 of the lower accumulator in order to locate the function except for the two cases mentioned:

(1) Argument and function stored in the same word.

(2) Address of function compiled by giving "D" address other than the first in the band.

It is possible to achieve the results of a table look-up operation without actually using the table look-up code (84). This type of operation is feasible if the arguments to be used are four digit numbers that fall in a block between the limits of 0000 and 1999 or 0000 and 0999 (or can be modified in some manner to meet these conditions).

One application that might lend itself to this method of programming is cost distribution by department number. If it is assumed that department numbers run from 0001 to 0200, it is then possible to assign magnetic drum location 0001 to department 0001, location 0002 to department 00002, etc. By using department number which had been read into the machine from a card, as a data address, it is possible to locate the associated magnetic drum storage location without doing a table look-up.

This is one method of doing table look-up without using code 84. Modifications of this basic approach are possible but are dependent on the type of information processed.

Code 84 performs an automatic table look-up using the "D" address as the location of the first table argument and the "I" address as the address of the next instruction to be executed. The argument for which a search is to be made must be in the distributor. The address of the table argument equal to or higher (if no equal exists) to the argument given is placed in positions 8–5 of the lower accumulator. The search argument remains, unaltered, in the distributor.

Clearing drum

The following routine can be used to clear effectively the entire drum to minus zeros. Minus zeros are used to differentiate between a location that was never loaded and a location that might be cleared to plus zero by a store instruction in the program.

Location of Instruction	Instruction			Operation Abbrev.
	OP	Data	Instr.	
8000	70	0004	xxxx	RD
0004	61	0008	0007	RSU
0007	69	0006	0005	LD
0005	24	0000	8003	STD
8003	10	0001	8003	SU

This routine will clear all drum locations to zeros with the exceptions of location 0000 which will contain the constant 01 0000 8000 and location 0001 which will contain the constant 00 0001 0000.

If a stop instruction (operation 01) is stored in location 0000, the failure to load all the instructions of the program will cause the machine to stop. The stop will occur as soon as the first missing instruction is referred to once the program has been started.

In any memory location where an instruction is missing, there will be zeros. When this location is referred to, the zeros in positions 10 and 9 become a no-op code (00) and the machine will go immediately to the "I" address of the instruction (positions 4–1). Since the "I" address portion is also zeros, the address of the next instruction is 0000. Location 0000 contains a stop code and the machine will stop at that point if the programmed switch is set to "Stop."

While there have been shown and described and pointed out the fundamental novel features of the invention as applied to a preferred embodiment, it will be understood that various omissions and substitutions and changes in the form and details of the device illustrated and in its operation may be made by those skilled in the art, without departing from the spirit of the invention. It is the intention, therefore, to be limited only as indicated by the scope of the following claims.

What is claimed is:

1. In a data processing machine: table look-up mechanism comprising program storage means adapted to have manifestations of program data stored therein, said manifestations of program data including the manifestations of an address of an argument of a table to be searched; second storage means adapted to have manifestations of tabular data comprising the arguments of the table to be searched stored therein; third storage means adapted to have manifestations of data comprising the argument to be located stored therein; merging means having first and second data manifestation inputs and a data manifestation output and adapted to merge manifestations of data from said first input with manifestations of data from said second input and manifest merged data at said output; first data manifestation transfer means under control of said program storage means adapted to be conditioned in accordance with manifestations of program data stored in said program storage means for successively communicating manifestations of data comprising successive arguments succeeding the argument located at the address included in said manifestations of program data from said second storage means to said first input; second data manifestation transfer means for conveying the manifestations of data from said third storage means to said second input simultaneously with each communication of manifestations of data from said second storage means to said first input; third data manifestation transfer means adapted to transfer manifestations of data from said program storage means to said first input; data manifesting means for manifesting values equal to the difference between the value of the address included

in said manifestations of program data and the values of the addresses of the arguments communicated from said second storage means to said first input; fourth data manifestation transfer means for transferring data from said data manifesting means to said second input; and means under control of said output for disabling said first and said second data transfer means and enabling said third and said fourth data transfer means in response to a predetermined signal manifested at said output so that the address included in said manifestations of program data is merged with the value manifested by said data manifesting means.

2. In a data processing machine, first data manifestation storage means, second data manifestation storage means, third data manifestation storage means, data manifestation merging means having first and second inputs and an output, data manifesting means, a first data manifestation conveying link under control of said first data manifestation storage means adapted to be conditioned in accordance with manifestations of data stored in said first data manifestation storage means and adapted to convey manifestations of data sequentially from said second storage means to said first input, a second data manifestation conveying link adapted to convey manifestations of data repeatedly from said third data manifestation storage means to said second input, means for enabling said first and said second data manifestation conveying links, a third data manifestation conveying link adapted to convey manifestations of data from said first data manifestation storage means to said first input, a fourth data manifestation conveying link adapted to convey manifestations of data from said data manifesting means to said second input, and switching means under control of said output for disabling said first and said second data manifestation conveying links and for enabling said third and fourth data manifestation conveying links in response to a predetermined output condition.

3. In a data processing machine which is adapted for performing a table look-up operation under the control of stored manifestations of program values, said manifestations of program values each comprising an address portion, the combination of a rotating magnetic drum on which is defined a plurality of address positions arranged in tracks about said drum, each track having a home position; means for recording manifestations of a plurality of arguments of tabular data in consecutive ascending order in chosen address positions beginning with the home position of one of said tracks, program manifestation storage means for storing manifestations of a program value containing the address of the home position of one of said tracks having manifestations of tabular data stored therein, address selecting means under control of said program manifestation storage means adapted to be conditioned in accordance with the address portion of manifestations of a program value stored in said program manifestation storage means for sequentially selecting the address position on said drum corresponding to said address portion of said program value manifestation and said chosen succeeding address positions so that said manifestations of arguments are sequentially selected in ascending order, means for manifesting a value equal to the number of manifestations of arguments sequentially selected, auxiliary storage means for storing manifestations of an argument for which the corresponding address on said drum is sought, an adder for concurrently receiving and adding manifestations of two values, a first data manifestation conveying link adapted to sequentially convey the selected manifestations of arguments from said drum to said adder, a second data manifestation conveying link adapted to repeatedly convey manifestations of the complement of the argument from said auxiliary storage means to said adder concurrently with the conveyance of manifestations of said selected arguments, a third data manifestation conveying link connected between said program manifestation storage means and said adder, a fourth data manifestation conveying link connected between said manifesting means and said adder, and means responsive to a carry from the highest ordered position of said adder to disable said first and said second data manifestation conveying links and to enable said third and said fourth data manifestation conveying links respectively to concurrently convey manifestations of the address portion of said program value and the value manifested by said manifesting means to said adder so that manifestations of the two values are added to produce manifestations of the address sought.

4. Apparatus according to claim 3, characterized further in that a data manifestation transfer link is provided between said adder and said magnetic drum so that manifestations of an address produced by said adder may be recorded on said drum for subsequent use.

5. Apparatus according to claim 3, in combination with an accumulator for storing manifestations of a program value, an output for said adder, a data manifestation transfer link for transferring address manifestations from said output to replace the address portion of the manifestations of said program value stored in said accumulator and means for transferring data manifestations from said accumulator to said magnetic drum so that manifestations of said modified program value may be subsequently used.

6. Apparatus according to claim 5, in combination with a data manifestation transfer link for transferring manifestations of program values from said magnetic drum to said program manifestation storage means.

