Dick M

Course DA3020-1
CYBER CP COMPASS

STUDENT HANDOUT

PROPRIETARY NOTICE

REVISION RECORD

REVISION	DESCRIPTION
01	
(01, 1978)	Manual Release
A	
(04-01-78)	Manual Update
B	
(04-30-78)	Manual Update
C	
(03, 1979)	Manual Update
D	
(12, 1980)	Manual Update
Publication No. DA3020-1	

REVISION LETTERS I, O, Q AND X ARE NOT USED

CYBER CP COMPASS

Address comments concerning this manual to:

CONTROL DATA CORPORATION
National Coordinator
5001 West 80th Street
Bloomington, Minnesota 55437
Attn: Curtis Vicha

or use Comment Sheet in the back of this manual.

GENERAL OVERVIEW

COURSE TITLE: CYBER CP COMPASS

COURSE NUMBER: DA3020

COURSE LENGTH: 5 DAYS

DESCRIPTION:

THIS COURSE INTRODUCES THE STUDENT TO THE COMPASS ASSEMBLY LANGUAGE INCLUDING DATA FORMATS, MACRO INSTRUCTIONS AND SUBROUTINE PROGRAMMING AND LINKAGE ROUTINES.

PREREQUISITES:

DA2000 OR FA2000, CYBER 70/170 INTRODUCTIONS OR PREVIOUS COMPUTER EXPERIENCE.

COURSE OBJECTIVES:

UPON COMPLETION, THE STUDENT WILL BE EXPECTED TO WRITE AND DEBUG A COMPASS COMPUTATIONAL ROUTINE UTILIZING THE CP INSTRUCTION SET, DATA FORMATS, PSEUDO INSTRUCTIONS, PROGRAMMER MACROS AND INCORPORATING THE PASSING OF PARAMETERS AND LINKING OF SUBROUTINES.

COURSE CHART - CP COMPASS

1	Intro and Pretest	Review	Data Items	Error Exits	Conditional Assembly
2	Hardware and Job Overview	Floating Point Arithmetic	Pseudo Operations		
3	Symbol Definition	Floating Point Instructions		Macro's	Parameter Passing
4	Lunch	Lunch	Lunch	Lunch	Lunch
5	Instructions	Floating Point Examples	Pseudo Operations	OPDEFS Micro's	Parameter Passing
6	Lab 1	Lab 2	Lab 3	Lab 4	Lab 5
7					

CYBER CP COMPASS

COURSE OUTLINE

I. INTRODUCTION

II. CENTRAL PROCESSOR OVERVIEW

A. Job flow in system
B. Control Point
C. Registers
D. Arithmetic Unit - functional & unified
E. Instruction cycle

III. JOB OVERVIEW

A. Coding conventions
B. Instruction format
C. Job structure
D. Control Statements

IV. INSTRUCTIONS

A. Integer Arithmetic
B. Sets
C. Jumps
D. Boolean
E. Shifts

V. FLOATING POINT ARITHMETIC

A. Data Format
B. Conversion
C. Instructions

COURSE OUTLINE (CONT.)

VI. DATA ITEMS

A. DATA
B. CON
C. DIS
D. LITERAL
E. VFD

VII. PSEUDO OPERATIONS

A. EQU/SET
B. REP
C. LOC

VIII. ERROR EXITS

A. SPECIAL FORMS
B. EXIT MODES

IX. MACROS

A. OPDEF's
B. PSEUDO OPS
C. MICROS
D. MACROS

X. CONDITIONAL ASSEMBLY

A. IFC
B. IFPP
C. IF - TEST SYMBOL

XI. SUBROUTINES

A. PARAMETER PASSING
B. COMPASS ROUTINES
C. FORTRAN INTERFACE
D. COBOL INTERFACE

LESSON 1

INTRODUCTION

LESSON REVIEW:

THE OBJECTIVES AND FORMAT OF THE CLASS ARE DISCUSSED. THE STUDENTS ARE GIVEN A BRIEF PRETEST TO CHECK THEIR BACKGROUND.

REFERENCES:

NONE

TRAINING AIDS:

NONE

PROJECTS:

PRETEST IN STUDENT HANDOUT

OBJECTIVES:

AT THE COMPLETION OF THIS UNIT THE STUDENT WILL BE ABLE TO:

1. LIST THE OBJECTIVES OF THIS COURSE
2. STATE WHAT SKILLS HE/SHE EXPECTS FROM THIS COURSE.

LESSON 2

CENTRAL PROCESSOR OVERVIEW

LESSON PREVIEW:

THE COMPONENTS OF THE CENTRAL PROCESSOR AND HOW AN INSTRUCTION IS EXECUTED IS DISCUSSED. IN ADDITION A JOB IN EXECUTION IS TRACED THROUGH ITS FLOW IN THE SYSTEM.

REFERENCES:

CHAPTER 1	COMPASS REF. MAN. #60492600
CHAPTER 2	CYBER 170 HARDWARE REF. #60420000

TRAINING AIDS:

VISUAL SET V2

PROJECTS:

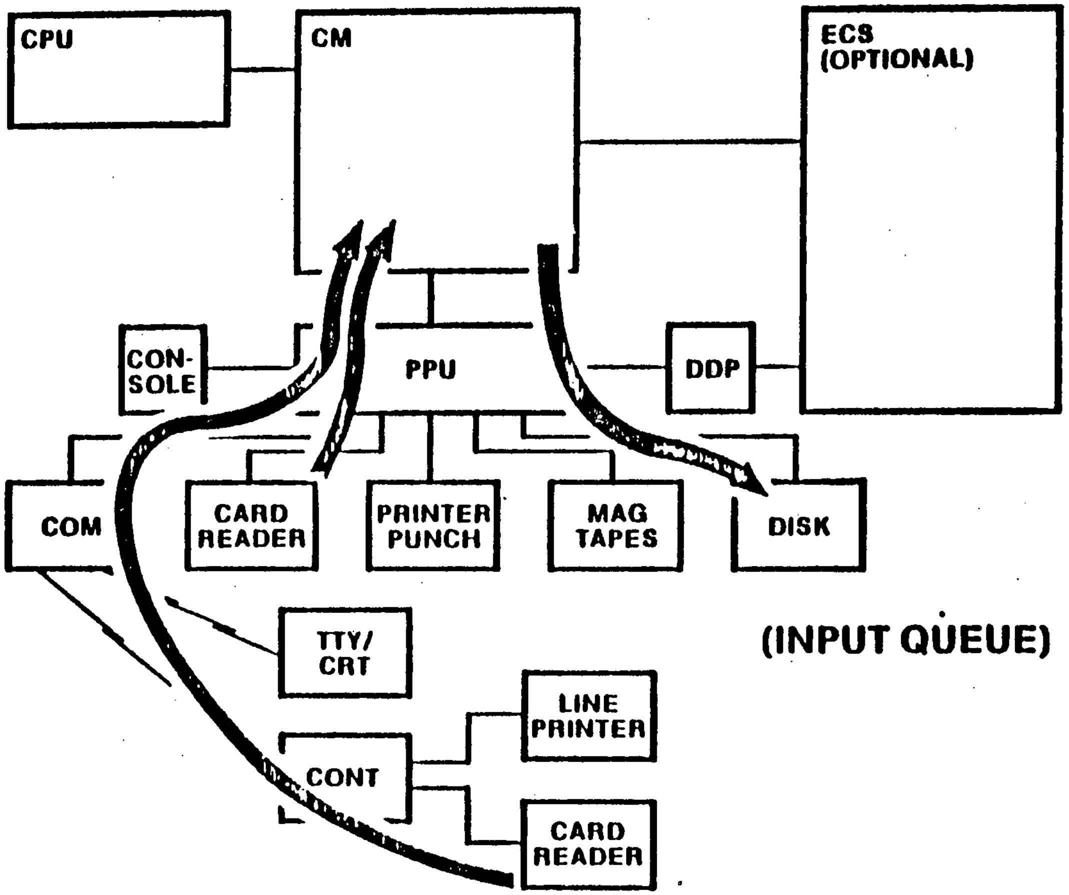
NONE

OBJECTIVES:

AT THE COMPLETION OF THIS LESSON THE STUDENT WILL BE ABLE TO:

1. DESCRIBE THE FIVE PHASES OF A JOB IN THE SYSTEM.
2. EXPLAIN THE CONTROL POINT.
3. LIST THE COMPONENTS AND THEIR FUNCTIONS OF THE CENTRAL PROCESSOR.
4. EXPLAIN HOW AN INSTRUCTION IS OPERATED ON BY THE COMPONENTS OF THE CENTRAL PROCESSOR.

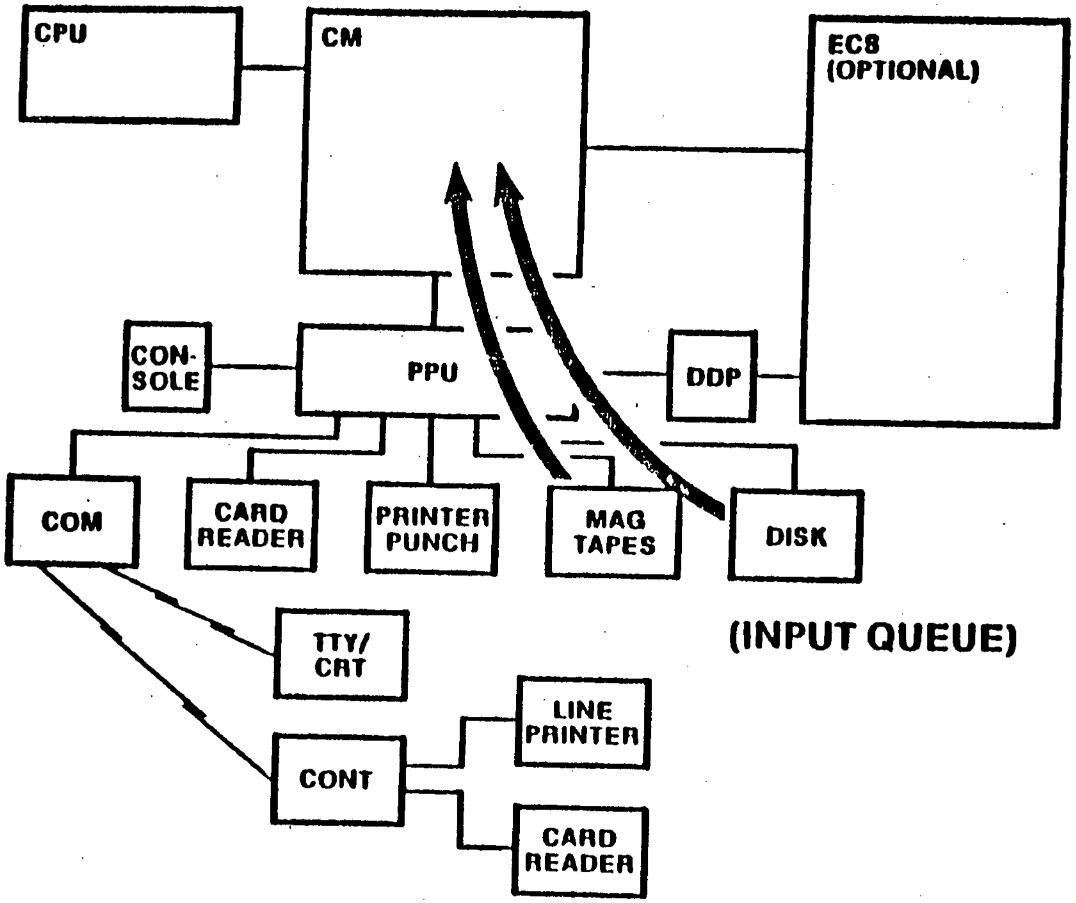
CDC CYBER 170 SERIES



(INPUT QUEUE)

BATCH JOB INPUT

CDC CYBER 170 SERIES



CPU

CM

ECS
(OPTIONAL)

CON-
SOLE

PPU

DDP

COM

CARD
READER

PRINTER
PUNCH

MAG
TAPES

DISK

(INPUT QUEUE)

TTY/
CRT

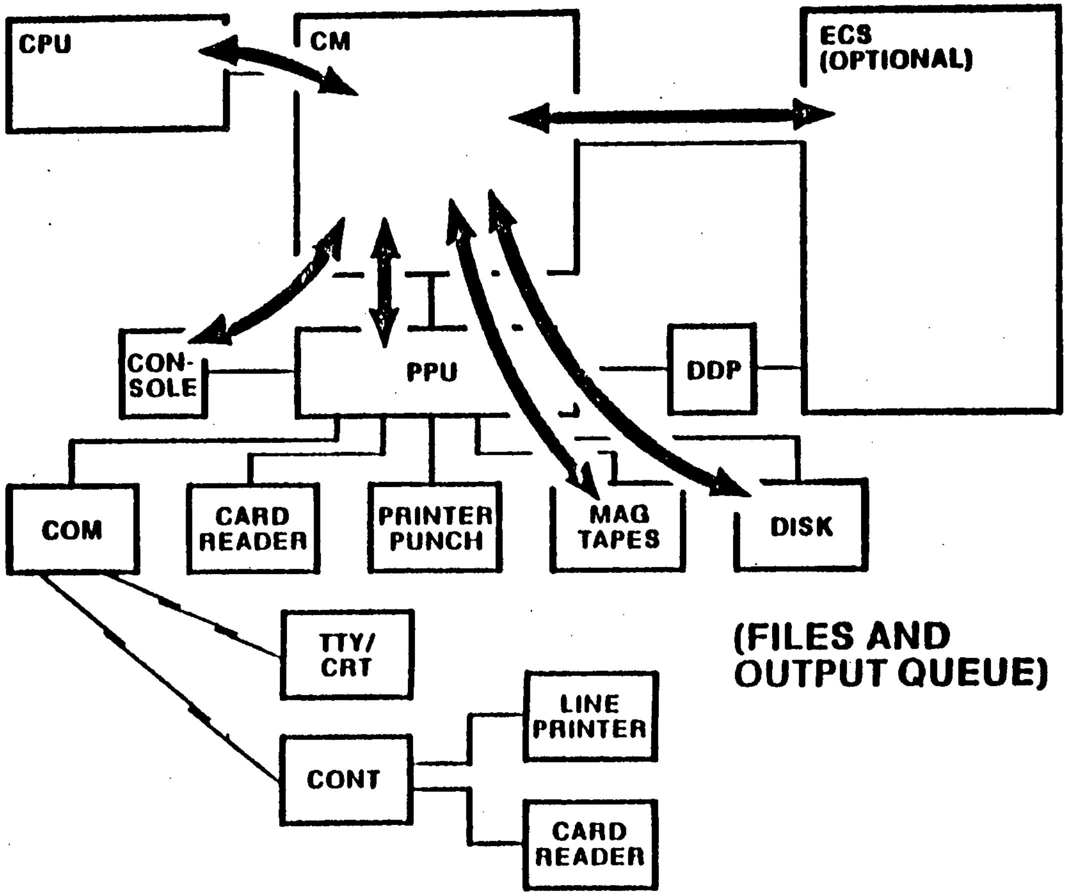
CONT

LINE
PRINTER

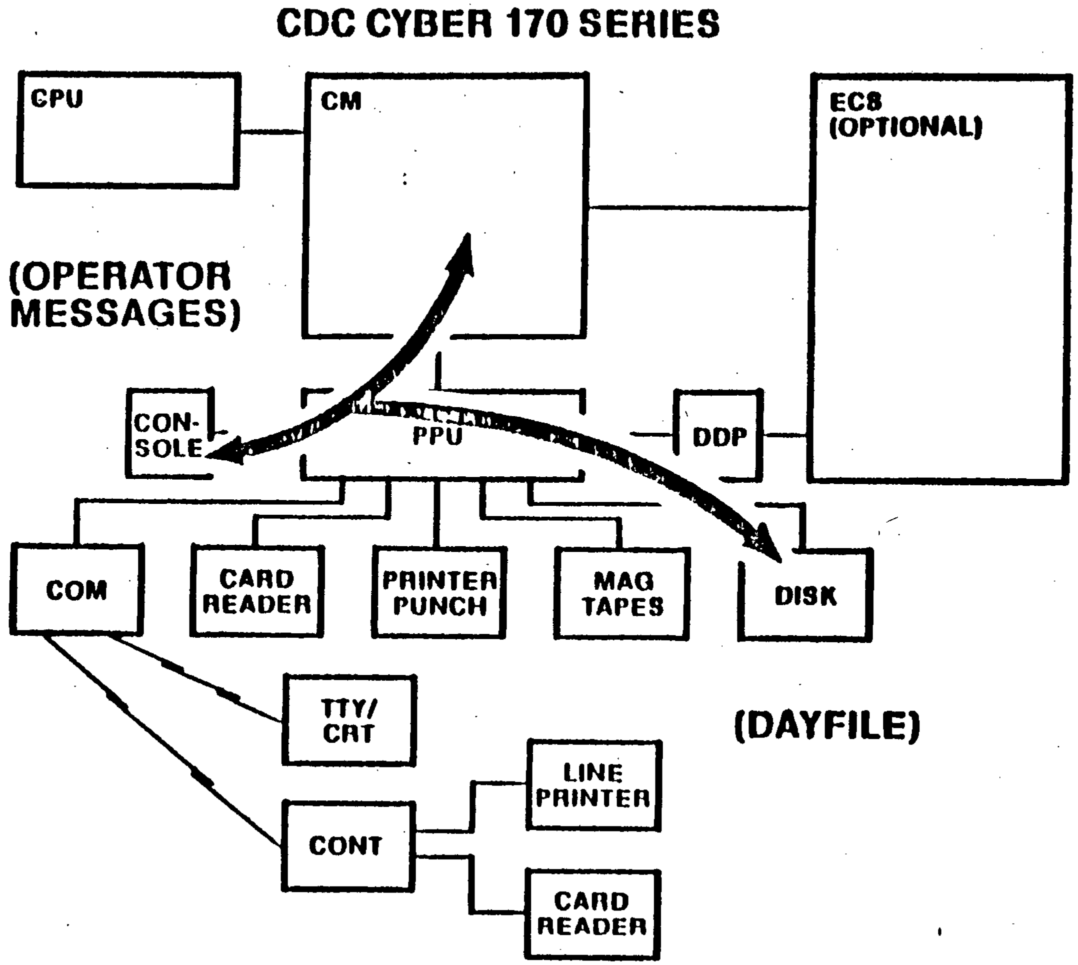
CARD
READER

LOADING A JOB FOR EXECUTION

CDC CYBER 170 SERIES

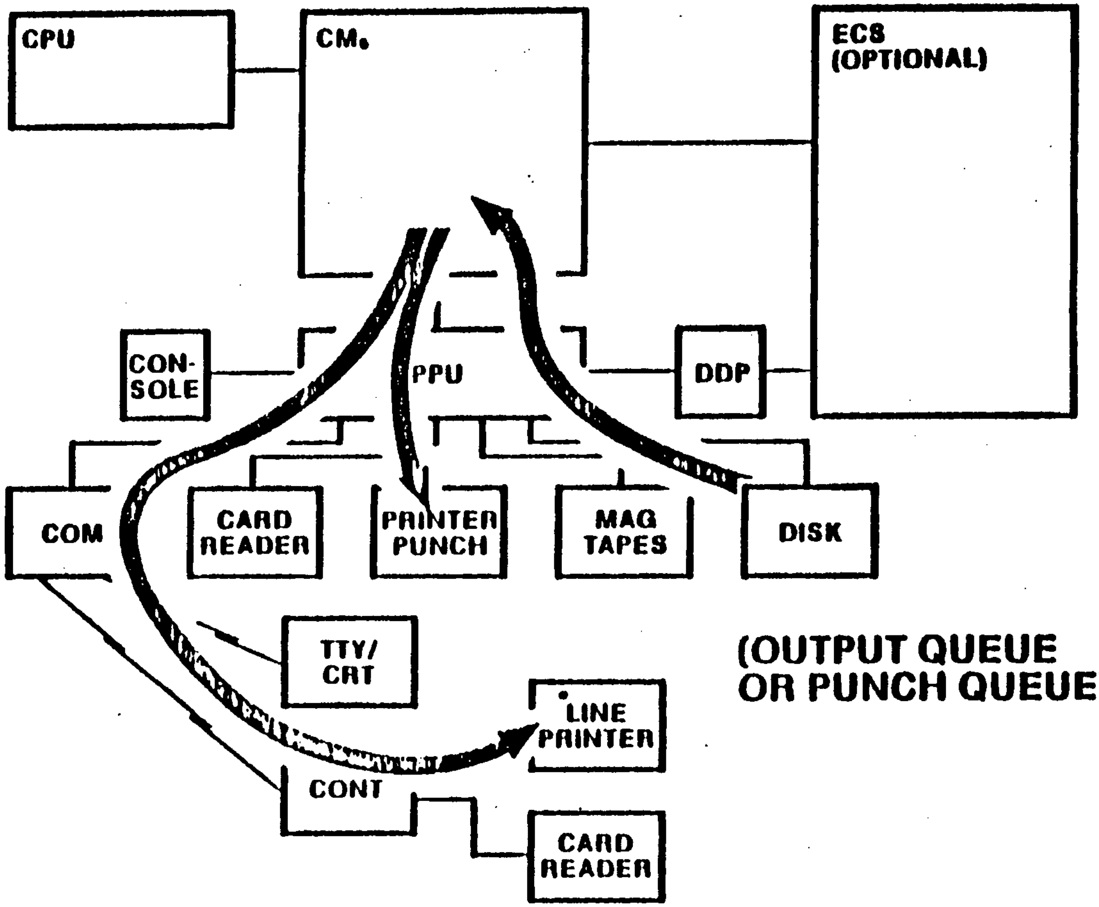


USER JOB EXECUTION



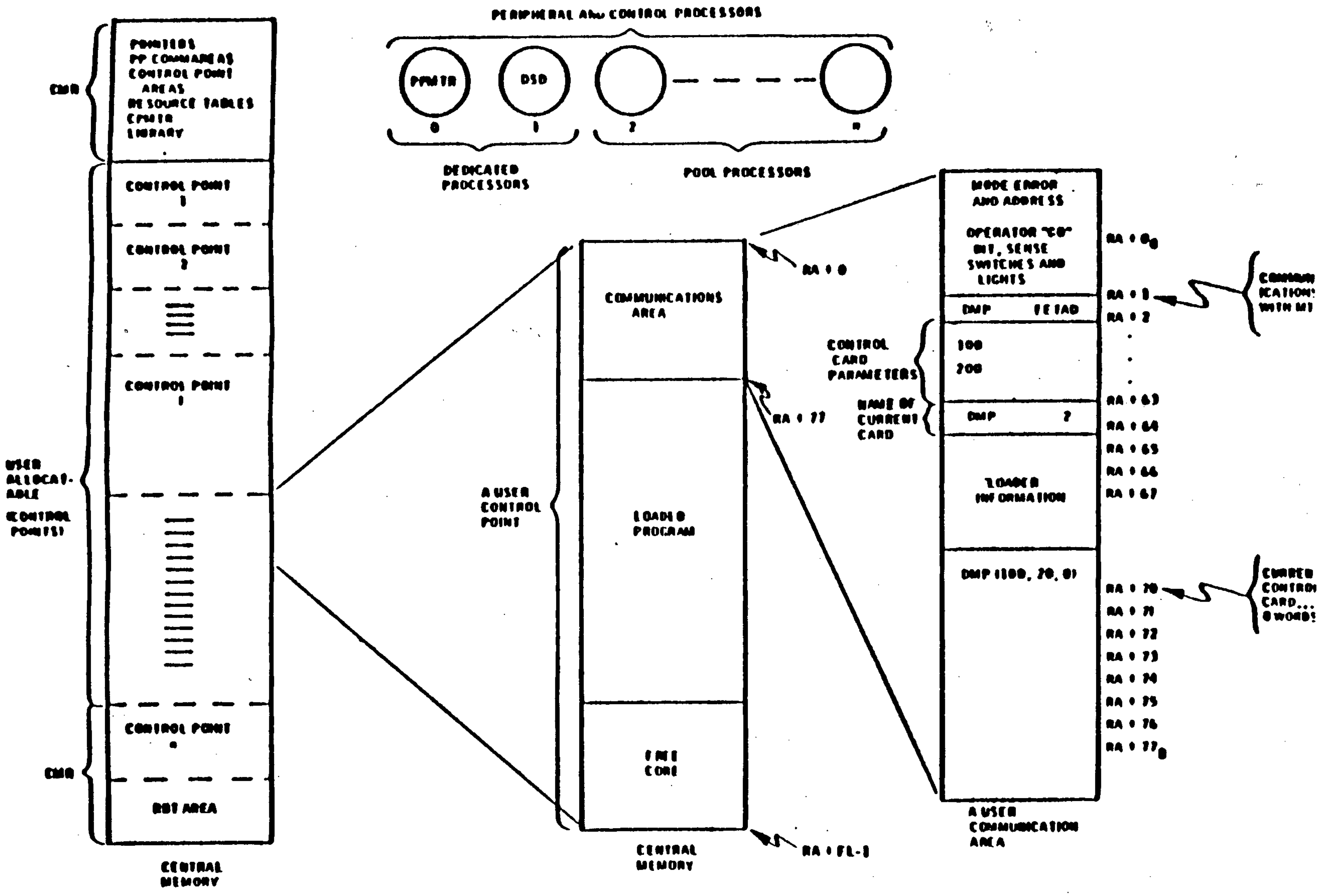
OPERATOR AND DAYFILE MESSAGES

CDC CYBER 170 SERIES



(OUTPUT QUEUE OR PUNCH QUEUE)

BATCH OUTPUT

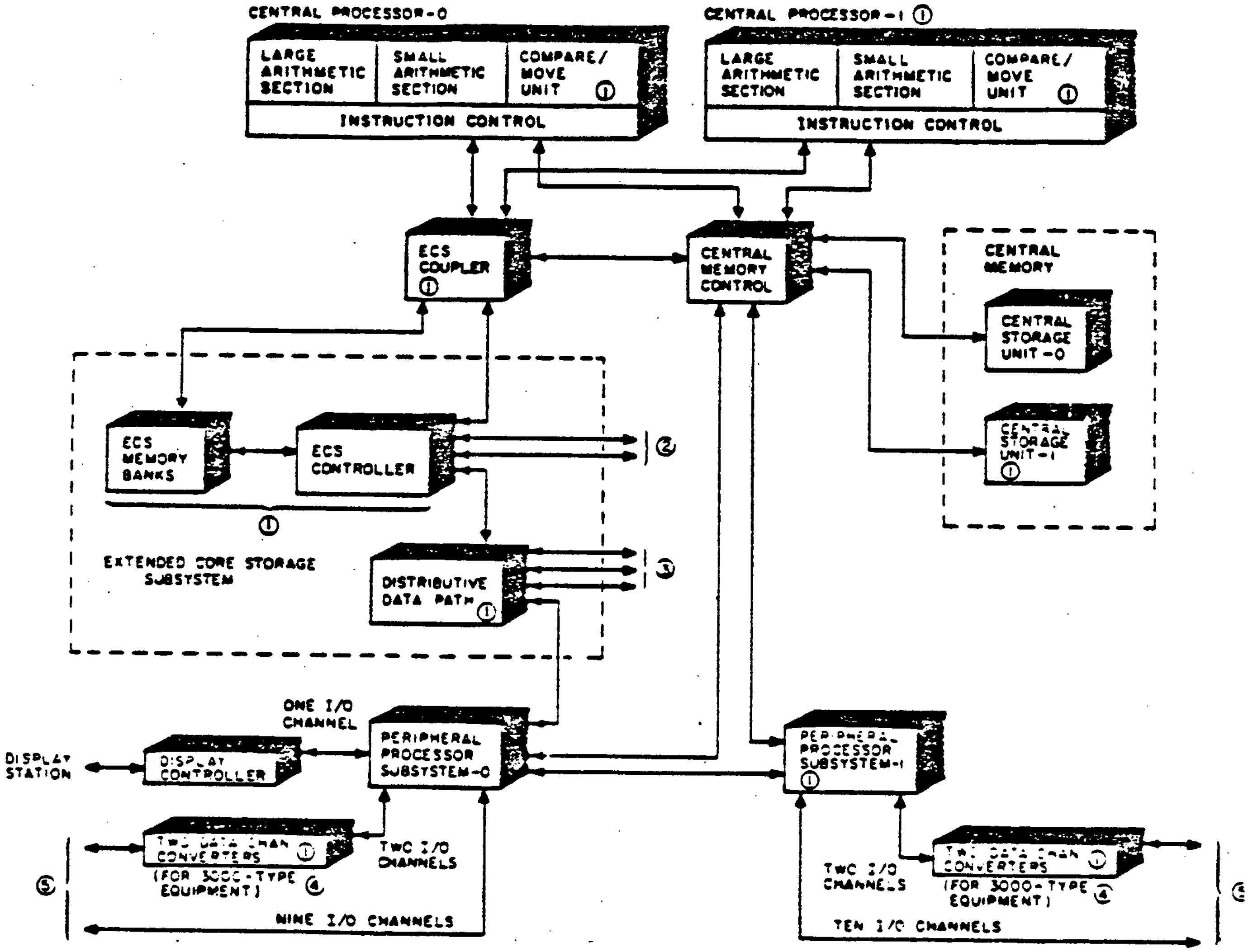

PERIPHERAL AND CONTROL PROCESSORS
PPMTR
DSD
0
1
2
n
DEDICATED PROCESSORS
POOL PROCESSORS
CMR
POINTERS
PP COMM AREAS
CONTROL POINT AREAS
RESOURCE TABLES
CPMTR
LIBRARY
CONTROL POINT 1
CONTROL POINT 2
CONTROL POINT i
CONTROL POINT n
USER ALLOCATABLE
(CONTROL POINTS)
CMR
RBT AREA
CENTRAL MEMORY
A USER CONTROL POINT
COMMUNICATIONS AREA
LOADED PROGRAM
FREE CORE
RA + 0
RA + 77
RA + FL-1
CENTRAL MEMORY
MODE ERROR AND ADDRESS
OPERATOR "GO" BIT, SENSE SWITCHES AND LIGHTS
RA + 0₈
RA + 1
RA + 2
DMP FETAD
CONTROL CARD PARAMETERS
100
200
RA + 63
NAME OF CURRENT CARD
DMP ?
RA + 64
RA + 65
RA + 66
RA + 67
LOADER INFORMATION
DMP (100, 20, 0)
RA + 70
RA + 71
RA + 72
RA + 73
RA + 74
RA + 75
RA + 76
RA + 77₈
COMMUN ICATIONS WITH MT
CURREN CONTROL CARD... 8 WORDS
A USER COMMUNICATION AREA

MODEL 171 SYSTEM

The model 171 basic computer system (figure 1-9) has a serial CP-0 with a serial CP-1 option. Each CP contains large and small arithmetic sections, instruction control, and a compare/move unit. The CPs communicate with each PPS and ECS, if installed, through CM. CM is under control of the CMC.

If the optional ECS is installed, it provides additional memory capabilities, short access times, and fast transfer rates to and from CM.

The PPS-0 performs all I/O operations and uses an instruction set separate from that of the CP to execute independent programs in each of 10 PPs. The PPs have individual memories and communicate with each other and any of 12 I/O channels. The PPs may be expanded from 10 to 14, 17, or 20 by adding PPS-1. This option expands the number of I/O channels from 12 to 24.



NOTES

① OPTIONAL EQUIPMENT.

② TWO PORTS AVAILABLE FOR USE BY OTHER SYSTEMS OR DISTRIBUTIVE DATA PATHS.

③ THREE PORTS AVAILABLE AS OPTIONS FOR USE BY OTHER SYSTEMS.

④ EXTERNAL DATA CHANNEL CONVERTERS MAY BE ADDED IN ADDITION TO THOSE IN PPS-0 AND PPS-1.

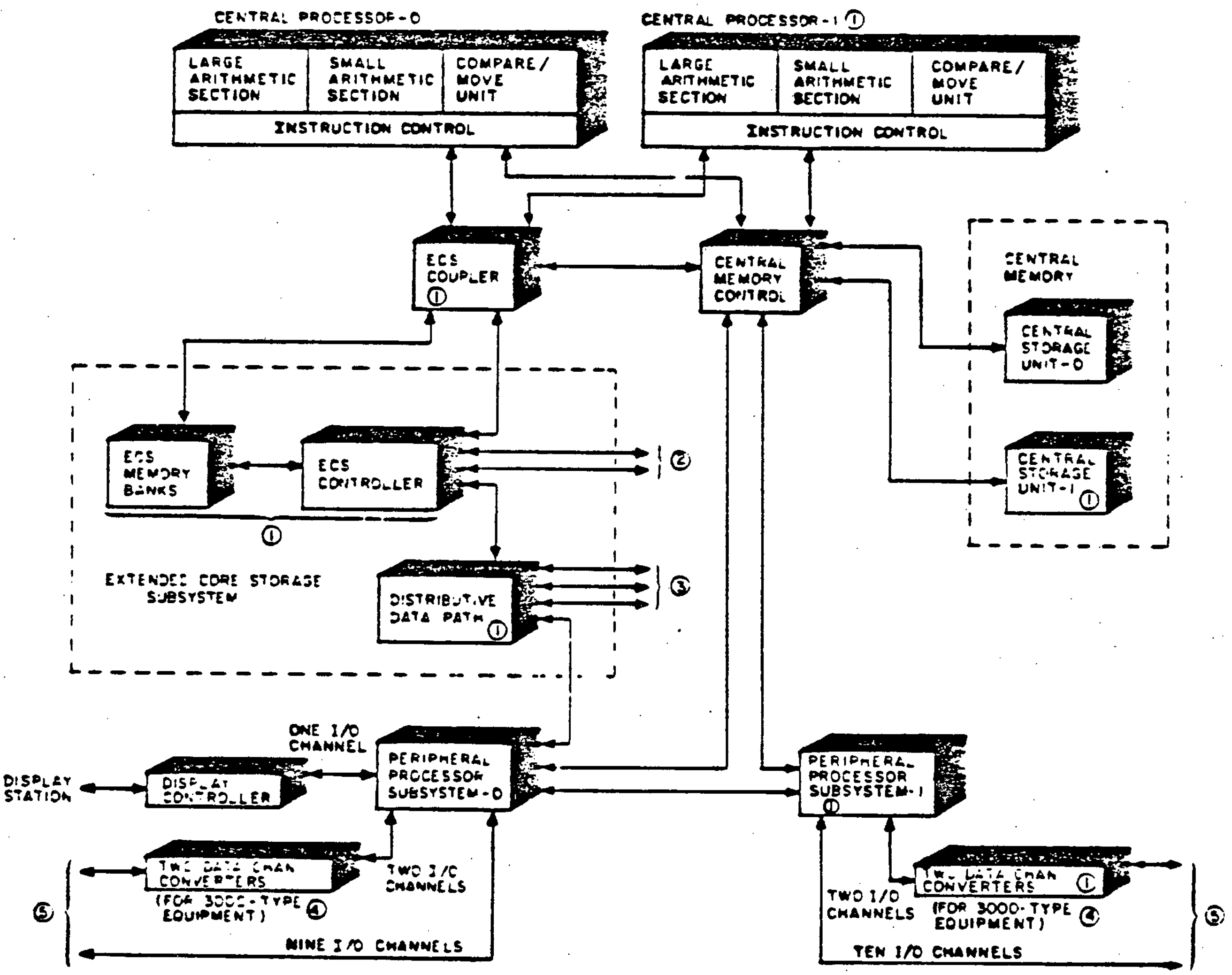
⑤ PERIPHERAL EQUIPMENT.

Figure 1-9. Model 171 Computer System

MODEL 172 SYSTEM

The model 172 basic computer system (figure 1-10) is functionally similar to model 171, except that the CP provides faster operation. The model 172 also has a second CP option. Basic equipment in model 172 includes a compare/move unit in the CP and two DCCs in PPS-0. This equipment is available only as options in model 171.

Original and later model 172 systems have some differences that result from development of additional equipment. Figure 1-10 represents the later model 172 system. The original model 172 basic memory has 32,768 words and requires two increments to reach the 65,536 words of the basic memory of later model 172 systems.



NOTES

① OPTIONAL EQUIPMENT.

② TWO PORTS AVAILABLE FOR USE BY OTHER SYSTEMS OR DISTRIBUTIVE DATA PATHS

③ THREE PORTS AVAILABLE AS OPTIONS FOR USE BY OTHER SYSTEMS

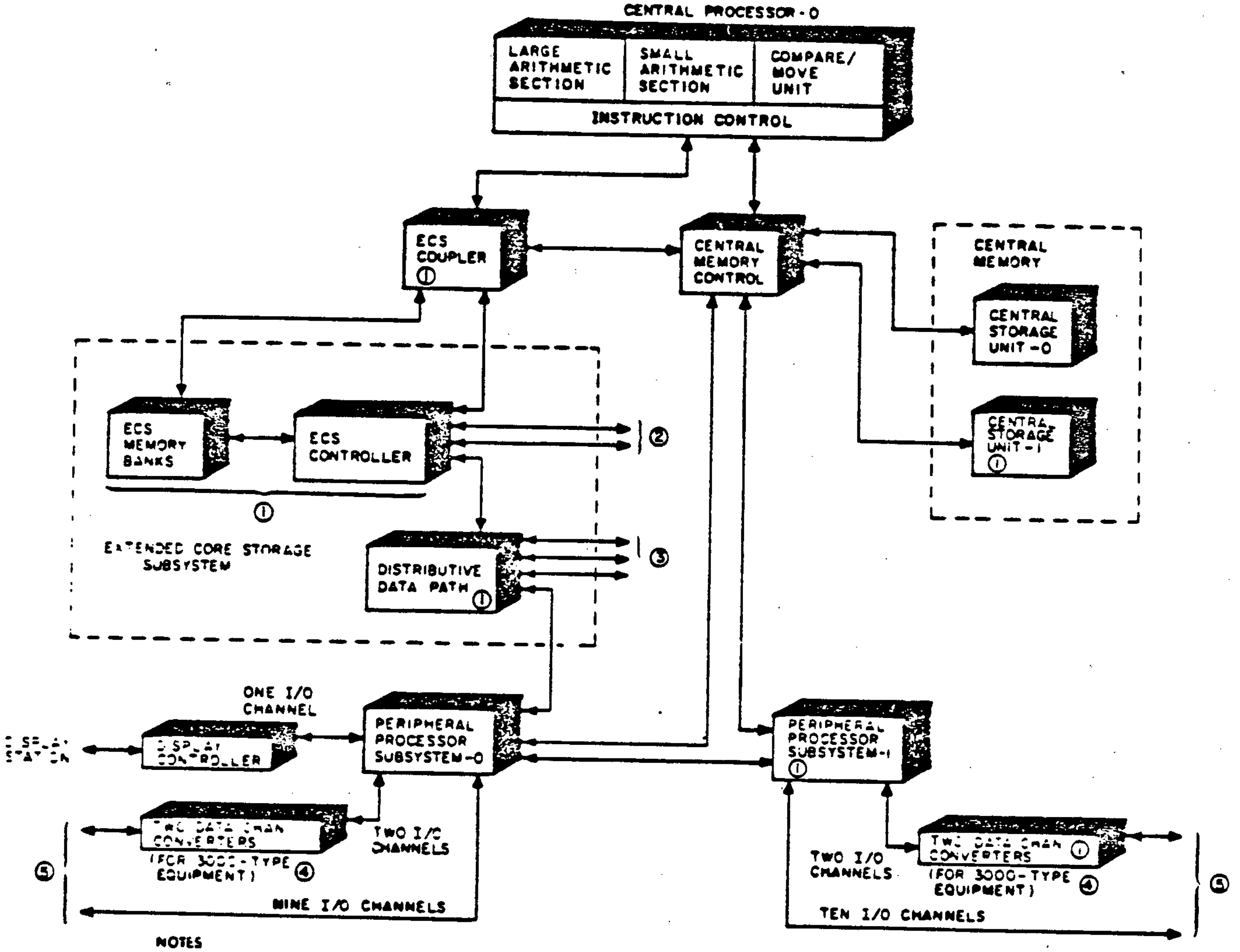
④ EXTERNAL DATA CHANNEL CONVERTERS MAY BE ADDED IN ADDITION TO THOSE IN PPS-0 AND PPS-1

⑤ PERIPHERAL EQUIPMENT

Figure 1-10. Model 172 Computer System

MODEL 173 SYSTEM

The model 173 basic computer system (figure 1-11) is functionally similar to model 172, except that the CP provides faster operation.



NOTES

① OPTIONAL EQUIPMENT.

② TWO PORTS AVAILABLE FOR USE BY OTHER SYSTEMS OR DISTRIBUTIVE DATA PATHS.

③ THREE PORTS AVAILABLE AS OPTIONS FOR USE BY OTHER SYSTEMS.

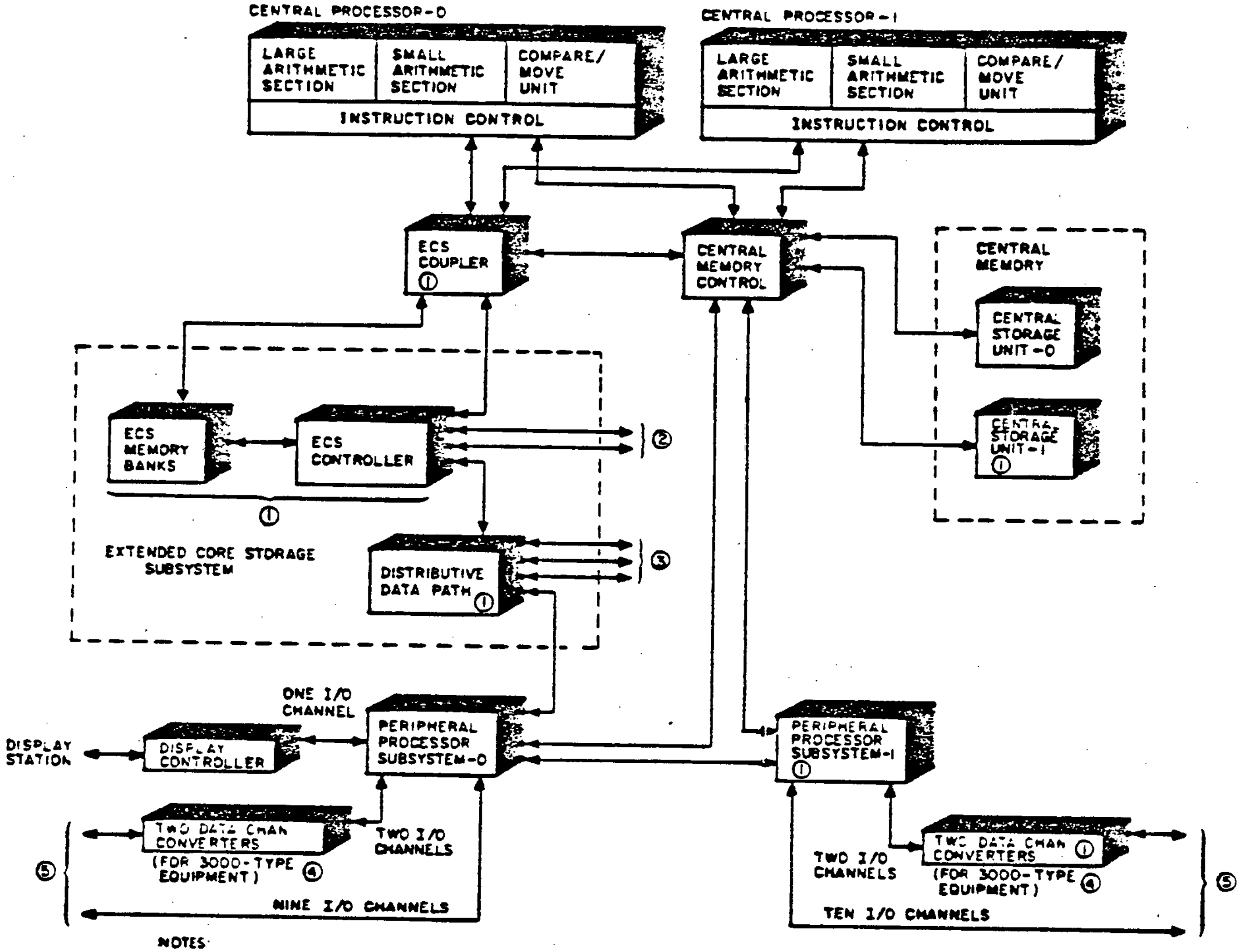
④ EXTERNAL DATA CHANNEL CONVERTERS MAY BE ADDED IN ADDITION TO THOSE IN PPS-0 AND PPS-1

⑤ PERIPHERAL EQUIPMENT

Figure 1-11. Model 173 Computer System

MODEL 174 SYSTEM

The model 174 basic computer system (figure 1-12) is functionally similar to model 173, except that the system provides faster operation. Model 174 differs basically from model 173 by having a second CP. The ECS, CM, PPS, and I/O options are the same as for model 173.



NOTES:

① OPTIONAL EQUIPMENT.

② TWO PORTS AVAILABLE FOR USE BY OTHER SYSTEMS OR DISTRIBUTIVE DATA PATHS.

③ THREE PORTS AVAILABLE AS OPTIONS FOR USE BY OTHER SYSTEMS.

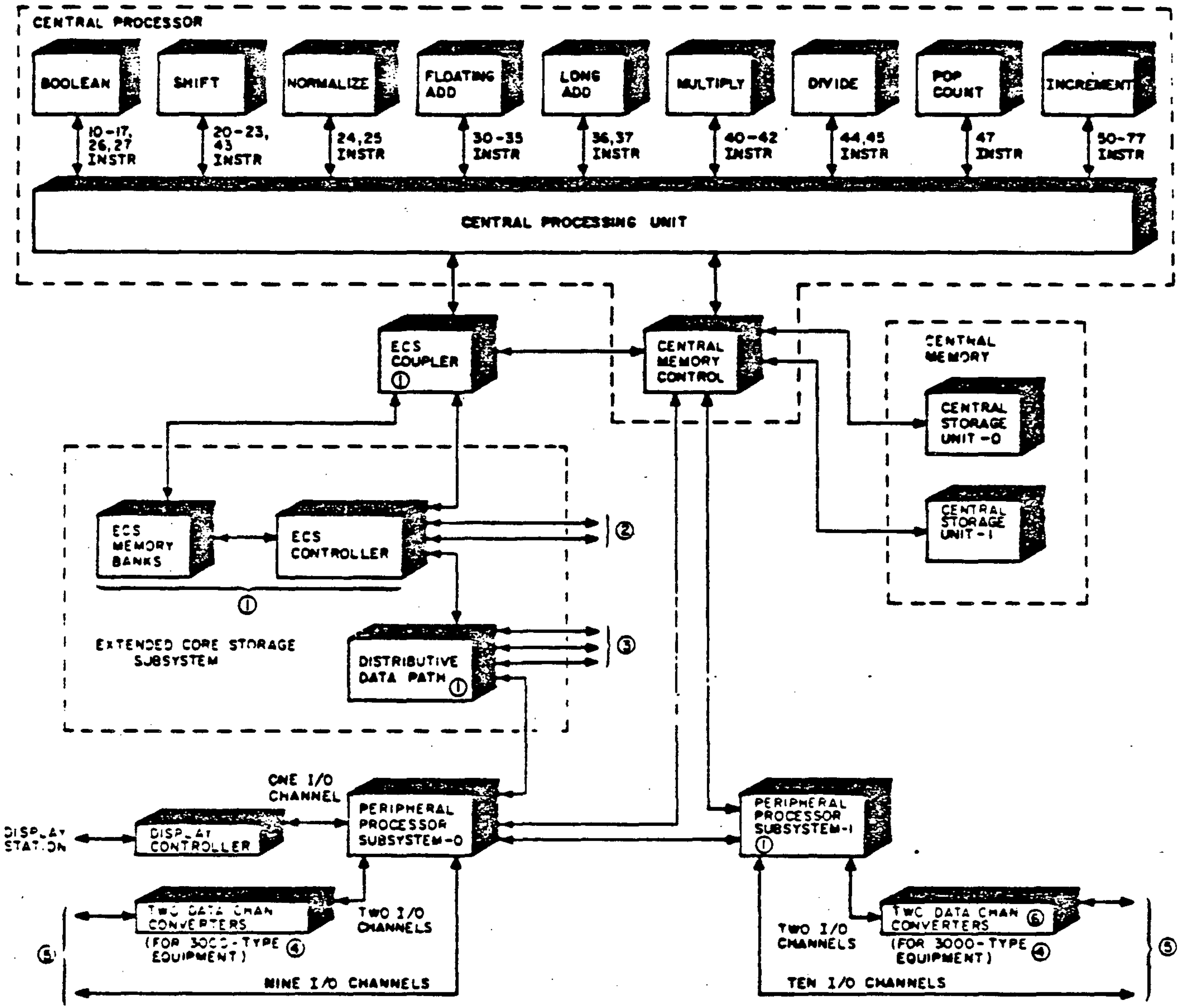
④ EXTERNAL DATA CHANNEL CONVERTERS MAY BE ADDED IN ADDITION TO THOSE IN PPS-0 AND PPS-1.

⑤ PERIPHERAL EQUIPMENT.

Figure 1-12. Model 174 Computer System

MODEL 175 SYSTEM

The model 175 basic computer system (figure 1-13) is functionally similar to model 173 and its options except in the CP. In place of the serial CP, the model 175 CP contains nine functional units, a central processing unit (CPU), and the CMC. The nine functional units operate in parallel as independent specialized arithmetic units, providing maximum overlap of instruction retrieval and execution. The basic model 175 has two CSUs that provide 16 independent banks of memory.



NOTES

① OPTIONAL EQUIPMENT

② TWO PORTS AVAILABLE FOR USE BY OTHER SYSTEMS OR DISTRIBUTIVE DATA PATHS

③ THREE PORTS AVAILABLE AS OPTIONS FOR USE BY OTHER SYSTEMS

④ EXTERNAL DATA CHANNEL CONVERTERS MAY BE ADDED IN ADDITION TO THOSE IN THE PPS.

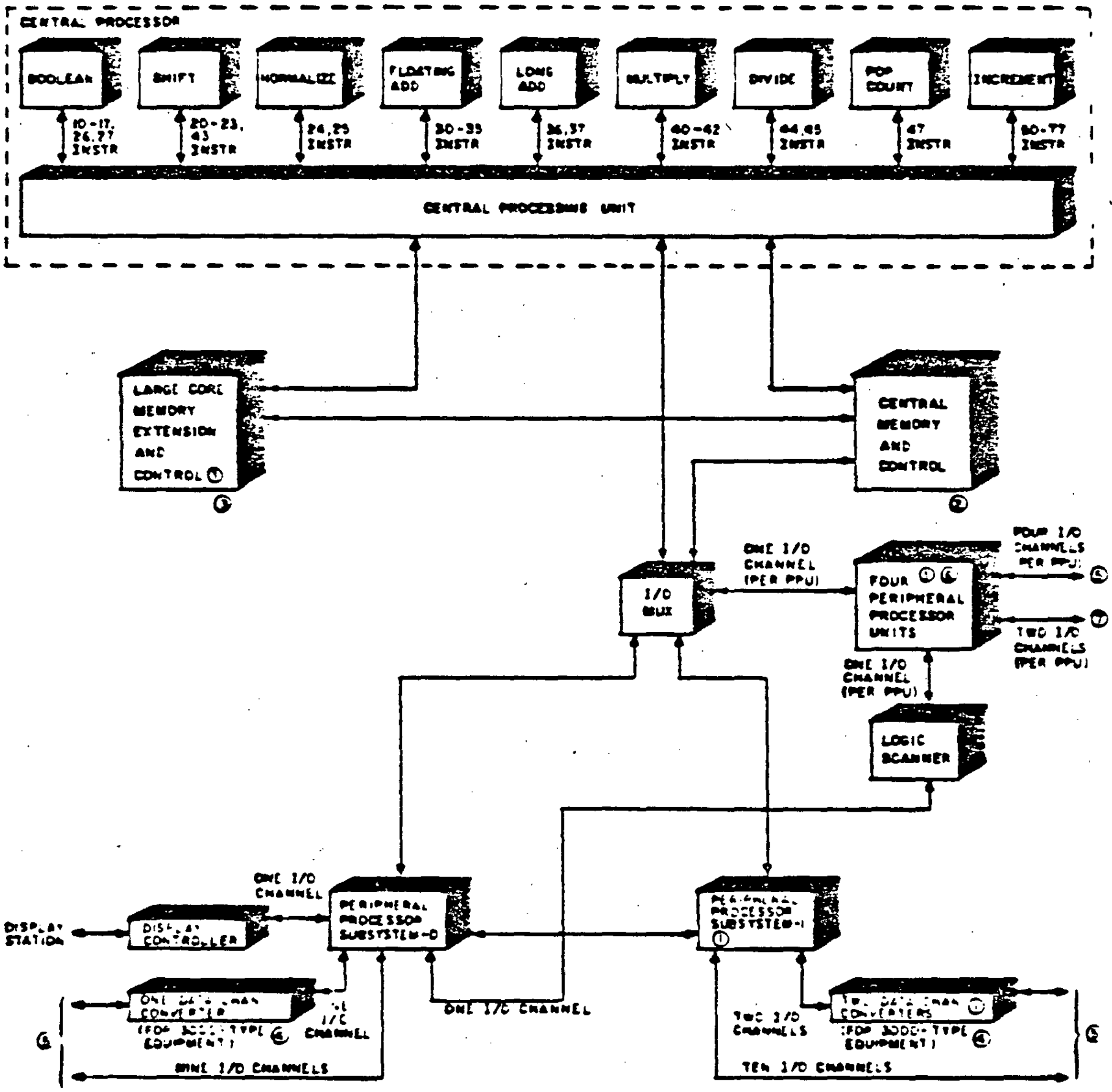
⑤ PERIPHERAL EQUIPMENT

⑥ OPTIONAL EQUIPMENT FOR MODELS 175A AND 175B NOT AVAILABLE FOR MODEL 175C

Figure 1-13. Model 175 Computer System

MODEL 176 SYSTEM

The model 176 basic computer system (figure 1-14) is functionally similar to model 175 in the areas of the CP and PPS. Model 176 differs basically from model 175 in the use of an LCME option in the basic system instead of having an ECS option. The CM is still optionally expandable but does not have separate CSUs as in other models. The CM and LCME each contain their own control functions. Other major differences include the option of adding from 4 to a total of 13 PPUs, an I/O multiplexer, and a logic scanner to permit PPS communication with the PPUs.


CENTRAL PROCESSOR
BOOLEAN
SHIFT
NORMALIZE
FLOATING ADD
LONG ADD
MULTIPLY
DIVIDE
POP COUNT
INCREMENT
10-17, 26,27 INSTR
20-23, 43 INSTR
24,25 INSTR
30-35 INSTR
36,37 INSTR
40-42 INSTR
44,45 INSTR
47 INSTR
50-77 INSTR
CENTRAL PROCESSING UNIT
LARGE CORE MEMORY EXTENSION AND CONTROL ①
③
CENTRAL MEMORY AND CONTROL
②
ONE I/O CHANNEL (PER PPU)
I/O MUX
FOUR PERIPHERAL PROCESSOR UNITS ① ⑥
FOUR I/O CHANNELS PER PPU
⑤
⑦
TWO I/O CHANNELS (PER PPU)
ONE I/O CHANNEL (PER PPU)
LOGIC SCANNER
ONE I/O CHANNEL
DISPLAY STATION
DISPLAY CONTROLLER
PERIPHERAL PROCESSOR SUBSYSTEM-0
PERIPHERAL PROCESSOR SUBSYSTEM-1 ①
ONE DATA CHAN CONVERTER (FOR 3000-TYPE EQUIPMENT) ④
ONE I/O CHANNEL
⑤
NINE I/O CHANNELS
ONE I/O CHANNEL
TWO I/O CHANNELS
TWO DATA CHAN CONVERTERS ① (FOR 3000-TYPE EQUIPMENT) ④
⑤
TEN I/O CHANNELS

NOTES

① OPTIONAL EQUIPMENT

② BASIC CM CONTAINS 16 INDEPENDENT BANKS OF 8,192 60-BIT WORDS EACH AND IS EXPANDABLE TO 12,288 TO 16,384 WORDS

③ BASIC LCME CONTAINS 2 INDEPENDENT BANKS OF 262,144 60-BIT WORDS AND IS EXPANDABLE TO 4 OR 8 INDEPENDENT BANKS OF 262,144 WORDS PER BANK

④ ADDITIONAL DATA CHANNEL CONVERTERS MAY BE ADDED EXTERNALLY TO PPS-0 AND PPS-1

⑤ PERIPHERAL EQUIPMENT

⑥ ADDITIONAL PERIPHERAL PROCESSOR UNITS ARE OPTIONAL

⑦ ONE CHANNEL FOR INTER-PPU COMMUNICATIONS AND ONE CHANNEL NOT USED

Figure 1-14. Model 176 Computer System

Functional Unit in Central Processor

Unit	Function
Boolean	Basic logical operations of transfer, logical product and sum, and logical difference.
Shift	Basic shifting operations. Left circular shifts, right end-off sign extension, plus the floating point work of normalization, pack and unpack. Also provides the mask generator.
F.P. add	Floating point addition and subtraction
Fix add	One's complement addition and subtraction on 60-bit fixed point numbers.
Multiply	Floating point multiplication
Divide	Floating point division
Increment	One's complement addition and subtraction of 18-bit numbers.
Normalize	Places a word in normalized format.
Pop Count	Count the number of one bits in a word.

CENTRAL PROCESSOR REGISTERS

A EIGHT 18-BIT ADDRESS REGISTERS

PREVIOUS COMPUTERS USED SINGLE-ACCUMULATOR DESIGN. IN THE 6000 DESIGN THIS WAS INEFFICIENT FOR LENGTH OF INSTRUCTION (15 OR 30 BIT) AND FETCH TIME...

SECOND PROBLEM WAS SINGLE-ADDRESS PROCEDURE THAT IMPLIES SINGLE ACCUMULATOR.

ADDITIONAL CP REGISTERS SOLVED PROBLEM

A0 - USED FOR COMMUNICATION WITH ECS

A1 - A5 - USED TO 'LDA' FUNCTION, LOAD ACCUMULATOR WITH INFORMATION FROM MEMORY.

A6, A7 - USED TO 'STA', PUT INFORMATION BACK INTO MEMORY

B EIGHT 18-BIT INCREMENT REGISTERS

ALSO SAME LENGTH AS ADDRESS REGISTERS, CAN BE USED FOR ADDRESS, INCREMENTS, COUTINERS, INDEXING, ETC.

X EIGHT 60-BIT OPERAND REGISTERS

ALL ARITHMETIC OPERATIONS PERFORMED ON OPERANDS IN X-REG.

CHOICE OF 60-BITS FOR INSTRUCTION PACKING AND FLOATING POINT PRECISION.

X0 - COMMUNICATION WITH ECS

X1 - X5 - VALUE OF WORD AT ADDRESS A1 THRU A-5 .... CONTENTS

X6, X7 - STORE CONTENTS OF X REG, IN ADDRESS A6 OR A7

ALL 24 REGISTERS ARE PROGRAMMABLE.

1. INSTRUCTIONS NEVER ACCESS MEMORY DIRECTLY. INSTEAD, CHANGING THE CONTENTS OF AN A-REGISTER CAUSES A MEMORY REFERENCE FOR THE CORRESPONDING X-REGISTER.

2. X AND A REGISTERS 1-5 ARE USED TO ACCESS MEMORY... PUTTING A 5 IN A2 CAUSES CONTENTS OF MEMORY LOCATION 5 TO BE PUT INTO X2.

3. A AND X REGISTERS 6,7 ARE USED TO STORE INTO MEMORY... PUT A 3 IN A6 CAUSES THE CONTENTS OF X6 TO BE PUT INTO LOCATION 3.

4. B0 IS ALWAYS ZERO.

LESSON 3

JOB OVERVIEW

LESSON PREVIEW:

THIS LESSON INTRODUCES MACHINE LANGUAGE PROGRAMMING, ASSEMBLY LANGUAGE PROGRAMMING, CODING CONVENTIONS, INSTRUCTION FORMATS, JOB STRUCTURE AND A FEW BASIC MNEMONICS.

REFERENCES:

CHAPTERS 1-3, 10, 11   COMPASS REF. MAN. #60492600

TRAINING AIDS:

VISUAL SET V3
PROGRAM COMPILATION OF DECK 3A

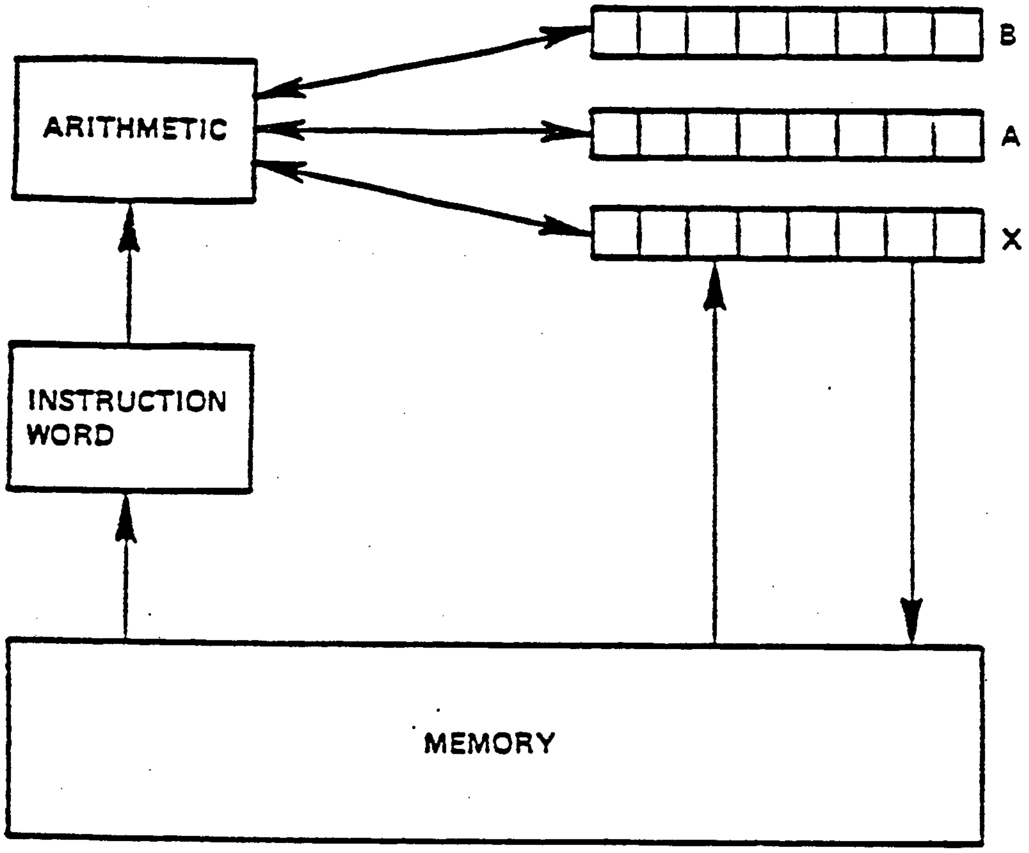
PROJECTS:

PROGRAMMING PROJECT #1
HOMEWORK

OBJECTIVES:

AT THE COMPLETION OF THIS LESSON THE STUDENT WILL BE ABLE TO:

1. WRITE A SIMPLE PROGRAM IN MACHINE LANGUAGE.
2. WRITE A SIMPLE PROGRAM IN ASSEMBLY LANGUAGE, I.E., COMPASS.
3. EXPLAIN THE INTERFACE BETWEEN THE ASSEMBLER AND THE LOADER.
4. FIND RESULTS IN A DUMP.
5. CONSTRUCT A JOB DECK TO RUN AND EXECUTE A COMPASS PROGRAM.
6. USE ARITHMETIC AND INCREMENT UNIT INSTRUCTIONS CORRECTLY.

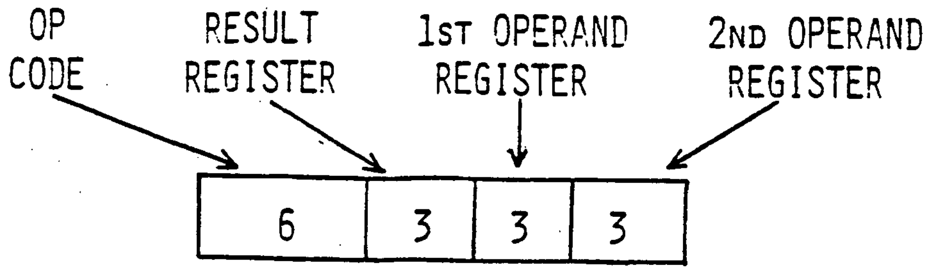


Instruction Flow

An instruction word is fetched from memory. The arithmetic unit will interpret the instruction word and perform required operations. Every instruction is an interregister instruction, i.e., the operands will be in registers and the results will be written to registers. When a memory read operation (an address to be put in A1 to A5) or a memory write operation (an address to be put in A6 or A7) is required, a transfer between memory and the appropriate X register will take place. Another instruction word is fetched.

