

## CONTENTS

- A FILL T/P MEMORY WITH A CONSTANT
- B LOAD T/P TO TYPEWRITER
- C DUMP T/P MEMORY TO TYPEWRITER
- D PUNCH BI-4 PAPER TAPE
- E TEST T/P SOCC TRANSMISSION
- F TEST T/P EXCHANGE UNIT
- G JUMP TO OPEN THE MACHINE
- H GENERALIZED PARITY ANALYSIS ROUTINE
- I BI-4 PAPER TAPE LOADER WITH CHECKSUM  
BI-4 " " DUMP WITH CHECKSUM WITH CHECKSUM
- J TEST ROUTINE PHASE SWITCH AND DUMP METHODS
- K SOCC TRANSMISSION & ERROR DETECTION  
(NOT TESTED)

TITLE: FILL TELE PROGRAMMER BLOCK WITH A CONSTANT

STORAGE: 51 words

PURPOSE: To fill a number of 7/10 blocks with a constant

OPERATION: This program may be loaded into any block starting at location 000. The tag of this block should be entered into location 001 of this block. The program is started from location 000. At stop 003 the tag of the first block to be filled is entered in the A register. Run from this location; at stop 006 the tag of the last block is entered in the A register. Run from this location; at stop 011 enter the constant to be used in filling the memory. Run from this point at halt 51 filling is completed.

FILL TELEPROGRAMMER MEMORY WITH A CONSTA PAGE 001

0000	0000	ORG	LDN	0	
0000	0020				** SET PROGRAM EXECUTION TAG
0001	0000	PRGTAG		0	
0002	0102		ATT	T1	
0003	0000		ERR		** STORE FIRST BLOCK TAG
0004	0141		STM	T1	
0005	0015		FSTBLK		
0006	0000		ERR		** STORE LAST BLOCK TAG
0007	0141		STM	T1	
0010	0044		LSTBLK		
0011	0000		ERR		** STORE CONSTANT WHICH WILL FILL MEMORY BLOCKS
0012	0141		STM	T1	
0013	0024		CONST		
0014	0020		LDN		** SET TAG 3 TO FIRST BLOCK
0015	0000	FSTBLK		0	
0016	0302		ATT	T3	
0017	0020		LDN		** INITIALIZE ML AT 0
0020	0000			0	
0021	0141		STM	T1	
0022	0026		ML		
0023	0020		LDN		** STORE COMSTANT
0024	0000	CONST			
0025	0341		STM	T3	
0026	0000	ML			
0027	0155		RAO	T1	
0030	0026		ML		
0031	0161		NZP	T1	
0032	0023		ML	-3	
0033	0121		LDM	T1	** SORE CONSTANT IN ML 377
0034	0024		CONST		
0035	0341		STM	T3	
0036	0377			377	
0037	0303		TTA	T3	
0040	0030		ADN		
0041	0001			1	
0042	0302		ATT	T3	
0043	0034		SBN		** IF TAG 3 LESS THAN LAST BLOCK GO TO STORE CONSTANT
0044	0000	LSTBLK			
0045	0163		NJP	T1	
0046	0023		ML	-3	
0047	0160		ZJP	T1	** IF TAG 3 EQUAL LAST BLOCK GO TO STORE CONSTANT
0050	0023		ML	-3	
0051	0077		HLT		** END OF PROGRAM

0100	T1	EQU	100		
0200	T2	EQU	200		
0300	T3	EQU	300		
0000	ERR	EQU		0	ERROR STOP
0001	SHA	EQU		1	SHIFT A LEFT ONE BIT
0002	ATT	EQU		2	A TO TAG REGISTER
0003	TTA	EQU		3	TAG REGISTER CONTENTS TO A
0004	ABR	EQU		4	A TO BUFFER ENTRANCE REGISTER
0005	ABX	EQU		5	A TO BUFFER EXIST REGISTER
0006	BER	EQU		6	CONTENTS OF BER REGISTER TO A
0007	CBC	EQU		7	CLEAR BUFFER CONTROLS
0010	LPN	EQU		10	LOGICAL PRODUCT NO ADDRESS
0011	LPM	EQU		11	LOGICAL PRODUCT MEMORY ADDRESS
0012	LPI	EQU		12	LOGICAL PRODUCT INDIRECT ADDRESS
0013	CIL	EQU		13	CLEAR INTERRUPT LOCKOUT
0014	LSN	EQU		14	LOGICAL SUM NO ADDRESS
0015	LSM	EQU		15	LOGICAL SUM MEMORY ADDRESS
0016	LSI	EQU		16	LOGICAL SUM INDIRECT ADDRESS
0020	LDN	EQU		20	LOAD A NO ADDRESS MODE
0021	LDM	EQU		21	LOAD A MEMORY
0022	LDI	EQU		22	LOAD A INDIRECT
0025	LCM	EQU		25	LOAD COMPLEMENT TO A MEMORY
0026	LCI	EQU		26	LOAD COMPLEMENT TO A INDIRECT
0030	ADN	EQU		30	ADD NO ADDRESS
0031	ADM	EQU		31	ADD MEMORY ADDRESS
0032	ADI	EQU		32	ADD INDIRECT ADDRESS
0034	SBN	EQU		34	SUBTRACT NO ADDRESS
0035	SBM	EQU		35	SUBTRACT MEMORY ADDRESS
0036	SBI	EQU		36	SUBTRACT INDIRECT ADDRESS
0041	STM	EQU		41	STORE A MEMORY
0042	STI	EQU		42	STORE A INDIRECT
0051	RAM	EQU		51	REPLACE ADD MEMORY ADDRESS
0055	RAO	EQU		55	REPLACE ADD ONE MEMORY ADDRESS
0060	ZJB	EQU		60	JUMP, IF CONTENTS OF A = 0
0060	ZJP	EQU		60	JUMP, IF CONTENTS OF A = 0
0061	NZP	EQU		61	JUMP, IF CONTENTS OF A ≠ 0
0062	PJP	EQU		62	JUMP, IF CONTENTS OF A ≥ 0 POSITIVE
0063	NJP	EQU		63	JUMP, IF CONTENTS OF A ≤ 0 NEGATIVE
0064	UJP	EQU		64	UNCONDITIONAL JUMP
0070	IBI	EQU		70	INITIATE BUFFER INPUT
0071	IBO	EQU		71	INITIATE BUFFER OUTPUT
0072	INN	EQU		72	INPUT NORMAL
0073	OUT	EQU		73	OUTPUT NORMAL
0074	OTN	EQU		74	OUTPUT, NO ADDRESS
0075	EXF	EQU		75	EXTERNAL FUNCTION
0076	INA	EQU		76	INPUT TO A
0077	HLT	EQU		77	HALT
0000		END			COMPLETE ASSEMBLY

**OSAS/OSAS-A CODING FORM**

PAGE NO. 1  
DATE \_\_\_\_\_  
PROGRAMMER \_\_\_\_\_

PROGRAM \_\_\_\_\_  
ROUTINE \_\_\_\_\_

2	LOCATION	10	OP	15	ADDRESS	23	ADDITIVE	31	COMMENTS
									FILL THE FOLLOWING BLOCKS
			DRG				0000		
0			LDN						** SET PROGRAM EXECUTION TAG
1	PLG TAG		LDN						
2			ATT				T1		
3			ERR						** STORE FIRST BLOCK TAG
4			STM				T1		
5			F.S.T.B.L.K.						
6			ERR						** STORE LAST BLOCK TAG
7			STM				T1		
10			L.S.T.B.L.K.						
11			ERR						** STORE CONSTANT TO FILL BLOCKS
14			STM				T1		
17			CONST						
18			LDN						** SET TAG 3 TO FIRST BLOCK
19	F.S.T.B.L.K.		LDN				0		
20			ATT				T3		
21			LDN						** INITIALIZE ML AT 0
24			STM				T1		
26			ML						

PAGE \_\_\_\_\_

TITLE: LOAD TELEPROGRAMMER VIA TYPEWRITER

SYNOPSIS: 200 WPROS

OBJECTIVE: To input program instructions or data into specific T/P memory locations.

PROCEDURE: This load program may be loaded into any block of the T/P memory starting with location 001. The tag of the block into which it has been loaded should be entered in to location 001 of the block.

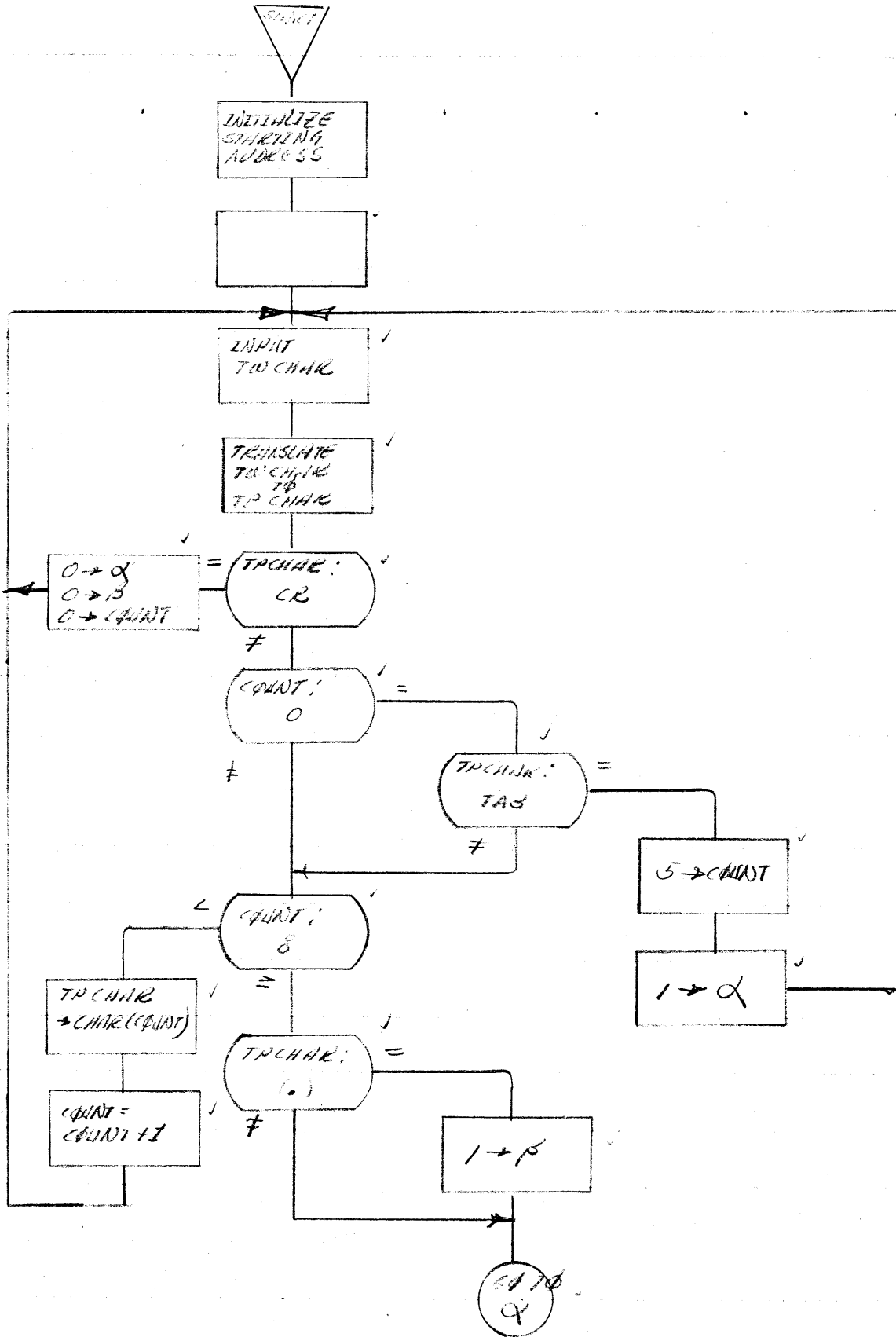
Loading via typewriter is accomplished as follows:

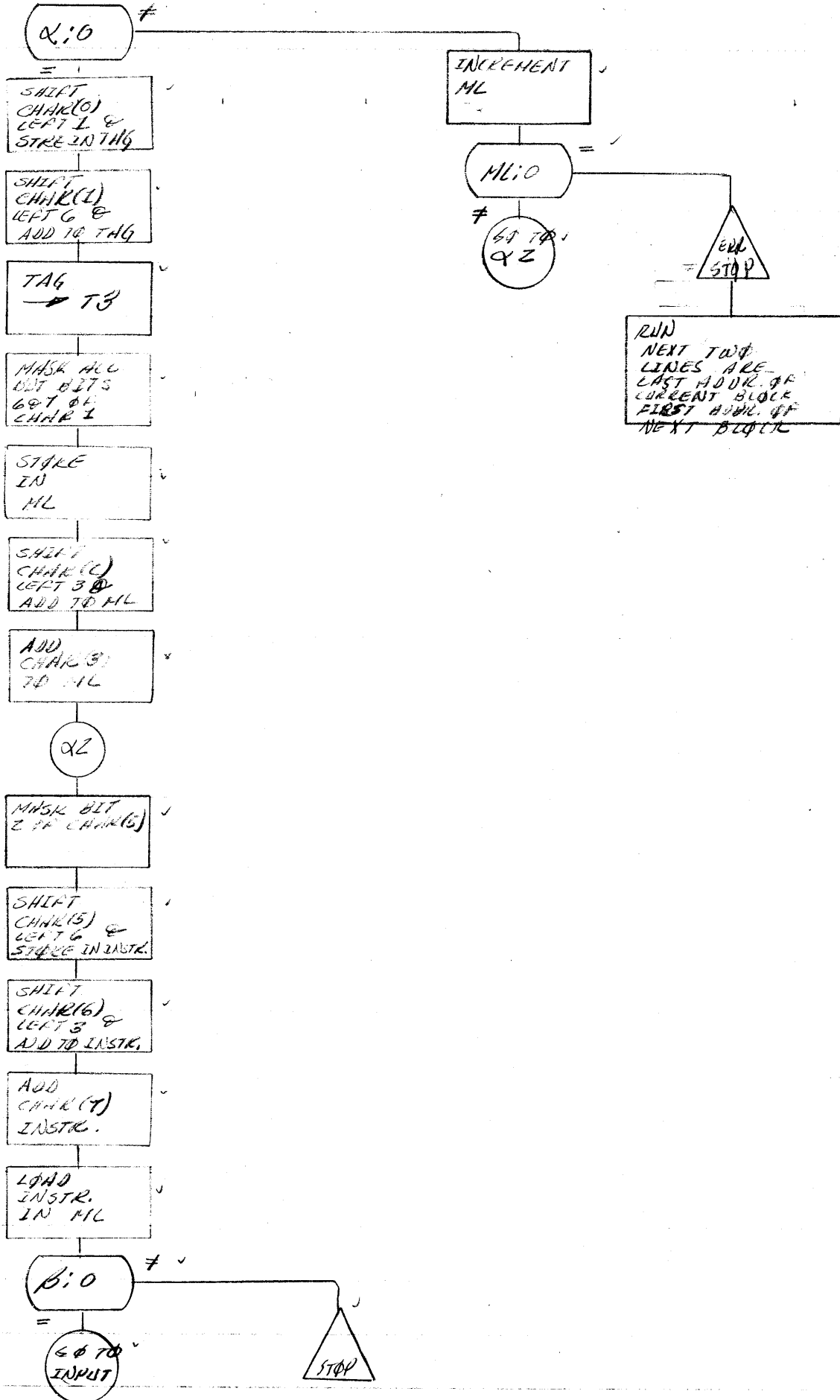
- A) set at least one tab on the typewriter
- B) the first character typed should be a carriage return
- C) the typewriter line should be:

MC	INSTK, OR DATA	
XXXX TAB XXX,		CARRIAGE RETURN

only actual numbers can be used in the MC and INSTK, OR DATA portions of the line. A character other than a period must follow INSTK OR DATA. Carriage return moves the INSTK, OR DATA in the M.L.

if it is desired to have INSTK, OR DATA entered in sequential locations only the first MC need be typed. a tab following a carriage return will load the following typed INSTK, OR DATA in the next M.L. when the next sequential location is word 317 of the current block an error stop will occur. from this stop the next two input lines must contain the MC of last address of current block and next block. This will load the last INSTK OR DATA and halt the program.





RUN  
NEXT TWO LINES ARE  
LAST ADDR. OF  
CURRENT BLOCK  
FIRST ADDR. OF  
NEXT BLOCK



TRANSITION BEDS ARE THE ONLY BEDS ARE  
 ALL OTHER TYPES OF BEDS ARE GREATER THAN 36

HEIGHT	BED THICK	BED THICK	HEIGHT	SHEDDING LIGHT DIST	ALBEDO
0	36	0	0	000 000	0
1	74	16	1	000 111	7
2	72	12	2	000 122	5
3	68	16	3	000 111	3
4	62	14	4	000 110	5
5	66	14	5	000 110	1
6	71	12	6	000 110	6
7	60	16	7	000 110	1

ALBEDO THE HEIGHTS ARE 1 TABLE

ALBEDO	HEIGHT
0	0
7	1
5	2
3	3
5	4
1	5
6	6
1	7



## LOAD TELEPROGRAMMER VIA TYPEWRITER

	0000		ORG		0	
	0000	0020	START	LDN		
	0001	0000	PRGTAG			TAG OF BLOCK WHERE THIS PROG IS LOADED
	0002	0102		ATT	TI	
	0003	0075	INPUT	EXF		** REQUEST T/W STATUS
0	0004	0042			42	
	0005	0040			40	
	0006	0076		INA		** STATUS NOT EQUAL ZERO
	0007	0161		NZP	TI	GO TO INPUT
	0010	0003		INPUT		
	0011	0075		EXF		** SELECT T/W INPUT
	0012	0042			42	
	0013	0020			20	
	0014	0076		INA		** INPUT T/W CHAR
	0015	0141		STM	TI	
	0016	0330		TWCHAR		
	0017	0034		SBN		** IF TW CHAR EQUAL . (42)
	0020	0042	PERIOD		42	GO TO DIRECT
	0021	0160		ZJP	TI	
	0022	0304		DIRECT		
	0023	0034		SBN		** IF TW CHAR EQUAL CR (45)
	0024	0003	CR		3	GO TO DIRECT
	0025	0160		ZJP	TI	
	0026	0304		DIRECT		
	0027	0034		SBN		** IF TW CHAR EQUAL TAB(51)
	0030	0004	TAB		4	
	0031	0160		ZJP	TI	
	0032	0304		DIRECT		
	0033	0034		SBN		** TW CHAR LESS THAN A
	0034	0005	NUMBER		5	NUMBER (56) GO TO ERROR
0	0035	0163		NJP	TI	
	0036	0312		ERROR		
	0037	0121		LDM	TI	** IF TW CHAR IS NOT EVEN
	0040	0330		TWCHAR		(INVALID CHAR) GO TO
	0041	0010		LPN		ERROR
	0042	0001			1	
	0043	0161		NZP	TI	
	0044	0312		ERROR		
	0045	0121		LDM	TI	** DETERMINE ARGUMENT FOR
	0046	0330		TWCHAR		TRANS. TABLE
	0047	0034		SBN		
	0050	0056			56	
	0051	0001		SHA		
	0052	0001		SHA		
	0053	0001		SHA		
	0054	0001		SHA		
	0055	0001		SHA		
	0056	0001		SHA		
	0057	0001		SHA		
	0060	0141		STM	TI	

0061	0067		ARGU		
0062	0020		LDN		
0063	0320		BASE		
0064	0151		RAM	TI	
0065	0067		ARGU		
0066	0121		LDM	TI	** TRANSLATE
0067	0000	ARGU			
0070	0141		STM	TI	
0071	0331		TPCHAR		
0072	0121	CRTST	LDM	TI	** IF TP CHAR NOT EQUAL CR(45) GO TO BEG TST
0073	0331		TPCHAR		
0074	0034		SBN		
0075	0045			45	
0076	0161		NZP	TI	
0077	0110		BEGTST		
0100	0141		STM	TI	** IF TP CHAR EQUAL CR(45) SET ALPHA, BETA AND COUNT TO ZERO, GO TO INPUT
0101	0343		ALPHA		
0102	0141		STM	TI	
0103	0344		BETA		
0104	0141		STM	TI	
0105	0345		COUNT		
0106	0164		UJP	TI	
0107	0003		INPUT		
0110	0121	BEGTST	LDM	TI	** IF COUNT EQUAL ZERO GO TO TABTST
0111	0345		COUNT		
0112	0160		ZJP	TI	
0113	0246		TABTST		
0114	0121	ENDTST	LDM	TI	** IF COUNT LESS THAN 8 GO TO STORE
0115	0345		COUNT		
0116	0034		SBN		
0117	0010			10	
0120	0163		NJP	TI	
0121	0264		STORE		
0122	0121	PRDTST	LDM	TI	** IF TP CHAR NOT EQUAL PERIOD (42) GO TO INPUT
0123	0331		TPCHAR		
0124	0034		SBN		
0125	0042			42	
0126	0161		NZP	TI	
0127	0132		ALPHA0		
0130	0155		RAO	TI	** SET BETA TO ONE
0131	0344		BETA		
0132	0121	ALPHA0	LDM	TI	** IF ALPHA NOT EQUAL ZERO GO TO ALPHA 3
0133	0343		ALPHA		
0134	0161		NZP	TI	
0135	0237		ALPHA3		
0136	0121	ALPHA1	LDM	TI	** SHIFT CHAR0 ONE BIT AND STORE IN TAG
0137	0332		CHAR0		
0140	0001		SHA		

0141	0141		STM	TI	
0142	0346		TAG		
0143	0121		LDM	TI	** SHIFT CHAR 1 SIX BITS AND
0144	0333		CHAR1		ADD TO TAG
0145	0001		SHA		
0146	0001		SHA		
0147	0001		SHA		
0150	0001		SHA		
0151	0001		SHA		
0152	0001		SHA		
0153	0151		RAM	TI	
0154	0346		TAG		
0155	0010		LPN		** STORE LOW ORDER TWO BITS
0156	0300			300	OF CHAR 1 IN HIGH ORDER
0157	0141		STM	TI	OF ML
0160	0231		ML		
0161	0121		LDM	TI	** SET TAG 3 TO TAG
0162	0346		TAG		
0163	0302		ATT	T3	
0164	0121		LDM	TI	** SHIFT CHAR 2 THREE BITS AND
0165	0334		CHAR2		ADD TO ML
0166	0001		SHA		
0167	0001		SHA		
0170	0001		SHA		
0171	0151		RAM	TI	
0172	0231		ML		
0173	0121		LDM	TI	** ADD CHAR 3 TO ML
0174	0335		CHAR3		
0175	0151		RAM	TI	
0176	0231		ML		
0177	0121	ALPHA2	LDM	TI	** SHIFT TWO LOW ORDER BITS
0200	0337		CHAR5		OF CHAR 5 TO HIGH ORDER
0201	0010		LPN		AND STORE IN INSTR
0202	0003			3	
0203	0001		SHA		
0204	0001		SHA		
0205	0001		SHA		
0206	0001		SHA		
0207	0001		SHA		
0210	0001		SHA		
0211	0141		STM	TI	
0212	0227		INSTR		
0213	0121		LDM	TI	** SHIFT CHAR 6 THREE BITS AND
0214	0340		CHAR6		ADD TO INSTR
0215	0001		SHA		
0216	0001		SHA		
0217	0001		SHA		
0220	0151		RAM	TI	