7. In a data processing machine which is adapted for performing table look-up operations under the control of stored manifestations of program words, said program words each comprising an address portion and an operation code portion, the combination of a rotating magnetic drum on which is defined a plurality of address positions arranged in tracks about said drum, each track having a home position, means for recording a plurality of manifestations of arguments of tabular data in consecutive ascending order in chosen address positions beginning with the home position of one of said tracks and for recording manifestations of program words in other address positions, an accumulator, means for entering manifestations of a program word from said drum into said accumulator, program manifestation storage means adapted to store manifestations of a program word, means for transferring manifestations of a program word containing the address of the home position of one of said tracks having manifestations of tabular data stored therein and containing a table look-up operation code from said drum to said program manifestation storage means, address selecting means under control of said program storage means adapted to be conditioned in accordance with manifestations of the value of the address portion of a program word stored in said program storage means for selecting the corresponding address position on said drum, means under control of said program manifestation storage means adapted to be conditioned in accordance with manifestations of the operation code portion of the program word stored in said program manifestation storage means for activating said address selecting means to sequentially select said chosen address positions succeeding the address position selected in accordance with manifestations of the address portion of the program word stored in said program storage so that manifestations of said arguments are sequentially selected in ascending order, means for manifesting a value equal to the number of address positions from the first selected argument manifestations through the last selected argument manifestations, auxiliary storage means for storing manifestations of an argument for which the corresponding address position on said drum is sought, an adder for concurrently receiving and adding manifesta-

tions of two values and having an output for transmitting manifestations of the sum of two added values, a first data manifestation conveying link for sequentially conveying manifestations of the selected arguments from said drum to said adder, a second data manifestation conveying link for repeatedly conveying manifestations of the complement of the argument stored in said auxiliary storage means to said adder concurrently with the conveyance of manifestations of said selected arguments, a third data manifestation conveying link coupling said program storage means to said adder, a fourth data manifestation conveying link coupling said manifesting means to said adder, means responsive to a carry at said output from the highest ordered position of said adder to disable said first and said second data manifestation conveying links and to enable said third and said fourth data manifestation conveying links respectively to concurrently convey manifestations of the value of the address portion of the program word from said program manifestations storage means and the value manifested by said manifesting means to said adder so that manifestations of the two values are added to produce at said output manifestations of the value of the address sought, means for transferring manifestations of the value from said output to said accumulator to replace the manifestations of the value of the address portion of the program word in said accumulator, and switching means for conditioning said recording means to record manifestations of the new word stored in said accumulator on said drum.

8. In a data processing machine which is adapted to be programmed by manifestations of internally stored program words, said program words each comprising an address value portion, an operation value portion, and an instruction value portion; the combination of a rotating magnetic drum on which is defined a plurality of address positions arranged in tracks about said drum, each track having a home position; means for recording manifestations of a plurality of arguments of tabular data in consecutive ascending order in chosen address positions beginning with manifestations of the first argument recorded adjacent the home position of one of said tracks and for recording manifestations of program words in other address positions; means for reading manifestations of data recorded on said drum; an accumulator; means for entering manifestations of a program word from said drum into said accumulator; program manifestation storage means adapted to store manifestations of a program word; an address register for initially storing manifestations of the instruction value portion of a program word, said instruction value comprising the value of the address position on said drum of manifestations of a program word; address selecting means under control of said address register for selecting an address position on said drum in accordance with manifestations of the value stored in said address register; means for entering manifestations of a selected program word from said drum into said program storage means; entry control means for causing manifestations of the instruction value in said address register to be replaced by manifestations of the address value portion of the program word in said program manifestation storage means, said address value portion comprising the value of the address of the first argument stored in a track of said drum when said operation value portion of said program value comprises a table look-up code; an operation register; means for entering manifestations of the operation value portion of the program word stored in said program manifestation storage means into said operation register; switching means under control of said operation register adapted to be conditioned in accordance with manifestations of the value stored in said operation register for activating said address selecting means to sequentially select said chosen address positions succeeding the address position selected in accordance with manifestations of the address value stored in said address register when said operation register has manifestations of a table look-up code operation value stored therein so that manifestations of said arguments are sequentially selected in ascending order; means for manifesting a value equal to the number of address positions on a track from the home position of the track through the last selected argument; a distributor for storing manifestations of an argument for which the corresponding address position on said drum is sought; an adder for concurrently receiving and adding manifestations of two values and having an output for transmitting manifestations of the sum of the two received manifestations of values; a first data manifestation conveying link for sequentially conveying manifestations of the values of the selected arguments from said drum to said adder; a second data manifestation conveying link for repeatedly conveying manifestations of the value of the complement of the argument manifestations stored in said distributor to said adder concurrently with the conveyance of manifestations of said selected arguments so that a carry will be produced from the output of said adder when manifestations of a selected argument equals or exceeds the value of the argument manifestations stored in said distributor; a third data manifestation conveying link coupling said program storage means to said adder; a fourth data manifestation conveying link coupling said manifesting means to said adder; means under the control of the output of said adder for disabling said first and said second data manifestation conveying links upon the occurrence of a carry from the highest ordered position of said adder and for enabling said third and said fourth data manifestation conveying links respectively to concurrently convey manifestations of the value of the address portion of the program word stored in said program manifestation storage means and manifestations of the value manifested by said manifesting means to said adder so that manifestations of the two last-named values are added to produce at said output manifestations of the value of the address sought; means for transferring manifestations of the address value from said output to said accumulator to replace the manifestations of the address portion of the program word in said accumulator so that manifestations of a modified program word are stored in said accumulator; means responsive to the carry from the highest ordered position of said adder for enabling said entry control means to replace the manifestations of the address value stored in said address register with manifestations of the instruction value portion of the program word stored in said program storage means whereby manifestations of the program word stored on said drum at the address position corresponding to manifestations of the instruction value in said address register is entered into said program manifestation storage means to replace the manifestations of the word previously stored therein and manifestations of the last-named instruction value in said address register is replaced by manifestations of the address value portion of the program word manifestations last entered into said program storage means; and means for transferring manifestations of the modified program word from said accumulator to the address position corresponding to the manifestations of the address value stored in said address register so that manifestations of said modified program word may be subsequently utilized.

9. In a data processing machine which is adapted to be programmed by manifestations of internally stored program words, said program words each comprising an address value portion, an operation value portion, and an instruction value portion; the combination of a rotating magnetic drum on which is defined a plurality of address positions arranged in a plurality of tracks about said drum, each track having a home position; means for recording manifestations of a plurality of arguments of tabular data in consecutive ascending order in chosen address positions extending over a plu-