INSTRUCTION FORMATS

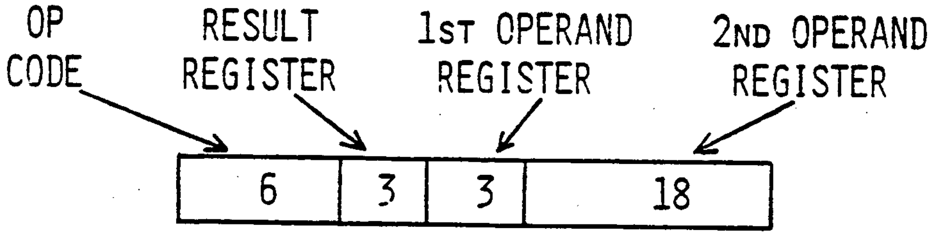
15 BIT INSTRUCTIONS



ALL ARITHMETIC INSTRUCTIONS AND SOME INCREMENT UNIT INSTRUCTIONS ARE 15 BITS. FOR EXAMPLE:

MACHINE CODE	COMPASS	
36123	IX1	X2+X3
40321	FX3	X2*X1
66200	SB2	B0+B0
56120	SA1	B2+B0

30 BIT INSTRUCTIONS

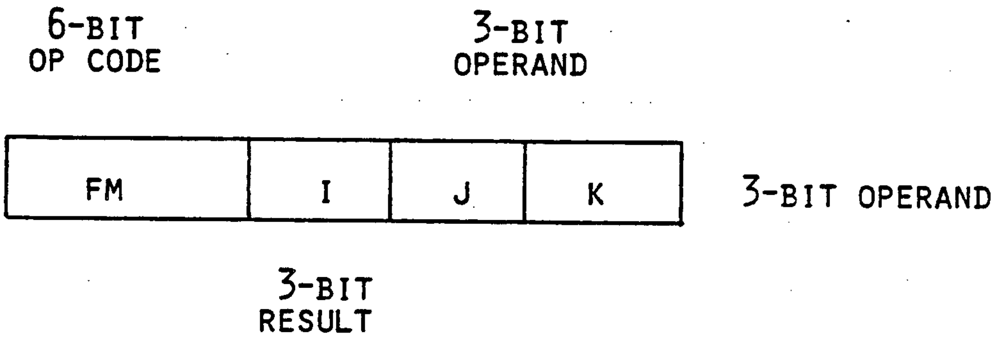


SOME INCREMENT UNIT INSTRUCTIONS AND ALL BRANCH INSTRUCTIONS ARE 30 BITS. FOR EXAMPLE:

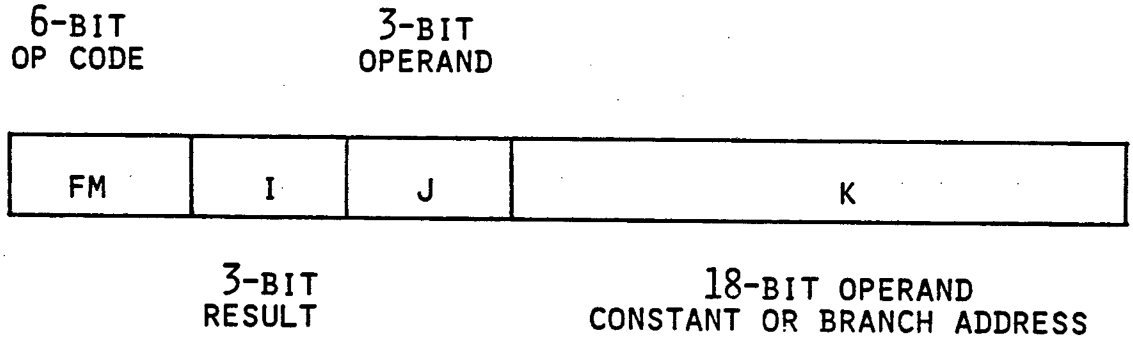
6111000001	SB1	B1+1
5160001000	SA6	B0+1000_8
0412000023	EQ	B1,B2,23_8

INSTRUCTION FORMATS

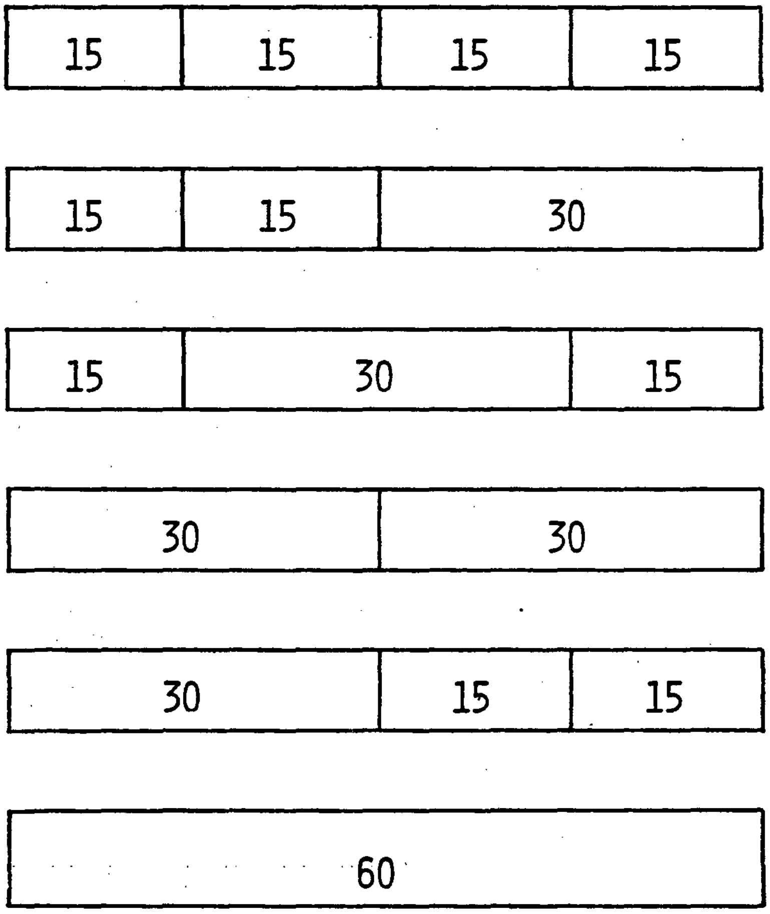
15-BIT INSTRUCTIONS



30-BIT INSTRUCTIONS



INSTRUCTION WORD FORMATS



MACHINE LANGUAGE PROGRAMMING

1. READ FOUR NUMBERS FROM MEMORY.

2. ADD THE NUMBERS.

3. STORE THE RESULT IN MEMORY.

```
100 | 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 5 0 |
    | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 |
    | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 5 |
    | 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 6 |
105 | 5 1 1 0 0 0 0 1 0 0|5 1 2 0 0 0 0 1 0 1 |
    | 5 1 3 0 0 0 0 1 0 2|5 1 4 0 0 0 0 1 0 3 |
    | 3 6 6 1 2|3 6 7 3 4|3 6 6 6 7|4 6 0 0 0 |
    | 5 1 6 0 0 0 0 1 0 4|                    |
```

EXERICSE 1

DIRECTIONS: WRITE A MACHINE LANGUAGE PROGRAM WHICH WILL EVALUATE (A&B)/C WHERE A=2.2, B=3.8, C=2.0. ASSUME THAT 2.2, 3.8 AND 2.0 ARE IN CORE AT LOCATIONS 100, 101, AND 102. STORE THE RESULT AT 103. THE PROGRAM STARTS AT 104.

```
100 |                          2.2 |
    |                          3.8 |
    |                          2.0 |
    |                          ___ |
    |------------------------------|
    |                              |
    |------------------------------|
    |                              |
    |------------------------------|
    |                              |
    |------------------------------|
    |                              |
    |------------------------------|
```

ASSEMBLY LANGUAGE

COMPASS CODING FORM

IN COMPASS THE PROGRAM WILL LOOK LIKE THIS:

```
                SA1           100
                SA2           101
                SA3           102
                SA4           103
                IX6           X1+X2
                IX7           X3+X4
                IX6           X6+X7
                SA6           104
```

WE STILL HAVE NO WAY TO GET THE DATA INTO MEMORY. THE DATA SPEUDO OP WILL DO THAT.

```
      LOCAD       DATA        -23,4,21,-1
```

THE ASSEMBLER WILL ASSIGN AN ADDRESS FOR LOCAD RELATIVE TO THE START OF THE PROGRAM. WE CAN USE THAT INSTEAD OF 100. NOW WE HAVE:

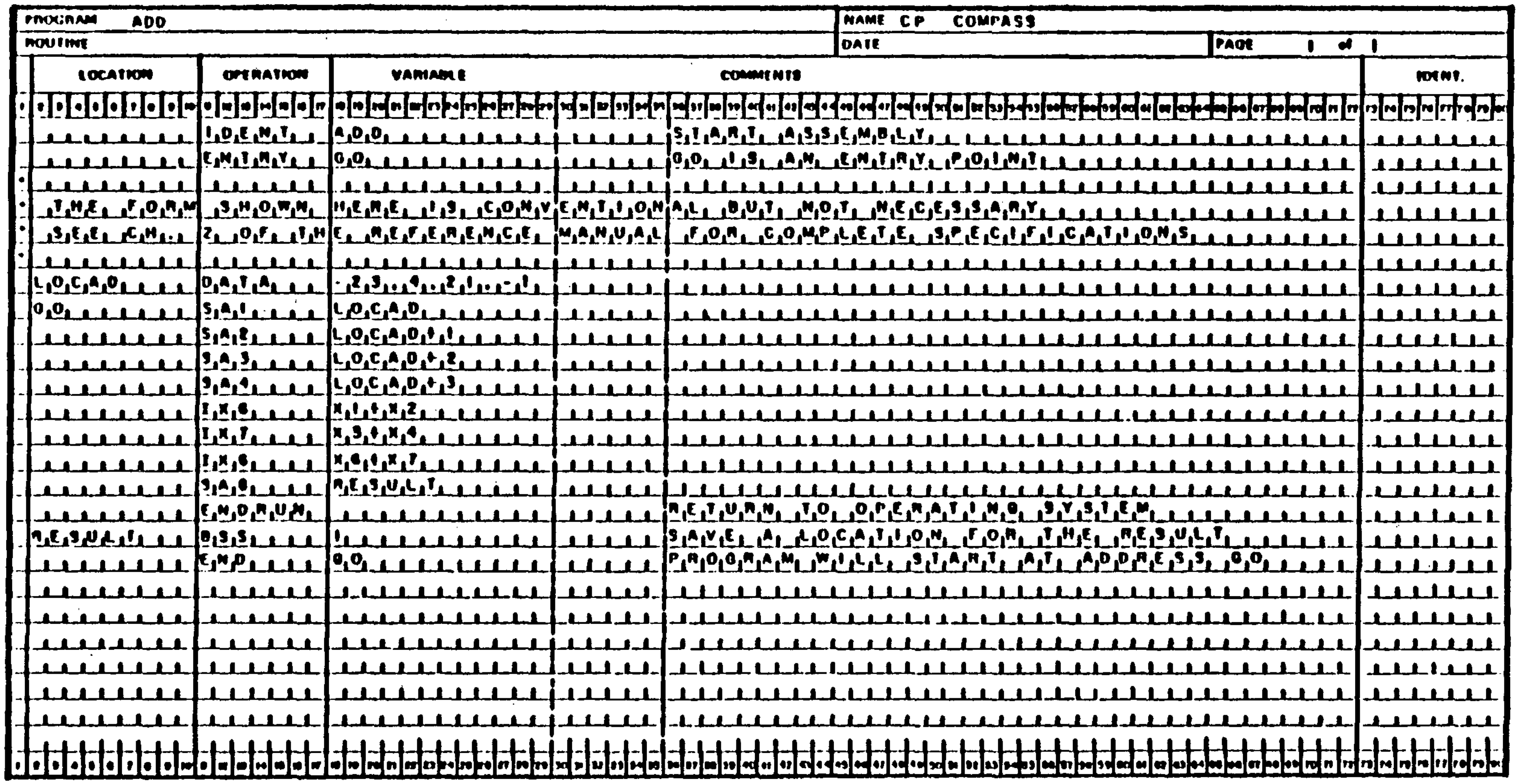
```
      LOCAD       DATA        -23,4,21,-1
                  SA1         LOCAD
                  SA2         LOCAD+1
                  SA3         LOCAD+2
                  SA4         LOCAD+3
                  IX6         X1+X2
                  IX7         X3+X4
                  IX6         X6+X7
                  SA6         RESULT
      RESULT      BSS         1
```

THE BSS SAVED ONE LOCATION FOR THE RESULT.

IF WE START THE ASSEMBLY WITH IDENT, END IT WITH END, AND ESTABLISH STARTING AND ENDING EXECUTION POINTS, WE'LL BE DONE. THE COMPLETE PROGRAM IS SHOWN ON THE NEXT PAGE.

CONTROL DATA CORPORATION

CYBER 70/6000/7000 COMPASS CODING FORM



PROGRAM ADD

NAME CP COMPASS

ROUTINE

DATE

PAGE 1 of 1

```
LOCATION  OPERATION VARIABLE                  COMMENTS                                      IDENT.
          IDENT     ADD                       START ASSEMBLY
          ENTRY     GO                        GO IS AN ENTRY POINT
*
* THE FORM SHOWN HERE IS CONVENTIONAL BUT NOT NECESSARY
* SEE CH. 2 OF THE REFERENCE MANUAL FOR COMPLETE SPECIFICATIONS
*
LOCAD     DATA      -23,4,21,-1
GO        SA1       LOCAD
          SA2       LOCAD+1
          SA3       LOCAD+2
          SA4       LOCAD+3
          IX6       X1+X2
          IX7       X3+X4
          IX6       X6+X7
          SA6       RESULT
          ENDRUN                              RETURN TO OPERATING SYSTEM
RESULT    BSS       1                         SAVE A LOCATION FOR THE RESULT
          END       GO                        PROGRAM WILL START AT ADDRESS GO
```

LOCATION FIELD

BEGIN: Column 1 or 2   END: First blank or Column 3   NORMAL: Columns 2-9

CONTENTS:

1. SYMBOL
   (1-8 characters, except LINKAGE SYMBOL, 7 characters, or 3 for PP)

2. A NAME
   (a SYMBOL for a BLOCK, MACRO, MICRO, or Instruction Bracket)

3. +
   (Forces upper)

4. -
   (Cancels some force-uppers)

OPERATION FIELD

BEGIN: COLUMN 3-29 END: FIRST BLANK, OR COLUMN 29 IF FOLLOWED BY A VARIABLE FIELD ENTRY.

NORMAL: COLUMNS 11-16

CONTENTS:

1. CP OPERATION CODE
2. PP OPERATION CODE
3. PSEUDO-INSTRUCTION
4. MACRO NAME

VARIABLE FIELD

(ADDRESS FIELD)

BEGIN: BEFORE COLUMN 29   END: FIRST BLANK, OR COLUMN 72

NORMAL: COLUMN 18-29

CONTENTS: (DICTATED BY THE OPERATION CODE)

1. REGISTERS, SEPARATED BY OPERATOR + - * /

```
IX6 X1+X2
```

2. SUBFIELDS, SEPARATED BY COMMAS.

```
LX1 B2,X6
EQ B3,B6,ADDRESS
```

COMMENTS FIELD

BEGIN: After Variable Field
Column 30 if VAR Field is empty

END: Column 72

NORMAL: Columns 30-72

CONTENTS: Any combination of characters

LOCATION	OPERATION	VARIABLE
IGNORED	IDENT	1, 2, OR 3 SUBFIELDS

- MUST BE 1ST OPERATION OF A SUBPROGRAM
- CAN OCCUR ONLY ONCE PER SUBPROGRAM
- IF THE ASSEMBLER IS CALLED BY FORTRAN RATHER THAN BY A COMPASS CONTROL CARD, IDENT MUST APPEAR IN COL. 11-15.
- 1ST VARIABLE SUBFIELD --- MUST CONTAIN THE LINKAGE SYMBOL WHICH BECOMES THE SUBPROGRAM NAME.
- FOR RELOCATABLE ASSEMBLIES --- IGNORE VARIABLE SUBFIELDS 2 AND 3.
- FOR ABSOLUTE ASSEMBLIES --- 2ND SUBFIELD DEFINES THE FWA OF THE ABSOLUTE BINARY PROGRAM IMAGE. THE 3RD SUBFIELD CONTAINS THE ENTRY ADDRESS FOR ABSOLUTE CP ASSEMBLIES.

IGNORED	ENTRY	LINKAGE SYMBOLS SEPARATED BY COMMAS

- LINKAGE SYMBOLS ARE DECLARED TO THE LOADER AS ENTRY POINTS.
- EACH LINKAGE SYMBOL MUST BE DEFINED IN THE ASEEMBLY AS A NON-EXTERNAL SYMBOL.

SYMBOL OR BLANK	END	BLANK OR LINKAGE SYMBOL (TRANSFER ADDRESS)

- TERMINATES A SUBPROGRAM DECK
- CAUSES THE ASSEMBLER TO TERMINATE ANY COUNTER, CONDITIONAL ASSEMBLY, MACRO GENERATION, OR CODE DUPLICATION IN PROGRESS.
- ALL LOCAL BLOCKS ARE ASSIGNED IN ORIGIN RELATIVE TO THE PROGRAM ORIGIN IN THE ORDER IN WHICH THEY WERE INTRODUCED.
- THE TRANSFER ADDRESS DEFINES THE STARTING POINT OF EXECUTION OF A PROGRAM WHEN IT IS LOADED.

STORAGE ALLOCATION

SYMBOL OR BLANK	BSS	ABSOLUTE ADDRESS EXPRESSION

- LOCATION FIELD SYMBOL IS GIVEN THE CURRENT VALUE OF THE LOCATION COUNTER.
- THE VARIABLE FIELD EXPRESSION IS EVALUATED AND THE LOCATION AND ORIGIN COUNTERS ARE INCREMENTED BY THAT AMOUNT.
- SYMBOLS IN THE EXPRESSION MUST HAVE PREVIOUSLY BEEN DEFINED.
- IF ADDRESS EXPRESSION IS INCORRECT, NO SPACE WILL BE RESERVED, BUT A FORCE UPPER WILL OCCUR.
- BSS O FORCES UPPER WITHOUT ALLOCATING STORAGE.

SYMBOL OR BLANK	BSSZ	ABSOLUTE ADDRESS EXPRESSION

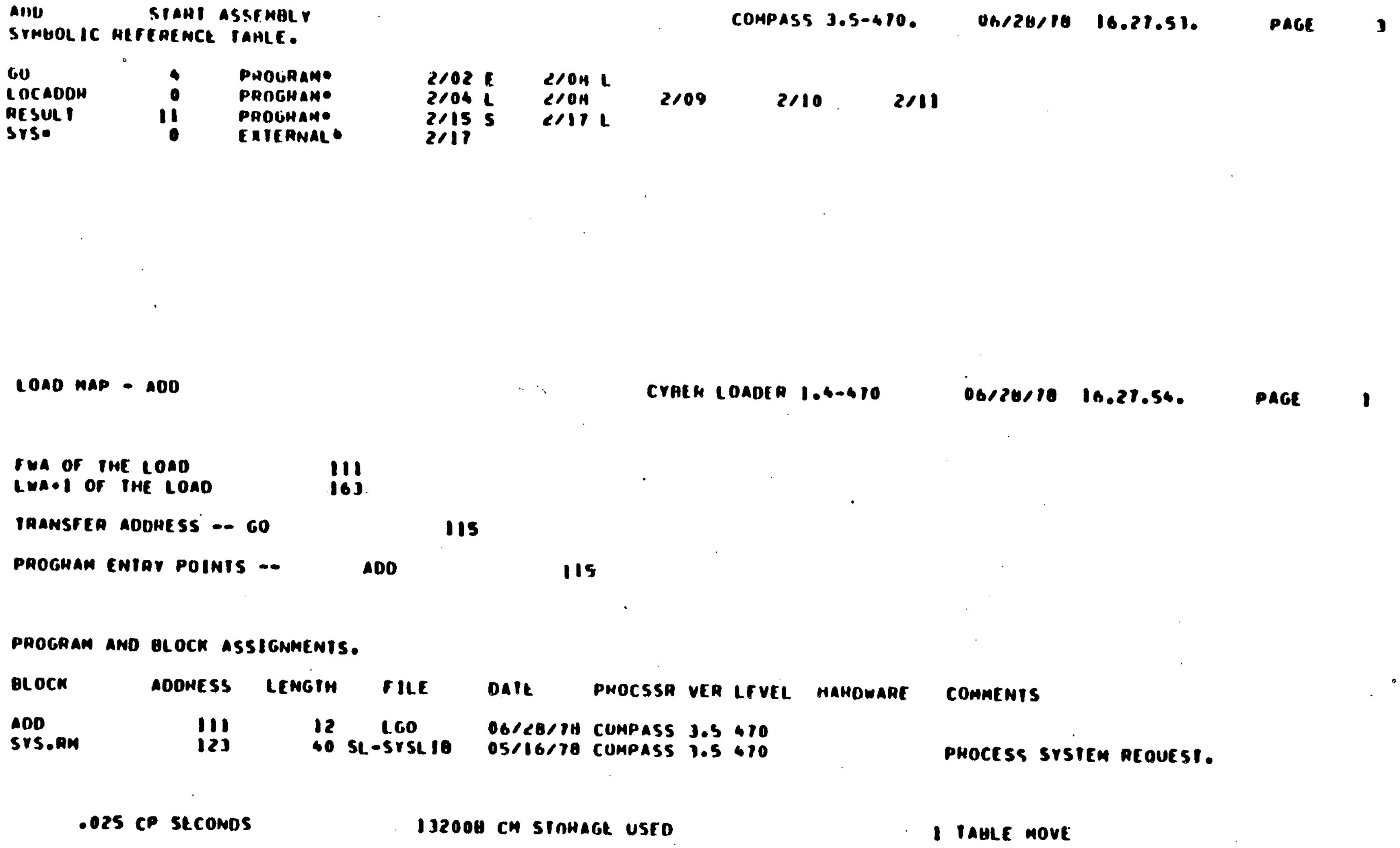
- IDENTICAL TO BSS EXCEPT THAT BSSZ RESERVES AN AREA OF ZERO FILLED WORDS.

BLANK OR SYMBOL	DATA	ABSOLUTE ADDRESS EXPRESSION

- SYMBOL IS GIVEN CURRENT VALUE OF LOCATION COUNTER.
- DATA ITEMS MAY BE OCTAL, DECIMAL, OR DISPLAY CODE CHARACTERS, AND MUST BE FULL-WORD VALUES.
- SEPARATED BY COMMAS.
- TERMINATED BY BLANKS.
- LITERALS MAY NOT BE USED.
- FORCES UPPER.

```
ADD         START ASSEMBLY                              COMPASS 3.5-470.     06/20/70  16.27.53.      PAGE     1
STORAGE ALLOCATION.

        ADDRESS   LENGTH           BINARY CONTROL CARDS.

              0       12           IDENT  ADD          START ASSEMBLY
             12                    END    GO           TERMINATE COMPASS ASSEMBLY


                              ENTRY POINTS.

                              GO              4.


                              EXTERNAL SYMBOLS.

                              SYS=
```

```
 ADD         START ASSEMBLY                              COMPASS 3.5-470.     06/20/70  16.27.53.      PAGE     2


                                             IDENT  ADD          START ASSEMBLY
                                             ENTRY  GO           GO IS AN ENTRY POINT
                                             LIST   D            SHOW DETAIL
     0  77777777777777777750      LOCADDR   DATA   -23,4,21.,-1 DATA FOR PROGRAM
     1  00000000000000000004
     2  00000000000000000025
     3  77777777777777777776
     4  5110000000 +              GO        SA1    LOCADDR      EXECUTION BEGINS HERE
                  5120000001 +              SA2    LOCADDR+1    X2 = 2ND DATA WORD
     5  5130000002 +                        SA3    LOCADDR+2    X3 = 3RD DATA WORD
                  5140000003 +              SA4    LOCADDR+3    X4 = 4TH DATA WORD
     6  36612                               IX6    X1+X2        X6 = 1ST SUM
              36734                         IX7    X3+X4        X7 = 2ND SUM
                  36667                     IX6    X6+X7        X6 = FINAL RESULT
     7  5160000011 +                        SA6    RESULT       STORE ANSWER
                  7160247021                ENDRUN              TERMINATE PROGRAM EXECUTION
    11                     1      RESULT    BSSZ   1            RESERVE WORD OF ZEROS FOR ANSWER

                                DEFAULT SYMBOLS DEFINED BY COMPASS.

     0 X                        SYS=

    12                                      END    GO           TERMINATE COMPASS ASSEMBLY

                      47300B SCM STORAGE USED            10 STATEMENTS         4 SYMBOLS
                             MODEL 174 ASSEMBLY      0.024 SECONDS            10 REFERENCES
```



```
ADD          START ASSEMBLY                                        COMPASS 3.5-470.     06/28/78  16.27.51.      PAGE      3
SYMBOLIC REFERENCE TABLE.

GO             4     PROGRAM*       2/02 E      2/08 L
LOCADDR        0     PROGRAM*       2/04 L      2/08        2/09        2/10        2/11
RESULT        11     PROGRAM*       2/15 S      2/17 L
SYS=           0     EXTERNAL*      2/17




LOAD MAP - ADD                                             CYBER LOADER 1.4-470      06/28/78  16.27.54.      PAGE      1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD          163

TRANSFER ADDRESS -- GO              115

PROGRAM ENTRY POINTS --       ADD              115


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK          ADDRESS   LENGTH     FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

ADD               111        12    LGO       06/28/78 COMPASS 3.5 470
SYS.RM            123        40 SL-SYSLIB    05/16/78 COMPASS 3.5 470                 PROCESS SYSTEM REQUEST.

     .025 CP SECONDS             13200B CM STORAGE USED                 1 TABLE MOVE
```

```
DUMP    RELATIVE                        DMP,111,177.

 00111   77777 77777 77777 77750     00000 00000 00000 00004     00000 00000 00000 00025
 00114   77777 77777 77777 77776     51100 00111 51200 00112     51300 00113 51400 00114     36612 36734 36667 46000

 00120   51600 00122 71602 47021     20650 01000 00125 46000     00000 00000 00000 00001     04000 00136 00000 00000

 00124   01300 00000 00000 00000     04000 00122 00000 00000     51100 00001 03110 00126     54610 04000 00124 46000

 00130   51100 00066 03310 00132     51100 00123 04000 00133     71100 00130 20160 46000     13661 13161 13661 46000

 00134   51600 00124 10611 46000     51100 00001 01000 00123     20652 01000 00125 46000     51100 00001 03110 00137

 00140   04004 00140 61000 46000     51100 00001 03110 00140     71602 20314 04000 00136     20150 36661 01000 00125

 00144   04004 00144 61000 46000     71602 20314 20652 36662     53160 20173 03310 00144     03010 00144 51100 00001

 00150   03110 00146 71100 00001     04000 00143 61000 46000     71603 24616 12661 20651     01000 00125 61000 46000

 00154   04004 00154 61000 46000     73660 20630 12161 73610     20123 03210 00152 20151     13116 20636 51600 00162

 00160   74660 36116 20123 46000     04000 00152 61000 46000     00000 00000 00000 00000     60000 00000 04004 00163

 00177  >60000 00000 04004 00177
```

```
  MFF   NP2-  CYB175-SN1    4LB7/R6A  05/15/78
16.27.51.DON00F0  FROM    /SH
16.27.51.IP  00000256 WORDS - FILE INPUT  , DC 04
16.27.51.DON.     PSD,0278,72CT011A,MILLER
16.27.52.COMPASS.
16.27.53. ASSEMBLY COMPLETE.   47300B SCM USED.
16.27.53.    0.071  CPU SECONDS  ASSEMBLY TIME.
16.27.53.LGO.
16.27.54.DMP,111,177.
16.27.54.OP  00000832 WORDS - FILE OUTPUT , DC 40
16.27.54.MS     3584  WORDS  (   10752 MAX USED)
16.27.54.CPA         .131 SEC.          .131 ADJ.
16.27.54.IO          .631 SEC.          .611 ADJ.
16.27.54.CM       10.085 KWS.           .615 ADJ.
16.27.54.SS                            1.379
16.27.54.PP        2.977 SEC.      DATE 06/28/78
16.27.54.EJ  END OF JOB, SH
```

ASSEMBLER/LOADER

THE ASSEMBLY PROCESS

The input to COMPASS is COMPASS source code as shown on the previous page. COMPASS makes two passes through the source to produce its output - a series of tables which will be used by the loader.

Pass 1 - Symbol Table

On the first pass, the assembler must evaluate all symbolic addresses and table the information so that text can be generated on the second pass. For each program, COMPASS uses 0 as a base address. The loader will have to relocate later when the actual load address is known.

```
          IDENT      ADD
          ENTRY      GO
LOCAD     DATA       -23,4,21,-1
GO        SA1        LOCAD
          SA2        LOCAD+1
          SA3        LOCAD+2
          SA4        LOCAD+3
          IX6        X1+X2
          IX7        X3+X4
          IX6        X6+X7
          SA6        RESULT
          ENDRUN
RESULT    BSS        1
          END        GO
```

LOCATION COUNTER	
POSITION COUNTER	

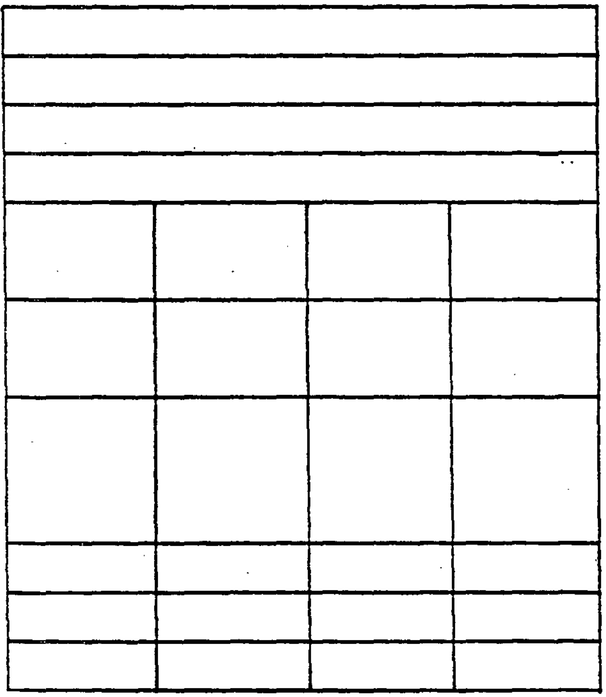
SYMBOL	ADDRESS	TY

Assembler/Pass 1

Pass 2 - Generated Tables

On the second pass through the source code, COMPASS will generate the tables that are needed by the loader. These tables will identify the program (PREFIX, PIDL), indicate how the core image should be generated (TEXT), and suggest the first instruction word to be executed (XFER).

SYMBOL	ADDRESS	TY
GO	000004	E/R
LOCAD	000000	R
RESULT	000011	R



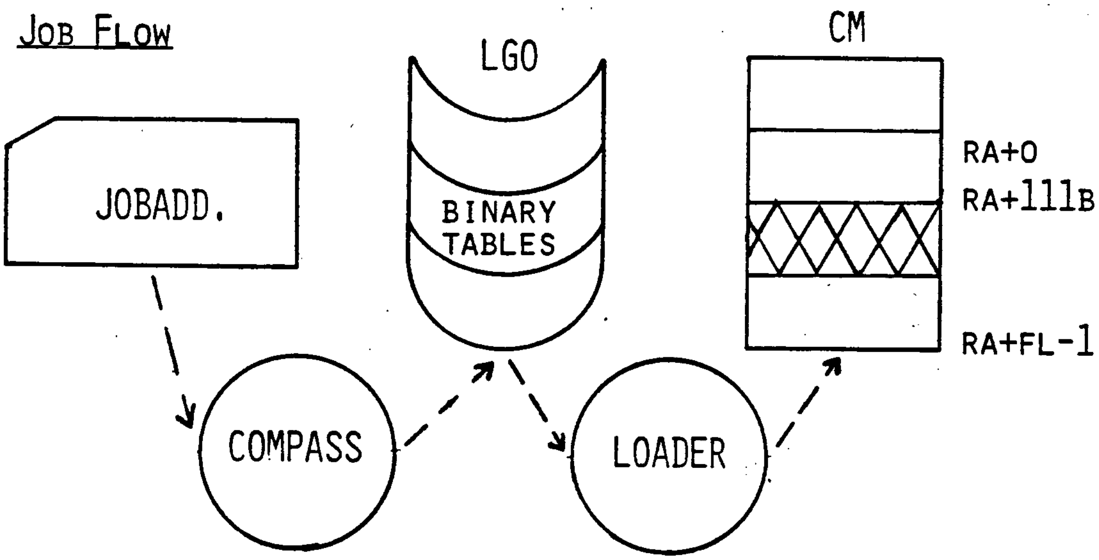
```
                IDENT   ADD             PREFIX/PIDL
                ENTRY   GO              ENTR
        LOCAD   DATA    -23,4,21,-1     TEXT


        GO      SA1     LOCAD           TEXT
                SA2     LOCAD+1         TEXT
                SA3     LOCAD+2         TEXT
                SA4     LOCAD+3         TEXT
                IX6     X1+X2           TEXT
                IX7     X3+X4           TEXT
                IX6     X6+X7           TEXT
                SA6     RESULT          TEXT
                ENDRUN                  TEXT
        RESULT  BSS     1               -SKIP-
                END     GO              XFER
```

COMPASS/LOADER INTERACTION

Deck Structure

```
JOBADD, T10.
COMPASS.
LGO.
7
 8
  9    IDENT ADD
          :
         END GO
6
 7
  8
   9
```

Job Flow



NOTE

COMPASS PREPARES ALL ADDRESSES AS IF THE JOB WOULD RUN AT RA+0. THE BINARY TABLES TELL THE LOADER WHICH ADDRESSES MUST BE FURTHER RELOCATED. IN THE EXAMPLE, 111 IS ADDED TO EACH ADDRESS BECAUSE 111 IS THE LOAD ADDRESS FOR THIS LOAD.

Dump Analysis

CHECK THE LOAD MAP AND DUMP ON PAGE 3-16. NOTICE THAT THE LOAD ADDRESS OF ADD IS 111B AND THAT THE RELOCATABLE ADDRESSES WERE MODIFIED BY THE LOADER IN PREPARATION FOR EXECUTION.

```
ERROR                                                                 COMPASS 3.5-470.      06/27/78  08.58.49.      PAGE      1
STORAGE ALLOCATION.

              ADDRESS   LENGTH                BINARY CONTROL CARDS.

                    0       41                IDENT   ERROR
                   41                         END     START        TRANSFER ADDRESS

                                     BLOCKS      TYPE        ADDRESS      LENGTH

                                     PROGRAM*    LOCAL             0          35
                                     LITERALS*   LOCAL            35           4

                                     ENTRY POINTS.

                                     START              0*            BUF                 7*

                                     EXTERNAL SYMBOLS.

                                     SUBI        SYS=
                                              IDENT   ERROR
                                              ENTRY   START,BUF
                                              EXT     SUB1
                                              LIST    S
                                        *
                                        *             THIS IS AN EXAMPLE OF AN ERROR LISTING
                                        *
        0  46000                         START   NO
                              7          FLAG    EQU     7
              5110000034 +                       SA1     TEMP
7                        64243                   SB2     B3+A,12
        1  7160000010                   *        SX6     10B           DATA
                    5160000007 +                 SA6     BUF           STORE IT
                              35 +       LITERAL LIT     2,4,6         LITERAL
        2  5110000036 +                          SA1     LITERAL+1     GET A LITERAL (2ND LITERAL)
                      10711                      BX7     X1            MOVE IT OVER
        3  5170000010 +                          SA7     BUF+1         STORE IT
                    5170000011 +                 SA,FLAG BUF+2
        4  0100000000 X                          RJ      SUB1          GO TO SUB1
                                                 IF      DEF,FLAG,1
        5  5110000040 +                          SA1     =100
                                                 ENDRUN
                    7160247021          SX6 3REND+4+1                                                        ENDRUN   .1
        6  20650                        LX6 40D                                                              ENDRUN   .1
             0100000000 X               RJ =XSYS=                                                            ENDRUN   .1
                                        ENDM                                                                 ENDRUN   .1
        7                     12        BUF      BSSZ    10            BUFFER IN PROGRAM
       21  0100000000 X                          RJ      SUB1          SHOW LINKAGE
       22  46000                        *        NO
             0100000000 X                        RJ      SUB1
                                        *        JUNK HERE
       23  5160000034 +                          SA6     =STEMP        BSS AT END OF PROGRAM
7                     20673                      LX5     273B          NONFATAL ERROR (NUMBER)
O      24  0000000000                            IX6     X6+10         FATAL ERROR (LETTER)
I      25                      5        B4       BSS     5             B4 ILLEGAL SYMBOL
U      32  5110000000                            SA1     UNDEF         UNDEFINED SYMBOL
       33  00000000000000000000 1      LGO.                            CONTROL CARD OUT OF PLACE SECTION 4.8.1 REF. MAN.

                                       DEFAULT SYMBOLS DEFINED BY COMPASS.

        0 X                            SYS=
       34                              TEMP

                                       CONTENT OF LITERALS BLOCK.

       35  00000000000000000002                 B
       36  00000000000000000004                 D
       37  00000000000000000006                 F
       40  00000000000000000144                A9

       41                                       END     START         TRANSFER ADDRESS

                         47300B SCM STORAGE USED          38 STATEMENTS         18 SYMBOLS
                                MODEL 174 ASSEMBLY      0.045 SECONDS          22 REFERENCES

                                         2 ERRORS IN ERROR
```

```
ERROR                                                             COMPASS 3.5-470.     06/27/78  08.58.49.      PAGE      3
ERROR DIRECTORY.

      O TYPE ERROR              OPERATION FIELD BAD.
             OCCURRED ON PAGES          2

      U TYPE ERROR              UNDEFINED SYMBOL.  VALUE ASSUMED 0.
             OCCURRED ON PAGES          2

      L TYPE ERROR              LOCATION SYMBOL BAD.  SYMBOL NOT DEFINED.
             OCCURRED ON PAGES          2

      T TYPE ERROR              ADDRESS VALUE EXCEEDS FIELD SIZE, RESULT TRUNCATED
             OCCURRED ON PAGES          2
```

```
ERROR                                                                         COMPASS 3.5-470.      06/27/78  08.58.49.       PAGE      4
SYMBOLIC REFERENCE TABLE.

        BUF           7      PROGRAM*           2/02 F     2/13 S     2/17 S     2/18 S     2/27 L
        B4           25      PROGRAM*           2/35 L
        FLAG          7                         2/09 D     2/19       2/20 F
        LGO.         33      PROGRAM*           2/37 L
        LITERAL      35      PROGRAM*           2/14 L     2/15
        START         0      PROGRAM*           2/02 F     2/08 L
        SUB1          0      EXTERNAL*          2/03 X     2/19       2/28       2/30
        SYS=          0      EXTERNAL*          2/25
        TEMP         34      PROGRAM*           2/10       2/32 S
U       UNDEF                   .               2/36
```

```
                                           CYBER LOADER 1.4-470      06/27/78  08.58.49.      PAGE     1
      LOAD MAP -


      FWA OF THE LOAD              111
      LWA+1 OF THE LOAD            111

      TRANSFER ADDRESS --            ADDRESS UNSPECIFIED

      PROGRAM ENTRY POINTS --                           0


   ********** ERROR SUMMARY

       FE0200***ATTEMPT TO LOAD SUPPRESSED BINARY
      NO PROGRAMS READ YET
      LAST FILE ACCESSED- LGO
      ERRORS IN ASSEMBLY


          .011 CP SECONDS           13200B CM STORAGE USED                  1 TABLE MOVE
```

```
DMPX.

 P    000734   A0  000150   B0  000000
 RA   125200   A1  000001   B1  000001      C(A1)=  0102  2401  0000  0000  0000      C(B1)=  0102  2401  0000  0000  0000
 FL   030000   A2  013021   B2  777776      C(A2)=  4000  0000  0002  0000  0111      C(B2)=
 EM   700700   A3  000277   B3  006214      C(A3)=  0000  0000  0000  0000  0000      C(B3)=  2300  0000  0000  0000  0000
 RE   000041   A4  000300   B4  006067      C(A4)=  0000  0000  0000  0000  0000      C(B4)=  2300  0000  0000  0000  0000
 FE   000000   A5  011033   B5  007070      C(A5)=  5100  0000  0001  0000  0014      C(B5)=  5110  0066  5154  7101  3221
 MA   000400   A6  000001   B6  005131      C(A6)=  0102  2401  0000  0000  0000      C(B6)=  2400  0000  0000  0000  0000
               A7  013021   B7  002440      C(A7)=  4000  0000  0002  0000  0111      C(B7)=  5000  0200  0024  4000  2441
 X0   7777  7777  7777  7700  0000
 X1   0000  0000  0000  0000  0000
 X2   0000  0000  0000  0000  0000
 X3   0000  0000  0000  0000  0000
 X4   0000  0000  0000  0000  0000
 X5   7777  7777  7777  7777  7774
 X6   0102  2401  0000  0000  0000
 X7   4000  0000  0002  0000  0111

      00000    00000 00000 00000 00000      01022 40100 00000 00000      00000 00000 00000 00000
      00054   -56110 03110 00054 56120      63310 67410 13414 21422      03140 00057 02300 00000      13443 03040 00060 67405
      00060    56154 63310 13415 21422      61447 77776 03040 00056      07040 00060 51600 00001      04000 00063 00000 00021
      00064    16250 31405 25230 00000      00000 00000 00010 06067      40000 00000 02000 02440      00000 00000 40000 00000
      00070    14071 75755 00000 00000      00000 00000 00000 00000
      00100   -51000 00000 01000 00014

      00634    22611 43700 51700 00340      54130 73710 54711 76160      03260 00637 77101 46000      64240 54771 02220 00000
      00640    53610 70217 77775 54710      04000 02707 00000 00000      54311 73711 46000 37473      03340 00640 54331 73230
      00644    03040 00647 37472 46000      03140 00640 55311 54211      73730 37472 03140 00640      10466 50117 77775 74210
      00650    20173 03310 00654 46000      03010 00641 01000 00752      52120 00002 10644 54311      04000 00642 61000 46000
      00654    71700 00014 01000 00543      04000 00652 61000 46000      04000 00663 61000 46000      04004 00657 61000 46000
      00660    51400 00657 04700 00657      52120 00004 53121 66767      63510 46000 61000 46000      50130 00002 54231 63410
      00664    63321 05350 00665 63330      56160 04340 00677 66661      10611 53620 46000 76230      05670 00664 61000 46000
      00670    73720 75231 01000 01160      20173 54720 03210 00676      54330 77643 63230 22316      77752 21674 13467 37643
      00674    03360 00676 71700 00014      01000 00543 61000 46000      64240 02220 00000 46000      55131 73620 20173 54620
      00700    03310 00703 03010 00703      01000 00746 61000 46000      65241 02220 00000 46000      75231 01000 01160 46000
      00704    71700 00014 01000 00543      65241 02220 00000 46000      04000 00712 61000 46000      04000 02670 00000 00000
      00710    51400 00707 52120 00004      53321 43460 63510 46000      50130 00002 54231 63410      63321 05350 00714 63330
      00714    56160 04340 00677 22601      66661 15714 53620 76230      03170 00713 04000 00670      04000 01742 00000 00000
      00720    71600 31117 51600 00554      66211 43052 71200 00115      01000 00752 61000 46000      51120 00342 15210 11101
      00724    03020 00730 71600 00003      12616 71200 00115 46000      01000 00525 61000 46000      71707 77603 01000 00543
      00730    67221 06200 00723 46000      04000 00717 61000 46000      04000 00744 00000 00000      01300 00000 00000 00000
      00734    04000 01745 00000 00000      51100 00001 03110 00735      54610 04000 00733 46000      51100 00732 10611 46000
      00740    51100 00066 03210 00742      71600 00130 20660 46000      51600 00733 51100 00001      10611 01000 00732 46000
      00744    20652 01000 00734 46000      51100 00001 03110 00745      04000 00573 00000 00000      51100 00001 03110 00746
      00750    71602 20314 04000 00744      20150 36661 01000 00734      04000 00723 00000 00000      71602 20314 20652 36662
      00754    53160 20173 03310 00752      03010 00752 51100 00001      03110 00754 74110 46000      04000 00751 61000 46000
      00760    20223 71603 24614 12662      20651 01000 00734 46000      04000 03640 00000 00000      43266 15262 20222 12212
      00764    03260 00760 13221 76610      20650 12226 73610 20636      51600 00770 74160 12212      04000 00760 61000 46000
      00770    00000 03646 00000 00001      04004 00771 61000 46000      20136 20614 12616 46000      51600 00777 71102 06501
      00774    20150 74460 12616 46000      01000 00734 61000 46000      04000 00771 61000 46000      00000 00000 00000 00000
      01000    00000 00000 61000 46000      71306 42465 43652 15336      71600 01012 72220 00002      20350 20222 12663 20136
      01004    12662 10711 53760 20252      20736 01000 00734 46000      76310 11323 01000 01364      54170 21136 37617 10233
      01010    03060 01000 43101 46000      04000 01000 61000 46000      00000 00000 00000 00000      04000 02227 00000 00000
      01014    04000 01013 61000 46000      00000 00000 61000 46000      10300 74400 10022 46000      51000 01025 61000 46000
      01020    01110 00000 04000 01023      54100 10033 53040 46000      04000 01015 61000 46000      43101 71200 02406 46000
      01024    01000 01715 61000 46000      00000 00000 00000 00000      00000 00000 61000 46000      10300 74400 51600 01025
      01030    10022 54060 61000 46000      01210 00000 04000 01033      10033 53040 04000 01026      43101 71200 02410 46000
      01034    01000 01715 61000 46000
```

```
  MFF   NR2-  CYB175-SN1    4LB7/R6B  05/15/78
 08.58.47.DON005G  FROM    /SH
 08.58.47.IP  00000384 WORDS - FILE INPUT  , DC 04
 08.58.47.DON.      PSD,0278,72CT011A,MILLER
 08.58.48.COMPASS.
 08.58.49.           3 WARNING MESSAGES IN ERROR
 08.58.49.           2 ERRORS IN ERROR
 08.58.49. ASSEMBLY ERRORS.     473008 SCM USED.
 08.58.49.     0.091  CPU SECONDS  ASSEMBLY TIME.
 08.58.49.LGO.
 08.58.49.           FATAL LOADER ERROR -  SEE MAP
 08.58.50.EXIT.
 08.58.50.OP  00001600 WORDS - FILE OUTPUT , DC 40
 08.58.50.MS    3584  WORDS  (    7168 MAX USED)
 08.58.50.CPA        .128 SEC.         .128 ADJ.
 08.58.50.IO         .624 SEC.         .624 ADJ.
 08.58.50.CM      10.499 KWS.          .640 ADJ.
 08.58.50.SS                          1.393
 08.58.50.PP        2.715 SEC.     DATE 06/27/78
 08.58.50.EJ  END OF JOB, SH
```

SUBPROGRAM STRUCTURE

Main Program

```
        IDENT   MAIN               ROUTINE IS NAMED MAIN
        ENTRY   EEEEEEE            ENTRY POINT
        EXT     XXXXXXX            XXXXXXX IS AN EXTERNAL SYMBOL
EEEEEEE   .                        ENTRY POINT OF MAIN ROUTINE, CONTAINS
          .                        THE FIRST INSTRUCTION
          .
          .
          .
        RJ      XXXXXXX            JUMP TO A SUBROUTINE
          .
          .
        ENDRUN                     STOP EXECUTION
        END     EEEEEEE            EEEEEEE IS A TRANSFER ADDRESS
```

Subroutine

```
        IDENT   SUB                ROUTINE IS NAMED SUB
        ENTRY   XXXXXXX            DECLARES XXXXXXX AS AN ENTRY POINT
XXXXXXX DATA    0                  DUMMY WORD MODIFIED BY RJ INSTRUCTION
          .
          .
        EQ      XXXXXXX            JUMP TO ENTRY POINT TO LEAVE SUB
        END                        NO TRANSFER ADDRESS
```

LESSON 4

INSTRUCTIONS

LESSON PREVIEW:

THIS LESSON INTRODUCES EACH GROUP OF INSTRUCTIONS, EXCEPT FLOATING POINT, AND DISCUSSES EACH IN DETAIL.

REFERENCES:

CHAPTER 8 COMPASS REF. #60492600

TRAINING AIDS:

VISUAL SET V4
PROGRAM COMPILATION LISTINGS OF DECK 5A

PROJECTS:

PROGRAMMING PROJECT #2
HOMEWORK

OBJECTIVES:

AT THE END OF THIS LESSON THE STUDENT WILL BE ABLE TO:

1. LIST THE COMPASS INSTRUCTIONS GROUPS, FUNCTIONS AND THE ARITHMETIC UNIT THAT EXECUTES THEM.
2. EXPLAIN INTEGER ARITHMETIC ON THE CYBER 70/170.
3. MANIPULATE DATA IN WORDS USING BOOLEAN LOGIC.

SET INSTRUCTION

INCREMENT UNIT

Set A Register to	A Register	± Operand	(50)
	B Register	± Operand	(51)
	X Register	± Operand	(52)
	X Register	+ B Register	(53)
	A Register	± B Register	(54-55)
	B Register	± B Register	(56-57)
Set B Register to	A Register	± Operand	(60)
	B Register	± Operand	(61)
	X Register	± Operand	(62)
	X Register	+ B Register	(63)
	A Register	± B Register	(64-65)
	B Register	± B Register	(66-67)
Set X Register to	A Register	± Operand	(70)
	B Register	± Operand	(71)
	X Register	± Operand	(72)
	X Register	+ B Register	(73)
	A Register	± B Register	(74-75)
	B Register	± B Register	(76-77)

SET INSTRUCTIONS

- * Use the two increment units
- * Have an 18-bit ADDER

ALLOWABLE OPERATIONS

30-Bit Instructions

$$\text{SET}\left.\begin{matrix}A\\B\\X\end{matrix}\right\}\text{ TO }\left.\begin{matrix}A\\B\\X\end{matrix}\right\}\begin{matrix}\\+K\\-\end{matrix}$$

A constant used as an operand or
A tag instead of a second source register

15-Bit Instructions

$$\begin{array}{ll}X & +B\\ \left.\begin{matrix}A\\B\end{matrix}\right\} & \begin{matrix}\pm B\\ \\ \end{matrix}\end{array}$$

Combination of 2 source registers

- * Overflow is ignored in the increment unit.
- * Any operands coming from X registers are truncated to 18-bits before going into the unit.
- * Operations X ± X, A ± A, X-B, or X ± A are illegal.

```
LOADNR                                                    COMPASS 3.5-470.     06/27/78  16.17.43.      PAGE     1
STORAGE ALLOCATION.

           ADDRESS   LENGTH             BINARY CONTROL CARDS.

                 0        6          IDENT      LOADNR
                 6                      END     LOADNR      TRANSFER ADDRESS

                                 ENTRY POINTS.

                                 LOADNR            0+

                                 EXTERNAL SYMBOLS.

                                 SYS=
```

```
LOADNR                                                    COMPASS 3.5-470.     06/27/78  16.17.43.      PAGE     2

                                       IDENT      LOADNR
                                          ENTRY   LOADNR
                                 *
                                 *        LOAD A NUMBER FROM CENTRAL MEMORY
                                 *        INTO A X REGISTER -
                                 *        TRANSFER THE NUMBER TO AN APPROPRIATE REGISTER -
                                 *        STORE THE NUMBER INTO CENTRAL MEMORY
                                 *
      0  5110000003 +            LOADNR   SA1     DAT          LOAD THE NUMBER
                   10611                  BX6     X1           TRANSFER THE NUMBER
      1  5160000005 +                     SA6     DAT+2        STORE NUMBER
                   7160247021             ENDRUN               TERMINATE JOB NORMALLY
      3  00000000000001234567    DAT      DATA    1234567B     NUMBER
      4                    2              BSSZ    2            PLACE TO STORE NUMBER
      6                                   END     LOADNR       TRANSFER ADDRESS

                     47300B SCM STORAGE USED            18 STATEMENTS        3 SYMBOLS
                            MODEL 174 ASSEMBLY       0.021 SECONDS           6 REFERENCES
```

```
LOADNR                                                    COMPASS 3.5-470.     06/27/78  16.17.43.      PAGE     3
SYMBOLIC REFERENCE TABLE.

DAT            3     PROGRAM*       2/09       2/11 S     2/13 L
LOADNR         0     PROGRAM*       2/02 E     2/09 L
SYS=           0     EXTERNAL*      2/13
```

```
LOAD MAP - LOADNR                                            CYBER LOADER 1.4-470       06/27/78  16.17.44.       PAGE      1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD          157

TRANSFER ADDRESS -- LOADNR            111

PROGRAM ENTRY POINTS --       LOADNR           111


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK       ADDRESS   LENGTH    FILE     DATE     PROCSSR VER LEVEL  HARDWARE   COMMENTS

LOADNR          111        6    LGO     06/27/78 COMPASS 3.5 470
SYS.RM          117       40 SL-SYSLIB  05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.


     .025 CP SECONDS              13200B CM STORAGE USED                      1 TABLE MOVE
```

```
DUMP    RELATIVE                          DMP(111,117)

    00111   51100 00114 10611 46000        51600 00116 71602 47021        20650 01000 00121 46000
    00114   00000 00000 00012 34567        00000 00000 00000 00000        00000 00000 00012 34567        04000 00132 00000 00000
```

```
  MFF   NB2-  CYB175-SN1    4LB7/R6B  05/15/78
16.17.41.DON00L9  FROM    /SH
16.17.41.IP  00000256 WORDS - FILE INPUT  , DC 04
16.17.41.DON.     PSD,0278,72CT011A,MILLER
16.17.42.COMPASS.
16.17.43. ASSEMBLY COMPLETE.   47300B SCM USED.
16.17.43.    0.069  CPU SECONDS  ASSEMBLY TIME.
16.17.43.LGO.
16.17.44.DMP(111,117)
16.17.44.OP  00000640 WORDS - FILE OUTPUT , DC 40
16.17.44.MS    3584  WORDS  (    7168 MAX USED)
16.17.44.CPA         .122 SEC.          .122 ADJ.
16.17.44.IO          .628 SEC.          .628 ADJ.
16.17.44.CM         9.760 KWS.          .595 ADJ.
16.17.44.SS                            1.346
16.17.44.PP         3.224 SEC.     DATE 06/27/78
16.17.44.EJ  END OF JOB, SH
```

```
    PROGRAM TESTSET    74/176  OPT=1                          FTN 4.7+470        06/27/78  16.17.57        PAGE      1

  1                PROGRAM    TESTSET(OUTPUT)
                   COMMON //ANSWER(4)
                   CALL SETS
                   PRINT 1,(ANSWER(I),I=1,4)
  5              1 FORMAT(1X,4(5X,O20))
                   END

  SETS                                                       COMPASS 3.5-470.     06/27/78  16.17.58.       PAGE      1
  STORAGE ALLOCATION.

            ADDRESS    LENGTH                BINARY CONTROL CARDS.

                  0         14               IDENT   SETS
                 14                          END

                                 BLOCKS      TYPE       ADDRESS    LENGTH

                                 PROGRAM*    LOCAL            0        14
                                 //          COMMON           0         4

                                 ENTRY POINTS.

                                 SETS               0*
```

```
  SETS                                                      COMPASS 3.5-470.     06/27/78  16.17.58.       PAGE      2

                                        IDENT   SETS
                                        ENTRY   SETS
      0  46000                    SETS  NO
      1  5110000010 +           +       SA1     BUF+0
                   5120000011 +         SA2     BUF+1
      2  5130000012 +                   SA3     BUF+2
                   5140000013 +         SA4     BUF+3
      3  6110200000                     SB1     200000B
                   73611                SX6     B1+X1
      4  5160000000 C                   SA6     ANS
                   73721                SX7     B1+X2
      5  5170000001 C                   SA7     ANS+1
                   73631                SX6     B1+X3
      6  5160000002 C                   SA6     ANS+2
                   73741                SX7     B1+X4
      7  5170000003 C                   SA7     ANS+3
                   0400000000 +         EQ      SETS
     10  00000000000006123456     BUF   DATA    6123456B
     11  00000000000001623456           CON     1623456B
     12  00000000000001312345           DATA    1312345B
     13  00000000000006312345           CON     6312345B
                                        USE     //
      0                      4    ANS   BSS     4
     14                                 END

                 47300B SCM STORAGE USED             24 STATEMENTS         3 SYMBOLS
                          MODEL 174 ASSEMBLY      0.029 SECONDS           13 REFERENCES
```

```
  MFF   NB2-  CYB175-SN1    4LBT/R6B  05/15/78
16.17.55.DONOOMA  FROM    /SH
16.17.55.IP  00000320 WORDS - FILE INPUT  , DC 04
16.17.55.DON.      PSD,027A,72CT011A,MILLER
16.17.56.FTN,R=0.
16.17.59.      .103 CP SECONDS COMPILATION TIME
16.17.59.MAP,PART.
16.17.59.LGO.
16.18.03.      END  TESTSET
16.18.03.      .020 CP SECONDS EXECUTION TIME
16.18.03.OP  00001152 WORDS - FILE OUTPUT , DC 40
16.18.03.MS    3584  WORDS  (   17920 MAX USED)
16.18.03.CPA        .272 SEC.          .272 ADJ.
16.18.03.IO        1.582 SEC.         1.582 ADJ.
16.18.03.CM       29.518 KWS.         1.801 ADJ.
16.18.03.SS                           3.666
16.18.03.PP        6.529 SEC.     DATE 06/27/78
16.18.03.EJ  END OF JOB, SH
```

```
LOAD MAP - TESTSET                                                     CYBER LOADER 1.4-470        06/27/78  16.18.01.       PAGE       1


    FWA OF THE LOAD              111
    LWA+1 OF THE LOAD           7257

    TRANSFER ADDRESS -- TESTSET          2173

    PROGRAM ENTRY POINTS --       TESTSET          2173


    PROGRAM AND BLOCK ASSIGNMENTS.

    BLOCK        ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

    TESTSET          111     2077    LGO      06/27/78 FTN      4.7 470    767X I     PROGRAM   OPT=1
    SETS            2210       14    LGO      06/27/78 COMPASS 3.5 470
    SYSAID=         2224        1 SL-FORTRAN  05/02/78 COMPASS 3.5 470               LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
    /STP.END/       2225        1
    /FCL.C./        2226       25
    /Q8.IO./        2253       77
    Q2NTRY=         2352        0 SL-FORTRAN  05/02/78 COMPASS 3.5 470               FCL INITIALIZATION ROUTINE.
    /FCL=ENT/       2352       42
    COMIO=          2414       14 SL-FORTRAN  05/02/78 COMPASS 3.5 470               COMMON CODED I/O ROUTINES AND CONSTANTS.
    FCL=FDL         2430       40 SL-FORTRAN  05/02/78 COMPASS 3.5 470               FCL CAPSULE LOADING
    FMTAP=          2470      377 SL-FORTRAN  05/02/78 COMPASS 3.5 470               CRACK APLIST AND FORMAT FOR KODER/KRAKER.
    FORUTL=         3067       46 SL-FORTRAN  05/02/78 COMPASS 3.5 470               FCL MISC. UTILITIES.
    GETFIT=         3135       61 SL-FORTRAN  05/02/78 COMPASS 3.5 470               LOCATE AN FIT GIVEN A FILE NAME.
    KODER=          3216      461 SL-FORTRAN  05/02/78 COMPASS 3.5 470               OUTPUT FORMAT INTERPRETER.
    OUTC=           3677      150 SL-FORTRAN  05/02/78 COMPASS 3.5 470               FORMATTED WRITE FORTRAN RECORD.
    FECMSK=         4047       41 SL-FORTRAN  05/02/78 COMPASS 3.5 470               INITIALIZE CONSTANTS.
    FLTOUT=         4110      315 SL-FORTRAN  05/02/78 COMPASS 3.5 470               COMMON FLOATING OUTPUT CODE
    FORSYS=         4425      302 SL-FORTRAN  05/02/78 COMPASS 3.5 470               FORTRAN OBJECT LIBRARY UTILITIES.
    OUTCOM=         4727      204 SL-FORTRAN  05/02/78 COMPASS 3.5 470               COMMON OUTPUT CODE
    CMF.ALF         5133      160 SL-SYSLIB   05/02/78 COMPASS 3.5 470               CMM V1.1 - ALLOCATE FIXED.
    CMF.CSF         5313        6 SL-SYSLIB   05/02/78 COMPASS 3.5 470               CMM V1.1 - CHANGE SPECS FIXED.
    CMM.FFA         5321       14 SL-SYSLIB   05/02/78 COMPASS 3.5 470               CMM V1.1 - FIXED FREE ALGORITHM.
    CMF.FRF         5335       36 SL-SYSLIB   05/02/78 COMPASS 3.5 470               CMM V1.1 - FREE FIXED.
    CMM.R           5373      213 SL-SYSLIB   05/02/78 COMPASS 3.5 470               CMM V1.1 - RESIDENT SUBROUTINES.
    CMF.SLF         5606       22 SL-SYSLIB   05/02/78 COMPASS 3.5 470               CMM V1.1 - SHRINK AT LWA FIXED.
    CTL$RM          5630      601 SL-SYSLIB   05/02/78 COMPASS 3.5 470               CRM CONTROLLING ROUTINE.
    ERR$RM          6431       25 SL-SYSLIB   05/02/78 COMPASS 3.5 470               CRM ERROR PROCESSOR ENTRY.
    LIST$RM         6456       66 SL-SYSLIB   05/02/78 COMPASS 3.5 470               CRM - ALLOCATE SPACE FOR LIST OF FILES
    /FDL.COM/       6544       14
    FDL.RES         6560      211 SL-SYSLIB   05/16/78 COMPASS 3.5 470               FAST DYNAMIC LOADER RESIDENT.
    FDL.MMI         6771      222 SL-SYSLIB   05/16/78 COMPASS 3.5 470               FDL MEMORY MANAGER INTERFACE.
    SYS.RM          7213       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470               PROCESS SYSTEM REQUEST.
     //             7253        4


          .117 CP SECONDS               23300B CM STORAGE USED                       11 TABLE MOVES



00000000000000123456     00000000000000023457     77777777777777512345     77777777777777512345
```

ARITHMETIC IN THE INCREMENT UNIT

INSTRUCTION SX6 B1+X1

$$\begin{array}{lcrr}
(X1) & = & 00000000000006123456_8 & 123456 \\
(B1) & = & \underline{\qquad\qquad\qquad\qquad 200000}_8 & \underline{200000} \\
(X6) & = & 00000000000000323456_8 & 323456
\end{array}$$

TRUNCATION AND SIGN EXTENSION

INSTRUCTION SX7 B1+X2

$$\begin{array}{lcrrr}
(X2) & = & 00000000000001623456_8 & & 623456 \\
(B1) & = & \underline{\qquad\qquad\qquad\qquad 200000}_8 & & \underline{200000} \\
(X7) & = & 00000000000000023457_8 & 1 & 023456 \\
 & & \text{END-AROUND-CARRY} & \llcorner & \longrightarrow 1 \\
 & & \text{TRUNCATION AND SIGN EXTENSION} & & 023457
\end{array}$$

INSTRUCTION SX6 B1+X3

$$\begin{array}{lcrr}
(X3) & = & 00000000000001312345_8 & 312345 \\
(B1) & = & \underline{\qquad\qquad\qquad\qquad 200000}_8 & \underline{200000} \\
(X6) & = & 77777777777777512345_8 & 512345
\end{array}$$

SIGN EXTENSION AND TRUNCATION

INSTRUCTION SX7 B1+X4

$$\begin{array}{lcrr}
(X4) & = & 00000000000006312345_8 & 312345 \\
(B1) & = & \underline{\qquad\qquad\qquad\qquad 200000}_8 & \underline{200000} \\
(X7) & = & 77777777777777512345_8 & 512345
\end{array}$$

SIGN EXTENSION AND TRUNCATION

LONG ADD UNIT

(INTEGER ARITHMETIC)

ADD A "X" REGISTER TO A "X" REGISTER

SUBTRACT A "X" REGISTER FROM A "X" REGISTER

60-BIT ARITHMETIC

ADD UNIT

(FLOATING POINT ARITHMETIC)

ADD - TWO "X" REGISTERS, SINGLE PRECISION OR DOUBLE PRECISION.

ADD - TWO "X" REGISTERS, SINGLE PRECISION ROUNDED.

SUBTRACT - TWO "X" REGISTERS, SINGLE PRECISION OR DOUBLE PRECISION.

SUBTRACT - TWO "X" REGISTERS, SINGLE PRECISION ROUNDED.