0221	0227		INSTR		
0222	0121		LDM	TI	** ADD CHAR 7 TO INSTR
0223	0341		CHAR7		
0224	0151		RAM	TI	
0225	0227		INSTR		
0226	0020		LDN		** STORE INSTR IN ML
0227	0000	INSTR			
0230	0341		STM	T3	
0231	0000	ML			
0232	0121		LDM	TI	** IF BETA EQUAL ZERO
0233	0344		BETA		GO TO INPUT
0234	0160		ZJP	TI	
0235	0003		INPUT		
0236	0077		HLT		** STOP
0237	0155	ALPHA3	RAO	TI	** INCREMENT ML, IF NOT
0240	0231		ML		ZERO GO TO ALPHA 2
0241	0161		NZP	TI	
0242	0177		ALPHA2		
0243	0000	GAMMA0	ERR		** STOP, GO TO INPUT.
0244	0164		UJP	TI	NEXT TWO INPUT LINES
0245	0003		INPUT		ARE FOR LAST ADDRESS OF
					CURRENT BLOCK AND
					FIRST ADDRESS OF NEXT BLOCK
0246	0121	TABTST	LDM	TI	** IF TP CHAR NOT EQUAL
0247	0331		TPCHAR		TAB(51) GO TO ENDTST
0250	0034		SBN		
0251	0051			51	
0252	0061		NZP		
0253	0114		ENDTST		
0254	0020		LDN		** IF TP CHAR EQUAL TAB (51)
0255	0005			5	SET COUNT TO 5 AND
0256	0141		STM	TI	ALPHA TO 1, GO TO INPUT
0257	0345		COUNT		
0260	0155		RAO	TI	
0261	0343		ALPHA		
0262	0164		UJP	TI	
0263	0003		INPUT		
0264	0020	STORE	LDN		** INITIALIZE POSITION LOCATION
0265	0332		CHAR0		
0266	0141		STM	TI	
0267	0277		POS		
0270	0121		LDM	TI	** INCREMENT POSITION BY
0271	0345		COUNT		COUNT
0272	0151		RAM	TI	
0273	0277		POS		
0274	0121		LDM	TI	** STORE CHAR IN APPROPRIATE
0275	0331		TPCHAR		POSITION
0276	0141		STM	TI	

0277	0000	POS			
0300	0155		RAO	TI	** INCREMENT COUNT
0301	0345		COUNT		GO TO INPUT
0302	0164		UJP	TI	
0303	0003		INPUT		
0304	0121	DIRECT	LDM	TI	** SET TPCHAR EQUAL TO TWCHAR
0305	0330		TWCHAR		GO TO CRTST
0306	0141		STM	TI	
0307	0331		TPCHAR		
0310	0164		UJP	TI	
0311	0072		CRTST		
0312	0020	ERROR	LON		** ERROR IN TW CHAR SET
0313	0007			7	TP CHAR TO 7, GO TO
0314	0141		STM	TI	CRTST
0315	0331		TPCHAR		
0316	0164		UJP	TI	
0317	0114		ENDTST		
0320	0000	BASE		0	** TRANSLATION TABLE
0321	0007			7	
0322	0004			4	
0323	0003			3	
0324	0005			5	
0325	0002			2	
0326	0006			6	
0327	0001			1	

PROGRAM PARAMETERS  
 TYPEWRITER CHARACTER  
 TELEPROGRAMMER CHARACTER  
 TYPED CHARACTER 0  
 TYPED CHARACTER 1  
 TYPED CHARACTER 2  
 TYPED CHARACTER 3  
 TYPED CHARACTER 4  
 TYPED CHARACTER 5  
 TYPED CHARACTER 6  
 TYPED CHARACTER 7  
 TYPED CHARACTER 10  
 LINE FORMAT FLAG  
 END OF INPUT FLAG  
 COUNT OF INPUT CHARACTERS  
 ML TAG OF INPUTED INSTR.

0330	0000	TWCHAR			
0331	0000	TPCHAR			
0332	0000	CHAR0			
0333	0000	CHAR1			
0334	0000	CHAR2			
0335	0000	CHAR3			
0336	0000	CHAR4			
0337	0000	CHAR5			
0340	0000	CHAR6			
0341	0000	CHAR7			
0342	0000	CHAR10			
0343	0000	ALPHA			
0344	0000	BETA			
0345	0000	COUNT			
0346	0000	TAG			

0100	T1	EQU	100		
0200	T2	EQU	200		
0300	T3	EQU	300		
0000	ERR	EQU		0	ERROR STOP
0001	SHA	EQU		1	SHIFT A LEFT ONE BIT
0002	ATT	EQU		2	A TO TAG REGISTER
0003	TTA	EQU		3	TAG REGISTER CONTENTS TO A
0004	ABR	EQU		4	A TO BUFFER ENTRANCE REGISTER
0005	ABX	EQU		5	A TO BUFFER EXIST REGISTER
0006	BER	EQU		6	CONTENTS OF BER REGISTER TO A
0007	CBC	EQU		7	CLEAR BUFFER CONTROLS
0010	LPN	EQU		10	LOGICAL PRODUCT NO ADDRESS
0011	LPM	EQU		11	LOGICAL PRODUCT MEMORY ADDRESS
0012	LPT	EQU		12	LOGICAL PRODUCT INDIRECT ADDRESS
0013	CIL	EQU		13	CLEAR INTERRUPT LOCKOUT
0014	LSN	EQU		14	LOGICAL SUM NO ADDRESS
0015	LSM	EQU		15	LOGICAL SUM MEMORY ADDRESS
0016	LSI	EQU		16	LOGICAL SUM INDIRECT ADDRESS
0020	LDN	EQU		20	LOAD A NO ADDRESS MODE
0021	LDM	EQU		21	LOAD A MEMORY
0022	LDI	EQU		22	LOAD A INDIRECT
0025	LCM	EQU		25	LOAD COMPLEMENT TO A MEMORY
0026	LCI	EQU		26	LOAD COMPLEMENT TO A INDIRECT
0030	ADN	EQU		30	ADD NO ADDRESS
0031	ADM	EQU		31	ADD MEMORY ADDRESS
0032	ADI	EQU		32	ADD INDIRECT ADDRESS
0034	SBN	EQU		34	SUBTRACT NO ADDRESS
0035	SBM	EQU		35	SUBTRACT MEMORY ADDRESS
0036	SBI	EQU		36	SUBTRACT INDIRECT ADDRESS
0041	STM	EQU		41	STORE A MEMORY
0042	STI	EQU		42	STORE A INDIRECT
0051	RAM	EQU		51	REPLACE ADD MEMORY ADDRESS
0055	RAO	EQU		55	REPLACE ADD ONE MEMORY ADDRESS
0060	ZJB	EQU		60	JUMP, IF CONTENTS OF A = 0
0060	ZJP	EQU		60	JUMP, IF CONTENTS OF A ≠ 0
0061	NZP	EQU		61	JUMP, IF CONTENTS OF A ≠ 0
0062	PJP	EQU		62	JUMP, IF CONTENTS OF A 0 POSITIVE
0063	NJP	EQU		63	JUMP, IF CONTENTS OF A 0 NEGATIVE
0064	UJP	EQU		64	UNCONDITIONAL JUMP
0070	IBI	EQU		70	INITIATE BUFFER INPUT
0071	IBO	EQU		71	INITIATE BUFFER OUTPUT
0072	INN	EQU		72	INPUT NORMAL
0073	OUT	EQU		73	OUTPUT NORMAL
0074	OTN	EQU		74	OUTPUT, NO ADDRESS
0075	EXF	EQU		75	EXTERNAL FUNCTION
0076	INA	EQU		76	INPUT TO A
0077	HLT	EQU		77	HALT
0000		END			COMPLETE ASSEMBLY



TITLE: DUMP TELETYPEWRITER BLOCKS VIA TYPEWRITER

STORAGE: 209 WORDS

PURPOSE: To dump the contents of specific T/P memory locations.

OPERATION: This dump program may be loaded into any block of the T/P starting with location 000. The top of the block into which it has been loaded should be entered in the location set of the blocks.

Dump via typewriter is accomplished as follows:

A) set at least one tab on the typewriter.

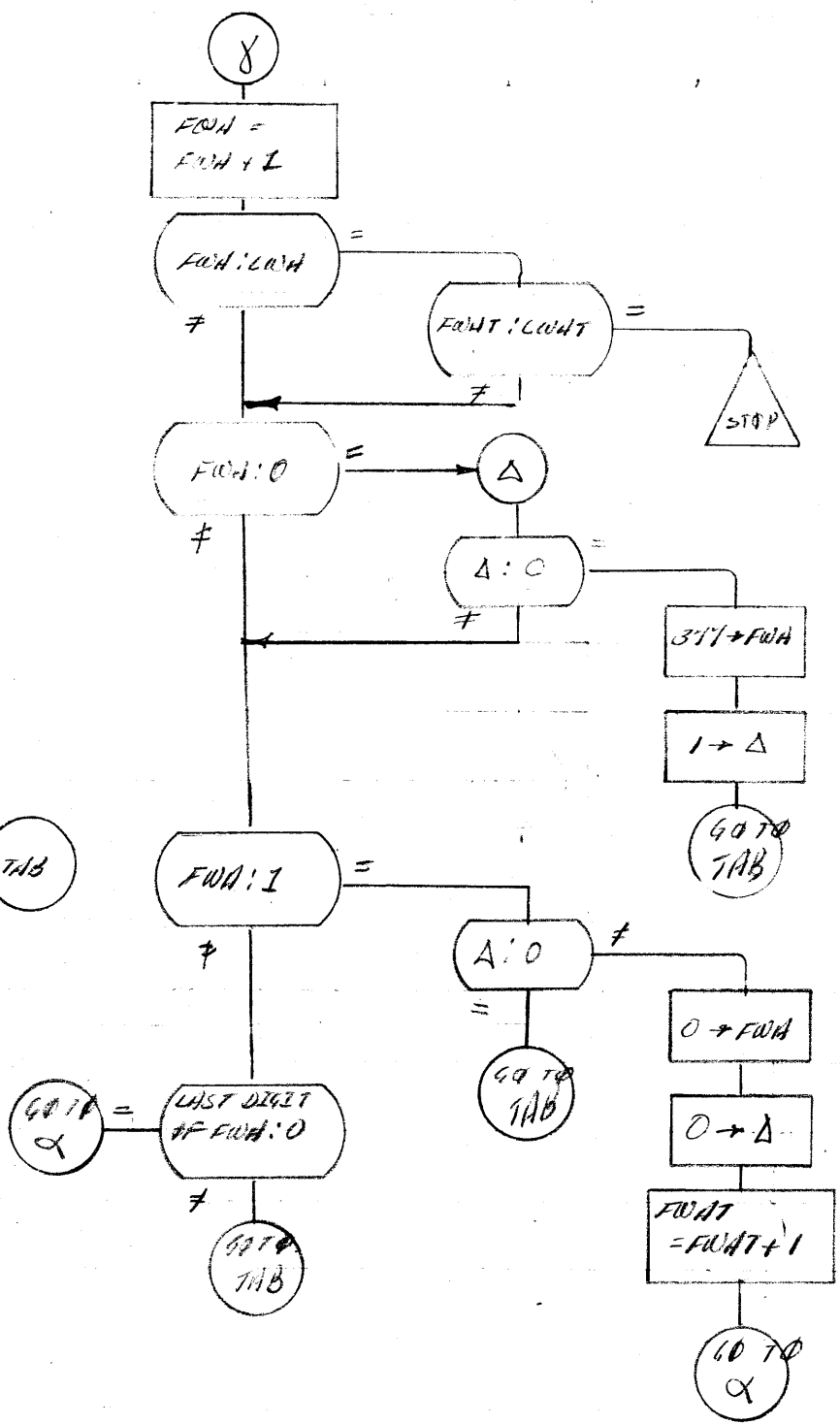
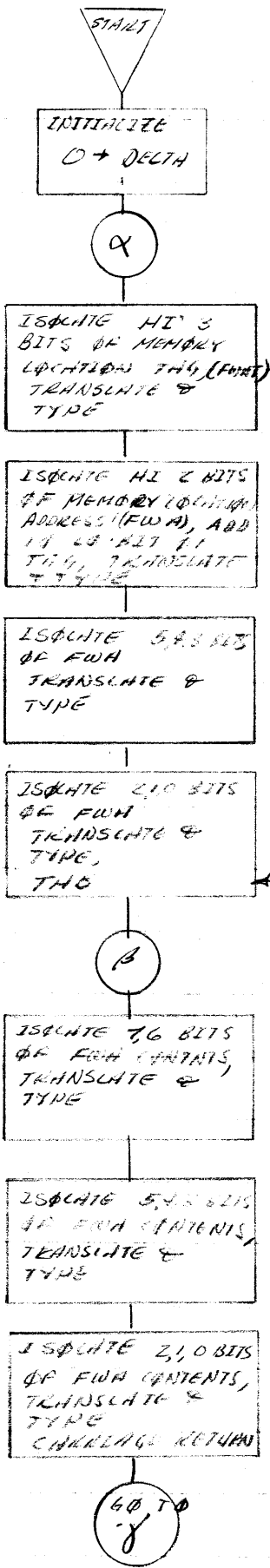
B) enter the range of the dump addresses into program block locations as follows:

first word tag      into  
first word address into  
last word tag      into  
last word address into

C) typed format will be:

ML                      COMMENTS  
XXXX                    XXX

the first ML and subsequent MLs which are multiples of 8 will be typed.



Address	Code	Label	ORG	Value	Comments
0000	0000			0	
0000	0020	START	LDN		
0001	0000	PRGTAG			TAG OF BLOCK WHERE THIS PROG IS LOADED
0002	0102		ATT	TI	
0003	0020		LDN		** INITIALIZE END OF BLOCK FLAG
0004	0000			0	
0005	0141		STM	TI	
0006	0354		DELTA		
0007	0020		LDN		** PERFORM T/W CR
0010	0045			45	
0011	0141		STM	TI	
0012	0333		OUTPUT		
0013	0020		LDN		
0014	0017		ALPHA1		
0015	0164		UJP	TI	
0016	0277		TATI		
0017	0121	ALPHA1	LDM	TI	** PLACE HI 3 BITS OF FIRST WORD ADDRESS TAG (FWAT) IN CHAR AND GO TO TRANSLATE AND TYPE (TAT)
0020	0350			FWAT	
0021	0001		SHA		
0022	0001		SHA		
0023	0001		SHA		
0024	0001		SHA		
0025	0001		SHA		
0026	0001		SHA		
0027	0001		SHA		
0030	0141		STM	TI	
0031	0347		CHAR		
0032	0020		LDN		
0033	0036		ALPHA2		
0034	0164		UJP	TI	
0035	0303		TAT		
0036	0121	ALPHA2	LDM	TI	** PLACE LO BIT OF FWAT IN HI BIT OF CHAR
0037	0347		CHAR		
0040	0001		SHA		
0041	0001		SHA		
0042	0001		SHA		
0043	0141		STM	TI	
0044	0347		CHAR		
0045	0121		LDM	TI	** ADD HI 2 BITS OF FIRST WORD ADDRESS (FWA) TO LO 2 BITS OF CHAR AND GO TO TAT
0046	0351		FWA		
0047	0001		SHA		
0050	0001		SHA		
0051	0141		STM	TI	
0052	0346		TEMP		
0053	0010		LPN		
0054	0003			3	
0055	0151		RAM	TI	
0056	0347		CHAR		

0057	0020		LDN		
0060	0063		ALPHA3		
0061	0164		UJP	T1	
0062	0303		TAT		
0063	0121	ALPHA3	LDM	T1	** PLACE 5, 4, 3 BITS OF FWA
0064	0346		TEMP		IN CHAR AND GO TO TAT
0065	0001		SHA		
0066	0001		SHA		
0067	0001		SHA		
0070	0141		STM	T1	
0071	0346		TEMP		
0072	0141		STM	T1	
0073	0347		CHAR		
0074	0020		LDN		
0075	0100		ALPHA4		
0076	0164		UJP	T1	
0077	0303		TAT		
0100	0121	ALPHA4	LDM	T1	** PLACE 2, 1, 0 BITS OF FWA
0101	0346		TEMP		IN CHAR AND GO TO TAT
0102	0001		SHA		
0103	0001		SHA		
0104	0001		SHA		
0105	0141		STM	T1	
0106	0347		CHAR		
0107	0020		LDN		
0110	0113		ALPHA5		
0111	0164		UJP	T1	
0112	0303		TAT		
0113	0020	ALPHA5	LDN		** PERFORM T/W TAB AND
0114	0051			S1	GO TO TAT
0115	0141		STM	T1	
0116	0333		OUTPUT		
0117	0020		LDN		
0120	0123		BETA1		
0121	0164		UJP	T1	
0122	0277		TAT1		
0123	0121	BETA1	LDM	T1	** PLACE 7 AND 6 BITS OF
0124	0350		FWAT		CONTENTS OF FWA IN
0125	0302		ATT	T3	LO BITS OF CHAR AND
0126	0020		LDN		GO TO TAT+6
0127	0150		BETA2		
0130	0141		STM	T1	
0131	0335		RETURN		
0132	0121		LDM	T1	
0133	0351		FWA		
0134	0141		STM	T1	
0135	0137		ML		
0136	0321		LDM	T3	

0137	0000	ML			
0140	0001		SHA		
0141	0001		SHA		
0142	0141		STM	TI	
0143	0347		CHAR		
0144	0010		LPN		
0145	0003			3	
0146	0164		UJP	TI	
0147	0311		TAT	+6	
0150	0121	BETA2	LDM	TI	** PLACE 5, 4, 3 BITS OF
0151	0347		CHAR		CONTENTS OF FWA IN CHAR
0152	0001		SHA		AND GO TO TAT
0153	0001		SHA		
0154	0001		SHA		
0155	0141		STM	TI	
0156	0347		CHAR		
0157	0020		LDN		
0160	0163		BETA3		
0161	0164		UJP	TI	
0162	0303		TAT		
0163	0121	BETA3	LDM	TI	** PLACE 2, 1, 0 BITS OF
0164	0347		CHAR		CONTENTS OF FWA IN CHAR
0165	0001		SHA		AND GO TO TAT
0166	0001		SHA		
0167	0001		SHA		
0170	0141		STM	TI	
0171	0347		CHAR		
0172	0020		LDN		
0173	0176		BETA4		
0174	0164		UJP	TI	
0175	0303		TAT		
0176	0020	BETA4	LDN		** PERFORM T/W CR
0177	0045			45	
0200	0141		STM	TI	
0201	0333		OUTPUT		
0202	0020		LDN		
0203	0206		GAMMA1		
0204	0164		UJP	TI	
0205	0277		TATI		
0206	0155	GAMMA1	RAO	TI	** INCREMENT FWA, IF FWA
0207	0351		FWA		NOT EQUAL LWA GO TO GAMMA 2
0210	0135		SBM	TI	
0211	0353		LWA		
0212	0161		NZP	TI	
0213	0223		GAMMA2		
0214	0121		LDM	TI	** IF FWA EQUAL LWA AND
0215	0350		FWAT		FWAT EQUAL LWAT STOP,
0216	0135		SBM	TI	IF NOT GO TO GAMMA 2

0217	0352		LWAT		
0220	0161		NZP	TI	
0221	0223		GAMMA2		
0222	0077		HLT		
0223	0121	GAMMA2	LDM	TI	** IF FWA EQUAL ZERO GO
0224	0351		FWA		TO DELTA 1
0225	0160		ZJP	TI	
0226	0245		DELTA1		
0227	0121	GAMMA3	LDM	TI	** IF FWA EQUAL 1 GO TO
0230	0351		FWA		DELTA2
0231	0034		SBN		
0232	0001			I	
0233	0160		ZJP	TI	
0234	0261		DELTA2		
0235	0121	GAMMA4	LDM	TI	** IF LAST DIGIT OF FWA
0236	0351		FWA		EQUAL 0 GO TO ALPHA1,
0237	0010		LPN		IF NOT GO TO ALPHAS
0240	0007			7	
0241	0160		ZJP	TI	
0242	0017		ALPHA1		
0243	0164		UJP	TI	
0244	0113		ALPHAS		
0245	0121	DELTA1	LDM	TI	** IF DELTA NOT EQUAL 0 GO
0246	0354		DELTA		TO GAMMA3, IF DELTA
0247	0161		NZP	TI	EQUAL 0 SET FWA EQUAL
0250	0227		GAMMA3		377 AND SET DELTA EQUAL 1
0251	0020		LDN		
0252	0377			377	
0253	0141		STM	TI	
0254	0351		FWA		
0255	0155		RAO	TI	
0256	0354		DELTA		
0257	0164		UJP	TI	
0260	0113		ALPHAS		
0261	0121	DELTA2	LDM	TI	** IF DELTA EQUAL 0 GO TO
0262	0354		DELTA		ALPHAS, IF DELTA NOT EQUAL
0263	0160		ZJP	TI	0 SET FWA TO 0, SET DELTA
0264	0113		ALPHAS		TO 0 AND INCREMENT FWAT
0265	0020		LDN		
0266	0000			0	
0267	0141		STM	TI	
0270	0351		FWA		
0271	0141		STM	TI	
0272	0354		DELTA		
0273	0155		RAO	TI	
0274	0350		FWAT		
0275	0164		UJP	TI	
0276	0017		ALPHA1		

0277	0141	TATI	STM	TI	
0300	0335		RETURN		
0301	0164		UJP	TI	
0302	0321		STATUS		
0303	0141	TAT	STM	TI	** STORE RETURN ADDRESS
0304	0335		RETURN		
0305	0121		LDM	TI	** MASK OFF ALL BUT BITS
0306	0347		CHAR		0, 1, 2 OF CHAR
0307	0010		LPN		
0310	0007			7	
0311	0030		ADN		** MAKE UP ARGUMENT FOR
0312	0336		BASE		TRANSLATION TABLE
0313	0141		STM	TI	
0314	0316		ARGU		
0315	0121		LDM	TI	** TRANSLATE AND SET UP TO
0316	0000	ARGU			TYPE CHAR
0317	0141		STM	TI	
0320	0333		OUTPUT		
0321	0075	STATUS	EXF		** REQUEST T/W STATUS
0322	0042			42	
0323	0040			40	
0324	0076		INA		** IF STATUS NOT ZERO
0325	0161		NZP	TI	GO TO STATUS
0326	0321		STATUS		
0327	0075		EXF		** SELECT T/W OUTPUT
0330	0042			42	
0331	0010			10	
0332	0074		OTN		** TYPE CHAR, GO TO
0333	0000	OUTPUT			RETURN
0334	0164		UJP	TI	
0335	0000	RETURN			
0336	0056	BASE		56	** TRANSLATION TABLE
0337	0074			74	
0340	0070			70	
0341	0064			64	
0342	0062			62	
0343	0066			66	
0344	0072			72	
0345	0060			60	
0346	0000	TEMP			PROGRAM PARAMETERS
0347	0000	CHAR			TEMPORARY STORAGE
0350	0000	FWAT			CHAR BEING TRANSLATED
0351	0000	FWA			FIRST WORD TAG
0352	0000	LWAT			FIRST WORD ADDRESS
0353	0000	LWA			LAST WORD TAG
0354	0000	DELTA			LAST WORD ADDRESS
					END OF BLOCK FLAG