rality of said tracks with manifestations of the first argument recorded on said drum at the address position adjacent the home position of a first of said plurality of tracks and for recording manifestions of program words in other address positions; means for reading data recorded on said drum; an accumulator; means for entering manifestations of a program word from said drum into said accumulator, program manifestation storage means adapted to store manifestations of a program word; an address register for initially storing manifestations of the instruction value portion of a program word, said instruction value corresponding to the address position on said drum of manifestations of a program word; address selecting means under control of said address register for selecting an address position on said drum in accordance with manifestations of the value stored in said address register; means for entering manifestations of a selected program word from said drum into said program storage means; entry control means for causing manifestations of the instruction value in said address register to be replaced by manifestations of the address value portion of the program word in said program storage means, said address value portion comprising the address of the manifestations of the first argument stored in said first track of said drum when said operation value portion of the manifestations of said program word stored in said program storage means comprises a table look-up code; an operation register; means for entering manifestations of the operation value portion of the manifestations of the program word stored in said program storage means into said operation register; switching means under control of said operation register adapted to be conditioned in accordance with manifestations of the value stored in said operation register for actuating said address selecting means to sequentially select said chosen address positions of said first track succeeding the address position selected in accordance with the manifestations of the address value stored in said address register when said operation register has manifestations of a table look-up code operation value stored therein so that manifestations of said arguments recorded in said first track are sequentially selected in ascending order; first manifesting means for manifesting a value equal to the number of address positions between the home positions of two adjacent tracks; second manifesting means for manifesting a value equal to the number of address positions on a track between the home position of the track and the last selected argument manifestation inclusive; a distributor for storing manifestations of an argument for which the corresponding address position on said drum is sought; an adder for concurrently receiving and adding manifestations of two values and having an output for transmitting manifestations of the sum of the manifestations of the two received values; a first data manifestation conveying link for sequentially conveying manifestations of the values of the selected arguments from said drum to said adder; a second data manifestation conveying link for repeatedly conveying manifestations of the value of the complement of the argument manifestation stored in said distributor to said adder concurrently with the conveyance of manifestations of said selected arguments so that a carry will be produced from the output of the highest ordered position of said adder when the value of a selected argument equals or exceeds the value of the argument manifestation stored in said distributor; a third data manifestation conveying link coupling said program storage means to said adder; a fourth data manifestation conveying link coupling said first manifesting means to said adder; a fifth data manifestation conveying link coupling said second manifesting means to said adder; means under the control of the output of said adder for temporarily disabling said first and said second data manifestation conveying links upon the selection of a predetermined number of argument manifestation containing address positions of said first track without the occurrence of a carry from the highest ordered position of said adder and for temporarily enabling said third and said fourth data manifestation conveying links respectively to concurrently convey manifestations of the value of the address portion of the program word manifestations stored in said program storage means and the value manifested by said first manifesting means to said adder so that the two last-named manifestations of values are added to produce at said output manifestations of the value of the address of the first address position of a second track of said plurality of tracks; means for transferring manifestations of the value of the address of the first address position of the second track of said plurality of tracks to said program storage means to replace the manifestations of the address portion of the program word manifestations stored in said program storage means so that manifestations of the value in said address register will be replaced by manifestations of the value of the address position adjacent the home position of said second track whereby manifestations of the arguments stored in the chosen address positions of said second track are sequentially selected by said address selecting means; means under the control of the output of said adder for disabling said first and said second data manifestation conveying links upon the occurrence of a carry from the highest ordered position of said adder and for enabling said third and said fifth data manifestation conveying links respectively to concurrently convey manifestations of the value of the address portion of the program word manifestations stored in said program storage means and the value manifested by said second manifesting means to said adder so that the two last-named values are added to produce at said output manifestations of the value of the address sought; means for transferring manifestations of the address value from said output to said accumulator to replace the address portion manifestations of the program word manifestations in said accumulator so that manifestations of a modified program word are stored in said accumulator; means responsive to the carry from the highest ordered position of said adder for enabling said entry control means to replace the address value manifestations stored in said address register with manifestations of the instruction value portion of the program word manifestations stored in said program storage means whereby the program word manifestations stored on said drum at the address position corresponding to the instruction value manifestations in said address register are entered into said program storage means to replace the value manifestations previously stored therein and the last-named instruction value manifestations in said address register are replaced by the address value portion of the program word manifestations last entered into said program storage means; and means for transferring manifestations of the modified program value from said accumulator to the address position corresponding to the address value manifestations of stored in said address register so that manifestations of said modified program value may be subsequently utilized.

10. In a data processing machine which is adapted for performing a table look-up operation under the control of stored manifestations of program words, said program words each comprising an address portion, the combination of a rotating magnetic drum on which is defined a plurality of address positions arranged in tracks about said drum, each track having a home position, means for recording manifestations of a plurality of arguments of tabular data in consecutive numeric order in chosen address positions beginning with the home position of one of said tracks, program manifestation storage means for storing manifestations of a program word containing the address of the home position of one of said tracks hav-

ing manifestations of tabular data stored therein, address selecting means under control of said program manifestation storage means adapted to be conditioned in accordance with the manifestations of the address portion of the manifestations of a program word stored in said program manifestation storage means for sequentially selecting the address position on said drum corresponding to said address position of the manifestation of said program word and said chosen succeeding address positions so that manifestations of said arguments are sequentially selected in numeric order, means for manifesting a value equal to the number of manifestations of arguments sequentially selected, auxiliary storage means for storing manifestations of an argument for which the corresponding address on said drum is sought, an adder for concurrently receiving and adding manifestations of two values, a first data manifestation conveying link adapted to sequentially convey manifestations of the selected arguments from said drum to said adder, a second data manifestation conveying link adapted to repeatedly convey the manifestations of the value of the argument from said auxiliary storage means to said adder concurrently with the conveyance of the manifestations of the selected arguments, said first or said second data manifestation conveying link being adapted to convey data manifestations in complement form, a third data manifestation conveying link connected between said program storage means and said adder, a fourth data manifestation conveying link connected between said manifesting means and said adder, and means responsive to a carry from the highest ordered position of said adder to disable said first an said second data manifestation conveying links and to enable said third and said fourth data manifestation conveying links respectively, to thereby concurrently convey the manifestations of the address portion of said program word and the value manifested by said manifesting means to said adder so that manifestations of the two values are added to produce the manifestation of the address sought.

11. In a data processing machine which is adapted for performing table look-up operations under the control of manifestation of stored program words, said program words each comprising an address portion and an operation code portion, the combination of a rotating magnetic drum on which is defined a plurality of address positions arranged in tracks about said drum, each track having a home position, means for recording manifestations of a plurality of arguments of tabular data in consecutive numeric order in chosen address positions beginning with the home position of one of said tracks and for recording manifestations of program words in other address positions, an accumulator, means for entering manifestations of a program value from said drum into said accumulator, program manifestation storage means adapted to store the manifestations of a program word, means for transferring manifestations of a program word containing the address of the home position of one of said tracks having manifestations of tabular data stored therein and a table look-up operation code from said drum to said program manifestation storage means, address selecting means under control of said program manifestation storage means adapted to be conditioned in accordance with the manifestations of the value of the address portion of manifestations of a program word stored in said program manifestation storage means for first selecting the corresponding address portion on said drum, means under control of said program manifestation storage means adapted to be conditioned in accordance with the operation code portion of the manifestations of the program word stored in said program manifestation storage means for activating said address selecting means to sequentially select said chosen address positions succeeding the address position selected in accordance with the address portion of the manifestations of the program word stored in said program manifestation storage so that manifestations of said arguments are sequentially selected in numeric order, means for manifesting a value equal to the number of address positions from the manifestations of the first selected argument through the manifestations of the last selected argument, auxiliary storage means for storing manifestations of an argument for which the corresponding address position on said drum is sought, an adder for concurrently receiving and adding manifestations of two values and having an output for transmitting the manifestations of the sum of the two added manifestations of values, a first data manifestation conveying link for sequentially conveying manifestations of the values of the manifestations of the selected arguments from said drum to said adder, a second data manifestation conveying link for repeatedly conveying the manifestations of the value of the argument manifestations stored in said auxiliary storage means to said adder concurrently with the conveyance of manifestations of said selected arguments, said first data manifestation conveying link or said second data manifestation conveying being adapted to convey manifestations of data in the complement form, a third data manifestation conveying link coupling said program storage means to said adder, a fourth data manifestation conveying link coupling said manifesting means to said adder, means responsive to a carry at said output from the highest ordered position of said adder to disable said first and said second data manifestation conveying link and to enable said third and said fourth data manifestation conveying link respectively to concurrently convey the manifestations of the value of the address portion of the program word from said program manifestation storage means and the value manifested by said manifesting means to said adder so that manifestations of the two values are added to produce at said output the manifestations of the value of the address sought, means for transferring the manifestations of the value from said output to said accumulator to replace the value manifestations of the address portion of the program word manifestations in said accumulator, and switching means for conditioning said recording means to record the new word manifestations stored in said accumulator on said drum.