96-BIT ADDER PLUS EXPONENT

MULTIPLY UNIT

(FLOATING POINT ARITHMETIC)

FX	MULTIPLY SINGLE PRECISION (1ST HALF OF ADDER)
RX	MULTIPLY SINGLE PRECISION ROUNDED
DX	MULTIPLY DOUBLE PRECISION (2ND HALF OF ADDER)

96-BIT ADDER PLUS EXPONENETS

DIVIDE UNIT

(Floating Point Arithmetic)

FX	Divide
RX	Rounded divide
NO	No operation
CX	Count one bits

BRANCH UNIT

(JUMPS)

ZR	Test the value of a "X" or "B" register
NZ	
PL	
NG	
IR	Test the value of a "X" register
OR	
DF	
ID	
EQ	Compare a "B" register to a "B" register
NE	
GE	
LT	
PS	Program Stop
JP	Go to K+Bi
RJ	Return Jump
XJ	Central Exchange Jump
RE	READ EXTENDED CORE STORAGE
WE	WRITE EXTENDED CORE STORAGE

```
JUMP                                                          COMPASS 3.5-470.     06/27/78  09.00.32.       PAGE      2

                                                   IDENT   JUMP
                                                   ENTRY   ENTER
                                           *
                                           *
                                           *       ILLUSTRATE VARIOUS USES OF JUMP INSTRUCTION
                                           *
                                           *
                                           *       EXAMPLE 1 - JUMP TABLE - JUMPS TO ONE OF THREE ROUTINES
                                           *                                DEPENDING UPON THE VALU PASSES IN B1
                                           *
     0                     1       ENTER   BSSZ    1
     1  63110                              SB1     X1
            6120000003                     SB2     3
     2  0710000000 X                       NG      B1,=XERROR
                  0612000000 X             GE      B1,B2,=XERROR
     3  0211000004 +                       JP      TABLE+B1
                                           *
     4  0400000000 X               TABLE   EQ      =XROUTIN0
     5  0400000000 X                       EQ      =XROUTIN1   NOTE THE FORCING UPPER
     6  0400000000 X                       EQ      =XROUTIN2   COULD BE      JP   =XROUTIN2
                                           *
                                           *
                                           *       EXAMPLE 2 - JUMP TO MEMORY LOCATION HOLDADR
                                           *
                                           *
     7  6110000000 C                       SB1     HOLDADR
                  0211000000               JP      B1          EQUIVALENT TO JP  0+B1
                                           *
                                           *
                                           *       EXAMPLE 3 - NON INDEXED JUMP
                                           *
                                           *
    10  0200000000 +                       JP      ENTER       EQUIVALENT TO    JP  B0+ENTER
    11  0400000000 +                       EQ      ENTER       EQUIVALENT TO    EQ  B0,B0,ENTER
                                           *
                                           *
                                           USE     /DON/
     0                     1       HOLDADR BSS     1
    12                                     END     ENTER

                  473000B SCM STORAGE USED              39 STATEMENTS          7 SYMBOLS
                        MODEL 174 ASSEMBLY           0.034 SECONDS            13 REFERENCES


JUMP                                                          COMPASS 3.5-470.     06/27/78  09.00.32.       PAGE      3.
SYMBOLIC REFERENCE TABLE.

ENTER           0      PROGRAM*        2/02 E     2/11 L     2/33       2/34
ERROR           0      EXTERNAL*       2/14       2/15
HOLDADR         0      DON             2/26       2/30 L
ROUTIN0         0      EXTERNAL*       2/18
ROUTIN1         0      EXTERNAL*       2/19
ROUTIN2         0      EXTERNAL*       2/20
TABLE           4      PROGRAM*        2/16       2/18 L
```

```
                                                           COMPASS 3.5-470.      06/27/78  09.00.32.       PAGE      1
JUMP
STORAGE ALLOCATION.

          ADDRESS   LENGTH              BINARY CONTROL CARDS.

                0       12              IDENT  JUMP
               12                       END    ENTER


                                    BLOCKS     TYPE       ADDRESS     LENGTH

                                    PROGRAM*   LOCAL            0         12
                                    DON        COMMON           0          1


                                    ENTRY POINTS.

                                    ENTER            0*


                                    EXTERNAL SYMBOLS.

                                    ROUTIN2    ROUTIN1    ROUTIN0    ERROR
```

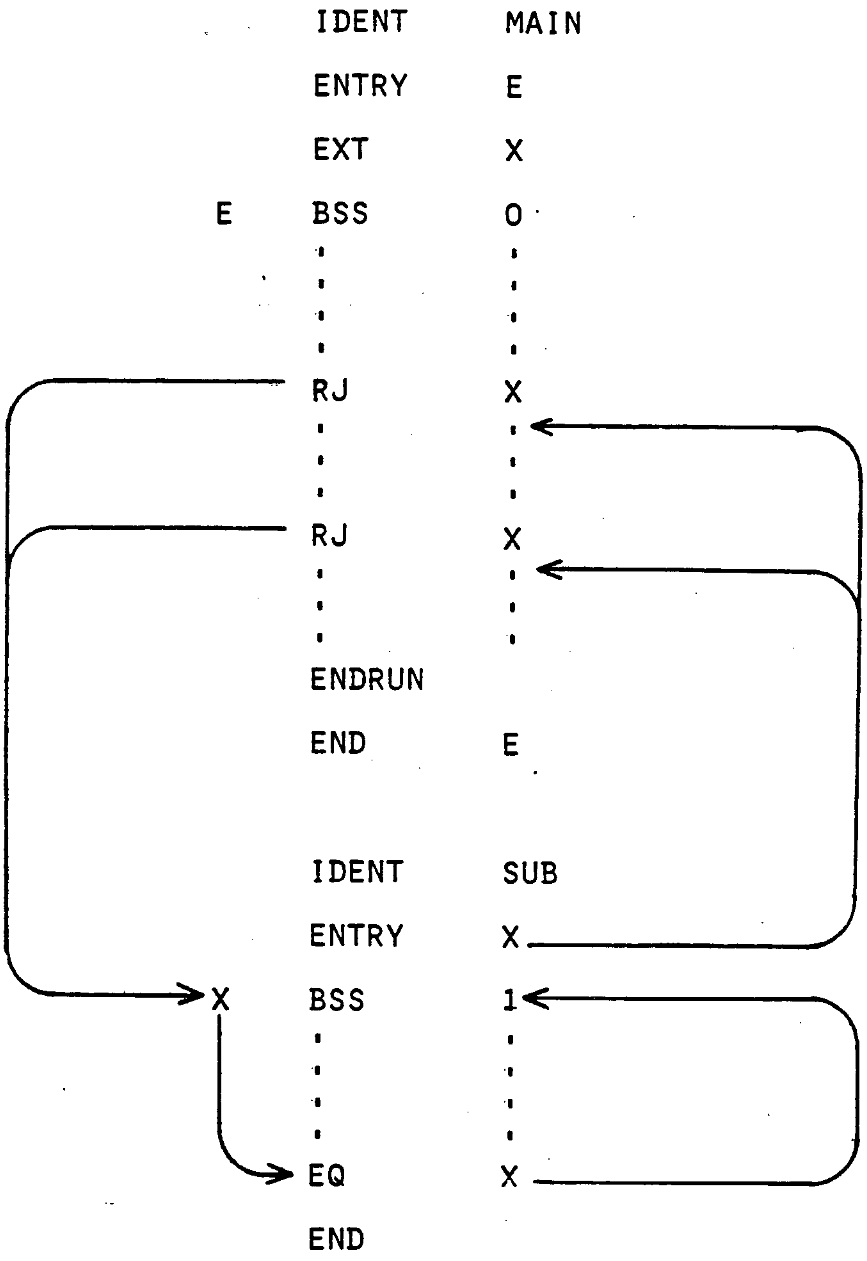
```
 MFF   NA2-  CYB175-SN1    4L87/R68  05/15/78
09.00.31.DON005J  FROM    /SH
09.00.31.IP  00000256 WORDS - FILE INPUT  , DC 04
09.00.31.DON.     PSD,0278,72CT011A,MILLER
09.00.32.COMPASS.
09.00.33. ASSEMBLY COMPLETE.   47300B SCM USED.
09.00.33.    0.085  CPU SECONDS  ASSEMBLY TIME.
09.00.33.OP  00000640 WORDS - FILE OUTPUT , DC 40
09.00.33.MS    3584  WORDS  (   10752 MAX USED)
09.00.33.CPA        .109 SEC.          .108 ADJ.
09.00.33.IO         .474 SEC.          .474 ADJ.
09.00.33.CM       8.514 KWS.           .519 ADJ.
09.00.33.SS                           1.102
09.00.33.PP       2.291 SEC.     DATE 06/27/78
09.00.33.EJ  END OF JOB, SH
```

```
JUMPS                                                               COMPASS 3.5-470.     06/27/78  08.56.05.      PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH                BINARY CONTROL CARDS.

                0       15                IDENT  JUMPS
               15                         END    JUMP


                                  ENTRY POINTS.

                                  JUMP            0+


                                  EXTERNAL SYMBOLS.

                                  SYS=
```

```
JUMPS                                                               COMPASS 3.5-470.     06/27/78  08.56.05.      PAGE      2


                                           IDENT   JUMPS
                                           ENTRY   JUMP
                                  *
                                  *
                                  *        CODE GENERATED BY VARIOUS JUMP INSTRUCTIONS
                                  *
                                  *
      0  0700000014 +              JUMP    JP      TAG         UNCONDITIONAL JUMP
      1  0400000014 +                      EQ      B0,B0,TAG
                   0400000014 +            EQ      TAG
      2  0100000014 +                      RJ      TAG         VOIDS STACK IN ALL SYSTEMS
                                  *
      3  0410000014 +                      EQ      B1,B0,TAG   CONDITIONAL JUMP B REGISTER EQUAL ZERO
                   0410000014 +            ZR      B1,TAG
                                  *
      4  0520000014 +                      NE      B2,B0,TAG   CONDITIONAL JUMP B REGISTER UNEQUAL TO ZERO
                   0520000014 +            NZ      B2,TAG
                                  *
      5  0630000014 +                      GE      B3,B0,TAG   POSITIVE B REGISTER TEST
                   0630000014 +            LE      B0,B3,TAG
      6  0630000014 +                      PL      B3,TAG
                                  *
                   0740000014 +            LT      B4,B0,TAG   NEGATIVE B REGISTER TEST
      7  0740000014 +                      GT      B0,B4,TAG
                   0740000014 +            NG      B4,TAG
                                  *
     10  0311000014 +                      NZ      X1,TAG      UNEQUAL ZERO X REGISTER TEST
                   0301000014 +            ZR      X1,TAG      EQUAL   ZERO X REGISTER TEST
     11  0321000014 +                      PL      X1,TAG      POSITIVE     X REGISTER TEST
                   0331000014 +            NG      X1,TAG      NEGATIVE     X REGISTER TEST
                                  *
     12  7160247021                        ENDRUN
     14                    1        TAG    BSS     1           A LOCATION TO JUMP TO
     15                                    END     JUMP

                   47300B SCM STORAGE USED           37 STATEMENTS        3 SYMBOLS
                        MODEL 174 ASSEMBLY       0.041 SECONDS           22 REFERENCES
```

```
JUMPS                                                                COMPASS 3.5-470.     06/27/78  08.56.05.       PAGE      3
SYMBOLIC REFERENCE TABLE.

JUMP            0      PROGRAM*        2/02 E     2/08 L
SYS=            0      EXTERNAL*       2/33
TAG            14      PROGRAM*        2/08       2/11       2/16       2/20       2/24       2/28       2/33 L
                                       2/09       2/13       2/17       2/21       2/25       2/29
                                       2/10       2/14       2/19       2/23       2/27       2/30
```

```
  MFF   NB2-  CYB175-SN1    4LB7/R6B  05/15/78
08.56.01.DON0044  FROM    /SH
08.56.01.IP  00000256 WORDS - FILE INPUT  , DC 04
08.56.01.DON.     PSD.0278.72CT011A.MILLER
08.56.04.COMPASS.
08.56.05. ASSEMBLY COMPLETE.   47300B SCM USED.
08.56.05.    0.093  CPU SECONDS  ASSEMBLY TIME.
08.56.05.OP  00000576 WORDS - FILE OUTPUT , DC 40
08.56.05.MS     3584  WORDS  (    7168 MAX USED)
08.56.05.CPA         .126 SEC.          .126 ADJ.
08.56.05.IO          .473 SEC.          .473 ADJ.
08.56.05.CM         8.531 KWS.          .520 ADJ.
08.56.05.SS                            1.121
08.56.05.PP         4.012 SEC.     DATE 06/27/78
08.56.05.EJ  END OF JOB, SH
```



```
        IDENT   MAIN
        ENTRY   E
        EXT     X
E       BSS     0
        .       .
        .       .
        .       .
        .       .
        RJ      X
        .       .
        .       .
        .       .
        RJ      X
        .       .
        .       .
        .       .
        ENDRUN
        END     E

        IDENT   SUB
        ENTRY   X
X       BSS     1
        .       .
        .       .
        .       .
        .       .
        EQ      X
        END
```

LINKAGE · ENTRY,EXT,RJ

```
RJTEST                                                    COMPASS 3.5-470.     06/27/78  08.52.06.      PAGE     2

                                          IDENT   RJTEST
                                          ENTRY   RJTEST
                                   .
                                   .
                                   .      EXAMPLE OF CODE GENERATED BY A RETURN JUMP RJ INSTRUCTION
                                   .
                                   .
    0  0100000002 +               RJTEST  RJ      SUB
    1  0200000000                         JP      0
                                   .
                                   .
                                   .
    2  77777777777777777777       SUB     DATA    -0
    3  0400000002 +                       EQ      SUB
    4                                     END     RJTEST

                   47300B SCM STORAGE USED            15 STATEMENTS          2 SYMBOLS
                            MODEL 174 ASSEMBLY      0.012 SECONDS            5 REFERENCES

      LOAD MAP - RJTEST                               CYBER LOADER 1.4-470       06/27/78  08.52.06.      PAGE     1


      FWA OF THE LOAD              111
      LWA+1 OF THE LOAD            115

      TRANSFER ADDRESS -- RJTEST           111

      PROGRAM ENTRY POINTS --       RJTEST           111


      PROGRAM AND BLOCK ASSIGNMENTS.

      BLOCK        ADDRESS    LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

      RJTEST          111        4     LGO      06/27/78 COMPASS 3.5 470


          .007 CP SECONDS              13200B CM STORAGE USED                   1 TABLE MOVE




                DUMP     RELATIVE                        DMP(111,117)

                  00111    01000 00113 61000 46000         02000 00000 61000 46000          04000 00112 00000 00000
                  00114    04000 00113 61000 46000         60000 00000 04004 00115   00117>60000 00000 04004 00117
```

```
                                                                              COMPASS 3.5-470.      06/27/78  08.52.06.        PAGE      1
        RJTEST
        STORAGE ALLOCATION.

                        ADDRESS    LENGTH                   BINARY CONTROL CARDS.

                              0         4                   IDENT   RJTEST
                              4                             END     RJTEST


                                                  ENTRY POINTS.

                                                  RJTEST          0*


DMPX.

 P    000000   A0   000200   B0   000000       C(A1)=   0000   0000   0000   0000   0000       C(B1)=   0000   0000   0000   0000   0000
 RA   125200   A1   000001   B1   000001       C(A2)=   1505   1520   0000   0000   0061       C(B2)=   0000   0000   0000   0000   0000
 FL   000200   A2   000060   B2   000002       C(A3)=   0000   0000   0000   0000   0000       C(B3)=
 EM   700700   A3   000057   B3   012733       C(A4)=   0000   0000   0000   0000   0000       C(B4)=
 RE   000041   A4   000001   B4   000201       C(A5)=   0100   0001   1361   0004   6000       C(B5)=   0100   0001   1361   0004   6000
 FE   000000   A5   000111   B5   000111       C(A6)=   0000   0000   0000   0000   0000       C(B6)=
 MA   000400   A6   000001   B6   000200       C(A7)=   0000   0000   0000   0000   0000       C(B7)=
               A7   000001   B7   027756
  X0  0000   0000   0000   0000   0000
  X1  0000   0000   0000   0000   0000
  X2  1505   1520   0000   0000   0061
  X3  0000   0000   0000   0000   0000
  X4  0000   0000   0000   0000   0000
  X5  6000   0000   0004   0040   0000
  X6  1505   1520   0000   0000   0061
  X7  0060   0000   0000   0000   0000

     00000    00000 00000 00000 00000                                          64550 02550 00000 46000          00000 00000 00000 00000
     00054   *56110 03110 00054 54710        51100 00001 03110 00055           07040 00060 51600 00001          04000 00063 00000 00021
     00060    15051 52000 00000 00061        00000 00200 00000 00001           40000 00000 02000 00111          00000 00000 40000 00000
     00064    14071 70000 00000 00000        00000 00000 00000 00115
     00070    14071 75755 00000 00000        00000 00000 00000 00000
     00100   *54000 00000 01000 00001


                                       MFF   NB2-  CY8175-SN1    4LB7/A6B  05/15/78
                                      08.52.04.DON0840  FROM     /SH
                                      08.52.04.IP  00000192 WORDS - FILE INPUT   . DC 04
                                      08.52.04.DON.      PSD.0278.72CT011A.MILLER
                                      08.52.06.COMPASS.
                                      08.52.06. ASSEMBLY COMPLETE.   47300B SCM USED.
                                      08.52.06.      0.062  CPU SECONDS  ASSEMBLY TIME.
                                      08.52.06.LGO.
                                      08.52.07.ERROR MODE =00.  ADDRESS =000000
                                      08.52.07.EXIT.
                                      08.52.07.DMP(111.117)
                                      08.52.07.OP  00000768 WORDS - FILE OUTPUT . DC 40
                                      08.52.07.MS      3584  WORDS  (    10752 MAX USED)
                                      08.52.07.CPA          .100 SEC.              .100 ADJ.
                                      08.52.07.IO           .619 SEC.              .619 ADJ.
                                      08.52.07.CM          9.242 KWS.              .563 ADJ.
                                      08.52.07.SS                                 1.283
                                      08.52.07.PP          2.805 SEC.      DATE 06/27/78
```

BOOLEAN UNIT

BX TRANSMIT A "X" REGISTER TO ANOTHER "X" REGISTER; OR TRANSMIT COMPLEMENT FROM A "X" REGISTER TO ANOTHER "X" REGISTER.

TRANSMIT A "X" REGISTER AND THE COMPLEMENT OF ANOTHER "X" REGISTER.

TRANSMIT	
LOGICAL PRODUCT	(AND)
LOGICAL SUM	(OR)
LOGICAL DIFFERENCE	(EXCLUSIVE OR)

60-BIT ARITHMETIC

BOOLEAN LOGICAL PRODUCT

THE BOOLEAN LOGICAL PRODUCT IS USED MAINLY FOR MASKING PURPOSES. THIS FUNCTION COMPARES TWO X REGISTERS BIT BY BIT. IF THE BITS IN BOTH REGISTERS ARE TRUE (I.E., SET TO 1), THEN THE CORRESPONDING BIT IS SET IN THE RESULT REGISTER.

FOR EXAMPLE, ASSUME THE FOLLOWING BIT PATTERNS:

```
    (X1)=00000 . . . . . . . 01010 11100

    (X2)=00000 . . . . . . . 00000 11111
```

BX3 X1*X2 WOULD PRODUCE:

```
    (X3)=00000 . . . . . . . 00000 11100
```

IN THIS WAY A PROGRAMMER CAN MASK OUT A PORTION OF ONE X REGISTER BY SETTING ZERO BIT IN A SECOND REGISTER AND PERFORMING A BOOLEAN LOGICAL PRODUCT OPERATION INTO A THIRD REGISTER. THE THIRD REGISTER CONTAINS ONLY THE PORTION OF THE FIRST X REGISTER THAT THE PROGRAMMER WANTS TO EXAMINE.

FOR EXAMPLE, ASSUME THAT (X1)=0000 0111 2222 3333 4444. A PROGRAMMER WOULD LIKE TO EXTRACT THE FIRST EIGHT OCTAL DIGITS OF THIS REGISTER AND ZERO-OUT THE REMAINDER OF THE CONTENTS. THIS COULD BE DONE AS FOLLOWS:

```
    MX2   24        (X2)=7777 7777 0000 0000 0000

    BX3   X1*X2     (X3)=0000 0111 0000 0000 0000
```

TO EXTRACT THE CONTENTS OF THE LOWEST SIX OCTAL DIGITS, THE FOLLOWING INSTRUCTIONS COULD BE USED:

```
    SX2   777777B   (X2)=0000 0000 0000 0077 7777

    BX3   X1*X2     (X3)=0000 0000 0000 0033 4444
```

COMPLEMENTS CAN ALSO BE USED TO CLEAR SPECIFIED AREAS OF A WORD:

```
    SX2   770000B   (X2)=0000 0000 0000 0077 0000

    BX3   -X2*X1    (X3)=0000 0111 2222 3300 4444
```

BOOLEAN LOGICAL SUM

THE BOOLEAN LOGICAL SUM IS USED FOR INSERTING DATA INTO A PORTION OF A WORD. THIS FUNCTION COMPARES TWO X REGISTERS BIT BY BIT. IF EITHER REGISTER HAS A TRUE BIT (I.E., A BIT SET TO 1), THE CORRESPONDING BIT IN THE RESULT REGISTER IS SET TO 1.

FOR EXAMPLE, ASSUME THE FOLLOWING BIT PATTERNS:

```
    (X1)=0000 . . . . . . . . . 01110 00001

    (X2)=0000 . . . . . . . . . 00000 11001

BX3   X1+X2 WOULD PRODUCE:

    (X3)=00000  . . . . . . . . 01110 11001
```

A PROGRAMMER CAN USE A BOOLEAN SUM INSTRUCTION TO ENSURE THAT A SPECIFIC BIT IS SET IN AN X REGISTER. TO SET BIT 0, THIS PROCEDURE COULD BE USED:

```
    (X1)=1111 2222 3333 4444 5555

SX2   1

    (X2)=0000 0000 0000 0000 0001

BX3   X2+X1

    (X3)=1111 2222 3333 4444 5555
```

TO SET BIT 59, THIS PROCEDURE COULD BE USED:

```
    (X1)=1111 2222 3333 4444 5555

MX2   1

    (X2)=4000 0000 0000 0000 0000

BX3   X2+X1

    (X3)=5111 2222 3333 4444 5555
```

BOOLEAN LOGICAL DIFFERENCE

THE BOOLEAN LOGICAL DIFFERENCE IS USED FOR COMPARING BIT PATTERNS AND COMPLEMENTING PORTIONS OF X REGISTERS. THIS FUNCTION COMPARES TWO X REGISTERS BIT BY BIT AND SETS A BIT IN THE RESULT REGISTER IF THE BITS IN THE OPERAND REGISTERS ARE DIFFERENT. THUS IT IS A TEST FOR DIFFERENCE.

FOR EXAMPLE, ASSUME THE FOLLOWING BIT PATTERNS:

```
    (X1)=0000 . . . . . . . 01110 00001

    (X2)=0000 . . . . . . . 01110 10000
```

BX3   X1-X2 WOULD PRODUCE:

```
    (X3)=00000  . . . . . . 00000 10001
```

THE BOOLEAN LOGICAL PRODUCT IS USED FOR MASKING.

THE BXI INSTRUCTION COMPARES TWO X REGISTERS, BIT-BY-BIT.

IF BOTH BITS ARE TRUE, SET TO 1, THEN THE CORRESPONDING BIT IN THE RESULT REGISTER IS ALSO SET TO 1.

J *	K →	I
1	1	1
1	0	0
0	1	0
0	0	0

A PROGRAMMER CAN USE THIS TO MASK PORTIONS OF A REGISTER BY SETTING THE CORRESPONDING BITS TO ZERO IN A SECOND REGISTER AND PERFORMING THE LOGICAL PRODUCT OPERATION.

THE BOOLEAN INSTRUCTION IS ALSO USED TO TRANSMIT ENTIRE WORD FROM ONE X REGISTER TO ANOTHER.

BOOLEAN LOGICAL SUM.

INSERT DATA INTO PORTION OF A WORD.

TWO REGISTERS ARE COMPARED ON A BIT-BY-BIT BASIS.

IF EITHER OR BOTH BITS ARE TRUE, SET TO 1, THEN THE CORRESPONDING BIT IN THE RESULT REGISTER IS ALSO SET TO 1.

J	+ K	⟶ I
1	1	1
1	0	1
0	1	1
0	0	0

SPECIFIC BITS CAN BE SET USING THIS METHOD.

BOOLEAN LOGICAL DIFFERENCE.

INSERT DATA INTO A PORTION OF A WORD.

TWO REGISTERS ARE COMPARED ON A BIT-BY-BIT BASIS.

IF BOTH BITS ARE TRUE, SET TO 1 OR IF BOTH BITS ARE FALSE, SET TO 0. THEN THE CORRESPONDING BIT IN THE RESULT REGISTER IS SET TO 0. IF THE BITS BEING COMPARED ARE UNLIKE, THEN THE BIT IN THE RESULT REGISTER IS SET TO 1.

J -	K →	I
1	1	0
1	0	1
0	1	1
0	0	0

BOOLEAN INSTRUCTIONS

HERE IS A LIST OF THE BOOLEAN INSTRUCTIONS:

BX2	X1	BX2	-X1	TRANSFERS
BX3	X2+X1	BX3	-X2*X1	LOGICAL AND
BX3	X2+X1	BX3	-X2+X1	LOGICAL OR
BX3	X2-X1	BX3	-X2-X1	LOGICAL EXCLUSIVE OR
MX4	12			MASK: SET UPPER BITS
CX3	X4			COUNT ONE BITS

TO TRANSMIT THE ENTIRE 60 BITS FROM ONE X REGISTER TO ANOTHER, THE BOOLEAN TRANSFER INSTRUCTION IS USED. FOR EXAMPLE, ASSUME:

```
(X1)  0000 5555 0000 0000 0000
```

THE INSTRUCTION BX2 X1 PRODUCES:

```
(X2)  0000 5555 0000 0000 0000
```

THE INSTRUCTION BX2 -X1 PRODUCES:

```
(X2)  7777 2222 7777 7777 7777
```

SHIFT UNIT

LX	Left Circular
AX	Right End-Off

Shift count may be in the instruction or a "B" register. A negative shift count in a "B" register reverses the direction of the shift. The destination register may be different from the source register.

MX	Form Mask

Floating Point Numbers

NX	Normalize
PX	Pack
UX	Unpack
ZX	Normalize rounded

SHIFT INSTRUCTIONS:

1. LEFT SHIFT, WHICH MOVES BITS LEFTWARD AND END-AROUND (FROM THE TOP OF THE WORD TO THE BOTTOM).

   FOR EXAMPLE, IF THE BIT STRUCTURE OF AN X REGISTER IS:

   11110 00000 00000 00011

   A LEFT SHIFT OF THREE WOULD PRODUCE:

   10000 00000 00000 . . . . 0 11111

2. RIGHT SHIFT, WHICH MOVES BITS RIGHTWARD AND BITS ON THE RIGHT END ARE LOST AS THE SHIFT PROGRESSES. BIT 59 (THE SIGN BIT) IS EXTENDED.

   FOR EXAMPLE, IF THE BIT STRUCTURE OF AN X REGISTER IS:

   11110 00000 00101

   A RIGHT SHIFT OF THREE WOULD PRODUCE:

   11111 11000 00000

THERE ARE EIGHT COMPASS INSTRUCTIONS WHICH PRODUCE LEFT AND RIGHT SHIFTS. THE FOLLOWING EXAMPLES SHOW THE EFFECT OF EACH INSTRUCTION.

ASSUME THAT BEFORE EACH OPERATION:

    (X1)=70700 00000 00000 00007

LX1 6 PRODUCES A LEFT SHIFT OF SIX BITS (TWO OCTAL DIGITS).

    (X1) NOW=70000 00000 00000 00770

LX1 -6 PRODUCES A LEFT SHIFT OF 54 (60-6) BITS; THIS LOOKS LIKE A RIGHT END-AROUND SHIFT OF SIX BITS.

    (X1) WOULD BE 07707 00000 00000 00000

AX1 6 PRODUCES A RIGHT SHIFT OF SIX BITS.

    (X1)=77707 00000 00000 00000

AX1 -6 PRODUCES A RIGHT SHIFT OF 54 (60-6) BITS.

    (X1)=77777 77777 77777 77770

LX2 X1,B1 (B1)=6 PRODUCES A LEFT SHIFT OF SIX BITS.

    (X2)=70000 00000 00000 00770

LX2 X1,B1 (B1)=-6 PRODUCES A RIGHT END OFF SHIFT OF SIX BITS.

    (X2)=77707 00000 00000 00000

AX2 X1,B1 (B1)=6 PRODUCES A RIGHT END OFF SHIFT OF SIX BITS.

    (X2)=77707 00000 00000 00000

AX2 X1,B1 (B1)=-6 PRODUCES A LEFT SHIFT OF SIX BITS.

    (X2) 70000 00000 00000 00770

```
                                            IDENT     SHIFT
                                             ENTRY    SHIFT
 0   00000 000000000000001        NUMBER    DATA      1,2,-0,10
 4         \                      SHIFT     BSS       0
                                *
                                *   LEFT SHIFT  (LXI  JK PLACES)
                                *
 4   51100 00000 +                          SA1       NUMBER
                20103                       LX1       3
                     20105                  LX1       5
 5   51100 00000 +                          SA1       NUMBER
                20173                       LX1       59
 6   51100 00000 +                          SA1       NUMBER
                20174                       LX1       60
                                *
                                *   ARITHMETIC RIGHT SHIFT (AXI  JK PLACES)
                                *
 7   51200 00001 +                          SA2       NUMBER+1
                21201                       AX2       1
10   51200 00023 +                          SA2       =37777777777777777777B
                21272                       AX2       58
                                *
                                *   LEFT SHIFT XK NOMINALLY BJ PLACES
                                *
11   61100 00003                            SB1       3
                5130000000 +                SA3       NUMBER
12   22313                                  LX3       B1,X3
          6110777774                *       SB1       777774B
13   51300 00003 +                          SA3       NUMBER+3
                22313                       LX3       B1,X3
                                *
                                *   ARITHMETIC RIGHT XHIFT XK NOMINALLY BJ PLACES
                                *
14   61100 00003                            SB1       3
                5140000003 +                SA4       NUMBER+3
15   23414                                  AX4       B1,X4
          6110777774                        SB1       777774B
16   51400 00000 +                          SA4       NUMBER
                23414                       AX4       B1,X4
17   61100 00073                            SB1       59
                5140000002 +                SA4       NUMBER+2
20   23414                                  AX4       B1,X4
          6110001100                        SB1       1100B
21   51400 00003 +                          SA4       NUMBER+3
                23414                       AX4       B1,X4
22   01300 01040000000000                   ENDRUN
24                                          END       SHIFT

          27700B SCM STORAGE USED                48 STATEMENTS          2 SYMBOL
                     MODEL 76 ASSEMBLY         0.025 SECONDS           13 REFERE
```

EXAMPLES OF SHIFTS

```
LX3     6        Shift X3 left 6 bits
AX3     6        Shift X3 right 6 bits
```

(B1) = +3   (B2) = -3

```
LX3     X1,B1    Left, 3 bits
AX3     X1,B1    Right, 3 bits
LX3     X1,B2    Right, 3 bits
AX3     X1,B2    Left, 3 bits
```

Into   Out of   Shift count

SHIFT INSTRUCTIONS - WITHOUT B REGISTERS

```
                                                IDENT   SHIFT
                                                ENTRY   SHIFT
0  40000000000000000000          0              DATA    40000000000000000000B
                                 .
                                 .
1  5110000000 +                  SHIFT          SA1     0
               10011                            BX0     X1
                      10211                     BX2     X1
2  10311                                        BX3     X1
        10411                                   BX4     X1
               10511                            BX5     X1
                      10611                     BX6     X1
3  10711                                        BX7     X1
                                 .
        21100                                   AX1     0               .RIGHT SHIFT 0 IE. NO CHANGE
               20274                            LX2     60              .LEFT SHIFT 60 BITS IE. NO CHANGE
                      21312                     AX3     10
4  20412                                        LX4     10              .LEFT SHIFT 10 BITS
        21562                                   AX5     -10             .RIGHT SHIFT (60-10) BITS
               20662                            LX6     -10             .LEFT SHIFT (60-10) BITS END AROUND
5  0200000000                                   JP      0
6                                               END     SHIFT


                                 X0   4000  0000  0000  0000  0000
                                 X1   4000  0000  0000  0000  0000
                                 X2   4000  0000  0000  0000  0000
                                 X3   7776  0000  0000  0000  0000
                                 X4   6000  0000  0000  0000  1000
                                 X5   7777  7777  7777  7777  7000
                                 X6   0002  0000  0000  0000  0000
                                 X7   4000  0000  0000  0000  0000
```

```
 SHIFTB                                                                  COMPASS 3.5-470.     06/27/78  16.18.08.       PAGE      2

                                                    IDENT   SHIFTB
                                                    ENTRY   SHIFTB
                                   6 +    CONSTANT LIT      40000000000000000000B
      0  5110000006 +                     SHIFTB    SA1     CONSTANT
                    10011                           BX0     X1
      1  6110000006                                 SB1     6
                    6120777771                      SB2     -6
      2  6130777700                                 SB3     -63            -63 DECIMAL
                    6140777677                      SB4     -64            -64 DECIMINAL
      3  6150777676                                 SB5     -65            -65 DECIMAL
                    22210                           LX2     B1,X0          SHIFT LEFT 6 BITS
                         22320                      LX3     B2,X0          SHIFT RIGHT 6 BITS
      4  22430                                      LX4     B3,X0          RIGHT SHIFT 63 PLACES
              22540                                 LX5     B4,X0          CLEAR DESTINATION REGISTER
                   22650                            LX6     B5,X0
      5  0200000000                                 JP      0
      7                                             END     SHIFTB

                         47300B SCM STORAGE USED               17 STATEMENTS          2 SYMBOLS
                                 MODEL 174 ASSEMBLY      0.018 SECONDS          4 REFERENCES
```

```
 DMPX.

  P    000000   A0   000200   B0   000000
  RA   220200   A1   000117   B1   000006      C(A1)=  4000  0000  0000  0000  0000     C(B1)=  0000  0000  0000  0000  0000
  FL   000200   A2   000060   B2   777771      C(A2)=  1505  1520  0000  0000  0061     C(B2)=
  EM   700700   A3   000057   B3   777700      C(A3)=  0000  0000  0000  0000  0000     C(B3)=
  RE   000045   A4   000001   B4   777677      C(A4)=  0000  0000  0000  0000  0000     C(B4)=
  FE   000000   A5   000111   B5   777676      C(A5)=  5110  0001  1710  0114  6000     C(B5)=
  MA   000600   A6   000001   B6   000200      C(A6)=  0000  0000  0000  0000  0000     C(B6)=
                A7   000001   B7   027756      C(A7)=  0000  0000  0000  0000  0000     C(B7)=
  X0   4000   0000   0000   0000   0000
  X1   4000   0000   0000   0000   0000
  X2   0000   0000   0000   0000   0040
  X3   7740   0000   0000   0000   0000
  X4   7777   7777   7777   7777   7777
  X5   0000   0000   0000   0000   0000
  X6   0000   0000   0000   0000   0000
  X7   0000   0000   0000   0000   0000

     00000    00000 00000 00000 00000
     00054   -56110 03110 00054 54710         51100 00001 03110 00055          64550 02550 00000 46000          00000 00000 00000 00000
     00060    15051 52000 00000 00061         00000 00200 00000 00001          07040 00060 51600 00001          04000 00063 00000 00021
     00064    14071 70000 00000 00000         00000 00000 00000 00120          40000 00000 02000 00111          00000 00000 40000 00000
     00070    14071 75755 00000 00000         00000 00000 00000 00000
     00100   -54000 00000 01000 00001
```

```
SHIFTB                                                    COMPASS 3.5-470.     06/27/78  16.18.08.      PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH              BINARY CONTROL CARDS.

                0        7              IDENT   SHIFTB
                7                       END     SHIFTB

                               BLOCKS     TYPE       ADDRESS    LENGTH

                               PROGRAM*  LOCAL            0          6
                               LITERALS* LOCAL            6          1

                               ENTRY POINTS.

                               SHIFTB            0*
                                                  CYBER LOADER 1.4-470      06/27/78  16.18.08.      PAGE      1
 LOAD MAP - SHIFTB


 FWA OF THE LOAD              111
 LWA+1 OF THE LOAD            120

 TRANSFER ADDRESS -- SHIFTB           111

 PROGRAM ENTRY POINTS --        SHIFTB             111


 PROGRAM AND BLOCK ASSIGNMENTS.

 BLOCK        ADDRESS   LENGTH     FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

 SHIFTB          111        7      LGO       06/27/78 COMPASS 3.5 470


      .008 CP SECONDS             13200B CM STORAGE USED                      1 TABLE MOVE

                                                              MFF   N02-  CYB175-SN1    4L87/R68  05/15/78
                                                             16.18.06.DON00MR  FROM    /SM
                                                             16.18.06.IP  00000320 WORDS - FILE INPUT  , DC 04
                                                             16.18.06.DON.     PSD,0278,72CT011A,MILLER
                                                             16.18.07.COMPASS.
                                                             16.18.08. ASSEMBLY COMPLETE.   47300B SCM USED.
                                                             16.18.08.    0.063  CPU SECONDS  ASSEMBLY TIME.
                                                             16.18.08.LGO.
                                                             16.18.08.ERROR MODE =00.  ADDRESS =000000
                                                             16.18.09.OP  00000768 WORDS - FILE OUTPUT  , DC 40
                                                             16.18.09.MS     3584  WORDS  (   10752 MAX USED)
                                                             16.18.09.CPA         .096 SEC.          .096 ADJ.
                                                             16.18.09.IO          .613 SEC.          .613 ADJ.
                                                             16.18.09.CM         9.261 KWS.          .565 ADJ.
                                                             16.18.09.SS                            1.275
                                                             16.18.09.PP         2.911 SEC.     DATE 06/27/78
                                                             16.18.09.EJ  END OF JOB, SM
```

```
TSTSHFT                                                          COMPASS 3.5-410.      06/27/78  09.00.54.       PAGE      1
STORAGE ALLOCATION.

            ADDRESS   LENGTH              BINARY CONTROL CARDS.

                  0       32              IDENT   TSTSHFT
                 32                       END     TSTSHFT

                                 ENTRY POINTS.

                                 TSTSHFT         14*

                                 EXTERNAL SYMBOLS.

                                 SYS=
TSTSHFT                                                                        COMPASS 3.5-470.     06/27/78  09.00.54.      PAGE      2

                                                       IDENT   TSTSHFT
                                                       ENTRY   TSTSHFT
       0                       14        ANSWER        BSSZ    12
      14  43001                          TSTSHFT       MX0     1
               10600                                   BX6     X0
                    10700                              BX7     X0
 7                       20644                         LX6     100           LEAST SIGNIFICANT 6 BITS USED 44(OCTAL)=36(DECIMAL)
 7    15  21744                                        AX7     100           LEAST SIGNIFICANT 6 BITS USED 44(OCTAL)=36(DECIMAL)
               5160000000 +                            SA6     ANSWER+0
      16  5170000001 +                                 SA7     ANSWER+1
                    10600                              BX6     X0
                         10700                         BX7     X0
 7    17  20633                                        LX6     -100          LEAST SIGNIFICANT 6 BITS USED 33(OCTAL)=27(DECIMAL)
 7             21733                                   AX7     -100          LEAST SIGNIFICANT 6 BITS USED 33(OCTAL)=27(DECIMAL)
                    5160000002 +                       SA6     ANSWER+2
      20  5170000003 +                                 SA7     ANSWER+3
                    10600                              BX6     X0
                         10700                         BX7     X0
      21  20664                                        LX6     20+32         52(DECIMAL) = 64(OCTAL)
               21764                                   AX7     20+32         52(DECIMAL) = 64(OCTAL)
                    5160000004 +                       SA6     ANSWER+4
      22  5170000005 +                                 SA7     ANSWER+5
                    10600                              BX6     X0
                         10700                         BX7     X0
      23  20614                                        LX6     32-20         12(DECIMAL) = 14(OCTAL)
               21714                                   AX7     32-20         12(DECIMAL) = 14(OCTAL)
                    5160000006 +                       SA6     ANSWER+6
      24  5170000007 +                                 SA7     ANSWER+7
                    10600                              BX6     X0
                         10700                         BX7     X0
      25  20660                                        LX6     20-32         -12 = (60-12) = 48(DECIMAL) = 60(OCTAL)
               21760                                   AX7     20-32         -12 = (60-12) = 48(DECIMAL) = 60(OCTAL)
                    5160000010 +                       SA6     ANSWER+8
      26  5170000011 +                                 SA7     ANSWER+9
                    10600                              BX6     X0
                         10700                         BX7     X0
      27  20644                                        LX6     160-184       -24 = (60-24) = 36(DECIMAL) = 44(OCTAL)
               21744                                   AX7     160-184       -24 = (60-24) = 36(DECIMAL) = 44(OCTAL)
                    5160000012 +                       SA6     ANSWER+10
      30  5170000013 +                                 SA7     ANSWER+11
                    7160247021                         ENDRUN
      32                                               END     TSTSHFT

                         47300B SCM STORAGE USED               45 STATEMENTS          3 SYMBOLS
                                    MODEL 174 ASSEMBLY      0.055 SECONDS            16 REFERENCES



TSTSHFT                                                                        COMPASS 3.5-470.     06/27/78  09.00.54.      PAGE      3
ERROR DIRECTORY.

     7 TYPE ERROR                       ADDRESS VALUE EXCEEDS FIELD SIZE, RESULT TRUNCATED
          OCCURRED ON PAGES            2
```

```
LOAD MAP - TSTSHFT                                        CYBER LOADER 1.4-470      06/27/78  09.00.54.      PAGE     1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD          203

TRANSFER ADDRESS -- TSTSHFT          125

PROGRAM ENTRY POINTS --      TSTSHFT          125


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK          ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

TSTSHFT           111       32    LGO       06/27/78 COMPASS 3.5 470
SYS.RM            143       40 SL-SYSLIB    05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.


      .025 CP SECONDS              13200B CM STORAGE USED                     1 TABLE MOVE




DUMP     RELATIVE                           DMP(111,147)

   00111   00000 00040 00000 00000       77777 77777 77400 00000       00000 00000 04000 00000
   00114   77777 77774 00000 00000       00100 00000 00000 00000       77777 77777 77777 77600       00000 00000 00000 04000

   00120   77774 00000 00000 00000       00004 00000 00000 00000       77777 77777 77777 74000       00000 00040 00000 00000

   00124   77777 77777 77400 00000       43001 10600 10700 20644       21744 51600 00111 46000       51700 00112 10600 10700

   00130   20633 21733 51600 00113       51700 00114 10600 10700       20664 21764 51600 00115       51700 00116 10600 10700

   00134   20614 21714 51600 00117       51700 00120 10600 10700       20660 21760 51600 00121       51700 00122 10600 10700

   00140   20644 21744 51600 00123       51700 00124 71602 47021       20650 01000 00145 46000       04000 00156 00000 00000

   00144   01300 00000 00000 00000       04000 00143 00000 00000       51100 00001 03110 00146       54610 04000 00144 46000
```

```
                                                                 COMPASS 3.5-470.     06/27/78  09.00.54.       PAGE      4
TSTSHFT
SYMBOLIC REFERENCE TABLE.

ANSWER          0      PROGRAM*         2/03 L     2/10 S     2/16 S     2/22 S     2/28 S     2/34 S     2/40 S
                                        2/09 S     2/15 S     2/21 S     2/27 S     2/33 S     2/39 S
SYS=            0      EXTERNAL*        2/42
TSTSHFT        14      PROGRAM*         2/02 E     2/04 L
```

```
  MFF   NB2-  CYB175-SN1    4LB7/H6B  05/15/78
09.00.52.DON005K  FROM    /SH
09.00.52.IP  00000320 WORDS - FILE INPUT  , DC 04
09.00.52.DON.     PSD,0278,72CT011A,MILLER
09.00.53.COMPASS.
09.00.54.          4 WARNING MESSAGES IN TSTSHFT
09.00.54. ASSEMBLY COMPLETE.   47300B SCM USED.
09.00.54.     0.108  CPU SECONDS  ASSEMBLY TIME.
09.00.54.LGO.
09.00.54.DMP(111,147)
09.00.54.OP  00000960 WORDS - FILE OUTPUT  , DC 40
09.00.54.MS    3584  WORDS  (   10752 MAX USED)
09.00.54.CPA        .162 SEC.          .162 ADJ.
09.00.54.IO         .634 SEC.          .634 ADJ.
09.00.54.CM       10.871 KWS.          .663 ADJ.
09.00.54.SS                           1.460
09.00.54.PP        2.737 SEC.     DATE 06/27/78
09.00.54.EJ  END OF JOB, SH
```

```
PROGRAM WRITEIT    74/74   OPT=1                           FTN 4.7+485        01/08/79  10.27.43        PAGE     1

                    PROGRAM WRITEIT(OUTPUT)
                    COMMON /BLOCK/ ANSWER(25)
                    CALL INSTRS
                    PRINT 101
                    DO 10 I=1,25
                    J=I-1
                 10 PRINT 100,J,ANSWER(I)
                100 FORMAT(* ANS+*I2,O23)
                101 FORMAT(1H1)
                    END
```

```
INSTRS
STORAGE ALLOCATION.                                         COMPASS 3.6-476.    01/08/79  10.29.06.      PAGE     1

          ADDRESS   LENGTH               BINARY CONTROL CARDS.

                0       47               IDENT  INSTRS
               47                        END

                               BLOCKS     TYPE       ADDRESS    LENGTH

                               PROGRAM*   LOCAL            0        47
                               BLOCK      COMMON           0        31

                               ENTRY POINTS.

                               INSTRS          3+
                                                       IDENT   INSTRS
                                                       ENTRY   INSTRS
                                                       LIST    -B,-R
                                             *
                                             *
                                                       COMMENT  SET - SHIFT - MASK - BOOLEAN - INSTRUCTION PROBLEMS
                                             *
                                             *
                                                       LIST    S
     0  00000000000000004444         BUF      DATA    4444B,5555B,6666B
     1  00000000000000005555
     2  00000000000000006666
                                             *
                                             *         EXAMPLE OF SET/BOOLEAN/MASK/SHIFT
                                             *
     3  00000000000000000000         INSTRS   DATA    0
     4  6110033333                            SB1     033333B
                  6120355555                  SB2     355555B
     5  76612                                 SX6     B1+B2
              5160000000 C                    SA6     ANS

     6  6140777774                   NUMBER1  SB4     777774B
                  76740                       SX7     B4
     7  5170000001 C                          SA7     ANS+1

    10  5110000002 +                 NUMBER2  SA1     BUF+2
                  36711                       IX7     X1+X1
    11  5170000002 C                          SA7     ANS+2

    12  47611                        NUMBER3  CX6     X1
              5160000003 C                    SA6     ANS+3

    13  5110000000 +                 NUMBER4  SA1     BUF
                  74710                       SX7     A1+B0
    14  5170000004 C                          SA7     ANS+4

    15  5130000000 +                 NUMBER5  SA3     BUF
                  5140000001 +                SA4     BUF+1
    16  11634                                 BX6     X3*X4
              5160000005 C                    SA6     ANS+5

    17  12734                        NUMBER6  BX7     X3+X4
              5170000006 C                    SA7     ANS+6

    20  13634                        NUMBER7  BX6     X3-X4
              5160000007 C                    SA6     ANS+7

    21  14633                        NUMBER8  BX6     -X3
              5160000010 C                    SA6     ANS+8

    22  20314                        NUMBER9  LX3     12
              10733                           BX7     X3
                  5170000011 C                SA7     ANS+9
```

```
23  43606                    NUMBER10 MX6    6
         5160000012 C                 SA6    ANS+10

24  5110000000 +             NUMBER11 SA1    BUF
              21103                   AX1    3
                   10611              BX6    X1
25  5160000013 C                      SA6    ANS+11

26  21101                    NUMBER12 AX1    1
         10711                        BX7    X1
              5170000014 C            SA7    ANS+12

27  7110000463               NUMBER13 SX1    463B
              6120000006              SB2    6
30  22621                             LX6    X1,B2
         5160000015 C                 SA6    ANS+13

31  6110777774               NUMBER14 SB1    -3
              22711                   LX7    X1,B1
32  5170000016 C                      SA7    ANS+14

33  23621                    NUMBER15 AX6    X1,B2
         5160000017 C                 SA6    ANS+15

34  23711                    NUMBER16 AX7    X1,B1
         5170000020 C                 SA7    ANS+16

35  6110000106               NUMBER17 SB1    106B
              7110000007              SX1    7
36  22711                             LX7    X1,B1
         5170000021 C                 SA7    ANS+17
37  23711                    NUMBER18 AX7    X1,B1
         5170000022 C                 SA7    ANS+18

40  43766                    NUMBER19 MX7    -6
         5170000023 C                 SA7    ANS+19

                             *
                             *        EXAMPLE OF INCORRECT USAGE
                             *
41  5150000000 +             NUMBER20 SA5    BUF
              20560                   LX5    -12
                   10755              BX7    X5
42  5170000024 C                      SA7    ANS+20

43  5150000000 +             NUMBER21 SA5    BUF
              21566                   AX5    -6
                   10655              BX6    X5
44  5160000025 C                      SA6    ANS+21

45  5140000000 +             NUMBER22 SA4    BUF
              20442                   LX4    162
                   10644              BX6    X4
46  5160000026 C                      SA6    ANS+22
```

```
INSTRS                                                                COMPASS 3.6-476.     01/08/79  10.29.06.      PAGE     4

                      0400000003 +           EQ     INSTRS
                                             USE    /BLOCK/
        0                       31   ANS     BSSZ   25
       47                                    END
        .
                        475008 CM   STORAGE USED           111 STATEMENTS        25 SYMBOLS
                                    MODEL 74 ASSEMBLY    0.330 SECONDS           58 REFERENCES
```

```
INSTRS
ERROR DIRECTORY.                                                      COMPASS 3.6-476.     01/08/79  10.29.06.      PAGE     5

     7 TYPE ERROR
            OCCURRED ON PAGES        ADDRESS VALUE EXCEEDS FIELD SIZE, RESULT TRUNCATED
                                         3
```

```
INSTRS
SYMBOLIC REFERENCE TABLE.                                             COMPASS 3.6-476.     01/08/79  10.29.06.      PAGE     6

ANS            0    BLOCK
                                 2/20 S    2/31 S    2/43 S    2/53 S    3/11 S    3/23 S    3/33 S    3/49 S
                                 2/24 S    2/35 S    2/46 S    3/02 S    3/16 S    3/26 S    3/36 S    3/54 S
BUF            0    PROGRAM*     2/28 S    2/40 S    2/49 S    3/07 S    3/20 S    3/31 S    3/44 S    4/04 L
                                 2/10 L    2/33      2/38      3/41      3/51
INSTRS         3    PROGRAM*     2/26      2/37      3/04      3/46
NUMBER1        6    PROGRAM*     2/02 E    2/16 L    4/02
NUMBER10      23    PROGRAM*     2/22 L
NUMBER11      24    PROGRAM*     3/01 L
NUMBER12      26    PROGRAM*     3/04 L
NUMBER13      27    PROGRAM*     3/09 L
NUMBER14      31    PROGRAM*     3/13 L
NUMBER15      33    PROGRAM*     3/18 L
NUMBER16      34    PROGRAM*     3/22 L
NUMBER17      35    PROGRAM*     3/25 L
NUMBER18      37    PROGRAM*     3/28 L
NUMBER19      40    PROGRAM*     3/32 L
NUMBER2       10    PROGRAM*     3/35 L
NUMBER20      41    PROGRAM*     2/26 L
NUMBER21      43    PROGRAM*     3/41 L
NUMBER22      45    PROGRAM*     3/46 L
NUMBER3       12    PROGRAM*     3/51 L
NUMBER4       13    PROGRAM*     2/30 L
NUMBER5       15    PROGRAM*     2/33 L
NUMBER6       17    PROGRAM*     2/37 L
NUMBER7       20    PROGRAM*     2/42 L
NUMBER8       21    PROGRAM*     2/45 L
NUMBER9       22    PROGRAM*     2/48 L
                                 2/51 L
```

```
LOAD MAP - WRITEIT                                 CYBER LOADER 1.4-485      01/08/79  10.29.20.      PAGE     1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD         7150

TRANSFER ADDRESS -- WRITEIT       2224

PROGRAM ENTRY POINTS --     WRITEIT        2224


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK      ADDRESS   LENGTH    FILE     DATE     PROCSSR VER LEVEL HARDWARE  COMMENTS

/BLOCK/         111      31
WRITEIT         142    2120   LGO      01/08/79 FTN     4.7 485   666X I    PROGRAM   OPT=1
INSTRS         2262      47   LGO      01/08/79 COMPASS 3.6 476             SET - SHIFT - MASK - BOOLEAN - INSTRUCTION PROB
/STP.END/      2331       1
/FCL.C./       2332      26
/Q8.IO./       2360     101
Q2NTRY=        2461       0 SL-FORTRAN 11/16/78 COMPASS 3.6 485             FCL INITIALIZATION ROUTINE.
/FCL=ENT/      2461      40
COMIO=         2521      33 SL-FORTRAN 11/16/78 COMPASS 3.6 485             COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL=FDL        2554      40 SL-FORTRAN 11/16/78 COMPASS 3.6 485             FCL CAPSULE LOADING
FEIFST=        2614       3 SL-FORTRAN 11/16/78 COMPASS 3.6 485              CONVERTED DATA STORAGE
FLTOUT=        2617     311 SL-FORTRAN 11/16/78 COMPASS 3.6 485             COMMON FLOATING OUTPUT CODE
FORSYS=        3130     301 SL-FORTRAN 11/16/78 COMPASS 3.6 485             FORTRAN OBJECT LIBRARY UTILITIES.
OUTCOM=        3431     154 SL-FORTRAN 11/16/78 COMPASS 3.6 485             COMMON OUTPUT CODE
SYSAID=        3605       1 SL-FORTRAN 11/16/78 COMPASS 3.6 485             LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
FECMSK=        3606      41 SL-FORTRAN 11/16/78 COMPASS 3.6 485             INITIALIZE CONSTANTS.
FMTAP=         3647     357 SL-FORTRAN 11/16/78 COMPASS 3.6 485             CRACK APLIST AND FORMAT FOR KODER/KRAKER.
FORUTL=        4226      46 SL-FORTRAN 11/16/78 COMPASS 3.6 485             FCL MISC. UTILITIES.
GETFIT=        4274      57 SL-FORTRAN 11/16/78 COMPASS 3.6 485             LOCATE AN FIT GIVEN A FILE NAME.
KODER=         4353     451 SL-FORTRAN 11/16/78 COMPASS 3.6 485             OUTPUT FORMAT INTERPRETER.
OUTC=          5024     150 SL-FORTRAN 11/16/78 COMPASS 3.6 485             FORMATTED WRITE FORTRAN RECORD.
/FDL.COM/      5174      14
FDL.RES        5210     211 SL-SYSLIB  11/02/78 COMPASS 3.6 485             FAST DYNAMIC LOADER RESIDENT.
FDL.MMI        5421     222 SL-SYSLIB  11/02/78 COMPASS 3.6 485             FDL MEMORY MANAGER INTERFACE.
CPU.SYS        5643      40 SL-SYSLIB  11/15/78 COMPASS 3.6 476             PROCESS SYSTEM REQUEST.
CMF.ALF        5703     160 SL-SYSLIB  11/15/78 COMPASS 3.6 485             CMM V1.1 - ALLOCATE FIXED.
CMF.CSF        6063       6 SL-SYSLIB  11/15/78 COMPASS 3.6 485             CMM V1.1 - CHANGE SPECS FIXED.
CMM.FFA        6071      14 SL-SYSLIB  11/15/78 COMPASS 3.6 485             CMM V1.1 - FIXED FREE ALGORITHM.
CMF.FRF        6105      36 SL-SYSLIB  11/15/78 COMPASS 3.6 485             CMM V1.1 - FREE FIXED.
CMM.R          6143     214 SL-SYSLIB  11/15/78 COMPASS 3.6 485             CMM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF        6357      22 SL-SYSLIB  11/15/78 COMPASS 3.6 485             CMM V1.1 - SHRINK AT LWA FIXED.
CTL$RM         6401     433 SL-SYSLIB  11/16/78 COMPASS 3.6 485             CRM CONTROLLING ROUTINE.
ERR$RM         7034      25 SL-SYSLIB  11/16/78 COMPASS 3.6 485             CRM ERROR PROCESSOR ENTRY.
LIST$RM        7061      67 SL-SYSLIB  11/16/78 COMPASS 3.6 485             CRM - ALLOCATE SPACE FOR LIST OF FILES


     .335 CP SECONDS                23600B CM STORAGE USED                   12 TABLE MOVES
```

```
ANS+ 0   77777777777777411110
ANS+ 1   77777777777777777774
ANS+ 2   00000000000000015554
ANS+ 3   00000000000000000010
ANS+ 4   00000000000000002262
ANS+ 5   00000000000000004444
ANS+ 6   00000000000000005555
ANS+ 7   00000000000000001111
ANS+ 8   77777777777777773333
ANS+ 9   00000000000044440000
ANS+10   77000000000000000000
ANS+11   00000000000000000444
ANS+12   00000000000000000222
ANS+13   00000000000000046300
ANS+14   00000000000000000046
ANS+15   00000000000000000004
ANS+16   00000000000000004630
ANS+17   00000000000000000700
ANS+18   00000000000000000000
ANS+19   77777777777777777700
ANS+20   44440000000000000000
ANS+21   00000000000000000000
ANS+22   00001111000000000000
ANS+23   00000000000000000000
ANS+24   00000000000000000000
```

```
 MFS  NH1-  CYB74-SN108    5C/R08    11/14/78
10.27.37.DON00K4  FROM    /OH
10.27.37.IP  00000448 WORDS - FILE INPUT  , DC 04
10.27.37.DON,T5.   001A,6883,1896,MILLER
10.27.43.REWIND,OUTPUT.
10.27.43.FTN,R=0.
10.29.07.         1 WARNING MESSAGE IN  INSTRS
10.29.07.       .616 CP SECONDS COMPILATION TIME
10.29.07.MAP,PART.
10.29.07.LGO.
10.29.24.      END  WRITEIT
10.29.24.       .090 CP SECONDS EXECUTION TIME
10.29.24.OP  00001856 WORDS - FILE OUTPUT , DC 40
10.29.24.MS    3584  WORDS  (   17920 MAX USED)
10.29.24.CPA        .957 SEC.          .957 ADJ.
10.29.24.CPB        .220 SEC.          .220 ADJ.
10.29.24.IO        1.170 SEC.         1.170 ADJ.
10.29.24.CM       40.762 KWS.         2.487 ADJ.
10.29.24.SS                           4.836
10.29.24.PP        8.565 SEC.     DATE 01/08/79
10.29.24.EJ  END OF JOB, OH
```

LESSON GUIDE 5
CYBER CP COMPASS
FLOATING POINT ARITHMETIC

LESSON PREVIEW:

THIS LESSON EXAMINES THE FLOATING POINT NUMBER FORMAT AND HARDWARE INSTRUCTIONS THAT MANIPULATE THEM. SINGLE AND DOUBLE PRECISION ARITHMETIC AND THE INSTRUCTIONS TO CONVERT BETWEEN INTEGER AND FLOATING POINT NUMBERS ARE ALSO INCLUDED.

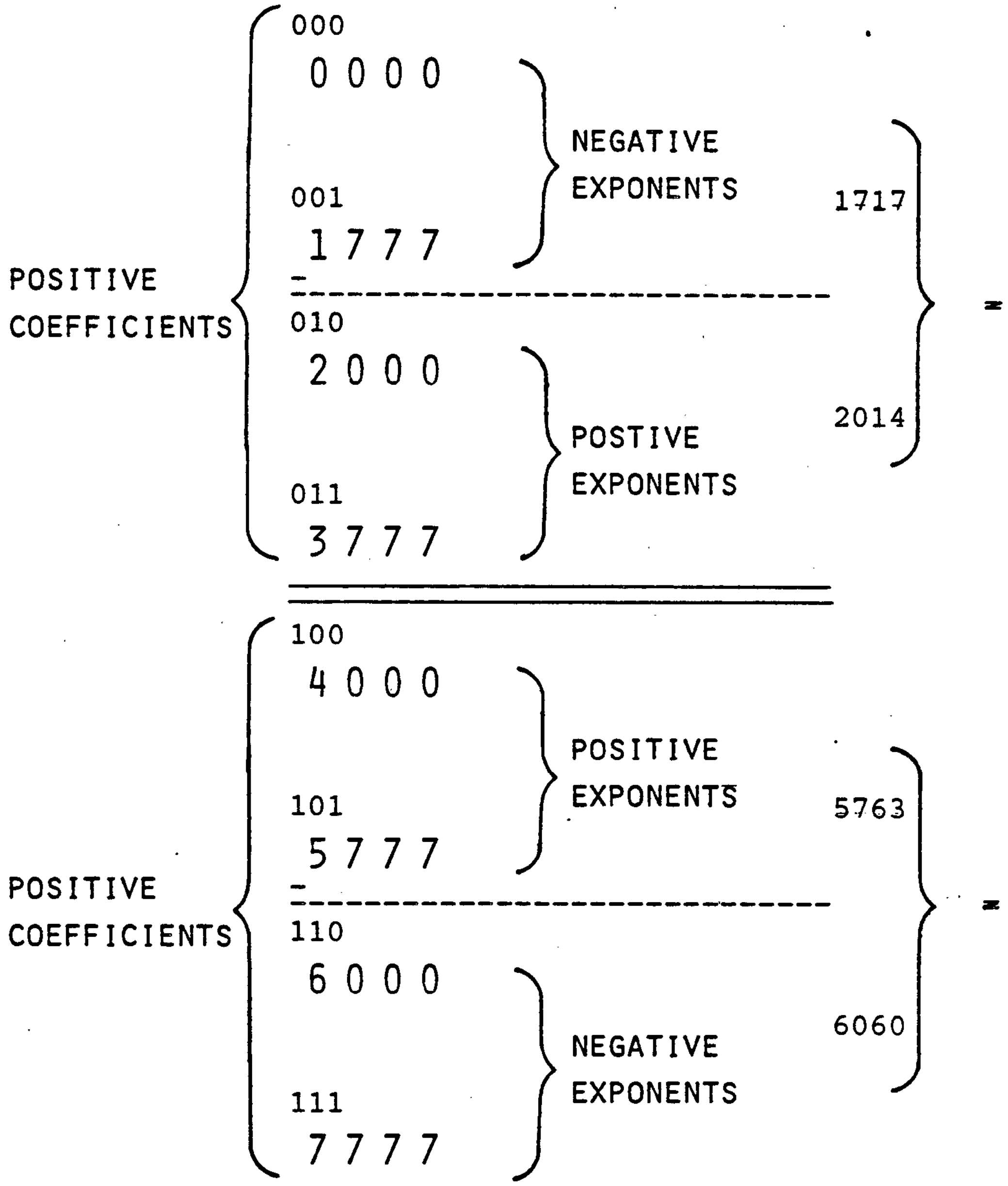
OBJECTIVES:

1. PERFORM ARITHMETIC OPERATION USING THE FXI AND RXI INSTRUCTIONS.
2. RECOGNIZE SPECIAL EXPONENTS AND THEIR EFFECT ON THE PROGRAM.
3. BE AWARE OF DOUBLE PRECISION NUMBERS AND DOUBLE PRECISION OPERATIONS.
4. WRITE PROGRAMS USING FLOATING POINT NUMBERS.
5. BE ABLE TO CONVERT INTEGERS TO AND FROM FLOATING POINT BOTH MANUALLY AND USING PACK AND NORMALIZE INSTRUCTIONS.

LEARNING TASKS:

1. ATTEND THE LECTURE.
2. ANSWER QUESTIONS (EXERCISES 1, 2, AND 3) THROUGHOUT THE LESSON.
3. WRITE AND DEBUG PROGRAM NO. 10 OR 11 FROM STUDENT PROBLEMS.
4. READ COMPASS STUDENT HANDOUT LESSON GUIDE 6 AND COMPASS REFERENCE MANUAL, CHAPTER 8, PAGES 34 THROUGH 44.

RANGE OF EXPONENTS



* NUMBERS WITH COMPONENTS WITHIN THESE RANGES MAY BE CONVERTED TO INTEGERS WITHOUT LOSING THEIR VALUES.

FLOATING POINT ARITHMETIC
BASIC INSTRUCTION SET

THE FOLLOWING INSTRUCTIONS ARE RELEVANT TO WORKING WITH FLOATING POINT NUMBERS:

PACK	PX2	X1,B1
NORMALIZE	NX3	X2,B2
ROUNDED NORMALIZE	ZX3	X2,B2
UNPACK	UX4	X3,B3
LEFT SHIFT	LX5	X4,B3
UNROUNDED ARITHMETIC	FX3	X1+X2
	FX3	X1-X2
	FX3	X1*X2
	FX3	X1/X2
ROUNDED ARITHMETIC	RX3	X1+X2
	RX3	X1-X2
	RX3	X1*X2
	RX3	X1/X2
DOUBLE PRECISION ARITHMETIC	DX3	X1+X2
	DX3	X1-X2
	DX3	X1*X2

DATA FORMAT

PACKING IS THE PROCESS BY WHICH AN INTEGER NUMBER IS CONVERTED TO A FLOATING POINT NUMBER (I.E., A NUMBER WITH A COEFFICIENT AND EXPONENT).

ASSUME THAT A PROGRAMMER WOULD LIKE TO CONVERT THE INTEGER NUMBER 1 TO THE FLOATING POINT 1.0 OR $1*2^{0}$. WE NEED THE FOLLOWING INSTRUCTIONS:

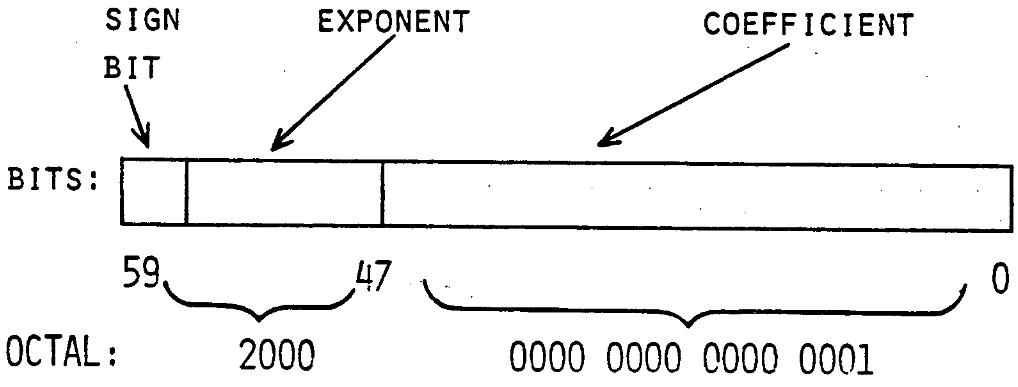
```
                                      (COEFFICIENT)
SX1    1        (X1)=0000 0000 0000 0000 0001

SB1    0        (B1)=000000     (EXPONENT)

PX2    X1,B1    (X2)=2000 0000 0000 0000 000 1
```

X2 CONTAINS THREE LOGICAL UNITS:

A SIGN BIT: BIT 59

AN EXPONENT, BIASED BY 2000: BITS 58 TO 48

A COEFFICIENT: BITS 47 TO 0



THE HARDWARE WILL BIAS ALL EXPONENTS BY 2000B. THEREFORE:

$1*2^{0}$=2000 0000 0000 0000 0001

$1*2^{1}$=2001 0000 0000 0000 0001

$1*2^{5}$=2005 0000 0000 0000 0001

Floating Point Arithmetic
Pack

THE IMPORTANCE OF THE BIAS BECOMES EVIDENT WHEN IT IS NECESSARY TO DEAL WITH NEGATIVE EXPONENTS. FOR EXAMPLE:

$1*2^{-0}$=1777 0000 0000 0000 0001
$1*2^{-1}$=1776 0000 0000 0000 0001
$1*2^{-5}$=1772 0000 0000 0000 0001

THE INSTRUCTIONS NECESSARY FOR PRODUCING $1*2^{-5}$ ARE:

```
SX1   1       (X1)=0000 0000 0000 0000 0001
SB1   -5      (B1)=777772
PX2   X1,B1   (X2)=1772 0000 0000 0000 0001
```

WITH THE ABOVE EXAMPLES, THE PACK INSTRUCTION PERFORMED THE FOLLOWING OPERATIONS.