0100	T1	EQU	100		
0200	T2	EQU	200		
0300	T3	EQU	300		
0000	ERR	EQU		0	ERROR STOP
0001	SHA	EQU		1	SHIFT A LEFT ONE BIT
0002	ATT	EQU		2	A TO TAG REGISTER
0003	TTA	EQU		3	TAG REGISTER CONTENTS TO A
0004	ABR	EQU		4	A TO BUFFER ENTRANCE REGISTER
0005	ABX	EQU		5	A TO BUFFER EXIST REGISTER
0006	BER	EQU		6	CONTENTS OF BER REGISTER TO A
0007	CBC	EQU		7	CLEAR BUFFER CONTROLS
0010	LPN	EQU		10	LOGICAL PRODUCT NO ADDRESS
0011	LPM	EQU		11	LOGICAL PRODUCT MEMORY ADDRESS
0012	LPI	EQU		12	LOGICAL PRODUCT INDIRECT ADDRESS
0013	CIL	EQU		13	CLEAR INTERRUPT LOCKOUT
0014	LSN	EQU		14	LOGICAL SUM NO ADDRESS
0015	LSM	EQU		15	LOGICAL SUM MEMORY ADDRESS
0016	LSI	EQU		16	LOGICAL SUM INDIRECT ADDRESS
0020	LDN	EQU		20	LOAD A NO ADDRESS MODE
0021	LDM	EQU		21	LOAD A MEMORY
0022	LOI	EQU		22	LOAD A INDIRECT
0025	LCM	EQU		25	LOAD COMPLEMENT TO A MEMORY
0026	LCI	EQU		26	LOAD COMPLEMENT TO A INDIRECT
0030	ADN	EQU		30	ADD NO ADDRESS
0031	ADM	EQU		31	ADD MEMORY ADDRESS
0032	ADI	EQU		32	ADD INDIRECT ADDRESS
0034	SBN	EQU		34	SUBTRACT NO ADDRESS
0035	SBM	EQU		35	SUBTRACT MEMORY ADDRESS
0036	SBI	EQU		36	SUBTRACT INDIRECT ADDRESS
0041	STM	EQU		41	STORE A MEMORY
0042	STI	EQU		42	STORE A INDIRECT
0051	RAM	EQU		51	REPLACE ADD MEMORY ADDRESS
0055	RAO	EQU		55	REPLACE ADD ONE MEMORY ADDRESS
0060	ZJB	EQU		60	JUMP, IF CONTENTS OF A = 0
0060	ZJP	EQU		60	JUMP, IF CONTENTS OF A ≠ 0
0061	NZP	EQU		61	JUMP, IF CONTENTS OF A ≠ 0
0062	PJP	EQU		62	JUMP, IF CONTENTS OF A 0 POSITIVE
0063	NJP	EQU		63	JUMP, IF CONTENTS OF A 0 NEGATIVE
0064	UJP	EQU		64	UNCONDITIONAL JUMP
0070	IBI	EQU		70	INITIATE BUFFER INPUT
0071	IBO	EQU		71	INITIATE BUFFER OUTPUT
0072	INN	EQU		72	INPUT NORMAL
0073	OUT	EQU		73	OUTPUT NORMAL
0074	OTN	EQU		74	OUTPUT, NO ADDRESS
0075	EXF	EQU		75	EXTERNAL FUNCTION
0076	INA	EQU		76	INPUT TO A
0077	HLT	EQU		77	HALT
0000	END				COMPLETE ASSEMBLY



TITLE: PUNCH BI-4 PAPER TAPE

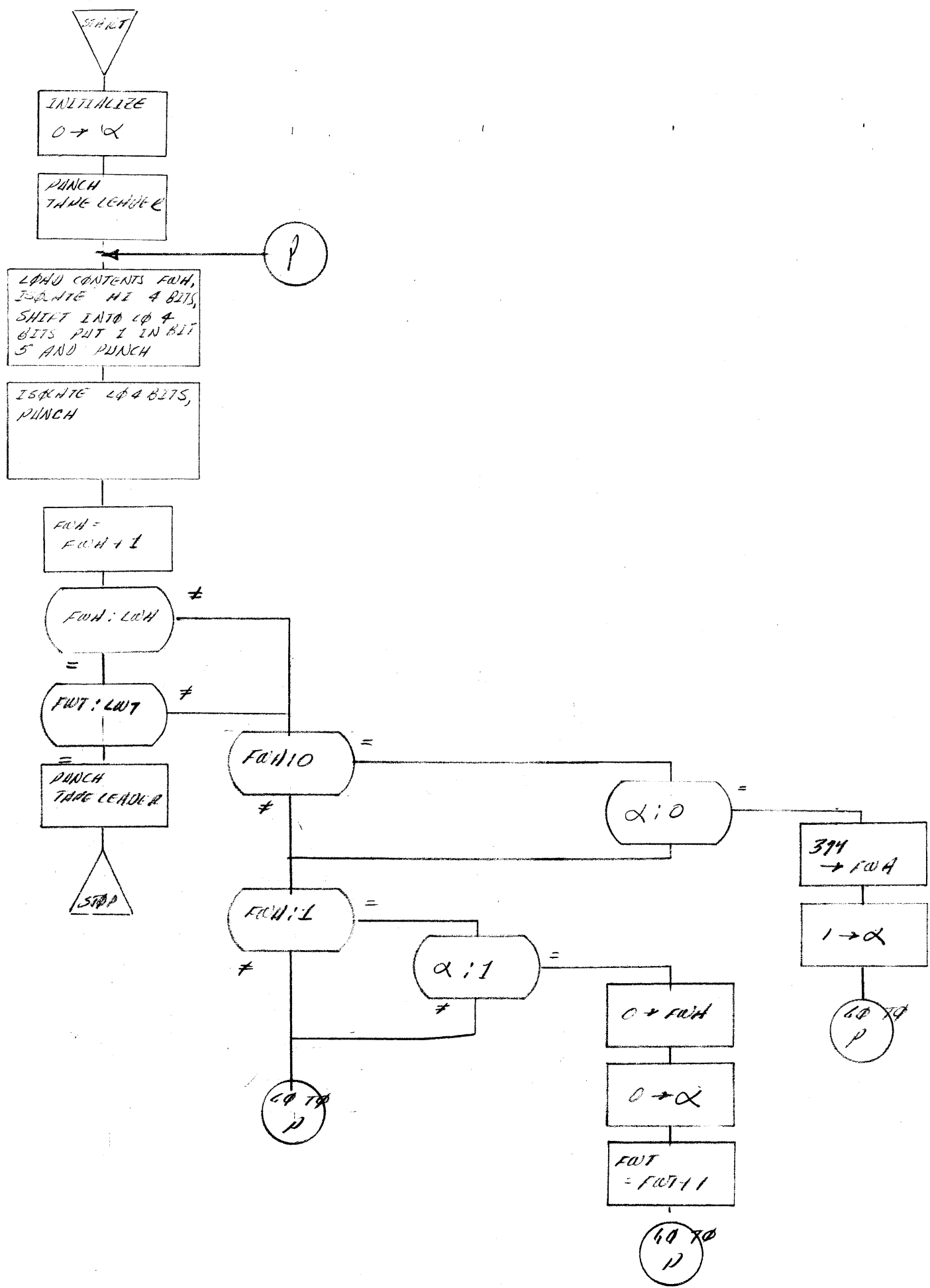
STORAGE: 101 WORDS

PURPOSE: To punch a paper tape in BI-4 format from specific memory locations. BI-4 format calls for splitting the 8 bit T/P word into 4 bits punched into sequential frames of a paper tape. The high order 4 bits are punched first with a one bit in the fifth level of the tape.

OPERATIONS: This program may be loaded into any block of the T/P starting with locations 000. The tag of the block into which it has been loaded should be entered into location 001.

The range of punch addresses is entered into program block locations as follows:

- first word tag into 164
- first word address into 165
- last word tag into 166
- last word address into 167



Address	Code	Label	Operation	Value	Comment
0000	0020		LDN	0	** SET PROGRAM EXECUTION TAG
0001	0000	PRGTAG			
0002	0102		ATT	T1	
0003	0075		EXF		** SELECT PAPER TAPE PUNCH
0004	0041			41	
0005	0004			4	
0006	0020		LDN		** INITIALIZE END OF BLOCK FLAG
0007	0000			0	
0010	0141		STM	T1	
0011	0170		ALPHA		
0012	0020		LDN		** PERFORM PUNCH LEADER
0013	0016		PUNCH	-3	
0014	0164		UJP	T1	
0015	0146		LEADER		
0016	0121		LDM	T1	** SET TAG 3 TO FIRST WORD TAG (FWT)
0017	0164		FWT		
0020	0302		ATT	T3	
0021	0121	PUNCH	LDM	T1	** CONTENTS OF FIRST WORD ADDRESS PLACED IN WORD
0022	0165		FWA		
0023	<del>7777</del>	D141 <i>CONNECTED TO BI-4 TAPE</i>	STM	T1	
0024	0026		ML		
0025	0321		LDM	T3	
0026	0000	ML			
0027	0141		STM	T1	
0030	0046		WORD		
0031	0010		LPN		** PUNCH HI 4 BITS OF WORD WITH 1 IN BIT 5
0032	0360			360	
0033	0001		SHA		
0034	0001		SHA		
0035	0001		SHA		
0036	0001		SHA		
0037	0030		ADN		
0040	0020			20	
0041	0141		STM	T1	
0042	0044		1STOUT		
0043	0074		OTN		
0044	0000	1STOUT			
0045	0020		LDN		** PUNCH LO 4 BITS OF WORD
0046	0000	WORD			
0047	0010		LPN		
0050	0017			17	
0051	0141		STM	T1	
0052	0054		2NDOUT		
0053	0074		OTN		
0054	0000	2NDOUT			
0055	0155		RAO	T1	** INCREMENT FWA; IF NOT EQUAL TO LWA GO TO PUNCH 1
0056	0165		FWA		

0057	0135		SBM	TI	
0060	0167		LWA		
0061	0161		NZP	TI	
0062	0076		PUNCH1		
0063	0121		LDM	TI	** IF INCREMENTED FWA EQUAL LWA
0064	0164		FWT		AND FWT EQUAL LWT STOP.
0065	0135		SBM	TI	IF INCREMENTED FWA EQUAL LWA
0066	0166		LWT		AND FWT NOT EQUAL LWT
0067	0161		NZP	TI	GO TO PUNCH 1
0070	0076		PUNCH1		
0071	0020		LDN		
0072	0075		STOP		
0073	0064			UJP	TI
0074	0146		LEADER		
0075	0077	STOP	HLT		
0076	0121	PUNCH1	LDM	TI	** IF FWA EQUAL 0 GO TO
0077	0165		FWA		ALPHA1
0100	0160		ZJP	TI	
0101	0112		ALPHA1		
0102	0121	PUNCH2	LDM	TI	** IF FWA EQUAL 1 GO TO
0103	0165		FWA		ALPHA2 OTHERWISE GO TO
0104	0034		SBN		PUNCH
0105	0001			I	
0106	0160		ZJP	TI	
0107	0126		ALPHA2		
0110	0164		UJP	TI	
0111	0021		PUNCH		
0112	0121	ALPHA1	LDM	TI	** IF ALPHA NOT EQUAL 0
0113	0170		ALPHA		GO TO PUNCH2, IF FWPHA
0114	0161		NZP	TI	EQUAL 0 SET FWA TO 377
0115	0102		PUNCH2		AND ALPHA TO 1
0116	0020		LDN		
0117	0377			377	
0120	0141		STM	TI	
0121	0165		FWA		
0122	0155		RAO	TI	
0123	0170		ALPHA		
0124	0164		UJP	TI	
0125	0021		PUNCH		
0126	0121	ALPHA2	LDM	TI	** IF ALPHA NOT EQUAL 0
0127	0170		ALPHA		GO TO PUNCH. IF ALPHA
0130	0034		SBN		EQUAL 0 SET FWA TO 0,
0131	0001			I	SET ALPHA TO 0 AND
0132	0161		NZP	TI	INCREMENT FWT
0133	0021		PUNCH		
0134	0020		LDN		
0135	0000			0	
0136	0141		STM	TI	

0137	0165		FWA		
0140	0141		STM	TI	
0141	0170		ALPHA		
0142	0155		RAO	TI	
0143	0164		FWT		
0144	0164		UJP	TI	
0145	0016		PUNCH	-3	
0146	0141	LEADER	STM	TI	** PUNCH LEADER
0147	0163		RETURN		
0150	0020		LDN		
0151	0233			233	
0152	0141		STM	TI	
0153	0171		COUNT		
0154	0074	BLANK	OTN		
0155	0000			0	
0156	0155		RAO	TI	
0157	0171		COUNT		
0160	0161		NZP	TI	
0161	0154		BLANK		
0162	0164		UJP	TI	
0163	0000	RETURN			

PROGRAM PARAMETERS  
 FIRST WORD TAG  
 FIRST WORD ADDRESS  
 LAST WORD TAG  
 LAST WORD ADDRESS  
 END OF BLOCK FLAG  
 LEADER FRAME COUNT

0164	0000	FWT		
0165	0000	FWA		
0166	0000	LWT		
0167	0000	LWA		
0170	0000	ALPHA		
0171	0000	COUNT		

0100	T1	EQU	100		
0200	T2	EQU	200		
0300	T3	EQU	300		
0000	ERR	EQU		0	ERROR STOP
0001	SHA	EQU		1	SHIFT A LEFT ONE BIT
0002	ATT	EQU		2	A TO TAG REGISTER
0003	TTA	EQU		3	TAG REGISTER CONTENTS TO A
0004	ABR	EQU		4	A TO BUFFER ENTRANCE REGISTER
0005	ABX	EQU		5	A TO BUFFER EXIST REGISTER
0006	BER	EQU		6	CONTENTS OF BER REGISTER TO A
0007	CBC	EQU		7	CLEAR BUFFER CONTROLS
0010	LPN	EQU		10	LOGICAL PRODUCT NO ADDRESS
0011	LPM	EQU		11	LOGICAL PRODUCT MEMORY ADDRESS
0012	LPI	EQU		12	LOGICAL PRODUCT INDIRECT ADDRESS
0013	CIL	EQU		13	CLEAR INTERRUPT LOCKOUT
0014	LSN	EQU		14	LOGICAL SUM NO ADDRESS
0015	LSM	EQU		15	LOGICAL SUM MEMORY ADDRESS
0016	LSI	EQU		16	LOGICAL SUM INDIRECT ADDRESS
0020	LDN	EQU		20	LOAD A NO ADDRESS MODE
0021	LDM	EQU		21	LOAD A MEMORY
0022	LDI	EQU		22	LOAD A INDIRECT
0025	LCM	EQU		25	LOAD COMPLEMENT TO A MEMORY
0026	LCI	EQU		26	LOAD COMPLEMENT TO A INDIRECT
0030	ADN	EQU		30	ADD NO ADDRESS
0031	ADM	EQU		31	ADD MEMORY ADDRESS
0032	ADI	EQU		32	ADD INDIRECT ADDRESS
0034	SBN	EQU		34	SUBTRACT NO ADDRESS
0035	SBM	EQU		35	SUBTRACT MEMORY ADDRESS
0036	SBI	EQU		36	SUBTRACT INDIRECT ADDRESS
0041	STM	EQU		41	STORE A MEMORY
0042	STI	EQU		42	STORE A INDIRECT
0051	RAM	EQU		51	REPLACE ADD MEMORY ADDRESS
0055	RAO	EQU		55	REPLACE ADD ONE MEMORY ADDRESS
0060	ZJB	EQU		60	JUMP, IF CONTENTS OF A = 0
0060	ZJP	EQU		60	JUMP, IF CONTENTS OF A = 0
0061	NZP	EQU		61	JUMP, IF CONTENTS OF A ≠ 0
0062	PJP	EQU		62	JUMP, IF CONTENTS OF A 0 POSITIVE
0063	NJP	EQU		63	JUMP, IF CONTENTS OF A 0 NEGATIVE
0064	UJP	EQU		64	UNCONDITIONAL JUMP
0070	IBI	EQU		70	INITIATE BUFFER INPUT
0071	IBO	EQU		71	INITIATE BUFFER OUTPUT
0072	INN	EQU		72	INPUT NORMAL
0073	OUT	EQU		73	OUTPUT NORMAL
0074	OTN	EQU		74	OUTPUT, NO ADDRESS
0075	EXF	EQU		75	EXTERNAL FUNCTION
0076	INA	EQU		76	INPUT TO A
0077	HLT	EQU		77	HALT
0000		END			COMPLETE ASSEMBLY

TITLE: 7857 8062 T/P TRANS+25524NS

STORAGE: House keeping 260, Interrupts & Idle 18,  
Buffer 49, Normal 23, Character 34

PURPOSE: To test the functioning of 8062 DEC's in  
transmission of data between T/P's.

OPERATION:

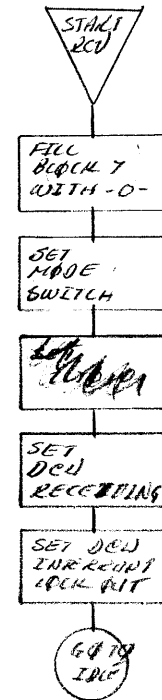
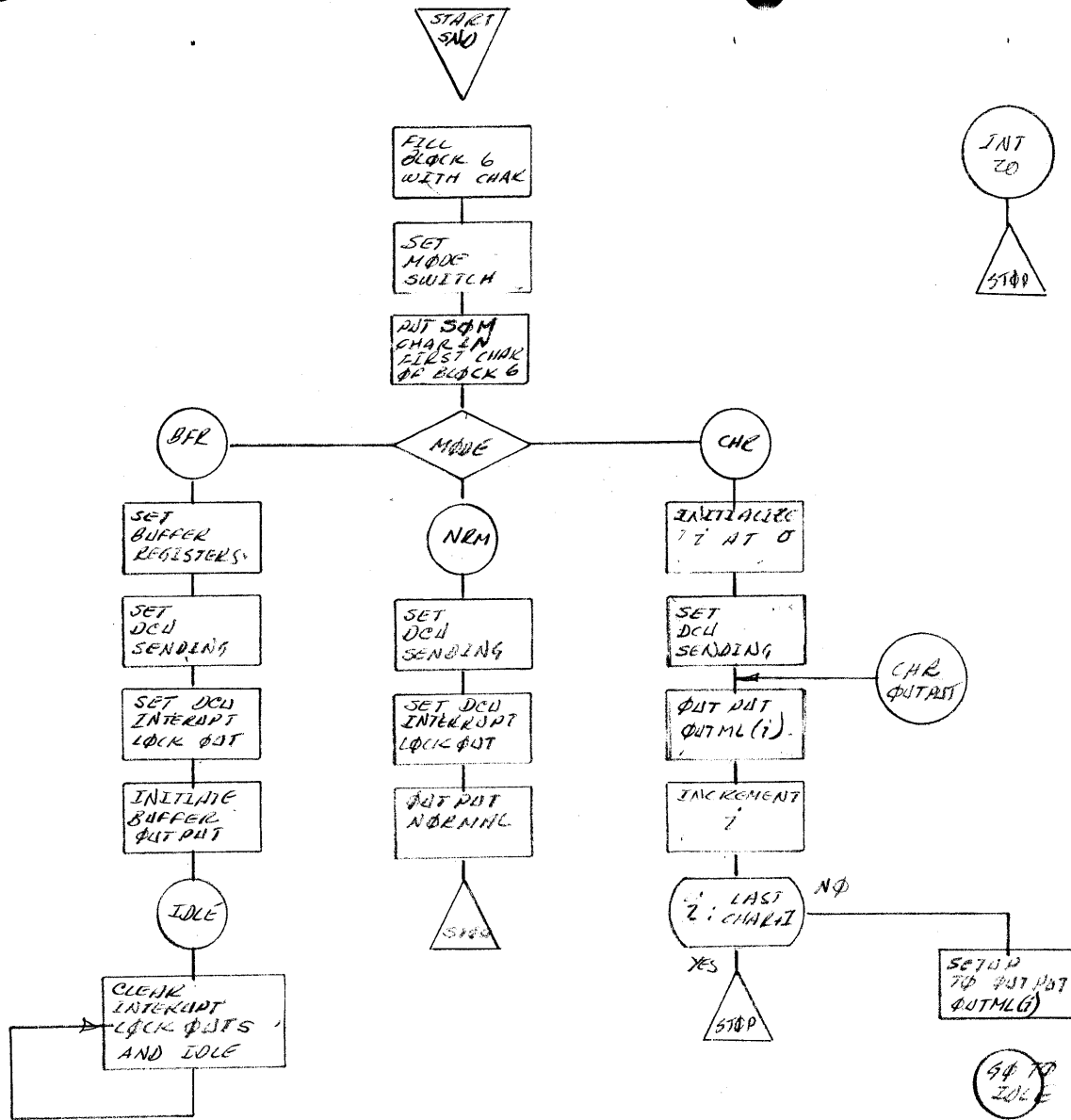
- A) LOADING - Load the program into both T/P's starting  
in location 8001.
- B) RUNNING - The mode of transmission is entered into  
both T/P's at location 8044. The mode  
codes are:  
100 - Buffer mode  
200 - Normal mode  
300 - Character mode

The character to be transmitted is entered  
into location 110. This character is placed  
into location 8001 thru 8376 of the send  
computer by the program.  
One of the T/P's is chosen to be a receiver  
and started from 8380. The other T/P  
is the sender and started from  
location 8100. The character will be  
transmitted and placed in locations 8401  
thru 8776 of the receiving T/P.