12. In a data processing machine which is adapted to be programmed by internally stored manifestations of program words, said program words each comprising an address value portion, an operation value portion, and an instruction value portion: the combination of a rotating magnetic drum on which is defined a plurality of address positions arranged in tracks about said drum, each track having a home position; means for recording a plurality of manifestations of arguments of tabular data in consecutive numeric order in chosen address positions beginning with manifestations of the first argument recorded adjacent the home position of one of said tracks and for recording manifestations of program words in other address positions; means for reading data manifestations recorded on said drum; an accumulator; means for entering manifestations of a program value from said drum into said accumulator; program manifestation storage means adapted to store manifestations of a program word; an address register for initially storing the instruction value portion of manifestations of a program word, said instruction value corresponding to the address position on said drum of the manifestations of a program word; address selecting means under control of said address register for selecting an address position on said drum in accordance with the manifestations of the value stored in said address register; means for entering manifestations of a selected program word from said drum into said program manifestation storage means; entry control means for causing the instruction value manifestations in said address register to be replaced by the address value portion of the program word manifestations in said program storage means, said address value portion comprising the address of the manifestations of the first argument stored in a track of said

drum when said operation value portion of said program word comprises a table look-up code; an operation register; means for entering the operation value portion of the manifestations of the program word stored in said program manifestation storage means into said operation register; switching means under control of said operation register adapted to be conditioned in accordance with the manifestation of the value stored in said operation register for actuating said address selecting means to sequentially select said chosen address positions succeeding the address position selected in accordance with the manifestations of the address value stored in said address register when said operation register manifestations of has a table look-up code operation value stored therein so that manifestations of said arguments are sequentially selected in numeric order; means for manifesting a value equal to the number of address positions on a track from the home position of the track through the last selected argument manifestations; a distributor for storing manifestations of an argument for which the corresponding address position on said drum is sought; an adder for concurrently receiving and adding manifestations of two values and having an output for transmitting the manifestations of the sum of the two received values; a first data manifestation conveying link for sequentially conveying the manifestations of the value of the selected arguments from said drum to said adder; a second data manifestation conveying link for repeatedly conveying the manifestations of the value of the argument stored in said distributor to said adder concurrently with the conveyance of the manifestations of the values of said selected arguments so that a carry will be produced from the output of the highest ordered position of said adder when the value of the selected argument is equal to or next adjacent to the value of the manifestations of the argument stored in said distributor, said first data manifestation conveying link or said second data manifestation conveying link being adapted to convey data manifestations in the complement form; a third data manifestation conveying link coupling said program storage means to said adder; a fourth data manifestation conveying link coupling said manifesting means to said adder; means under the control of the output of said adder for disabling said first and said second data manifestation conveying links upon the occurrence of a carry from the highest order position of said adder and for enabling said third and said fourth data manifestation conveying links respectively to concurrently convey the manifestations of the value of the address portion of the program value manifestations stored in said program manifestation storage means and the value manifested by said manifesting means to said adder so that the manifestations of the two last-named values are added to produce at said output the manifestations of the value of the address sought; means for transferring the manifestations of the address value from said output to said accumulator to replace the address portion of the program word manifestations in the accumulator so that manifestations of a modified program word are stored in the accumulator; means responsive to the carry from the highest order position of said adder for enabling said entry control means to replace the address value manifestations stored in said address register with the instruction value portion of the program word manifestations stored in said program manifestation storage means whereby manifestations of a program word stored on said drum at the address position corresponding to the manifestations of the instruction value in said address register are entered into said program manifestation storage means to replace the value manifestations previously stored therein and the last-named instruction value manifestations in said address register are replaced by the address value portion of the program word manifestations last entered into said program manifestation storage

means; and means for transferring the manifestations of the modified program word from said accumulator to the address position corresponding to the manifestations of the address value stored in said address register so that manifestations of said modified program word may be subsequently utilized.

13. In a data processing machine which is adapted to be programmed by internally stored manifestations of program values, said program values each comprising an address value portion, an operation value portion, and an instruction value portion; the combination of a rotating magnetic drum on which is defined a plurality of address positions arranged in a plurality of tracks about said drum, each track having a home position; means for recording a plurality of manifestations of arguments of tabular data in consecutive numeric order in chosen address positions extending over a plurality of said tracks with the first argument manifestations recorded on said drum at the address position adjacent the home position of a first of said plurality of tracks and for recording manifestations of program values in other address positions; means for reading manifestations of data recorded on said drum; an accumulator; means for entering manifestations of a program value from said drum into said accumulator; program manifestation storage means adapted to store manifestations of a program value; an address register for initially storing the instruction value portion of manifestations of a program value, said instruction value corresponding to the address position on said drum of manifestations of a program value; address selecting means under control of said address register for selecting an address position on said drum in accordance with the value manifestations stored in said address register; means for entering manifestations of a selected program value from said drum into said program manifestation storage means; entry control means for causing the manifestations of the instruction value in said address register to be replaced by the address value portion of the program value manifestations in said program manifestation storage means, said address value portion comprising the address of the manifestations of of the first argument stored in said first track of said drum when said operation value portion of the manifestations of said program value stored in said program manifestation storage means comprises a table look-up code; an operation register; means for entering the operation value portion of the program value manifestations stored in said program manifestation storage means into said operation register; switching means under control of said operation register adapted to be conditioned in accordance with the value manifestations stored in said operation register for actuating said address selecting means to sequentially select the chosen address positions of said first track succeeding the address position selected in accordance with the address value manifestations stored in said address register when said operation register has manifestations of a table look-up code operation value stored therein so that manifestations of said arguments recorded in said first track are sequentially selected in numeric order; first manifesting means for manifesting a value equal to the number of address positions between the home position of two succeeding tracks; second manifesting means for manifesting a value equal to the number of address positions on a track from the home position of the track to the manifestations of the last selected argument of the said track; a distributor for storing manifestations of an argument for which the corresponding address position on said drum is sought; an adder for concurrently receiving and adding manifestations of two values and having an output for transmitting the manifestations of the sum of the two received manifestations of values; a first data manifestation conveying link for sequentially conveying manifestations of the values of the selected manifestations of arguments from said drum to said adder; a second data conveying link for

repeatedly conveying the manifestations of the value of the argument manifestations stored in said distributor to said adder concurrently with the conveyance of the manifestations of the values of manifestations of said selected arguments so that a carry will be produced from the output of the highest ordered position of said adder when the value of the manifestations of a selected argument equals or lies next adjacent the value of the manifestations of the argument stored in said distributor; a third data manifestation conveying link coupling said program storage means to said adder; a fourth data manifestation conveying link coupling said first manifesting means to said adder; a fifth data manifestation conveying link coupling said second manifesting means to said adder; means under the control of the output of said adder for temporarily disabling said first and said second data manifestation conveying links upon the selection of a predetermined number of argument manifestation containing address positions of said first track without the occurrence of a carry from the highest ordered position of said adder and for temporarily enabling said third and said fourth data manifestation conveying links respectively to concurrently convey the manifestations of the value of the address portion of the program value manifestations stored in said program storage means and the value manifested by said first manifesting means to said adder so that the manifestations of the two last-named values are added to produce at said output the manifestations of the value of the address of the first address position of a second track of said plurality of tracks; means for transferring the manifestations of the value of the address of the first address position of the second track of said plurality of tracks to said program storage means to replace the address portion of the program value manifestations stored in said program storage means so that the value manifestations in said address register will be replaced by the value manifestations of the address position adjacent the home position of said second track whereby the argument manifestations stored in the chosen address positions of said second track are sequentially selected by said address selecting means; means under the control of the output of said adder for disabling said first and said second data manifestation conveying links on the occurrence of a carry from the highest order position of said adder and for enabling said third and said fifth data manifestation conveying links respectively to concurrently convey the manifestations of the value of the address portion of the program value manifestations stored in said program manifestation storage means and the value manifested by said second manifesting means to said adder so that manifestations of the two last-named values are added to produce at said output the manifestations of the value of the address sought; means for transferring manifestations of the address value from said output to said accumulator to replace the address portion of the program value manifestations in said accumulator so that a modified program value manifestations are stored in said accumulator; means responsive to the carry from the highest ordered position of said adder for enabling said entry control means to replace the address value manifestations stored in said address register with the instruction value portion of the program value manifestations stored in said program storage means whereby the program value manifestations stored on said drum at the address position manifestations corresponding to the instruction value manifestations in said address register are entered into said program storage means to replace the value manifestations previously stored therein and the last-named instruction value manifestations in said address register are replaced by the address value portion of the program value manifestations last entered into said program storage means; and means for transferring the modified program value manifestations from said accumulator to the address position corresponding to the address value manifestations stored in said address register so that said modified program value manifestations may be subsequently utilized.