1. IT TRANSFERRED THE LOWEST 48 BITS OF X1 TO X2 TO FORM THE COEFFICIENT. THE UPPER 12 BITS OF X1, EXCEPT FOR THE SIGN BIT, ARE NOT USED.
2. IT TRANSFERRED THE LOWEST 11 BITS OF B1 TO BITS 48 TO 58 OF X2 TO FORM THE EXPONENT. THE REMAINING BITS OF B1 ARE IGNORED.
3. BIT 58 IN X2 WAS TOGGLED. THIS EFFECTIVELY BIASED THE EXPONENT BY 2000 OCTAL.
4. THE SIGN OF X1 WAS TRANSFERRED FROM X1.
5. IF THE SIGN OF X1 HAD BEEN NEGATIVE, THE 11-BIT EXPONENT WOULD HAVE BEEN COMPLEMENTED (SEE STEP 2).

THE SAME PROCEDURE THUS WORKS FOR NEGATIVE NUMBERS. FOR EXAMPLE:

```
SX1   -1      (X1)=7777 7777 7777 7777 7776
SB1   2       (B1)=000002
PX2   X1,B1   (X2)=5775 7777 7777 7777 7776
SX1   -1      (X1)=7777 7777 7777 7777 7776
SB1   -2      (B1)=777775
PX2   X1,B1   (X2)=6002 7777 7777 7777 7776
```

IT BECOMES APPARENT THAT THERE ARE SOME LIMITS TO THE SIZE OF AN EXPONENT AND COEFFICIENT.

THE COEFFICIENT IS ORDINARILY LIMITED TO 48 BITS OF SIGNIFICANCE. (LATER WE WILL SEE HOW DOUBLE PRECISION GIVES 96 BITS OF SIGNIFICANCE.)

THE EXPONENT REACHES ITS UPPER LIMITS FOR A POSITIVE NUMBER AT 3776. WHEN A NUMBER HAS AN EXPONENT OF 3777 OR $N*2^{1777}$ IT HAS OVERFLOWED THE MACHINE. SUCH A NUMBER IS IDENTIFIED AS <u>INFINITE</u> OR <u>OUT OF RANGE</u>. THUS THE FOLLOWING NUMBER WOULD BE INFINITE:

```
3777 0000 0000 0000 0001
```

ON A 7000 TYPE MACHINE (CYBER 76), AN ERROR WOULD BE IMMEDIATELY NOTED AND THE USER WOULD USUALLY ABORT AT THIS POINT. ON A 6000 MACHINE (LOWER CYBER), THE PROGRAM WOULD USUALLY ABORT WHEN IT ATTEMPTED TO USE THIS NUMBER IN A FLOATING POINT OPERATION.

A NEGATIVE NUMBER ALSO HAS AN UPPER LIMIT. IF $1*2^{1777}$ IS INFINITE, SO IS $-1*2^{1777}$. THAT NUMBER WOULD BE REPRESENTED AS:

```
4000 7777 7777 7777 7776
```

THUS, 3777 AND 4000 ARE EXPONENTS THAT ARE INFINITE.

IT IS PERMISSIBLE TO HAVE EXPONENTS REPRESENTING $N*2-^{0}$. THESE EXPONENTS ARE 1777 FOR POSITIVE NUMBERS, 6000 FOR NEGATIVE NUMBERS. THESE NUMBERS ARE IDENTIFIED AS <u>INDEFINITE</u> ON A 6000 MACHINE AND <u>OUT OF RANGE</u> ON A 7000. ANY USE OF THESE NUMBERS PRODUCES AN ERROR AND USUALLY AN ABORT.

UNDERFLOW IS REPRESENTED BY EXPONENTS OF 0000 FOR POSITIVE NUMBERS AND 7777 FOR NEGATIVE NUMBERS. NO ERROR CONDITION IS PRODUCED ON A 6000 MACHINE; A 7000 WILL PRODUCE AN ERROR CONDITION WHEN UNDERFLOW OCCURS. AN ATTEMPT TO USE A NUMBER WITH AN EXPONENT OF 0000 OR 7777 WILL NOT PRODUCE AN ERROR ON A 6000. HOWEVER THE RESULT OF ANY OPERATION USING THE NUMBER WILL ALSO HAVE AN EXPONENT OF 0000 OR 7777. THUS, IF:

```
        (X1)=0000 5555 5555 5555 5555

        (X2)=2000 1111 1111 1111 1111

  FX3   X1+X2 PRODUCES (X3)=0000 0000 0000 0000 0000
```

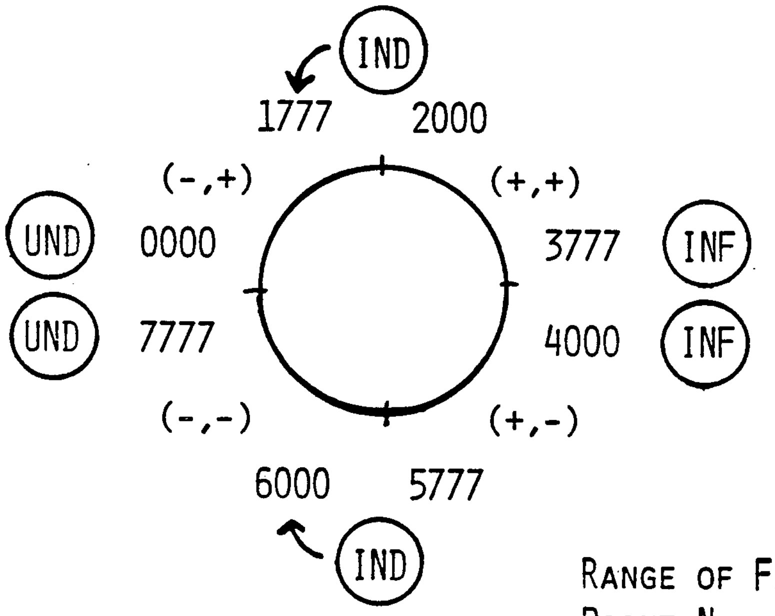
THE PROGRAMMER IS RESPONSIBLE FOR DEVELOPING HIS OWN TEST FOR UNDERFLOW.

FLOATING POINT ARITHMETIC
EXERCISES - SPECIAL EXPONENTS

EXERCISE 1

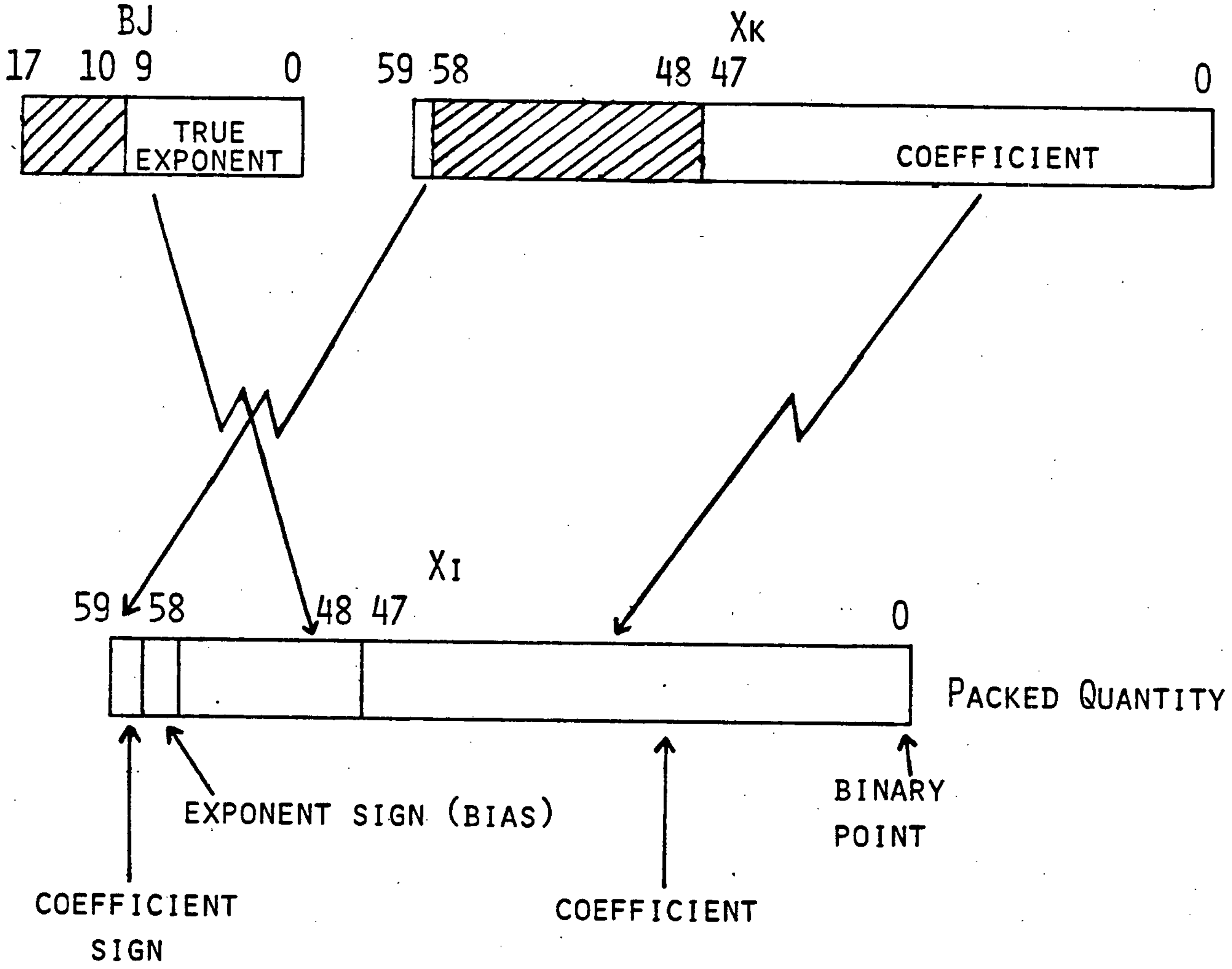
DIRECTIONS: FILL IN THE MISSING ITEMS ON THIS CHART (REFER TO CHART AT BOTTOM OF PAGE FOR RANGES).

UPPER 12 BITS	EXPONENTS (+ OR -)	COEFFICIENT (+ OR -)
7777	UNDERFLOW	NEGATIVE
7776	NEGATIVE	NEGATIVE
6001		
	INDEFINITE	NEGATIVE
5777	POSITIVE	
4001		
4000		
	INFINITE	
3776		
2000		
	INDEFINITE	
1776		
0000		



RANGE OF FLOATING POINT NUMBERS

PACKING: PXI BJ,XK



HOW TO PACK A FLOATING POINT NUMBER:

$$(X1) = 00000000000000000004_8$$

PX2 X1,B0

$$(X2) = \underbrace{2000}_{\text{EXP BIASED}}\underbrace{0000000000000004}_{\text{TAKEN FROM LOWER 48 BITS OF X1 COEFFICIENT}}{}_8$$

0000 ← 10-BIT EXP TAKEN FROM BO

$$\begin{array}{r} 010 \\ +\ 2000_8 \\ \hline 2000_8 \end{array}$$

$$\overbrace{010\ 000\ 000\ 000}^{2000_8} \quad \text{(BINARY)}$$

SIGN OF EXP.

BIAS A POSITIVE EXPONENT BY ADDING 2000_8 TO THE TRUE EXPONENT.

THIS SETS THE EXPONENT SIGN BIT TO A 1 AND REPRESENTS THE TRUE EXPONENT IN THE REMAINING 10 BITS.

HOW TO NORMALIZE A PACKED NUMBER

$(X2) = 20000000000000000004_8$

NX3 X2,B2

$(X3) = \underbrace{1722}_{\text{ADJUSTED EXPONENT}}\underline{400000000}0000000_8$ $(B2) = 000055_8$

ADJUSTED EXPONENT

COEFFICIENT IS SHIFTED LEFT UNTIL ITS MOST SIGNIFICANT ONE BIT IS IN BIT POS. 47

1777_8 BIAS MAKES EXP. SIGN BIT 0 AND COMPLIMENTS EXP.

-55_8 EXP. (NO. OF BITS SHIFTED LEFT)

$$\begin{array}{r} 1777_8 \\ -55_8 \\ \hline 1722_8 \end{array}$$

$\overset{\uparrow}{\text{ool}}\ \underbrace{\text{111 olo olo}}$ ← (BINARY)

SIGN OF EXPONENT	10-BIT COMPLEMENTED EXPONENT

THE TRUE VALUE OF THE NUMBER IS:

$$4000000000000000 \times 2^{-55_8}$$

BIAS A NEGATIVE EXPONENT BY SUBTRACTING IT FROM 1777_8.

THIS MAKES THE EXPONENT SIGN BIT A 0 AND REPRESENTS THE TRUE EXPONENT IN COMPLEMENTED FORM.

HOW TO PACK AND NORMALIZE A NEGATIVE NUMBER

** PACK THE NUMBER AS THOUGH IT WERE POSITIVE --- THEN COMPLEMENT THE ENTIRE WORD ***

EXAMPLE:

(X3) = -4

PACK IT:

(X3) = 20000000000000000004_8

NORMALIZE IT:

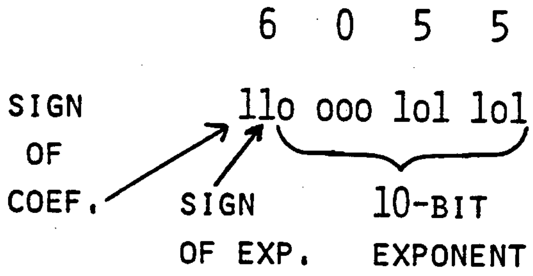
(X3) = 17224000000000000000_8

COMPLEMENT IT:

(X3) = 60553777777777777777_8

LOGICALLY THIS IS THE OPERATION ACCOMPLISHED

THE BINARY EXPONENT FIELD IS:



THE COEFFICIENT IS COMPLEMENTED:

3777777777777777_8 (48 BITS)

EXAMPLES OF PACKED AND NORMALIZED NUMBERS

NOTE: ALL NUMBERS ARE OCTAL

$40_8 = 40_8 \times 2^0$

20000000000000000040_8 PACK IT
17254000000000000000_8 NORM. IT

$.004_8 = 4_8 \times 2^{-11_8}$

17660000000000000004_8 PACK IT
17114000000000000000_8 NORM. IT

$-40_8 = -40_8 \times 2^0$

20000000000000000040_8 PACK IT
17254000000000000000_8 NORM. IT
60523777777777777777_8 COMPLEMENT IT

$-.004_8 = -4_8 \times 2^{-11_8}$

17660000000000000004_8 PACK IT
17114000000000000000_8 NORM. IT
60663777777777777777_8 COMPLEMENT IT

$1.4_8 = 14_8 \times 2^{-3}$

17740000000000000014_8 PACK IT
17206000000000000000_8 NORM. IT

$-1.4_8 = -14_8 \times 2^{-3}$

17740000000000000014_8 PACK IT
17206000000000000000_8 NORM. IT
60571777777777777777_8 COMPLEMENT IT

NORMALIZATION - TWO INSTRUCTIONS WILL PRODUCE NORMALIZATION:

```
NX3   X2,B2

ZX3   X2,B2
```

NOTE

THE ZX3 INSTRUCTION ALSO ROUNDS, AS WILL BE DISCUSSED LATER.

NORMALIZING IS THE PROCESS BY WHICH THE HARDWARE LEFT SHIFTS A COEFFICIENT UNTIL THE MOST SIGNIFICANT BIT IS AT BIT POSITION 47 (I.E., UNTIL THE SIGN BIT AND BIT 47 ARE DIFFERENT). AS THE HARDWARE SHIFTS, THE EXPONENT IS DECREMENTED ACCORDINGLY. THE ORIGINAL AND NORMALIZED VALUE ARE THE SAME BUT THEIR INTERNAL REPRESENTATION DIFFERS; THAT IS, THEY HAVE DIFFERENT COEFFICIENTS AND EXPONENTS.

THESE ARE ALL WAYS OF REPRESENTING THE SAME VALUE:

1 * 2^{0} OR 2000 0000 0000 0000 0001

2 * 2^{-1} OR 1776 0000 0000 0000 0002

4 * 2^{-2} OR 1775 0000 0000 0000 0004

10B * 2^{-3} OR 1774 0000 0000 0000 0010

⋮ ⋮

4000 0000 0000 0000 * 2^{-47} OR 1720 4000 0000 0000 0000

THIS LAST VALUE IS THE QUANTITY 1 PACKED AND NORMALIZED. THESE INSTRUCTIONS WOULD ACCOMPLISH THIS:

```
 SX1   1         (X1)  =  0000 0000 0000 0000 0001

 PX2   X1, B0    (X2)  =  2000 0000 0000 0000 0001

 NX3   X2,B1*    (X3)  =  1720 4000 0000 0000 0000

                 (B1)  =  000057
```

AS THE MOST SIGNIFICANT BIT IS SHIFTED LEFT, THE SIGN BIT IS EXTENDED END-AROUND. THUS, WHILE THE NORMALIZED VALUE OF 1 IS:

```
1720 4000 0000 0000 0000
```

THE NORMALIZED VALUE OF -1 IS:

```
6057 3777 7777 7777 7777
```

IF THE B REGISTER IS SPECIFIED, IT CONTAINS THE NUMBER OF POSITIONS THE COEFFICIENT WAS SHIFTED BEFORE IT NORMALIZED.

To convert a floating point number to an integer, the programmer must unpack the floating point number and left shift the result.

For example, assume that a programmer wants to convert the number 5.0 (in X3) to the integer 5 (in X1). The procedure is as follows.

```
UX2   B2,X3
LX1   X2,B2
```

The unpack instruction transferred the lower 48 bits of X3 to X2 and extended the sign bit in the upper 12 bits:

```
(X3) = 1722 5000 0000 0000 0000
(X2) = 0000 5000 0000 0000 0000
```

Bits 48 through 57 of X3 are transferred to the lower 10 bits of the B register. Bit 58 is toggled and then used to fill in the rest of the B register bits.

```
(B2) = 777722 (-55)
```

The left shift produces an effective arithmetic right shift with sign extension of 55 octal or 45 decimal places.

```
(X1) = 0000 0000 0000 0000 0005
```

A negative number would be treated similarly but the exponent would be complemented first. For example, the complement of 5.0 seen above would be:

```
(X3) = 6055 2777 7777 7777 7777  (-5.0)
```

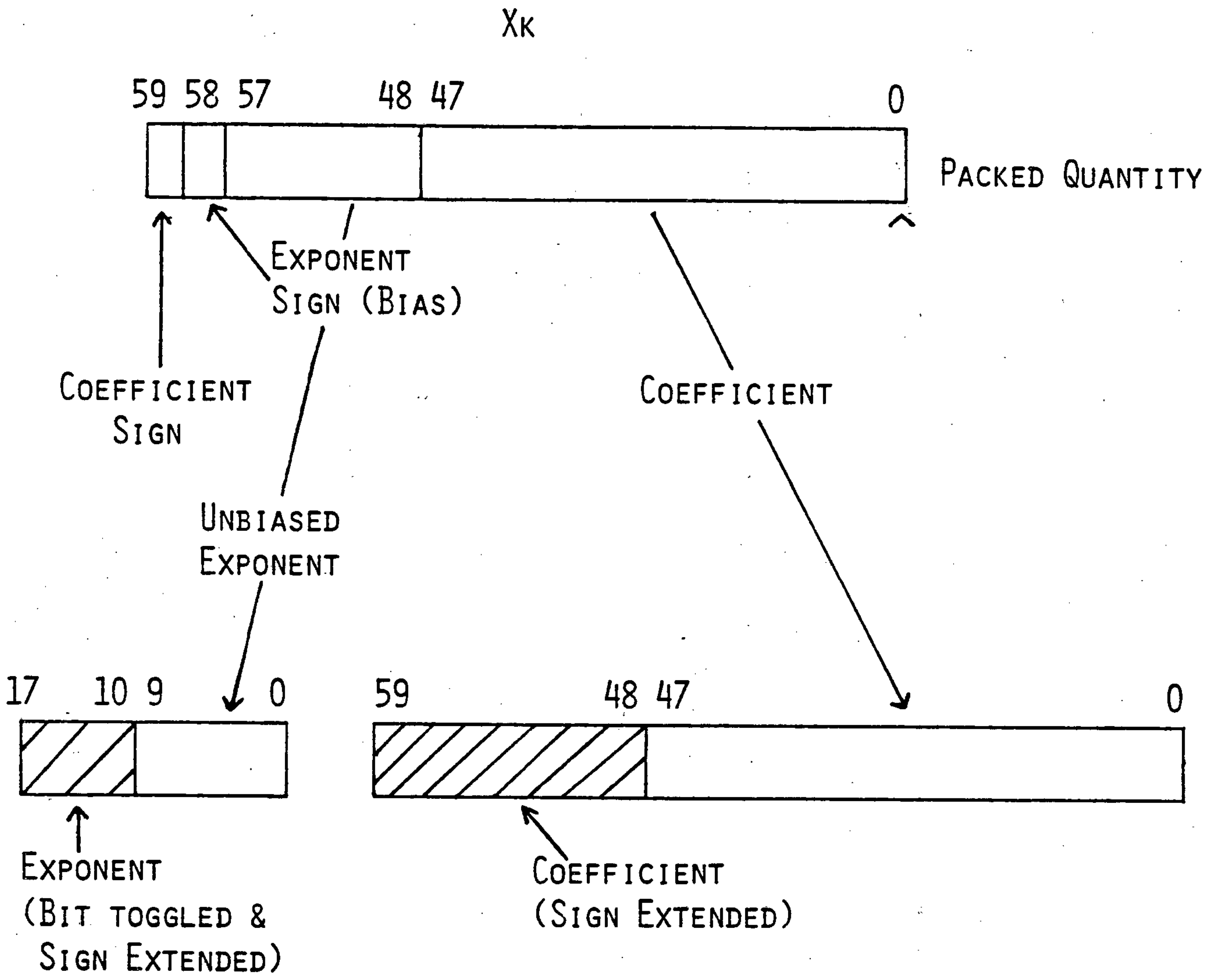
UX2 B2,X3 would produce:

```
(X2) = 7777 2777 7777 7777 7777
(B2) = 777722
```

LX1 X2,B2 would produce:

```
(X1) = 7777 7777 7777 7777 7772  (-5)
```

UNPACKING: UX_I B_J,X_K



UNPACKED QUANTITY

How to unpack a number

$$(X3) \;=\; 17224000000000000000_8$$

```
UX4     X3,B4
```

$$(X4) \;=\; \underbrace{0000}_{\text{SIGN OF COEF. EXTENDED}}\underbrace{4000000000000000}_{\text{COEF. FROM LOWER 48 BITS OF X3}}{}_8$$

$$(B4) \;=\; 777722_8 \;=\; -55_8 \;=\; -45_{10}$$

To unnormalize it: (There is no unnormalize instruction)

```
LX7     X4,B4
```

$$(X7) \;=\; 0000000000000\underrightarrow{0000004}{}_8$$

The coef. was shifted right 55_8 (45_{10}) bits due to the -55 in B4

ARITHMETIC

THE FLOATING POINT FUNCTIONS ARE:

FX3	X1+X2
FX3	X1-X2
FX3	X1*X1
FX3	X1/X2

FLOATING POINT ARITHMETIC TAKES PLACE IN A 96-BIT ACCUMULATOR. EACH FUNCTION HAS ITS OWN PROCEDURE. THE PROCESS BY WHICH THE HARDWARE ADDS WILL GENERALLY ILLUSTRATE HOW OTHER FUNCTIONS WORK.

ASSUME THAT WE WANT TO ADD 4.0 AND 1.0.

```
1.0  =  1720 4000 0000 0000 0000

4.0  =  1722 4000 0000 0000 0000
```

TO ADD THESE TWO NUMBERS THE HARDWARE PERFORMS THE FOLLOWING PROCEDURES:

1. BOTH ARGUMENTS ARE UNPACKED.

2. THE ARGUMENT WITH THE SMALLER EXPONENET IS PLACED IN THE 96-BIT ACCUMULATOR AND SHIFTED RIGHT BY THE DIFFERENCE IN EXPONENTS.

   THUS:

   ```
   1720 4000 0000 0000 0000
   ```

   BECOMES:

   ```
   1000 0000 0000 0000 0000 0000 0000 0000
   ```

3. THE COEFFICIENT WITH THE LARGER EXPONENT IS ENTERED INTO THE UPPER HALF OF THE ACCUMULATOR.

   THUS:

   1000 0000 0000 0000 0000 0000 0000 0000

   BECOMES:

   5000 0000 0000 0000 0000 0000 0000 0000

4. IF OVERFLOW OCCURS THE SUM IS RIGHT SHIFTED ONE PLACE AND THE EXPONENT IS INCREASED BY ONE.

5. THE UPPER 48 BITS OF THE ACCUMULATOR PLUS THE EXPONENT ARE RETURNED TO THE RESULT X REGISTER. THE RESULT IS:

   1722 5000 0000 0000 0000

ROUND FUNCTION

THE ROUNDING FUNCTIONS ARE:

```
RX3     X1+X2
RX3     X1*X2
RX3     X1-X2
RX3     X1/X2
```

THE ROUND FUNCTION IS IDENTICAL TO THE STANDARD FLOATING POINT FUNCTION EXCEPT THAT A BIT IS ADDED TO THE RESULT IN THE LEAST SIGNIFICANT POSITION. THIS BIT ADDITION REDUCES ERROR WHEN NUMBERS ARE OPERATED ON SEVERAL TIMES.

THE FOLLOWING PROGRAM MULTIPLIES THE NUMBER 1.55533 BY ITSELF A THOUSAND TIMES, USING BOTH THE STANDARD F FUNCTION AND THE R FUNCTION:

```
            IDENT     MULT
            ENTRY     START
RESULTS     BSSZ      2
NUM         DATA      1.55533
START       SA1       NUM
            SX6       1
            PX6       X6
            NX6       X6
            BX7       X6
            SB1       1
            SB2       1000
LOOP        FX6       X6*X1
            RX7       X7*X1
            SB1       B1+1
            NE        B1,B2,LOOP
            SA6       RESULTS
            SA7       RESULTS+1
            ENDRUN
            END       START
```

AFTER 1000 MULTIPLICATIONS, THE RESULTS OF THE FUNCTIONS ARE:

```
FX6   (X6)  =  3115 5774 6776 0716 4533

RX7   (X7)  =  3115 5774 6776 0716 5265
```

DOUBLE PRECISION FUNCTIONS

THE DOUBLE PRECISION FUNCTIONS ARE:

```
DX4   X1+X2

DX4   X1-X2

DX4   X1*X2
```

WHEN DOUBLE PRECISION FUNCTIONS ARE PERFORMED, THE HARDWARE RETURNS TO THE RESULT X REGISTER THE LOWER 48 BITS OF THE ACCUMULATOR PLUS THE EXPONENT.

TO OBTAIN BOTH THE UPPER 48 BITS AND THE LOWER 48 BITS OF THE 96 BIT ACCUMULATOR, THE PROGRAMMER MUST PERFORM TWO OPERATIONS:

```
FX3  X1+X2   (X3 WILL RECEIVE THE MOST SIGNIFI-
              CANT BITS - UPPER 48 BITS.)

DX4  X1+X2   (X4 WILL RECEIVE THE LEAST SIGNIFI-
              CANT BITS - LOWER 48 BITS.)
```

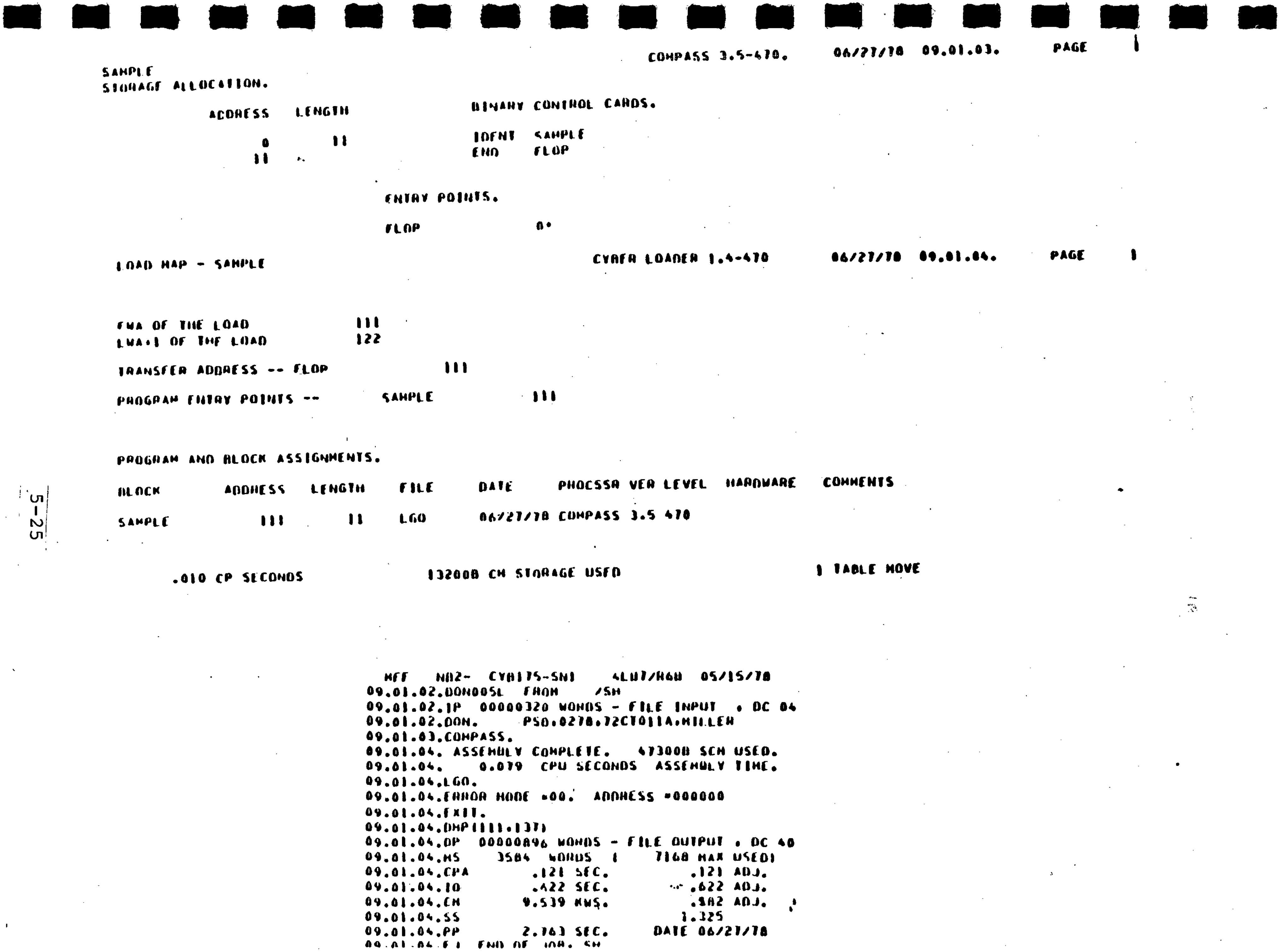
FOR EXAMPLE, IF A PROGRAMMER ADDS 1.0 IN X1 AND 4.0 IN X2 AND WANTS TO OBTAIN THE MOST AND LEAST SIGNIFICANT BITS, HE MUST PERFORM THESE OPERATIONS.

```
FX3  X1+X2   (X3) = 1722 5000 0000 0000 0000

DX4  X1+X2   (X4) = 1642 0000 0000 0000 0000
```

NOTE

THE EXPONENT IN X4 IS OFFSET BY 60B.

```
SAMPLE                                                       COMPASS 3.5-470.     06/27/78  09.01.03.      PAGE      2

                                              IDENT  SAMPLE
                                              ENTRY  FLOP
                                      *
                                      *
                                      *       EXAMPLE CODE FOR PROBLEM
                                      *
                                      *
       0  5110000005 +               FLOP     SA1    A            LOAD OPERAND
                    5120000006 +              SA2    B            LOAD OPERAND
       1  5130000007 +                        SA3    C            LOAD OPERAND
                    30412                     FX4    X1+X2
                         44643                FX5    X4/X3
       2  40611                               FX6    X1*X1
               40722                          FX7    X2*X2
                    30063                     FX0    X6+X3
                         30707                FX7    X0+X7
       3  40757                               FX7    X5*X7
               5170000010 +                   SA7    ANSWER       FLOATING POINT ANSWER
       4  0200000000                          JP     0
       5  17216000000000000000       A        DATA   3.0          FLOATING POINT DATA
       6  17224000000000000000       B        DATA   4.0          FLOATING POINT DATA
       7  17214000000000000000       C        DATA   2.0          FLOATING POINT DATA
      10                       1     ANSWER   BSS    1
      11                                      END    FLOP

                        47300B SCM STORAGE USED          24 STATEMENTS        9 SYMBOLS
                              MODEL 174 ASSEMBLY      0.027 SECONDS          10 REFERENCES

DMPX.

 P    000000  A0  000200  B0  000000
 RA   201600  A1  000116  B1  000001     C(A1)=  1721  6000  0000  0000  0000     C(B1)=  0000  0000  0000  0000  0000
 FL   000200  A2  000117  B2  000002     C(A2)=  1722  4000  0000  0000  0000     C(B2)=  0000  0000  0000  0000  0000
 EM   700700  A3  000120  B3  012733     C(A3)=  1721  4000  0000  0000  0000     C(B3)=
 RE   000045  A4  000001  B4  000201     C(A4)=  0000  0000  0000  0000  0000     C(B4)=
 FE   000000  A5  000111  B5  000111     C(A5)=  5110  0001  1651  2000  0117     C(B5)=  5110  0001  1651  2000  0117
 MA   000600  A6  000001  B6  000200     C(A6)=  0000  0000  0000  0000  0000     C(B6)=
              A7  000121  B7  027756     C(A7)=  1726  5720  0000  0000  0000     C(B7)=
 X0   1723  5400  0000  0000  0000
 X1   1721  6000  0000  0000  0000
 X2   1722  4000  0000  0000  0000
 X3   1721  4000  0000  0000  0000
 X4   1722  7000  0000  0000  0000
 X5   1721  7000  0000  0000  0000
 X6   1723  4400  0000  0000  0000
 X7   1726  5720  0000  0000  0000

    00000    00000 00000 00000 00000                                                                             00000 00000 00000 00000
    00054   -56110 03110 00054 54710     51100 00001 03110 00055      64550 02550 00000 46000      04000 00063 00000 00021
    00060    15051 52000 00000 00061     00000 00200 00000 00001      07040 00060 51600 00001      00000 00000 40000 00000
    00064    14071 70000 00000 00000     00000 00000 00000 00122      40000 00000 02000 00111
    00070    14071 75755 00000 00000     00000 00000 00000 00000
    00100   -54000 00000 01000 00001

  DUMP     RELATIVE                         DMP(111,117)

       00111    51100 00116 51200 00117      51300 00120 10412 44543      40611 40722 30063 30707
       00114    40757 51700 00121 46000      02000 00000 61000 46000      17216 00000 00000 00000      17224 00000 00000 00
```



```
                                                                 COMPASS 3.5-470.     06/27/78  09.01.03.       PAGE     1
SAMPLE
STORAGE ALLOCATION.

           ADDRESS   LENGTH                  BINARY CONTROL CARDS.

                 0        11                 IDENT   SAMPLE
                11                           END     FLOP

                                      ENTRY POINTS.

                                      FLOP             0*

 LOAD MAP - SAMPLE                                          CYBER LOADER 1.4-470       06/27/78  09.01.04.       PAGE     1


 FWA OF THE LOAD             111
 LWA+1 OF THE LOAD           122

 TRANSFER ADDRESS -- FLOP              111

 PROGRAM ENTRY POINTS --        SAMPLE             111


 PROGRAM AND BLOCK ASSIGNMENTS.

 BLOCK        ADDRESS   LENGTH    FILE      DATE     PROCSSR VER LEVEL  HARDWARE   COMMENTS

 SAMPLE           111       11    LGO      06/27/78 COMPASS 3.5 470


      .010 CP SECONDS             13200B CM STORAGE USED                          1 TABLE MOVE



                                  MFF   NB2-  CYB175-SN1    4LU7/R6B  05/15/78
                                 09.01.02.DON005L  FROM    /SH
                                 09.01.02.IP  00000320 WORDS - FILE INPUT  , DC 04
                                 09.01.02.DON.     PSO,0278,72CT011A,MILLER
                                 09.01.03.COMPASS.
                                 09.01.04. ASSEMBLY COMPLETE.   47300B SCM USED.
                                 09.01.04.    0.079  CPU SECONDS  ASSEMBLY TIME.
                                 09.01.04.LGO.
                                 09.01.04.ERROR MODE =00.  ADDRESS =000000
                                 09.01.04.EXIT.
                                 09.01.04.DMP(111,137)
                                 09.01.04.OP  00000896 WORDS - FILE OUTPUT , DC 40
                                 09.01.04.MS     3584  WORDS  (    7168 MAX USED)
                                 09.01.04.CPA        .121 SEC.          .121 ADJ.
                                 09.01.04.IO         .422 SEC.          .622 ADJ.
                                 09.01.04.CM        9.539 KWS.          .582 ADJ.
                                 09.01.04.SS                           1.325
                                 09.01.04.PP        2.763 SEC.     DATE 06/27/78
                                 09.01.04.EJ  END OF JOB, SH
```

```
                                                                 COMPASS 3.5-470.     06/27/78  16.19.06.      PAGE      1
          INCNVRT
          STORAGE ALLOCATION.

                    ADDRESS   LENGTH                BINARY CONTROL CARDS.

                          0        2                IDENT  INCNVRT
                          2                         END    INCNVRT     TRANSFER ADDRESS


                                             ENTRY POINTS.

                                             INCNVRT         0*


          INCNVRT                                                 COMPASS 3.5-470.     06/27/78  16.19.06.      PAGE      2


                                                       IDENT   INCNVRT
                                                       ENTRY   INCNVRT
                                                       LIST    $
                                               *
                                               *       CONVERSION OF INTEGER NUMBER TO FLOATING POINT
                                               *       NUMBER AND BACK TO INTEGER
                                               *
               0  7100123456                 INCNVRT   SX0     123456B      PLACE INTEGER IN X0
                            27100                      PX1     X0           PACK X0 INTO X1, B0 ASSUMED
                                 24211                 NX2     X1,B1        NORMALIZE X1 INTO X2, SHIFT COUNT IN B1
               1  26322                                UX3     X2,B2        UNPACK X2 INTO X3, SHIFT COUNT IN B2
                       22423                           LX4     X3,B2        LEFT SHIFT X3 INTO X4 USING COUNT IN B2
                                               *
                            0200000000                 JP      0            ABNORMAL TERMINATION
               2                                       END     INCNVRT      TRANSFER ADDRESS

                                  47306B SCM STORAGE USED            15 STATEMENTS         1 SYMBOLS
                                          MODEL 174 ASSEMBLY      0.016 SECONDS          2 REFERENCES

     DMPX.

      P    000000  A0   000200  B0   000000
      RA   220200  A1   000001  B1   000040       C(A1)=  0000  0000  0000  0000  0000     C(B1)=  0000  0000  0000  0000  0000
      FL   000200  A2   000060  B2   777737       C(A2)=  1505  1520  0000  0000  0061     C(B2)=
      EM   700700  A3   000057  B3   012733       C(A3)=  0000  0000  0000  0000  0000     C(B3)=
      RE   000045  A4   000001  B4   000201       C(A4)=  0000  0000  0000  0000  0000     C(B4)=
      FE   000000  A5   000111  B5   000111       C(A5)=  7100  1234  5627  1002  4211     C(B5)=  7100  1234  5627  1002  4211
      MA   000600  A6   000001  B6   000200       C(A6)=  0000  0000  0000  0000  0000     C(B6)=
                   A7   000001  B7   027756       C(A7)=  0000  0000  0000  0000  0000     C(B7)=
      X0   0000   0000   0000   0012   3456
      X1   2000   0000   0000   0012   3456
      X2   1737   5162   7000   0000   0000
      X3   0000   5162   7000   0000   0000
      X4   0000   0000   0000   0012   3456
      X5   6000   0000   0004   0040   0000
      X6   1505   1520   0000   0000   0061
      X7   0000   0000   0000   0000   0000

         00000    00000 00000 00000 00000
         00054   -56110 03110 00054 54710        51100 00001 03110 00055        64550 02550 00000 46000        00000 00000 00000 00000
         00060    15051 52000 00000 00061        00000 00200 00000 00001        07040 00060 51600 00001        04000 00063 00000 00021
         00064    14071 70000 00000 00000        00000 00000 00000 00113        40000 00000 02000 00111        00000 00000 40000 00000
         00070    14071 75755 00000 00000        00000 00000 00000 00000
         00100   -54000 00000 01000 00001
```

```
INCNVRT                                                                   COMPASS 3.5-470.      06/27/78  16.19.06.        PAGE      3
SYMBOLIC REFERENCE TABLE.

INCNVRT        0     PROGRAM*        2/02 E    2/08 L
```

```
LOAD MAP - INCNVRT                                              CYBER LOADER 1.4-470        06/27/78  16.19.07.        PAGE      1


FWA OF THE LOAD                 111
LWA+1 OF THE LOAD               113

TRANSFER ADDRESS -- INCNVRT             111

PROGRAM ENTRY POINTS --        INCNVRT           111


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK          ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

INCNVRT            111        2    LGO       06/27/78 COMPASS 3.5 470


     .007 CP SECONDS               13200B CM STORAGE USED                          1 TABLE MOVE
```

```
  MFF   NB2-  CYB175-SN1    4L87/R68  05/15/78
16.19.04.DON00ME  FROM    /SM
16.19.04.IP  00000256 WORDS - FILE INPUT  , DC 04
16.19.04.DON.     PS9.027B.72CT011A.MILLER
16.19.05.COMPASS.
16.19.06. ASSEMBLY COMPLETE.   47300B SCM USED.
16.19.06.    0.068  CPU SECONDS  ASSEMBLY TIME.
16.19.06.LGO.
16.19.07.ERROR MODE =00.  ADDRESS =000000
16.19.07.OP  00000704 WORDS - FILE OUTPUT , DC 40
16.19.07.MS    3584  WORDS  (   10752 MAX USED)
16.19.07.CPA          .103 SEC.          .103 ADJ.
16.19.07.IO           .612 SEC.          .612 ADJ.
16.19.08.CM         9.307 KWS.           .567 ADJ.
16.19.08.SS                             1.283
16.19.08.PP         3.295 SEC.     DATE 06/27/78
16.19.08.EJ  END OF JOB, SM
```

```
BIGINT                                                        COMPASS 3.5-470.     06/27/78  08.50.56.      PAGE     1
STORAGE ALLOCATION.

          ADDRESS   LENGTH              BINARY CONTROL CARDS.

                0        5              IDENT  BIGINT
                5                       END    BIGINT      TRANSFER ADDRESS

                                  BLOCKS     TYPE      ADDRESS    LENGTH

                                  PROGRAM*  LOCAL            0         4
                                  LITERALS* LOCAL            4         1

                                  ENTRY POINTS.

                                  BIGINT          0*
```

```
 LOAD MAP - BIGINT                                        CYBER LOADER 1.4-470      06/27/78  08.50.57.      PAGE     1


 FWA OF THE LOAD            111
 LWA+1 OF THE LOAD          116

 TRANSFER ADDRESS -- BIGINT           111

 PROGRAM ENTRY POINTS --      BIGINT           111


 PROGRAM AND BLOCK ASSIGNMENTS.

 BLOCK       ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

 BIGINT          111        5    LGO      06/27/78 COMPASS 3.5 470


      .008 CP SECONDS             132008 CM STORAGE USED
```

```
   MFF   NB2-  CYB175-SN1    4L87/A6B  05/15/78
08.50.54.DON004J  FROM    /SH
08.50.54.IP  00000320 WORDS - FILE INPUT  , DC 04
08.50.54.DON.     PSD,0278,72CT011A,MILLER
08.50.56.COMPASS.
08.50.57. ASSEMBLY COMPLETE.   473008 SCM USED.
08.50.57.    0.082  CPU SECONDS  ASSEMBLY TIME.
08.50.57.LGO.
08.50.57.ERROR MODE =00.  ADDRESS =000000
08.50.58.OP  00000896 WORDS - FILE OUTPUT , DC 40
08.50.58.MS     3584  WORDS  (   10752 MAX USED)
08.50.58.CPA          .130 SEC.          .130 ADJ.
08.50.58.IO           .616 SEC.          .616 ADJ.
08.50.58.CM          9.770 KWS.          .596 ADJ.
08.50.58.SS                             1.343
08.50.58.PP          3.526 SEC.     DATE 06/27/78
08.50.58.EJ  END OF JOB, SH
```

```
                                           *************************************************************************
                                           *                                                                       *
                                           *                 CONVERSION OF A LARGE INTEGER TO FLOATING POINT TO INTEGER *
                                           *                                                                       *
                                           *************************************************************************

                                                     IDENT  BIGINT
                                                     ENTRY  BIGINT
                                           *
                                           *         CONVERSION OF A LARGE INTEGER (I.E. GREATER THAN 48 BITS) TO A
                                           *         FLOATING POINT NUMBER AND BACK TO AN INTEGER
                                           *
      0  46000                              BIGINT   NO
                                           *
                                           *         LOAD NUMBER INTO X1
                                           *
              5110000004 +                           SA1    =1234567123456712348
      1  6110000006                                  SB1    6             PLACE SHIFT COUNT IN B1
                        23211                        AX2    X1,B1         SHIFT BEFORE PACKING
                             27312                   PX3    X2,B1         PACK USING SHIFT COUNT IN B1
      2  24423                                       NX4    X3,B2         NORMALIZE X3 INTO X4, SHIFT COUNT TO B2
                                           *
                                           *         PACKED-NORMALIZED NUMBER NOW IN X4, SHIFT COUNT IN B2
                                           *
              26534                                  UX5    X4,B3         UNPACK X4 INTO X5, SHIFT COUNT TO B3
                   22635                             LX6    X5,B3         LEFT SHIFT I.E. FINISH UNPACKING
                        27701                        PX7    X1            PACKING WITHOUT SHIFTING
      3  0200000000                                  JP     0             ABNORMAL TERMINATION
      5                                              END    BIGINT        TRANSFER ADDRESS

                         47300B SCM STORAGE USED              29 STATEMENTS          1 SYMBOLS
                                  MODEL 174 ASSEMBLY        0.032 SECONDS            2 REFERENCES
DMPX.

  P    000000  A0   000200  B0   000000
  RA   240200  A1   000115  B1   000006     C(A1)=  0012  3456  7123  4567  1234     C(B1)=  0000  0000  0000  0000  0000
  FL   000200  A2   000060  B2   000002     C(A2)=  1505  1520  0000  0000  0061     C(B2)=  0000  0000  0000  0000  0000
  EM   700700  A3   000057  B3   000004     C(A3)=  0000  0000  0000  0000  0000     C(B3)=  0000  0000  0000  0000  0000
  RE   000051  A4   000001  B4   000201     C(A4)=  0000  0000  0000  0000  0000     C(B4)=
  FE   000000  A5   000111  B5   000111     C(A5)=  4600  0511  0000  1154  6000     C(B5)=  4600  0511  0000  1154  6000
  MA   001000  A6   000001  B6   000200     C(A6)=  0000  0000  0000  0000  0000     C(B6)=
               A7   000001  B7   027756     C(A7)=  0000  0000  0000  0000  0000     C(B7)=
  X0   0000   0000   0000   0000   0000
  X1   0012   3456   7123   4567   1234
  X2   0000   1234   5671   2345   6712
  X3   2006   1234   5671   2345   6712
  X4   2004   5162   7345   1627   3450
  X5   0000   5162   7345   1627   3450
  X6   0012   3456   7123   4567   1200
  X7   2000   3456   7123   4567   1234

     00000    00000 00000 00000 00000
     00054   +56110 03110 00054 54710       51100 00001 03110 00055      64550 02550 00000 46000      00000 00000 00000 00000
     00060    15051 52000 00000 00061       00000 00200 00000 00001      07040 00060 51600 00001      04000 00063 00000 00021
     00064    14071 70000 00000 00000       00000 00000 00000 00116      40000 00000 02000 00111      00000 00000 40000 00000
     00070    14071 75755 00000 00000       00000 00000 00000 00000
```

```
                                      IDENT   EXPONET
                                      ENTRY   EXPONET
                                      LIST    *
                              ***************************************************************
                              *                                                             *
                              *            RANGE OF EXPONENTS FOR VALID UNPACKING           *
                              *                                                             *
                              *            UX1    BJ,XK                                     *
                              *            LX1    BJ,XK                                     *
                              *                                                             *
                              *   IF TRUE EXPONENT, AND THUS BJ<-48 THEN ALL OF THE SIGNIFICANT BITS WILL
                              *   BE SHIFTED OFF THE RIGHT END OF THE WORD LEAVING +0 OR -0.
                              *                                                             *
                              *   IF TRUE EXPONENT IS POSITIVE, THEN THE SHIFT IS LEFT, INCREASING THE
                              *   VALUE OF THE COEFFICIENT.  IF THE SHIFT IS 12 BITS OR GREATER, YOU WILL
                              *   HAVE SHIFTED A NORMALIZED COEFFICIENT INTO SIGN POSITION AND THEN END
                              *   AROUND, THUS CHANGING THE VALUE OF THE NUMBER.
                              *                                                             *
                              *   THUS, A VALID RESULT IS ASSURED IF:  -47 ≤ TRUE EXPONENT ≤ +11 (DECIMAL)
                              *                                                             *
                              ***************************************************************
                              *
                              *        SHOW EXPONENTS OUT OF RANGE
                              *
 0  5110000003 +              EXPONET  SA1    =20147654321000000000B  LOAD VALUE INTO X1
              26221                    UX2    X1,B2          UNPACK X1 INTO X2, SHIFT COUNT TO B2
                   22322               LX3    X2,B2          LEFT SHIFT X2 INTO X3, B2 PLACES
                              *
 1  5140000004 +                       SA4    =17177654321000000000B  LOAD VALUE INTO X4
              26554                    UX5    X4,B5          UNPACK X4 INTO X5, SHIFT COUNT TO B5
                   22655               LX6    X5,B5          LEFT SHIFT X5 INTO X6, BY B5 PLACES
                              *
 2  0200000000                         JP     0              ABNORMAL TERMINATION

                              CONTENT OF LITERALS BLOCK.

 3  20147654321000000000      PL¬=ZH
 4  17177654321000000000      OO¬=ZH

 5                                     END    EXPONET        TRANSFER ADDRESS

             47300B SCM STORAGE USED              34 STATEMENTS          1 SYMBOL
                    MODEL 174 ASSEMBLY         0.036 SECONDS             2 REFERE
```

```
DMPX.

 P     000000   A0   000200   B0   000000
 RA    125200   A1   000114   B1   000001
 FL    000200   A2   000060   B2   000014
 EM    700700   A3   000057   B3   012733
 RE    000041   A4   000115   B4   000201
 FE    000000   A5   000111   B5   777717
 MA    000400   A6   000001   B6   000200
                A7   000001   B7   027756
 X0   0000   0000   0000   0000   0000
 X1   2014   7654   3210   0000   0000
 X2   0000   7654   3210   0000   0000
 X3   7654   3210   0000   0000   0000
 X4   1717   7654   3210   0000   0000
 X5   0000   7654   3210   0000   0000
 X6   0000   0000   0000   0000   0000
 X7   0000   0000   0000   0000   0000

     00000     00000 00000 00000 00000
     00054    *56110 03110 00054 54710
     00060     15051 52000 00000 00061
     00064     14071 70000 00000 00000
     00070     14071 75755 00000 00000
         0        00 0     01     000
```

```
EXPONET                                                            COMPASS 3.5-470.     06/27/78  08.52.31.      PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH            BINARY CONTROL CARDS.

                0        5            IDENT   EXPONET
                5                     END     EXPONET     TRANSFER ADDRESS

                                 BLOCKS     TYPE       ADDRESS     LENGTH

                                 PROGRAM*  LOCAL             0          3
                                 LITERALS* LOCAL             3          2

                                 ENTRY POINTS.

                                 EXPONET          0*
```

```
EXPONET                                                            COMPASS 3.5-470.     06/27/78  08.52.31.      PAGE      3
SYMBOLIC REFERENCE TABLE.

EXPONET       0      PROGRAM*        2/02 F     2/25 L
```

```
LOAD MAP - EXPONET                                       CYBER LOADER 1.4-470       06/27/78  08.52.32.      PAGE      1


FWA OF THE LOAD             111
LWA+1 OF THE LOAD           116

TRANSFER ADDRESS -- EXPONET          111

PROGRAM ENTRY POINTS --        EXPONET          111


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK         ADDRESS   LENGTH    FILE    DATE     PROCSSR VER LEVEL  H

EXPONET           111        5    LGO     06/27/78 COMPASS 3.5 470


     .007 CP SECONDS                  13200B CM STORAGE USED
```

```
   MFF   NB2-  CYB175-SN1    4LB7/R6B  05/15/78
08.52.29.DON004S  FROM    /SM
08.52.29.IP  00000384 WORDS - FILE INPUT  , DC 04
08.52.29.DON.        PSO.0278.72CT011A.MILLER
08.52.31.COMPASS.
08.52.31. ASSEMBLY COMPLETE.   47300B SCM USED.
08.52.31.     0.089  CPU SECONDS  ASSEMBLY TIME.
08.52.31.LGO.
08.52.32.ERROR MODE =00.  ADDRESS =000000
08.52.32.OP  00001024 WORDS - FILE OUTPUT , DC 40
08.52.32.MS     3584  WORDS  (    10752 MAX USED)
08.52.32.CPA          .131 SEC.            .130 ADJ.
08.52.32.IO           .619 SEC.            .619 ADJ.
08.52.32.CM          9.902 KWS.            .609 ADJ.
```

```
                                                           IDENT  TESTFPO
                                                           ENTRY  TESTFPO
                                                           LIST   S
                                                 *
                                                 *         TEST FLOATING POINT ZERO
                                                 *
       0  5110000005 +                           TESTFPO   SA1    =0.0
                    10011                                  BX0    X1
       1  5110000006 +                                     SA1    =3.5
                    31211                                  FX2    X1-X1          PRODUCE 0.0 IN X2
                          40321                            FX3    X2*X1          ZERO TIMES A NUMBER
       2  44421                                            FX4    X2/X1          ZERO DIVIDED BY A NUMBER
              44512                                        FX5    X1/X2          NUMBER DIVIDED BY ZERO
                    30651                                  FX6    X5+X1          USE PREVIOUS ANSWER
                          30761                            FX7    X6+X1          USE PREVIOUS ANSWE
       3                                                   ENDRUN
       3  7160247021                              SX6 3REND*4+1                                                  ENDRUN  .1
                    20650                         LX6 40D                                                        ENDRUN  .1
       4  0100000000 X                            RJ =XSYS=                                                      ENDRUN  .1
                                                  ENDM                                                           ENDRUN  .1

                                                 DEFAULT SYMBOLS DEFINED BY COMPASS.

       0 X                                       SYS=

                                                 CONTENT OF LITERALS BLOCK.

       5  00000000000000000000
       6  17217000000000000000                   00+

       7                                                   END    TESTFPO     TRANSFER ADDRESS

                                 47300B SCM STORAGE USED              21 STATEMENTS          2 SYMBOLS
                                            MODEL 174 ASSEMBLY      0.023 SECONDS            3 REFERENCES
 DMPX.

  P    000000   A0   000200   B0   000000
  RA   133600   A1   000117   B1   000001     C(A1)=  1721  7000  0000  0000  0000     C(B1)=  0000  0000  0000  0000  0000
  FL   000200   A2   000060   B2   000002     C(A2)=  1505  1520  0000  0000  0061     C(B2)=  0000  0000  0000  0000  0000
  EM   700700   A3   000057   B3   012733     C(A3)=  0000  0000  0000  0000  0000     C(B3)=
  RE   000045   A4   000001   B4   000201     C(A4)=  0000  0000  0000  0000  0000     C(B4)=
  FE   000000   A5   000111   B5   000111     C(A5)=  5110  0001  1610  0114  6000     C(B5)=  5110  0001  1610  0114  6000
  MA   000600   A6   000001   B6   000200     C(A6)=  0000  0000  0000  0000  0000     C(B6)=
                A7   000001   B7   027756     C(A7)=  0000  0000  0000  0000  0000     C(B7)=
  X0   0000  0000  0000  0000  0000
  X1   1721  7000  0000  0000  0000
  X2   1721  0000  0000  0000  0000
  X3   1721  0000  0000  0000  0000
  X4   1717  0000  0000  0000  0000
  X5   1777  0000  0000  0000  0000
  X6   1777  0000  0000  0000  0000
  X7   1777  0000  0000  0000  0000

      00000    00040 00114 00000 00000         00000 00000 00000 00000
[redacted]
```

```
TESTFPO                                                                   COMPASS 3.5-470.     06/27/78  08.51.01.      PAGE     1
STORAGE ALLOCATION.

          ADDRESS   LENGTH                    BINARY CONTROL CARDS.

                0        7                    IDENT   TESTFPO
                7                             END     TESTFPO     TRANSFER ADDRESS

                                    BLOCKS     TYPE       ADDRESS     LENGTH

                                    PROGRAM*  LOCAL            0          5
                                    LITERALS* LOCAL            5          2

                                    ENTRY POINTS.

                                    TESTFPO           0+

                                    EXTERNAL SYMBOLS.

                                    SYS=
```

```
LOAD MAP - TESTFPO                                          CYBER LOADER 1.4-470      06/27/78  08.51.02.      PAGE     1


FWA OF THE LOAD              111
LWA+1 OF THE LOAD            160

TRANSFER ADDRESS -- TESTFPO          111

PROGRAM ENTRY POINTS --       TESTFPO          111


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK         ADDRESS   LENGTH    FILE      DATE     PROCSSR VER LEVEL  HARDWA

TESTFPO          111 *       7    LGO      06/27/78 COMPASS 3.5 470
SYS.RM           120        40 SL-SYSLIB   05/16/78 COMPASS 3.5 470


     .026 CP SECONDS              13200B CM STORAGE USED
```

```
  MFF    NR2-  CYB175-SN1    4L87/R68  05/15/78
08.50.58.DON004L  FROM    /SH
08.50.58.IP  00000256 WORDS - FILE INPUT  , DC 04
08.50.58.DON.      PSO,0278,72CT011A,MILLER
08.51.00.COMPASS.
08.51.01. ASSEMBLY COMPLETE.   47300B SCM USED.
08.51.01.     0.069  CPU SECONDS  ASSEMBLY TIME.
08.51.01.LGO.
08.51.02.ERROR MODE =04.  ADDRESS =000114
08.51.02.OP  00001152 WORDS - FILE OUTPUT , DC 40
08.51.02.MS     3584  WORDS  (    10752 MAX USED)
08.51.02.CPA           .122 SEC.            .122 ADJ.
08.51.02.IO            .676 SEC.            .676 ADJ.
08.51.02.CM          10.046 KWS.            .613 ADJ.
08.51.02.SS                                1.411
08.51.02.PP           3.797 SEC.      DATE 06/27/78
08.51.02.EJ  END OF JOB, SH
```

```
ROUND                                                          COMPASS 3.6-476.     01/04/79  14.43.08.      PAGE     1

                                          IDENT  ROUND
                                          ENTRY  ROUND
                                          LIST   S
                                 *
                                 *        EXAMPLE SHOWING A ROUND NORMALIZE INSTRUCTION
                                 *
     0  76100                    ROUND    SX1    B0
             10611                        BX6    X1
                  5160000006 +            SA6    ANSWER
     1  27201                             PX2    X1
             10722                        BX7    X2
                  5170000007 +            SA7    ANSWER+1
     2  25302                             ZX3    X2
             10633                        BX6    X3
                  5160000010 +            SA6    ANSWER+2
     3  5110000012 +                      SA1    OUT
                  10711                   BX7    X1
     4  5170000001                        SA7    1
                  5120000001              SA2    1
     5  0312000004 +                      NZ     X2,*-1
     6                     4     ANSWER   BSSZ   4
    12  05160420                 OUT      VFD    24/4LENDP   TAKE THE PLACE OF ENDRUN MACRO
    13                                    END    ROUND

                     47500B CM  STORAGE USED               23 STATEMENTS         3 SYMBOLS
                                MODEL 74 ASSEMBLY       0.070 SECONDS            0 REFERENCES
```

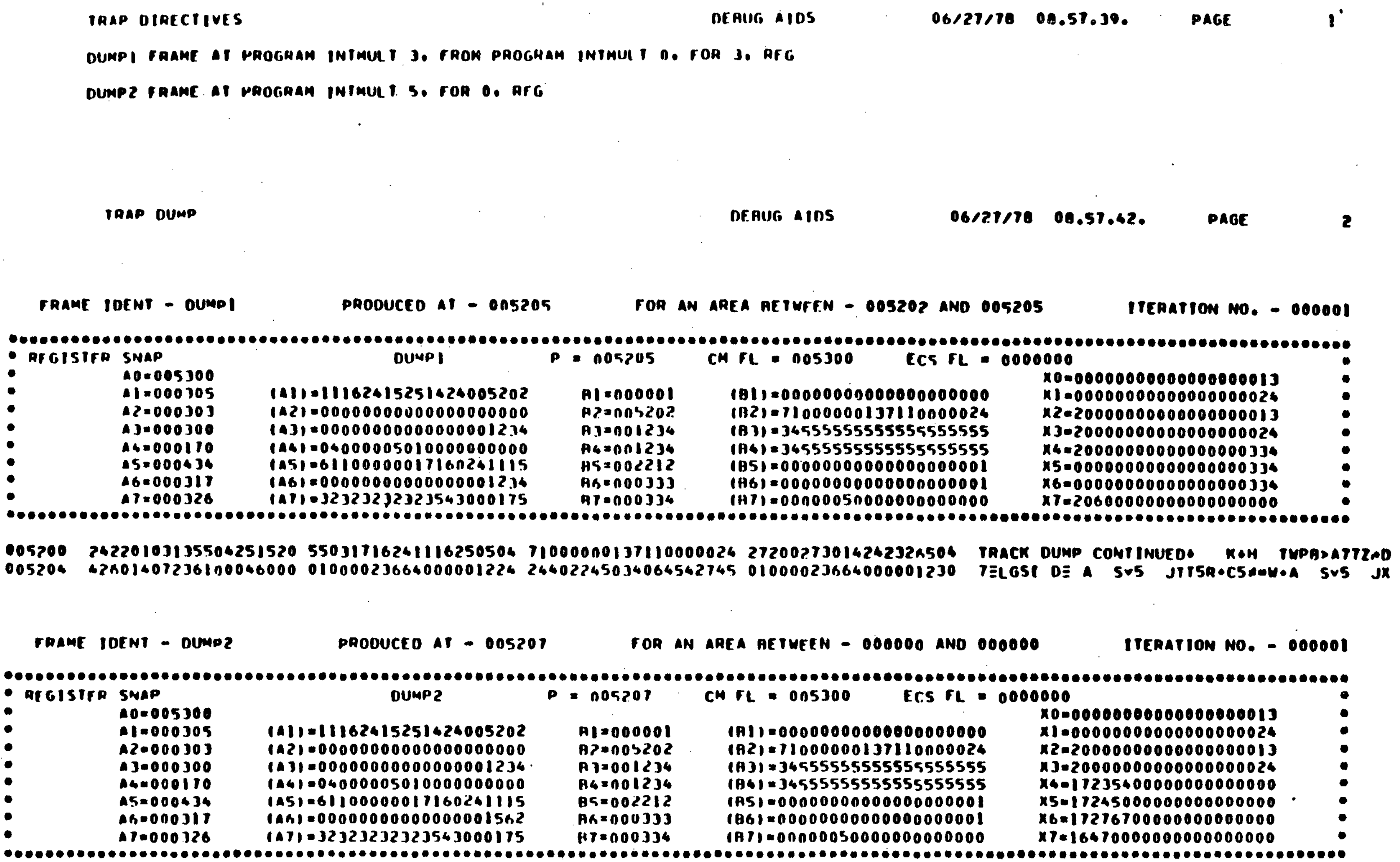
```
DUMP    RELATIVE                       DMP(0,127)

    00000   00000 00000 00000 00000   00002_33000 00000 00000 00001          34354 20000 00000 00017
    00004   00000 00000 00000 00000   00054_56110 03110 00054 54710          51100 00001 03110 00055         64550 02550 00000 46000
    00057   00000 00000 00000 00000
    00060   15051 52000 00000 00061         00000 00200 00000 00001          07040 00060 51600 00001         04000 00063 00000 00000
    00064   04152 00000 00000 00002         00000 00000 00000 00124          40000 00000 02000 00111         00000 00000 40000 00000
    00070   14071 75700 00000 00000         00000 00000 00000 00000    00100_54000 00000 01000 00001         00001 30000 00000 00124
    00102   00000 00000 00000 00000   00104_00000 00000 00000 00124          00000 00000 00000 00000   00110_22172 51604 00000 00111
    00111   76100 10611 51600 00117         27201 10722 51700 00120          25302 10633 51600 00121
    00114   51100 00123 10711 46000         51700 00001 51200 00001          03120 00115 61000 46000         00000 00000 00000 00000
    00120   20000 00000 00000 00000         17174 00000 00000 00000          00000 00000 00000 00000         05160 42000 61000 46000
    00124   60000 00000 04004 00124   00127>60000 00000 04004 00127
```

```
LOAD MAP - ROUND                                             CYBER LOADER 1.4-485      01/04/79  14.43.09.     PAGE     1

FWA OF THE LOAD            111
LWA+1 OF THE LOAD          124

TRANSFER ADDRESS -- ROUND          111

PROGRAM ENTRY POINTS --      ROUND          111


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS   LENGTH    FILE     DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

ROUND            111       13    LGO      01/04/79 COMPASS 3.6 476


     .023 CP SECONDS              13500B CM STORAGE USED                          1 TABLE MOVE
```

```
  MFS  NBI-  CYB74-SN108    5C/R0B    11/14/78
14.34.42.DON00G3  FROM    /OH
14.34.42.IP  00000256 WORDS - FILE INPUT  , DC 04
14.34.42.DON,T5.   001A,6883,1896,MILLER
14.43.07.COMPASS,LO=BR.
14.43.08. ASSEMBLY COMPLETE.   47500B  CM USED.
14.43.08.     0.200  CPU SECONDS  ASSEMBLY TIME.
14.43.08.LGO.
14.43.09.DMP(0,127)
14.43.09.OP  00000640 WORDS - FILE OUTPUT , DC 40
14.43.09.MS    3584  WORDS  (   10752 MAX USED)
14.43.09.CPA         .287 SEC.           .287 ADJ.
14.43.09.CPB         .061 SEC.           .061 ADJ.
14.43.09.IO          .310 SEC.           .310 ADJ.
14.43.09.CM        11.069 KWS.           .675 ADJ.
14.43.09.SS                             1.334
14.43.09.PP         5.030 SEC.     DATE 01/04/79
14.43.10.EJ  END OF JOB, OH
```

```
INTMULT                                                                  COMPASS 3.5-470.     06/27/78  08.57.39.      PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH               BINARY CONTROL CARDS.

                0       10               IDENT   INTMULT
               10                        END     INTMULT


                              ENTRY POINTS.

                              INTMULT          0*


                              EXTERNAL SYMBOLS.

                              SYS=
```

```
INTMULT                                                                  COMPASS 3.5-470.     06/27/78  08.57.39.      PAGE      2


                                         IDENT   INTMULT
                                         ENTRY   INTMULT
                                         LIST    *
                              *
                              *
                              *          USING DOUBLE PRECISION INSTRUCTION TO MULTIPLY INTEGERS
                              *
                              *
    0  7100000013              INTMULT   SX0     13B           PLACE 13 OCTAL IN X0
                 7110000024              SX1     24B    .      PLACE 24 OCTAL IN X1
    1  27200                             PX2     X0            CONVERT 13 OCTAL TO A PACKED FLOATING POINT NUMBER
           27301                         PX3     X1            CONVERT 24 OCTAL TO A PACKED FLOATING POINT NUMBER
                 42423                   DX4     X2*X3         DOUBLE PRECISION MULTIPLY -
                                                               LEAST SIGNIFICANT DIGITS IN X4
                       26504             UX5     X4            UNPACK X4 CONTENTS INTO X5 -
                                                                SHIFT COUNT GOING TO B0
    2  42601                             DX6     X0*X1         DOUBLE PRECISION MULTIPLY OF 13B AND 24B
           40723                         FX7     X2*X3         SINGLE PRECISION MULTIPLY OF X2 AND X3 CONTENTS
    3  46000                   TRKONE    NO
    4  24402                   *         NX4     X2
           24503                         NX5     X3
                 40645                   FX6     X4*X5
                       42745             DX7     X4*X5
    5  46000                   TRKTWO    NO
    6                          *         ENDRUN
    6  7160247021               SX6 3REND*4+1                                                      ENDRUN  .1
                 20650          LX6 40D                                                            ENDRUN  .1
    7  0100000000 X             RJ =XSYS=                                                          ENDRUN  .1
                                ENDM                                                               ENDRUN  .1

                              DEFAULT SYMBOLS DEFINED BY COMPASS.

    0 X                       SYS=

   10                                    END     INTMULT

                     47300B SCM STORAGE USED             30 STATEMENTS          4 SYMBOLS
                            MODEL 174 ASSEMBLY           0.035 SECONDS          5 REFERENCES
```



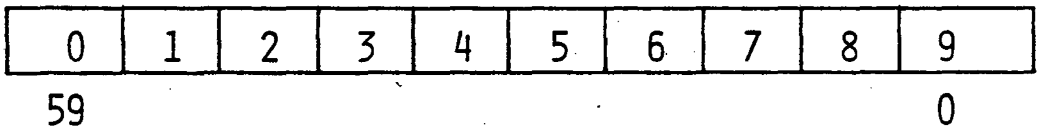
```
TRAP DIRECTIVES                                               DEBUG AIDS           06/27/78  08.57.39.      PAGE           1

DUMP1 FRAME AT PROGRAM INTMULT 3, FROM PROGRAM INTMULT 0, FOR 3, REG

DUMP2 FRAME AT PROGRAM INTMULT 5, FOR 0, REG




 TRAP DUMP                                                     DEBUG AIDS           06/27/78  08.57.42.      PAGE           2



FRAME IDENT - DUMP1            PRODUCED AT - 005205        FOR AN AREA BETWEEN - 005202 AND 005205        ITERATION NO. - 000001

•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
• REGISTER SNAP                          DUMP1           P = 005205     CM FL = 005300     ECS FL = 0000000               •
•              A0=005300                                                                  X0=00000000000000000013         •
•              A1=000305     (A1)=11162415251424005202      B1=000001     (B1)=00000000000000000000     X1=00000000000000000024         •
•              A2=000303     (A2)=00000000000000000000      B2=005202     (B2)=71000000137110000024     X2=20000000000000000013         •
•              A3=000300     (A3)=00000000000000001234      B3=001234     (B3)=34555555555555555555     X3=20000000000000000024         •
•              A4=000170     (A4)=04000005010000000000      B4=001234     (B4)=34555555555555555555     X4=20000000000000000334         •
•              A5=000434     (A5)=61100000017160241115      B5=002212     (B5)=00000000000000000001     X5=00000000000000000334         •
•              A6=000317     (A6)=00000000000000001234      B6=000333     (B6)=00000000000000000001     X6=00000000000000000334         •
•              A7=000326     (A7)=32323232323543000175      B7=000334     (B7)=00000050000000000000     X7=20600000000000000000         •
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••

005200  24220103135504251520 55031716241116250504 71000000137110000024 27200273014242326504  TRACK DUMP CONTINUED•   K•H  TWPB>ATTZ∧D
005204  42601407236100046000 01000023664000001224 24402245034064542745 01000023664000001230  7≡LGS( D≡ A  S∨5  JTTSR•C5≠=W•A  S∨5  JX


FRAME IDENT - DUMP2            PRODUCED AT - 005207        FOR AN AREA BETWEEN - 000000 AND 000000        ITERATION NO. - 000001

•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
• REGISTER SNAP                          DUMP2           P = 005207     CM FL = 005300     ECS FL = 0000000               •
•              A0=005300                                                                  X0=00000000000000000013         •
•              A1=000305     (A1)=11162415251424005202      B1=000001     (B1)=00000000000000000000     X1=00000000000000000024         •
•              A2=000303     (A2)=00000000000000000000      B2=005202     (B2)=71000000137110000024     X2=20000000000000000013         •
•              A3=000300     (A3)=00000000000000001234      B3=001234     (B3)=34555555555555555555     X3=20000000000000000024         •
•              A4=000170     (A4)=04000005010000000000      B4=001234     (B4)=34555555555555555555     X4=17235400000000000000         •
•              A5=000434     (A5)=61100000017160241115      B5=002212     (B5)=00000000000000000001     X5=17245000000000000000         •
•              A6=000317     (A6)=00000000000000001562      B6=000333     (B6)=00000000000000000001     X6=17276700000000000000         •
•              A7=000326     (A7)=32323232323543000175      B7=000334     (B7)=00000050000000000000     X7=16470000000000000000         •
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

```
INTMULT                                                             COMPASS 3.5-470.     06/27/78  08.57.39.      PAGE      3
SYMBOLIC REFERENCE TABLE.