Transmission has been accomplished when  
both T/P's stop at

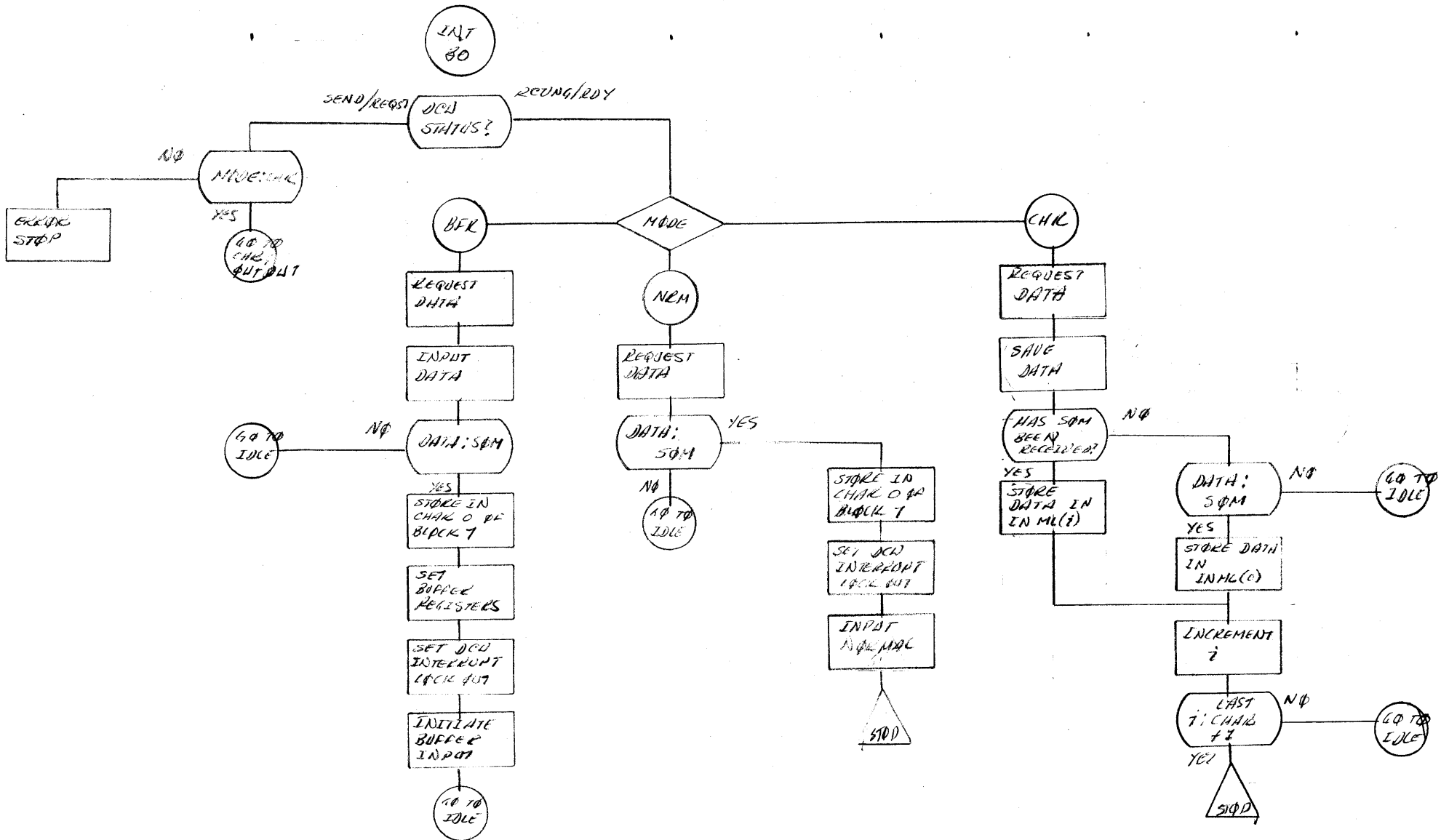
- 8001 - buffer
- 8002 - normal
- 8003 - character

8001 - 8376 of send and 8401 - 8776 of receive  
are filled with character 8000 & 8400  
hold mode codes.



GO TO IDLE





0001	0001	ORG	1	
0001	0077	HLT		** END OF BUFFER TRANSMISSION
0002	0077	HLT		** END OF NORMAL TRANSMISSION
0003	0077	HLT		** END OF CHARACTER TRANSMISSION
	0020	ORG	20	INSTRUCTIONS AT INTERRUPTS
0020	0064	UJP		
0021	0001		1	
	0030	ORG	30	
0030	0064	UJP		
0031	0206	INT30		

0077	0077		ORG	77	SEND ROUTINE GENERAL
0077	0000	MODE			MODE OF TRANSMISSION
0100	0020	SND	LDN		** SET TAG3 TO 6
0101	0006			6	
0102	0302		ATT	T3	
0103	0020		LDN		** PUT A CHARACTER IN ML
0104	0001			1	3001 THRU 3376
0105	0041		STM		
0106	0112		SFILL		
0107	0020		LDN		
0110	0000	CHAR			
0111	0341		STM	T3	
0112	0000	SFILL			
0113	0055		RAO		
0114	0112		SFILL		
0115	0061		NZP		
0116	0107		CHAR	-1	
0117	0021		LDM		** PUT MODE IN ML 3000
0120	0077		MODE		
0121	0341		STM	T3	
0122	0000			0	
0123	0021		LDM		** GO TO APPROPRIATE SEND
0124	0077		MODE		MODE ROUTINE
0125	0001		SHA		IF T1 = 1 BUFFER
0126	0001		SHA		IF T1 = 2 NORMAL
0127	0102		ATT	T1	IF T1 = 3 CHARACTER
0130	0164		UJP	T1	
0131	0000			0	

	0150	RCV	ORG	LDN	150	RECEIVE ROUTINE GENERAL
0150	0020			LDN		** SET TAG 3 TO 7
0151	0007				7	
0152	0302			ATT	T3	
0153	0020			LDN		** PUT 0 IN ML
0154	0000				0	3400 THRU 3776
0155	0041			STM		
0156	0162			RFILL		
0157	0020			LDN		
0160	0000	ZERO			0	
0161	0341			STM	T3	
0162	0000	RFILL				
0163	0055			RAO		
0164	0162			RFILL		
0165	0061			NZP		
0166	0157			ZERO	-1	
0167	0021			LDM		** SET TAG 1 FOR APPROPROATE
0170	0077			MODE		RECEIVE MODE ROUTINE
0171	0001			SHA		IF T2 = 1 BUFFER
0172	0001			SHA		IF T2 = 2 NORMAL
0173	0202			ATT	T2	IF T2 = 3 CHARACTER
0174	0075			EXF		** SET DCU RECEIVING
0175	0036				36	
0176	0000				0	
0177	0075	IDLE		EXF		** CLEAR DCU AND T/P INTERRUPTS
0200	0036				36	AND IDLE
0201	0005				5	
0202	0013			CIL		
0203	0002			ATT		
0204	0064			UJP		
0205	0204			IDLE	+5	
0206	0075	INT30		EXF		** REQUEST DCU STATUS
0207	0036				36	
0210	0003				3	
0211	0076			INA		** IF STATUS IS SENDING/ REQUEST
0212	0034			SBN		GO TO OUTCHR MINUS 1
0213	0005				5	
0214	0160			ZJP	T1	
0215	1407			OUTCHR	-5	
0216	0034			SBN		** IF STATUS IS RECEIVING/READY
0217	0005				5	GO TO APPROPRIATE
0220	0260			ZJP	T2	RECEIVE ROUTINE
0221	0150				150	
0222	0000			ERR		** ERROR STOP ON STATUS

0400	0400	ORG	400	BUFFER SEND
0400	0020	SNDBFR	LDN	** SET BUFFER ENTRANCE
0401	0000		0	REGISTER TO 3000
0402	0104		ABR T1	
0403	0400		SNDBFR	
0404	0020		LDN	** SET BUFFER EXIT
0405	0376		376	REGISTER TO 3376
0406	0105		ABX T1	
0407	0404		SNDBFR +4	
0410	0075		EXF	** SET DCU SENDING
0411	0036		36	
0412	0001		1	
0413	0075		EXF	** SET DCU INTERRUPT L/O
0414	0036		36	
0415	0004		4	
0416	0171	OUTBFR	IBO T1	** INITIATE BUFFER OUTPUT
0417	0416		OUTBFR	AND GO TO IDLE
0420	0064		UJP	
0421	0202		IDLE +3	
0550	0550	ORG	550	BUFFER RECEIVE
0550	0075	RCVBFR	EXF	** REQUEST DATA
0551	0036		36	
0552	0002		2	
0553	0076		INA	** IF DATA NOT EQUAL MODE
0554	0341		STM T3	(SOM CODE) GO TO IDLE
0555	0000		0	
0556	0035		SBM	
0557	0077		MODE	
0560	0061		NZP	
0561	0177		IDLE	
0562	0020		LDN	** SET BUFFER ENTRANCE
0563	0001		1	REGISTER TO 3401
0564	0204		ABR T2	
0565	0562		RCVBFR +12	
0566	0020		LDN	** SET BUFFER EXIT
0567	0377		377	REGISTER TO 3776
0570	0205		ABX T2	
0571	0566		RCVBFR +16	
0572	0075		EXF	** SET DCU INTERRUPT L/O
0573	0036		36	
0574	0004		4	
0575	0270	INBFR	IBI T2	** INITIATE BUFFER INPUT
0576	0575		INBFR	AND GO TO IDLE
0577	0064		UJP	
0600	0202		IDLE +3	

1000	0075	SNDNRM	ORG	EXF	1000	NORMAL SEND
						** SET DCU SENDING
1001	0036				36	
1002	0001				1	
1003	0075			EXF		** SET DCU INTERRUPT L/O
1004	0036				36	
1005	0004				4	
1006	0373			OUT	T3	** NORMAL OUTPUT
1007	0000				0	
1010	0376				376	
1011	0064			UJP		** GO TO END OF NORMAL TRANS.
1012	0002				2	
	1150		ORG			
1150	0075	RCVNRM		EXF	1150	NORMAL RECEIVE
						** REQUEST DATA
1151	0036				36	
1152	0002				2	
1153	0076			INA		** IF DATA NOT EQUAL MODE
1154	0341			STM	T3	(SOM CODE) GO TO IDLE
1155	0000				0	
1156	0035			SBM		
1157	0077			MODE		
1160	0061			NZP		
1161	0177			IDLE		
1162	0075			EXF		** SET DCU INTERRUPT L/O
1163	0036				36	
1164	0004				4	
1165	0372			INN	T3	** NORMAL INPUT
1166	0001				1	
1167	0376				376	
1170	0064			UJP		** GO TO END OF NORMAL TRANS.
1171	0002				2	

1400	1400	ORG	1400	CHARACTER SEND
1400	0020	SNDCHR	LDM	** INITIALIZE OUTMLI AT 0
1401	0000		0	
1402	0141		STM T1	
1403	1410		OUTMLI	
1404	0075		EXF	** SET DCU SENDING
1405	0036		36	
1406	0001		1	
1407	0321		LDM T3	** CHARACTER OUTPUT
1410	0000	OUTMLI		
1411	0141		STM T1	
1412	1414		OUTCHR	
1413	0074		OTN	
1414	0000	OUTCHR		
1415	0155		RAO T1	** INCREMENT OUTMLI, IF 0
1416	1410		OUTMLI	GO TO END OF CHARACTER TRANS.
1417	0060		ZJP	
1420	0003		3	
1421	0064		UJP	
1422	0177		IDLE	
1550	1550	ORG	1550	CHARACTER RECEIVE
1550	0075	RCVCHR	EXF	** REQUEST DATA
1551	0036		36	
1552	0002		2	
1553	0076		INA	** STORE DATA
1554	0241		STM T2	
1555	1610		DATA	
1556	0321		LDM T3	** IF SOM CODE HAS NOT
1557	0000		0	BEEN RECEIVED GOT
1560	0035		SBM	TST SOM
1561	0077		MODE	
1562	0261		NZP T2	
1563	1576		TSTSOM	
1564	0221		LDM T2	** IF SOM CODE HAS BEEN
1565	1610		DATA	RECEIVED, STORE DATA
1566	0341		STM T3	IN APPROPRIATE ML
1567	0000	INMLI		
1570	0255		RAO T2	
1571	1567		INMLI	
1572	0061		NZP	
1573	0177		IDLE	
1574	0064		UJP	** GO TO END OF CHARACTER TRANS.
1575	0003		3	
1576	0221	TSTSOM	LDM T2	** IF DATA NOT EQUAL MODE
1577	1610		DATA	(SOM CODE) GO TO IDLE
1600	0341		STM T3	
1601	0000		0	
1602	0035		SBM	

1603 0077

1604 0061

1605 0177

1606 0264

1607 1570

1610 0000 DATA

MODE

NZP

IDLE

UJP T2

INMLI +1

\*\* IF DATA EQUAL MODE

GO TO INMLI PLUS 1

TEMP. INPUT DATA STORAGE



0100	T1	EQU	100		
0200	T2	EQU	200		
0300	T3	EQU	300		
0000	ERR	EQU		0	ERROR STOP
0001	SHA	EQU		1	SHIFT A LEFT ONE BIT
0002	ATT	EQU		2	A TO TAG REGISTER
0003	TTA	EQU		3	TAG REGISTER CONTENTS TO A
0004	ABR	EQU		4	A TO BUFFER ENTRANCE REGISTER
0005	ABX	EQU		5	A TO BUFFER EXIST REGISTER
0006	BER	EQU		6	CONTENTS OF BER REGISTER TO A
0007	CBC	EQU		7	CLEAR BUFFER CONTROLS
0010	LPN	EQU		10	LOGICAL PRODUCT NO ADDRESS
0011	LPM	EQU		11	LOGICAL PRODUCT MEMORY ADDRESS
0012	LPI	EQU		12	LOGICAL PRODUCT INDIRECT ADDRESS
0013	CIL	EQU		13	CLEAR INTERRUPT LOCKOUT
0014	LSN	EQU		14	LOGICAL SUM NO ADDRESS
0015	LSM	EQU		15	LOGICAL SUM MEMORY ADDRESS
0016	LSI	EQU		16	LOGICAL SUM INDIRECT ADDRESS
0020	LON	EQU		20	LOAD A NO ADDRESS MODE
0021	LDM	EQU		21	LOAD A MEMORY
0022	LDI	EQU		22	LOAD A INDIRECT
0025	LCM	EQU		25	LOAD COMPLEMENT TO A MEMORY
0026	LCI	EQU		26	LOAD COMPLEMENT TO A INDIRECT
0030	ADN	EQU		30	ADD NO ADDRESS
0031	ADM	EQU		31	ADD MEMORY ADDRESS
0032	ADI	EQU		32	ADD INDIRECT ADDRESS
0034	SBN	EQU		34	SUBTRACT NO ADDRESS
0035	SBM	EQU		35	SUBTRACT MEMORY ADDRESS
0036	SBI	EQU		36	SUBTRACT INDIRECT ADDRESS
0041	STM	EQU		41	STORE A MEMORY
0042	STI	EQU		42	STORE A INDIRECT
0051	RAM	EQU		51	REPLACE ADD MEMORY ADDRESS
0055	RAO	EQU		55	REPLACE ADD ONE MEMORY ADDRESS
0060	ZJB	EQU		60	JUMP, IF CONTENTS OF A = 0
0060	ZJP	EQU		60	JUMP, IF CONTENTS OF A ≠ 0
0061	NZP	EQU		61	JUMP, IF CONTENTS OF A ≠ 0
0062	PJP	EQU		62	JUMP, IF CONTENTS OF A 0 POSITIVE
0063	NJP	EQU		63	JUMP, IF CONTENTS OF A 0 NEGATIVE
0064	UJP	EQU		64	UNCONDITIONAL JUMP
0070	IBI	EQU		70	INITIATE BUFFER INPUT
0071	IBO	EQU		71	INITIATE BUFFER OUTPUT
0072	INN	EQU		72	INPUT NORMAL
0073	OUT	EQU		73	OUTPUT NORMAL
0074	OTN	EQU		74	OUTPUT, NO ADDRESS
0075	EXF	EQU		75	EXTERNAL FUNCTION
0076	INA	EQU		76	INPUT TO A
0077	HLT	EQU		77	HALT
0000		END			COMPLETE ASSEMBLY

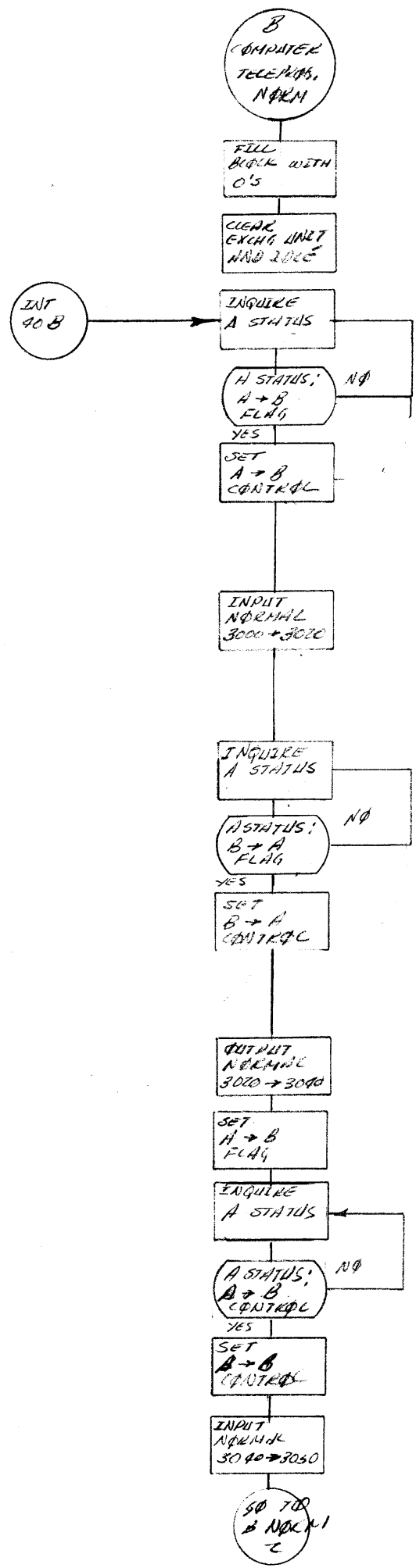
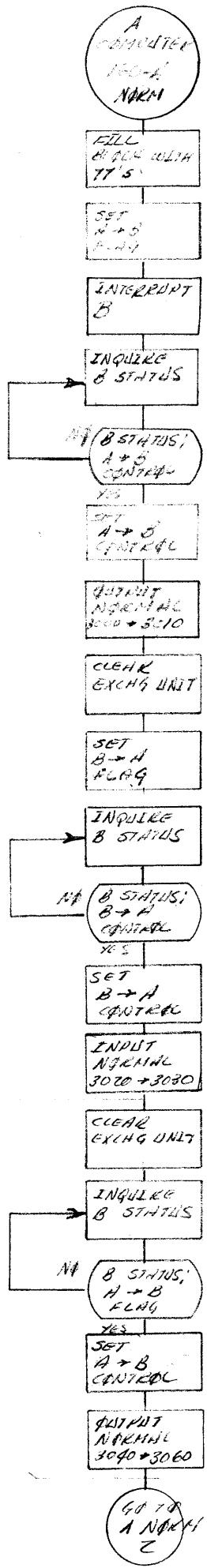
TITLE: TEST EXCHANGE UNIT OPERATION

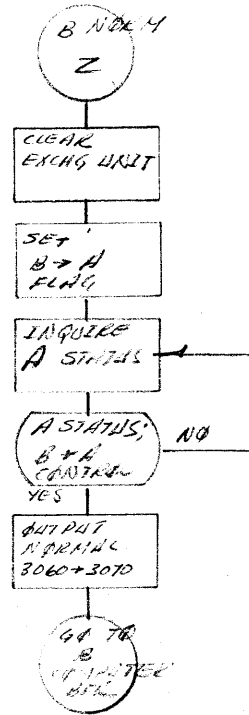
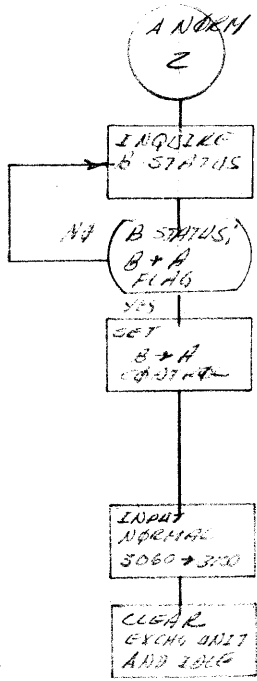
STORAGE: 160-A words, T/P words

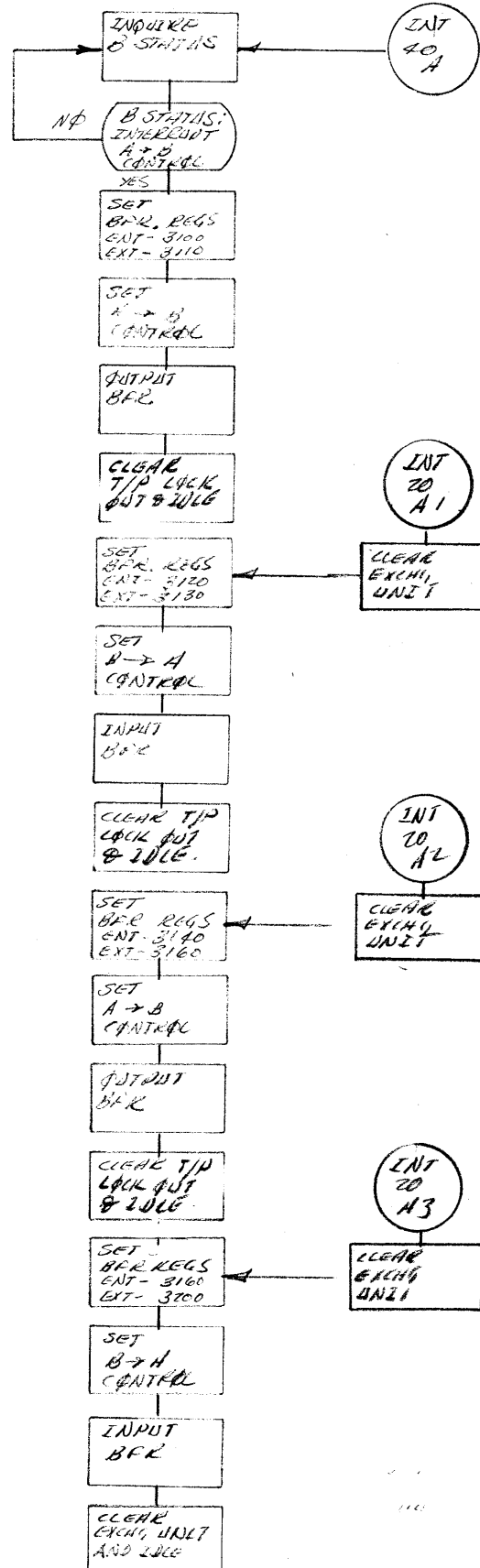
PURPOSE: To test the functioning of an exchange unit's operation between a 160-A computer and a TeleProgrammer.