14. Apparatus according to claim 9, in combination with means for recording manifestations of functions of the arguments of the table in address positions on said drum a constant number of address positions removed from the address positions of the manifestations of the arguments, means for storing the manifestations of the value of the number of address positions the manifestations of the arguments are removed from their corresponding functions, and means for concurrently entering the manifestations of the last-named value and the manifestations of the value of the address of an argument into said adder to produce the manifestations of the value of the address position in which the manifestations of the function of the argument is stored.

15. In a data processing machine which is adapted for locating the address of the manifestations of an argument of a table, the combination of means for recording the manifestations of arguments of the table in ascending order in successive address positions of a storage medium, storage means for storing the manifestations of the value of the address position of the first recorded argument manifestations, means for storing manifestations of the value of an argument for which an address is sought, means for successively comparing the manifestations of the value of the argument for which an address is sought with consecutive recorded manifestations of arguments starting with the first recorded argument manifestations and adapted to manifest an equal to or greater than signal, means for manifesting a value equal to the number of address positions from the address position of the first recorded argument manifestations through the address position of the first recorded argument manifestations producing an equal to or greater than signal in said comparing means, and means responsive to said equal to or greater than signal for adding the manifestations of the value manifested by said manifesting means to the value manifestations stored in said storage means to produce the manifestations of the value of the address sought.

16. In a data processing machine which is adapted for locating the manifestations of a function of an argument of a table, the combination of a rotating magnetic drum on which is defined a plurality of consecutively arranged address positions, means for recording manifestations of the arguments of the table in ascending order in a first group of address positions, means for recording manifestations of corresponding functions of the arguments in other groups of address positions removed from the address positions of the arguments by constant numbers of address positions, storage means for storing the manifestations of the value of the first address position of said first group of address positions, means for storing manifestations of the value of an argument for which a function is sought, means for successively comparing the manifestations of the value of the argument for which a function is sought with consecutive recorded manifestations of arguments starting with the first address position of said first group and adapted to manifest an equal to or greater than signal, means for manifesting a value equal to the number of address positions from the address position of the first recorded argument manifestations through the address position of the first recorded argument manifestations producing an equal to or greater than signal in said comparing means, means responsive to said equal to or greater than signal for adding to the stored value manifestations of the value manifested by said manifesting means thereby storing the value manifestations of the address position corresponding to the argument for which a function is sought, and means subsequently acting to add to the last-named stored value manifestations of a value equal to the number of address positions by which the recorded manifestations of arguments are removed from the desired recorded mani-

festations of functions to thereby produce the manifestations of the value of the address of the function sought.

17. In a data processing machine which is adapted for locating the manifestations of a function of an argument of a table; the combination of a rotating magnetic drum on which is defined a plurality of groups of consecutively arranged address positions; means for recording manifestations of the arguments of the table in ascending order in a plurality of said groups, each group having the manifestations of an argument recorded in the first address position thereof and in a predetermined number of consecutive address positions thereafter; means for recording corresponding manifestations of functions of the arguments in other groups of address positions removed from the address positions of the manifestations of the arguments by constant numbers of address positions; storage means for storing the manifestations of the value of the first address position of the first group in which manifestations of arguments are recorded; means for storing manifestations of the value of an argument for which a function is sought; means for successively comparing the manifestations of the value of the argument for which a function is sought with the manifestations of the values of the arguments located at successive address positions in said groups starting with the first address position of each group and adapted to produce an equal to or greater than signal, means responsive in the absence of an equal to or greater than signal for adding to the manifestations of the value stored in said storage means the manifestations of a value equal to the number of address positions in a group if the manifestation of an argument is not found in the first group having a value equal to or greater than the value of the argument for which a function is sought thereby storing the manifestations of the value of the first address position of a second group of argument manifestation containing address positions, means responsive to an equal to or greater than signal for adding to the value manifestations stored in said storage means manifestations of a value equal to the number of address positions from the first address position of the second group through the first address position of the second group containing the manifestations of an argument having a value equal to or greater than the value of the argument for which an address is sought thereby storing the manifestations of the value of the address position corresponding to the argument for which a function is sought, and means for subsequently adding to the manifestations of the last-named stored value the manifestations of a value equal to the number of address positions by which the recorded manifestations of arguments are removed from the desired recorded manifestations of functions to thereby produce the manifestations of the value of the address of the function sought.

18. Apparatus according to claim 16, characterized further in that said means for adding to the stored value manifestations comprises a one digit matrix adder adapted to concurrently receive and add two manifestations of values.

19. In a data processing machine which is adapted for locating the address of the manifestations of an argument of a table, the combination of a rotating magnetic drum on which is defined a plurality of address positions, means for recording manifestations of the arguments of the table in numeric order in successive address positions on said drum, means for reading recorded data manifestations from said drum, storage means for storing manifestations of the value of the address position of the first recorded argument manifestations, means for storing manifestations of the value of an argument for which an address is sought, means for successively comparing manifestations of the value of the argument for which an address is sought with the manifestations of the values of consecutive recorded manifestations of arguments starting with the first recorded argument manifestations and adapted to produce an equal to or next adjacent signal, means for manifesting a value equal to the number of address positions from the address position of the first ıecorded argument manifestations through the address position of the first recorded argument manifestations having a value equal to or next adjacent to the value of the argument manifestations for which an address is sought, and means responsive to said equal to or next adjacent signal for adding the manifestations of the value manifested by said manifesting means to the manifestations of the value stored in said storage means to produce the manifestations of the value of the address sought.

20. In a data processing machine which is adapted for locating a function of an argument of a table, the combination of a rotating magnetic drum on which is defined a plurality of consecutively arranged address positions, means for recording manifestations of the arguments of the table in numeric order in a first group of address positions, means recording corresponding manifestations of functions of the arguments in other groups of address positions removed from the address positions of the arguments by constant numbers of address positions, storage means for storing the manifestations of the value of the first address position of said first group, means for storing manifestations of the value of an argument for which a function is sought, means for successively comparing the manifestations of the value of the argument for which a function is sought with the manifestations of the values of consecutive recorded manifestations of arguments starting with the manifestations of the value of the argument manifestations recorded in the first address position of said first group and adapted to produce an equal to or next adjacent signal, means responsive to an equal to or next adjacent signal for adding to the manifestations of the value stored in said storage means the manifestations of a value equal to the number of address positions from the address position of the first recorded argument manifestations through the address position of the first recorded argument manifestations having a value equal to or next adjacent to the value of the argument for which a function is sought thereby storing the manifestations of the value of the address position corresponding to the argument for which a function is sought, and means for subsequently adding to the manifestations of the last-named stored value the manifestations of a value equal to the number of address positions by which the recorded manifestations of arguments are removed from the desired corresponding recorded manifestations of functions to thereby produce the manifestations of the value of the address of the manifestations of the function sought.