INTMULT        0      PROGRAM*        2/02 I     2/03 L
SYS=           0      EXTERNAL*       2/28
TRPONE         3      PROGRAM*        2/19 L
TRPTWO         5      PROGRAM*        2/24 L


 LOAD MAP - TRAPPER                                             CYBER LOADER 1.4-470       06/27/78  08.57.42.      PAGE      1



 FWA OF THE LOAD              111
 LWA+1 OF THE LOAD           5252

 TRANSFER ADDRESS -- INTMULT          5202
 DEBUG ENTRY USED -- TRPSETR           434

 PROGRAM ENTRY POINTS --       TRAPPER          5202



 PROGRAM AND BLOCK ASSIGNMENTS.

 BLOCK        ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

 TRAPPER          111     5071 SL-SYSLIB   05/16/78 COMPASS 3.5 470
 INTMULT         5202       10    LGO      06/27/78 COMPASS 3.5 470
 SYS.RM          5212       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                   PROCESS SYSTEM REQUEST.


      .056 CP SECONDS            17600B CM STORAGE USED                      1 TABLE MOVE

                                   MFF   NR2-  CYB175-SN1    4L87/R68  05/15/78
                                  08.57.36.DON0049  FROM    /SH
                                  08.57.36.IP  00000320 WORDS - FILE INPUT  , DC 04
                                  08.57.36.DON.     PSD,0278,72CT011A,MILLER
                                  08.57.39.TRAP(L=PUTOUT)
                                  08.57.39.COMPASS.
                                  08.57.40. ASSEMBLY COMPLETE.   47300B SCM USED.
                                  08.57.40.    0.080  CPU SECONDS  ASSEMBLY TIME.
                                  08.57.40.LGO.
                                  08.57.42.REWIND(PUTOUT)
                                  08.57.42.COPYBF,PUTOUT.
                                  08.57.43.EOI ENCOUNTERED AFTER COPY OF FILE
                                  08.57.43.          0, RECORD  3
                                  08.57.43.OP  00001280 WORDS - FILE OUTPUT , DC 40
                                  08.57.43.MS    3584  WORDS  (    17920 MAX USED)
                                  08.57.43.CPA        .211 SEC.          .211 ADJ.
                                  08.57.43.IO         .763 SEC.          .763 ADJ.
                                  08.57.43.CM      12.083 KWS.           .737 ADJ.
                                  08.57.43.SS                           1.712
                                  08.57.43.PP        5.423 SEC.     DATE 06/27/78
                                  08.57.43.EJ  END OF JOB, SH
```

COMPARE MOVE UNIT (CMU)

THE COMPARE MOVE UNIT IS A STANDARD CPU HARDWARE COMPONENT OF THE CYBER 70 SERIES MODEL 72 AND 73; MODELS 172, 173, 174; AND OPTIONAL ON THE MODEL 76 COMPUTER SYSTEM. IT PROVIDES THE CAPABILITY TO MOVE AND COMPARE DATA FIELDS IN STORAGE WITHOUT HAVING TO USE THE REGISTERS.

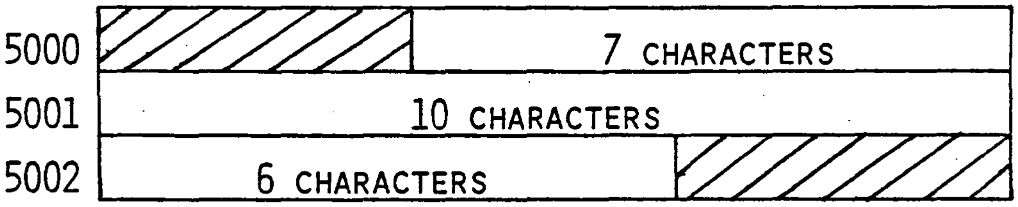
THERE ARE TEN 6-BIT CHARACTER POSITIONS IN EACH 60-BIT WORD. THESE POSITIONS ARE NUMBERED 0 THROUGH 9 FROM LEFT TO RIGHT RESPECTIVELY. THE 4-BIT CHARACTER ADDRESSES OF THESE POSITIONS ARE (IN BINARY) 0000, 0001, ..., 1000, AND 1001. CHARACTER ADDRESSES 1010 THROUGH 1111 ARE ILLEGAL AND CAUSE THE INSTRUCTIONS TO GIVE AN ADDRESS OUT OF RANGE CONDITION.



STORAGE WORD

DATA FIELDS MAY SPAN WORD BOUNDARIES AND MAY START OR END AT ANY POSITION IN A 60-BIT WORD.

EXAMPLE:



THE FIELD ABOVE STARTS AT CHARACTER 3 IN WORD 50000 AND ENDS WITH CHARACTER 5 IN WORD 50002. THE FIELD HAS A LENGTH OF 23_{10} CHARACTERS. ONE LIMITATION FOR

USING THE CMU IS THAT THE DATA FIELD MUST NOT BE IN AN OPERATING REGISTER OR IN ECS/LCM.

COMPASS 3.X PROVIDES SYMBOLIC FORMS OF FOUR CMU INSTRUCTIONS. THEY ARE:

1) INDIRECT MOVE - IM
2) DIRECT MOVE - DM
3) COMPARE COLLATED - CC
4) COMPARE UNCOLLATED - CU

OF THE ABOVE, ONLY THE INDIRECT MOVE (IM) INSTRUCTION HAS THE SAME TYPE OF SYNTAX AND SOMANTICS AS OTHER CPU INSTRUCTIONS. THE OTHERS ARE TRCATED AS PSEUDO INSTRUCTIONS BY COMPASS.

Indirect Move (IM)

This is a 30-bit instruction that moves the content of a data field to another data field according to a descriptor word. Maximum length of the data field that could be moved by this instruction is 8191_{10} characters. The descriptor word contains the length and addresses of the data fields. COMPASS forces the instruction to the upper left of a word because it is executed as a pass by the hardware if it is not the first instruction of a word. The next instruction is also forced upper in the next word, because the lower half of a word containing an indirect move i not executed.

Format:

Operation	Variable	Description	Size	Octal Code
IM	Bj	Move per descriptor at Bj	30 bits	464j000000
IM	K	Move per descriptor at K	30 bits	4640K
IM	Bj±K	Move per descriptor at Bj±K	30	464jK

Execution: The descriptor word is fetched from storage location (Bj)±K. If the data field length is zero, the instruction is executed as a pass but the execution time is longer. Otherwise, the content of the source field is moved to the destination field. If the two fields overlap, the results are undefined. The X0 register is used for intermediate storage during execution of the instruction and is cleared upon completion of the instruction.

A pseudo instruction MD is used to generate a descriptor word for use by the indirect move instruction. The MD instruction has the following format:

LOCATION	OPERATION	VARIABLES
LOCSYM	MD	L, X_S, C_S, K_D, C_D

L IS THE ABSOLUTE ADDRESS EXPRESSION; ITS VALUE, IN THE RANGE$\leq 0 <$L 8191, IS THE DATA FIELD LENGTH IN CHARACTERS. THE UPPER 9 BITS ARE PLACED IN BITS 56-48 OF THE DESCRIPTOR WORD WHILE THE LOWER 4 BITS ARE PLACED IN BITS 29-26.

K_S IS ANY EXPRESSION, THE FIRST WORD ADDRESS OF THE SOURCE FIELD.

C_S IS THE ABSOLUTE EXPRESSION, THE STARTING CHARACTER POSITION OF THE SOURCE FIELD WITHIN THE WORD AT LOCATION K_S.

K_D IS ANY EXPRESSION, THE FIRST WORD ADDRESS OF THE DESTINATION FIELD.

C_D IS THE ABSOLUTE EXPRESSION, THE STARTING CHARACTER POSITION OF THE DESTINATION FIELD WITHIN THE WORD AT LOCATION K_D.

INDIRECT MOVE DESCRIPTOR WORD FORMAT:

59	56	47	28	25	21	17	0
0	L_U	K_S	L_L	C_S	C_D	K_D	

WHERE:

L_U: UPPER 9 BITS OF VALUE OF L.

L_L: LOWER 4 BITS OF VALUE OF L.

DIRECT MOVE (DM)

THE DIRECT MOVE PSEUDO INSTRUCTION GENERATES A CMU INSTRUCTION THAT MOVES A DATA FIELD IN STORAGE TO ANOTHER LOCATION IN STORAGE. THIS INSTRUCTION DIFFERS FROM THE INDIRECT MOVE IN SEVERAL WAYS. IT IS A 60-BIT INSTRUCTION THAT CANNOT BE SPLIT BETWEEN WORDS AND THE DESCRIPTOR WORD IS PART OF THE INSTRUCTION. FURTHERMORE, THE LENGTH OF THE DATA FIELD IT CAN MOVE IS LIMITED TO A MAXIMUM OF 127_{10} CHARACTERS.

INSTRUCTION FORMAT:

LOCATION	OPERATIONS	VARIABLE
LOCSYM	DM	L, K_S, C_S, K_D, C_D

59	50	47	29	25	21	17	0
465	L_U	K_S	L_L	C_S	C_D	K_D	

L IS THE ABSOLUTE ADDRESS EXPRESSION; ITS VALUE, IN THE RANGE $0 \le L \le 127$, IS THE DATA FIELD LENGTH IN CHARACTERS.

L_U IS THE UPPER 3 BITS OF THE VALUE OF L.

L_L, K_S, C_S, K_D, C_D: SAME AS IN THE MD INSTRUCTION.

EXECUTION: SAME AS IM, EXCEPT THAT THE DESCRIPTOR IS IN THE INSTRUCTION WORD ITSELF.

COMPARE COLLATED (CC)

THE COMPARE COLLATED INSTRUCTION COMPARES THE CONTENTS OF TWO DATA FIELDS, ONE CHARACTER AT A TIME, FROM LEFT TO RIGHT, UNTIL A PAIR OF CORRESPONDING CHARACTERS ARE FOUND TO HAVE UNEQUAL COLLATING VALUES, OR UNTIL THE DATA FIELDS ARE EXHAUSTED. IT IS A 60-BIT INSTRUCTION THAT OCCUPIES ONE FULL WORD (IT CANNOT BE SPLIT BETWEEN TWO WORDS) AND CONTAINS ITS OWN DATA FIELD DESCRIPTOR.

IT USES REGISTER A0 TO CONTAIN THE FIRST WORD ADDRESS OF A TABLE IN STORAGE THAT CONTAINS THE COLLATING VALUES TO BE USED IN COMPARING CHARACTERS. THE RESULT OF THE COMPARISON IS PLACED IN REGISTER X0.

FORMAT:

LOCATION	OPERATION	VARIABLES
LOCASYM	CC	L, K_A, C_A, K_B, C_B

59	50	47	29	25	21	17	0
466	L_U	K_A	L_L	C_A	C_B	K_B	

L, L_U, L_L ARE SAME AS IN THE DM INSTRUCTION.

K_A IS ANY EXPRESSION, THE FIRST WORD ADDRESS OF THE FIRST DATA FIELD.

C_A IS THE ABSOLUTE EXPRESSION, THE STARTING CHARACTER POSITION OF THE FIRST DATA FIELD WITHIN THE WORD AT LOCATION K_A.

K_B IS ANY EXPRESSION, THE FIRST WORD ADDRESS OF THE SECOND DATA FIELD.

C_B IS THE ONLY ABSOLUTE EXPRESSION, THE STARTING CHARACTER POSITION OF THE SECOND DATA FIELD WITHIN THE WORD AT LOCATION K_B.

EXECUTION: THE FIRST WORD ADDRESS OF THE COLLATING TABLE IS OBTAINED FROM REGISTER A0. THE CONTENTS OF THE DATA FIELDS ARE COMPARED FROM LEFT TO RIGHT, ONE CHARACTER AT A TIME FROM EACH FIELD, UNTIL TWO UNEQUAL CHARACTERS ARE FOUND. THE COLLATING VALUE OF EACH CHARACTER IS OBTAINED FROM THE COLLATING TABLE. IF THESE VALUES ARE EQUAL, THE COMPARE CONTINUES UNTIL ANOTHER CHARACTER PAIR IS UNEQUAL OR UNTIL ALL CHARACTERS HAVE BEEN COMPARED. IF THE COLLATING VALUES ARE UNEQUAL, THE TWO DATA FIELDS ARE UNEQUAL AND THE FILED WITH A LARGER COLLATING VALUE IS THE GREATER OF THE TWO FIELDS. THE COLLATING VALUES ARE TREATED AS 6-BIT UNSIGNED INTEGERS.

NOTE THAT TWO UNEQUAL CHARACTERS COULD HAVE THE SAME COLLATING VALUE AND WOULD COMPARE EQUAL. UPON COMPLETION, REGISTER X0 CONTAINS A 60-BIT SIGNED INTEGER AS FOLLOWS:

$X0 = L - N > 0$ IF FIELD A > FIELD B

$X0 = +0$ IF FIELD A = FIELD B

$X0 = N - L < 0$ IF FIELD A < FIELD B

WHERE N IS THE NUMBER OF PAIRS OF CHARACTERS THAT COMPARED EQUAL.

IF $L=0$, THEN $X0=+0$.

THE FORMAT OF THE COLLATING TABLE IS AS FOLLOWS:

	59	53	47	41	35	29	23	17	11 0
(A0)	00	01	02	03	04	05	06	07	
(A0) +1	10	11	12	13	14	15	16	17	
⋮									
(A0) +7	70	71	72	73	74	75	76	77	

COMPARE UNCOLLATED (CU)

THE COMPARE UNCOLLATED INSTRUCTION COMPARES THE CONTENTS OF TWO DATA FIELDS, ONE CHARACTER AT A TIME, FROM LEFT TO RIGHT, UNTIL A PAIR OF CORRESPONDING CHARACTERS ARE FOUND TO HAVE UNEQUAL VALUES, OR UNTIL THE DATA FIELDS ARE EXHAUSTED. IT IS A 60-BIT INSTRUCTION THAT OCCUPIES ONE FULL WORD (IT CANNOT BE SPLIT BETWEEN TWO WORDS) AND CONTAINS ITS OWN DATA FIELD DESCRIPTOR. THE RESULT OF THE COMPARISON IS PLACED IN REGISTER X0.

FORMAT:

LOCATION	OPERATION	VARIABLE
LOCASYM	CU	L, K_A, C_A, K_B, C_B

59	50	47	28	25	21	17 0
467	L_U	K_A	L_L	C_A	C_B	K_B

EXECUTION: SAME AS THE CC INSTRUCTION EXCEPT THAT A0 AND THE COLLATING TABLE ARE NOT USED. INSTEAD, THE CHARACTERS ARE COMPARED DIRECTLY WITH EACH CHARACTER REGARDED AS A 6-BIT UNSIGNED BINARY INTEGER.

REGISTER X0 IS SET IN THE SAME MANNER AS BY THE CC INSTRUCTION.

Lesson 6

Lesson Preview:

In this lesson several important definition and special symbols that are an integral part of COMPASS are discussed. The pseudo-ops EQU, SET, CON, and VFD are introduced.

References:

Chapter 4	COMPASS Reference Manual #60492600

Training Aids:

Visual set V6

Projects:

Homework

Objectives:

At the completion of this lesson the student will be able to:

1. Define names and symbols and give examples of each.
2. Denote the differences between data and constants.
3. Define expressions, give examples and evaluate them.
4. Identify absolute, relocatable and evaluatable expressions.
5. Use data generation pseudo-ops correctly.

```
                                          IDENT  NUMDATA
                                          LIST   D
                                  ***********************************************************
                                  *                                                         *
                                  *                                                         *
                                  *       TEST DECK OF NUMERIC DATA GENERATED               *
                                  *                                                         *
                                  *                                                         *
                                  ***********************************************************
                                  *
     0  00000000000000000007              DATA   7            POSITIVE INTEGER
     1  77777777777777777766              DATA   -9           NEGATIVE INTEGER
     2  00000000000000000010              DATA   10B          POST RADIX
     3  00000000000000000010              DATA   B10          PRE RADIX
     4  17224000000000000000              DATA   4.0          FLOATING POINT NUMBER - SINGLE PRECISION
     5  17224000000000000000              DATA   4.EE         FLOATING POINT NUMBER - DOUBLE PRECISION
     6  16420000000000000000
     7  60553777777777777777              DATA   -4.0         NEGATIVE FLOATING POINT NUMBER
    10  17104061115645706520              DATA   0.004        FLOATING POINT DECIMAL
    11  17224400000000000000              DATA   4.5          FLOATING POINT  FRACTION = 9/2 = 11/2B
    12  12345671234567444444              DATA   12345671234567444444B  INTEGER NUMBER
    13  17224077153337337741              DATA   4.1234567123456712EE  DOUBLE PRECISION FLOATING POINT NUMBER
    14  16427752426164147264
                                  *
                                  ***********************************************************
                                  *                                                         *
                                  *                                                         *
                                  *       USE E FOR POWERS OF 10                            *
                                  *                                                         *
                                  *                                                         *
                                  ***********************************************************
                                  *
    15  00000000000000116100              DATA   4E4          40000 INTEGER
    16  00000000000000116100              DATA   4EE4         COMPARES TO 4E4
    17  17374704000000000000              DATA   4.E4         40000 FLOATING POINT SINGLE PRECISION
    20  17374704000000000000              DATA   4.EE4        40000 FLOATING POINT DOUBLE PRECISION
    21  16570000000000000000
                                  *
                                  ***********************************************************
                                  *                                                         *
                                  *                                                         *
                                  *       USE S FOR BINARY SHIFTING IN WORDS                *
                                  *                                                         *
                                  *                                                         *
                                  ***********************************************************
                                  *
    22  00000000000000000024              DATA   10S1         001 010 0 (BINARY=12B*2 = 24B)
    23  00000000000000000240              DATA   12BS4        001 010 000 0 (BINARY = 12B*16B = 240B)
                                  *
                                  ***********************************************************
                                  *                                                         *
                                  *                                                         *
                                  *       USE P SCALING FOR UNNORMALIZED NUMBERS            *
                                  *                                                         *
                                  *                                                         *
                                  ***********************************************************
                                  *
```

```
NUMDATA                                                     COMPASS 3.5-470.     06/28/78  07.34.59.      PAGE      3

     24  17255000000000000000                 DATA    40.0        NORMALIZED FLOATING POINT NUMBER
     25  20000000000000000050                 DATA    40.P0       UN-NORMALIZED FLOATING POINT NUMBER (POSITIONED)
     26  20030000000000000005                 DATA    40.P-3      SHIFTED RIGHT
     27  17740000000000000500                 DATA    40.P3       SHIFTED LEFT
     30  00000000000000000000                 DATA    0.04P0      NO SIGNIFICANT BITS IN REGISTER
     31  17710000000000000002                 DATA    0.04P6      LEFT SHIFT
                                   *
                                   ***********************************************************************************
                                   *                                                                                 *
                                   *                                                                                 *
                                   *          EVALUATION ORDER IS F THEN S THEN P                                    *
                                   *                                                                                 *
                                   *                                                                                 *
                                   ***********************************************************************************
                                   *
     32  17760000000000000002                 DATA     1.2P1S-5E1
     33  60017777777777777775                 DATA    -1.2P1S-5E1
     34  00000000000000000000                 DATA     DEES
     35  00000000000000000000
     36                                       END

                        47300B SCM STORAGE USED              71 STATEMENTS          0 SYMBOLS
                               MODEL 174 ASSEMBLY        0.066 SECONDS              0 REFERENCES

                                      1 ERROR IN  NUMDATA


NUMDATA                                                     COMPASS 3.5-470.     06/28/78  07.34.59.      PAGE      4
ERROR DIRECTORY.

      A TYPE ERROR                 ADDRESS FIELD BAD.
           OCCURRED ON PAGES          3


                                                 MFF   NB2-  CYB175-SN1    4LB7/R6B  05/15/78
                                               07.34.57.DON001W  FROM    /SH
                                               07.34.57.IP  00000640 WORDS - FILE INPUT  , DC 04
                                               07.34.57.DON.     PSC,027B,72CT011A.MILLER
                                               07.34.59.COMPASS.
                                               07.34.59.           1 ERROR IN  NUMDATA
                                               07.34.59. ASSEMBLY ERRORS.     47300B SCM USED.
                                               07.34.59.    0.116  CPU SECONDS  ASSEMBLY TIME.
                                               07.34.59.OP  00001088 WORDS - FILE OUTPUT , DC 40
                                               07.34.59.MS    3584  WORDS  (   10752 MAX USED)
                                               07.34.59.CPA        .143 SEC.         .143 ADJ.
                                               07.34.59.IO         .488 SEC.         .488 ADJ.
                                               07.34.59.CM       9.359 KWS.          .571 ADJ.
                                               07.34.59.SS                          1.202
                                               07.34.59.PP        2.196 SEC.     DATE 06/28/78
                                               07.34.59.EJ  END OF JOB, SH
```

```
DATA                                                                        COMPASS 3.5-470.     06/27/78  08.49.25.      PAGE      2

                                                       IDENT  DATA
                                                       ENTRY  DATA
                                                       LIST   S
                                               *
                                               *       USE OF DATA PSEUDO OP ******CHARACTER DATA******
                                               *
     0  01020304050600000000        TYPEC      DATA   6CABCDEF    LEFT JUSTIFIED WITH ZERO FILL.  FOR DATA ITEMS OR
     1  01020304050607100000                   DATA   9CABCDEFGHI LITERALS. 12 ZERO BITS ARE GUARANTEED AT THE END OF
     2  00000000000000000000
                                               *                  THE STRING EVEN IF ANOTHER WORD MUST BE ALLOCATED.
                                                                  FOR A CONSTANT. THE ZERO BITS ARE NOT GUARANTEED!
                                                                  C IS THE SAME AS L FOR A CONSTANT
                                                       SPACE  2

     3  01020304050655555555        TYPEH      DATA   6HABCDEF    LEFT JUSTIFIED WITH BLANK FILL
     4  01020304050607101155                   DATA   9HABCDEFGHI LEFT JUSTIFIED WITH BLANK FILL
                                                       SPACE  2

     5  55555555010203040506        TYPEA      DATA   6AABCDEF    RIGHT JUSTIFIED WITH BLANK FILL
     6  55010203040506071011                   DATA   9AABCDEFGHI RIGHT JUSTIFIED WITH BLANK FILL
                                                       SPACE  2

     7  00000000010203040506        TYPER      DATA   6RABCDEF    RIGHT JUSTIFIED WITH ZERO FILL
    10  00010203040506071011                   DATA   9RABCDEFGHI RIGHT JUSTIFIED WITH ZERO FILL
                                                       SPACE  2

    11  01020304050600000000        TYPEL      DATA   6LABCDEF    LEFT JUSTIFIED WITH ZERO FILL
    12  01020304050607101100                   DATA   9LABCDEFGHI LEFT JUSTIFIED WITH ZERO FILL
                                                       SPACE  2

    13  01020304050600000000        TYPEZ      DATA   6ZABCDEF    LEFT JUSTIFIED WITH ZERO FILL.  FOR DATA ITEMS OR
    14  01020304050607101100                   DATA   9ZABCDEFGHI LITERALS. SIX ZERO BITS ARE GUARANTEED AT THE END
    15  01020304050607101112                   DATA 10ZABCDEFGHIJ OF THE STRING EVEN IF ANOTHER WORD MUST BE ALLO-
    16  00000000000000000000
                                                                  CATED.  FOR A CONSTANT. THE BITS ARE NOT GUARANTEED
    17  01020304050600000000                   DATA   +6CABCDEF   SIGN.N.TYPE.STRING
    20  76757473727177777777                   DATA   -6CABCDEF   SIGN.N.TYPE.STRING
    21  01020304050600000000                   DATA    6CABCDEF   SIGN.N.TYPE.STRING
                                                       SPACE  1

    22  04050601020355555555                   DATA   +H*DEFABC*  SIGN.TYPE.DELIMITER.STRING.DELIMITER
    23  73727176757422222222                   DATA   -H*DEFABC*  SIGN.TYPE.DELIMITER.STRING.DELIMITER
    24  04050601020355555555                   DATA    H*DEFABC*  SIGN.TYPE.DELIMITER.STRING.DELIMITER
                                                       SPACE  1

    25  5110010203                             SA1    3HABC       N.TYPE STRING
                                                       SPACE  1

    26  5110000067 +                DATA       SA1    =+4ADEFG    =.SIGN.N.TYPE.STRING
                  5110000070 +                 SA1    =-4ADEFG    =.SIGN.N.TYPE.STRING
    27  5110000067 +                           SA1    =04ADEFG    =.SIGN.N.TYPE.STRING
```

```
                                                 EJECT
                                         *
                                         *       USE OF DATA PSEUDO OP ******NUMERIC DATA******
                                         *

      30  0000000000000000000125                 DATA   85          VALUE
      31  7777777777777777777774                 DATA   -3          SIGN,VALUE
      32  17204000000000000000                   DATA   1.0         SIGN,VALUE
      33  17225000000000000000                   DATA   5.0         SIGN,VALUE
      34  17245046314631463146                   DATA   D20.3       SIGN,PRERADIX,VALUE
      35  17245046314631463146                   DATA   +D20.3      SIGN,PRERADIX,VALUE
      36  60532731463146314631                   DATA   -D20.3      SIGN,PRERADIX,VALUE
      37  17244060000000000000                   DATA    B20.3      SIGN,PRERADIX,VALUE
      40  60533717777777777777                   DATA   -B20.3      SIGN,PRERADIX,VALUE
      41  17244060000000000000                   DATA   +B20.3      SIGN,PRERADIX,VALUE
      42  17136314631463146315                   DATA   5.0E-2      SIGN,VALUE+MODIFIER
      43  17307640000000000000                   DATA   5.0E2       SIGN,VALUE+MODIFIER
      44  17307640000000000000                   DATA   5.0EE2      SIGN,VALUE+MODIFIER
      45  16500000000000000000
      46  17307640000000000000                   DATA   +5.0EE2     SIGN,VALUE+MODIFIER
      47  16500000000000000000
      50  60470137777777777777                   DATA   -5.0EE2     SIGN,VALUE+MODIFIER
      51  61277777777777777777
      52  20000000000000000001                   DATA   1.0P0       SIGN,VALUE+MODIFIER
      53  57777777777777777776                   DATA   -1.0P0      SIGN,VALUE+MODIFIER
      54  17740000000000000400                   DATA   3.2E1P3     SIGN,VALUE+MODIFIER
      55  17200400000000000000                   DATA   0.1BP47     SIGN,VALUE+MODIFIER

                                         *
                                         *       N IS OMITTED
                                         *
      56  55555555555515111605                   DATA   A*MINE*
      57                                         DATA   H**
      57  05222217225511165520                   DATA   L*ERROR IN PDQ*
      60  04210000000000000000
                                         *
                                         *       N=0
                                         *
      61  55555555555512171016                   DATA   0AJOHN
      62  12171016000000000000                   DATA   0LJOHN DOE  SPACE ACTS AS A DELIMITOR
      63                                         DATA   0L JOHN DOE SPACE ACTS AS A DELIMITOR
7     63  7110021714                             SX1    0RSYMBOL
7                   7120010203                   SX2    0HABCDEFGHIJ
      64  7130000060                             SX3    1R.+1
A     65  00000000000000000000                   DATA   1R.+1
      66  00000000000000000123          MINE     DATA   123B

                                         CONTENT OF LITERALS BLOCK.

      67  55555555555504050607                DEFG
      70  22222222222273727170           RRRRRR><<+

      71                                         END

                             47300B SCM STORAGE USED            84 STATEMENTS         8 SYMBOLS
                                        MODEL 174 ASSEMBLY    0.088 SECONDS         9 REFERENCES
```

```
DATA                                                          COMPASS 3.5-470.    06/27/78  08.49.25.      PAGE      1
STORAGE ALLOCATION.

              ADDRESS   LENGTH                BINARY CONTROL CARDS.

                    0       71                IDENT   DATA
                   71                         END

                                       BLOCKS      TYPE        ADDRESS    LENGTH

                                       PROGRAM*    LOCAL             0        67
                                       LITERALS*   LOCAL            67         2


                                       ENTRY POINTS.

                                       DATA                  26*
DATA                                                          COMPASS 3.5-470.    06/27/78  08.49.25.      PAGE      4


                                            1 ERROR IN  DATA


DATA                                                          COMPASS 3.5-470.    06/27/78  08.49.25.      PAGE      5
ERROR DIRECTORY.

     A TYPE ERROR                     ADDRESS FIELD BAD.
          OCCURRED ON PAGES                  3

     T TYPE ERROR                     ADDRESS VALUE EXCEEDS FIELD SIZE, RESULT TRUNCATED
          OCCURRED ON PAGES                  3




DATA                                                          COMPASS 3.5-470.    06/27/78  08.49.25.      PAGE      6
SYMBOLIC REFERENCE TABLE.

DATA           26      PROGRAM*        2/02 F     2/54 L
MINF           66      PROGRAM*        3/46 L
TYPEA           5      PROGRAM*        2/21 L
TYPEC           0      PROGRAM*        2/07 L
TYPEH           3      PROGRAM*        2/16 L
TYPEL          11      PROGRAM*        2/31 L
TYPER           7      PROGRAM*        2/26 L
TYPEZ          13      PROGRAM*        2/36 L
```

```
   MFF   NP2-  CY8175-SN1    4L97/RAH  05/15/78
 08.49.23.DON004P  FROM    /SH
 08.49.23.IP  00000576 WORDS - FILE INPUT  , DC 04
 08.49.23.DON.     PSD.0278.72CT0114.MILLER
 08.49.25.COMPASS(LO=$$$$)
 08.49.25.          2 WARNING MESSAGES IN DATA
 08.49.25.          1 ERROR IN  DATA
 08.49.26. ASSEMBLY ERRORS.     47300B SCM USED.
 08.49.26.    0.139  CPU SECONDS  ASSEMBLY TIME.
 08.49.26.OP  00001216 WORDS - FILE OUTPUT , DC 40
 08.49.26.MS    3584  WORDS  (    7168 MAX USED)
 08.49.26.CPA       .166 SEC.         .166 ADJ.
 08.49.26.IO        .491 SEC.         .491 ADJ.
 08.49.26.CM       9.933 KWS.         .606 ADJ.
 08.49.26.SS                         1.264
 08.49.26.PP       2.839 SEC.     DATE 06/27/78
 08.49.26.EJ  END OF JOB, SH
```

```
                                      IDENT  CON
                                      ENTRY  CON
                              *****************************************************************
                              *                                                               *
                              *       CON - GENERATE CONSTANTS                                *
                              *                                                               *
                              *       THE ≠CON≠ PSEUDO INSTRUCTION GENERATES ONE OR MORE FULL WORDS OF *
                              *       BINARY DATA IN THE BLOCK IN USE.  IT DIFFERS FROM DATA IN THAT IT *
                              *       GENERATES EXPRESSION VALUES RATHER THAN DATA ITEMS AND DIFFERS FROM *
                              *       VFD IN THAT THE FIELD SIZE IS FIXED.                    *
                              *                                                               *
                              *LOCATION OPERATION VARIABLE SUBFIELDS                          *
                              *                                                               *
                              *SYM      CON       EXP1,EXP2,.....,EXPN                        *
                              *                                                               *
                              *SYM    IF PRESENT, SYM IS ASSIGNED THE VALUE OF THE LOCATION COUNTER AFTER THE *
                              *       FORCE UPPER OCCURS.                                     *
                              *                                                               *
                              *EXP1   AN ABSOLUTE, RELOCATEABLE, OR EXTERNAL EXPRESSION THE VALUE OF WHICH WILL
                              *       BE INSERTED INTO A FIELD HAVING A SIZE OF ONE WORD.  FOR PPU ASSEMBLY, *
                              *       FLOATING POINT IS NOT ALLOWED. FOR CPU ASSEMBLY, DOUBLE PRECISION IS NOT*
                              *       ALLOWED.                                                *
                              *                                                               *
                              *****************************************************************
      0  00000000000000000000      CON      CON    0
      1  00000000000000000006               CON    6
      2  00000000000000000003               CON    3
      3  00000000000000000024               CON    20
      4  00000000000000000055               CON    1R
      5  00000000000000000033               CON    1R0
      6  00000000000000000064               CON    1R≠.
      7  00000000000000000033               CON    1R0,1L0,1H0,1A0
     10  33000000000000000000
     11  33555555555555555555
     12  55555555555555555533
     13  00000000000000000060               CON    1R.+1
A    14  00000000000000000000               DATA   1R.+1
     15  7160247021                         ENDRUN
     17                                     END    CON

              47300B SCM STORAGE USED               40 STATEMENTS         2 SYMBOLS
                     MODEL 174 ASSEMBLY         0.044 SECONDS             3 REFERENCES

                                     1 ERROR IN  CON
```

```
CON                                            COMPASS 3.5-470.     06/27/78  16.29.23.      PAGE      3
ERROR DIRECTORY.

     A TYPE ERROR              ADDRESS FIELD BAD.
          OCCURRED ON PAGES          2
```

```
CON                                                                COMPASS 3.5-470.     06/27/78  16.29.23.      PAGE      1
STORAGE ALLOCATION.

           ADDRESS   LENGTH              BINARY CONTROL CARDS.

                 0       17              IDENT   CON
                17                       END     CON

                                 ENTRY POINTS.

                                 CON                0*

                                 EXTERNAL SYMBOLS.

                                 SYS=
```

```
CON                                                                COMPASS 3.5-470.     06/27/78  16.29.23.      PAGE      4
SYMBOLIC REFERENCE TABLE.

CON              0      PROGRAM*          2/02 F     2/25 L
SYS=             0      EXTERNAL*         2/39
```

```
  MFF   NA2-  CYB175-SN1    4L87/R6B  05/15/78
16.29.20.DON00MU  FROM     /SH
16.29.20.IP  00000384 WORDS - FILE INPUT  , DC 04
16.29.20.DON.     PSC,0278,72CT011A,MILLER
16.29.22.COMPASS.
16.29.23.           1 ERROR IN  CON
16.29.23. ASSEMBLY ERRORS.     47300B SCM USED.
16.29.23.    0.093  CPU SECONDS  ASSEMBLY TIME.
16.29.23.OP  00000704 WORDS - FILE OUTPUT , DC 40
16.29.23.MS    3584  WORDS  (     7168 MAX USED)
16.29.23.CPA        .115 SEC.          .115 ADJ.
16.29.23.IO         .478 SEC.          .478 ADJ.
16.29.23.CM        0.606 KWS.          .525 ADJ.
16.29.23.SS                           1.118
16.29.23.PP        2.950 SEC.     DATE 06/27/78
16.29.23.EJ  END OF JOB, SH
```

```
                                                                 COMPASS 3.5-470.     06/27/78  16.29.35.        PAGE        1
DIS
STORAGE ALLOCATION.
                ADDRESS   LENGTH                 BINARY CONTROL CARDS.

                      0        22                IDENT  DIS
                     22                          END    FIRST

                                         ENTRY POINTS.

                                         FIRST              0*

                                         EXTERNAL SYMBOLS.

                                         SYS=
```

```
                                                                 COMPASS 3.5-470.     06/27/78  16.29.35.        PAGE        2
   DIS

                                                     IDENT  DIS
                                                     ENTRY   FIRST
                                           *
                                           *
                                           *         USE OF DIS PSEUDO INSTRUCTION
                                           *
                                           *
                                                     LIST   D
      0  01020304050607101112               FIRST    DIS    2,ABCDEFGHIJKABCDEFGHIJ
      1  13010203040506071011
      2  01141405160114140516                        DIS    2,ALLENALLENALLEN
      3  01141405165555555555
      4  10172704310000000000                        DIS    0,*HOWDY*
      5  03171520012323550530                        DIS    3,/COMPASS EXAMPLE GOOD/
      6  01152014055507171704
      7  00000000000000000000
                                           *
                                           *         USE OF BSSZ PSEUDO OPERATION
                                           *
     10                         10          SAVE     BSSZ   8
     20                                     SAVEONE  BSSZ
     20  7160247021                                  ENDRUN
                                          DEFAULT SYMBOLS DEFINED BY COMPASS.

       0 R                                SYS=

                                                     END    FIRST
     22
                              47300B SCM STORAGE USED              22 STATEMENTS          4 SYMBOLS
                                         MODEL 174 ASSEMBLY      0.026 SECONDS            5 REFERENCES
```

DIS for Display Code

```
DIS                                                                   COMPASS 3.5-470.     06/27/78  16.29.35.      PAGE     1
SYMBOLIC REFERENCE TABLE.

FIRST           0      PROGRAM*          2/02 F     2/09 L
SAVE           10      PROGRAM*          2/20 L
SAVEONE        20      PROGRAM*          2/21 L
SYS=            0      EXTERNAL*         2/23
```

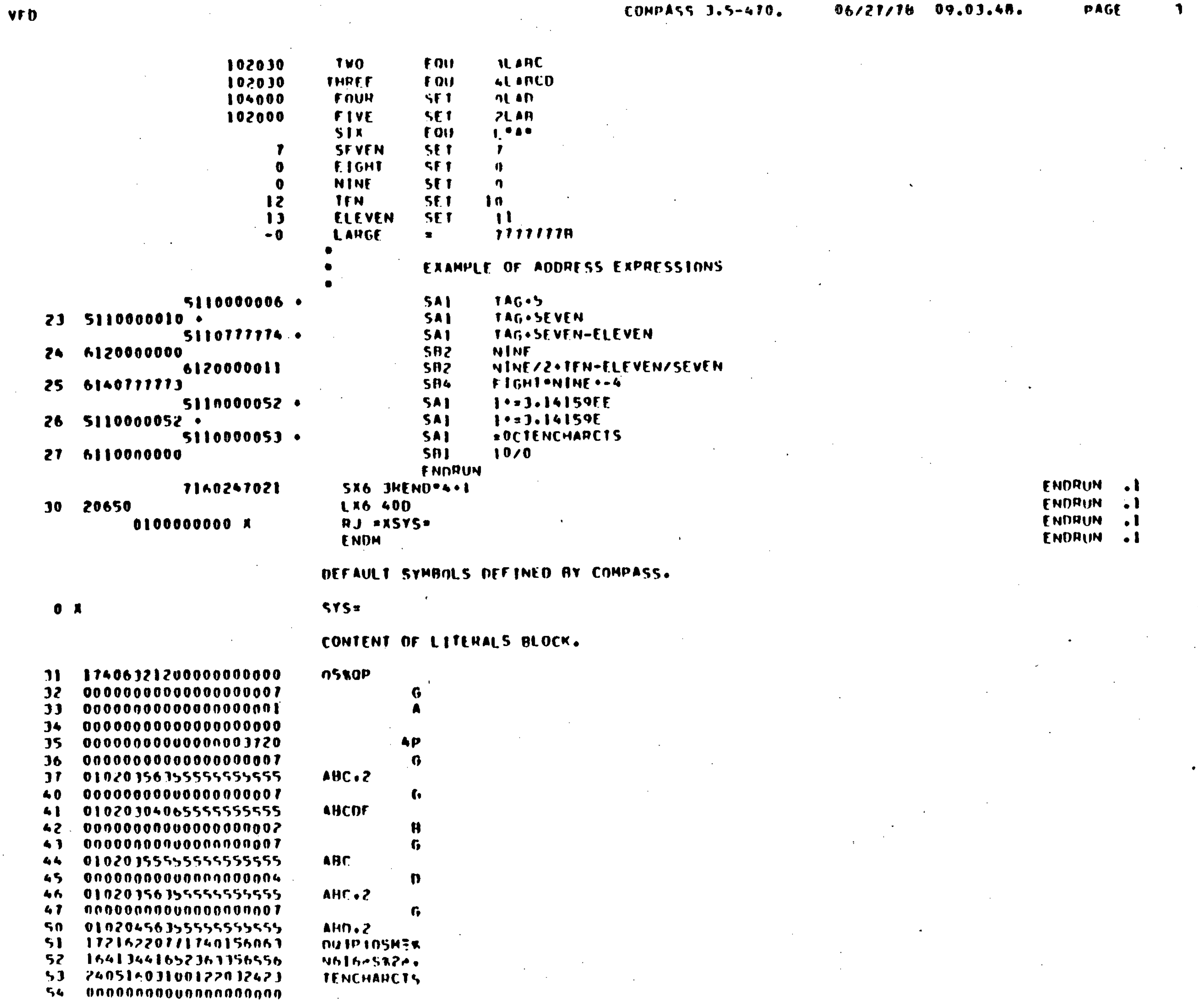
```
  MFF   NB2-  CYB175-SN1     4LB7/B6D  05/15/78
16.29.32.DON00MV  FROM     /SH
16.29.32.IP  00000256 WORDS - FILE INPUT  , DC 04
16.29.32.DON.     PSD.0278.72CT011A.MILLER
16.29.34.COMPASS.
16.29.35. ASSEMBLY COMPLETE.   473008 SCM USED.
16.29.35.    0.071  CPU SECONDS  ASSEMBLY TIME.
16.29.35.OP  00000448 WORDS - FILE OUTPUT , DC 40
16.29.35.MS    3504  WORDS  (   10752 MAX USED)
16.29.35.CPA        .095 SEC.          .095 ADJ.
16.29.35.IO         .470 SEC.          .470 ADJ.
16.29.35.CM        7.947 KWS.          .484 ADJ.
16.29.35.SS                           1.051
16.29.35.PP        3.148 SEC.     DATE 06/27/78
16.29.35.EJ  END OF JOB, SH
```

```
LITERAL                                                                        COMPASS 3.5-470.     06/27/78  16.30.59.      PAGE      2

                                                           IDENT  LITERAL
                                                           ENTRY  ST
                                                   *
                                                   *
                                                   *       USE OF LITERAL PSEUDO OPERATION
                                                   *
                                                   *
                                                           LIST   L,R,G,D
   0  5130000020 +                           ST            SA3    =1.5
                 5140000021 +                              SA4    =1.0
   1  30734                                                FX7    X3+X4
          24707                                            NX7    X7
                 5170000017 +                              SA7    SAVE+6
   2  5110000022 +                                         SA1    =0.0
                 31711                                     FX7    X1-X1
   3  5170000011 +                                         SA7    SAVE
                 24707                                     NX7    X7
   4  5170000012 +                                         SA7    SAVE+1
                 5120000023 +                              SA2    =3.5
   5  30621                                                FX6    X2+X1
          24606                                            NX6    X6
                 5160000014 +                              SA6    SAVE+3
   6  40712                                                FX7    X1*X2
          5170000015 +                                     SA7    SAVE+4
                      44612                                FX6    X1/X2
   7  5160000016 +                                         SA6    SAVE+5
                                                          ENDRUN
                 7160247021                SX6 3REND*4+1
  10  20650                                LX6 400
          0100000000 X                     RJ =XSYS=
  11  77777777777777777777              SAVE          DATA    -0
  12                         6                        BSSZ    6

                                        DEFAULT SYMBOLS DEFINED BY COMPASS.

   0 X                                  SYS=

                                        CONTENT OF LITERALS BLOCK.

  20  17206000000000000000              OP≡
  21  17204000000000000000              OPS
  22  00000000000000000000
  23  17217000000000000000              00+

  24                                                    END     ST

                          47300B SCM STORAGE USED              33 STATEMENTS          3 SYMBOLS
                                 MODEL 174 ASSEMBLY          0.036 SECONDS          10 REFERENCES
```

```
  MFF   NB2-  CYB175-SN1    4L07/R6B  05/15/78
16.30.56.DON00MX  FROM    /SH
16.30.56.IP  00000320 WORDS - FILE INPUT  , DC 0
16.30.56.DON.     PSD,0278,72CT011A,MILLER
16.30.58.COMPASS.
16.31.00. ASSEMBLY COMPLETE.   47300B SCM USED.
16.31.00.     0.081  CPU SECONDS  ASSEMBLY TIME.
16.31.00.LGO.
16.31.01.DMP(111,137)
16.31.01.EXIT.
16.31.01.OP  00000896 WORDS - FILE OUTPUT , DC 4
16.31.01.MS     3584  WORDS  (    7168 MAX USED)
16.31.01.CPA         .127 SEC.             .127 ADJ.
16.31.01.IO          .632 SEC.             .632 ADJ.
16.31.01.CM        10.086 KWS.             .615 ADJ.
16.31.01.SS                               1.376
16.31.01.PP         3.375 SEC.     DATE 06/27/78
16.31.01.EJ  END OF JOB, SH



                                         ENDRUN  .1
                                         ENDRUN  .1
                                         ENDRUN  .1
```

```
LITERAL                                                                   COMPASS 3.5-470.     06/27/78  16.30.59.      PAGE      3
SYMBOLIC REFERENCE TABLE.

SAVE          11     PROGRAM*       2/13 S    2/16 S    2/18 S    2/22 S    2/24 S    2/26 S    2/31 L
ST             0     PROGRAM*       2/02 E    2/09 L
SYS=           0     EXTERNAL*      2/30
```

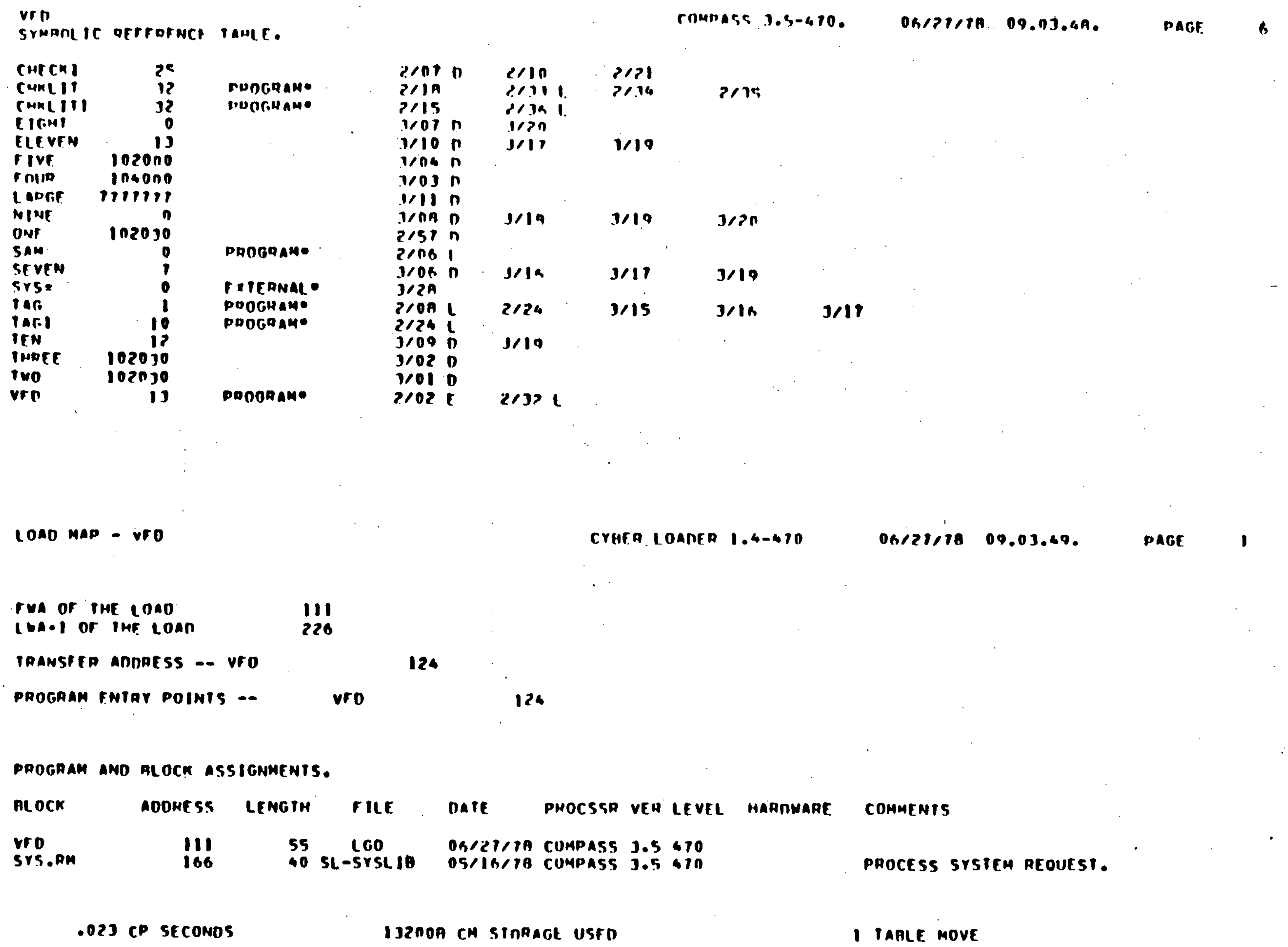
```
LOAD MAP - LITERAL                            CYBER LOADER 1.4-470      06/27/78  16.31.01.      PAGE      1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD          175

TRANSFER ADDRESS -- ST              111

PROGRAM ENTRY POINTS --      LITERAL          111


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS  LENGTH    FILE       DATE     PROCSSR VER LEVEL  HARDWARE   COMMENTS

LITERAL          111       24   LGO       06/27/78 COMPASS 3.5 470
SYS.RM           135       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.


      .023 CP SECONDS             13200B CM STORAGE USED                        1 TABLE MOVE

DUMP    RELATIVE                          DMP(111,137)

   00111   51300 00131 51400 00132       30734 24707 51700 00130      51100 00133 31711 46000       40712 51700 00126 44612
   00114   51700 00122 24707 46000       51700 00123 51200 00134      30621 24606 51600 00125

   00120   51600 00127 71602 47021       20650 01000 00137 46000      00000 00000 00000 00000   00125-17217 00000 00000 00000

   00126   00000 00000 00000 00000   00130-17215 00000 00000 00000      17206 00000 00000 00000       17204 00000 00000 00000

   00133   00000 00000 00000 00000       04000 00150 00000 00000      01300 00000 00000 00000       04000 00122 00000 00000
   00134   17217 00000 00000 00000
                                                          COMPASS 3.5-470.     06/27/78  16.30.59.      PAGE      1
  LITERAL
  STORAGE ALLOCATION.
                                                   BINARY CONTROL CARDS.
              ADDRESS   LENGTH
                                                   IDENT   LITERAL
                    0        24                    END     ST
                   24

                                         BLOCKS     TYPE      ADDRESS     LENGTH

                                         PROGRAM*  LOCAL           0          20
                                         LITERALS* LOCAL          20           4


                                         ENTRY POINTS.

                                         ST                 0+


                                         EXTERNAL SYMBOLS.

                                         SYS=
                                                  IDENT  VFD
                                                  ENTRY  VFD
                                      *
                                      *          USE OF VFD PSEUDO INSTRUCTION
                                      *
     0  00000000000000000100          SAM        DATA   100B
                           25         CHECK1     EQU    25B
     1  07537                         TAG        VFD    15/7537B
            10711                                BX7    X1
                  0000000000                     VFD    42/CHECK1
     2  0025
           15310611140500                        VFD    42/0LMYFILE
                           00                    VFD    18/1001B
     3  1001
            5120000032 +                         SA2    CHKLIT1
                        15310                    VFD    42/0LMYFILE,18/1001B
     4  611140500001001
     5  5120000033 +                             SA2    CHKLIT+1
                  0000000010                     VFD    27/1,10/2
     6  010
          00000000000052                         VFD    40/CHECK1
                        00030                    VFD    10/3,30/4
     7  000000004
    10  5130000001 +                  TAG1       SA3    TAG
                  0102032301                     VFD    18/3HABC,24/4HSAM ,18/0
    11  1555000000                               VFD    17/3HABC
    12  02040A                        *          VFD    18/3HSAM
                  230115              -
                                      *
                                      *          USE OF LITERAL PSEUDO INSTRUCTION
                                      *
    13  5110000031 +                  VFD        SA1    =10.5E4
                           32 +       CHKLIT     LIT    7,1,0,20E2
                  5120000032 +                   SA2    CHKLIT
    14  5130000034 +                             SA3    CHKLIT+2
                           32 +       CHKLIT1    LIT    7,1,0,20E2
                           36 +                  LIT    7,5HABC,2
                           40 +                  LIT    7,5HABCDE,2
                           43 +                  LIT    7,5HABC
                           45 +                  LIT    4,5HABC,2
                           47 +                  LIT    7,5HABD,2
                  5130000032 +                   SA3    =7
    15  5140000031 +                             SA4    =10.5E4
                  5150000042 +                   SA5    =2
                                      *
                                      *          EXAMPLE OF CONSTANTS
                                      *
    16  5211000055                               SA1    X1+1H
                  5121010203                     SA2    A1+3HABC
    17  5132010255                               SA3    A2+3LAB
                  5143010203                     SA4    A3+3HABC
7   20  5154020304                               SA5    A4+4HABCD
                  5120513252                     SA2    0L(7)
    21  6110010455                               SB1    3LAD
                  6120010203                     SB2    3HABC
7   22  6130020304                               SB3    4HABCD
```



```
VFD                                                              COMPASS 3.5-470.     06/27/78  09.03.48.      PAGE       1

                         102030    TWO       EQU    3LABC
                         102030   THREE      EQU    4LABCD
                         104000    FOUR      SET    0LAD
                         102000    FIVE      SET    2LAB
2                                  SIX       EQU    1.*A*
                              7    SEVEN     SET    7
                              0    EIGHT     SET    8
                              0    NINE      SET    9
                             12    TEN       SET   10
                             13    ELEVEN    SET    11
                             -0    LARGE     =      7777777B
                                  *
                                  *          EXAMPLE OF ADDRESS EXPRESSIONS
                                  *
                   5110000006 *              SA1    TAG+5
       23  5110000010 *                      SA1    TAG+SEVEN
                   5110777774 *              SA1    TAG+SEVEN-ELEVEN
       24  6120000000                        SB2    NINE
                   6120000011                SB2    NINE/2+TEN-ELEVEN/SEVEN
       25  6140777773                        SB4    EIGHT*NINE+-4
                   5110000052 *              SA1    1+=3.14159EE
       26  5110000052 *                      SA1    1+=3.14159E
                   5110000053 *              SA1    =0CTENCHARCTS
       27  6110000000                        SB1    10/0
                                             ENDRUN
                   7160247021      SX6 3RENDR*4+1                                                     ENDRUN   .1
       30  20650                   LX6 40D                                                            ENDRUN   .1
               0100000000 X        RJ =XSYS=                                                          ENDRUN   .1
                                   ENDM                                                               ENDRUN   .1

                                 DEFAULT SYMBOLS DEFINED BY COMPASS.

        0 X                      SYS=

                                 CONTENT OF LITERALS BLOCK.

       31  17406321200000000000  OS%OP
       32  00000000000000000007         G
       33  00000000000000000001         A
       34  00000000000000000000
       35  00000000000000003720        4P
       36  00000000000000000007         G
       37  01020356355555555555  ABC.2
       40  00000000000000000007         G
       41  01020304065555555555  ABCDF
       42  00000000000000000002         B
       43  00000000000000000007         G
       44  01020355555555555555  ABC
       45  00000000000000000004         D
       46  01020356355555555555  ABC.2
       47  00000000000000000007         G
       50  01020456355555555555  ABD.2
       51  17216220771740156063  OQIPIOSM≡%
       52  16413441652363356556  N616-S%2-.
       53  24051403100122012423  TENCHARCTS
       54  00000000000000000000
```

]

```
VFD                                                             COMPASS 3.5-470.     06/27/78  09.03.48.       PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH              BINARY CONTROL CARDS.

                0       55              IDENT   VFD
               55                       END     VFD

                                 BLOCKS     TYPE       ADDRESS     LENGTH

                                 PROGRAM*  LOCAL             0         31
                                 LITERALS* LOCAL            31         24

                                 ENTRY POINTS.

                                 VFD              13*

                                 EXTERNAL SYMBOLS.

                                 SYS=

 VFD                                                             COMPASS 3.5-470.     06/27/78  09.03.48.       PAGE      4


      55                                    END     VFD

                         47300B SCM STORAGE USED                81 STATEMENTS        19 SYMBOLS
                                MODEL 174 ASSEMBLY       0.091 SECONDS         40 REFERENCES

VFD                                                             COMPASS 3.5-470.     06/27/78  09.03.48.       PAGE      5
ERROR DIRECTORY.
                                                                                       MFF   NB2-  CYB175-SN1    4LB7/R6B  05/15/78
        2 TYPE ERROR               ADDRESS ERROR ON SYMBOL DEFINITION.                 09.03.41.DON00SR  FROM    /SH
             OCCURRED ON PAGES           3                                             09.03.42.IP  00000384 WORDS - FILE INPUT  , DC (
        7 TYPE ERROR               ADDRESS VALUE EXCEEDS FIELD SIZE, RESULT TRUNCATED 09.03.42.DON.    PSN.0278.72CT011A.MILLER
             OCCURRED ON PAGES           2,        3                                   09.03.47.COMPASS(LO=$$$$)
                                                                                       09.03.48.            4 WARNING MESSAGES IN VFD
                                                                                       09.03.48. ASSEMBLY COMPLETE.   47300B SCM USED.
                                                                                       09.03.48.    0.148  CPU SECONDS  ASSEMBLY TIME.
                                                                                       09.03.48.LGO.
                                                                                       09.03.49.ERROR MODE =01.  ADDRESS =000130
                                                                                       09.03.50.OP  00001920 WORDS - FILE OUTPUT , DC 4
                                                                                       09.03.50.MS    3584  WORDS  (    7168 MAX USED)
                                                                                       09.03.50.CPA         .254 SEC.          .254 ADJ.
                                                                                       09.03.50.IO          .659 SEC.          .659 ADJ.
                                                                                       09.03.50.CM       12.164 KWS.           .742 ADJ.
                                                                                       09.03.50.SS                            1.656
                                                                                       09.03.50.PP         6.945 SEC.      DATE 06/27/78
                                                                                       09.03.50.EJ  END OF JOB, SH
```



```
VFD                                                              COMPASS 3.5-470.     06/27/78  09.03.48.     PAGE     6
SYMBOLIC REFERENCE TABLE.