OPERATION: The 160-A is designated the A computer. The TeleProgrammer is designated the B computer. Programmes are loaded into the respective computers in locations 0000<sup>bank 1</sup>. The T/P is the initial slave and is started from location 0000. The 160-A is then started from location 0000. Normal and buffer exchanges will take place going in both directions with transmission length disparities occurring at both computers. Exchange is terminated when both computers stop at location 0000. A successful exchange of information has occurred when an examination of computer memory reveals the following:

160-A contains	11's	from	location	3000	thru	3017
"	0's	"	"	3020	"	3027
"	77's	"	"	3030	"	3057
"	0's	"	"	3060	"	3077
"	77's	"	"	3100	"	3117
"	0's	"	"	3120	"	3127
"	77's	"	"	3130	"	3157
"	0's	"	"	3160	"	3167
"	77's	"	"	3170	"	3177
T/P contains	77's	from	location	3000	thru	3007
"	0's	"	"	3010	"	3037
"	77's	"	"	3040	"	3047
"	0's	"	"	3030	"	3077
"	77's	"	"	3100	"	3107
"	0's	"	"	3110	"	3127
"	77's	"	"	3140	"	3147
"	0's	"	"	3150	"	3177







INT  
40  
A

INT  
20  
A1

INT  
20  
A2

INT  
20  
A3

INT  
20  
A9/B9

CLEAR  
EXCHG  
UNIT

STOP

B  
COMPUTER  
BFR.

SET  
BFR REGS  
ENT-3100  
EXT-3120

INTERCEPT  
A to B CONTROL

INPUT  
BFR.

CLEAR TIP  
LOCK OUT  
AND LDC

SET  
BFR REGS  
ENT-3120  
EXT-3140

SET  
A to A  
CONTROL

OUTPUT  
BFR

CLEAR TIP  
LOCK OUT  
AND LDC

SET  
BFR REGS  
ENT-3140  
EXT-3150

SET  
B to A  
CONTROL

INPUT  
BFR

CLEAR TIP  
LOCK OUT  
AND LDC

SET  
BFR REGS  
ENT-3160  
EXT-3170

SET  
B to A  
CONTROL

OUTPUT  
BFR

CLEAR  
EXCHG UNIT  
AND LDC

INT  
20  
B1

CLEAR  
EXCHG  
UNIT

INT  
20  
B2

CLEAR  
EXCHG  
UNIT

INT  
20  
B3

CLEAR  
EXCHG  
UNIT

O

3200

ADDRESS

11	0	1	0
12	1	1	0
13	0	1	0
14	1	1	0
15	0	1	0
16	1	1	0
17	0	1	0
18	1	1	0
19	0	1	0
20	1	1	0
21	0	1	0
22	1	1	0
23	0	1	0
24	1	1	0
25	0	1	0
26	1	1	0
27	0	1	0
28	1	1	0
29	0	1	0
30	1	1	0
31	0	1	0
32	1	1	0
33	0	1	0
34	1	1	0
35	0	1	0
36	1	1	0
37	0	1	0
38	1	1	0
39	0	1	0
40	1	1	0
41	0	1	0
42	1	1	0
43	0	1	0
44	1	1	0
45	0	1	0
46	1	1	0
47	0	1	0
48	1	1	0
49	0	1	0
50	1	1	0
51	0	1	0
52	1	1	0
53	0	1	0
54	1	1	0
55	0	1	0
56	1	1	0
57	0	1	0
58	1	1	0
59	0	1	0
60	1	1	0
61	0	1	0
62	1	1	0
63	0	1	0
64	1	1	0
65	0	1	0
66	1	1	0
67	0	1	0
68	1	1	0
69	0	1	0
70	1	1	0
71	0	1	0
72	1	1	0
73	0	1	0
74	1	1	0
75	0	1	0
76	1	1	0
77	0	1	0
78	1	1	0
79	0	1	0
80	1	1	0
81	0	1	0
82	1	1	0
83	0	1	0
84	1	1	0
85	0	1	0
86	1	1	0
87	0	1	0
88	1	1	0
89	0	1	0
90	1	1	0
91	0	1	0
92	1	1	0
93	0	1	0
94	1	1	0
95	0	1	0
96	1	1	0
97	0	1	0
98	1	1	0
99	0	1	0
100	1	1	0

3010

3167

B

3010

11	0	1	0
12	1	1	0
13	0	1	0
14	1	1	0
15	0	1	0
16	1	1	0
17	0	1	0
18	1	1	0
19	0	1	0
20	1	1	0
21	0	1	0
22	1	1	0
23	0	1	0
24	1	1	0
25	0	1	0
26	1	1	0
27	0	1	0
28	1	1	0
29	0	1	0
30	1	1	0
31	0	1	0
32	1	1	0
33	0	1	0
34	1	1	0
35	0	1	0
36	1	1	0
37	0	1	0
38	1	1	0
39	0	1	0
40	1	1	0
41	0	1	0
42	1	1	0
43	0	1	0
44	1	1	0
45	0	1	0
46	1	1	0
47	0	1	0
48	1	1	0
49	0	1	0
50	1	1	0
51	0	1	0
52	1	1	0
53	0	1	0
54	1	1	0
55	0	1	0
56	1	1	0
57	0	1	0
58	1	1	0
59	0	1	0
60	1	1	0
61	0	1	0
62	1	1	0
63	0	1	0
64	1	1	0
65	0	1	0
66	1	1	0
67	0	1	0
68	1	1	0
69	0	1	0
70	1	1	0
71	0	1	0
72	1	1	0
73	0	1	0
74	1	1	0
75	0	1	0
76	1	1	0
77	0	1	0
78	1	1	0
79	0	1	0
80	1	1	0
81	0	1	0
82	1	1	0
83	0	1	0
84	1	1	0
85	0	1	0
86	1	1	0
87	0	1	0
88	1	1	0
89	0	1	0
90	1	1	0
91	0	1	0
92	1	1	0
93	0	1	0
94	1	1	0
95	0	1	0
96	1	1	0
97	0	1	0
98	1	1	0
99	0	1	0
100	1	1	0

O

1ST EXCHG

- 001 A → B FLG ✓
- INTC B ✓
- 002 B → A FLG ✓
- SET A → B CONT ✓

2ND EXCHG

- CLEAR A ✓
- SET B → A FLG ✓
- SET B → A CONT
- CLEAR A

3RD EXCHG

- STATUS OF B
- SET A → B CONT

STATUS OF B

SET B → A FLG

STATUS OF A

SET A → B CONT ✓

STATUS OF A ✓

SET B → A CONT

SET A → B FLG

STATUS OF A

SET A → B CONT

CLEAR B

SET B → A FLG

STATUS OF A

SET B → A CONT

0000	0000	ORG	0	160A(A COMPUTER) PROGRAM
0000	7700	FINISH HLT		** END OF TEST
0001	0000	CHAR		** CHARACTER TO BE EXCHANGED
	0020	ORG	20	INSTRUCTIONS AT INTERRUPTS
0020	0000	INT20		
0021	7500	EXC	5300	** CLEAR EXCHANGE UNIT AND GO
0022	5300			
0023	7100	JFI		TO APPROPRIATE BUFFER
0024	0202			EXCHANGE ROUTINE
0025	0222	BFRX2		
0026	0242	BFRX3		
0027	0000	BFRX4		
	0040	FINISH		
0040	0000	ORG	40	
0041	7100	JFI		** GO TO BFRXI
0042	0155	BFRXI		

	0050		ORG		50	** 160A NORMAL EXCHANGE
0050	2200		LDC	START		** PUT PROGRAM START IN A
0051	0054					
0052	0140		SBUO			** SET ALL BANKS TO 0
0053	0070		ACJO			AND GO TO START
0054	2200	START	LDC		7101	** INITIALIZE INT20 RETURN
0055	7101					
0056	4100		STM	INT20	+3	
0057	0023					
0060	2200		LDC		3000	** PUT CHARACTER IN MLS
0061	3000					
0062	0105	FILL1	ATE	FILL1		3000 THROUGH 3177
0063	0062					
0064	2200		LDC		3200	
0065	3200					
0066	0106	FILL2	ATX	FILL2		
0067	0066					
0070	2001		LDD	CHAR		
0071	0100	FILL3	BLS	FILL3		
0072	0071					
0073	7500	NRMX1	EXC		5320	** INTERRUPT B EXCHG UNIT
0074	5320					
0075	7500		EXC		5301	** SET A TO B FLAG
0076	5301					
0077	7500		EXC		5340	** INQUIRE B EXCHANGE STATUS
0100	5340					
0101	7600		INA			** B STATUS NOT EQUAL A TO B
0102	0705		SBN		5	CONTROL GO TO NRMX1 + 2
0103	6506		NZR	NRMX1	+2	
0104	7500		EXC		5305	** SET A TO B CONTROL
0105	5305					
0106	7343		OUT	NX1FWA		** OUTPUT NORMAL 3000 THRU 3007
0107	3010				3010	
0110	7500	NRMX2	EXC		5300	** CLEAR EXCHANGE UNIT
0111	5300					
0112	7500		EXC		5310	** SET B TO A FLAG
0113	5310					
0114	7500		EXC		5340	** INQUIRE B EXCHANGE STATUS
0115	5340					
0116	7600		INA			** B STATUS NOT EQUAL B TO A
0117	0714		SBN		14	CONTROL GO TO NRMX2 + 2
0120	6506		NZR	NRMX2	+2	
0121	7500		EXC		5314	** SET B TO A CONTROL
0122	5314					
0123	7227		INP	NX2FWA		** INPUT NORMAL 3020 THRU 3030
0124	3030				3030	
0125	7500	NRMX3	EXC		5300	** CLEAR EXCHANGE UNIT



0126	5300					
0127	7500	EXC		5340	** INQUIRE B EXCHANGE STATUS	
0130	5340					
0131	7600	INA			** B STATUS NOT A TO B	
0132	0701	SBN		1	FLAG GO TO NRMX3 + 1	
0133	6504	NZR	NRMX3	+2		
0134	7500	EXC		5305	** SET A TO B CONTROL	
0135	5305					
0136	7315	OUT	NX3FWA		** OUTPUT NORMAL 3040 THRU 3057	
0137	3060			3060		
0140	7500	NRMX4	EXC	5340	** INQUIRE B EXCHANGE STATUS	
0141	5340					
0142	7600	INA			** B STATUS NOT B TO A	
0143	0710	SBN		10	FLAG GO TO NRMX4	
0144	6504	NZR	NRMX4			
0145	7500	EXC		5314		
0146	5314					
0147	7205	INP	NX4FWA		** INPUT NORMAL 3060 THRU 3077	
0150	3100			3100		
0151	3000	NX1FWA		3000	** FIRST WORD ADR. NORMAL EXCHG 1	
0152	3020	NX2FWA		3020	** FIRST WORD ADR. NORMAL XCHG 2	
0153	3040	NX3FWA		3040	** FIRST WORD ADR. NORMAL XCHG 3	
0154	3060	NX4FWA		3060	** FIRST WORD ADR. NORMAL XCHG 4	

						160A BUFFER EXCHANGE
0155	7500	BFRX1	EXC		5340	** INQUIRE B EXCHANGE STATUS
0156	5340					
0157	7600		INA			** B STATUS NOT INTERRUPT A
0160	0725		SBN		25	SETTING A TO B CONTROL GO
0161	6504		NZR	BFRX1		TO BFRX1
0162	2200		LDC		3100	** SET BUFFER REGS
0163	3100					
0164	0105	SET1	ATE	SET1		ENT TO 3100
0165	0164					
0166	2200		LDC		3110	EXT TO 3110
0167	3110					
0170	0106	SET2	ATX	SET2		
0171	0170					
0172	7500		EXC		5305	** SET A TO B CONTROL
0173	5305					
0174	7300	BFRXIS	IBO	BFRXIS		** BUFFER OUTPUT
0175	0174					
0176	0120	IDLE	CIL			** IDLE LOOP
0177	0007		NOP7			
0200	7101		JFI		1	
0201	0177			IDLE	+1	
0202	2200	BFRX2	LDC		3120	** SET BUFFER REGS
0203	3120					
0204	0105	SET3	ATE	SET3		ENT # 3120
0205	0204					
0206	2200		LDC		3130	EXT # 3130
0207	3130					
0210	0106	SET4	ATX	SET4		
0211	0210					
0212	5500		AOM	INT20	+3	** SET INT20 RETURN
0213	0023					
0214	7500		EXC		5314	** SET B TO A CONTROL
0215	5314					
0216	7200	BFRX2S	IBI	BFRX2S		** BUFFER INPUT AND IDLE
0217	0216					
0220	7101		JFI		1	
0221	0176			IDLE		
0222	2200	BFRX3	LDC		3140	** SET BUFFER REGS
0223	3140					
0224	0105	SET5	ATE	SET5		ENT # 3140
0225	0224					
0226	2200		LDC		3160	EXT # 3160
0227	3160					
0230	0106	SET6	ATX	SET6		
0231	0230					
0232	5500		AOM	INT20	+3	** SET INT20 RETURN
0233	0023					

```
0234 7500          EXC          5305  ** SET A TO B CONTROL
0235 5305
0236 7300  BFRX3S IBO  BFRX3S          ** BUFFER OUTPUT AND IDLE
0237 0236
0240 7101          JFI          I
0241 0176          IDLE
0242 2200  BFRX4  LDC          3160  ** SET BUFFER REGS
0243 3160
0244 0105  SET7  ATE  SET7          ENT ≠ 3160
0245 0244
0246 2200          LDC          3200  EXT ≠ 3200
0247 3200
0250 0106  SET8  ATX  SET8
0251 0250
0252 5500          ADM  INT20 +3  ** SET INT20 RETURN
0253 0023
0254 7500          EXC          5314  ** SET B TO A CONTROL
0255 5314
0256 7200  BFRX4S IBI  BFRX4S          ** BUFFER INPUT AND IDLE
0257 0256
0260 7101          JFI          I
0261 0176          IDLE
      0000          END          COMPLETE ASSEMBLY
```

0000	0000		ORG		0	TELEPROGRAMMER (B COMPUTER) PROGRAM
0000	0077	FINISH		HLT		** END OF TEST
0020	0020		ORG		20	INSTRUCTIONS AT INTERRUPTS
0020	0075			EXF		** CLEAR EXCHANGE UNIT
0021	0053				53	
0022	0000				0	
0023	0164			UJP	T1	** GO TO INT20X
0024	0000	INT20X				
	0040		ORG		40	
0040	0164			UJP	T1	** GO TO NRMX1
0041	0101			NRMX1		

## TELEPROGRAMMER NORMAL EXCHANGE

0050	0050	ORG	LDN	50	
0050	0020		LDN		** SET TAG 3 TO 6
0051	0006			6	
0052	0302		ATT	T3	
0053	0020		LDN		** SET TAG 1 TO 0
0054	0000			0	
0055	0102		ATT	T1	
0056	0020		LDN		** PUT 0 IN MLS
0057	0000			0	3000 THROUGH 3376
0060	0141		STM	T1	
0061	0065		TPFILL		
0062	0020		LDN		
0063	0000			0	
0064	0341		STM	T3	
0065	0000	TPFILL			
0066	0155		RAO	T1	
0067	0065		TPFILL		
0070	0161		NZP	T1	
0071	0062		TPFILL	-3	
0072	0075	IDLE	EXF		** CLEAR EXCHANGE UNIT, TELEPROGRAMMER,
0073	0053			53	AND IDLE
0074	0000			0	
0075	0013		CIL		
0076	0002		ATT		
0077	0164		UJP	T1	
0100	0076		IDLE	+4	
0101	0075	NRMXI	EXF		** INQUIRE A EXCHANGE STATUS
0102	0053			53	
0103	0040			40	
0104	0076		INA		** A STATUS NOT EQUAL INTERRUPT AND
0105	0034		SBN		A TO B FLAG GO TO NRMXI
0106	0021			21	
0107	0161		NZP	T1	
0110	0101		NRMXI		
0111	0075		EXF		** SET A TO B CONTROL AND
0112	0053			53	INPUT NORM. 3000 THROUGH
0113	0005			5	3017
0114	0372		INN	T3	
0115	0000			0	
0116	0020			20	
0117	0075	NRMX2	EXF		** INQUIRE A EXCHANGE STATUS
0120	0053			53	
0121	0040			40	
0122	0076		INA		** A STATUS NOT EQUAL B TO A
0123	0034		SBN		FLAG GO TO NRMX2
0124	0010			10	
0125	0161		NZP	T1	

0126	0117		NRMX2		
0127	0075		EXF		** SET B TO A CONTROL AND
0130	0053			53	OUTPUT NORMAL 3020 THROUGH
0131	0014			14	3037
0132	0373		OUT	T3	
0133	0020			20	
0134	0040			40	
0135	0075	NRMX3	EXF		** SET A TO B FLAG
0136	0053			53	
0137	0001			1	
0140	0075		EXF		** INQUIRE A EXCHANGE STATUS
0141	0053			53	
0142	0040			40	
0143	0076		INA		** A STATUS NOT EQUAL A TO B
0144	0034		SBN		CONTROL GO TO NRMX3 + 3
0145	0005			5	
0146	0161		NZP	T1	
0147	0140		NRMX3	+3	
0150	0075		EXF		** SET A TO B CONTROL AND
0151	0053			53	INPUT NORMAL 3040 THROUGH
0152	0005			5	3047
0153	0372		INN	T3	
0154	0040			40	
0155	0050			50	
0156	0075	NRMX4	EXF		** CLEAR EXCHANGE UNIT
0157	0053			53	
0160	0000			0	
0161	0075		EXF		** SET B TO A FLAG
0162	0053			53	
0163	0010			10	
0164	0075		EXF		** INQUIRE A EXCHANGE STATUS
0165	0053			53	
0166	0040			40	
0167	0076		INA		** A STATUS NOT EQUAL B TO A
0170	0034		SBN		CONTROL GO TO NRMX4 + 6
0171	0014			14	
0172	0161		NZP	T1	
0173	0164		NRMX4	+6	
0174	0075		EXF		** SET B TO A CONTROL AND
0175	0053			53	OUTPUT NORMAL 3060 THROUGH
0176	0014			14	3067
0177	0373		OUT	T3	
0200	0060			60	
0201	0070			70	

					TELEPROGRAMMER BUFFER EXCHANGE
0202	0020	SET1	LDN		** SET BUFFER REGS
0203	0100			100	ENT TO 3100
0204	0104		ABR	T1	EXT TO 3120
0205	0202		SET1		
0206	0020	SET2	LDN		
0207	0120			120	
0210	0105		ABX	T1	
0211	0206		SET2		
0212	0020		LDN		** SET INT20X
0213	0225		SET3		
0214	0041		STM		
0215	0024		INT20X		
0216	0075		EXF		** INTERRUPT A EXCHANGE SETTING B EXCHANGE A TO B CONTROL
0217	0053			53	
0220	0025			25	
0221	0170	BFRX1	IBI	T1	** BUFFER INPUT AND IDLE
0222	0221		BFRX1		
0223	0164		UJP	T1	
0224	0075		IDLE	+3	
0225	0020	SET3	LDN		** SET BUFFER REGS
0226	0120			120	ENT TO 3120
0227	0104		ABR	T1	EXT TO 3140
0230	0225		SET3		
0231	0020	SET4	LDN		
0232	0140			140	
0233	0105		ABX	T1	
0234	0231		SET4		
0235	0020		LDN		** SET INT20X
0236	0250		SET5		
0237	0041		STM		
0240	0024		INT20X		
0241	0075		EXF		** SET B TO A CONTROL, OUTPUT BUFFER AND IDLE
0242	0053			53	
0243	0014			14	
0244	0171	BFRX2	IBO	T1	
0245	0244		BFRX2		
0246	0164		UJP	T1	
0247	0075		IDLE	+3	
0250	0020	SET5	LDN		** SET BUFFER REGS
0251	0140			140	ENT TO 3140
0252	0104		ABR	T1	EXT TO 3150
0253	0250		SET5		
0254	0020	SET6	LDN		
0255	0150			150	
0256	0105		ABX	T1	
0257	0254		SET6		
0260	0020		LDN		** SET INT20X

0261	0273		SET7		
0262	0041		STM		
0263	0024		INT20X		
0264	0075		EXF		** SET B TO A CONTROL,
0265	0053			53	INPUT BUFFER AND IDLE
0266	0005			5	
0267	0170	BFRX3	IBI	T1	
0270	0267		BFRX3		
0271	0164		UJP	T1	
0272	0075		IDLE	+3	
0273	0020	SET7	LDN		** SET BUFFER REGS
0274	0160			160	ENT TO 3160
0275	0104		ABR	T1	EXT TO 3170
0276	0273		SET7		
0277	0020	SET8	LDN		
0300	0170			170	
0301	0105		ABX	T1	
0302	0277		SET8		
0303	0020		LDN		** SET INT20X
0304	0000		FINISH		
0305	0041		STM		
0306	0024		INT20X		
0307	0075		EXF		** SET B TO A CONTROL,
0310	0053			53	OUTPUT BUFFER AND IDLE
0311	0014			14	
0312	0171	BFRX4	IBO	T1	
0313	0312		BFRX4		
0314	0064		UJP		
0315	0075		IDLE	+3	



0100	T1	EQU	100		
0200	T2	EQU	200		
0300	T3	EQU	300		
0000	ERR	EQU		0	ERROR STOP
0001	SHA	EQU		1	SHIFT A LEFT ONE BIT
0002	ATT	EQU		2	A TO TAG REGISTER
0003	TTA	EQU		3	TAG REGISTER CONTENTS TO A
0004	ABR	EQU		4	A TO BUFFER ENTRANCE REGISTER
0005	ABX	EQU		5	A TO BUFFER EXIST REGISTER
0006	BER	EQU		6	CONTENTS OF BER REGISTER TO A
0007	CBC	EQU		7	CLEAR BUFFER CONTROLS
0010	LPN	EQU		10	LOGICAL PRODUCT NO ADDRESS
0011	LPM	EQU		11	LOGICAL PRODUCT MEMORY ADDRESS
0012	LPI	EQU		12	LOGICAL PRODUCT INDIRECT ADDRESS
0013	CIL	EQU		13	CLEAR INTERRUPT LOCKOUT
0014	LSN	EQU		14	LOGICAL SUM NO ADDRESS
0015	LSM	EQU		15	LOGICAL SUM MEMORY ADDRESS
0016	LSI	EQU		16	LOGICAL SUM INDIRECT ADDRESS
0020	LDN	EQU		20	LOAD A NO ADDRESS MODE
0021	LDM	EQU		21	LOAD A MEMORY
0022	LDI	EQU		22	LOAD A INDIRECT
0025	LCM	EQU		25	LOAD COMPLEMENT TO A MEMORY
0026	LCI	EQU		26	LOAD COMPLEMENT TO A INDIRECT
0030	ADN	EQU		30	ADD NO ADDRESS
0031	ADM	EQU		31	ADD MEMORY ADDRESS
0032	ADI	EQU		32	ADD INDIRECT ADDRESS
0034	SBN	EQU		34	SUBTRACT NO ADDRESS
0035	SBM	EQU		35	SUBTRACT MEMORY ADDRESS
0036	SBI	EQU		36	SUBTRACT INDIRECT ADDRESS
0041	STM	EQU		41	STORE A MEMORY
0042	STI	EQU		42	STORE A INDIRECT
0051	RAM	EQU		51	REPLACE ADD MEMORY ADDRESS
0055	RAO	EQU		55	REPLACE ADD ONE MEMORY ADDRESS
0060	ZJB	EQU		60	JUMP, IF CONTENTS OF A = 0
0060	ZJP	EQU		60	JUMP, IF CONTENTS OF A = 0
0061	NZP	EQU		61	JUMP, IF CONTENTS OF A ≠ 0
0062	PJP	EQU		62	JUMP, IF CONTENTS OF A 0 POSITIVE
0063	NJP	EQU		63	JUMP, IF CONTENTS OF A 0 NEGATIVE
0064	UJP	EQU		64	UNCONDITIONAL JUMP
0070	IBI	EQU		70	INITIATE BUFFER INPUT
0071	IBO	EQU		71	INITIATE BUFFER OUTPUT
0072	INN	EQU		72	INPUT NORMAL
0073	OUT	EQU		73	OUTPUT NORMAL
0074	OTN	EQU		74	OUTPUT, NO ADDRESS
0075	EXF	EQU		75	EXTERNAL FUNCTION
0076	INA	EQU		76	INPUT TO A
0077	HLT	EQU		77	HALT
0000		END			COMPLETE ASSEMBLY

### Example 3 A Return Jump Technique

The following technique provides for transfer of control from a Main Program to a sub-program and return with the ability to use any Tag Register in either the Main Program or Sub-program. The technique assumes that if a jump from the sub-program to another sub-program is desired; the programmer has the responsibility of saving and restoring the initial contents of the "Jump Area", (see flowchart), before jumping back to the Main Program.