21. In a data processing machine adapted to be programmed in accordance with internally stored manifestations of program values, said program values each comprising an operation value portion, an address value portion, and an instruction value portion; the combination of a rotating magnetic drum on which is defined a plurality of address positions, means for recording manifestations of program values at selected ones of said address positions and other data manifestations at other of said address positions, means for reading recorded data manifestations from said drum, program manifestation storage means adapted to store manifestations of a program value, an address register adapted to be conditioned in accordance with manifestations of a value stored therein, an operation register adapted to be conditioned in accordance with manifestations of an operation value stored therein, first entry control means under control of said address register for causing the manifestations of a program value stored at the address position corresponding to the manifestations of the value stored in said address register to be entered into said program storage means, second entry control means adapted to cause the address value portion of the program value manifestations

stored in said program manifestation storage means to be entered into said address register to replace the manifestations of the value previously stored therein and to cause the operation value portion of the program value manifestations stored in said program manifestation storage means to be entered into said operation register, switching means under control of said address register adapted to be conditioned in accordance with manifestations of an address value stored in said address register for selecting the address of data manifestations to be processed by said machine in accordance with the condition of said operation register, and third entry control means for causing the instruction value portion of the program value manifestations stored in said program manifestations storage means to be entered into said address register to replace the manifestations of the address value previously stored therein and to enable said first entry control means whereby the program value manifestations at the address on said drum corresponding to the value manifestations stored in said address register are entered into said program manifestation storage means.

22. In a data processing machine adapted to be programmed in accordance with internally stored manifestations of program values, said program values each comprising an address value portion and an instruction value portion; the combination of a rotating magnetic drum on which is defined a plurality of address positions, means for recording manifestations of program values at selected ones of said address positions and other data manifestations at other of said address positions, means for reading recorded data manifestations from said drum, program manifestations storage means for storing manifestations of program data, an address register adapted to be conditioned in accordance with manifestations of a value stored therein, first entry control means adapted to cause the address value portion of the manifestations of the program value stored in said program storage means to be entered into said address register, means for manifesting one or the other of two conditions, second entry control means under control of said manifesting means for causing the address value portion of the manifestations of the program value to remain in said address register or for causing the instruction value portion of the program value manifestations stored in said program storage means to be entered into said address register to replace the manifestations of the address value previously stored therein in accordance with the one or the other of the two conditions manifested by said manifesting means, third entry means under the control of said address register and adapted to be conditioned in accordance with the manifestations of the value stored in said address register for causing the program value manifestations stored on said drum at the address position corresponding to the value manifestations stored in said address register to be entered into said program storage means.

23. Apparatus according to claim 21, in combination with manifesting means for manifesting a first or a second condition, means under control of said manifesting means for disabling said third entry control means upon the manifestation of said first condition so that the address value manifestations stored in said address register remain stored therein, and ineffective upon the manifestation of said second condition so that manifestations of one or the other of two program values depending on the condition manifested by said manifesting means is entered into said program manifestation storage means from the address location defined by the contents of said address register.

24. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the operation to be performed and the address of the next succeeding program step, the combination comprising, program storage means for storing a program step, control means responsive to manifestations of data stored in said program storage means for controlling said machine, data manifestation storage means for storing a plurality of other program steps and manifestations of other data, auxiliary storage means for storing a program step, first transmitting means responsive to said control means for transmitting data manifestations from said data manifestation storage means to said program storage means in a sequence determined by the manifestations of data of successive program steps, error detecting means for continuously monitoring the operation of said machine, and means responsive to said error detecting means for interrupting the operation of said first transmitting means upon the occurrence of an error and for transmitting data manifestations from said auxiliary storage means to said program storage means to thereby alter the sequence of operations of said machine.

25. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the operation to be performed and the address of the next succeeding program step, the combination comprising, program storage means for storing a program step, control means responsive to manifestations of data stored in said program storage means for controlling said machine, data manifestation storage means, first transmitting means responsive to said control means for transmitting data manifestations from said data manifestation storage means to said program storage means in a sequence determined by the manifestations of data of successive program steps, auxiliary storage means for storing manifestations of data comprehending the address of the first program step of an alternate sequence of program steps, error detecting means for continuously monitoring the operation of said machine, and means responsive to said error detecting means for interrupting the operation of said first transmitting means upon the occurrence of an error and for transmitting data manifestations from said auxiliary storage means to said program storage means, to thereby initiate an alternate sequence of operations.

26. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the operation to be performed and the address of the next succeeding program step, the combination comprising, program storage means for storing a program step, control means responsive to manifestations of data stored in said program storage means for controlling said machine, data manifestation storage means, first transmitting means responsive to said control means for transmitting data manifestations from said data manifestation storage means to said program storage means in a sequence determined by the manifestations of data of successive program steps, auxiliary storage means for storing manifestations of data comprehending the address of the first program step of a group of program steps in which an error may occur, error detecting means for continuously monitoring the operation of said machine, and means responsive to said error detecting means for interrupting the operation of said first transmitting means upon the occurrence of an error and for transmitting data manifestations from said auxiliary storage means to said program storage means, to thereby repeat the group of operations in which the error occurred.

27. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the operation to be performed and the address of the next succeeding program step, the combination comprising, program storage means for storing a program step, control means responsive to manifestations of data stored in said program storage means for controlling said machine, data manifestation storage means, first transmitting means responsive to said control means for transmitting data manifestations from said data manifestation storage means to said program storage means in a sequence determined by the manifestations of data of successive program steps, auxiliary storage means for storing manifestations of data

comprehending the address of a program step, error detecting means for continuously monitoring the operation of said machine, and means responsive to said error detecting means for interrupting the operation of said first transmitting means upon the occurrence of an error and for transmitting data manifestations from said auxiliary storage means to said program storage means, to thereby start a sequence of operations beginning with the program steps stored in said auxiliary storage means.

28. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the operation to be performed and the address of the next succeeding program step, the combination comprising, program storage means for storing a program step, control means responsive to manifestations of data stored in said program storage means for controlling said machine, data manifestation storage means, first transmitting means responsive to said control means for transmitting data manifestations from said data manifestation storage means to said program storage means in a sequence determined by the manifestations of data of successive program steps, auxiliary storage means for storing manifestations of data comprehending the address of the first program step of the sequence, error detecting means for continuously monitoring the operation of said machine, and means responsive to said error detecting means for interrupting the operation of said first transmitting means upon the occurrence of an error and for transmitting data manifestations from said auxiliary storage means to said program storage means, to thereby cause the machine to repeat the sequence of operations preceding the operation in which the error occurred.

29. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the addresses of first and second positions in storage and the operation to be performed by said machine, the combination comprising, storage means having a plurality of addressable storage positions for storing program steps and manifestations of data to be operated upon by said machine, a program register having a first storage position for storing manifestations of data comprehending the address of a first position in said storage means and a second storage position for storing manifestations of data comprehending the address of a second position in said storage means, a first bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, a second bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, an address register adapted to store manifestations of data for addressing a position in said storage means in accordance with the manifestations of data stored in said address register, means responsive to the manifestation of the first stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said first position of said program register and responsive to the manifestation of the second stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said second position of said program register, means for transmitting manifestations of data from said program register to said address register in accordance with the stable state manifested by said first bi-stable device, a first channel responsive to the manifestation of the first stable state by said second bi-stable device for transmitting data from the storage position of said storage means addressed by said address register to said program register, a calculator section, and a second channel responsive to the manifestation of the second stable state by said second bi-stable device for transmitting data manifestations from the storage position of said storage means addressed by said address register to said calculator section.

30. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the addresses of first and second positions in storage and the operation to be performed by said machine, the combination comprising, storage means having a plurality of addressable storage positions for storing program steps and manifestations of data to be operated upon by said machine, a program register having a first storage position for storing manifestations of data comprehending the address of a first position in said storage means and a second storage position for storing manifestations of data comprehending the address of a second position in said storage means, a first bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, a second bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, an address register adapted to store manifestations of data for addressing a position in said storage means in accordance with the manifestations of data stored in said address register, means responsive to the manifestation of the first stable state by said first bi-stable device for conditioning said address register to rereceive data manifestations from said first position of said program register and responsive to the manifestation of the second stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said second position of said program register, means for transmitting manifestations of data from said program register to said address register in accordance with the stable state manifested by said first bi-stable device, and a first channel responsive to the manifestation of the first stable state by said second bi-stable device for transmitting data from an addressed storage position of said storage means to said program register, a calculator section, a second channel adapted to be conditioned to transmit data manifestations from an addressed storage position of said storage means to said calculator section and to manifest a complete signal upon the completion of a transmission, means responsive to a change in the stable state of said first bi-stable device for changing the stable state of said second bi-stable device, and means responsive to a manifested complete signal for reversing the stable state of said first bi-stable device.

31. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the addresses of first and second positions in storage and the operation to be performed by said machine, the combination comprising, storage means having a plurality of addressable storage positions for storing program steps and manifestations of data to be operated upon by said machine, a program register having a first storage position for storing manifestations of data comprehending the address of a first position in said storage means and a second storage position for storing manifestations of data comprehending the address of a second position in said storage means, a first bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, a second bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, an address register adapted to store manifestations of data for addressing a position in said storage means in accordance with the manifestations of data stored in said address register, means responsive to the manifestation of the first stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said first position of said program register and responsive to the manifestation of the second stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said second position of said program register, means for transmitting manifestations of data from said program register to said address

register in accordance with the stable state manifested by said first bi-stable device, a first channel responsive to the manifestation of the first stable state by said second bi-stable device for transmitting data from the storage position of said storage means addressed by said address register to said program register, a calculator section, a second channel responsive to the manifestation of the second stable state by said second bi-stable device for transmitting data manifestations from the storage position of said storage means addressed by said address register to said calculator section, means under control of said first and second channels for producing a signal upon the completion of a data manifestation transmission by one of said channels, means responsive to a signal produced by said signal producing means for reversing the stable state of said first bi-stable device, and means responsive to a reversal in the stable state of said first bi-stable device for reversing the stable state of said second bi-stable device, whereby manifestations of data from alternately addressed storage positions in said storage means are alternately transmitted to said program register and to said calculator section.

32. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the addresses of first and second positions in storage and the operation to be performed by said machine, the combination comprising, storage means having a plurality of addressable storage positions for storing program steps and manifestations of data to be operated upon by said machine, a program register having a first storage position for storing manifestations of data comprehending the address of a first position in said storage means and a second storage position for storing manifestations of data comprehending the address of a second position in said storage means, a first bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, a second bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, an address register adapted to store manifestations of data for addressing a position in said storage means in accordance with the manifestations of data stored in said address register, means responsive to the manifestation of the first stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said first position of said program register and responsive to the manifestation of the second stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said second position of said program register, means for transmitting manifestations of data from said program register to said address register in accordance with the stable state manifested by said first bi-stable device, a first channel responsive to the manifestation of the first stable state by said second bi-stable device for transmitting data from the storage position of said storage means addressed by said address register to said program register, a calculator section, a second channel adapted to be conditioned to transmit data manifestations from the storage position of said storage means addressed by said address register to said calculator section and to manifest a complete signal upon the completion of a transmission, an operation register for storing manifestations of data comprehending the operation to be performed by said machine, means under control of said operation register for testing said calculator section and manifesting a condition of said calculator, means responsive to a change in the stable state of said first bi-stable device for changing the stable state of said second bi-stable device, means responsive to a manifested complete signal operative to reverse the stable state of said first bi-stable device, and means responsive to a predetermined manifestation by said testing means for blocking the operation of said last-named means and for reversing the stable state of said second bi-stable device, whereby manifestations of data from two successively addressed positions in said storage means are transmitted to said program register upon the occurrence of a predetermined condition.

33. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the addresses of first and second positions in storage and the operation to be performed by said machine, the combination comprising, storage means having a plurality of addressable storage positions for storing program steps and manifestations of data to be operated upon by said machine, a program register having a first storage position for storing manifestations of data comprehending the address of a first position in said storage means and a second storage position for storing manifestations of data comprehending the address of a second position in said storage means, a first bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, a second bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, an address register adapted to store manifestations of data for addressing a position in said storage means in accordance with the manifestations of data stored in said address register, means responsive to the manifestation of the first stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said first position of said program register and responsive to the manifestation of the second stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said second position of said program register, means for transmitting manifestations of data from said program register to said address register in accordance with the stable state manifested by said first bi-stable device, a first channel responsive to the manifestation of the first stable state by said second bi-stable device for transmitting data from the storage position of said storage means addressed by said address register to said program register, a calculator section including a component adapted to be tested for one or the other of two specific states of said component, a second channel responsive to the manifestation of the second stable state by said second bi-stable device for transmitting data manifestations from the storage position of said storage means addressed by said address register to said calculator section, means under control of said first and second channels for producing a first signal upon the completion of a data manifestation transmission by one of said channels, restart means responsive to said first signal for reversing the stable state of said first bi-stable device and adapted to be disabled, means responsive to a reversal in the stable state of said first bi-stable device for reversing the stable state of said second bi-stable device, an operation register adapted to store manifestations of data comprehending a test to be performed, means under control of said operation register for testing said component and producing a second signal indicative of a first state of said component, and means responsive to said second signal for disabling said restart means and for reversing the stable state of said second bi-stable device.

34. Apparatus according to claim 33 wherein said first bi-stable circuit comprises a latch circuit and said second bi-stable device comprises a pair of interconnected latch circuits.

35. Apparatus according to claim 33 wherein said component comprises an accumulator adapted to be tested for manifestations of a plus or a minus sign.

36. Apparatus according to claim 33 wherein said component comprises an accumulator adapted to be tested for manifestations of a zero or a no zero condition of a predetermined denominational order.

37. Apparatus according to claim 33 wherein said component comprises an accumulator adapted to be tested

for manifestations of zeros in all denominational orders or not zeros in all denominational orders.

38. Apparatus according to claim 33 wherein said component comprises a distributor adapted to be tested for manifestations of a plus or a minus sign.

39. Apparatus according to claim 33 wherein said component comprises an accumulator adapted to be tested for an overflow or a no overflow condition.

40. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the addresses of first and second positions in storage and the operation to be performed by said machine, the combination comprising, storage means having a plurality of addressable storage positions for storing program steps and manifestations of data to be operated upon by said machine, a program register having a first storage position for storing manifestations of data comprehending the address of a first position in said storage means and a second storage position for storing manifestations of data comprehending the address of a second position in said storage means, a first bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, a second bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition, an address register adapted to store manifestations of data for addressing a position in said storage means in accordance with the manifestations of data stored in said address register, means responsive to the manifestation of the first stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said first position of said program register and responsive to the manifestation of the second stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said second position of said program register, means for transmitting manifestations of data from said program register to said address register in accordance with the stable state manifested by said first bi-stable device, a first channel responsive to the manifestation of the first stable state by said second bi-stable device for transmitting data from the storage position of said storage means addressed by said address register to said program register, a calculator section including means for producing a calculation complete signal, a second channel responsive to the manifestation of the second stable state by said second bi-stable device for transmitting data manifestations from the storage position of said storage means addressed by said address register to said calculator section, means under control of said first channel for producing a second signal upon the completion of a data manifestation transmission by said first channel; means responsive to the coincidence of said second signal and a calculation complete signal for reversing the stable state of said first bi-stable device; means under control of said second channel for reversing the stable state of said first bi-stable device upon the completion of a data manifestation transmission by said second channel, and means responsive to a reversal in the stable state of said first bi-stable device for reversing the stable state of said second bi-stable device, whereby data manifestations may be transmitted to said program register simultaneously with the operation of said calculator section.

41. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the addresses of first and second positions in storage and the operation to be performed by said machine; the combination comprising: storage means having a plurality of addressable storage positions for storing program steps and manifestations of data to be operated upon by said machine; a program register having a first storage position for storing manifestations of data comprehending the address of a first position in said storage means, a second storage position for storing manifestations of data comprehending the address of a second position in said storage means and a third storage position for storing manifestations of data comprehending the operation to be performed by said machine; a first bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition; a second bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition; an address register adapted to store manifestations of data for addressing a position in said storage means in accordance with the manifestations of data stored in said address register; an operation register adapted to store manifestations of data for controlling the operation of said machine; means responsive to the manifestation of the first stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said first position of said program register and for conditioning said operation register to receive data manifestations from said third position of said program register, and responsive to the manifestation of the second stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said second position of said program register; means for transmitting manifestations of data from said program register to said address register and said operation register in accordance with the stable state manifested by said first bi-stable device; a first channel responsive to the manifestation of the first stable state by said second bi-stable device for transmitting data from the storage position of said storage means addressed by said address register to said program register; a calculator section including means for producing a calculation complete signal; a second channel responsive to the manifestation of the second stable state by said second bi-stable device for transmitting data manifestations from the storage position of said storage means addressed by said address register to said calculator section; means under control of said first channel for producing a second signal upon the completion of a data manifestation transmission by said first channel; means responsive to the coincidence of said second signal and a calculation complete signal for reversing the stable state of said first bi-stable device; means under control of said second channel for reversing the stable state of said first bi-stable device upon the completion of a data manifestation transmission by said second channel; means responsive to a reversal in the stable state of said first bi-stable device for reversing the stable state of said second bi-stable device; means under control of said operation register for producing a third signal; and means responsive to the coincidence of said third signal and said second signal for reversing the stable state of said first bi-stable device, whereby a plurality of operations may be simultaneously performed by said machine.

42. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the addresses of first and second positions in storage and the operation to be performed by said machine; the combination comprising: storage means having a plurality of addressable storage positions for storing program steps and manifestations of data to be operated upon by said machine; a program register having a first storage position for storing manifestations of data comprehending the address of a first position in said storage means, a second storage position for storing manifestations of data comprehending the address of a second position in said storage means and a third storage position for storing manifestations of data comprehending the operation to be performed by said machine; a first bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition; a second bi-stable device adapted to be set in a first or a second stable state for manifesting a first or a second stable state condition; an address register adapted to store manifestations of data for addressing a position in said storage means in accordance with the manifestations of

data stored in said address register; an operation register adapted to store manifestations of data for controlling the operation of said machine; means responsive to the manifestation of the first stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said first position of said program register and for conditioning said operation register to receive data manifestations from said third position in said program register and responsive to the manifestation of the second stable state by said first bi-stable device for conditioning said address register to receive data manifestations from said second position of said program register; means for transmitting manifestations of data from said program register to said address register and said operation register in accordance with the stable state manifested by said first bi-stable device; a first channel responsive to the manifestation of the first stable state by said second bi-stable device for transmitting data from the storage position of said storage means addressed by said address register to said program register; a calculator section including means for producing a calculation complete signal; a second channel responsive to the manifestation of the second stable state by said second bi-stable device for transmitting data manifestations from the storage position of said storage means addressed by said address register to said calculator section; means under control of said first channel for producing a second signal upon the completion of a data manifestation transmission by said first channel; means responsive to the coincidence of said second signal and a calculation complete signal for reversing the stable state of said bi-stable device, means under control of said second channel for reversing the stable state of said first bi-stable device upon the completion of a data manifestation transmission by said second channel; means responsive to a reversal in the stable state of said first bi-stable device for reversing the stable state of said second bi-stable device; means under control of said operation register for producing a third signal; means responsive to the coincidence of said second and said third signals for reversing the stable state of said first bi-stable device; and an interlock for preventing simultaneous use of a single component of said calculator section.

43. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the operation to be performed and the address of two storage positions in said machine; the combination comprising: storage means having a plurality of addressable storage positions for storing program steps and other data manifestations; a program register for storing a program step comprised of manifestations of data comprehending the address of the next succeeding program step of a first sequence of program steps and the address of the first program step of an alternate sequence of program steps, the last program step of said alternate sequence of program steps including manifestations of data comprehending the address of said next step of said first sequence of steps; data manifestation transmitting means for transmitting data manifestations from said storage means to said program register; an address register for controlling said data manifestation transmitting means to transmit data from an address position in said storage means in accordance with the data manifestation stored in said address register; condition manifesting means for manifesting first and second conditions; and means responsive to said condition manifesting means for transmitting the manifestations of data comprehending the address of the next program step of said first sequence of program steps from said program register to said address register in response to the manifestation of the first condition and for transmitting data manifestations comprehending the first step of said alternate sequence of steps from said program register to said address register in response to the manifestation of the second condition, whereby a

predetermined group of other program steps may be inserted into a sequence of program steps.

44. In a data processing machine of the type controlled by stored program steps each comprised of manifestations of data comprehending the operation to be performed and the address of two storage positions in said machine; the combination comprising: storage means having a plurality of addressable storage positions for storing program steps and other data manifestations; a program register for storing a program step comprised of manifestations of data comprehending the address of the next succeeding program step of a first sequence of program steps and the address of the first program step of an alternate sequence of program steps, the last program step of said second sequence of program steps including manifestations of data comprehending the address of a predetermined program step of said first sequence of steps; data manifestation transmitting means for transmitting data manifestations from said storage means to said program register; an address register for controlling said data manifestation transmitting means to transmit data from an address position in said storage means in accordance with the data manifestation stored in said address register; condition manifesting means for manifesting first and second conditions; and means responsive to said condition manifesting means for transmitting the manifestations of data comprehending the address of the next program step of said first sequence of program steps from said program register to said address register in response to the manifestation of the first condition and for transmitting data manifestations comprehending the first step of said alternate sequence of steps from said program register to said address register in response to the manifestation of the second condition, whereby an alternate group of program steps may be substituted for a predetermined group of program steps of a sequence.

45. Apparatus according to claim 44 wherein said program register to address register transmitting means comprises a bi-stable device adapted to be set in one or the other of two stable states.

46. Apparatus according to claim 45 further characterized by the provision of a calculator section and wherein said condition manifesting means is under control of said calculator.

References Cited in the file of this patent

UNITED STATES PATENTS

2,444,042	Hartley	June 29, 1948
2,500,294	Phelps	Mar. 14, 1950
2,580,768	Hamilton	Jan. 1, 1952
2,604,262	Phelps	July 22, 1952
2,634,052	Bloch	Apr. 7, 1953
2,657,856	Edwards	Nov. 3, 1953
2,679,638	Bensky	May 25, 1954
2,692,728	Strickler	Oct. 26, 1954
2,755,459	Carbrey	July 17, 1956
2,800,278	Thomas	July 23, 1957

FOREIGN PATENTS

737,863	Great Britain	Oct. 5, 1955

OTHER REFERENCES

EDVAC Progress Report #2; Moore School, U. of Pa.; declassified Feb. 13, 1947; page 1-4-6 and Fig. PY-0-216 relied on.

Twenty-four Digit Parallel Computer with Magnetic Drum Memory, Engineering Research Associates, Incorporated, 1949, two volumes, pages 10 to 11, 20 to 21, 31 to 33, 42 to 43 and Figs. 3.2-2, 3.4-2, 3.4-4, 3.5-1, and 3.5-2 are relevant.

(Other references on following page)

OTHER REFERENCES

Keister et al.: The Design of Switching Circuits, D. Van Nostrand Co., 1951, pages 280–299 relied on.

Pearcey: "The Functional Design of an Automatic Computer," Proc. of Conference on Automatic Computing Machines, University of Sydney, Australia, April 1952, pages 127–138, Figs. 1, 3 relied on.

Description of a Magnetic Drum Calculator, volume XXV of the Annals of the Computation Laboratory of Harvard University, Harvard University Press, 1952. (This contains the description of the Mark III, pages 22 to 23, 30 to 31 and 82 to 85 relevant.)

Greenwald et al.: "Seac," Proc. I.R.E., October 1953, pages 1300–1308 relied on.