CHECK1        25                         2/07 D    2/10      2/21
CHKLIT        12      PROGRAM*           2/18      2/33 L    2/34      2/35
CHKLIT1       32      PROGRAM*           2/15      2/36 L
EIGHT          0                         3/07 D    3/20
ELEVEN        13                         3/10 D    3/17      3/19
FIVE      102000                         3/04 D
FOUR      104000                         3/03 D
LARGE    7777777                         3/11 D
NINE           0                         3/08 D    3/18      3/19      3/20
ONE       102030                         2/57 D
SAM            0      PROGRAM*           2/06 I
SEVEN          7                         3/06 D    3/16      3/17      3/19
SYS=           0      EXTERNAL*          3/28
TAG            1      PROGRAM*           2/08 L    2/24      3/15      3/16      3/17
TAG1          10      PROGRAM*           2/24 L
TEN           12                         3/09 D    3/19
THREE     102030                         3/02 D
TWO       102030                         3/01 D
VFD           13      PROGRAM*           2/02 E    2/32 L



LOAD MAP - VFD                                            CYBER LOADER 1.4-470      06/27/78  09.03.49.     PAGE     1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD          226

TRANSFER ADDRESS -- VFD              124

PROGRAM ENTRY POINTS --        VFD              124


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK         ADDRESS   LENGTH    FILE     DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

VFD              111        55    LGO      06/27/78 COMPASS 3.5 470
SYS.RM           166        40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.


     .023 CP SECONDS              13200B CM STORAGE USED                      1 TABLE MOVE
```

```
DMPX.

P    000000   A0   000100   B0   000000
PA   123200   A1   000055   B1   000001     C(A1)=  5110   0000   0103   1100   0055     C(B1)=  0000   0000   0000   0000   0000
FL   000300   A2   010204   B2   000002     C(A2)=                                       C(B2)=  0000.  0000   0000   0000   0000
EM   700700   A3   000143   B3   012733     C(A3)=  0000   0000   0000   0000   0007     C(B3)=
PE   000041   A4   000142   B4   000301     C(A4)=  1740   6321   2000   0000   0000     C(B4)=
FE   000000   A5   000153   B5   000124     C(A5)=  0000   0000   0000   0000   0002     C(B5)=  5110   0001   4251   2000   0143
MA   000400   A6   000001   B6   000300     C(A6)=  0000   0000   0000   0000   0000     C(B6)=
              A7   000001   B7   027756     C(A7)=  0000   0000   0000   0000   0000     C(B7)=
X0   0000    0000    0000    0000    0000
X1   5110    0000    0101    1100    0055
X2   0000    0000    0000    0000    0000
X3   0000    0000    0000    0000    0007
X4   1740    6321    2000    0000    0000
X5   0000    0000    0000    0000    0002
X6   1505    1520    0000    0000    0061
X7   0000    0000    0000    0000    0000

   00000     00010 00130 00000 00000        00000 00000 00000 00000
   00054    -56110 01110 00054 54710        51100 00001 03110 00055        64550 02550 00000 46000        00000 00000 00000 00000
   00060     15051 52000 00000 00061        00000 00300 00000 00001        07040 00060 51600 00001        04000 00063 00000 00021
   00064     14071 70000 00000 00000        00000 00000 00000 00226        40000 00000 02000 00111        00000 00000 40000 00000
   00070     14071 75755 00000 00000        00000 00000 00000 00000
   00100    -54000 00000 01000 00001

   00100     54000 00000 01000 00001        00011 50000 00000 00226        00000 00000 00000 00000
   00104    -00000 00000 00000 00226        00000 00000 00000 00000
   00110    -26060 40000 00000 00124        00000 00000 00000 00100        07537 10711 00000 00000        00251 53106 11140 50000
   00114     10010 51200 00143 15310.       61114 05000 01001 46000        51200 00144 00000 00010        01000 00000 00005 20030
   00120     00000 00040 61000 46000        51300 00112 01020 32301        15550 00000 61000 46000        02040 60000 23011 50000
   00124     51100 00142 51200 00143        51300 00145 51300 00143        51400 00142 51500 00153        52110 00055 51210 10203
   00130     51320 10255 51430 10203        51540 20304 51205 13252        61100 10455 61200 10203        61300 20304 51100 00117
   00134     51100 00121 51100 00106        61200 00000 61200 00011        61407 77773 51100 00163        51100 00163 51100 00164
   00140     61100 00000 71602 47021        20650 01000 00170 46000        17406 32120 00000 00000        00000 00000 00000 00007
   00144     00000 00000 00000 00001        00000 00000 00000 00000        00000 00000 00000 03720        00000 00000 00000 00007
   00150     01020 35635 55555 55555        00000 00000 00000 00007        01020 30406 55555 55555        00000 00000 00000 00002
   00154     00000 00000 00000 00007        01020 35555 55555 55555        00000 00000 00000 00004        01020 35635 55555 55555
   00160     00000 00000 00000 00007 .      01020 45635 55555 55555        17216 22077 17401 56063        16413 44165 23633 56556
   00164     24051 60310 01220 32423        00000 00000 00000 00000        54110 20123 03310 00167        04000 00173 61000 46000
   00170     04004 00170 61000 46000        51100 00001 03110 00171        54610 04000 00167 46000        51100 00066 03310 00175
   00174     51100 00166 04000 00176        71100 00130 20160 46000        13661 13161 13661 46000        51600 00167 10611 46000
   00200     51100 00001 01000 00166        20652 01000 00170 46000        51100 00001 03110 00202        04004 00203 61000 46000
   00204     51100 00001 03110 00203        71602 20314 04000 00201        20150 36661 01000 00170        04004 00207 61000 46000
   00210     71602 20314 20652 36662        53160 20173 03110 00207        03010 00207 51100 00001        03110 00211 71100 00001
   00214     04000 00206 61000 46000        71603 24616 12661 20651        01000 00170 61000 46000        04004 00217 61000 46000
   00220     73660 20630 12161 73610        20123 03210 00215 20151        13116 20636 51600 00225        74660 36116 20123 46000
   00224     04000 00215 61000 46000        00000 00000 00000 00000        60000 00000 04004 00226        60000 00000 04004 00227
   00230     60000 00000 04004 00230
```

LESSON 7

PSEUDO OPERATIONS

LESSON PREVIEW:

THE LESSON COVERS SOME OF THE BASIS PSEUDO OPERATIONS USED IN ADDING COSMETIC EFFECTS TO A LISTING. ALSO SOME OF THE BASIC PSEUDO OPS USED IN NORMAL PROGRAMMING PRACTICES WILL BE COVERED.

REFERENCES:

CHAPTER 4	COMPASS REFERENCE MANUAL #60492600

TRAINING AIDS:

VISUAL SET 7

PROJECTS:

OBJECTIVES:

AT THE COMPLETION OF THE LESSON THE STUDENT WILL BE ABLE TO:

1. WRITE A PROGRAM USING GOOD DOCUMENTATION TECHNIQUES.
2. USE PSEUDO OPERATIONS THAT WILL ALLOW FOR WRITING MEANINGFUL EFFECIENT PROGRAMS.

```
PSEUDO INSTRUCTION
STORAGE ALLOCATION.                                    COMPASS 3.6-476.    01/09/79  08.34.39.     PAGE      1

          ADDRESS   LENGTH            BINARY CONTROL CARDS.

                0        2            IDENT   TITLE
                2                     END     TITLE

                            ENTRY POINTS.

                            TITLE            0+

                            EXTERNAL SYMBOLS.

                            SYS≠
```

```
                                                 IDENT  TITLE
                                                 ENTRY  TITLE
                                                 LIST   S
                                         *****************************************************************
                                         *                                                               *
                                         *       TITLE-ASSEMBLY LISTING TITLE                            *
                                         *                                                               *
                                         *       THE FIRST TITLE PSEUDO INSTRUCTION ESTABLISHES THE TITLE THAT*
                                         *       WILL BE PRINTED ON EACH PAGE OF THE LISTING.  A SUBSEQUENT  *
                                         *       TITLE INSTRUCTION GENERATES A SUBTITLE AND CAUSES A PAGE EJECT.
                                         *       IF THE SUBPROGRAM DOES NOT INCLUDE A TITLE INSTRUCTION, COMPASS
                                         *       PRINTS THE VARIABLE FIELD OF THE FIRST IDENT PSEUDO INSTRUCTION,
                                         *       AS THE TITLE.  A TITLE INSTRUCTION WITHOUT A CHARACTER STRING*
                                         *       PRODUCES AN UNTITLED LISTING.  A NAME IN THE LOCATION FIELD  *
                                         *       INTRODUCES A NEW SUBPROGRAM SUB-TITLE.                       *
                                         *                                                               *
                                         *LOCATION OPERATION VARIABLE-SUBFIELDS                          *
                                         *                                                               *
                                         *NAME     TITLE    STRING                                       *
                                         *                                                               *
                                         *    NAME    NEW SUBPROGRAM SUB-SUBTITLE TO BE PRINTED IN CHARACTER  *
                                         *            POSITION 70-79 ON THE SECOND LINE OF THE PAGE.  A BLANK *
                                         *            NAME CLEARS THE SUB-SUBTITLE.                           *
                                         *                                                               *
                                         *    STRING  COMPASS SEARCHES THE COLUMNS FOLLOWING THE BLANK THAT TERM-
                                         *            INATES THE OPERATION FIELD.  IF IT DOES NOT FIND A NONBLANK
                                         *            CHARACTER BEFORE THE DEFAULT COMMENTS COLUMN (SEE COL IN- *
                                         *            STRUCTION), IT TAKES THE CHARACTERS STARTING WITH THE DE- *
                                         *            FAULT COMMENTS COLUMN MINUS ONE UP TO THE END OF THE STATE-
                                         *            MENT.  OTHERWISE, THE TITLE OR SUBTITLE BEGINS WITH THE   *
                                         *            FIRST NONBLANK CHARACTER FOLLOWING TITLE AND CONTINUES TO *
                                         *            THE END OF THE STATEMENT OR TO 62 CHARACTERS.  ANY CHARACT-
                                         *            ERS BEYOND THE 62ND ARE LOST.  A BLANK STRING PRODUCES AN *
                                         *            UNTITLED LISTING.                                     *
                                         *                                                               *
                                         *****************************************************************
 0  46000                                 TITLE    NO
                                                 ENDRUN
       7160247021                          SX6 3REND*4+1                                                  ENDRUN  .1
                    20650                  LX6 40D                                                        ENDRUN  .1
 1  0100000000 X                           RJ =XSYS=                                                      ENDRUN  .1
                                           ENDM                                                           ENDRUN  .1
                                                 TITLE  PSEUDO INSTRUCTION
                                         *****************************************************************
                                         *                                                               *
                                         *       IDENT-SUBPROGRAM IDENTIFICATION                         *
                                         *                                                               *
                                         *       AN IDENT PSEUDO INSTRUCTION OF THE FOLLOWING FORM IS THE  *
                                         *       FIRST STATEMENT OF A SUBPROGRAM RECOGNIZED BY THE ASSEMBLER. *
                                         *       USUALLY, ANY LINE PRECEDING THE FIRST IDENT OR BETWEEN AN END*
                                         *       AND IDENT ARE ASSUMED TO BE COMMENTS.  HOWEVER, WHEN COMPASS *
                                         *       HAS BEEN CALLED BY SOME OTHER LANGUAGE PROCESSOR SUCH AS  *
                                         *       FORTRAN, THE ASSEMBLER RETURNS CONTROL TO THE PROCESSOR WHEN *
                                         *       THE STATEMENT FOLLOWING END IS NOT IDENT.  FOR A RELOCATABLE *
                                         *       SUBPROGRAM, COMPASS FLAGS ANY SUBSEQUENT USE OF IDENT BEFORE *
                                         *       END AS AN ERROR.  FOR AN ABSOLUTE SUBPROGRAM, A SECOND FORM OF
                                         *       IDENT DESCRIBED UNDER BINARY CONTROL IS AVAILABLE FOR OVERLAY*
```

```
*         GENERATION.                                                   *
*                                                                       *
*         THE FORMAT OF IDENT VARIES ACCORDING TO THE TYPE OF ASSEMBLY.*
*                                                                       *
*         CPU RELOCATEABLE FORMAT:                                      *
*                                                                       *
*LOCATION OPERATION VARIABLE-SUBFIELDS                                  *
*                                                                       *
*         IDENT  NAME                                                   *
*                                                                       *
*         CPU ABSOLUTE FORMAT                                           *
*                                                                       *
*LOCATION OPERATION VARIABLE-SUBFIELDS                                  *
*                                                                       *
*         IDENT  NAME,ORIGIN,ENTRY,L1,L2                                *
*                                                                       *
*         7600 PPU ABSOLUTE FORMAT:                                     *
*                                                                       *
*LOCATION OPERATION VARIABLE-SUBFIELDS                                  *
*                                                                       *
*         IDENT  NAME,ORIGIN,ENTRY,PPU                                  *
*                                                                       *
*         6000 SERIES PPU ABSOLUTE FORMAT:                              *
*                                                                       *
*LOCATION OPERATION VARIABLE-SUBFIELDS                                  *
*                                                                       *
*         IDENT  NAME,ORIGIN                                            *
*                                                                       *
*    NAME      NAME OF THE SUBPROGRAM OR OVERLAY.  THE PARAMETER IS     *
*              REQUIRED.  FOR A CPU RELOCATABLE OR ABSOLUTE ASSEMBLY,   *
*              NAME CAN BE 1-7 CHARACTERS, OF WHICH THE FIRST MUST BE   *
*              ALPHABETIC (A-Z) AND THE LAST MUST NOT BE A COLON.       *
*                                                                       *
*              FOR A CYBER 70/MODEL 76 OR 7600 PPU ASSEMBLY, NAME CAN   *
*              BE 1-7 CHARACTERS.  FOR A CYBER 70/MODEL 72,73,74 OR 6000
*              -SERIES PPU ASSEMBLY NAME CAN BE 1-3 CHARACTERS.  IN     *
*              EITHER CASE, THERE IS NO RESTRICTION ON THE FIRST        *
*              CHARACTER, BUT THE LAST CHARACTER MUST NOT BE A COLON.   *
*                                                                       *
*    ORIGIN    AN EXPRESSION SPECIFYING THE FIRST WORD ADDRESS OF THE   *
*              ABSOLUTE PROGRAM OR OVERLAY.  THE OVERLAY LOADER TABLE   *
*              AND ALL CODE ASSEMBLED STARTING AT THIS ADDRESS AND      *
*              ENDING WITH THE NEXT SEGMENT, NONBLANK IDENT, OR END     *
*              INSTRUCTION COMPRISES THE OVERLAY. FOR A SINGLE ENTRY    *
*              POINT CPU PROGRAM ADDRESS FOR THE OVERLAY IS ORIGIN-1.   *
*              THE WORD AT ORIGIN-1 IS OVERLAYED BY THE 50(OCTAL) LOADER
*              CONTROL TABLE.  FOR A MULTIPLE ENTRY POINT CPU PROGRAM,  *
*              THE LOAD ADDRESS FOR THE ABSOLUTE OVERLAY IS ORIGIN-WC-1,
*              WHERE WC IS THE NUMBER OF ENTRY POINTS IN THE 51(OCTAL)  *
*              LOADER TABLE.                                            *
*                                                                       *
*              FOR A PPU SUBPROGRAM, THE LOAD ADDRESS IS ORIGIN-5.      *
*              FIVE 12-BIT PPU WORDS ARE OVERLAYED BY THE 60-BIT LOADER*
*              TABLE.                                                   *
*                                                                       *
*              DATA CAN BE GENERATED IN LOCATIONS STARTING WITH ORIGIN  *
*              AND ABOVE, BUT NOT BELOW ORIGIN.  THE ORIGIN SUBFIELD    *
*                  DOES NOT SERVE THE SAME FUNCTION AS ORG NOR DOES IT      *
*                  REPLACE ORG FOR SETTING THE ORIGIN COUNTER.              *
*                                                                           *
*                  IF THE ORIGIN FIELD IS NULL FOR AN ABSOLUTE SUBPROGRAM,  *
*                  THE ASSEMBLER USES ADDRESS 000000 RA(S) AS THE ORIGIN    *
*                  FOR A CPU PROGRAM AND 0000 AS THE ORIGIN FOR A PPU PROG. *
*                                                                           *
*                  FOR A RELOCATABLE SUBPROGRAM, THE SUBFIELD IS IGNORED... *
*                  THE LOADER AUTOMATICALLY RELOCATES THE FIRST SUBPROGRAM  *
*                  TO BE LOADED STARTING AT RA(S)+100(OCTAL), THE SECOND    *
*                  SUBPROGRAM STARTING AT THE FIRST AVAILABLE LOCATION      *
*                  FOLLOWING THE FIRST SUBPROGRAM, ETC.                     *
*                                                                           *
*      ENTRY       FOR A CYBER 70/MODEL 76 OR 7600 PPU ASSEMBLY OR FOR AN   *
*                  ABSOLUTE CPU ASSEMBLY, THIS SUBFIELD CONTAINS AN EXPRES- *
*                  SION SPECIFYING THE SUBPROGRAM ENTRY ADDRESS,WHICH CAN   *
*                  BE SYMBOLIC.                                             *
*                                                                           *
*      L1,L2       ABSOLUTE EXPRESSIONS SPECIFYING THE LEVEL NUMBERS OF THE *
*                  OVERLAY.  L1 IS THE PRIMARY LEVEL(0-63) AND L2 IS THE    *
*                  SECONDARY LEVEL (0-63).  WHEN THE FIRST IDENT IDENTIFIES *
*                  THE MAIN OVERLAY, L1 AND L2 CAN BE OMITTED.  IF L1 IS    *
*                  OMITTED, IT IS SET TO 00.  IF L2 IS OMITTED, IT IS SET   *
*                  TO 00.                                                   *
*                                                                           *
*                  BECAUSE THE FIRST IDENT PRECEDES ANY USE OF THE BASE     *
*                  PSEUDO INSTRUCTION, THE LEVEL NUMBERS ON THE IDENT ARE   *
*                  EVALUATED AS DECIMAL UNLESS SPECIFICALLY DESIGNATED AS   *
*                  OCTAL BY A POST RADIX.                                   *
*                                                                           *
*      PPU         ABSOLUTE EXPRESSION SPECIFYING THE NUMBER OF THE PPU ON  *
*                  WHICH THIS PROGRAM IS TO BE LOADED.  ON THE FIRST IDENT, *
*                  THIS NUMBER IS EVALUATED AS DECIMAL UNLESS SPECIFICALLY  *
*                  DESIGNATED AS OCTAL.                                     *
*                                                                           *
*                  A LOCATION FIELD SYMBOL, IF PRESENT, IS IGNORED.         *
*                                                                           *
*****************************************************************************
```

```
A123456789 TITLE SPACE PSEUDO INSTRUCTION.
```

```
PSEUDO INSTRUCTION                                         COMPASS 3.6-476.     01/09/79  08.34.39.      PAGE      5
SPACE PSEUDO INSTRUCTION.                                  A123456789

                         A123456789 TITLE SPACE PSEUDO INSTRUCTION.




PSEUDO INSTRUCTION
EJECT PSEUDO INSTRUCTION                                   COMPASS 3.6-476.     01/09/79  08.34.39.      PAGE      6

                              TITLE  EJECT PSEUDO INSTRUCTION
*************************************************************************
*                                                                       *
*            EJECT-EJECT PAGE AND BEGIN NEW SUB-TITLE                   *
*                                                                       *
*            THE EJECT PSEUDO INSTRUCTION ADVANCES PRINTER PAPER TO A NEW *
*            PAGE BEFORE PRINTING.  THEN, PAGE HEADINGS ARE PRINTED AND *
*            LISTING CONTINUES.  EJECT HAS NO EFFECT,OTHER THAN SETTING THE
*            SUB-SUBTITLE, IF IT IS GENERATED BY "DUP,ECHO,RMT,XTEXT", OR *
*            A MACRO OR OPDEF EXPANSION, AND THE CORRESPONDING "LIST"   *
*            OPTIONS ARE NOT ALL SELECTED.                              *
*                                                                       *
*LOCATION OPERATION VARIABLE-SUBFIELDS                                  *
*                                                                       *
*NAME     EJECT                                                         *
*                                                                       *
*    NAME  NEW PROGRAM SUB-SUBTITLE FOR THE PAGE WILL BE PRINTED IN     *
*          CHARACTER POSITIONS 70-79 OF THE SECOND LINE OF THE PAGE.  A*
*          BLANK NAME CLEARS THE SUB-TITLE.                             *
*                                                                       *
*          AN ENTRY IN THE VARIABLE FIELD, IF PRESENT, IS IGNORED.      *
*                                                                       *
*************************************************************************
```

```
TEJECT    EJECT
************************************************************************
*                                                                      *
*         TTL-NEW ASSEMBLY LISTING TITLE                               *
*                                                                      *
*         THE "TTL" PSEUDO INSTRUCTION INTRODUCES A NEW MAIN TITLE TO BE
*         PRINTED ON EACH PAGE OF THE LISTING, AND CLEARS THE SUBTITLE.*
*                                                                      *
*LOCATION OPERATION VARIABLE-SUBFIELDS                                 *
*                                                                      *
*NAME      TTL      STRING                                             *
*                                                                      *
*    STRING  COMPASS SEARCHES THE COLUMNS FOLLOWING THE BLANK THAT TER-*
*            MINATES THE OPERATING FIELD.  IF IT DOES NOT FIND A NONBLANK
*            CHARACTER BEFORE THE DEFAULT COMMENTS COLUMN (SEE COL PSEUDO
*            INSTRUCTION), IT TAKES THE CHARACTERS STARTING WITH THE DE-
*            FAULT COMMENTS COLUMN MINUS ONE UP TO THE STATEMENT END.  *
*            OTHERWISE, THE TITLE BEGINS WITH THE FIRST NONBLANK CHARACTER
*            FOLLOWING "TTL" AND CONTINUES TO THE END OF THE STATEMENT *
*            OR TO THE 62ND CHARACTER.  ANY CHARACTER BEYOND THE 62ND  *
*            ARE LOST.  A BLANK STRING PRODUCES AN UNTITLED LISTING.   *
*                                                                      *
*    NAME    NEW SUB-SUBTITLE TO BE PRINTED IN CHARACTER POSITION 70-79*
*            ON THE SECOND LINE OF THE PAGE.  A BLANK NAME CLEARS THE  *
*            SUB-SUBTITLE.                                             *
*                                                                      *
*            "TTL" DOES NOT CAUSE A PAGE EJECT.                        *
*                                                                      *
************************************************************************
```

```
          EJECT
 MILLER   TTL     TTL PSEUDO INSTRUCTION
*********************************************************************
*                                                                   *
*         NOREF - OMIT SYMBOL REFERENCES                            *
*                                                                   *
*         THE "NOREF" PSEUDO INSTRUCTION CAUSES THE SYMBOLS NAMED IN THE   *
*         VARIABLE FIELD TO BE SUPPRESSED FROM THE SYMBOLIC REFERENCE TABLE.  *
*                                                                   *
*LOCATION OPERATION VARIABLE SUBFIELDS                              *
*                                                                   *
*         NOREF     SYM1,SYM2,......SYMN                            *
*                                                                   *
*         SYMI      ONE OR MORE SYMBOLS DEFINED IN THE SUBPROGRAM.  IF A     *
*                   SYMBOL QUALIFIER IS IN EFFECT WHEN THE "NOREF" IS ENCOUNTER-
*                   ED, THE SYMBOLS ARE ASSUMED TO BE QUALIFIED BY THE QUALIFIER
*                   IN USE.  ALTERNATIVELY, SYMI CAN BE A NON-BLANK QUALIFIER  *
*                   SYMBOL ENCLOSED BY SLANT BARS, /QUALIFIER/, IN WHICH CASE  *
*                   ALL SYMBOLS QUALIFIED BY THE SPECIFIED QUALIFIER ARE       *
*                   SUPPRESSED FROM THE SYMBOLIC REFERENCE TABLE.              *
*                                                                   *
*         A LOCATION FIELD  FIELD SYMBOL, IF PRESENT, IS IGNORED.   *
*                                                                   *
*********************************************************************


******************************************************************
*                                                                *
*         USE-ESTABLISH AND USE BLOCK                            *
*                                                                *
*         USE ESTABLISHES. A NEW BLOCK OR RESUMES USE OF AN ALREADY  *
*         ESTABLISHED BLOCK.  THE BLOCK IN USE IS THE BLOCK INTO WHICH *
*         CODE IS SUBSEQUENTLY ASSEMBLED.  A USER MAY ESTABLISH UP TO  *
*         252 BLOCKS.                                            *
*                                                                *
*LOCATION OPERATION VARIABLE-SUBFIELDS                           *
*                                                                *
*         USE    BLOCK                                           *
*                                                                *
*  BLOCK   IDENTIFIES BLOCK TO BE USED, AS FOLLOWS:              *
*                                                                *
*          0 OR BLANK  NOMINAL BLOCK(ABSOLUTE OR ZERO)           *
*          //          BLANK COMMON BLOCK, FOR A RELOCATABLE SUBPROGRAM,
*                      THIS BLOCK CANNOT CONTAIN DATA.  THE ONLY STORAGE
*                      ALLOCATION INSTRUCTIONS ' IAT CAN FOLLOW ARE BSS *
*                      AND ORG.  THE BSSZ INSTRUCTION IS ILLEGAL BECAUSE
*                      IT PRESETS THE BLOCK TO ZEROS.            *
*                                                                *
*         /NAME/       LABELED COMMON BLOCK.  A NAME CAN BE A MAXIMUM  *
*                      OF 7 CHARACTERS AND CANNOT INCLUDE BLANK OR COMMA.
*                      THE FIRST AND LAST CHARACTERS MUST NOT BE A COLON
*                      CONVENTIONS IMPOSED BY THE LOADER OR OTHER  *
*                      ASSEMBLERS OR COMPILERS COULD FURTHER RESTRICT  *
*                      THE USE OF NAMES.                         *
*                                                                *
*         NAME         LOCAL BLOCK.  A NAME CAN BE 1-8 CHARACTERS,   *
*                      EXCLUDING BLANK OR COMMA.  USE OF THIS NAME   *
*                        ENCLOSED BY THE BRACKETS DOES NOT CAUSE THE BLOCK
*                        TO BECOME A LABELED COMMON BLOCK.  FOR EXAMPLE, *
*                        USE A AND USE /A/ ARE DIFFERENT BLOCKS.         *
*                                                                        *
*          *             BLOCK IN USE PRIOR TO CURRENT USE,USELCM,ORG, OR*
*                        ORGC.                                           *
*                                                                        *
*          A LOCATION SYMBOL, IF PRESENT, IS IGNORED.                    *
*                                                                        *
**************************************************************************
 USELESS  BSS     0
 USELESS  NOREF  USELESS      THIS SYMBOL SHOULD NOT OCCUR IN THE SYMBOLIC
                              REFERENCE TABLE
          SPACE  20
```

2

```
**************************************************************************
*                                                                        *
*           SPACE-SKIP LINES AND BEGIN NEW SUB-TITLE                     *
*                                                                        *
*           THE "SPACE" PSEUDO INSTRUCTION SPACES THE ASSEMBLER LISTING. *
*           WHEN A PAGE IS FULL, AN EJECT OCCURS AND LISTING RESUMES ON THE
*           NEXT PAGE.  A SPACE IMMEDIATELY FOLLOWING AN "EJECT" IS IG-  *
*           NORED.  "SPACE" HAS NO EFFECT, OTHER THAN SETTING THE SUB-SUB-
*           TITLE, IF IT IS GENERATED BY "DUP, ECHO, RMT, XTEXT", OR A   *
*           MACRO OR OPDEF EXPANSION, AND THE CORRESPONDING "LIST" OPTIONS
*           ARE NOT ALL SELECTED.                                        *
*                                                                        *
*LOCATION OPERATION VARIABLE-SUBFIELDS                                   *
*                                                                        *
*NAME       SPACE     SCNT,RCNT                                          *
*                                                                        *
*     NAME  NEW SUBPROGRAM SUBTITLE WILL BE PRINTED IN CHARACTER 70-79 ON
*           THE SECOND LINE OF THE NEXT PAGE HEADING.  A BLANK NAME CLEARS
*           THE SUB-SUBTITLE.                                            *
*                                                                        *
*     SCNT  AN ABSOLUTE EXPRESSION SPECIFYING A POSITIVE INTEGER NUMBER *
*           OF SPACES BETWEEN THE MOST RECENT LINE AND THE NEXT LINE OF *
*           PRINTOUT.  IF BASE IS M, SCNT IS ASSUMED TO BE DECIMAL.  IF *
*          SCNT IS OMITTED OR ZERO, NO LINE IS SKIPPED.                     *
*                                                                           *
*    RCNT  AN ABSOLUTE EXPRESSION SPECIFYING A POSITIVE INTEGER NUMBER *
*          OF LINES THAT MUST BE REMAINING ON THE PAGE FOLLOWING SPACING.
*          IF BASE IS M, RCNT IS ASSUMED TO BE DECIMAL.                     *
*                                                                           *
*          IF SCNT + RCNT EXCEEDS THE NUMBER OF LINES ON THE PAGE BEFORE
*          SPACING OCCURS, THE "SPACE" ACTS LIKE AN "EJECT".  NOTE THAT*
*          EITHER THE EJECT OCCURS OR THE NUMBER OF SPACES ARE SKIPPED *
*                                                                           *
*          BLANK CARDS CAN ALSO BE USED TO SPACE THE LISTING.               *
*                                                                           *
*****************************************************************************
          SPACE  10,30




*********************************************************************************
*                                                                               *
*         XREF - REFERENCE SYMBOLIC ADDRESS                                     *
*                                                                               *
*         THE "XREF" PSEUDO INSTRUCTION PROVIDES THE OPTIONS OF HAVING THE      *
*         SYMBOLIC REFERENCE TABLE  CONTAIN REFERENCES TO SYMBOLS ACCORDING TO *
*         (1)  LOCATION COUNTER ADDRESS                                         *
*         (2)  PAGE AND LINE NUMBER                                             *
*         (3)  BOTH                                                             *
*         FOR THE FORMAT OF THE SYMBOLIC REFERENCE TABLE REFER TO SECTION 11.8.*
*                                                                               *
*LOCATION OPERATION VARIABLE SUBFIELDS                                          *
*                                                                               *
*         XREF      STRING                                                      *
*                                                                               *
*    STRING  AN OPTIONAL CHARACTER STRING, THE FIRST CHARACTER OF WHICH IN-     *
*            DICATES HOW SYMBOLS ARE TO BE REFERENCED.                          *
*            A  THE SYMBOLIC REFERENCE TABLE LIST ADDRESSES ONLY.  FLAGS ARE    *
*               NOT INCLUDED.                                                   *
*            B  THE SYMBOLIC REFERENCE TABLE LISTS REFERENCES TO SYMBOLS ACCORD-
*               ING TO PAGE NUMBER, LINE, AND ADDRESS.  FLAGS ARE INCLUDED.     *
*            P  THE SYMBOLIC REFERENCE TABLE LISTS REFERENCES TO SYMBOLS        *
*               ACCORDING TO PAGE AND LINE NUMBER. FLAGS ARE INCLUDED.          *
*                                                                               *
*      A LOCATION FIELD SYMBOL IF PRESENT, IS IGNORED.                          *
*                                                                               *
*      IF THE STRING IS OMITTED OR IF NO XREF IS ISSUED, THE SYMBOLIC REFERENCE*
*      TABLE CONTAINS REFERENCES ACCORDING TO PAGE AND LINE NUMBERS AND INCLUDES
*      FLAGS.  THE LAST XREF ENCOUNTERED IN A SUBPROGRAM DETERMINES THE FORM OF*
*      THE LISTING FOR THE ENTIRE SUBPROGRAM.                                   *
*                                                                               *
*********************************************************************************
 USELESS  XREF     BOTH PAGE NUMBERS LINE, AND ADDRESS SHOULD BE LISTED.
```

```
  TTL PSEUDO INSTRUCTION                                       COMPASS 3.6-476.     01/09/79  08.34.39.      PAGE     11
  NOTE THIS IS A NEW TITLE

                                        TITLE  NOTE THIS IS A NEW TITLE


TTL PSEUDO INSTRUCTION                                       COMPASS 3.6-476.     01/09/79  08.34.39.      PAGE     13

                                *              NOT SELECTED, THE "CTEXT" DOES NOT AFFECT TITLING.                   *
                                *                                                                                   *
                                *              THE SUBTITLE BEGINS WITH THE FIRST NONBLANK CHARACTER FOLLOWING      *
                                *              "CTEXT" OR IN THE DEFAULT COMMENTS COLUMN (SEE COL PSEUDO IN-        *
                                *              STRUCTION) MINUS ONE, WHICHEVER COMES FIRST, AND CONTINUES TO THE*
                                *              END OF THE STATEMENT OR TO 62 CHARACTERS.  ANY CHARACTER BEYOND      *
                                *              THE 62ND CHARACTER ARE LOST.                                         *
                                *                                                                                   *
                                *************************************************************************************
                                          SPACE  10




                                          COMMENT NOTICE WHAT THIS COMMENT DOES TO THE PREVIOUS COMMENT
                                          SPACE  2

                                ************************************************************************************
                                *                                                                                  *
                                *         ENDX - DISABLE LISTING OF COMMON DECK TEXT                               *
                                *                                                                                  *
                                *         THE ENDX PSEUDO INSTRUCTION CLEARS THE "XTEXT" FLAG FOR LIST CONTROL     *
                                *         AND CAUSES LISTING TO RESUME, STARTING WITH THE INSTRUCTION AFTER        *
                                *         "ENDX", WHEN THE X LIST OPTION HAS BEEN SELECTED.                        *
                                *                                                                                  *
                                *LOCATION OPERATION VARIABLE SUBFIELDS
                                *                                                                                  *
                                *         ENDX                                                                     *
                                *                                                                                  *
                                *      ENTIRES IN THE LOCATION FIELD OR VARIABLE FIELD, IF PRESENT, ARE IGNORED.
                                *                                                                                  *
                                ************************************************************************************
                                DEFAULT SYMBOLS DEFINED BY COMPASS.

          0 X                   SYS=

          2                               END    TITLE

                      47500B CM STORAGE USED             432 STATEMENTS          4 SYMBOLS
                                MODEL 74 ASSEMBLY       0.949 SECONDS            4 REFERENCES
```

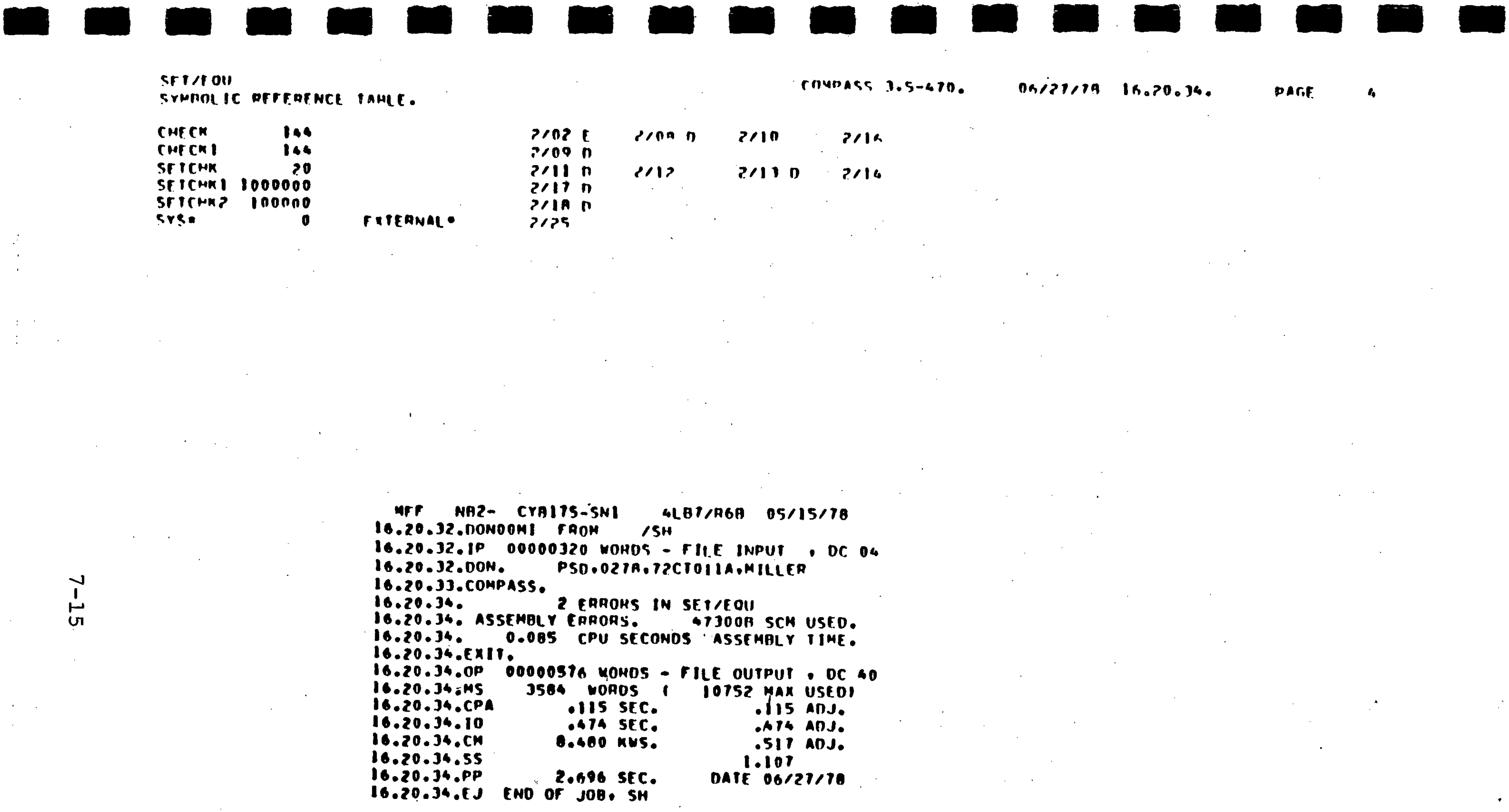
```
               TITLE
**********************************************************************
*                                                                    *
*              COMMENT-PREFIX TABLE COMMENT                          *
*                                                                    *
*              THE COMMENT PSEUDO INSTRUCTION INSERTS THE CHARACTER STRING  *
*              SPECIFIED IN THE VARIABLE FIELD INTO THE EIGHTH THROUGH FOUR-*
*              TEENTH WORD OF THE PRFX TABLE IN THE OBJECT PROGRAM.  THE    *
*              PREFIX TABLE, AND THUS THE COMMENT, IS IGNORED BY THE LOADER *
*              BUT IDENTIFIES THE SECTION.  IF A SUBPROGRAM CONTAINS MORE   *
*              THAN ONE COMMENT INSTRUCTION, THE NEW COMMENTS ARE APPENDED  *
*              TO THE TABLE FOR THE MOST RECENT BINARY CONTROL CARD.  IF THE*
*              SUBPROGRAM CONTAINS A NOLABEL INSTRUCTION, THE COMMENT INST- *
*              RUCTION IS MEANINGLESS.  COMMENT INSTRUCTIONS FOLLOWING SEG  *
*              AND BLANK IDENT PSEUDO INSTRUCTIONS ARE IGNORED WITHOUT NOTI-*
*              FICATION.                                                    *
*                                                                    *
*LOCATION OPERATION VARIABLE-SUBFIELDS                               *
*                                                                    *
*              COMMENT STRING                                        *
*                                                                    *
*       STRING  COMPASS SEARCHES THE COLUMNS FOLLOWING THE BLANK THAT TERM-
*               INATES THE OPERATION FIELD.  IF IT DOES NOT FIND A NONBLANK
*               CHARACTER BEFORE THE DEFAULT.COMMENTS COLUMN (SEE COL PSEUDO
*               INSTRUCTION), IT TAKES THE CHARACTER STARTING WITH THE DE-*
*               FAULT COMMENTS COLUMN MINUS ONE.  OTHERWISE, THE CHARACTER*
*               STRING BEGINS WITH THE FIRST NONBLANK CHARACTER FOLLOWING *
*               THE OPERATION FIELD.  IN EITHER CASE, THE LAST CHARACTER  *
*               OF THE STRING IS THE LAST NONBLANK CHARACTER OF THE STATE-*
*               MENT.  1 TO 10 BLANKS ARE APPENDED ON THE RIGHT SO THAT THE
*               STRING IS FOLLOWED BY AT LEAST ONE BLANK AND THE LENGTH OF*
*               THE STRING IS A MULTIPLE OF 10 CHARACTERS.  IF THE VARIABLE
*               AND COMMENTS FIELDS ARE ALL BLANKS, THE STRING CONSIST OF *
*               10 BLANKS.  IF THE STRING LENGTH IS MORE THAN 70 CHARACTERS
*               ALL CHARACTERS BEYOND THE 70TH ARE LOST.                  *
*                                                                    *
*              A LOCATION SYMBOL, IF PRESENT, IS IGNORED.            *
*                                                                    *
**********************************************************************
               COMMENT  THIS EXAMPLE ILLUSTRATES THE TITLE - TTL - SPACE - COMMENT **
**********************************************************************
*                                                                    *
*              CTEXT - ENABLE LISTING OF COMMON DECK TEXT            *
*                                                                    *
*              THE CTEXT PSEUDO INSTRUCTION SETS THE XTEXT FLAG FOR LIST CONTROL.  *
*                                                                    *
*            NOTE: WHEN THE FLAG IS SET, EXTERNAL TEXT IS LISTED ONLY IF THE X LIST
*                  OPTION IS SELECTED.                               *
*                                                                    *
*NAME          OPERATION VARIABLE SUBFIELDS                          *
*                                                                    *
*         NAME  IF X LIST OPTION IS SELECTED, NAME(OPTIONAL) IS TREATED AS A SUB- *
*               SUBTITLE, OTHERWISE IT IS IGNORED.                   *
*                                                                    *
*         STRING IF THE VARIABLE FIELD IS NONBLANK AND THE X LIST OPTION IS   *
*               SELECTED,THE "CTEXT" IS TREATED AS A SUBTITLE.  THE "CTEXT" IN- *
*               STRUCTION GENERATES A SUBTITLE AND CAUSES A PAFE EJECT.  IF X IS *
```

```
TTL PSEUDO INSTRUCTION                                                        COMPASS 3.6-476.     01/09/79  08.34.39.      PAGE    14
SYMBOLIC REFERENCE TABLE.

SYS=           0     EXTERNAL*      2/41         1
TITLE          0     PROGRAM*       2/02 E       0,    2/37 L         0
USELESS        2     PROGRAM*       9/11 L       2


   LOAD MAP - TITLE                                                    CYBER LOADER 1.4-485       01/09/79  08.34.41.      PAGE     1


   FWA OF THE LOAD               111
   LWA+1 OF THE LOAD             153

   TRANSFER ADDRESS -- TITLE               111

   PROGRAM ENTRY POINTS --       TITLE             111


   PROGRAM AND BLOCK ASSIGNMENTS.

   BLOCK        ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

   TITLE           111        2    LGO      01/09/79 COMPASS 3.6 476
   CPU.SYS         113       40 SL-SYSLIB   11/15/78 COMPASS 3.6 476                  THIS EXAMPLE ILLUSTRATES THE TITLE - TTL - SPAC
                                                                                      PROCESS SYSTEM REQUEST.


        .060 CP SECONDS               13500B CM STORAGE USED                          1 TABLE MOVE




                        MFS  NB1-  CYB74-SN108    5C/R0B    11/14/78
                       08.34.34.DON004U  FROM    /OH
                       08.34.34.IP  00003584 WORDS - FILE INPUT  , DC 04
                       08.34.34.DON,TS.    001A,6883,1896,MILLER
                       08.34.37.REWIND,OUTPUT.
                       08.34.38.COMPASS.
                       08.34.41. ASSEMBLY COMPLETE.   47500B  CM USED.
                       08.34.41.    1.075  CPU SECONDS  ASSEMBLY TIME.
                       08.34.41.LGO.
                       08.34.41.OP  00005696 WORDS - FILE OUTPUT , DC 40
                       08.34.42.MS    7168  WORDS  (   10752 MAX USED)
                       08.34.42.CPA      1.172 SEC.        1.172 ADJ.
                       08.34.42.CPB       .057 SEC.         .057 ADJ.
                       08.34.42.IO        .436 SEC.         .436 ADJ.
                       08.34.42.CM      35.401 KWS.        2.165 ADJ.
                       08.34.42.SS                         3.832
                       08.34.42.PP       2.850 SEC.    DATE 01/09/79
                       08.34.42.EJ  END OF JOB, OH
```

```
SET/EQU                                                      COMPASS 3.5-470.     06/27/78  16.20.34.      PAGE      1
STORAGE ALLOCATION.

            ADDRESS  LENGTH              BINARY CONTROL CARDS.

                  0        4             IDENT  SET/EQU
                  4                      END    CHECK


                               ENTRY POINTS.

                               CHECK          144


                               EXTERNAL SYMBOLS.

                               SYS=
```

```
       SET/EQU                                                     COMPASS 3.5-470.     06/27/78  16.20.34.      PAGE      2


                                                IDENT  SET/EQU
                                                ENTRY  CHECK
                                      *
                                      *
                                      *         USE OF SET AND EQU PSEUDO INSTRUCTION
                                      *
                                      *
                               144     CHECK    EQU    100
                               144     CHECK1   =      100
            0 6130000144                        SB3    CHECK
                                12     SETCHK   SET    10
                        6140000012              SB4    SETCHK
                                20     SETCHK   SET    20B
            1 6150000020                        SB5    SETCHK
D                                      CHECK    EQU    200B
                        6160000144              SB6    CHECK
                           1000000     SETCHK1  SET    262144
                            100000     SETCHK2  =      100000B
                                                COMMENT            THIS IS AN EXAMPLE OF A COMMENT
                                      *                            THIS IS AN EXAMPLE OF A COMMENT
                                                                   THIS IS AN EXAMPLE OF A COMMENT BLANK PSEUDO OP
                                50              COL    40          THIS IS AN EXAMPLE OF THE COL PSEUDO OPERATION
O           2 0000000000                                           THIS IS AN EXAMPLE OF THE COL PSEUDO OPERATION
                        7160247021              ENDRUN
            4                                   END    CHECK

                           47300B SCM STORAGE USED             28 STATEMENTS        6 SYMBOLS
                                      MODEL 174 ASSEMBLY      0.028 SECONDS        12 REFERENCES

                                               2 ERRORS IN SET/EQU
```

```
SET/EQU                                                      COMPASS 3.5-470.     06/27/78  16.20.34.      PAGE      3
ERROR DIRECTORY.

      O TYPE ERROR             OPERATION FIELD BAD.
             OCCURRED ON PAGES        2

      D TYPE ERROR             DOUBLY DEFINED SYMBOL.  THE FIRST DEFINITION HOLDS
             OCCURRED ON PAGES        2
```



```
SET/EQU                                                                      COMPASS 3.5-470.     06/27/78  16.20.34.       PAGE     4
SYMBOLIC REFERENCE TABLE.

CHECK        144                       2/02 E    2/08 D    2/10      2/16
CHECK1       144                       2/09 D
SETCHK        20                       2/11 D    2/12      2/13 D    2/14
SETCHK1 1000000                        2/17 D
SETCHK2  100000                        2/18 D
SYS=           0      EXTERNAL=        2/25
```

```
  MFF   NR2-  CYB175-SN1    4LB7/R68  05/15/78
16.20.32.DON00MI  FROM    /SH
16.20.32.IP  00000320 WORDS - FILE INPUT  , DC 04
16.20.32.DON.     PSD,027R,72CT011A,MILLER
16.20.33.COMPASS.
16.20.34.          2 ERRORS IN SET/EQU
16.20.34. ASSEMBLY ERRORS.     47300B SCM USED.
16.20.34.    0.085  CPU SECONDS  ASSEMBLY TIME.
16.20.34.EXIT.
16.20.34.OP  00000576 WORDS - FILE OUTPUT , DC 40
16.20.34.MS     3584  WORDS  (   10752 MAX USED)
16.20.34.CPA         .115 SEC.          .115 ADJ.
16.20.34.IO          .474 SEC.          .474 ADJ.
16.20.34.CM         8.480 KWS.          .517 ADJ.
16.20.34.SS                            1.107
16.20.34.PP         2.696 SEC.     DATE 06/27/78
16.20.34.EJ  END OF JOB, SH
```

```
                                                IDENT  REPEAT
                                                LIST   D,M,S,X
                                                ENTRY  BETTY
                                                REP    S/SAM,D/TOM,B/5,C/2         REP PSEUDO OP
  0  00000000000000000001       SAM     DATA   1,2,3,4,5
  1  00000000000000000002
  2  00000000000000000003
  3  00000000000000000004
  4  00000000000000000005
  5  5120000064                 BETTY   SA2    64B
               74620                    SX6    A2
                    10722               BX7    X2
  6  5160000000 C                       SA6    ANS
               5170000001 C             SA7    ANS+1
  7  5130000012 +               LOOP    SA3    WORD
               10733                    BX7    X3
 10  5170000001                         SA7    1
               5140000001               SA4    1
 11  0314000007 +                       NZ     X4,LOOP
 12  05160460000000000000       WORD    VFD    18/3LEND,3/6,39/0
 13  01020304050607101112               DATA   10LABCDEFGHIJ
                    D-D                 BASE   D
 14                 12          TOM     BSSZ   10
                                        USE    /ANS/        USE LABELED COMMON
  0                 10          ANS     BSSZ   8
                                        USE    //           USE BLANK COMMON
  0                 10                  BSS    8
 26                                     END    BETTY

                 47300B SCM STORAGE USED              24 STATEMENTS          6 SYMBOLS
                       MODEL 174 ASSEMBLY          0.029 SECONDS           13 REFERENCES
```

REP

```
DUMP    RELATIVE                         DMP(111,177)

    00111    00000 00000 00000 00064     14071 70000 00000 00000     00000 00000 00000 00000   00121-00000 00000 00000 00001

    00122    00000 00000 00000 00002     00000 00000 00000 00003
    00124    00000 00000 00000 00004     00000 00000 00000 00005     51200 00064 74620 10722       51600 00111 51700 00112

    00130    51300 00133 10733 46000     51700 00001 51400 00001     03140 00130 61000 46000       05160 46000 00000 00000

    00134    01020 30405 06071 01112     00000 00000 00000 00001     00000 00000 00000 00002       00000 00000 00000 00003

    00140    00000 00000 00000 00004     00000 00000 00000 00005     00000 00000 00000 00001       00000 00000 00000 00002

    00144    00000 00000 00000 00003     00000 00000 00000 00004     00000 00000 00000 00005       60000 00000 04004 00147

    00177   >60000 00000 04004 00177
```

```
REPEAT                                                          COMPASS 3.5-470.     06/27/78  16.28.10.      PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH                 BINARY CONTROL CARDS.

                0       26                 IDENT  REPEAT
               26                          END    BETTY

                                   BLOCKS     TYPE       ADDRESS    LENGTH

                                   PROGRAM*   LOCAL            0        26
                                   ANS        COMMON           0        10
                                   //         COMMON           0        10

                                   ENTRY POINTS.

                                   BETTY              5*
```

```
LOAD MAP - REPEAT                                          CYBER LOADER 1.4-470       06/27/78  16.28.10.      PAGE      1


FWA OF THE LOAD              111
LWA+1 OF THE LOAD            157

TRANSFER ADDRESS -- BETTY           126

PROGRAM ENTRY POINTS --       REPEAT           126


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS   LENGTH     FILE     DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

/ANS/            111       10
REPEAT           121       26     LGO     06/27/78 COMPASS 3.5 470
 //              147       10


     .012 CP SECONDS              132008 CM STORAGE USED
```

```
  MFF   NB2-  CYB175-SN1    4LB7/R6B  05/15/78
16.28.08.DON00MR  FROM    /SH
16.28.08.IP  00000320 WORDS - FILE INPUT  , DC 04
16.28.08.DON.      PSD,0278,72CT011A,MILLER
16.28.10.COMPASS.
16.28.10. ASSEMBLY COMPLETE.   47300B SCM USED.
16.28.10.    0.083  CPU SECONDS  ASSEMBLY TIME.
16.28.10.LOAD(LGO)
16.28.10.EXECUTE(BETTY)
16.28.11.DMP(111,177)
16.28.11.OP  00000768 WORDS - FILE OUTPUT , DC 40
16.28.11.MS     3584  WORDS  (   10752 MAX USED)
16.28.11.CPA         .120 SEC.          .120 ADJ.
16.28.11.IO          .609 SEC.          .609 ADJ.
16.28.11.CM        9.840 KWS.          .600 ADJ.
16.28.11.SS                            1.330
16.28.11.PP        3.316 SEC.     DATE 06/27/78
16.28.11.EJ  END OF JOB, SH
```

```
                                                    IDENT  LOC
                                                    ENTRY  SAM
      0                      12                     BSSZ   10
     12  00000000000000000001         GAM           DATA   1,2,3,4
     16  46000                        SAM           NO
L    12                                             LOC    GAM
L    12  5110000000                                 SA1    0
                   5120000001                       SA2    1
L    13  5130000013 +                               SA3    *
                   5140000015 +                     SA4    *+2
                                                    USE    *
     21  10611                                      BX6    X1
              74630                                 SX6    A3
                   74740                            SX7    A4
     22  5160000033 +                               SA6    ANS+6
                   5170000034 +                     SA7    ANS+7
     23  7160247021                                 ENDRUN
     25                      12       ANS           BSSZ   10
     37                                             END    SAM

                    47300B SCM STORAGE USED          22 STATEMENTS         4 SYMBOLS
                          MODEL 174 ASSEMBLY      0.024 SECONDS            8 REFERENCES
```

```
DUMP    RELATIVE                     DMP(111,147)

   00111    00000 00000 00000 00000   00123-00000 00000 00000 00001                                      46000 46000 61000 46000
   00124    00000 00000 00000 00002         00000 00000 00000 00003      00000 00000 00000 00004

   00130    51100 00000 51200 00001         51300 00124 51400 00126      10611 74630 74740 46000         51600 00144 51700 00145

   00134    71602 47021 20650 46000         01000 00152 61000 46000      00000 00000 00000 00000   00144-00000 00000 00000 00124

   00145    00000 00000 00000 00126         00000 00000 00000 00000   00147-00000 00000 00000 00000
```

```
LOC                                                        COMPASS 3.5-470.     06/27/78  09.04.34.      PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH              BINARY CONTROL CARDS.

                0       37              IDENT  LOC
               37                       END    SAM

                                   ENTRY POINTS.

                                   SAM               16+

                                   EXTERNAL SYMBOLS.

                                   SYS=


  LOAD MAP - LOC                                        CYBER LOADER 1.4-470       06/27/78  09.04.35.      PAGE      1


  FWA OF THE LOAD             111
  LWA+1 OF THE LOAD           210

  TRANSFER ADDRESS -- SAM              127

  PROGRAM ENTRY POINTS --       LOC                127


  PROGRAM AND BLOCK ASSIGNMENTS.

  BLOCK       ADDRESS   LENGTH    FILE     DATE     PROCSSR VER LEVEL  HARDWARE   COMMENTS

  LOC             111       37    LGO     06/27/78 COMPASS 3.5 470
  SYS.RM          150       40 SL-SYSLIB  05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.


       .025 CP SECONDS              13200B CM STORAGE USED
                                                                     MFF   NB2-  CYB175-SN1    4LB7/868  05/15/78
                                                                   09.04.32.DON005U  FROM     /SH
                                                                   09.04.32.IP  00000256 WORDS - FILE INPUT  , DC 04
                                                                   09.04.32.DON.     PSD,0278,72CT011A,MILLER
                                                                   09.04.34.COMPASS.
                                                                   09.04.34. ASSEMBLY COMPLETE.   47300B SCM USED.
                                                                   09.04.34.     0.073  CPU SECONDS  ASSEMBLY TIME.
                                                                   09.04.34.LGO.
                                                                   09.04.35.DMP(111,147)
                                                                   09.04.35.OP  00000704 WORDS - FILE OUTPUT , DC 40
                                                                   09.04.35.MS     3584  WORDS  (     7168 MAX USED)
                                                                   09.04.35.CPA          .132 SEC.           .132 ADJ.
                                                                   09.04.35.IO           .628 SEC.           .628 ADJ.
                                                                   09.04.35.CM          9.987 KWS.           .609 ADJ.
                                                                   09.04.36.SS                              1.371
                                                                   09.04.36.PP          3.537 SEC.     DATE 06/27/78
                                                                   09.04.36.EJ  END OF JOB, SH
```

```
CHECK1                                                                       COMPASS 3.5-470.     06/27/78  16.28.34.      PAGE      1
STORAGE ALLOCATION.

           ADDRESS   LENGTH               BINARY CONTROL CARDS.

                 0        7               IDENT   CHECK1
                 7                        END     START1

                                 ENTRY POINTS.

                                 START1           2+            TAG             0+

                                 EXTERNAL SYMBOLS.

                                 SYS=      START2



CHECK1                                                                       COMPASS 3.5-470.     06/27/78  16.28.34.      PAGE      3
SYMBOLIC REFERENCE TABLE.

HOLD          6     PROGRAM*       2/34 S
START1        2     PROGRAM*       2/02 E     2/32 L
START2        0     EXTERNAL*      2/35
SYS=          0     EXTERNAL*      2/39
TAG           0     PROGRAM*       2/02 E     2/30 L     2/32
```

```
CHECK1                                                          COMPASS 3.5-470.     06/27/78  16.28.34.      PAGE      2

                                                 IDENT  CHECK1
                                                 ENTRY  START1,TAG
                                                 LIST   *
                                         ***************************************************************************
                                         *                                                                         *
                                         *       DEFAULT SYMBOLS                                                   *
                                         *                                                                         *
                                         *  WHEN A SYMBOL REFERENCE IS PRECEDED BY =S OR =X AND THE SYMBOL IS NOT DEFINED
                                         *  IN THE SUBPROGRAM, COMPASS DEFINES THE SYMBOL OR DECLARES IT AS AN EXTERNAL *
                                         *  SYMBOL, RESPECTIVELY, AT THE END OF ASSEMBLY.  THE =X FORM IS DEFINED BY    *
                                         *  DEFAULT IN RELOCATEABLE ASSEMBLIES ONLY.                               *
                                         *                                                                         *
                                         *  =SSYMBOL  IF SYMBOL IS NOT DEFINED, COMPASS ASSIGNS AN ADDRESS AT THE END OF*
                                         *            THE ZERO BLOCK.  ALL SUBSEQUENT REFERENCES TO THE SYMBOL, WHETHER *
                                         *            PRECEDED BY =S OR NOT, ARE TO THE LOCATION OF THE WORD.  A DEFAULT*
                                         *            SYMBOL CANNOT BE USED WHERE A PREVIOUSLY DEFINED SYMBOL IS REQUIRED
                                         *                                                                         *
                                         *            IF THE SYMBOL IS DEFINED BY CONVENTIONAL METHOD, COMPASS DOES NOT *
                                         *            DEFINE IT AGAIN BUT USES THE PROGRAMMER DEFINITION.          *
                                         *                                                                         *
                                         *  =XSYMBOL  THIS OPTION PERMITS  A PROGRAMMER TO DEFINE HIS SYMBOLS IN A SUB- *
                                         *            ROUTINE OR LINK TO THEM IN ANOTHER SUBPROGRAM.  IF THE PROGRAMMER *
                                         *            DEFINES THE SYMBOL, THE ASSEMBLER USES THE PROGRAM DEFINITION.  IF*
                                         *            THE PROGRAMMER DOES NOT DEFINE THE SYMBOL, THE ASSEMBLER ASSUMES  *
                                         *            THAT THE SYMBOL IS EXTERNAL AS THOUGH DECLARED IN AN *EXT* PSEUDO *
                                         *            INSTRUCTION.  A SYMBOL PREFIXED BY =X MUST CONFORM TO THE REQUIRE-*
                                         *            MENTS FOR EXTERNAL SYMBOLS.                                  *
                                         *                                                                         *
                                         ***************************************************************************
 0  77777777777777777777          TAG       DATA   -0,-1
 1  77777777777777777776
 2  5110000000 +                  START1    SA1    TAG
            10711                           BX7    X1            TRANSMIT (X1) TO X7
 3  5170000006 +                            SA7    =SHOLD
            0200000000 X                    JP     =XSTART2
 4                                          ENDRUN
 4  7160247021                     SX6 3REND*4+1                                                           ENDRUN   .1
            20650                  LX6 40D                                                                 ENDRUN   .1
 5  0100000000 X                   RJ =XSYS=                                                               ENDRUN   .1
                                   ENDM                                                                    ENDRUN   .1

                                 DEFAULT SYMBOLS DEFINED BY COMPASS.

 6                               HOLD
 0 X                             START2
 0 X                             SYS=

 7                                          END    START1

                      50400B SCM STORAGE USED            40 STATEMENTS          5 SYMBOLS
                             MODEL 174 ASSEMBLY       0.058 SECONDS             8 REFERENCES
```

```
CHECK2                                                   COMPASS 3.5-470.     06/27/78  16.28.34.      PAGE     1
STORAGE ALLOCATION.

          ADDRESS   LENGTH             BINARY CONTROL CARDS.

                0        4             IDENT   CHECK2
                4                      END

                                 ENTRY POINTS.

                                 START2          0*

                                 EXTERNAL SYMBOLS.

                                 SYS=      TAG
```

```
CHECK2                                                   COMPASS 3.5-470.     06/27/78  16.28.34.      PAGE     2

                                              IDENT   CHECK2
                                              ENTRY   START2
      0  5110000001 X            START2       SA1     =XTAG+1
                   10711                      BX7     X1
      1  5170000003 +                         SA7     SAVE
                   7160247021                 ENDRUN
      3                     1    SAVE         BSSZ    1
      4                                       END

                   47300B SCM STORAGE USED            11 STATEMENTS          4 SYMBOLS
                          MODEL 174 ASSEMBLY       0.015 SECONDS             6 REFERENCES
```

```
CHECK2                                                   COMPASS 3.5-470.     06/27/78  16.28.34.      PAGE     3.
SYMBOLIC REFERENCE TABLE.

SAVE          3      PROGRAM*        2/05 S     2/07 L
START2        0      PROGRAM*        2/02 E     2/03 L
SYS=          0      EXTERNAL*       2/07
TAG           0      EXTERNAL*       2/03
```

```
LOAD MAP - CHECK1                                          CYBER LOADER 1.4-470        06/27/78  16.28.35.       PAGE      1


FWA OF THE LOAD             111
LWA+1 OF THE LOAD           164

TRANSFER ADDRESS -- START1          113

PROGRAM ENTRY POINTS --        CHECK1          113


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS  LENGTH     FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

CHECK1          111        7    LGO      06/27/78 COMPASS 3.5 470
CHECK2          120        4    LGO      06/27/78 COMPASS 3.5 470
SYS.RM          124       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                   PROCESS SYSTEM REQUEST.