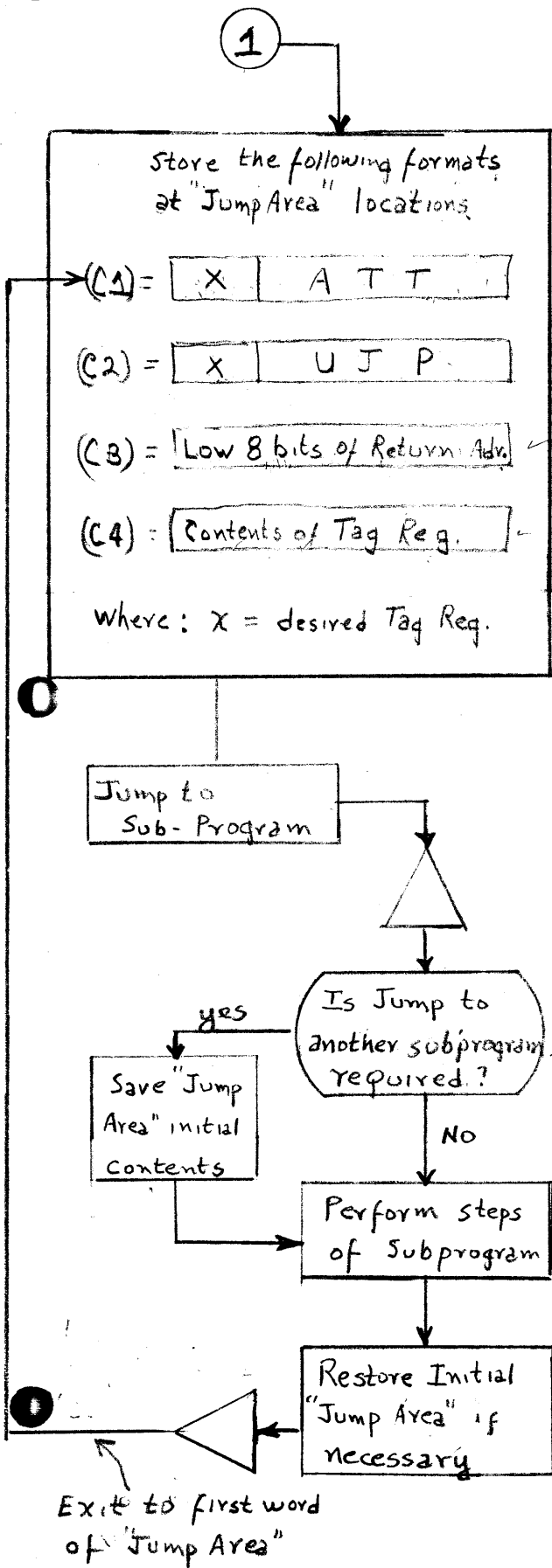
The technique consists in saving the 12 bit Return Address of the Main Program in two consecutive words of a "Jump Area". (The lower 8 bits are stored in one word and the contents of the Tag Register — containing the upper 4 bits — in the following word). The Unconditional Jump exit instruction is then created and stored in the two words preceding the Return Address words.

The total technique requires 25 words:

17 in the Main Program, 4 in the Subprogram, and

4 in the "Jump Area".

# Flow Chart — Return Jump Technique — Program



End Portion of Main Program:

```

{ O L D N } Store lower 8
{ Ret. Addr. } bits of Return
                Address at
                location, C3, in
                "Jump Area"
{ O S T M }
{ C 3. }

-----
P1 { X T T A } Tag Reg. to "A"

{ O S T M } Store Tag Reg.
{ C 4 } contents at C4

{ O L D N } Create and
{ X U J P } store word
                format shown
                at C2
{ O S T M }
{ C 2 }

{ O L D N } Create and
{ X A T T } store word
                format shown
                at C1
{ O S T M }
{ C 1 }

{ T U J P } Jump to
{ S S S } Sub program
    
```

End Portion of Subprogram:

```

{ O L D M } Restore Main
{ C 4 } Program Tag Reg.
                Contents to A
                and jump to C1
{ O U J P }
{ C 1 }
    
```

TITLE: GENERALISE PARITY ANALYSIS ROUTINE

STORAGE: GENERAL HOUSE KEEPING - 13 WORDS  
HORIZONTAL PARITY - 68 WORDS  
VERTICAL PARITY - 48 WORDS  
PARAMETERS - 12 WORDS  
TOTAL - 141 WORDS

PURPOSE: To provide a general routine wherein horizontal or vertical parities can be either generated or compared. The horizontal parity may be located in any bit position of the T/P word. The vertical parity word includes all bits of the message words and is the last word of a message block. The parity mode may be odd or even.

OPERATION: The activity (to generate or compare) this routine is to perform is indicated by the ACTIVITY parameter which is set by the main program. The parity mode (is odd or even) to be applied by this routine is indicated by the PMODE parameter which is also set by the main program. The horizontal parity bit position is indicated in the HPPPS parameter by a 1 bit in the appropriate position of an otherwise all zero bit word (i.e. 01000000 indicates bit 6 to be the parity bit). This is also set by the main program. FSTCHR and LSTCHR parameters indicate the first and last words of the message block. LSTCHR being the address of the message's vertical parity word. These are also set by the main program. Upon encountering the first discrepancy during the compare routine, an error flag is set and control is immediately returned to the main program.

1 6 5 4 3 2 1 0  
 p b<sub>6</sub> b<sub>5</sub> b<sub>4</sub> b<sub>3</sub> b<sub>2</sub> b<sub>1</sub> b<sub>0</sub>  
 0 b<sub>6</sub> b<sub>5</sub> b<sub>4</sub> b<sub>3</sub> b<sub>2</sub> b<sub>1</sub> b<sub>0</sub>  
 b<sub>6</sub> b<sub>5</sub> b<sub>4</sub> b<sub>3</sub> b<sub>2</sub> b<sub>1</sub> b<sub>0</sub> 0  
 b<sub>5</sub> b<sub>4</sub> b<sub>3</sub> b<sub>2</sub> b<sub>1</sub> b<sub>0</sub> 0 b<sub>6</sub>  
 b<sub>4</sub> b<sub>3</sub> b<sub>2</sub> b<sub>1</sub> b<sub>0</sub> 0 b<sub>6</sub> b<sub>5</sub>  
 b<sub>3</sub> b<sub>2</sub> b<sub>1</sub> b<sub>0</sub> 0 b<sub>6</sub> b<sub>5</sub> b<sub>4</sub>  
 b<sub>2</sub> b<sub>1</sub> b<sub>0</sub> 0 b<sub>6</sub> b<sub>5</sub> b<sub>4</sub> b<sub>3</sub>  
 b<sub>1</sub> b<sub>0</sub> 0 b<sub>6</sub> b<sub>5</sub> b<sub>4</sub> b<sub>3</sub> b<sub>2</sub>  
 b<sub>0</sub> 0 b<sub>6</sub> b<sub>5</sub> b<sub>4</sub> b<sub>3</sub> b<sub>2</sub> b<sub>1</sub>

$$\sum_{i=0}^6 E_{b_i} \rightarrow \sum_{i=0}^6 E_{b_i} = (\text{PARITY})$$

$$\sum_{i=0}^6 E_{b_i} = (\text{PARITY})$$

$\sum_{i=0}^6 E_{b_i}$  is the PARITY SUM.

A PARITY SUM of 0 indicates an even number of bits  
 " " " " " 1 " " " " " " " " " " " "

Thus depending on the PARITY Mode (ie odd or even) a logical sum or logical product may be done with the PARITY SUM to yield the correct PARITY BIT as indicated below

MODE =	EVEN		ODD	
	even	odd	even	odd
number of bits =	even	odd	even	odd
PARITY SUM =	0	1	0	1
logical prod =	1	1		
logical sum =			1	1
PARITY BIT =	0	1	1	0

MARK UP PARITIES

0 → D'PRTY

0 → H'PRTY

-7 → BIT COUNT

CHAR(C) → A

ZERO OUT PARITY POS.

STORE IN TEMP

TEMP → A

SHIFT 1 BIT

STORE IN TEMP

LOGICAL SUM TO H'PRTY

STORE IN H'PRTY

INCREMENT BIT COUNT

≠ BIT COUNT: 0

= ISOLATE PARITY POS. IN H'PRTY

PARITY MODE?

EVEN LOGICAL SUM 1 TO PARITY POS. OF H'PRTY

ADD H'PRTY TO TEMP

ACTIVITY? GENERATE PARITIES

GO TO C

H

1

TEMP → CHAR(C)

LOGICAL SUM TEMP TO D'PRTY

INCREMENT C

C INDICATES LAST CHAR?

NO GO TO H

PARITY MODE?

EVEN LOGICAL SUM 1 TO D'PRTY

LOGICAL SUM 1 TO D'PRTY

ACTIVITY? GENERATE PARITIES

STORE D'PRTY IN MESSAGE

COMPARE PARITIES

D'PRTY: READ UP

RETURN

≠ SET ERROR FLAG

C

TEMP: CHAR(C)

= GO TO 2

≠ SET ERROR FLAG

RETURN

COMPARE PARITIES

ADDRESS TAG ASSIGNMENTS

\* T1 = TAG USED IN EXECUTION OF ROUTINE AND IS SET BY MAIN PROGRAM

\* T2 = TAG USED IN ADDRESSING MESSAGE BLOCK WORDS.

0 BLOCK PARAMETERS SET BY MAIN PROGRAM

```

0360 0360      ORG      360
0360 0000      ACTVITY
0361 0000      P-ODE
0362 0000      HPPOS
0363 0000      FSTCHR
0364 0000      LSTCHR
0365 0000      CHAR
0366 0000      PRGTAG
0367 0000      ERROR
    
```

\* ACTIVITY THE ROUTINE IS TO PERFORM (0 = GENERATE PARITIES, 1 = COMPARE PARITIES)  
 \* PARITY MODE(0 = EVEN, 1 = ODD)  
 \* POSITION OF HORIZ PARITY BIT  
 \* ADDR. OF FIRST CHAR. OF MESSAGE BLOCK  
 \* ADDR. OF LAST CHAR. OF MESSAGE BLOCK (VERT PARITY WORD)  
 \* ADDR. OF CHAR. CURRENTLY BEING ANALYZED (SET BY ROUTINE)  
 \* CONTENTS OF TAG USED IN EXECUTION OF MAIN PROGRAM  
 \* INDICATES AN ERROR IN PARITY DETECTED IN COMPARE (SET BY ROUTINE)

```

0050 0141      START      STM      50
0051 0254      RETURN
0052 0121      LDM      TI
0053 0363      FSTCHR
0054 0041      STM
0055 0365      CHAR
0056 0020      LDN
0057 0000
0060 0041      STM
0061 0367      ERROR
0062 0141      STM      TI
0063 0255      VPRTY
0064 0020      HORIZI    LDN
0065 0000
0066 0141      STM      TI
0067 0256      HPRTY
0070 0020      LDN
0071 0370
0072 0141      STM      TI
0073 0257      BIT
0074 0125      LCM      TI
0075 0362      HPPOS
0076 0141      STM      TI
    
```

\*\* SET RETURN  
 \*\* INITIALIZE FOR FIRST WORD ON WHICH PARITIES ARE TO BE GENERATED.  
 \*\* SET ERROR TO 0  
 \*\* SET VPRTY TO 0  
 \*\* SET HPRTY TO 0  
 \*\* SET BIT COUNTER TO -8  
 \*\* PUT 0 IN PARITY POS. OF WORD AND STORE IN TEMP.

0077	0103		HORIZ2		
0100	0222		LDI	T2	
0101	0365		CHAR		
0102	0010		LPN		
0103	0000	HORIZ2			
0104	0041		STM		
0105	0260		TEMP		
0106	0121	HORIZ3	LDM	TI	** LOGICAL SUM SHIFTED
0107	0260		TEMP		WORD TO HPRTY
0110	0001		SHA		
0111	0141		STM	TI	
0112	0260		TEMP		
0113	0115		LSM	TI	
0114	0256		HPRTY		
0115	0141		STM	TI	
0116	0256		HPRTY		
0117	0155		RAO	TI	** INCREMENT BIT COUNTER
0120	0257		BIT		IF NOT 0 GO TO HORIZ3
0121	0161		NZP	TI	
0122	0106		HORIZ3		
0123	0121		LDM	TI	** MASK OFF ALL BUT
0124	0256		HPRTY		PARITY POS BIT OF
0125	0111		LPM	TI	HPRTY
0126	0362		HPPOS		
0127	0141		STM	TI	
0130	0362		HPPOS		
0131	0120		LDN	TI	** IF PARITY MOVE EVEN
0132	0361		PMODE		GO TO EHPRTY
0133	0160		ZJP	TI	
0134	0144		EHPRTY		
0135	0121	OPRTY	LDM	TI	** SET ODD PARITY BIT
0136	0362		HPPOS		AND GO TO HORIZ4
0137	1500	LSM	TI		
0140	0100				
0141	0362		HPPOS		
0142	0164		UJP	TI	
0143	0150		HORIZ4		
0144	0121	EHPRTY	LDM	TI	** SET EVEN PARITY BIT
0145	0362		HPPOS		
0146	0111		LPM	TI	
0147	0362		HPPOS		
0150	0151	HORIZ4	RAM	TI	** PLACE HORIZ PARITY BIT
0151	0260		TEMP		IN WORD
0152	0121		LDM	TI	** IF ACTIVITY IS COMPARE
0153	0360		ACTVTY		GO TO COMPRI.
0154	0161		NZP	TI	
0155	0226		COMPRI		
0156	0121	VERTI	LDM	TI	** REPLACE CHAR WITH PARITY



0157	0260		TEMP		BIT BACK INTO MEMORY
0160	0242		STI	T2	
0161	0365		CHAR		
0162	0115		LSM	T1	** LOGICAL SUM TEMP TO
0163	0255		VPRTY		VPRTY
0164	0141		STM	T1	
0165	0255		VPRTY		
0166	0055		RAO		** INCREMENT CHAR IF NOT
0167	0365		CHAR		LAST CHARACTER PLUS 1
0170	0135		SBM	T1	GO TO HORIZI
0171	0364		LSTCHR		
0172	0163		NJP	T1	
0173	0064		HORIZI		
0174	0121		LDM	T1	** IF PARITY MODE EVEN
0175	0361		PMODE		GO TO EVPRTY
0176	0160		ZJP	T1	
0177	0206		EVPRTY		
0200	0121	OVPRTY	LDM	T1	** SET ODD VERT PARITY WORD
0201	0255		VPRTY		
0202	0014		LSN		
0203	0377			377	
0204	0164		UJP	T1	
0205	0212		VERT2		
0206	0121	EVPRTY	LDM	T1	** SET EVEN VERT PARITY WORD
0207	0255		VPRTY		
0210	0010		LPN		
0211	0377			377	
0212	0141	VERT2	STM	T1	** STORE IN VPRTY
0213	0255		VPRTY		
0214	0120		LDM	T1	** IF ACTIVITY IS COMPARE
0215	0360		ACTVTY		GO TO COMPR2
0216	0161		NZP	T1	
0217	0240		COMPR2		
0220	0121		LDM	T1	** PUT VPRTY IN MESSAGE BLOCK.
0221	0255		VPRTY		
0222	0241		STM	T2	
0223	0365		CHAR		
0224	0164		UJP	T1	** GO TO FINISH
0225	0250		FINISH		
0226	0222	COMPRI	LDI	T2	** IF PARITY GENERATED EQUAL
0227	0365		CHAR		PARITY IN WORD GO TO
0230	0135		SBM	T1	VERT 1
0231	0260		TEMP		
0232	0160		ZJP	T1	
0233	0156		VERT1		
0234	0055		RAO		** IF PARITY GENERATED NOT
0235	0367		ERROR		EQUAL PARITY IN WORD; SET
0236	0164		UJP	T1	ERROR TO NON ZERO AND

0237	0250		FINISH		GO TO FINISH
0240	0222	COMPR2	LDI	T2	** IF VPRTY GENERATED EQUAL
0241	0365		CHAR		VPRTY IN MESSAGE GO TO
0242	0135		SBM	T1	FINISH
0243	0255		VPRTY		
0244	0160		ZJP	T1	
0245	0250		FINISH		
0246	0055		RAO		** IF VPRTY GENERATED NOT EQUAL
0247	0367		ERROR		VPRTY IN MESSAGE SET ERROR
					TO NON ZERO
0250	0021	FINISH	LDM		** END ROUTINE AND RETURN TO
0251	0366		PRGTAG		MAIN PROGRAM.
0252	0102		ATT	T1	
0253	0164		UJP	T1	
0254	0000	RETURN			
					ROUTINE BLOCK PARAMETERS
					USED BY ROUTINE
0255	0000	VPRTY			* VERTICAL PARITY WORD
0256	0000	HPRTY			* HORIZONTAL PARITY BIT
0257	0000	BIT			* WORD BIT POSITION COUNTER
0260	0000	TEMP			* TEMPORARY STORAGE

0100	T1	EQU	100	
0200	T2	EQU	200	
0300	T3	EQU	300	
0000	ERR	EQU	0	ERROR STOP
0001	SHA	EQU	1	SHIFT A LEFT ONE BIT
0002	ATT	EQU	2	A TO TAG REGISTER
0003	TTA	EQU	3	TAG REGISTER CONTENTS TO A
0004	ABR	EQU	4	A TO BUFFER ENTRANCE REGISTER
0005	ABX	EQU	5	A TO BUFFER EXIST REGISTER
0006	BER	EQU	6	CONTENTS OF BER REGISTER TO A
0007	CBC	EQU	7	CLEAR BUFFER CONTROLS
0010	LPN	EQU	10	LOGICAL PRODUCT NO ADDRESS
0011	LPM	EQU	11	LOGICAL PRODUCT MEMORY ADDRESS
0012	LPI	EQU	12	LOGICAL PRODUCT INDIRECT ADDRESS
0013	CIL	EQU	13	CLEAR INTERRUPT LOCKOUT
0014	LSN	EQU	14	LOGICAL SUM NO ADDRESS
0015	LSM	EQU	15	LOGICAL SUM MEMORY ADDRESS
0016	LSI	EQU	16	LOGICAL SUM INDIRECT ADDRESS
0020	LDN	EQU	20	LOAD A NO ADDRESS MODE
0021	LDM	EQU	21	LOAD A MEMORY
0022	LDI	EQU	22	LOAD A INDIRECT
0025	LCM	EQU	25	LOAD COMPLEMENT TO A MEMORY
0026	LCI	EQU	26	LOAD COMPLEMENT TO A INDIRECT
0030	ADN	EQU	30	ADD NO ADDRESS
0031	ADM	EQU	31	ADD MEMORY ADDRESS
0032	ADI	EQU	32	ADD INDIRECT ADDRESS
0034	SBN	EQU	34	SUBTRACT NO ADDRESS
0035	SBM	EQU	35	SUBTRACT MEMORY ADDRESS
0036	SBI	EQU	36	SUBTRACT INDIRECT ADDRESS
0041	STM	EQU	41	STORE A MEMORY
0042	STI	EQU	42	STORE A INDIRECT
0051	RAM	EQU	51	REPLACE ADD MEMORY ADDRESS
0055	RAO	EQU	55	REPLACE ADD ONE MEMORY ADDRESS
0060	ZJB	EQU	60	JUMP, IF CONTENTS OF A = 0
0060	ZJP	EQU	60	JUMP, IF CONTENTS OF A = 0
0061	NZP	EQU	61	JUMP, IF CONTENTS OF A ≠ 0
0062	PJP	EQU	62	JUMP, IF CONTENTS OF A 0 POSITIVE
0063	NJP	EQU	63	JUMP, IF CONTENTS OF A 0 NEGATIVE
0064	UJP	EQU	64	UNCONDITIONAL JUMP
0070	IBI	EQU	70	INITIATE BUFFER INPUT
0071	IBO	EQU	71	INITIATE BUFFER OUTPUT
0072	INN	EQU	72	INPUT NORMAL
0073	OUT	EQU	73	OUTPUT NORMAL
0074	OTN	EQU	74	OUTPUT, NO ADDRESS
0075	EXF	EQU	75	EXTERNAL FUNCTION
0076	INA	EQU	76	INPUT TO A
0077	HLT	EQU	77	HALT
0000		END		COMPLETE ASSEMBLY

TEST: 80-4 PAPER TAPE CHECKER WITH CHECKSUM

SIZE: 170 WORDS

HARDWARE: 8092 & 350 PAPER TAPE READER

PURPOSE: To load a 80-4 paper into 8092 and generate a logical sum check sum of the tape contents, and display this check sum in the A register. This check sum can be compared with the check sum generated when the tape being loaded was dumped from the 160-1 with an appropriate dump routine.

OPERATING INSTRUCTIONS:

- 1) Load the loader in any block of the 8092 core starting in location 000. It will be loaded when the P register displays the starting address  $(S)+170$ .
- 2) Place the tape to be loaded in the 350.
  - a) Enter the starting address of the loader in P
  - b) Enter the low 4 bits of this address in A
  - c) Run to halt at  $S+1$
  - d) Enter the low 4 bits of the sum of the program to be loaded in A
  - e) Run to halt at  $S+3$
  - f) Enter the low 8 bits of the sum of the program to be loaded in A

g) Line is held at S+54. The checksum will be displayed in A

### 3) Error Stops

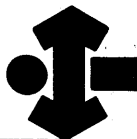
- a) S+54 - Should the program stop running at this location before the last frame is read a tape format error has been encountered, specifically two successive bad data frames have been encountered.
- b) S+74 - Error in tape format; two successive bad data frames have been encountered.
- c) S+100 - Tape is about to be loaded over the loader, the loader will not be destroyed and may be used again by going to Z above.

160-A BI-4 DUMP

DISPLAYING LOGICAL  
SUM CHECK SUM,

	MC	(MC)				
DUMP	0000	42 17	STF	MC	**	STORE FWA IN MC
	1	0000	EXR		**	STORE LWA+1 IN LWA+1
	2	4205	STF	LWA+1		
	3	0400	LDN	0	**	SET CHECKSUM EQUALZER
	4	4206	STF	CHKSUM		
DUMP 1	5	2212	LDF	MC	**	IF FWA NOT EQUAL LWA+1 JUMP TO DUMP 2
	6	3600	ORC			
LWA+1	7	0000				
	10	6106	NEF	DUMP 2		
	11	2200	LDC		**	DISPLAY 8 BITS OF CHECKSUM IN A REG. AND HALT
CHKSUM	12	0000				
	13	1200	LPC			
	14	0317	STI			
	15	1111	HCT			
DUMP 2	16	2100	LDM		**	LOAD CONTENTS OF FWA AND STORE AT WORD
MC	17	0000				
	20	4200	STC			
WORD	1	0000				
	2	17 10	SCB	CHKSUM	**	LOGICAL SUM WORD TO CHECKSUM
	3	43 11	STB	CHKSUM		
	4	75 00	EXC		**	SELECT PUNCH 4 BITS OF WORD
	5	41 04	4104			
	6	23 05	LDB	WORD	**	PUT OUT TO TAPE HI 4 BITS OF WORD WITH 5TH LEVEL PUNCH
	7	01 11	LSC			
	10	0103	LSZ			
	11	02 17	LPN	17		
	12	06 20	ADN	20		
	13	76 77	PTA	11		
	14	23 13	LDB	WORD	**	PUT OUT TO TAPE LO ORDER 4 BITS OF WORD IN DUMP,
	15	02 17	ILPN	17		
	16	76 77	PTA			
	17	57 20	RAB	MC	**	INCREMENT MC AND JUMP TO DUMP
	10	65 33	NEB	DUMP 1		
	11	00 00	EXR			
	12					
	13					
	14					
	15					
	16					
	17					

CHK SUM = 5671  
LWAH = 1039



CONTROL DATA CORPORATION

160 DATA PROCESSOR

REPLACE  
ADD CHECK SUM IN 1001  
BI-4

PAGE NO. \_\_\_\_\_  
DATE \_\_\_\_\_  
PROGRAMMER \_\_\_\_\_

M.L.	F	E	COMMENTS	M.L.	F	E	COMMENTS
	0	4.2	1.5	STF 15		0	Star FWH in ML
	1	0.0	0.0	ERR		1	Star LWAH+1 in LWAH1
	2	4.2	0.5	STF 3 loc		2	
	3	0.4	0.0	LWN 2001		3	Set CHRSUM to -0-
	4	4.2	0.6	STF 6 loc		4	
*1	5	3.2	1.0	LDF 10 loc		5	If FWH not equal LWAH1
	6	3.6	0.0	SBC		6	jump to *2
LWAH1	7	0.0	0.0			7	
	0	6.1	0.4	NZF 4 loc		0	
	1	2.2	0.0	LDC		1	Display CHRSUM in 11 regis
CHRSUM	2	0.0	0.0	CHECKSUM		2	and halt
	3	0.0	0.0	HLT		3	
*2	4	2.1	0.0	LDM		4	Load contents of ML
ML	5	0.0	0.0			5	and store in WWD
	6	4.2	0.0	STC		6	
WWD	7	0.0	0.0			7	
-	0	5.3	0.6			0	Replace add WWD to CHRSUM
	1	7.5	0.0	select punch		1	select punch
	2	4.1	0.4			2	
	3	2.3	0.4	LDB 4 loc		3	Output to tape 4 bits
	4	0.1	7.1	LS 2 punch		4	of WWD with level 5
	5	0.1	0.8	LS 2 punch		5	punches
	6	0.2	1.7	LPN 11		6	
	7	0.6	3.0	ADN 10		7	
	0	7.0	7.7	PRINT (H)		0	
	1	2.3	1.2	LDB 10 loc		1	output to tape 2 bits
	2	0.2	1.7	LPN 11		2	of WWD
	3	7.6	7.7	PRINT (H)		3	
	4	5.7	1.7	RAB 11 loc		4	Increment ML and jump
	5	6.5	3.0	NZB 20 loc		5	to *1
	6	0.0	0.0			6	
	7					7	

PROGRAM  
ROUTINE

PAGE

METHODS OF TESTING A SWITCH AND JUMPING TO APPROPRIATE ROUTINE PHASES

METHOD A				METHOD B			
LOC	OP	ADR		LOC	OP	ADR	
0	PRG5WT		** SWITCH	0	PRG5WT		** SWITCH
1	LDM	T1		1	LDM	T1	
2		PRG5WT		2		PRG5WT	
3	ADN			3	SBN		
4		PRGPHS		4		1	
5	STH			5	ZJP	T1	
6		SCRTCH		6		PRG1	
7	LDI	T1		7	SBN		
8		SCRTCH		8		1	
9	STH	T1		9	ZJP	T1	
10		PRGPHS		10		PRG2	
11	ZJP	T1		11	SBN		
12	PRGPHS			12		1	
13	PRG1		** ROUTINE PHASE 1	13	ZJP	T1	
14	PRG2		" " 2	14		PRG3	
15	PRG3		" " 3	15	PRG4		
16	PRG4		" " 4				

A COMPARISON OF STORAGE REQUIREMENTS

A = 13 + PHASES  
 B = 4(PHASES) - 1

ROUTINE PHASES	STORAGE ID	
	A	B
3	16	11
4	17	15
5	18	19
6	19	23
7	20	27
8	21	31

A COMPARISON OF TIME TO GET TO PHASE

A = 4(17)  
 B = 4[3 + (2+1)(PHASE-1)]

ROUTINE PHASE	TIME, μSEC	
	A	B
3	68	28 44 36
4	68	28 44 48
5	68	28 44 60
6	68	28 44 72
7	68	28 44 84
8	68	28 44 96



0000  
0000 0064  
0001 0137

CON 0  
UJP

CHECK

T/P SEND NICK AND DEL 1/16/64

GO TO ROUTINE CHECK

			CON	60	
0	0060	0011	WBHD	11	WRITE, BINARY, HIGH DENSITY (ODD)
	0061	0013	WCHD	13	WRITE( CODED, HIGH DENSITY (EVEN)
	0062	0010	WBLD	10	WRITE BINARY LOW DENSITY (ODD)
	0063	0012	WCLD	12	WRITE, CODED, LOW DENSITY (EVEN)
	0064	0000	SAVE		WRITE CODE, IF MAG/TAPE INPUT
	0065	0000	SAVE1		RECORD LENGTH OF READ
	0066	0000	SAVE2		TYPE OF TRANSMISSION
	0067	0000	TIMES		NUMBER OF OUTPUT FROM BUFFER
	0070	0000	TIMES1		NUMBER OF OUTPUTS COMPLEMENTED
	0071	0000	TIMES2		NUMBER OF OUTPUTS COMPLEMENTED FOR RESE
	0072	0000	TIMES3		USED FOR NUMBER OF OUTPUTS ONLY
	0073	0000	TEMP		WORKING AREA
	0074	0000	TEMP1		WORKING AREA
	0075	0000	TEMP2		WORKING AREA
	0076	0000	TEMP3		WORKING AREA
	0077	0000	TEMP4		WORKING AREA
	0100				
			PRG	100	
	0100	0000	FLAG		SEND, XLATE AND RECIEVE WORD
	0101	0000	FLAG1		INPUT FLAG FOR 1ST TIMES
	0102	0000	FLAG2		OUTPUT FLAG FOR 1ST TIME
	0103	0000	FLAG3		MAG/TAPE INPUT FLAG
	0104	0360	BLOCK	360	NUMBER OF WORD FOR BUFFER AREA
	0105	0000	INFUT		FUTURE INPUT
	0106	0000	INFUT1		FUTURE INPUT
	0107	0000	INFUT2		FUTURE INPUT
0	0110	1412	INMAG	INPUT2	INPUT MAG/TAPE STARTING ADDRESS
	0111	0000	OUFUT		SAME AS INPUT
	0112	0000	OUFUT1		SAME AS INPUT
	0113	0000	OUFUT2		SAME AS INPUT
	0114	0000	OUTPR		STARTING ADDRESS LINE PRINTER OUTPUT
	0115	0000	OUTFU		FUTURE OUTPUT
	0116	0000	OUTMT		STARTING ADDRESS MAG/TAPE OUTPUT
	0117	0005	FILL	5	
	0120	0366	FILL1	366	

0121	0020	DELR	LDN		JUMP TO RECIEVE ROUTINE
0122	0007			7	
0123	0102		ATT	100	
0124	0164		UJP	100	
0125	3473			RCV	
0126	0020	INPUT	LDN		JUMP TO INPUT ROUTINE
0127	0003			3	
0130	0202		ATT	200	
0131	0264		UJP	200	
0132	0000	INPUT I			7777
0133	0020	OTPUT	LDN		JUMP TO OUTPUT ROUTINE
					LINE PRINTER 166
					MAG/TAPE
0134	0002		ATT		
0135	0261		UJP	NICK	
0136	0000	OTPUT I			7777

0137	0020	CHECK	LDN		ROUTINE CHECK
0140	0001			1	
0141	0302		ATT	300	SET TAG3
0142	0030		ADN		
0143	0001			1	
0144	0102		ATT	100	SET TAG 1
0145	0030		ADN		
0146	0001			1	
0147	0202		ATT	200	SET TAG 2
0150	0321		LDM	300	PANEL TRANSMISSION
0151	0403			TYPE	
0152	0010		LPN		YES JUMP
0153	0020			20	
0154	0061		NZP		
0155	0166			CHECKA	
0156	0075		EXF		
0157	0035			35	
0160	0001			1	
0161	0076		INA		
0162	0010		LPN		STOP**CTP**LAMP
0163	0022			22	
0164	0061		NZP		
0165	0175			NICK	PANEL ACTION
0166	0164	CHECKA	UJP	100	RETURN FROM REC//SEND
0167	1361	CHECKI		CHECK2	
0170	0020	DELS	LDN		
0171	0007			7	
0172	0102		ATT	T1	
0173	0164		UJP	T1	
0174	3420		SND		
0175	0075	NICK	EXF		
0176	0035			35	
0177	0000			0	
0200	0074		OTN		
0201	0100			100	
0202	0077		HLT		
0203	0075	VINCE	EXF		
0204	0035			35	
0205	0000			0	
0206	0074		OTN		
0207	0077		HLT		

			PRG	400		
0400	0275	SYNCH1		275	TRANSMISSION SYNCH-WORD 1	
0401	0275	SYNCH2		275	TRANSMISSION SYNCH-WORD 2	
0402	0000	COUNT			NUMBER OF WORDS TO SEND +1	
0403	0000	TYPE			TYPE OF TRANSMISSION SENDING	
0404	0000	LENGTH			IF MAG/TAPE, INPUT RECORD LENGTH	
0405	0000	NUMBER			NUMBER OF OUTPUTS AND WRITE MODE	
0406	0000	TYPE1			HARDWARE INPUT USED AND PANEL CHANGES	
0407	0000	BLR		240D	INPUT/OUTPUT AREA USED	
0767	0000	BLR		8D	AREA FOR PARITY CHECK, TRANSMISSION ON	

			PRG	1000	
1000	0020	PAN	LDN		PANEL ROUTINE
1001	0000			0	
1002	0041		STM		SET FLAGS TO STARTING POINT
1003	0100			FLAG	
1004	0041		STM		
1005	0101			FLAG1	
1006	0041		STM		
1007	0102			FLAG2	
1010	0041		STM		
1011	0103			FLAG3	
1012	0341		STM	300	
1013	0403			TYPE	
1014	0075	PAN1	EXF		PANEL STATUS NUMBER 1
1015	0035			35	
1016	0001			1	
1017	0076		INA		
1020	0163		NJP	100	SEND BUTTON PUSHED
1021	1027			PAN2	
1022	0001		SHA		
1023	0163		NJP	100	XLATE BUTTON PUSHED
1024	1033			PAN3	
1025	0064		UJP		WAIT REMOTE RESPONSE
1026	0121			DELR	
1027	0055	PAN2	RAO		SET FLAG FOR SEND
1030	0100			FLAG	
1031	0164		UJP	100	
1032	1037			PAN14	
1033	0020	PAN3	LDN		
1034	0002			2	
1035	0041		STM		SET FLAG FOR XLATE
1036	0100			FLAG	
1037	0075	PAN14	EXF		FIND INPUT/OUTPUT
1040	0035			35	
1041	0002			2	
1042	0076		INA		
1043	0041		STM		SAVE INPUT*OUTPUT BITS
1044	0073			TEMP	
1045	0010		LPN		SAVE INPUT BITS
1046	0007			7	
1047	0160		ZJB	100	MAG/TAPE FOR INPUT
1050	1105			PAN7	
1051	0001		SHA		
1052	0001		SHA		SHIFT BITS TO HIGH ORDER OF WORD
1053	0001		SHA		
1054	0001		SHA		
1055	0001		SHA		
1056	0163	PAN4	NJP	100	NEG; GET UNPUT

1057	1073		PAN5	
1060	0001	SHA		SHIFT INPUT BIT
1061	0041	STM		
1062	0074		TEMP1	SAVE INPUT BIT
1063	0155	RAO	100	ADD ON TO START OF INPUT
1064	1074		PAN6	
1065	0155	RAO	100	ADD ONE TO START OF OUTPUT
1066	1161		PAN13	
1067	0021	LDM		LOAD INPUT BIT
1070	0074		TEMP1	
1071	0164	UJP	100	TRY AGAIN
1072	1056		PAN4	
1073	0021	LDM		LOAD STARTING ADDRESS
1074	0105		PAN6	STORE STARTING ADDRESS
1075	0041	STM		
1076	0132		INPUT1	
1077	0020	LDM		SET STARTING ADDRESS BACK
1100	0105		INFUT	
1101	0141	STM	100	
1102	1074		PAN6	STORE STARTING ADDRESS BACK
1103	0164	UJP	100	
1104	1117		PAN8	
1105	0021	LDM		GET STARTING ADDRESS OF INPUT
1106	0110		INMAG	
1107	0041	STM		STORE STARTING FOR JUMP
1110	0132		INPUT1	
1111	0055	RAO		SET FLAG FOR MAG/TAPE INPUT
1112	0103		FLAG3	
1113	0021	LDM		
1114	0116		OUTMT	
1115	0041	STM		SET STARTING ADDRESS FOR OUTPUT
1116	0136		OUTPUT1	
1117	0021	LDM		BRING OUTPUT BITS BACK
1120	0073		TEMP	
1121	0010	LPN		SAVE OUTPUT BITS
1122	0070		70	
1123	0001	SHA		
1124	0001	SHA		
1125	0160	ZJB	100	OUTPUT SAME AS INPUT
1126	1154		PAN12	
1127	0163	NJP	100	JUMP TO LOAD OUTPUT
1130	1142		PAN10	
1131	0001	SHA		
1132	0041	STM		
1133	0074		TEMP1	
1134	0155	RAO	100	ADD ONE TO START ADDRESS
1135	1143		PAN11	
1136	0021	LDM		BRING OUTPUT BITS BACK

1137	0074		TEMP1	
1140	0164		100	TRY AGAIN
1141	1127		PAN9	
1142	0021	PAN10	LDM	LOAD OUTPUT ADDRESS
1143	0114	PAN11	OUTPR	
1144	0041		STM	STORE OUTPUT ADDRESS
1145	0136		OTPUT1	
1146	0020		LDN	SET STARTING ADDRESS
1147	0114		OUTPR	
1150	0141		STM	100 STORE STARTING ADDRESS
1151	1143		PAN11	
1152	0161		NZP	100 GO TO ROUTINE SET
1153	1172		SET	
1154	0021	PAN12	LDM	
1155	0103		FLAG3	
1156	0161		NZP	100 OUTPUT ADDRESS WAS SET
1157	1172		SET	
1160	0021		LDM	
1161	0111	PAN13	OUFUT	
1162	0041		STM	
1163	0136		OTPUT1	
1164	0020		LDN	
1165	0111		OUFUT	
1166	0141		STM	100
1167	1161		PAN13	
1170	0064		UJP	
1171	0175		NICK	



1172	0020	SET	LDN		ROUTINE SET
1173	0001			1	SET INPUT FLAG
1174	0041		STM		
1175	0101			FLAG1	
1176	0041		STM		SET OUPUT FLAG2
1177	0102			FLAG2	
1200	0021		LDM		CHECK IF INPUT MAG/TAPE
1201	0103			FLAG3	
1202	0160		ZJB	100	INPUT OTHER THEN MAG/TAPE
1203	1345			SET8	
1204	0075		EXF		GET READ STATUS FOR WRITE
1205	0011			11	
1206	0002			2	
1207	0076		INA		SAVE STATUS
1210	0041		STM		
1211	0075			TEMP2	
1212	0001		SHA		SHIFT WORD
1213	0163		NJP	100	BINARY READ SET
1214	1221			SET1	
1215	0020		LDN		
1216	0002			2	
1217	0151		RAM	100	ADVANCE COUNT BY TWO
1220	1232			SET3	CODED READ SET
1221	0021	SET1	LDM		
1222	0075			TEMP2	
1223	0010		LPN		
1224	0001			1	
1225	0160		ZJB	100	(ODD) PARITY SET
1226	1231			SET2	
1227	0155		RAO	100	ADVANCE COUNT BY ONE
1230	1232			SET3	(EVEN) PARITY SET
1231	0021	SET2	LDM		
1232	0060	SET3		WBHD	
1233	0341		STM	300	STORE IN HEADER FOR TRANSMISSION
1234	0405			NUMBER	
1235	0020		LDN		
1236	0060			WBHD	SET START ADDRESS BACK
1237	0141		STM	100	
1240	1232			SET3	
1241	0075		EXF		READ ONE RECORD
1242	0011			11	
1243	0034			34	
1244	0372		INN	300	
1245	0007			7	FWA ADDERSS OF BUFFER
1246	0367			367	
1247	0034		SBN		SUB ONE FROM LAS GOOD RECORD
1250	0001			1	
1251	0041		STM		SAVE RECORD LENGTH

1252	0065		SAVE I	
1253	0241	STM	200	
1254	1462		INPUT7	STORE LENGTH IN INPUT ROUTINE
1255	0341	STM	300	STORE LENGTH IN HEADER FOR TRANSMISSION
1256	0404		LENGTH	
1257	0020	LDM		NUMBER OF READS FROM
1260	0001		1	BUFFER AREA
1261	0041	STM		
1262	0067		TIMES	DIVIDE LENGTH INTO BLOCK
1263	0021	LDM		
1264	0065		SAVE I	
1265	0163	SET4 NJP	100	FIND NUMBER OF READS
1266	1303		SETS	
1267	0001	SHA		AND STORE IN HEADER
1270	0041	STM		
1271	0065		SAVE I	
1272	0021	LDM		
1273	0067		TIMES	LOAD COUNT
1274	0001	SHA		SHIFT COUNT
1275	0041	STM		STORE COUNT BACK
1276	0067		TIMES	
1277	0021	LDM		LOAD LENGTH
1300	0065		SAVE I	
1301	0164	UJP	100	TRY AGAIN
1302	1265		SET4	
1303	0021	SET5 LDM		BLOCK==360
1304	0104		BLOCK	
1305	0035	SET6 SBM		SUB SHIFT LENGTH
1306	0065		SAVE I	
1307	0163	NJP	100	
1310	1333		SET7	
1311	0160	ZJB	100	
1312	1333		SET7	
1313	0041	STM		SAVE FOR NEXT
1314	0074		TEMPI	
1315	0335	SBM	300	TRY AGAIN
1316	0404		LENGTH	
1317	0163	NJP	100	
1320	1333		SET7	
1321	0055	RAO		ADD ONE TO COUNT
1322	0067		TIMES	
1323	0321	LDM	300	
1324	0404		LENGTH	
1325	0041	STM		STORE WORD LENGTH
1326	0065		SAVE I	
1327	0021	LDM		
1330	0074		TEMPI	
1331	0164	UJP	100	TRY AGAIN

1332	1305			SET6	
1333	0021	SET7	LDM		NUMBER OF OUTPUTS FROM BUFFER AREA
1334	0067			TIMES	
1335	0001		SHA		
1336	0001		SHA		
1337	0351		RAM	300	SET TIMES IN HEADER
1340	0405			NUMBER	
1341	0020		LDN		
1342	0002			2	
1343	0341		STM	300	SET TYPE OF INPUT IN HEADER
1344	0406			TYPE1	
1345	0020	SET8	LDN		
1346	0301			301	
1347	0341		STM	300	START OF TRANSMISSION
1350	0403			TYPE	
1351	0041		STM		SAVE TYPE OF TRANSMISSION
1352	0066			SAVE2	
1353	0025		LCM		COMPLEMENT NUMBER OF TIMES
1354	0067			TIMES	
1355	0041		STM		STORE FOR TEST
1356	0070			TIMES1	
1357	0041		STM		STORE FOR RESET
1360	0071			TIMES2	

1361	0021	CHECK2	LDM		TEST IF XLATE
1362	0100			FLAG	
1363	0034		SBN		
1364	0002			2	
1365	0260		ZJB	200	COMPUTER SET TO XLATE
1366	1412			INPUT2	
1367	0321		LDM	300	
1370	0403			TYPE	
1371	0160		ZJB	100	NO ACTION GO BACK TO PAN
1372	1000			PAN	
1373	0021		LDM		
1374	0100			FLAG	
1375	0060		ZJB		COMPUTER IN RECIEVING MODE
1376	0175			NICK	
1377	0321		LDM	300	
1400	0403			TYPE	
1401	0001		SHA		
1402	0263		NJP	200	TRANSMISSION O.K. SEND MORE
1403	1412			INPUT2	
1404	0001		SHA		TRANSMISSION
1405	0063		NJP		NO GOOD RE SEND
1406	0175			NICK	
1407	0001		SHA		
1410	0063		NJP		PANEL ACTION ON REPLY
1411	0175			NICK	