      .025 CP SECONDS              132008 CM STORAGE USED                         1 TABLE MOVE

DUMP    RELATIVE                          DMP(111,127)

   00111   77777 77777 77777 77777         77777 77777 77777 77776         51100 00111 10711 46000
   00114   51700 00117 02000 00120         71602 47021 20650 46000         01000 00126 61000 46000         77777 77777 77777 77777

   00120   51100 00112 10711 46000         51700 00123 71602 47021         20650 01000 00126 46000         77777 77777 77777 77776

   00124   04000 00137 00000 00000         01300 00000 00000 00000         04000 00123 00000 00000         51100 00001 03110 00127


                                                                    MFF   NO2-  CY8175-SN1    4L87/R68  05/15/78
                                                                   16.28.32.DON00MS  FROM    /SH
                                                                   16.28.32.IP  00000512 WORDS - FILE INPUT  , DC 04
                                                                   16.28.32.DON.      PSD,0278,72CT011A,MILLER
                                                                   16.28.33.COMPASS.
                                                                   16.28.34. ASSEMBLY COMPLETE.   50400B SCM USED.
                                                                   16.28.34.     0.130  CPU SECONDS  ASSEMBLY TIME.
                                                                   16.28.34.LGO.
                                                                   16.28.35.DMP(111,127)
                                                                   16.28.35.OP  00001216 WORDS - FILE OUTPUT , DC 40
                                                                   16.28.36.MS    3584  WORDS  (    7168 MAX USED)
                                                                   16.28.36.CPA        .182 SEC.          .181 ADJ.
                                                                   16.28.36.IO         .642 SEC.          .642 ADJ.
                                                                   16.28.36.CM       11.400 KWS.          .695 ADJ.
                                                                   16.28.36.SS                           1.520
                                                                   16.28.36.PP        3.093 SEC.     DATE 06/27/78
                                                                   16.28.36.EJ  END OF JOB, SH
   PROGRAM INITIAL    74/74    OPT=1                            FTN 4.7+485        01/04/79  12.26.38        PAGE     1

                     PROGRAM INITIAL(OUTPUT)
                     COMMON /ANSWER/ BUFFER(16)
                     CALL START
                     PRINT 101
                     PRINT 100,(BUFFER(I),I=1,16)
                 100 FORMAT(4O22)
                 101 FORMAT(1H1)
                     END
```

```
   SUBROUTINE ERRMSG1     74/74    OPT=1                        FTN 4.7+485        01/04/79  12.26.38        PAGE     1

                     SUBROUTINE ERRMSG1
                     PRINT 100
                 100 FORMAT(1H1,* ECS WRITE ERROR*)
                     RETURN
                     END
```

```
   SUBROUTINE ERRMSG2     74/74    OPT=1                        FTN 4.7+485        01/04/79  12.26.38        PAGE     1

                     SUBROUTINE ERRMSG2
                     PRINT 100
                 100 FORMAT(1H1,* ECS READ ERROR*)
                     RETURN
                     END
```

```
                                                       IDENT   LOC7600
                                                       LIST    -B,-R
                                                       ENTRY   START
                                                       EXT     ERRMSG1,ERRMSG2
                                                       USE     /ANSWER/
        0                        20          ANSWER    BSSZ    16
                                                       USE     0
        0   00000000000000000001             OPERAND   DATA    1,2,3,4,5,6,7,8
       10   00000000000000000000             START     DATA    0
       11   5100000015 +                               SA0     ADD
                       7100000000                      SX0     0
       12   0120000044                                 WE      LBLOCKT
                       0100000000 X                    RJ      ERRMSG1
       13   0110000012                                 RE      LBLOCK1
                       0100000000 X                    RJ      ERRMSG2
       14   0400000015 +                               EQ      ADD
       15   5110000000 +                     ADD       SA1     OPERAND+0
                       5120000001 +                    SA2     OPERAND+1
       16   5130000002 +                               SA3     OPERAND+2
                       5140000003 +                    SA4     OPERAND+3
       17   36612                                      IX6     X1+X2
                 36734                                 IX7     X3+X4
                       5160000000 C                    SA6     ANSWER+0
       20   5170000001 C                               SA7     ANSWER+1
                       5110000004 +                    SA1     OPERAND+4
       21   5120000005 +                               SA2     OPERAND+5
                       5130000006 +                    SA3     OPERAND+6
       22   5140000007 +                               SA4     OPERAND+7
                       36612                           IX6     X1+X2
                            36734                      IX7     X3+X4
       23   5160000002 C                               SA6     ANSWER+2
                       5170000003 C                    SA7     ANSWER+3
       24   5100000015 +                               SA0     ADD
                       7100000012                      SX0     LBLOCK1
       25   0110000010                                 RE      LBLOCK2
                       0100000000 X                    RJ      ERRMSG2
       26   0100000015 +                     END1BLK   RJ      ADD
L      15                                              LOC     ADD
L      15   00000000000000000000             SUBT      DATA    0
L      16   5110000000 +                               SA1     OPERAND+0
                       5120000001 +                    SA2     OPERAND+1
L      17   5130000002 +                               SA3     OPERAND+2
                       5140000003 +                    SA4     OPERAND+3
L      20   37621                                      IX6     X2-X1
                 37734                                 IX7     X3-X4
                       5160000004 C                    SA6     ANSWER+4
L      21   5170000005 C                               SA7     ANSWER+5
                       5100000015 +                    SA0     ADD
L      22   7100000022                                 SX0     LBLOCK1+LBLOCK2
L      23   0110000010                                 RE      LBLOCK3
                       0100000000 X                    RJ      ERRMSG2
L      24   0100000015 +                     END2BLK   RJ      ADD
L      15                                              LOC     ADD
L      15   00000000000000000000             MULT      DATA    0
```

```
  L    16 5110000000 +                SA1  OPERAND+0
                  5120000001 +        SA2  OPERAND+1
  L    17 5130000002 +                SA3  OPERAND+2
                  5140000003 +        SA4  OPERAND+3
  L    20 42612                       IX6  X1*X2
              42734                   IX7  X3*X4
                  5160000006 C        SA6  ANSWER+6
  L    21 5170000007 C                SA7  ANSWER+7
                  5100000015 +        SA0  ADD
  L    22 7100000032                  SX0  LBLOCK1+LBLOCK2+LBLOCK3
  L    23 0110000012                  RE   LBLOCK4
                  0100000000 X        RJ   ERRMSG2
  L    24 0100000015 +        END3BLK RJ   ADD
  L    15                             LOC  ADD
  L    15 00000000000000000000DIVIDE  DATA 0
  L    16 5110000000 +                SA1  OPERAND+0
                  5120000001 +        SA2  OPERAND+1
  L    17 5130000002 +                SA3  OPERAND+2
                  5140000003 +        SA4  OPERAND+3
  L    20 27101                       PX1  X1
              27202                   PX2  X2
                  24101               NX1  X1
                      24202           NX2  X2
  L    21 44621                       FX6  X2/X1
              27303                   PX3  X3
                  27404               PX4  X4
                      24303           NX3  X3
  L    22 24404                       NX4  X4
              44743                   FX7  X4/X3
                  5160000010 C        SA6  ANSWER+8
  L    23 5170000011 C                SA7  ANSWER+9
                  26666               UX6  X6,B6
                      26777           UX7  X7,B7
  L    24 22666                       LX6  X6,B6
              22777                   LX7  X7,B7
                  5160000012 C        SA6  ANSWER+10
  L    25 5170000013 C                SA7  ANSWER+11
  L    26 0400000010 +        END4BLK EQ   START
                        12    LBLOCK1 EQU  END1BLK-ADD+1
                        10    LBLOCK2 EQU  END2BLK-SUBT+1
                        10    LBLOCK3 EQU  END3BLK-MULT+1
                        12    LBLOCK4 EQU  END4BLK-DIVIDE+1
                        44    LBLOCKT EQU  LBLOCK1+LBLOCK2+LBLOCK3+LBLOCK4
                        12    MAXBLK  MAX  LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4
                        10    MINBLK  MIN  LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4
       61                             END

           1100B ECS  46700B CM  STORAGE USED           100 STATEMENTS       20 SYMBOLS
                                 MODEL 73 ASSEMBLY     0.347 SECONDS          0 REFERENCES
```

```
 LOAD MAP - INITIAL                                        CYBER LOADER 1.4-485       01/04/79  12.31.17.       PAGE     1


 FWA OF THE LOAD            111
 LWA+1 OF THE LOAD         7166

 TRANSFER ADDRESS -- INITIAL          2213

 PROGRAM ENTRY POINTS --       INITIAL         2213


 PROGRAM AND BLOCK ASSIGNMENTS.

 BLOCK         ADDRESS   LENGTH     FILE        DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

 /ANSWER/          111       20
 INITIAL           131     2105     LGO       01/04/79 FTN       4.7 485     666X I     PROGRAM   OPT=1
 ERRMSG1          2236       14     LGO       01/04/79 FTN       4.7 485     666X I     SUBROUTINEOPT=1
 ERRMSG2          2252       14     LGO       01/04/79 FTN       4.7 485     666X I     SUBROUTINEOPT=1
 LOC7600          2266       61     LGO       01/04/79 COMPASS 3.6 476
 /STP.END/        2347        1
 /FCL.C./         2350       26
 /QB.IO./         2376      101
 Q2NTRY=          2477        0 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  FCL INITIALIZATION ROUTINE.
 /FCL=ENT/        2477       40
 COMIO=           2537       33 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  COMMON CODED I/O ROUTINES AND CONSTANTS.
 FCL=FDL          2572       40 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  FCL CAPSULE LOADING
 FEIFST=          2632        3 SL-FORTRAN  11/16/78 COMPASS 3.6 485                   CONVERTED DATA STORAGE
 FLTOUT=          2635      311 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  COMMON FLOATING OUTPUT CODE
 FORSYS=          3146      301 SL-FORTRAN  11/15/78 COMPASS 3.6 485                  FORTRAN OBJECT LIBRARY UTILITIES.
 OUTCOM=          3447     -154 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  COMMON OUTPUT CODE
 SYSAID=          3623        1 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
 FECMSK=          3624       41 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  INITIALIZE CONSTANTS.
 FMTAP=           3665      357 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  CRACK APLIST AND FORMAT FOR KODER/KRAKER.
 FORUTL=          4244       46 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  FCL MISC. UTILITIES.
 GETFIT=          4312       57 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  LOCATE AN FIT GIVEN A FILE NAME.
 KODER=           4371      451 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  OUTPUT FORMAT INTERPRETER.
 OUTC=            5042      150 SL-FORTRAN  11/16/78 COMPASS 3.6 485                  FORMATTED WRITE FORTRAN RECORD.
 /FDL.COM/        5212       14
 FDL.RES          5226      211 SL-SYSLIB   11/02/78 COMPASS 3.6 485                  FAST DYNAMIC LOADER RESIDENT.
 FDL.MMI          5437      222 SL-SYSLIB   11/02/78 COMPASS 3.6 485                  FDL MEMORY MANAGER INTERFACE.
 CPU.SYS          5661       40 SL-SYSLIB   11/15/78 COMPASS 3.6 476                  PROCESS SYSTEM REQUEST.
 CMF.ALF          5721      160 SL-SYSLIB   11/15/78 COMPASS 3.6 485                  CMM V1.1 - ALLOCATE FIXED.
 CMF.CSF          6101        6 SL-SYSLIB   11/15/78 COMPASS 3.6 485                  CMM V1.1 - CHANGE SPECS FIXED.
 CMM.FFA          6107       14 SL-SYSLIB   11/15/78 COMPASS 3.6 485                  CMM V1.1 - FIXED FREE ALGORITHM.
 CMF.FRF          6123       36 SL-SYSLIB   11/15/78 COMPASS 3.6 485                  CMM V1.1 - FREE FIXED.
 CMM.R            6161      214 SL-SYSLIB   11/15/78 COMPASS 3.6 485                  CMM V1.1 - RESIDENT SUBROUTINES.
 CMF.SLF          6375       22 SL-SYSLIB   11/15/78 COMPASS 3.6 485                  CMM V1.1 - SHRINK AT LWA FIXED.
 CTL$RM           6417      433 SL-SYSLIB   11/16/78 COMPASS 3.6 485                  CRM CONTROLLING ROUTINE.
 ERR$RM           7052       25 SL-SYSLIB   11/16/78 COMPASS 3.6 485                  CRM ERROR PROCESSOR ENTRY.
 LIST$RM          7077       67 SL-SYSLIB   11/16/78 COMPASS 3.6 485                  CRM - ALLOCATE SPACE FOR LIST OF FILES


      .330 CP SECONDS                    23600B CM STORAGE USED                    12 TABLE MOVES
```

```
00000000000000000003  00000000000000000007  00000000000000000013  00000000000000000017
00000000000000000001  77777777777777777776  00000000000000000002  00000000000000000014
17214000000000000000  17205252525252525252  00000000000000000002  00000000000000000001
00000000000000000000  00000000000000000000  00000000000000000000  00000000000000000000
```

```
  MFS  NHI-  CYB74-SN108    5C/ROB    11/14/78
12.26.16.DONOOHT  FROM    /OH
12.26.16.IP  00000768 WORDS - FILE INPUT  , DC 04
12.26.16.DON,ECI.  001A,6883,1896,MILLER
12.26.38.HEWIND,OUTPUT.
12.26.38.FTN,R=0.
12.31.14.      .706 CP SECONDS COMPILATION TIME
12.31.14.MAP,PART.
12.31.15.LGO.
12.31.19.     END  INITIAL
12.31.19.      .069 CP SECONDS EXECUTION TIME
12.31.19.EXIT.
12.31.19.OP  00001536 WORDS - FILE OUTPUT , DC 40
12.31.19.MS    3584  WORDS  (   14336 MAX USED)
12.31.19.CPA        .764 SEC.         .764 ADJ.
12.31.19.CPB        .453 SEC.         .453 ADJ.
12.31.20.IO        2.081 SEC.        2.081 ADJ.
12.31.20.CM       62.324 KWS.        3.803 ADJ.
12.31.20.EC        1.685 KWS.         .051 ADJ.
12.31.20.SS                          7.154
12.31.20.PP       15.234 SEC.     DATE 01/04/79
12.31.20.EJ  END OF JOB, OH
```

```
                                                 IDENT  ERRXX
                                                 ENTRY  ERRXX
                                       ********************************************************************************
                                       *                                                                              *
                                       *         ERRXX - CONDITIONALLY SET ERROR FLAG                                 *
                                       *                                                                              *
                                       *         AN ERRXX PSEUDO INSTRUCTION PRODUCES AN ASSEMBLY ERROR WHEN A CON-   *
                                       *         DITION DETECTED DURING THE SECOND PASS OF THE ASSEMBLER IS TRUE.     *
                                       *                                                                              *
                                       *LOCATION ERRXX  AEXP                                                          *
                                       *                                                                              *
                                       *      FLAG  A SINGLE ALPHANUMERIC CHARACTER DENOTING THE ERROR TYPE.  THE FLAG*
                                       *            IS PLACED IN THE LISTING TO THE LEFT OF THE LINE FOR THE "ERR".   *
                                       *            THE FLAG CAN DENOTE A FATAL OR NONFATAL ERROR.  A FATAL ERROR     *
                                       *            CAUSES "COMPASS" TO SUPPRESS GENERATION OF THE BINARY DECK UNLESS *
                                       *            THE "D" MODE OPTION IS SELECTED ON THE "COMPASS" CONTROL CARD.  IF*
                                       *            NO FLAG IS SPECIFIED, OR THE CHARACTER IS NOT ONE OF THOSE GIVEN  *
                                       *            IN SECTION 11.9 OF THE REFERENCE MANUAL, "COMPASS" USES "P".      *
                                       *                                                                              *
                                       *            XX    DEFINES CONDITION UNDER WHICH AEXP VALUE IS ERRONEOUS.      *
                                       *                                                                              *
                                       *                  XX        ERROR CONDITION                                   *
                                       *                  NG OR MI  VALUE OF EXPRESSION IS NEGATIVE                   *
                                       *                  NZ        VALUE OF EXPRESSION IS NONZERO                    *
                                       *                  PL        VALUE OF EXPRESSION IS POSITIVE                   *
                                       *                  ZR        VALUE OF EXPRESSION IS ZERO                       *
                                       *                                                                              *
                                       *      AEXP  ABSOLUTE EXPRESSION.  IT CANNOT CONTAIN EXTERNAL SYMBOLS OR REFER-*
                                       *            ENCES TO BLANK COMMON.  THE TEST IS MADE IN PASS TWO OF THE AS-   *
                                       *            SEMBLER.  RELOCATEABLE ADDRESSES ARE ASSIGNED VALUES RELATIVE TO  *
                                       *            PROGRAM ORIGIN RATHER THAN TO THE BLOCK IN WHICH THEY ARE DEFINED.*
                                       *                                                                              *
                                       *                                      NOTE:                                   *
                                       *                                                                              *
                                       *            ERRXX IS THE ONLY CONDITIONAL INSTRUCTION FOR WHICH THE TEST IS   *
                                       *            MADE IN PASS TWO.  THEREFORE, THIS IS THE ONLY PSEUDO INSTRUCTION *
                                       *            THAT CAN BE USED TO DETERMINE PPU OVERFLOW IF THE PPU PROGRAM HAS *
                                       *            LITERALS AND "USE" BLOCKS.                                        *
                                       *                                                                              *
                                       ********************************************************************************
            0  00000000000000000000    ERRXX     DATA   0
                                  1     B        EQU    1
                                 -1     C        =      -1
                                  0     D        =      0
P                                -1     Z        ERRNG  C
P                                -1     Y        ERRMI  C
P                                 1     X        ERRNZ  B
P                                 1     W        ERRPL  B
6                                 0     6        ERRZR  D
P                                27     Q        ERRPL  BUFF-ERRXX-120B
            1  7160247021                        ENDRUN
            3                   144     BUF      BSS    100
          147                           BUFF     BSS    0
          147                                    END

                            47500B CM  STORAGE USED               57 STATEMENTS        7 SYMBOLS
                                       MODEL 74 ASSEMBLY       0.206 SECONDS           0 REFERENCES
```

```
ERRXX                                                  COMPASS 3.6-476.     01/09/79  12.27.56.      PAGE      2

                                   5 ERRORS IN ERRXX
```

```
ERRXX                                                  COMPASS 3.6-476.     01/09/79  12.27.56.      PAGE      3
ERROR DIRECTORY.

    P TYPE ERROR              CONSULT LISTINGS FOR REASON BEHIND P-ERROR.
           OCCURRED ON PAGES          1

    6 TYPE ERROR              LOCATION FIELD MEANINGLESS.
           OCCURRED ON PAGES          1
```

```
  MFS  N81-  CYB74-SN108    SC/ROB    11/14/78
12.27.53.DON00FL  FROM    /OH
12.27.53.IP  00000512 WORDS - FILE INPUT  , DC 04
12.27.53.DON,TS.   001A,6883,1896,MILLER
12.27.55.REWIND,OUTPUT.
12.27.55.COMPASS,LO=BR..
12.27.57.          1 WARNING MESSAGE IN  ERRXX
12.27.57.          5 ERRORS IN ERRXX
12.27.57. ASSEMBLY ERRORS.     47500B  CM USED.
12.27.57.    0.385  CPU SECONDS  ASSEMBLY TIME.
12.27.57.OP  00000704 WORDS - FILE OUTPUT , DC 40
12.27.57.MS    3584  WORDS  (   10752 MAX USED)
12.27.57.CPA       .291 SEC.          .291 ADJ.
12.27.57.CPB       .174 SEC.          .174 ADJ.
12.27.57.IO        .305 SEC.          .305 ADJ.
12.27.57.CM      14.978 KWS.          .914 ADJ.
12.27.57.SS                          1.685
12.27.57.PP       3.370 SEC.     DATE 01/09/79
12.27.57.EJ  END OF JOB, OH
```

LESSON 8

ERROR EXITS

LESSON PREVIEW:

THE VARIOUS ERROR EXITS ARE DETECTED IN THE HARDWARE AS WELL AS SOFTWARE.

REFERENCES:

CHAPTER II	COMPASS	#60492600

TRAINING AIDS:

VISUAL SET V8
PROGRAM COMPILATION & EXECUTION LISTING OF DECK 8A WITH DUMP.

PROJECTS:

HOMEWORK

OBJECTIVES:

AT THE COMPLETION OF THIS LESSON THE STUDENT WILL BE ABLE TO:

1. DETERMINE THE CAUSE OF EACH ERROR MADE.
2. LOCATE ITEMS INVOLVED WITH THE ERROR IN THE DUMP.
3. UTILIZE THE EXCHANGE JUMP PACKAGE SOLVING THE ERROR MADE PROBLEM.

SPECIAL FORMS

CYBER FLOATING POINT HARDWARE IS VERY SPECIFIC IN THE VALUES RETURNED AS A RESULT OF ARITHMETIC OPERATIONS WHICH ATTEMPT TO GENERATE INDEFINITE OR OUT-OF-RANGE NUMBERS.

UNDERFLOW

0 0 0 0 EXPONENT (OR 7 7 7 7)

INFINITY

3 7 7 7 EXPONENT (OR 4 0 0 0)

INDEFINITE

1 7 7 7 EXPONENT (OR 6 0 0 0)

NOTES ON "SPECIAL FORMS"

THESE ARE THE OUT-OF-RANGE AND INDEFINITE NUMBERS GENERATED AS A RESULT OF FLOATING POINT OPERATIONS.

INFINITE - THE NUMBER IS TOO LARGE (OVERFLOW)
ZERO FORM - THE NUMBER IS TOO SMALL (UNDERFLOW)
INDEFINITE - THE COMPUTER DOES NOT KNOW WHAT RANGE THE OPERAND LIES IN (FOR EXAMPLE, IT MAY HAVE BEEN PRODUCED AS A RESULT OF USING AN INFINITE OPERAND)

SPECIAL FORMS MAY BE HANDLED IN THREE WAYS:

1. CHECK THE OPERAND WHEN IT IS GENERATED BY USING JUMPS

   IR, OR } INFINITY TESTS
   DF, ID } INDEFINITE TESTS
   ZR, NZ } UNDERFLOW TESTS

OR

2. ALLOW THE HARDWARE TO ABORT THE JOB WHEN AN ATTEMPT IS MADE TO USE A SPECIAL FORM BY SELECTING AN EXIT MODE. (THE EXIT MODES WILL CATCH USE OF INFINITE AND INDEFINITE FORMS. THEY WILL NOT CATCH UNDERFLOW.)

OR

3. IGNORE THE ERROR. BY DESELECTING THE EXIT MODE, THE INFINITE OR INDEFINITE OPERANDS MAY BE USED IN SUBSEQUENT OPERATIONS. THIS OF COURSE WILL PERPETUATE ERRORS.

READ THE INFORMATION ABOUT EXIT MODE AND SPECIAL FORMS IN THE REFERENCE MANUAL - 60100000 (6000/CYBER-70)
- 60420000 (CYBER-170)

Notes on Hardware Error Exits

Certain error conditions are detected by the hardware:

```
   50     49     48
                    /Referencing address out of range
   x      x      x                  (error code 1)
   |       \
   |        Using infinite operand (error code 2)
   |
    Using indefinite operand (error code 4)
```

When the error is detected the hardware will:
(6000/Cyber-70/Cyber-170)

1. Put the code in RA+0 bits 50-48

2. Put P+1 * in RA+0 bits 47-30

3. Set P to 0

The Software will:

1. Discover that the CPU is not running and that P=0. (MTR says this is illegal).

2. Pick up the error mode and address from RA+0 and format a dayfile message.

3. Give the DMPX and abort the job.

Look at the following example which shows all of these conditions.

Also see Cyber Reference Manual.

* Referring to what P was at the time the error was discovered in the functional unit.

```
                                          IDENT   INDEFIN
                                          ENTRY   INDEFIN
     0  17770000000000000000      INDEF   VFD     12/1777B,48/0  INDEFINITE OPERAND
     1  17214000000000000000      NUMBER  DATA    2.0
     2  5110000000 +              INDEFIN SA1     INDEF
                  5120000001 +            SA2     NUMBER
     3  44312                             FX1     X1/X2        USE INDEFINITE OPERAND IN DIVIDE
     4  46000                 +           NO
     5  46000                 +           NO
     6  46000                 +           NO
     7  46000                 +           NO
            7160247021                    ENDRUN               NORMAL TERMINATION
    11                                    END     INDEFIN      TRANSFER ADDRESS

                      47300B SCM STORAGE USED             16 STATEMENTS         4 SYMBOLS
                              MODEL 174 ASSEMBLY      0.019 SECONDS           7 REFERENCES

DMPX.

 P    000000  A0  000200  B0  000000
 RA   510400  A1  000111  B1  000001     C(A1)=  1777  0000  0000  0000  0000      C(B1)=  0000  0000  0000  0000  0000
 FL   000200  A2  000112  B2  000002     C(A2)=  1721  4000  0000  0000  0000      C(B2)=  0000  0000  0000  0000  0000
 EM   700700  A3  000057  B3  012733     C(A3)=  0000  0000  0000  0000  0000      C(B3)=
 RE   000055  A4  000001  B4  000201     C(A4)=  0000  0000  0000  0000  0000      C(B4)=
 FE   000000  A5  000113  B5  000113     C(A5)=  5110  0001  1151  2000  0112      C(B5)=  5110  0001  1151  2000  0112
 MA   001200  A6  000001  B6  000200     C(A6)=  0000  0000  0000  0000  0000      C(B6)=
              A7  000001  B7  027756     C(A7)=  0000  0000  0000  0000  0000      C(B7)=
 X0   0000  0000  0000  0000  0000
 X1   1777  0000  0000  0000  0000
 X2   1721  4000  0000  0000  0000
 X3   1777  0000  0000  0000  0000
 X4   0000  0000  0000  0000  0000
 X5   6000  0000  0004  0040  0000
 X6   1505  1520  0000  0000  0061
 X7   0000  0000  0000  0000  0000

    00000    00040 00117 00000 00000       00000 00000 00000 00000
    00054   *56110 03110 00054 54710       51100 00001 03110 00055       64550 02550 00000 46000       00000 00000 00000 00000
    00060    15051 52000 00000 00061       00000 00200 00000 00001       07040 00060 51600 00001       04000 00063 00000 00021
    00064    14071 70000 00000 00000       00000 00000 00000 00162       40000 00000 02000 00111       00000 00000 40000 00000
    00070    14071 75755 00000 00000       00000 00000 00000 00000
    00100   *54000 00000 01000 00001

    00100    54000 00000 01000 00001       00005 10000 00000 00162       00000 00000 00000 00000
    00104   *00000 00000 00000 00162       00000 00000 00000 00000
    00110   *11160 40506 11160 00113       17770 00000 00000 00000       17214 00000 00000 00000       51100 00111 51200 00112
    00114    44312 46000 61000 46000       46000 46000 61000 46000
    00120   *46000 71602 47021 20650       01000 00124 61000 46000       54110 20123 03310 00123       04000 00127 61000 46000
    00124    04004 00124 61000 46000       51100 00001 03110 00125       54610 04000 00123 46000       51100 00066 03310 00131
    00130    51100 00122 04000 00132       71100 00130 20160 46000       13661 13161 13661 46000       51600 00123 10611 46000
    00134    51100 00001 01000 00122       20652 01000 00124 46000       51100 00001 03110 00136       04004 00137 61000 46000
    00140    51100 00001 03110 00137       71602 20314 04000 00135       20150 36661 01000 00124       04004 00143 61000 46000
    00144    71602 20314 20652 36662       53160 20173 03310 00143       03010 00143 51100 00001       03110 00145 71100 00001
    00150    04000 00142 61000 46000       71603 24616 12661 20651       01000 00124 61000 46000       04004 00153 61000 46000
    00154    73660 20630 12161 73610       20123 03210 00151 20151       13116 20636 51600 00161       74660 36116 20123 46000
    00160    04000 00151 61000 46000       00000 00000 00000 00000       60000 00000 04004 00162       60000 00000 04004 00163
    00164    60000 00000 04004 00164       60000 00000 04004 00165       60000 00000 04004 00166       60000 00000 04004 00167
```

```
INDEFIN                                                              COMPASS 3.5-470.     06/27/78  08.55.10.       PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH              BINARY CONTROL CARDS.

                0       11              IDENT   INDEFIN
               11                       END     INDEFIN     TRANSFER ADDRESS

                                  ENTRY POINTS.

                                  INDEFIN           2*

                                  EXTERNAL SYMBOLS.

                                  SYS=

 LOAD MAP - INDEFIN                                         CYBER LOADER 1.4-470       06/27/78  08.55.11.       PAGE      1


 FWA OF THE LOAD             111
 LWA+1 OF THE LOAD           162

 TRANSFER ADDRESS -- INDEFIN          113

 PROGRAM ENTRY POINTS --       INDEFIN          113


 PROGRAM AND BLOCK ASSIGNMENTS.

 BLOCK        ADDRESS   LENGTH     FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

 INDEFIN         111        11    LGO      06/27/78 COMPASS 3.5 470
 SYS.RM          122        40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                   PROCESS SYSTEM REQUEST.


      .021 CP SECONDS              13200B CM STORAGE USED                  MFF   NB2-  CYB175-SN1    4L87/R68  05/15/78
                                                                     08.55.07.DON0042  FROM    /SH
                                                                     08.55.07.IP  00000192 WORDS - FILE INPUT  , DC 04
                                                                     08.55.07.DON.     PSD,0278,72CT011A,MILLER
                                                                     08.55.10.COMPASS.
                                                                     08.55.10. ASSEMBLY COMPLETE.   47300B SCM USED.
                                                                     08.55.10.    0.069  CPU SECONDS  ASSEMBLY TIME.
                                                                     08.55.10.LGO.
                                                                     08.55.11.ERROR MODE =04.  ADDRESS =000117
                                                                     08.55.12.OP  00000960 WORDS - FILE OUTPUT , DC 40
                                                                     08.55.12.MS     3584  WORDS  (    7168 MAX USED)
                                                                     08.55.12.CPA         .121 SEC.          .120 ADJ.
                                                                     08.55.12.IO          .639 SEC.          .639 ADJ.
                                                                     08.55.12.CM         9.878 KWS.          .602 ADJ.
                                                                     08.55.12.SS                            1.362
                                                                     08.55.12.PP         4.148 SEC.     DATE 06/27/78
                                                                     08.55.12.EJ  END OF JOB, SH
```

```
                                           IDENT  TSTADDR
                                           ENTRY  TSTADDR
                                   ******************************************************************
                                   *                                                                *
                                   *     ADDRESS OUT OF RANGE ERRORS                                *
                                   *                                                                *
                                   *     THE LOAD OR STORE INSTRUCTION WHICH CAUSES THIS ERROR MAY BE SEVERAL  *
                                   *     INSTRUCTIONS BEHIND THE ADDRESS IN THE DAYFILE.  FOR EXAMPLE, THE     *
                                   *     FOLLOWING PROGRAM WAS LOADED AT 111.  THE GUILTY INSTRUCTION WAS AT 112.*
                                   *     THE HARDWARE DID NOT DISCOVER IT UNTIL THE INSTRUCTION AT 113 WAS     *
                                   *     EXECUTED.                                                  *
                                   *                                                                *
                                   ******************************************************************
    0  46000                       TSTADDR  NO.
             6110777013                     SB1     -500
    1  5111000001 +                         SA1     B1+*
                  46000                     NO
    2  46000                       +        NO
    3  46000                       +        NO
    4  46000                       +        NO
             7160247021                     ENDRUN                NORMAL TERMINATION
    6                                       END     TSTADDR       TRANSFER ADDRESS

                     473000 SCM STORAGE USED              25 STATEMENTS          2 SYMBOLS
                             MODEL 174 ASSEMBLY        0.025 SECONDS             3 REFERENCES

DMPX.

 P    000000  A0   000200  B0   000000
 RA   206600  A1   777125  B1   777013     C(A1)=                                   C(B1)=
 FL   000200  A2   000060  B2   000002     C(A2)=  1505  1520  0000  0000  0061     C(B2)=  0000  0000  0000  0000  0000
 EM   700700  A3   000057  B3   012733     C(A3)=  0000  0000  0000  0000  0000     C(B3)=
 RE   000055  A4   000001  B4   000201     C(A4)=  0000  0000  0000  0000  0000     C(B4)=
 FE   000000  A5   000111  B5   000111     C(A5)=  4600  0611  0777  0134  6000     C(B5)=  4600  0611  0777  0134  6000
 MA   001200  A6   000001  B6   000200     C(A6)=  0000  0000  0000  0000  0000     C(B6)=
              A7   000001  B7   027756     C(A7)=  0000  0000  0000  0000  0000     C(B7)=
 X0   0000  0000  0000  0000  0000
 X1   0000  0000  0000  0000  0000
 X2   1505  1520  0000  0000  0061
 X3   0000  0000  0000  0000  0000
 X4   0000  0000  0000  0000  0000
 X5   6000  0000  0004  0040  0000
 X6   1505  1520  0000  0000  0061
 X7   0000  0000  0000  0000  0000

    00000    00010 00113 00000 00000        00000 00000 00000 00000
    00054   *56110 03110 00054 54710        51100 00001 03110 00055        64550 02550 00000 46000        00000 00000 00000 00001
    00060    15051 52000 00000 00061        00000 00200 00000 00001        07040 00060 51600 00001        04000 00063 00000 00021
    00064    14071 70000 00000 00000        00000 00000 00000 00157        40000 00000 02000 00111        00000 00000 40000 00001
    00070    14071 75755 00000 00000        00000 00000 00000 00000
    00100   *54000 00000 01000 00001

    00100    54000 00000 01000 00001        00004 60000 00000 00157        00000 00000 00000 00000
    00104   *00000 00000 00000 00157        00000 00000 00000 00000
    00110   *24232 40104 04220 00111        46000 61107 77013 46000        51110 00112 46000 46000        46000 46000 61000 46000
    00114   *46000 71602 47021 20450
```

```
                                                          COMPASS 3.5-470.     06/27/78  09.03.50.      PAGE      1
TSTADDR
STORAGE ALLOCATION.

          ADDRESS   LENGTH               BINARY CONTROL CARDS.

                0        6               IDENT   TSTADDR
                6                        END     TSTADDR     TRANSFER ADDRESS


                                ENTRY POINTS.

                                TSTADDR          0+


                                EXTERNAL SYMBOLS.

                                SYS=

                                                      CYBER LOADER 1.4-470      06/27/78  09.03.51.      PAGE      1
  LOAD MAP - TSTADDR



  FWA OF THE LOAD              111
  LWA+1 OF THE LOAD            157

  TRANSFER ADDRESS -- TSTADDR          111

  PROGRAM ENTRY POINTS --      TSTADDR          111



  PROGRAM AND BLOCK ASSIGNMENTS.

  BLOCK         ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

  TSTADDR          111        6    LGO      06/27/78 COMPASS 3.5 470                  PROCESS SYSTEM REQUEST.
  SYS.RM           117       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470

                                                                              MFF   NB2-  CYB175-SN1    4L87/R68  05/15/78
                                                                             09.03.47.DON00ST  FROM    /SH
        .021 CP SECONDS            13200B CM STORAGE USED                    09.03.47.IP  00000320 WORDS - FILE INPUT  , DC 04
                                                                             09.03.47.DON.      PSD,0278,72CT011A,MILLER
                                                                             09.03.50.COMPASS.
                                                                             09.03.51. ASSEMBLY COMPLETE.   47300B SCM USED.
                                                                             09.03.51.    0.073  CPU SECONDS  ASSEMBLY TIME.
                                                                             09.03.51.LGO.
                                                                             09.03.51.ERROR MODE =01.  ADDRESS =000113
                                                                             09.03.52.OP  0000108B WORDS - FILE OUTPUT , DC 40
                                                                             09.03.52.MS    3584  WORDS  (    7168 MAX USED)
                                                                             09.03.52.CPA        .126 SEC.          .126 ADJ.
                                                                             09.03.52.IO         .643 SEC.          .643 ADJ.
                                                                             09.03.52.CM        9.840 KWS.          .600 ADJ.
                                                                             09.03.52.SS                           1.371
                                                                             09.03.52.PP        4.473 SEC.     DATE 06/27/78
                                                                             09.03.52.EJ  END OF JOB, SH
```

```
OUTRANG                                                          COMPASS 3.6-476.     01/08/79  10.29.04.      PAGE      1

                                               IDENT  OUTRANG
                                               ENTRY  OUTRANG
                                       **************************************************************************
                                       *                                                                        *
                                       *     ADDRESS OUT OF RANGE                                               *
                                       *                                                                        *
                                       *     UNDER CONTROL OF A MODE 6 CONTROL CARD EXECUTION CONTINUES WHEN ADDRESSES
                                       *     OUT OF RANGE ARE REFERENCED.                                       *
                                       *     1.  LOADING FROM AN ADDRESS OUT OF RANGE CAUSES A LOAD FROM ABSOLUTE *
                                       *         ADDRESS 0.                                                     *
                                       *                                                                        *
                                       *     2.  STORING OUT OF RANGE IS A NO-OPERATION (NO STORE IS DONE)      *
                                       *                                                                        *
                                       **************************************************************************
   0  7170000007                       OUTRANG  SX7    7
               5170000000                       SA7    0
   1  7150001111                                SX5    1111B
               5150377776                       SA5    377776B
   2  7170006666                                SX7    6666B
               5170377776                       SA7    377776B
   3  46000                            +        NO
   4  46000                            +        NO
           7130003333                           SX3    3333B
   5  46000                            +        NO
   6  46000                            +        NO
           7160247021                           ENDRUN
  10                                            END    OUTRANG

                    47500B CM  STORAGE USED              31 STATEMENTS         2 SYMBOLS
                               MODEL 74 ASSEMBLY      0.095 SECONDS            0 REFERENCES
```

```
LOAD MAP - OUTRANG                                           CYBER LOADER 1.4-485       01/08/79  10.29.05.      PAGE      1


FWA OF THE LOAD                 111
LWA+1 OF THE LOAD               161

TRANSFER ADDRESS -- OUTRANG            111

PROGRAM ENTRY POINTS --        OUTRANG         111


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS   LENGTH     FILE      DATE      PROCSSR VER LEVEL  HARDWARE    COMMENTS

OUTRANG        111       10     LGO      01/08/79 COMPASS 3.6 476
CPU.SYS        121       40 SL-SYSLIB    11/15/78 COMPASS 3.6 476                    PROCESS SYSTEM REQUEST.

     .055 CP SECONDS                 13500B CM STORAGE USED                       1 TABLE MOVE
```

```
DUMP      EXCHANGE PACKAGE                      DMP.

P    000123  A0  000200  B0  000000
RA   175600  A1  000001  B1  000001  C(A1)=  0000  0000  0000  0000  0000     C(B1)=  0000  0000  0000  0000  0000
FL   000200  A2  000060  B2  000002  C(A2)=  1505  1520  0000  0000  0061     C(B2)=  0000  0000  0000  0000  0000
EM   000600  A3  000057  B3  013274  C(A3)=  0000  0000  0000  0000  0000     C(B3)=     * * OUT OF RANGE * *
RE   000037  A4  000001  B4  000201  C(A4)=  0000  0000  0000  0000  0000     C(B4)=     * * OUT OF RANGE * *
FE   000000  A5  377776  B5  000111  C(A5)=     * * OUT OF RANGE * *          C(B5)=  7170  0000  0751  7000  0000
MA   001000  A6  000122  B6  000200  C(A6)=  0130  0000  0000  0000  0000     C(B6)=     * * OUT OF RANGE * *
             A7  377776  B7  027756  C(A7)=     * * OUT OF RANGE * *          C(B7)=     * * OUT OF RANGE * *
X0   0000  0000  0000  0000  0000
X1   0516  0420  0000  0000  0000
X2   1505  1520  0000  0000  0061
X3   0000  0000  0000  0000  3333
X4   0000  0000  0000  0000  0000
X5   0000  0000  0000  0000  0000
X6   0516  0420  0000  0000  0000
X7   0000  0000  0000  0000  6666
   00000    00000 00000 00000 00007         00000 00000 00000 00000   00054_56110 03110 00054 54710         51100 00001 03110 00055
   00056    64550 02550 00000 46000         00000 00000 00000 00000
   00060    15051 52000 00000 00061         00000 00200 00000 00001         07040 00060 51600 00001         04000 00063 00000 00000
   00064    04152 00000 00000 00000         00000 00000 00000 00161         40000 00000 02000 00111         00000 00000 40000 00000
   00070    14071 75700 00000 00000         00000 00000 00000 00000   00100_54000 00000 01000 00001

   00100    54000 00000 01000 00001         00005 00000 00000 00161         00000 00000 00000 00000   00104_00000 00000 00000 00161
   00105    00000 00000 00000 00000   00110_17252 42201 16070 00111         71700 00007 51700 00000         71500 01111 51503 77776
   00113    71700 06666 51703 77776
   00114    46000 46000 61000 46000         46000 71300 03333 46000         46000 46000 61000 46000         46000 71602 47021 20650
   00120    01000 00123 61000 46000         04000 00134 00000 00000         01300 00000 00000 00000         04000 00121 00000 00000
   00124    51100 00001 03110 00124         54610 04000 00122 46000         51100 00066 03310 00130         51100 00121 04000 00131
   00130    71100 00130 20160 46000         13661 13161 13661 46000         51600 00122 10611 46000         51100 00001 01000 00121
   00134    20652 01000 00123 46000         51100 00001 03110 00135         04004 00136 61000 46000         51100 00001 03110 00136
   00140    71602 20314 04000 00134         20150 36661 01000 00123         04004 00142 61000 46000         71602 20314 20652 36662
   00144    53160 20173 03310 00142         03010 00142 51100 00001         03110 00144 71100 00001         04000 00141 61000 46000
   00150    71603 24616 12661 20651         01000 00123 61000 46000         04004 00152 61000 46000         73660 20630 12161 73610
   00154    20123 03210 00150 20151         13116 20636 51600 00160         74660 36116 20123 46000         04000 00150 61000 46000
   00160    00000 00000 00000 00000         60000 00000 04004 00161   00177>60000 00000 04004 00177
```

```
  MFS  NB1-  CYB74-SN108     5C/R08     11/14/78
10.25.10.DUN00K1  FROM    /OH
10.25.10.IP  00000384 WORDS - FILE INPUT  , DC 04
10.25.10.DON,T5.   001A,6883,1896,MILLER
10.25.19.REWIND,OUTPUT.
10.27.49.COMPASS,LO=BR.
10.29.04. ASSEMBLY COMPLETE.   47500B  CM USED.
10.29.04.    0.228  CPU SECONDS  ASSEMBLY TIME.
10.29.04.MODE,6.
10.29.04.LGO.
10.29.05.  WE GOT THIS FAR
10.29.05.DMP.
10.29.05.OP  00000832 WORDS - FILE OUTPUT , DC 40
10.29.05.MS    3584  WORDS  (    7168 MAX USED)
10.29.05.CPA         .445 SEC.           .445 ADJ.
10.29.05.CPB         .282 SEC.           .282 ADJ.
10.29.06.IO          .316 SEC.           .316 ADJ.
10.29.06.CM       12.493 KWS.            .762 ADJ.
10.29.06.SS                             1.806
10.29.06.PP       10.161 SEC.     DATE 01/08/79
10.29.06.EJ  END OF JOB, OH
```

```
 TESTRAO                                                                COMPASS 3.5-470.     06/27/78  08.30.20.      PAGE      2

                                                         IDENT   TESTRAO
                                                         ENTRY   TESTRAO
        0  71006666660400000115      DATA     DATA    71006666660400000115B
        1  5110000000 +             TESTRAO   SA1     DATA
                     10711                    BX7     X1
        2  5170000000                         SA7     0
                     0400000000               EQ      0
        3  46000                     +        NO
        4  46000                     +        NO
        5  46000                     +        NO
        6  46000                     +        NO
               7160247021                     ENDRUN
       10                                     END     TESTRAO

                        47300B SCM STORAGE USED            16 STATEMENTS           3 SYMBOLS
                               MODEL 174 ASSEMBLY        0.015 SECONDS             5 REFERENCES
 DUMP     EXCHANGE PACKAGE                    DMP.

  P    000123  A0   000200  B0   000000
  RA   344400  A1   000001  B1   000001  C(A1)=  0000  0000  0000  0000  0000      C(B1)=  0000  0000  0000  0000  0000
  FL   000200  A2   000060  B2   000002  C(A2)=  1505  1520  0000  0000  0061      C(B2)=  0000  0000  0000  0000  0000
  FM   700700  A3   000057  B3   012733  C(A3)=  0000  0000  0000  0000  0000      C(B3)=    * * OUT OF RANGE * *
  RE   000055  A4   000001  B4   000201  C(A4)=  0000  0000  0000  0000  0000      C(B4)=    * * OUT OF RANGE * *
  FE   000000  A5   000112  B5   000112  C(A5)=  5110  0001  1110  7114  6000      C(B5)=  5110  0001  1110  7114  6000
  MA   001200  A6   000122  B6   000200  C(A6)=  0130  0000  0000  0000  0000      C(B6)=    * * OUT OF RANGE * *
               A7   000000  B7   027756  C(A7)=  7100  6666  6604  0000  0115      C(B7)=    * * OUT OF RANGE * *
 X0   7777  7777  7777  7766  6666
 X1   0516  0420  0000  0000  0000
 X2   1505  1520  0000  0000  0061
 X3   0000  0000  0000  0000  0000
 X4   0000  0000  0000  0000  0000
 X5   6000  0000  0004  0040  0000
 X6   0516  0420  0000  0000  0000
 X7   7100  6666  6604  0000  0115
   00000     71006 66666 04000 00115          00000 00000 00000 00000   00054-56110 03110 00054 54710          51100 00001 03110 00055

   00056     64550 02550 00000 46000          00000 00000 00000 00000
   00060     15051 52000 00000 00061          00000 00200 00000 00001         07040 00060 51600 00001          04000 00063 00000 00021

   00064     04152 00000 00000 00000          00000 00000 00000 00161         40000 00000 02000 00111          00000 00000 40000 00000

   00070     14071 75755 00000 00000          00000 00000 00000 00000   00100-54000 00000 01000 00001

   00100     54000 00000 01000 00001          00005 00000 00000 00161         00000 00000 00000 00000   00104-00000 00000 00000 00161

   00105     00000 00000 00000 00000   00110-24052 32422 01330 00112         71006 66666 04000 00115          51100 00111 10711 46000

   00113     51700 00000 04000 00000
   00114     46000 46000 61000 46000   00117-46000 71602 47021 20650
   00120     01000 00123 61000 46000          04000 00134 00000 00000         01300 00000 00000 00000          04000 00121 00000 00000

   00124     51100 00001 03110 00124          54610 04000 00122 46000         51100 00068 03310 00130          51100 00121 04000 00131

   00130     71100 00130 20160 46000          13661 13161 13661 46000         51600 00122 10611 46000          51100 00001 01000 00121

   00134     20652 01000 00123 46000          51100 00001 03110 00135         04004 00136 61000 46000          51100 00001 03110 00136
```

```
                                                                        COMPASS 3.5-470.     06/27/78  08.38.20.       PAGE      1
TESTRAO
STORAGE ALLOCATION.

           ADDRESS   LENGTH                BINARY CONTROL CARDS.

                 0        10               IDENT   TESTRAO
                10                         END     TESTRAO

                                    ENTRY POINTS.

                                    TESTRAO           1*

                                    EXTERNAL SYMBOLS.

                                    SYS=
```

```
                                                                  CYBER LOADER 1.4-470      06/27/78  08.38.21.      PAGE      1
 LOAD MAP - TESTRAO


 FWA OF THE LOAD                111
 LWA+1 OF THE LOAD              161

 TRANSFER ADDRESS -- TESTRAO            112

 PROGRAM ENTRY POINTS --       TESTRAO           112


 PROGRAM AND BLOCK ASSIGNMENTS.

 BLOCK          ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

 TESTRAO           111         10    LGO      06/27/78 COMPASS 3.5 470
 SYS.PM            121         40 SL-SYSLIB   05/16/78 COMPASS 3.5 470
                                                                                        MFF   NB2-  CYB175-SN1    4LB7/R6B  05/15/78
                                                                                      08.38.17.DON0031  FROM     /SH
                                                                                      08.38.17.IP  00000192 WORDS - FILE INPUT  . DC 04
                                                                                      08.38.17.DON.       PSO.0278.72CT011A.MILLER
      .022 CP SECONDS               13200B CM STORAGE USED                            08.38.19.COMPASS.
                                                                                      08.38.20. ASSEMBLY COMPLETE.   47300B SCM USED.
                                                                                      08.38.20.     0.066  CPU SECONDS  ASSEMBLY TIME.
                                                                                      08.38.20.LGO.
                                                                                      08.38.21.DMP.
                                                                                      08.38.21.OP  00000896 WORDS - FILE OUTPUT . DC 40
                                                                                      08.38.21.MS    3584  WORDS  (   10752 MAX USED)
                                                                                      08.38.21.CPA        .114 SEC.         .114 ADJ.
                                                                                      08.38.21.IO         .631 SEC.         .631 ADJ.
                                                                                      08.38.21.CM        9.665 KWS.         .589 ADJ.
                                                                                      08.38.21.SS                          1.335
                                                                                      08.38.21.PP        4.289 SEC.     DATE 06/27/78
                                                                                      08.38.21.EJ  END OF JOB. SH
```

LESSON 9

MACROS

LESSON PREVIEW:

THIS LESSON COVERS THE DEFINITION AND CALLS OF MACROS AND OPDEFS, THE USE OF MICROS AND PSEUDO-OPS, AND CONDITIONAL ASSEMBLY.

REFERENCES:

CHAPTER 4-7	COMPASS REFERENCE MAN. #60492600

TRAINING AIDS:

VISUAL SET V9

PROJECTS:

PROGRAMMING PROJECT 6

OBJECTIVES:

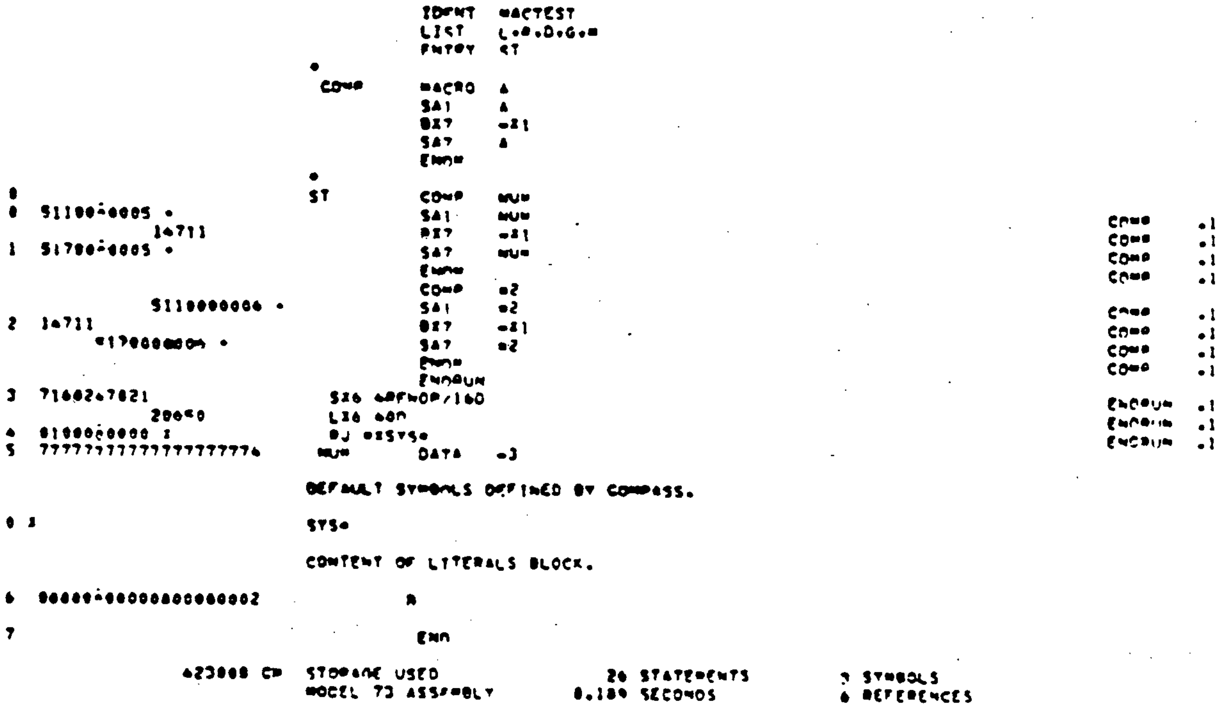
AT THE COMPLETION OF THIS LESSON THE STUDENT WILL BE ABLE TO:

1. WRITE PROGRAMS THAT UTILIZE:
   - A. MACROS
   - B. OPDEFS
   - C. MICROS
   - D. PSEUDO-OPS

2. USE CONDITIONAL ASSEMBLY

DEFINITION

Often a sequence of code should be assembled many places in a program. Rather than writing out the code each time it is needed, it would be helpful to give a name to the sequence and then just use the name each time you want the assembler to bring in the code. A sequence of instructions that will be assembled whenever called is a MACRO. For added convenience the programmer could specify certain elements in the MACRO as variable; that is they can change. Each time the MACRO is called, different values could be given for these variables or parameters. The assembler would then insert the given values for the parameters as the code is assembled. For example:



```
                                             IDENT   MACTEST
                                             LIST    L,R,D,G,M
                                             ENTRY   ST
                                   *
                                    COMP     MACRO   A
                                             SA1     A
                                             BX7     -X1
                                             SA7     A
                                             ENDM
                                   *
0                                  ST        COMP    NUM
0  51100-0005 +                              SA1     NUM                              COMP     .1
            14711                            BX7     -X1                              COMP     .1
1  51700-0005 +                              SA7     NUM                              COMP     .1
                                             ENDM                                     COMP     .1
                                             COMP    =2
            5110000006 +                     SA1     =2                               COMP     .1
2  14711                                     BX7     -X1                              COMP     .1
       5170000006 +                          SA7     =2                               COMP     .1
                                             ENDM                                     COMP     .1
                                             ENDDUM
3  7160247021                       SX6 ARFNOP/100                                    ENDDUM   .1
            20050                   LX6 6ON                                           ENDDUM   .1
4  0100000000 X                     RJ =XSYS=                                         ENDDUM   .1
5  77777777777777777776            NUM       DATA    -3

                                   DEFAULT SYMBOLS DEFINED BY COMPASS.

0 X                                SYS=

                                   CONTENT OF LITERALS BLOCK.

6  00000000000000000002                      R

7                                            END

                    623000 CM  STORAGE USED             26 STATEMENTS           3 SYMBOLS
                               MODEL 73 ASSEMBLY        0.189 SECONDS           6 REFERENCES
```

MACRO

- Type 1 format
- Single "instruction" to call a sequence of code
- Like an in-line subroutine

I.E.
SUM macro

The Call:

N WORDS TO SUM

```
        SBI     10
        SUM     BUF,A,
        .
        .
BUF     BSS     10
A       BSS     1
```

CALL

ACTUAL PARAMETERS

MACRO CODE INSERTED HERE BY ASSEMBLER

The Macro Definition:

MACRO NAME

FORMAL PARAMETERS

```
SUM     MACRO   DATA,ANS
        LOCAL   ADD
        SX7     0
ADD     SB1     B1-1
        SA1     DATA+B1
        1X7     X1+X7
        PL      B1,ADD
        SA7     ANS
        ENDM
```

LOCAL SYMBOL

CODE TO BE INSERTED IN LINE WHEN THE MACRO IS CALLED

```
                                                    COMPASS 3.5-470.     06/27/78  09.13.58.      PAGE      1
MOV7600
STORAGE ALLOCATION.

       ADDRESS   LENGTH              BINARY CONTROL CARDS.

             0      100              IDENT  MOV7600
           100                       END    ADD

                         ENTRY POINTS.                              MFF   NB2-  CYB175-SN1    4LB7/H6B  05/15/78
                                                                   09.13.56.DON006U  FROM    /SH
                         ADD              30+                      09.13.56.IP  00000512 WORDS - FILE INPUT  , DC 04
                                                                   09.13.56.DON.     PSC.027B.72CT011A.MILLER
                                                                   09.13.57.COMPASS.
                         EXTERNAL SYMBOLS.                         09.13.58. ASSEMBLY COMPLETE.   47300B SCM USED.
                                                                   09.13.58.    0.179  CPU SECONDS  ASSEMBLY TIME.
                         SYS=                                      09.13.58.LGO.
                                                                   09.13.59.DMP(111,177)
                                                                   09.13.59.EXIT.
                                                                   09.13.59.OP  00001792 WORDS - FILE OUTPUT , DC 40
                                                                   09.13.59.MS     3584  WORDS  (   10752 MAX USED)
                                                                   09.13.59.CPA         .229 SEC.           .229 ADJ.
                                                                   09.13.59.IO          .651 SEC.           .651 ADJ.
                                                                   09.13.59.CM       12.850 KWS.            .784 ADJ.
                                                                   09.13.59.SS                             1.665
                                                                   09.13.59.PP         2.964 SEC.     DATE 06/27/78
                                                                   09.13.59.EJ  END OF JOB, SH
```

```
MOV7600                                               COMPASS 3.5-470.     06/27/78  09.13.58.      PAGE      4
SYMBOLIC REFERENCE TABLE.

ADD             30     PROGRAM*        2/03 F     2/31 S     2/55 S     3/16 S     3/44
                                       2/16 L     2/41       3/02       3/20
ANSWER           0     PROGRAM*        2/14 L     2/23 S     2/31 S     2/49 S     3/10 S     3/36 S     3/42 S
                                       2/22 S     2/30 S     2/48 S     3/09 S     3/35 S     3/41 S
DIVIDE          66     PROGRAM*        3/14       3/21 L     3/47
END1BLK         43     PROGRAM*        2/41 L     3/44
END2BLK         54     PROGRAM*        3/02 L     3/45
END3BLK         65     PROGRAM*        3/20 L     3/46
END4BLK         76     PROGRAM*        3/43 L     3/47
LBLOCKT         50                     3/48 D
LBLOCK1         14                     3/44 D     3/48       3/49       3/50
LBLOCK2         11                     2/34       3/45 D     3/48       3/50
LBLOCK3         11                     2/52       3/46 D     3/48
LBLOCK4         12                     3/13       3/47 D     3/48
MAXBLK          14                     3/49 D
MINBLK          11                     3/50 D
MULT            55     PROGRAM*        2/53       3/03 L     3/46
OPERAND         20     PROGRAM*        2/15 L     2/18       2/25       2/42       2/45       3/05       3/22
                                       2/16       2/19       2/26       2/43       3/03       3/06       3/23
                                       2/17       2/24       2/27       2/44       3/04       3/21       3/24
SUBT            44     PROGRAM*        2/35       2/42 L     3/45
SYS=             0     EXTERNAL*       3/44
```

```
                                                 IDENT   MOV7600
                                                 LIST    M
                                                 ENTRY   ADD
                                         MOVE    MACRO   PARM1,PARM2,PARM3
                                                 LOCAL   MOVE1
                                                 SB1     0
                                                 SB2     PARM1
                                         MOVE1   SA5     PARM2+B1
                                                 BX6     X5
                                                 SA6     PARM3+B1
                                                 SB1     B1+1
                                                 NE      B1,B2,MOVE1
                                                 ENDM
    0                                20  ANSWER  BSSZ    16
   20  00000000000000000001              OPERAND DATA    1,2,3,4,5,6,7,8
   30  5110000020 +                      ADD     SA1     OPERAND+0
                     5120000021 +                SA2     OPERAND+1
   31  5130000022 +                              SA3     OPERAND+2
                     5140000023 +                SA4     OPERAND+3
   32  36612                                     IX6     X1+X2
            36734                                IX7     X3+X4
                     5160000000 +                SA6     ANSWER+0     STORE RESULT
   33  5170000001 +                              SA7     ANSWER+1     STORE RESULT
                     5110000024 +                SA1     OPERAND+4
   34  5120000025 +                              SA2     OPERAND+5
                     5130000026 +                SA3     OPERAND+6
   35  5140000027 +                              SA4     OPERAND+7
                     36612                       IX6     X1+X2
                          36734                  IX7     X3+X4
   36  5160000002 +                              SA6     ANSWER+2
                     5170000003 +                SA7     ANSWER+3
   37                                            MOVE    LBLOCK2,SUBT,ADD
   37  6110000000                                SB1     0                                       MOVE     .1
                     6120000011                  SB2     LBLOCK2                                 MOVE     .1
   40  5151000044 +                      ++000001   SA5     SUBT+B1                              MOVE     .1
                     10655                       BX6     X5                                      MOVE     .1
   41  5161000030 +                              SA6     ADD+B1                                  MOVE     .1
                     6111000001                  SB1     B1+1                                    MOVE     .1
   42  0512000040 +                              NE      B1,B2,++000001                          MOVE     .1
                                                 ENDM                                            MOVE     .1
   43  0400000030 +                      ENDIBLK EQ      ADD
   44  5110000020 +                      SUBT    SA1     OPERAND+0
                     5120000021 +                SA2     OPERAND+1
   45  5130000022 +                              SA3     OPERAND+2
                     5140000023 +                SA4     OPERAND+3
   46  37621                                     IX6     X2-X1
            37734                                IX7     X3-X4
                     5160000004 +                SA6     ANSWER+4     STORE RESULT
   47  5170000005 +                              SA7     ANSWER+5     STORE RESULT
                                                 MOVE    LBLOCK3,MULT,ADD
                     6110000000                  SB1     0                                       MOVE     .1
   50  6120000011                                SB2     LBLOCK3                                 MOVE     .1
   51  5151000055 +                      ++000002   SA5     MULT+B1                              MOVE     .1
                     10655                       BX6     X5                                      MOVE     .1
   52  5161000030 +                              SA6     ADD+B1                                  MOVE     .1
                     6111000001                  SB1     B1+1                                    MOVE     .1
   53  0512000051 +                              NE      B1,B2,++000002                          MOVE     .1
```

```
                                             ENDM                                                       MOVE     .1
 54  0400000030 +                 END2BLK    EQ       ADD
 55  5110000020 +                 MULT       SA1      OPERAND+0
              5120000021 +                   SA2      OPERAND+1
 56  5130000022 +                            SA3      OPERAND+2
              5140000023 +                   SA4      OPERAND+3
 57  42612                                   IX6      X1*X2
          42734                              IX7      X3*X4
              5160000006 +                   SA6      ANSWER+6      STORE RESULTS
 60  5170000007 +                            SA7      ANSWER+7      STORE RESULTS
                                             MOVE     LBLOCK4,DIVIDE,ADD
              6110000000                     SB1      0                                                 MOVE     .1
 61  6120000012                              SB2      LBLOCK4                                           MOVE     .1
 62  5151000066 +                 *+000003     SA5      DIVIDE+B1                                       MOVE     .1
              10655                          BX6      X5                                                MOVE     .1
 63  5161000030 +                            SA6      ADD+B1                                            MOVE     .1
              6111000001                     SB1      B1+1                                              MOVE     .1
 64  0512000062 +                            NE       B1,B2,*+000003                                    MOVE     .1
                                             ENDM                                                       MOVE     .1
 65  0400000030 +                 END3BLK    EQ       ADD
 66  5110000020 +                 DIVIDE     SA1      OPERAND+0
              5120000021 +                   SA2      OPERAND+1
 67  5130000022 +                            SA3      OPERAND+2
              5140000023 +                   SA4      OPERAND+3
 70  27101                                   PX1      X1
          27202                              PX2      X2
              24101                          NX1      X1
                  24202                      NX2      X2
 71  27303                                   PX3      X3
          27404                              PX4      X4
              24303                          NX3      X3
                  24404                      NX4      X4
 72  44621                                   FX6      X2/X1
          44743                              FX7      X4/X3
              5160000010 +                   SA6      ANSWER+8      STORE RESULT(FLOATING POINT)
 73  5170000011 +                            SA7      ANSWER+9      STORE RESULT(FLOATING POINT)
              26676                          UX6      X6,B7
                  26767                      UX7      X7,B6
 74  22676                                   LX6      X6,B7
          22767                              LX7      X7,B6
              5160000012 +                   SA6      ANSWER+10     STORE RESULT(INTEGER)
 75  5170000013 +                            SA7      ANSWER+11     STORE RESULT(INTEGER)
 76  7160247021                   END4BLK    ENDRUN
                          14      LBLOCK1    EQU      END1BLK-ADD+1
                          11      LBLOCK2    EQU      END2BLK-SUBT+1
                          11      LBLOCK3    EQU      END3BLK-MULT+1
                          12      LBLOCK4    EQU      END4BLK-DIVIDE+2
                          50      LBLOCKT    EQU      LBLOCK1+LBLOCK2+LBLOCK3+LBLOCK4
                          14      MAXBLK     MAX      LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4
                          11      MINBLK     MIN      LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4
100                                          END      ADD

                 47300B SCM STORAGE USED            111 STATEMENTS        21 SYMBOLS       000003 INVENTED SYMBOLS
                        MODEL 174 ASSEMBLY        0.122 SECONDS           78 REFERENCES
```

```
LOAD MAP - MOV7600                                              CYBER LOADER 1.4-470       06/27/78  09.13.59.      PAGE     1

FWA OF THE LOAD              111
LWA+1 OF THE LOAD            251

TRANSFER ADDRESS -- ADD               141

PROGRAM ENTRY POINTS --       MOV7600         141


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK          ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

MOV7600          111       100    LGO       06/27/78 COMPASS 3.5 470
SYS.RM           211        40 SL-SYSLIB    05/16/78 COMPASS 3.5 470                 PROCESS SYSTEM REQUEST.


     .029 CP SECONDS              132008 CM STORAGE USED                     1 TABLE MOVE



DUMP     RELATIVE                              DMP(111,177)

   00111    00000 00000 00000 00003       00000 00000 00000 00007       00000 00000 00000 00013
   00114    00000 00000 00000 00017       00000 00000 00000 00001       77777 77777 77777 77776       00000 00000 00000 00002

   00120    00000 00000 00000 00014       17214 00000 00000 00000       17205 25252 52525 25252       00000 00000 00000 00002

   00124    00000 00000 00000 00001       00000 00000 00000 00000 00131-00000 00000 00000 00001       00000 00000 00000 00002

   00133    00000 00000 00000 00003
   00134    00000 00000 00000 00004       00000 00000 00000 00005       00000 00000 00000 00006       00000 00000 00000 00007

   00140    00000 00000 00000 00010       51100 00131 51200 00132       51300 00133 51400 00134       27101 27202 24101 24202

   00144    27303 27404 24301 24404       44621 44743 51600 00121       51700 00122 26676 26767       22676 22767 51600 00123

   00150    51700 00124 61000 46000       71602 47021 20650 46000       01000 00213 61000 46000       05120 00151 61000 46000

   00154    04000 00141 61000 46000       51100 00131 51200 00132       51300 00133 51400 00134       37621 37734 51600 00115

   00160    51700 00116 61100 00000       61200 00011 61900 46000       51510 00166 10655 46000       51610 00141 61110 00001

   00164    05120 00162 61000 46000       04000 00141 61000 46000       51100 00131 51200 00132       51300 00133 51400 00134

   00170    42612 42734 51600 00117       51700 00120 61100 00000       61200 00012 61000 46000       51510 00177 10655 46000

   00174    51610 00141 61110 00001       05120 00173 61000 46000       04000 00141 61000 46000       51100 00131 51200 00132
```

MACRO

Type 2 format:

i.e.

LOAD macro

The Call:

```
                         LOADS X2
      LOC       LOAD       2,Y
LOCATION SYMBOL                 DATA FROM Y
      Y         DATA       3
```

The Macro Definition:

```
                                          LOCATION SYMBOL
                              MACRO NAME
                MACRO     LOAD,TAG,REG,ADDR
      TAG       SA.REG    ADDR
                ENDM                        FORMAL PARAMETERS
```

FORMAT II

```
MACRO                                                       COMPASS 3.6-476.     01/08/79  16.41.44.       PAGE     1

                                           IDENT  MACRO
                                           ENTRY  BEGIN
                                           LIST   M,-B,-R
                                           MACRO  SAM,P,Q,R,S,T
                                           SA1    T
                                           SA2    S
                                           IX5    X1*X2
                                           SA3    Q
                                           SA4    R
                                           IX6    X3*X4
                                           IX6    X5*X6
                                           SA6    P
                                  SAM      ENDM
   0                              BEGIN    BSS    0           PLACE TAG ON NEXT LOCATION
   0                              P+0      SAM    M,M+1,M+2,M+3
   0  5110000023 +                         SA1    M+3                                                  SAM      .1
              5120000022 +                 SA2    M+2                                                  SAM      .1
   1  42512                                IX5    X1*X2                                                SAM      .1
          5130000020 +                     SA3    M                                                    SAM      .1
   2  5140000021 +                         SA4    M+1                                                  SAM      .1
              42634                        IX6    X3*X4                                                SAM      .1
                   42656                   IX6    X5*X6                                                SAM      .1
   3  5160000036 +                         SA6    P+0                                                  SAM      .1
                                           ENDM                                                        SAM      .1
                                  P+1      SAM    M+3,M+4,M+5,M+6                                      SAM      .1
              5110000026 +                 SA1    M+6
   4  5120000025 +                         SA2    M+5                                                  SAM      .1
              42512                        IX5    X1*X2                                                SAM      .1
   5  5130000023 +                         SA3    M+3                                                  SAM      .1
              5140000024 +                 SA4    M+4                                                  SAM      .1
   6  42634                                IX6    X3*X4                                                SAM      .1
          42656                            IX6    X5*X6                                                SAM      .1
              5160000037 +                 SA6    P+1                                                  SAM      .1
                                           ENDM                                                        SAM      .1
                                           LIST   -M                                                   SAM      .1
   7  5110000031 +                P+2      SAM    M+6,M+7,M+8,M+9
              5110000035 +        P+3      SAM    I,I+1,I+2,I+3
  16  7160247021                           ENDRUN
  20  00000000000000000001        M        DATA   1,2,3,4,5,6,7,8,9,0
  32  00000000000000000001        I        DATA   1,2,3,3
  36                     4        P        BSSZ   4
  42                                       END    BEGIN

                    47500B CM  STORAGE USED                61 STATEMENTS
                               MODEL 73 ASSEMBLY         0.265 SECONDS
```

```
  MFS  NHI-  CYH74-SN108     SC/R08     11/14/78
16.38.07.DUN0003  FROM    /OH
16.38.07.IP  00000256 WORDS - FILE INPUT  , DC 04
16.38.07.DON,T5.   001A,6883,1896,MILLER
16.38.10.REWIND,OUTPUT.
16.38.10.COMPASS.
16.41.46. ASSEMBLY COMPLETE.   47500B  CM USED.
16.41.46.    0.410  CPU SECONDS  ASSEMBLY TIME.
16.41.46.LGO.
16.41.46.DMP(111,200)
16.41.46.OP  00000832 WORDS - FILE OUTPUT , DC 40
16.41.46.MS    3584  WORDS  (    7168 MAX USED)
16.41.46.CPA          .291 SEC.          .291 ADJ.
16.41.46.CPB          .266 SEC.          .266 ADJ.
16.41.46.IO           .317 SEC.          .317 ADJ.
16.41.46.CM         16.162 KWS.          .986 ADJ.
16.41.47.SS                             1.861
16.41.47.PP          2.942 SEC.     DATE 01/08/79
16.41.47.EJ  END OF JOB, OH
```

```
LOAD MAP - MACRO                                          CYBER LOADER 1.4-485      01/08/79  16.41.46.      PAGE     1

FWA OF THE LOAD               111
LWA+1 OF THE LOAD             213

TRANSFER ADDRESS -- BEGIN           111

PROGRAM ENTRY POINTS --       MACRO           111


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK       ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

MACRO          111       42     LGO       01/08/79 COMPASS 3.6 476
CPU.SYS        153       40  SL-SYSLIB    11/15/78 COMPASS 3.6 476                PROCESS SYSTEM REQUEST.