1412	0020	INPUT2	LDN		
1413	0000			0	
1414	0341		STM	300	
1415	0402			COUNT	
1416	0341		STM	300	ZERO OUTPUT NUMBER
1417	0405			NUMBER	
1420	0041		STM		RESET NUMBER OUTPUTS.
1421	0072			TIMES3	
1422	0341		STM	300	RESET TYPE1 TO ZERO
1423	0406			TYPE1	
1424	0021		LDM		ROUTINE INPUT MAG/TAPE
1425	0101			FLAG1	
1426	0260		ZJB	200	CHECK FLAG1 FOR 1ST TIME
1427	1437			INPUT5	
1430	0020		LDN		SET START OF MESS
1431	0203			203	
1432	0341		STM	300	STORE S.O.M. IN HEADER
1433	0403			TYPE	
1434	0075	INPUT4	EXF		REWIND TO LOAD POINT
1435	0011			11	
1436	0022			22	
1437	0075	INPUT5	EXF		STATUS ON TAPE/HANDLER NO.1
1440	0011			11	
1441	0002			2	
1442	0076		INA		
1443	0010		LPN		SAVE BUSY AND NOT READY BITS
1444	0202			202	
1445	0260		ZJB	200	
1446	1455			INPUT6	
1447	0010		LPN		
1450	0200			200	
1451	0260		ZJB	200	
1452	1455			INPUT6	
1453	0064		UJP		TAPE UNIT NOT READY
1454	0175			NICK	
1455	0075	INPUT6	EXF		READ RECORD
1456	0011			11	
1457	0034			34	
1460	0372		INN	300	
1461	0007	INPUTA		7	
1462	0000	INPUT7			INPUT AREA FOR READ
1463	0075	INPUT8	EXF		
1464	0011			11	STATUS OF READ
1465	0002			2	
1466	0076		INA		
1467	0041		STM		SAVE STATUS
1470	0074			TEMP1	
1471	0010		LPN		

1472	0200		200	
1473	0261	NZP	200	WAIT TILL READY
1474	1463		INPUT8	
1475	0021	LDM		BRING BACK STATUS
1476	0074		TEMP1	
1477	0010	LPN		
1500	0024		24	
1501	0260	ZJB	200	MORE READS NEEDED
1502	1511		INPUT9	
1503	0010	LPN		
1504	0004		4	
1505	0061	NZP		PARITY ON READ
1506	0175		NICK	
1507	0064	UJP		END OF FILE
1510	0175		NICK	
1511	0321	INPUT9 LDM	300	
1512	0404		LENGTH	
1513	0351	RAM	300	ADD ONE TO COUNTER
1514	0402		COUNT	
1515	0055	RAO		SAVE NUMBER OF TIMES
1516	0072		TIMES3	
1517	0055	RAO		BUFFER FILLED
1520	0070		TIMES1	
1521	0260	ZJB	200	YES JUMP
1522	1535		INPUT8	
1523	0221	LDM	200	
1524	1462		INPUT7	
1525	0241	STM	200	ADVANCE STARTING AND ENDING ADDRESS
1526	1461		INPUTA	
1527	0321	LDM	300	
1530	0404		LENGTH	
1531	0251	RAM	200	
1532	1462		INPUT7	
1533	0264	UJP	200	
1534	1437		INPUT5	READ AGAIN
1535	0020	INPUT8 LDM		STARTING ADDRESS OF INPUT AREA
1536	0007		7	
1537	0241	STM	200	1ST WORD FOR BUFFER
1540	1461		INPUTA	
1541	0321	LDM	300	
1542	0404		LENGTH	LAST WORD FOR BUFFER
1543	0241	STM	200	
1544	1462		INPUT7	
1545	0021	LDM		SET COUNTR TO START
1546	0071		TIMES2	
1547	0041	STM		
1550	0070		TIMES1	
1551	0021	LDM		SET NUMBER OF OUTPUTS

1552	0072			TIMES3	
1553	0001		SHA		
1554	0001		SHA		
1555	0341		STM	300	
1556	0405			NUMBER	
1557	0020	INPUTC	LDN		FILL INPUT/OUTPUT AREA
1560	0200			200	
1561	0351		RAM	300	
1562	0405	INPUTD		NUMBER	
1563	0255		RAO	200	
1564	1562			INPUTD	
1565	0035		SBM		
1566	0120			FILLI	
1567	0261		NZP	200	
1570	1557			INPUTC	
1571	0020		LDN		
1572	0405			NUMBER	
1573	0241		STM	200	
1574	1562			INPUTD	
1575	0021		LDM		TEST 1ST TIME FLAG
1576	0101			FLAGI	
1577	0261		NZP	200	
1600	1605			INPUTE	
1601	0020		LDN		SET TYPE IN TRANSMISSIOM
1602	0203			203	
1603	0341		STM	300	
1604	0403			TYPE	
1605	0020	INPUTE	LDN		SET FLAG BACK TO ZERO
1606	0000			0	
1607	0041		STM		
1610	0101			FLAGI	
1611	0020		LDN		
1612	0010			10	
1613	0351		RAM	300	
1614	0402			COUNT	
1615	0021		LDM		SEND OR XLATE MODE
1616	0100			FLAG	
1617	0034		SBN		
1620	0001			1	
1621	0060		ZJB		SEND MODE JUMP
1622	0170			DELS	
1623	0064		UJP		XLATE MODE JUMP.
1624	0175			NICK	

## 8062 T/P TRANSMISSION AND ERROR DECTION

- \*\* T1 = TAG REGISTER 1 USED  
IN THE EXECUTION OF EDC,  
SND, RCV AND CHR ROUTINES.  
THIS IS SET BY THE ROUTINE  
WHICH GIVES CONTROL TO EDC,  
SND, RCV AND CHR ROUTINES.  
FOR EDC T1 = 06  
FOR OTHERS T1 = 07
- \*\* T2 = TAG REGISTER 2 USED  
IN ADDRESSING ERROR  
DETECTION CODES. THIS IS  
SET BY CHR OR SND  
ROUTINES.  
ON TRANSMISSION T2 = T3  
ON RECEPTION T2 = T1
- \*\* T3 = TAG REGISTER 3 USED  
IN ADDRESSING CHARACTERS  
IN THE BLOCK WHICH IS  
TO BE TRANSMITTED. THIS  
IS SET BY SND OR RCV  
ROUTINES.  
T3 = 01



INSTRUCTIONS AT 30 INTERRUPT

0030	0020	LDN	30
0031	0006		6
0032	0103	TTA	TI
0033	0164	UJP	TI
0034	3400	INT30	

\*\* SET TAGI TO 6, AND GO TO INT30

		ORG	3000		ERROR DECTION CODE CALCULATIONS
3000	0000	M			**MATRIX ELEMENT COUNTER
3001	0020	EDC	LDM		**INITIALIZE M AT ZERO
3002	0000			0	
3003	0141		STM	T1	
3004	3000		M		
3005	0020		LDM		**INITIALIZE EDC S AT ZERO
3006	0000			0	
3007	0241		STM	T2	
3010	0367	ZROEDC		367	
3011	0155		RAO	T1	
3012	3010		ZROEDC		
3013	0161		NZP	T1	
3014	3005		ZROEDC	-3	
3015	0020		LDM		**INITIALIZE FIRST MATRIX ROW
3016	3210		MROWI		EXAMINATION
3017	0141		STM	T1	
3020	3022		ROW		
3021	0121	EDC1	LDM	T1	**LOAD ROW
3022	0000	ROW			
3023	0001	EDC2	SHA		**SHIFT ROW ONE BIT
3024	0141		STM	T1	AND STORE IN TROW
3025	3247		TROW		
3026	0010		LPN		**MASK ALL BUT BIT ONE
3027	0001			1	IF ZERO GO TO EDC4
3030	0160		ZJB	T1	
3031	3074		EDC4		
3032	0121		LDM	T1	**INITIALIZE CHAR AT M
3033	3000		M		
3034	0141		STM	T1	
3035	3053		CHAR		
3036	0020		LDM		**INITIALIZE OLDEDC, NEWEDC,
3037	0367			367	CODE R AND CODES AT 367
3040	0141		STM	T1	
3041	3051		OLDEDC		
3042	0141		STM	T1	
3043	3055		NEWEDC		
3044	0141		STM	T1	
3045	3141		CODER		
3046	0141		STM	T1	
3047	3143		CODES		
3050	0221	EDC3	LDM	T2	**CALCULATE ERROR DETECTION
3051	0000	OLDEDC			CODE
3052	0315		LSM	T3	
3053	0000	CHAR			
3054	0241		STM	T2	
3055	0000	NEWEDC			

3056	0155		RAO	TI	**INCREMENT CHAR BY ONE,
3057	3053		CHAR		IF CHAR GREATER OR
3060	0335		SBM	T3	EQUAL COUNT GO TO EDC4
3061	0402		COUNT		
3062	0160		ZJB	TI	
3063	3074		EDC4		
3064	0155		RAO	TI	**INCREMENT OLDEDC, IF
3065	3051		OLDEDC		ZERO GO TO EDC4, IF
3066	0160		ZJB	TI	NOT ZERO INCREMENT
3067	3074		EDC4		NEWEDC, GO TO EDC3
3070	0141		STM	TI	
3071	3055		NEWEDC		
3072	0164		UJP	TI	
3073	3050		EDC3		
3074	0155	EDC4	RAO	TI	**INCREMENT M BY ONE, IF
3075	3000		M		M GREATER OR EQUAL COUNT
3076	0335		SBM	T3	GO TO EDC6
3077	0402		COUNT		
3100	0160		ZJB	TI	
3101	3126		EDC6		
3102	0121		LDM	TI	**IF M GREATER OR EQUAL
3103	3000		M		TO 361 GO TO EDC6
3104	0034		SBN		
3105	0367			367	
3106	0160		ZJB	TI	
3107	3126		EDC6		
3110	0121		LDM	TI	**IF M IS DIVISABLE BY
3111	3000		M		8 GO TO EDC5
3112	0010		LPN		
3113	0007			7	
3114	0160		ZJB	TI	
3115	3122		EDC5		
3116	0121		LDM	TI	**LOAD TROW AND GO TO
3117	3247		TROW		EDC2
3120	0164		UJP	TI	
3121	3023		EDC2		
3122	0155	EDC5	RAO	TI	**INCREMENT ROW BY ONE
3123	3022		ROW		AND GO TO EDC1
3124	0164		UJP	TI	
3125	3021		EDC1		
3126	0203	EDC6	TTA	T2	**IF TAG 2 NOT EQUAL
3127	0034		SBN		ONE GO TO EDC7
3130	0001			1	
3131	0161		NZP	TI	
3132	3140		EDC7		
3133	0020		LDN		**IF TAG 2 EQUAL ONE
3134	0007			7	GO TO SND1
3135	0102		ATT	TI	

03136	0164		UJP	TI	
3137	3431		SNDI		
3140	0121	EDC7	LDM	TI	**COMPARE CALCULATED
3141	0000	CODER			EDCS ON RECEIPT
3142	0335		SBM	T3	WITH EDCS SENT IF
3143	0000	CODES			NOT EQUAL GO TO EDC10
3144	0161		NZP	TI	
3145	3156		EDC10		
3146	0155		RAO	TI	**INCREMENT CODER AND
3147	3141		CODER		CODES. IF NOT ZERO
3150	0141		STM	TI	GO TO EDC7. IF ZERO
3151	3143		CODES		GO TO EDC11.
3152	0161		NZP	TI	
3153	3140		EDC7		
3154	0164		UJP	TI	
3155	3173		EDC11		
3156	0020	EDC10	LDN		**STORE TRANS NG CODE
3157	0241			241	IN TYPE CHAR OF
3160	0341		STM	T3	HEADER ETC. AND GO TO
3161	0403		TYPE		SND ROUTINE.
3162	0020		LDN		
3163	0010			10	
3164	0341		STM	T3	
3165	0402		COUNT		
3166	0020		LDN		
3167	0007			7	
03170	0102		ATT	TI	
3171	0164		UJP	TI	
3172	3420		SND		
3173	0021	EDC11	LDM	0	**GO TO SET UP TO
3174	0403		TYPE		GIVE CONTROL TO
3175	0060		ZJB	0	NICK IF COMPUTER
3176	0137		CHECK		IS SENDING OR IDLE
3177	0021		LDM	0	**SET UP TO GIVE
3200	0100		FLAG		CONTROL TO VINCE
3201	0061		NZP	0	IF COMPUTER IS IN
3202	0137		CHECK		RECEIVE
3203	0020		LDN		**GO TO VINCE
3204	0000			0	
3205	0102		ATT	TI	
3206	0164		UJP	TI	
3207	0203		VINCE		
3210	0261	MROWI		261	**FIRST ROW OF MATRIX
3211	0350			350	
3212	0177			177	
3213	0220			220	
3214	0247			247	
3215	0325			325	

3216	0160	160
3217	0142	142
3220	0263	263
3221	0057	57
3222	0336	336
3223	0156	156
3224	0345	345
3225	0112	112
3226	0045	45
3227	0243	243
3230	0071	71
3231	0343	343
3232	0141	141
3233	0027	27
3234	0136	136
3235	0337	337
3236	0015	15
3237	0065	65
3240	0165	165
3241	0004	4
3242	0354	354
3243	0223	223
3244	0003	3
3245	0224	224
3246	0160	160
3247	0000	TROW

LAST ROW OF MATRIX  
\*\*TEMP. ROW STORAGE

		ORG		3400		TEST TRANSMISSION STATE	
3400	0075	INT30	EXF			** REQUEST DCU STATUS	
3401	0036				36		
3402	0003				3		
3403	0141		STM		TI	** SAVE STATUS	
3404	3412		STATUS				
3405	0034		SBN			** IF STATUS SENDING/REQUEST	
3406	0005				5	GO TO INT30S	
3407	0160		ZJP		TI		
3410	3463		INT30S				
3411	0020		LDN			** IF STATUS RECEIVING/READY	
3412	0000	STATUS				GO TO INT30R	
3413	0034		SBN				
3414	0012				12		
3415	0160		ZJP		TI		
3416	3524		INT30R				
3417	0000		ERR			** ERROR	

3420	0020	SND	LDN		SEND ROUTINE
3421	0001			I	** SET TAGS 2 AND 3 TO 1
3422	0202		ATT	T2	
3423	0302		ATT	T3	
3424	0020		LDN		** GO TO EDC
3425	0006			6	
3426	0102		ATT	T1	
3427	0164		UJP	T1	
3430	3001		EDC		
3431	0020	SND1	LDN		** SET C TO ZERO
3432	0000			0	
3433	0141		STM	T1	
3434	3632		C		
3435	0075		EXF		** REQUEST SEND DCU
3436	0036			36	
3437	0001			I	
3440	0075	SND2	EXF		** REQUEST STATUS DCU
3441	0036			36	
3442	0003			3	
3443	0076		INA		** IF STATUS NOT SENDING
3444	0010		LPN		GO TO SND1
3445	0001			I	
3446	0161		NZP	T1	
3447	3431		SND1		
3450	0321	SND3	LDM	T3	** SET UP CHAR TO BE SENT
3451	3632		C		
3452	0141		STM	T1	
3453	3464		SNDWD		
3454	0075	IDLE	EXF		** IDLE LOOP
3455	0036			36	
3456	0005			5	
3457	0013		CIL		
3460	0002		ATT		
3461	0164		UJP	T1	
3462	3454		IDLE		
3463	0074	INT30S	OTN		** OUTPUT
3464	0000	SNDWD			
3465	0121		LDM	T1	** PERFORM CHR ROUTINE
3466	3450		SND3		
3467	0141		STM	T1	
3470	3617		RETURN		
3471	0164		UJP	T1	
3472	3564		CHR		

3473	0020	RCV	LDN		RECEIVE ROUTINE
3474	0007			7	** SET TAG 2 TO 7
3475	0202		ATT	T2	
3476	0020		LDN		** SET TAG 3 TO 1
3477	0001			1	
3500	0302		ATT	T3	
3501	0020		LDN		** SET ALPHA AND C TO ZERO
3502	0000			0	
3503	0141		STM	T1	
3504	3633		ALPHA		
3505	0141		STM	T1	
3506	3632		C		
3507	0075		EXF		** STOP SENDING DCU
3510	0036			36	
3511	0000			0	
3512	0075		EXF		** REQUEST STATUS DCU
3513	0036			36	
3514	0003			3	
3515	0076		INA		** STATUS EQUAL ZERO
3516	0060		ZJP		GO TO CHECK (NICK)
3517	0137		CHECK		
3520	0034		SBN		** STATUS NOT RECEIVING/READY
3521	0012			12	GO TO IDLE
3522	0161		NZP	T1	
3523	3454		IDLE		
3524	0075	INT30R	EXF		** REQUEST DATA DCU
3525	0036			36	
3526	0002			2	
3527	0076		INA		** INPUT
3530	0141		STM	T1	
3531	3634		RCVWD		
3532	0121		LDM	T1	** IF ALPHA NOT ZERO
3533	3633		ALPHA		GO TO ECVI
3534	0161		NZP	T1	
3535	3550		RCVI		
3536	0121		LDM	T1	** IF RCVWD EQUAL ZERO
3537	3634		RCVWD		GO TO IDLE
3540	0160		ZJP	T1	
3541	3454		IDLE		
3542	0034		SBN		** IF RCVWD NOT EQUAL
3543	0275			275	SOT(275) GO TO IDLE
3544	0161		NZP	T1	
3545	3454		IDLE		
3546	0155		RAO	T1	** SET ALPHA TO 1
3547	3633		ALPHA		
3550	0121	RCVI	LDM	T1	** SET UP TO STORE CHAR
3551	3632		C		



0 3552	0141		STM	TI	
3553	3557		RCVCHR		
3554	0121		LDM	TI	** STORE CHAR
3555	3634		RCVWD		
3556	0341		STM	T3	
3557	0000	RCVCHR			
3560	0020		LDM		** PERFORM CHR ROUTINE
3561	3454		IDLE		
3562	0141		STM	TI	
3563	3617		RETURN		

Address	Code	Label	Op	Mode	Comment
3564	0155	CHR	RAO	T1	CHARACTER COUNTING
3565	3632		C		**INCREMENT C BY 1
3566	0160		ZJB	T1	**C EQUAL ZERO GO TO CHR2
3567	3620		CHR2		
3570	0034		SBN		**C LESS THAN OR EQUAL
3571	0007			7	HEADER GO TO CHRI
3572	0163		NJP	T1	
3573	3616		CHRI		
3574	0121		LDM	T1	**C LESS THAN OR EQUAL
3575	3632		C		COUNT GO TO CHRI
3576	0335		SBM	T3	
3577	0402		COUNT		
3600	0160		ZJB	T1	
3601	3616		CHRI		
3602	0163		NJP	T1	
3603	3616		CHRI		
3604	0121		LDM	T1	**C GREATER THAN OR
3605	3632		C		EQUAL TO 247 GO TO
3606	0034		SBN		CHRI
3607	0367			367	
3610	0162		PJP	T1	
3611	3616		CHRI		
3612	0020		LDN		**C LESS THAN 247 SET
3613	0367			367	C EQUAL 247
3614	0141		STM	T1	
3615	3632		C		
3616	0164	CHRI	UJP	T1	**RETURN
3617	0000	RETURN			
3620	0203	CHR2	TTA	T2	**T2 EQUAL 1 GO TO
3621	0034		SBN		RCV
3622	0001			1	
3623	0160		ZJB	T1	
3624	3473		RCV		
3625	0020		LDN		**T2 NOT EQUAL 1 GO TO
3626	0006			6	EDC
3627	0102		ATT	T1	
3630	0164		UJP	T1	
3631	3001		EDC		

0 3632 0000 C  
3633 0000 ALPHA  
3634 0000 RCVWD

ROUTINE PARAMETERS  
MESSAGE CHARACTER COUNTER  
FLAG TO DETECT START OF TRANS.  
WORD FROM DCU

0100	T1	EQU	100		
0200	T2	EQU	200		
0300	T3	EQU	300		
0000	ERR	EQU		0	ERROR STOP
0001	SHA	EQU		1	SHIFT A LEFT ONE BIT
0002	ATT	EQU		2	A TO TAG REGISTER
0003	TTA	EQU		3	TAG REGISTER CONTENTS TO A
0004	ABR	EQU		4	A TO BUFFER ENTRANCE REGISTER
0005	ABX	EQU		5	A TO BUFFER EXIST REGISTER
0006	BER	EQU		6	CONTENTS OF BER REGISTER TO A
0007	CBC	EQU		7	CLEAR BUFFER CONTROLS
0010	LPN	EQU		10	LOGICAL PRODUCT NO ADDRESS
0011	LPM	EQU		11	LOGICAL PRODUCT MEMORY ADDRESS
0012	LPI	EQU		12	LOGICAL PRODUCT INDIRECT ADDRESS
0013	CIL	EQU		13	CLEAR INTERRUPT LOCKOUT
0014	LSN	EQU		14	LOGICAL SUM NO ADDRESS
0015	LSM	EQU		15	LOGICAL SUM MEMORY ADDRESS
0016	LSI	EQU		16	LOGICAL SUM INDIRECT ADDRESS
0020	LDN	EQU		20	LOAD A NO ADDRESS MODE
0021	LDM	EQU		21	LOAD A MEMORY
0022	LDI	EQU		22	LOAD A INDIRECT
0025	LCM	EQU		25	LOAD COMPLEMENT TO A MEMORY
0026	LCI	EQU		26	LOAD COMPLEMENT TO A INDIRECT
0030	ADN	EQU		30	ADD NO ADDRESS
0031	ADM	EQU		31	ADD MEMORY ADDRESS
0032	ADI	EQU		32	ADD INDIRECT ADDRESS
0034	SBN	EQU		34	SUBTRACT NO ADDRESS
0035	SBM	EQU		35	SUBTRACT MEMORY ADDRESS
0036	SBI	EQU		36	SUBTRACT INDIRECT ADDRESS
0041	STM	EQU		41	STORE A MEMORY
0042	STI	EQU		42	STORE A INDIRECT
0051	RAM	EQU		51	REPLACE ADD MEMORY ADDRESS
0055	RAO	EQU		55	REPLACE ADD ONE MEMORY ADDRESS
0060	ZJB	EQU		60	JUMP, IF CONTENTS OF A = 0
0060	ZJP	EQU		60	JUMP, IF CONTENTS OF A = 0
0061	NZP	EQU		61	JUMP, IF CONTENTS OF A ≠ 0
0062	PJP	EQU		62	JUMP, IF CONTENTS OF A 0 POSITIVE
0063	NJP	EQU		63	JUMP, IF CONTENTS OF A 0 NEGATIVE
0064	UJP	EQU		64	UNCONDITIONAL JUMP
0070	IBI	EQU		70	INITIATE BUFFER INPUT
0071	IBO	EQU		71	INITIATE BUFFER OUTPUT
0072	INN	EQU		72	INPUT NORMAL
0073	OUT	EQU		73	OUTPUT NORMAL
0074	OTN	EQU		74	OUTPUT, NO ADDRESS
0075	EXF	EQU		75	EXTERNAL FUNCTION
0076	INA	EQU		76	INPUT TO A
0077	HLT	EQU		77	HALT
0000	END				COMPLETE ASSEMBLY