      .055 CP SECONDS              13500B CM STORAGE USED                   1 TABLE MOVE



DUMP    RELATIVE                         DMP(111,200)

   00111    51100 00134 51200 00133           42512 51300 00131 46000           51400 00132 42634 42656
   00114    51600 00147 51100 00137           51200 00136 42512 46000           51300 00134 51400 00135           42634 42656 51600 00150
   00120    51100 00142 51200 00141           42512 51300 00137 46000           51400 00140 42634 42656           51600 00151 51100 00146
   00124    51200 00145 42512 46000           51300 00143 51400 00144           42634 42656 51600 00152           71602 47021 20650 46000
   00130    01000 00155 61000 46000           00000 00000 00000 00001           00000 00000 00000 00002           00000 00000 00000 00003
   00134    00000 00000 00000 00004           00000 00000 00000 00005           00000 00000 00000 00006           00000 00000 00000 00007
   00140    00000 00000 00000 00010           00000 00000 00000 00011           00000 00000 00000 00000           00000 00000 00000 00001
   00144    00000 00000 00000 00002           00000 00000 00000 00003     00147_00000 00000 00000 00030
   00150    00000 00000 00000 01510           00000 00000 00000 00000           00000 00000 00000 00022           04000 00166 00000 00000
   00154    01300 00000 00000 00000           04000 00131 00000 00000           51100 00001 03110 00156           54610 04000 00154 46000
   00160    51100 00066 03310 00162           51100 00153 04000 00163           71100 00130 20160 46000           13661 13161 13661 46000
   00164    51600 00154 10611 46000           51100 00001 01000 00153           20652 01000 00155 46000           51100 00001 03110 00167
   00170    04004 00170 61000 46000           51100 00001 03110 00170           71602 20314 04000 00166           20150 36661 01000 00155
   00174    04004 00174 61000 46000           71602 20314 20652 36662           53160 20173 03310 00174           03010 00174 51100 00001
   00200    03110 00176 71100 00001
```

```
      PROGRAM MACDPTH    74/74   OPT=1                         FTN 4.7+485        01/09/79  12.55.45        PAGE     1

                    PROGRAM MACDPTH(OUTPUT)
                    COMMON ANSWER(24)
                    PRINT 100
                100 FORMAT(1H1)
                    CALL BEGIN
                    PRINT 20,(ANSWER(I),I=1,24)
                 20 FORMAT(4O24)
                    END
```

```
17214000000000000000    17216000000000000000    17214000000000000000    17216000000000000000
17176000000000000000    60556777777777777777    17176000000000000000    60573777777777777777
17247400000000000000    17235400000000000000    17247400000000000000    17235400000000000000
17177000000000000000    60547377777777777777    17177000000000000000    60573777777777777777
17265500000000000000    17244600000000000000    17265500000000000000    17244600000000000000
17177252525252525252    60547377777777777777    17177252525252525252    60573777777777777777
```

```
  MFS  NB1-  CYB74-SN108    SC/ROB    11/14/78
12.55.44.DON00GF  FROM    /OH
12.55.44.IP  00000320 WORDS - FILE INPUT  , DC 04
12.55.44.DON,T5.   001A,6883,1896,MILLER
12.55.45.FTN,R=0.
12.55.51.          1 WARNING MESSAGE IN  MACRO
12.55.51.       .687 CP SECONDS COMPILATION TIME
12.55.51.MAP,PART.
12.55.52.LGO.
12.56.24.      END  MACDPTH
12.56.24.       .077 CP SECONDS EXECUTION TIME
12.56.24.OP  00001920 WORDS - FILE OUTPUT , DC 40
12.56.25.MS    3584  WORDS  (   14336 MAX USED)
12.56.25.CPA         .987 SEC.           .987 ADJ.
12.56.25.CPB         .192 SEC.           .192 ADJ.
12.56.25.IO         1.166 SEC.          1.166 ADJ.
12.56.25.CM        42.524 KWS.          2.595 ADJ.
12.56.25.SS                             4.941
12.56.25.PP         8.340 SEC.     DATE 01/09/79
12.56.25.EJ  END OF JOB, OH
```

```
                                          IDENT  MACRO
                                          ENTRY  BEGIN
                                          LIST   M,-B,-R
                                 STORE    MACRO  P5,P6
                                          SA6    P5
                                          SA7    P6
                                          ENDM
                                 SAM      MACRO  PARM1,PARM2,PARM3,PARM4,PARM5,PARM6,PARM7,PARM8,PARM9,P
                                ,ARM10,PARM11,PARM12
                                          LOAD   PARM1,PARM2,PARM3,PARM4
                                          FX6    X1*X2
                                          FX7    X1+X2
                                          STORE  PARM5,PARM6
                                          NX6    X6
                                          NX7    X7
                                          STORE  PARM7,PARM8
                                          FX6    X3/X4
                                          FX7    X3-X4
                                          STORE  PARM9,PARM10
                                          NX6    X6
                                          NX7    X7
                                          STORE  PARM11,PARM12
                                          ENDM
                                 LOAD     MACRO  P1,P2,P3,P4
                                          SA1    P1
                                          SA2    P2
                                          SA3    P3
                                          SA4    P4
                                          ENDM
     0  77777777777777777777     BEGIN    DATA   -0
     1                                    SAM    M,M+1,M+2,M+3,ANS,ANS+1,ANS+2,ANS+3,ANS+4,ANS+5,ANS+6,A
                                ,NS+7
     1                                    LOAD   M,M+1,M+2,M+3                                          SAM     .1
     1  5110000032 +                      SA1    M                                                      LOAD    .2
                  5120000033 +            SA2    M+1                                                    LOAD    .2
     2  5130000034 +                      SA3    M+2                                                    LOAD    .2
                  5140000035 +            SA4    M+3                                                    LOAD    .2
                                          ENDM                                                          LOAD    .2
     3  40612                             FX6    X1*X2                                                  SAM     .1
             30712                        FX7    X1+X2                                                  SAM     .1
                                          STORE  ANS,ANS+1                                              SAM     .1
                  5160000000 C            SA6    ANS                                                    STORE   .2
     4  5170000001 C                      SA7    ANS+1                                                  STORE   .2
                                          ENDM                                                          STORE   .2
                  24606                   NX6    X6                                                     SAM     .1
                       24707              NX7    X7                                                     SAM     .1
     5                                    STORE  ANS+2,ANS+3                                            SAM     .1
     5  5160000002 C                      SA6    ANS+2                                                  STORE   .2
                  5170000003 C            SA7    ANS+3                                                  STORE   .2
                                          ENDM                                                          STORE   .2
     6  44634                             FX6    X3/X4                                                  SAM     .1
             31734                        FX7    X3-X4                                                  SAM     .1
                                          STORE  ANS+4,ANS+5                                            SAM     .1
                  5160000004 C            SA6    ANS+4                                                  STORE   .2
```

```
  7  5170000005 C                     SA7    ANS+5                                                       STORE    .2
                                      ENDM                                                               STORE    .2
               24606                  NX6    X6                                                          SAM      .1
                     24707            NX7    X7                                                          SAM      .1
 10                                   STORE  ANS+6,ANS+7                                                 SAM      .1
 10  5160000006 C                     SA6    ANS+6                                                       STORE    .2
               5170000007 C           SA7    ANS+7                                                       STORE    .2
                                      ENDM                                                               STORE    .2
                                      ENDM                                                               SAM      .1
 11                                   SAM    M+4,M+5,M+6,M+7,ANS+8,ANS+9,ANS+10,ANS+11,ANS+12,ANS+13
                            ,,ANS+14,ANS+15
 11                                   LOAD   M+4,M+5,M+6,M+7                                             SAM      .1
 11  5110000036 +                     SA1    M+4                                                         LOAD     .2
               5120000037 +           SA2    M+5                                                         LOAD     .2
 12  5130000040 +                     SA3    M+6                                                         LOAD     .2
               5140000041 +           SA4    M+7                                                         LOAD     .2
                                      ENDM                                                               LOAD     .2
 13  40612                            FX6    X1*X2                                                       SAM      .1
           30712                      FX7    X1+X2                                                       SAM      .1
                                      STORE  ANS+8,ANS+9                                                 SAM      .1
               5160000010 C           SA6    ANS+8                                                       STORE    .2
 14  5170000011 C                     SA7    ANS+9                                                       STORE    .2
                                      ENDM                                                               STORE    .2
               24606                  NX6    X6                                                          SAM      .1
                     24707            NX7    X7                                                          SAM      .1
 15                                   STORE  ANS+10,ANS+11                                               SAM      .1
 15  5160000012 C                     SA6    ANS+10                                                      STORE    .2
               5170000013 C           SA7    ANS+11                                                      STORE    .2
                                      ENDM                                                               STORE    .2
 16  44634                            FX6    X3/X4                                                       SAM      .1
           31734                      FX7    X3-X4                                                       SAM      .1
                                      STORE  ANS+12,ANS+13                                               SAM      .1
               5160000014 C           SA6    ANS+12                                                      STORE    .2
 17  5170000015 C                     SA7    ANS+13                                                      STORE    .2
                                      ENDM                                                               STORE    .2
               24606                  NX6    X6                                                          SAM      .1
                     24707            NX7    X7                                                          SAM      .1
 20                                   STORE  ANS+14,ANS+15                                               SAM      .1
 20  5160000016 C                     SA6    ANS+14                                                      STORE    .2
               5170000017 C           SA7    ANS+15                                                      STORE    .2
                                      ENDM                                                               STORE    .2
                                      ENDM                                                               SAM      .1
                                      LIST   -M
 21  5110000042 +                     SAM    M+8,M+9,M+10,M+11,ANS+16,ANS+17,ANS+18,ANS+19,ANS+20,AN
                            ,S+21,ANS+22,ANS+23
 31  0400000000 +                     EQ     BEGIN
 32  17204000000000000000     M       DATA   1.0,2.0,3.0,4.0,5.0,6.0,7.0,8.0,9.0,10.0,11.0,12.0
                                      USE    //
  0                    30     ANS     BSS    24
                                      USE    0
 46                                   END

                 47600B CM  STORAGE USED           121 STATEMENTS        3 SYMBOLS
                            MODEL 73 ASSEMBLY    0.454 SECONDS           0 REFERENCES
```

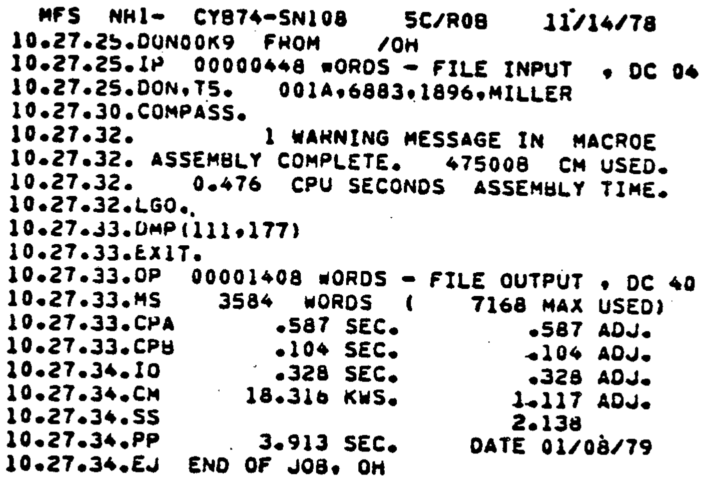
```
MACRO
ERROR DIRECTORY.                                                  COMPASS 3.6-476.     01/09/79  12.55.50.      PAGE      3

     3 TYPE ERROR              DUPLICATE MACRO DEFINITION.  NEW ONE OVERRIDES.
            OCCURRED ON PAGES         1



 LOAD MAP - MACDPTH
                                                              CYBER LOADER 1.4-485       01/09/79  12.56.22.      PAGE      1


 FWA OF THE LOAD             111
 LWA+1 OF THE LOAD          7133

 TRANSFER ADDRESS -- MACDPTH         2173

 PROGRAM ENTRY POINTS --       MACDPTH         2173


 PROGRAM AND BLOCK ASSIGNMENTS.

 BLOCK         ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

 MACDPTH           111     2105    LGO      01/09/79 FTN      4.7 485    666X I     PROGRAM   OPT=1
 MACRO            2216       46    LGO      01/09/79 COMPASS 3.6 476
 /STP.END/        2264        1
 /FCL.C./         2265       26
 /Q8.IO./         2313      101
 Q2NTRY=          2414        0 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FCL INITIALIZATION ROUTINE.
 /FCL=ENT/        2414       40
 COMIO=           2454       33 SL-FORTRAN  11/16/78 COMPASS 3.6 485               COMMON CODED I/O ROUTINES AND CONSTANTS.
 FCL=FDL          2507       40 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FCL CAPSULE LOADING
 FEIFST=          2547        3 SL-FORTRAN  11/16/78 COMPASS 3.6 485                CONVERTED DATA STORAGE
 FLTOUT=          2552      311 SL-FORTRAN  11/16/78 COMPASS 3.6 485               COMMON FLOATING OUTPUT CODE
 FORSYS=          3063      301 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FORTRAN OBJECT LIBRARY UTILITIES.
 OUTCOM=          3364      154 SL-FORTRAN  11/16/78 COMPASS 3.6 485               COMMON OUTPUT CODE
 SYSAID=          3540        1 SL-FORTRAN  11/16/78 COMPASS 3.6 485               LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
 FECMSK=          3541       41 SL-FORTRAN  11/16/78 COMPASS 3.6 485               INITIALIZE CONSTANTS.
 FMTAP=           3602      357 SL-FORTRAN  11/16/78 COMPASS 3.6 485               CRACK APLIST AND FORMAT FOR KODER/KRAKER.
 FORUTL=          4161       46 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FCL MISC. UTILITIES.
 GETFIT=          4227       57 SL-FORTRAN  11/16/78 COMPASS 3.6 485               LOCATE AN FIT GIVEN A FILE NAME.
 KODER=           4306      451 SL-FORTRAN  11/16/78 COMPASS 3.6 485               OUTPUT FORMAT INTERPRETER.
 OUTC=            4757      150 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FORMATTED WRITE FORTRAN RECORD.
 /FDL.COM/        5127       14
 FDL.RES          5143      211 SL-SYSLIB   11/02/78 COMPASS 3.6 485               FAST DYNAMIC LOADER RESIDENT.
 FDL.MMI          5354      222 SL-SYSLIB   11/02/78 COMPASS 3.6 485               FDL MEMORY MANAGER INTERFACE.
 CPU.SYS          5576       40 SL-SYSLIB   11/15/78 COMPASS 3.6 476               PROCESS SYSTEM REQUEST.
 CMF.ALF          5636      160 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - ALLOCATE FIXED.
 CMF.CSF          6016        6 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - CHANGE SPECS FIXED.
 CMM.FFA          6024       14 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - FIXED FREE ALGORITHM.
 CMF.FRF          6040       36 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - FREE FIXED.
 CMM.R            6076      214 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - RESIDENT SUBROUTINES.
 CMF.SLF          6312       22 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - SHRINK AT LWA FIXED.
 CTL$RM           6334      433 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CRM CONTROLLING ROUTINE.
 ERR$RM           6767       25 SL-SYSLIB   11/16/78 COMPASS 3.6 485               CRM ERROR PROCESSOR ENTRY.
 LIST$RM          7014       67 SL-SYSLIB   11/16/78 COMPASS 3.6 485               CRM - ALLOCATE SPACE FOR LIST OF FILES
  //              7103       30


       .331 CP SECONDS                  23500B CM STORAGE USED                          12 TABLE MOVES
```

3

```
                                               IDENT   MACROE
                                               ENTRY   ADD
                                               LIST    M,-B,-R
                                       MOVE    MACROE  LENGTH,FROM,TO
                                               LOCAL   MOVE1
                                               SB1     0
                                               SB2     LENGTH
                                       MOVE1   SA5     FROM+B1
                                               BX6     X5
                                               SA6     TO+B1
                                               SB1     B1+1
                                               NE      B1,B2,MOVE1
                                               ENDM
     0                        20       ANSWER  BSSZ    16
    20  00000000000000000001           OPERAND DATA    1,2,3,4,5,6,7,8
    30  5110000020 +                   ADD     SA1     OPERAND+0
                   5120000021 +                SA2     OPERAND+1
    31  5130000022 +                           SA3     OPERAND+2
                   5140000023 +                SA4     OPERAND+3
    32  36612                                  IX6     X1+X2
             36734                             IX7     X3+X4
                   5160000000 +                SA6     ANSWER+0     STORE RESULT
    33  5170000001 +                           SA7     ANSWER+1     STORE RESULT
                   5110000024 +                SA1     OPERAND+4
    34  5120000025 +                           SA2     OPERAND+5
                   5130000026 +                SA3     OPERAND+6
    35  5140000027 +                           SA4     OPERAND+7
                   36612                       IX6     X1+X2
                        36734                  IX7     X3+X4
    36  5160000002 +                           SA6     ANSWER+2
                   5170000003 +                SA7     ANSWER+3
    37                                         MOVE    LENGTH=LBLOCK2,FROM=SUBT,TO=ADD
    37  6110000000                             SB1     0                                        MOVE     .1
                   6120000011                  SB2     LBLOCK2                                  MOVE     .1
    40  5151000044 +                   '7000001    SA5     SUBT+B1                              MOVE     .1
                   10655                       BX6     X5                                       MOVE     .1
    41  5161000030 +                           SA6     ADD+B1                                   MOVE     .1
                   6111000001                  SB1     B1+1                                     MOVE     .1
    42  0512000040 +                           NE      B1,B2,'7000001                           MOVE     .1
                                               ENDM                                             MOVE     .1
    43  0400000030 +                   ENDIBLK EQ      ADD                                      MOVE     .1
    44  5110000020 +                   SUBT    SA1     OPERAND+0
                   5120000021 +                SA2     OPERAND+1
    45  5130000022 +                           SA3     OPERAND+2
                   5140000023 +                SA4     OPERAND+3
    46  37621                                  IX6     X2-X1
             37734                             IX7     X3-X4
                   5160000004 +                SA6     ANSWER+4     STORE RESULT
    47  5170000005 +                           SA7     ANSWER+5     STORE RESULT
                                               MOVE    FROM=MULT,TO=ADD,LENGTH=LBLOCK3
                   6110000000                  SB1     0                                        MOVE     .1
    50  6120000011                             SB2     LBLOCK3                                  MOVE     .1
    51  5151000055 +                   '7000002    SA5     MULT+B1                              MOVE     .1
                   10655                       BX6     X5                                       MOVE     .1
    52  5161000030 +                           SA6     ADD+B1                                   MOVE     .1
                   6111000001                  SB1     B1+1                                     MOVE     .1
    53  0512000051 +                           NE      B1,B2,'7000002                           MOVE     .1
                                                                                                MOVE     .1
```

```
MACROE                                                                    COMPASS 3.6-476.     01/08/79  10.27.31.      PAGE     2

                                                      ENDM
 54  0400000030 +                            END2BLK  EQ      ADD                                                   MOVE     .1
 55  5110000020 +                            MULT     SA1     OPERAND+0
                5120000021 +                          SA2     OPERAND+1
 56  5130000022 +                                     SA3     OPERAND+2
                5140000023 +                          SA4     OPERAND+3
 57  42612                                            IX6     X1*X2
          42734                                       IX7     X3*X4
                5160000006 +                          SA6     ANSWER+6     STORE RESULTS
 60  5170000007 +                                     SA7     ANSWER+7     STORE RESULTS
                                                      LIST    -M
                6110000000                            MOVE    TO=ADD,LENGTH=LBLOCK4,FROM=DIVIDE
 65  0400000030 +                            END3BLK  EQ      ADD
 66  5110000020 +                            DIVIDE   SA1     OPERAND+0
                5120000021 +                          SA2     OPERAND+1
 67  5130000022 +                                     SA3     OPERAND+2
                5140000023 +                          SA4     OPERAND+3
 70  27101                                            PX1     X1
          27202                                       PX2     X2
                24101                                 NX1     X1
                      24202                           NX2     X2
 71  27303                                            PX3     X3
          27404                                       PX4     X4
                24303                                 NX3     X3
                      24404                           NX4     X4
 72  44621                                            FX6     X2/X1
          44743                                       FX7     X4/X3
                5160000010 +                          SA6     ANSWER+8     STORE RESULT(FLOATING POINT)
 73  5170000011 +                                     SA7     ANSWER+9     STORE RESULT(FLOATING POINT)
                26676                                 UX6     X6,B7
                      26767                           UX7     X7,B6
 74  22676                                            LX6     X6,B7
          22767                                       LX7     X7,B6
                5160000012 +                          SA6     ANSWER+10    STORE RESULT(INTEGER)
 75  5170000013 +                                     SA7     ANSWER+11    STORE RESULT(INTEGER)
 76  7160247021                              END4BLK  ENDRUN
                           14                LBLOCK1  EQU     END1BLK-ADD+1
                           11                LBLOCK2  EQU     END2BLK-SUBT+1
                           11                LBLOCK3  EQU     END3BLK-MULT+1
                           12                LBLOCK4  EQU     END4BLK-DIVIDE+2
                           50                LBLOCKT  EQU     LBLOCK1+LBLOCK2+LBLOCK3+LBLOCK4
                           14                MAXBLK   MAX     LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4
                           11                MINBLK   MIN     LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4
100                                                   END     ADD

                     47500B CM  STORAGE USED            111 STATEMENTS       21 SYMBOLS       000003 INVENTED SYMBOLS
                                MODEL 73 ASSEMBLY     0.331 SECONDS           0 REFERENCES


MACROE
ERROR DIRECTORY.                                                          COMPASS 3.6-476.     01/08/79  10.27.31.      PAGE     3

      3 TYPE ERROR              DUPLICATE MACRO DEFINITION.  NEW ONE OVERRIDES.
              OCCURRED ON PAGES      1
```

```
LOAD MAP - MACROE                                   CYBER LOADER 1.4-485      01/08/79  10.27.33.      PAGE      1


FWA OF THE LOAD           111
LWA+1 OF THE LOAD         251

TRANSFER ADDRESS -- ADD             141

PROGRAM ENTRY POINTS --       MACROE          141



PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK       ADDRESS  LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

MACROE          111      100    LGO      01/08/79 COMPASS 3.6 476
CPU.SYS         211       40 SL-SYSLIB   11/15/78 COMPASS 3.6 476                PROCESS SYSTEM REQUEST.


      .090 CP SECONDS           13500B CM STORAGE USED                     1 TABLE MOVE




DUMP    RELATIVE                          DMP(111.177)

    00111    00000 00000 00000 00003       00000 00000 00000 00007       00000 00000 00000 00013
    00114    00000 00000 00000 00017       00000 00000 00000 00001       77777 77777 77777 77776       00000 00000 00000 00002
    00120    00000 00000 00000 00014       17214 00000 00000 00000       17205 25252 52525 25252       00000 00000 00000 00002
    00124    00000 00000 00000 00001       00000 00000 00000 00000 00131_00000 00000 00000 00001       00000 00000 00000 00002
    00133    00000 00000 00000 00003
    00134    00000 00000 00000 00004       00000 00000 00000 00005       00000 00000 00000 00006       00000 00000 00000 00007
    00140    00000 00000 00000 00010       51100 00131 51200 00132       51300 00133 51400 00134       27101 27202 24101 24202
    00144    27303 27404 24303 24404       44621 44743 51600 00121       51700 00122 26676 26767       22676 22767 51600 00123
    00150    51700 00124 61000 46000       71602 47021 20650 46000       01000 00213 61100 46000       05120 00151 61000 46000
    00154    04000 00141 61000 46000       51100 00131 51200 00132       51300 00133 51400 00134       37621 37734 51600 00115
    00160    51700 00116 61100 00000       61200 00011 61000 46000       51510 00166 10655 46000       51610 00141 61110 00001
    00164    05120 00162 61000 46000       04000 00141 61000 46000       51100 00131 51200 00132       51300 00133 51400 00134
    00170    42612 42734 51600 00117       51700 00120 61100 00000       61200 00012 61000 46000       51510 00177 10655 46000
    00174    51610 00141 61110 00001       05120 00173 61000 46000       04000 00141 61000 46000       51100 00131 51200 00132
```



```
  MFS  NH1-  CY874-SN108    5C/R08    11/14/78
10.27.25.DUN00K9  FROM    /OH
10.27.25.IP  00000448 WORDS - FILE INPUT  , DC 04
10.27.25.DON,T5.   001A,6883,1896,MILLER
10.27.30.COMPASS.
10.27.32.          1 WARNING MESSAGE IN  MACROE
10.27.32. ASSEMBLY COMPLETE.   475008  CM USED.
10.27.32.    0.476  CPU SECONDS  ASSEMBLY TIME.
10.27.32.LGO.
10.27.33.DMP(111,177)
10.27.33.EXIT.
10.27.33.OP  00001408 WORDS - FILE OUTPUT , DC 40
10.27.33.MS    3584  WORDS  (    7168 MAX USED)
10.27.33.CPA         .587 SEC.         .587 ADJ.
10.27.33.CPB         .104 SEC.         .104 ADJ.
10.27.34.IO          .328 SEC.         .328 ADJ.
10.27.34.CM        18.316 KWS.        1.117 ADJ.
10.27.34.SS                           2.138
10.27.34.PP         3.913 SEC.     DATE 01/08/79
10.27.34.EJ  END OF JOB, OH
```

OPDEF'S

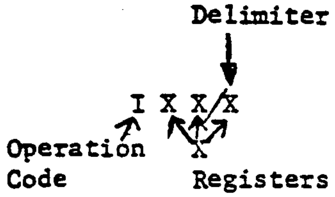
MACROS are sequences of code identified by a name: that is, whenever the name is used, the assembler brings in the associated code.

We can also associate a specific instruction Syntax with a sequence of code. Whenever an instruction matches the Syntax in the definition, the assembler will bring in the associated code.

For example, the instruction:

```
     DX6  X5/X4
```

follows the form:



There is no single hardware operation to perform this function (integer division). To handle this we set up a sequence of code called an OPDEF, which is similar to a MACRO.

The Syntax of OPDEF's must follow certain conventions. The Syntax can include both registers (A, B, X) and expressions (Q) as in the following example.

This OPDEF is designed to extract a certain set of bits from one X register and put them in another specified X register.

Syntax Parameters

OPDEF Definition

Syntax:

E = Operation Code

X = X register

Q = Expression

```
Exx,Q,Q      OPDEF   R1,R2, FIRST, LAST
              MX.R1   60-LAST-1
             BX.R1   -X.R1*X.R2
             AX.R1   FIRST
```

Variables:

R1 = Result Register

R2 = Operand Register

FIRST = beginning of bit stream (lower)

LAST = end of bit stream (upper)

```
ENDM
  .
  .
  .
  .
  .
  .
  .
  .
  .
```

OPDEF Call

```
EX2     X1,30,40
```

Extract bits 30 to 40 of X1 and put them in in X2

Since the Syntax of the instruction above matched the Syntax of the definition, the assembler generated these three lines of code inserting the given values (2, 1, 30, 40) for the dummy variables.

```
 MX.2   60-40-1     EX2     .1
BX.2    -X.2*X.1    EX2     .1
AX.2    30          EX2     .1
 ENDM               EX2     .1
```

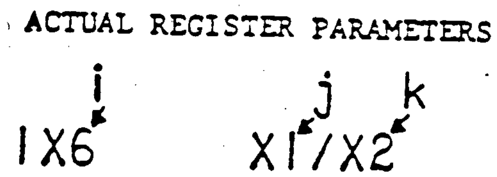
OPDEF

- A TYPE OF MACRO
- DEFINES A NEW INSTRUCTION or REDEFINES AN EXISTING INSTRUCTION

i.e.

INTEGER DIVIDE

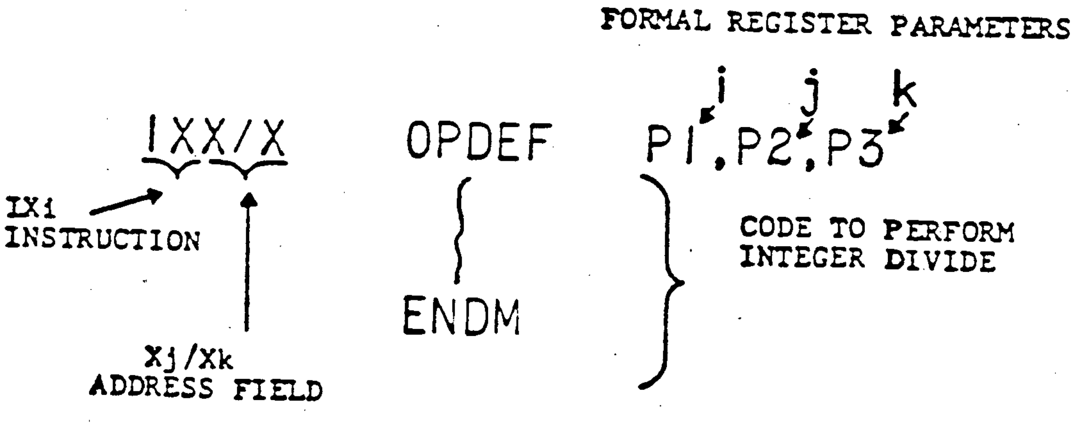
The Call:



ACTUAL REGISTER PARAMETERS

```
IX6        X1/X2
```

The Macro Definition:



FORMAL REGISTER PARAMETERS

```
IXX/X     OPDEF     P1,P2,P3

          ENDM
```

```
                                                           IDENT  OPDEF
                                                           ENTRY  INTDIVD
                                                           LIST   M
                                                   *
                                                   *
3                                                  *
                                                           OPDEF EXAMPLE - INTEGER DIVIDE
                                                    IXX/X  OPDEF  PARM1,PARM2,PARM3
                                                           PX.PARM2 X.PARM2
                                                           NX.PARM2 X.PARM2,B4
                                                           PX.PARM3 X.PARM3
                                                           NX.PARM3 X.PARM3,B5
                                                           RX.PARM1 X.PARM2/X.PARM3
                                                           UX.PARM1 X.PARM1,B6
                                                           LX.PARM1 X.PARM1,B6
                                                           ENDM
                                                   *
          0  46000                                 INTDIVD NO
                   5110000011 +                            SA1    NUMBERS
          1  5120000012 +                                  SA2    NUMBERS+1
                                                           IX6    X1/X2
                         27101                             PX.1 X.1                                                IX6     .1
                              24141                        NX.1 X.1,B4                                             IX6     .1
          2  27202                                         PX.2 X.2                                                IX6     .1
                  24252                                    NX.2 X.2,B5                                             IX6     .1
                        45612                              RX.6 X.1/X.2                                            IX6     .1
                              26666                        UX.6 X.6,B6                                             IX6     .1
          3  22666                                         LX.6 X.6,B6                                             IX6     .1
                                                           ENDM                                                    IX6     .1
                  5160000000 C                             SA6    OUT
          4  5130000013 +                                  SA3    NUMBERS+2
                        5150000014 +                       SA5    NUMBERS+3
          5                                                IX7    X3/X5
          5  27303                                         PX.3 X.3                                                IX7     .1
                  24343                                    NX.3 X.3,B4                                             IX7     .1
                        27505                              PX.5 X.5                                                IX7     .1
                              24555                        NX.5 X.5,B5                                             IX7     .1
          6  45735                                         RX.7 X.3/X.5                                            IX7     .1
                  26767                                    UX.7 X.7,B6                                             IX7     .1
                        22767                              LX.7 X.7,B6                                             IX7     .1
                                                           ENDM                                                    IX7     .1
          7  5170000001 C                                  SA7    OUT+1
                        7160247021                         ENDRUN
         11  00000000000000000010               NUMBERS    DATA   8,2,-5,3
                                                           USE    /ANSWERS/
          0                      2              OUT        BSSZ   2
         15                                                END    INTDIVD

                    47300B SCM STORAGE USED            49 STATEMENTS         4 SYMBOLS
                          MODEL 174 ASSEMBLY         0.049 SECONDS          11 REFERENCES
```

```
OPDEF                                                                      COMPASS 3.5-470.     06/27/78  16.31.13.      PAGE      3
ERROR DIRECTORY.

      3 TYPE ERROR                       DUPLICATE MACRO DEFINITION.  NEW ONE OVERRIDES.
            OCCURRED ON PAGES            2
```

```
LOAD MAP - OPDEF                                    CYBER LOADER 1.4-470      06/27/78  16.31.15.      PAGE      1


FWA OF THE LOAD              111
LWA+1 OF THE LOAD            170

TRANSFER ADDRESS -- INTDIVD           113

PROGRAM ENTRY POINTS --       OPDEF              113


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK          ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

/ANSWERS/         111        2
OPDEF             113       15    LGO      06/27/78 COMPASS 3.5 470
SYS.RM            130       40 SL-SYSLIB  05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.


     .023 CP SECONDS              13200B CM STORAGE USED                        1 TABLE MOVE



DUMP     RELATIVE                           DMP(111,177)

    00111    00000 00000 00000 00004          77777 77777 77777 77776          46000 51100 00124 46000
    00114    51200 00125 27101 24141          27202 24252 45612 24666          22666 51600 00111 46000          51300 00126 51500 00127

    00120    27303 24343 27505 24555          45735 24767 22767 46000          51700 00112 71602 47021          20650 01000 00132 46000

    00124    00000 00000 00000 00010          00000 00000 00000 00002          77777 77777 77777 77772          00000 00000 00000 00003

    00130    04000 00143 00000 00000          01300 00300 00000 00000          04000 00124 00000 00000          51100 00001 03110 00133

    00134    54610 04000 00131 46000          51100 00066 03110 00137          51100 00130 04000 00140          71100 00130 20160 46000

    00140    13661 13161 13661 46000          51600 00131 10611 46000          51100 00001 01000 00130          20652 01000 00132 46000

    00144    51100 00001 03110 00144          04004 00145 61000 46000          51100 00001 03110 00145          71602 20314 04000 00143

    00150    20150 36661 01000 00132          04004 00151 61000 46000          71602 20314 20652 36662          53160 20173 03310 00151

    00154    03010 00151 51100 00001          03110 00153 71100 00001          04000 00150 61000 46000          71603 24616 12661 20651

    00160    01000 00132 61000 46000          04004 00161 61000 46000          73660 20630 12161 73610          20123 03210 00157 20:51

    00164    13116 20636 51600 00167          74660 36116 20123 46000          04000 00157 61000 46000          00000 00000 00000 00000

    00170    60000 00000 04004 00170     00177>60000 00000 04004 00177
```

```
OPDEF                                                            COMPASS 3.5-470.     06/27/78  16.31.13.       PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH              BINARY CONTROL CARDS.

                0       15              IDENT  OPDEF
               15                       END    INTDIVD

                                BLOCKS     TYPE       ADDRESS    LENGTH

                                PROGRAM*   LOCAL            0        15
                                ANSWERS    COMMON           0         2

                                ENTRY POINTS.

                                INTDIVD          0*

                                EXTERNAL SYMBOLS.

                                SYS=
```

```
OPDEF                                                            COMPASS 3.5-470.     06/27/78  16.31.13.       PAGE      4
SYMBOLIC REFERENCE TABLE.

INTDIVD        0     PROGRAM*        2/02 E     2/17 L
NUMBERS       11     PROGRAM*        2/18       2/19       2/30       2/31       2/43 L
OUT            0     ANSWERS         2/29 S     2/41 S     2/45 L
SYS=           0     EXTERNAL*       2/43
```

```
  MFF   NB2-  CYB175-SN1    4LH7/H6B  05/15/78
16.31.11.DONOOMY  FROM    /SH
16.31.11.IP  00000256 WORDS - FILE INPUT  , DC 04
16.31.11.DON.     PSC.02/B.72CT011A.MILLER
16.31.12.COMPASS.
16.31.13.           1 WARNING MESSAGE IN  OPDEF
16.31.14. ASSEMBLY COMPLETE.   47300B SCM USED.
16.31.14.    0.102  CPU SECONDS  ASSEMBLY TIME.
16.31.14.LGO.
16.31.15.DMP(111,177)
16.31.15.OP  00000108 WORDS - FILE OUTPUT , DC 40
16.31.15.MS    3584  WORDS  (   10752 MAX USED)
16.31.15.CPA        .152 SEC.          .152 ADJ.
16.31.15.IO         .635 SEC.          .635 ADJ.
16.31.15.CM       10.583 KWS.          .645 ADJ.
16.31.15.SS                           1.433
16.31.15.PP        3.832 SEC.     DATE 06/27/78
16.31.15.EJ  END OF JOB, SH
```

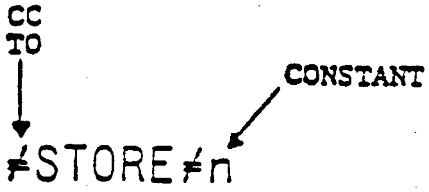
MICRO

- USE A SYMBOL NAME
- TO GET A STRING OF CHARACTERS

i.e.

STORE

The Call:



CC TO

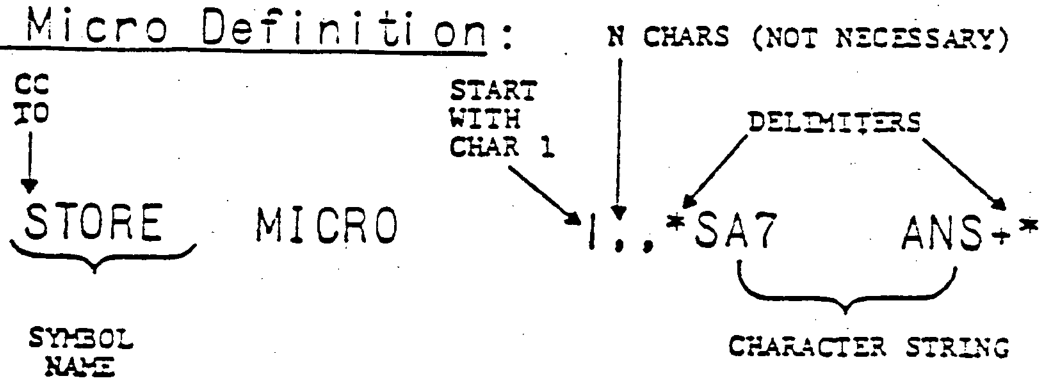
CONSTANT

≠STORE≠n

GENERATES THE CODE:

```
        SA7            ANS+n
```

The Micro Definition:



CC TO

N CHARS (NOT NECESSARY)

START WITH CHAR 1

DELIMITERS

```
STORE   MICRO    1,,*SA7      ANS+*
```

SYMBOL NAME

CHARACTER STRING

```
                                        IDENT  MICRO
                                 ************************************************************
                                 *                                                          *
                                 *       THE CHARACTER STRING OF THE MICRO IS INSERTED WHERE IT IS CALLED   *
                                 *       BY THE #NNNNNNNNNNNNNN#.  THE LIST A LISTS THE MICRO EXPANSION     *
                                 *                                                          *
                                 ************************************************************
                                        ENTRY  START
                                        LIST   A,D
                                 LOAD   MICRO  1,,*SA1        A1+B*
                                 STORE  MICRO  1,,*SA7        ANSWER+*

  0  6110000001                  START  SB1    1
               6120000002               SB2    2
                         12 +    DATA   LIT    2,4,5,6
  1  5110000012 +                       SA1    DATA
               10711                    BX7    X1
                                        #STORE#0
  2  5170000006 +                       SA7       ANSWER+0
                               .

                                        #LOAD#1
               54111                    SA1       A1+B1
                               .
                    10711               BX7    X1

                                        #STORE#1
  3  5170000007 +                       SA7       ANSWER+1
                               .

                                        #LOAD#2
               54112                    SA1       A1+B2
                               .
                    10711               BX7    X1

                                        #STORE#2
  4  5170000010 +                       SA7       ANSWER+2
                               .

               7160247021               ENDRUN
  6                      4     ANSWER   BSSZ   4
                               DEFAULT SYMBOLS DEFINED BY COMPASS.

  0 X                          SYS=

                               CONTENT OF LITERALS BLOCK.

 12  00000000000000000002               B
 13  00000000000000000004               D
 14  00000000000000000005               E
 15  00000000000000000006               F

 16                                     END    START

            41700B SCM STORAGE USED                41 STATEMENTS          4 SYMBOLS
```

```
LOAD MAP - MICRO                                        CYBER LOADER 1.4-470       06/27/78  16.31.28.       PAGE      1


FWA OF THE LOAD               111
LWA+1 OF THE LOAD             167

TRANSFER ADDRESS -- START              111

PROGRAM ENTRY POINTS --        MICRO             111


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS   LENGTH    FILE      DATE     PROCSSR VER LEVEL  HARDWARE   COMMENTS

MICRO           111        16    LGO      06/27/78 COMPASS 3.5 470
SYS.RM          127        40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.


     .023 CP SECONDS             13200B CM STORAGE USED                     1 TABLE MOVE
```

```
DUMP    RELATIVE                          DMP(111,127)

    00111   61100 00001 61200 00002        51100 00123 10711 46000        51700 00117 54111 10711
    00114   51700 00120 54112 10711        51700 00121 71602 47021        20650 01000 00131 46000        00000 00000 00000 00002

    00120   00000 00000 00000 00004        00000 00000 00000 00006        00000 00000 00000 00000        00000 00000 00000 00002

    00124   00000 00000 00000 00004        00000 00000 00000 00005        00000 00000 00000 00006        04000 00142 00000 00000
```

```
  MFF   NR2-  CY8175-SN1    4L87/H6B  05/15/78
16.31.25.DON00H7  FROM    /SH
16.31.25.IP  00000256 WORDS - FILE INPUT  . DC 04
16.31.25.DON.     P50,0278,72CT011A,MILLER
16.31.27.COMPASS.
16.31.27. ASSEMBLY COMPLETE.   47300B SCM USED.
16.31.27.    0.094  CPU SECONDS  ASSEMBLY TIME.
16.31.27.LGO.
16.31.28.DMP(111,127)
16.31.28.OP  00000896 WORDS - FILE OUTPUT . DC 40
16.31.28.MS    3584  WORDS  (   10752 MAX USED)
16.31.28.CPA          .146 SEC.          .146 ADJ.
16.31.28.IO           .632 SEC.          .632 ADJ.
16.31.28.CM      10.516 KWS.             .641 ADJ.
16.31.28.SS                             1.421
16.31.28.PP          3.074 SEC.     DATE 06/27/78
16.31.28.EJ  END OF JOB. SH
```

```
                                              IDENT  CONVERT
                                       ***********************************************************************
                                       *                                                                     *
                                       *                 MICRO AND DUP PSEUDO OPERATIONS                     *
                                       *                                                                     *
                                       *       CREATE A EXTERNAL BCD TO DISPLAY CODE CONVERSION TABLE        *
                                       *                                                                     *
                                       ***********************************************************************
                                              LIST   D,G
     0                                 TABLE  BSS    0
                          D^D                 CODE   DISPLAY CODE
                            1          D      SET    1
                           35                 DUP    35B
                                       CHAR   MICRO  D,1,$:1234567890=≤%[ /STUVWXYZ],($
                                              DATA   1R#CHAR#
                                       D      SET    D+1
                                              ENDD
     0  00000000000000000000                  DATA   1R:                                          *DUP*    .1
     1  00000000000000000034                  DATA   1R1                                          *DUP*    .1
     2  00000000000000000035                  DATA   1R2                                          *DUP*    .1
     3  00000000000000000036                  DATA   1R3                                          *DUP*    .1
     4  00000000000000000037                  DATA   1R4                                          *DUP*    .1
     5  00000000000000000040                  DATA   1R5                                          *DUP*    .1
     6  00000000000000000041                  DATA   1R6                                          *DUP*    .1
     7  00000000000000000042                  DATA   1R7                                          *DUP*    .1
    10  00000000000000000043                  DATA   1R8                                          *DUP*    .1
    11  00000000000000000044                  DATA   1R9                                          *DUP*    .1
    12  00000000000000000033                  DATA   1R0                                          *DUP*    .1
    13  00000000000000000054                  DATA   1R=                                          *DUP*    .1
    14  00000000000000000074                  DATA   1R≤                                          *DUP*    .1
    15  00000000000000000063                  DATA   1R%                                          *DUP*    .1
    16  00000000000000000061                  DATA   1R[                                          *DUP*    .1
    17  00000000000000000055                  DATA   1R                                           *DUP*    .1
    20  00000000000000000050                  DATA   1R/                                          *DUP*    .1
    21  00000000000000000023                  DATA   1RS                                          *DUP*    .1
    22  00000000000000000024                  DATA   1RT                                          *DUP*    .1
    23  00000000000000000025                  DATA   1RU                                          *DUP*    .1
    24  00000000000000000026                  DATA   1RV                                          *DUP*    .1
    25  00000000000000000027                  DATA   1RW                                          *DUP*    .1
    26  00000000000000000030                  DATA   1RX                                          *DUP*    .1
    27  00000000000000000031                  DATA   1RY                                          *DUP*    .1
    30  00000000000000000032                  DATA   1RZ                                          *DUP*    .1
    31  00000000000000000062                  DATA   1R]                                          *DUP*    .1
    32  00000000000000000056                  DATA   1R,                                          *DUP*    .1
    33  00000000000000000051                  DATA   1R(                                          *DUP*    .1
    34  00000000000000000055                  DATA   1R                                           *DUP*    .1
    35  00000000000000000065                  DATA   65B           DISPLAY CODE FOR →
                            1          D      SET    1
                           42                 DUP    42B
                                       CHAR   MICRO  D,1,/≡∧-JKLMNOPQR∨$*↑↓>≥ABCDEFGHI<.)≥ ¬/
                                              DATA   1R#CHAR#
                                       D      SET    D+1
                                              ENDD
    36  00000000000000000060                  DATA   1R≡                                          *DUP*    .1
    37  00000000000000000067                  DATA   1R∧                                          *DUP*    .1
    40  00000000000000000046                  DATA   1R-                                          *DUP*    .1
    41  00000000000000000012                  DATA   1RJ                                          *DUP*    .1
```

```
 42  00000000000000000013          DATA   1RK
 43  00000000000000000014          DATA   1RL                                              *DUP*   .1
 44  00000000000000000015          DATA   1RM                                              *DUP*   .1
 45  00000000000000000016          DATA   1RN                                              *DUP*   .1
 46  00000000000000000017          DATA   1RO                                              *DUP*   .1
 47  00000000000000000020          DATA   1RP                                              *DUP*   .1
 50  00000000000000000021          DATA   1RQ                                              *DUP*   .1
 51  00000000000000000022          DATA   1RR                                              *DUP*   .1
 52  00000000000000000066          DATA   1R∨                                              *DUP*   .1
 53  00000000000000000053          DATA   1R$                                              *DUP*   .1
 54  00000000000000000047          DATA   1R*                                              *DUP*   .1
 55  00000000000000000070          DATA   1R↑                                              *DUP*   .1
 56  00000000000000000071          DATA   1R↓                                              *DUP*   .1
 57  00000000000000000073          DATA   1R>                                              *DUP*   .1
 60  00000000000000000045          DATA   1R+                                              *DUP*   .1
 61  00000000000000000001          DATA   1RA                                              *DUP*   .1
 62  00000000000000000002          DATA   1RB                                              *DUP*   .1
 63  00000000000000000003          DATA   1RC                                              *DUP*   .1
 64  00000000000000000004          DATA   1RD                                              *DUP*   .1
 65  00000000000000000005          DATA   1RE                                              *DUP*   .1
 66  00000000000000000006          DATA   1RF                                              *DUP*   .1
 67  00000000000000000007          DATA   1RG                                              *DUP*   .1
 70  00000000000000000010          DATA   1RH                                              *DUP*   .1
 71  00000000000000000011          DATA   1RI                                              *DUP*   .1
 72  00000000000000000074          DATA   1R≤                                              *DUP*   .1
 73  00000000000000000057          DATA   1R.                                              *DUP*   .1
 74  00000000000000000052          DATA   1R)                                              *DUP*   .1
 75  00000000000000000075          DATA   1R≥                                              *DUP*   .1
 76  00000000000000000055          DATA   1R                                               *DUP*   .1
 77  00000000000000000077          DATA   1R;                                              *DUP*   .1
                          0-0      CODE   OTHER                                            *DUP*   .1
                        00-33      CHAR   0,33B        INTERCHANGE COLON AND
                        33-00      CHAR   33B,0        ZERO FOR CODE OTHER
                            1   D  SET    1
                           35      DUP    35B
                            CHAR   MICRO  D,1,$:1234567890=≤%[ /STUVWXYZ],($
                                   DATA   1R"CHAR"
                                D  SET    D+1
                                   ENDD
100  00000000000000000033          DATA   1R:
101  00000000000000000034          DATA   1R1                                              *DUP*   .1
102  00000000000000000035          DATA   1R2                                              *DUP*   .1
103  00000000000000000036          DATA   1R3                                              *DUP*   .1
104  00000000000000000037          DATA   1R4                                              *DUP*   .1
105  00000000000000000040          DATA   1R5                                              *DUP*   .1
106  00000000000000000041          DATA   1R6                                              *DUP*   .1
107  00000000000000000042          DATA   1R7                                              *DUP*   .1
110  00000000000000000043          DATA   1R8                                              *DUP*   .1
111  00000000000000000044          DATA   1R9                                              *DUP*   .1
112  00000000000000000000          DATA   1R0                                              *DUP*   .1
113  00000000000000000054          DATA   1R=                                              *DUP*   .1
114  00000000000000000074          DATA   1R<                                              *DUP*   .1
115  00000000000000000063          DATA   1R%                                              *DUP*   .1
116  00000000000000000061          DATA   1R[                                              *DUP*   .1
117  00000000000000000055          DATA   1R                                               *DUP*   .1
120  00000000000000000050          DATA   1R/                                              *DUP*   .1
                                                                                           *DUP*   .1
```

```
  122  00000000000000000024                  DATA    1RT                                                        *DUP*     .1
  123  00000000000000000025                  DATA    1RU                                                        *DUP*     .1
  124  00000000000000000026                  DATA    1RV                                                        *DUP*     .1
  125  00000000000000000027                  DATA    1RW                                                        *DUP*     .1
  126  00000000000000000030                  DATA    1RX                                                        *DUP*     .1
  127  00000000000000000031                  DATA    1RY                                                        *DUP*     .1
  130  00000000000000000032                  DATA    1RZ                                                        *DUP*     .1
  131  00000000000000000062                  DATA    1R]                                                        *DUP*     .1
  132  00000000000000000056                  DATA    1R.                                                        *DUP*     .1
  133  00000000000000000051                  DATA    1R(                                                        *DUP*     .1
  134  00000000000000000055                  DATA    1R                                                         *DUP*     .1
                          0-D                CODE    *
                            1      D         SET     1
                           35                DUP     35B
                                   CHAR      MICRO   D,1,$:1234567890=≤≠[ /STUVWXYZ].($
                                             DATA    1R"CHAR"
                                   D         SET     D+1
                                             ENDD
  135  00000000000000000000                  DATA    1R:                                                        *DUP*     .1
  136  00000000000000000034                  DATA    1R1                                                        *DUP*     .1
  137  00000000000000000035                  DATA    1R2                                                        *DUP*     .1
  140  00000000000000000036                  DATA    1R3                                                        *DUP*     .1
  141  00000000000000000037                  DATA    1R4                                                        *DUP*     .1
  142  00000000000000000040                  DATA    1R5                                                        *DUP*     .1
  143  00000000000000000041                  DATA    1R6                                                        *DUP*     .1
  144  00000000000000000042                  DATA    1R7                                                        *DUP*     .1
  145  00000000000000000043                  DATA    1R8                                                        *DUP*     .1
  146  00000000000000000044                  DATA    1R9                                                        *DUP*     .1
  147  00000000000000000033                  DATA    1R0                                                        *DUP*     .1
  150  00000000000000000054                  DATA    1R=                                                        *DUP*     .1
  151  00000000000000000074                  DATA    1R≤                                                        *DUP*     .1
  152  00000000000000000063                  DATA    1R≠                                                        *DUP*     .1
  153  00000000000000000061                  DATA    1R[                                                        *DUP*     .1
  154  00000000000000000055                  DATA    1R                                                         *DUP*     .1
  155  00000000000000000050                  DATA    1R/                                                        *DUP*     .1
  156  00000000000000000023                  DATA    1RS                                                        *DUP*     .1
  157  00000000000000000024                  DATA    1RT                                                        *DUP*     .1
  160  00000000000000000025                  DATA    1RU                                                        *DUP*     .1
  161  00000000000000000026                  DATA    1RV                                                        *DUP*     .1
  162  00000000000000000027                  DATA    1RW                                                        *DUP*     .1
  163  00000000000000000030                  DATA    1RX                                                        *DUP*     .1
  164  00000000000000000031                  DATA    1RY                                                        *DUP*     .1
  165  00000000000000000032                  DATA    1RZ                                                        *DUP*     .1
  166  00000000000000000062                  DATA    1R]                                                        *DUP*     .1
  167  00000000000000000056                  DATA    1R.                                                        *DUP*     .1
  170  00000000000000000051                  DATA    1R(                                                        *DUP*     .1
  171  00000000000000000055                  DATA    1R                                                         *DUP*     .1
  172                                  END

                  47300B SCM STORAGE USED            284 STATEMENTS         2 SYMBOLS
                           MODEL 174 ASSEMBLY      0.356 SECONDS          247 REFERENCES
```

```
CONVERT                                                     COMPASS 3.5-470.     06/27/78  08.40.57.      PAGE      1
STORAGE ALLOCATION.

         ADDRESS   LENGTH             BINARY CONTROL CARDS.

               0      172              IDENT   CONVERT
             172                   END

CONVERT                                                     COMPASS 3.5-470.     06/27/78  08.40.57.      PAGE      5
SYMBOLIC REFERENCE TABLE.

D          36                      2/12 D     2/34       2/56       3/14 D     3/30       3/54       4/12 D     4/34 D
                                   2/19       2/34 D     2/56 D     3/15       3/30 D     3/54 D     4/13 D     4/35
                                   2/19 D     2/35       2/57       3/15 D     3/31       3/55       4/20       4/35 D
                                   2/20       2/35 D     2/57 D     3/16       3/31 D     3/55 D     4/20 D     4/36
                                   2/20 D     2/36       3/01       3/16 D     3/34 D     3/56       4/21       4/36 D
                                   2/21       2/36 D     3/01 D     3/17       3/41       3/56 D     4/21 D     4/37
                                   2/21 D     2/37       3/02       3/17 D     3/41 D     3/57       4/22       4/37 D
                                   2/22       2/37 D     3/02 D     3/18       3/42       3/57 D     4/22 D     4/38
                                   2/22 D     2/38       3/03       3/18 D     3/42 D     4/01       4/23       4/38 D
                                   2/23       2/38 D     3/03 D     3/19       3/43       4/01 D     4/23 D     4/39
                                   2/23 D     2/39       3/04       3/19 D     3/43 D     4/02       4/24       4/39 D
                                   2/24       2/39 D     3/04 D     3/20       3/44       4/02 D     4/24 D     4/40
                                   2/24 D     2/40       3/05       3/20 D     3/44 D     4/03       4/25       4/40 D
                                   2/25       2/40 D     3/05 D     3/21       3/45       4/03 D     4/25 D     4/41
                                   2/25 D     2/41       3/06       3/21 D     3/45 D     4/04       4/26       4/41 D
                                   2/26       2/41 D     3/06 D     3/22       3/46       4/04 D     4/26 D     4/42
                                   2/26 D     2/42       3/07       3/22 D     3/46 D     4/05       4/27       4/42 D
                                   2/27       2/42 D     3/07 D     3/23       3/47       4/05 D     4/27 D     4/43
                                   2/27 D     2/43       3/08       3/23 D     3/47 D     4/06       4/28       4/43 D
                                   2/28       2/43 D     3/08 D     3/24       3/48       4/06 D     4/28 D     4/44
                                   2/28 D     2/44       3/09       3/24 D     3/48 D     4/07       4/29       4/44 D
                                   2/29       2/44 D     3/09 D     3/25       3/49       4/07 D     4/29 D     4/45
                                   2/29 D     2/45       3/10       3/25 D     3/49 D     4/08       4/30       4/45 D
                                   2/30       2/45 D     3/10 D     3/26       3/50       4/08 D     4/30 D     4/46
                                   2/30 D     2/46       3/11       3/26 D     3/50 D     4/09       4/31       4/46 D
                                   2/31       2/46 D     3/11 D     3/27       3/51       4/09 D     4/31 D     4/47
                                   2/31 D     2/47       3/12       3/27 D     3/51 D     4/10       4/32       4/47 D
                                   2/32       2/47 D     3/12 D     3/28       3/52       4/10 D     4/32 D     4/48
                                   2/32 D     2/48 D     3/13       3/28 D     3/52 D     4/11       4/33       4/48 D
                                   2/33       2/55       3/13 D     3/29       3/53       4/11 D     4/33 D
                                   2/33 D     2/55 D     3/14       3/29 D     3/53 D     4/12       4/34
TABLE      0     PROGRAM*          2/10 L

                                                           MFF   NB2-  CYB175-SN1    4LB7/R68  05/15/78
                                                         08.40.54.DON003M  FROM    /SH
                                                         08.40.54.IP  00000320 WORDS - FILE INPUT  , DC 04
                                                         08.40.54.DON.     PSD,0278,72CT011A,MILLER
                                                         08.40.56.COMPASS.
                                                         08.40.58. ASSEMBLY COMPLETE.   47300B SCM USED.
                                                         08.40.58.     0.418  CPU SECONDS  ASSEMBLY TIME.
                                                         08.40.58.OP  00002560 WORDS - FILE OUTPUT , DC 40
                                                         08.40.58.MS    3584  WORDS  (     7168 MAX USED)
                                                         08.40.58.CPA          .442 SEC.          .442 ADJ.
                                                         08.40.58.IO           .506 SEC.          .506 ADJ.
                                                         08.40.58.CM         16.572 KWS.         1.011 ADJ.
                                                         08.40.58.SS                             1.960
                                                         08.40.58.PP          3.151 SEC.     DATE 06/27/78
                                                         08.40.58.EJ  END OF JOB, SH
```

```
PROGRAM SWITCH      74/176  OPT=1                         FTN 4.7+470        06/27/78  16.32.03        PAGE      1

                  PROGRAM SWITCH(INPUT,OUTPUT)
                  COMMON A(12)
                  PRINT 100
              100 FORMAT(1H1)
                  READ 7,(A(I),I=1,12)
                  PRINT 7,(A(I),I=1,12)
                  CALL TRAPOS
                7 FORMAT(4F20.0)
                  END
```

```
SUBROUTINE UPRINT1     74/176  OPT=1                      FTN 4.7+470        06/27/78  16.32.03        PAGE      1

                  SUBROUTINE UPRINT1
                  COMMON /BLOCK1/STORE(12)
                  PRINT 5,(STORE(I),I=1,12)
                5 FORMAT(1X,4F20.4,////)
                  END
```

```
MICRO                                                   COMPASS 3.5-470.    06/27/78  16.32.05.       PAGE      1
STORAGE ALLOCATION.

            ADDRESS   LENGTH               BINARY CONTROL CARDS.

                  0       23               IDENT  MICRO
                 23                        END

                        BLOCKS     TYPE       ADDRESS    LENGTH

                        PROGRAM*   LOCAL            0        23
                        BLOCK1     COMMON           0        14
                        //         COMMON           0        14

                        ENTRY POINTS.

                        TRAPOS            0+

                        EXTERNAL SYMBOLS.

                        UPRINT1
```

```
                                                   IDENT  MICRO
                                                   ENTRY  TRAPOS
                                                   EXT    UPRINT1
                                                   LIST   M
                                                   LIST   A
                                          ROW1     MICRO  1,,*M,M+3,M+6,M+9*
                                          ROW2     MICRO  1,,*M+1,M+4,M+7,M+10*
                                          ROW3     MICRO  1,,*M+2,M+5,M+8,M+11*

                                          SAM      MACRO  A,B,C,D
                                                   SA1    A
                                                   SA2    B
                                                   SA3    C
                                                   SA4    D
                                                   BX6    X1
                                                   BX7    X2
                                                   SA6    B1
                                                   SA7    B1+1
                                                   BX6    X3
                                                   BX7    X4
                                                   SA6    B1+2
                                                   SA7    B1+3
                                                   ENDM

   0  00000000000000000000                TRAPOS   DATA   0
   1  6110000000 C                                 SB1    STORE
                                                   SAM    ≠ROW1≠
                                                   SAM    M,M+3,M+6,M+9

              5110000000 C                         SA1    M                                              SAM       .1
   2  5120000003 C                                 SA2    M+3                                            SAM       .1
              5130000006 C                         SA3    M+6                                            SAM       .1
   3  5140000011 C                                 SA4    M+9                                            SAM       .1
              10611                                BX6    X1                                             SAM       .1
                   10722                           BX7    X2                                             SAM       .1
   4  56610                                        SA6    B1                                             SAM       .1
          5171000001                               SA7    B1+1                                           SAM       .1
                   10633                           BX6    X3                                             SAM       .1
   5  10744                                        BX7    X4                                             SAM       .1
          5161000002                               SA6    B1+2                                           SAM       .1
   6  5171000003                                   SA7    B1+3                                           SAM       .1
                                                   ENDM                                                  SAM       .1

              6110000004 C                         SB1    STORE+4
                                                   LIST   -A
   7                                               SAM    ≠ROW2≠
   7  5110000001 C                                 SA1    M+1                                            SAM       .1
              5120000004 C                         SA2    M+4                                            SAM       .1
  10  5130000007 C                                 SA3    M+7                                            SAM       .1
              5140000012 C                         SA4    M+10                                           SAM       .1
  11  10611                                        BX6    X1                                             SAM       .1
          10722                                    BX7    X2                                             SAM       .1
              56610                                SA6    B1                                             SAM       .1
  12  5171000001                                   SA7    B1+1                                           SAM       .1
              10633                                BX6    X3                                             SAM       .1
                   10744                           BX7    X4                                             SAM       .1
  13  5161000002                                   SA6    B1+2                                           SAM       .1
```

```
MICRO                                                     COMPASS 1.5-470.    06/27/78  16.32.05.      PAGE     3

            5171000003             SA7    A1+3                                                        SAM      .1
                                   ENDM                                                               SAM      .1

   14  6110000010 C                SB1    STORE+B
                                   LIST   -M
            5110000002 C           SAM    #ROW3#

            0100000000 X           RJ     UPPRINT1
   22  0400000000 +                EQ     TRAPOS
                                   LIST   *
   23                      LASTWRD BSS    0           GET ADDRESS OF NEXT WORD
                     23    AA      DECMIC LASTWRD-TRAPOS,4  GET DECIMAL LENGTH OF PROGRAM
                     23    AB      OCTMIC LASTWRD-TRAPOS,6  GET OCTAL LENGTH OF PROGRAM
                           RAG5    MICRO  1,,*#AA# IS THE DECIMAL LENGTH OF THIS PROGRAM*
                           RAG5    MICRO  1,,*0019 IS THE DECIMAL LENGTH OF THIS PROGRAM*
                           RAG6    MICRO  1,,*#AB# IS THE OCTAL LENGTH OF THIS PROGRAM*
                           RAG6    MICRO  1,,*000023 IS THE OCTAL LENGTH OF THIS PROGRAM*

                     52    DECLENG MICCNT RAG5        NUMBER OF CHARACTERS IN RAG5
                     52    OCTLENG MICCNT RAG6        NUMBER OF CHARACTERS IN RAG6
                     52    RAG7    DECMIC DECLENG,4   GET DECIMAL CHARACTER COUNT
                     52    RAG8    OCTMIC OCTLENG,4   GET OCTAL CHARACTER COUNT
                           RAG9    MICRO  1,,*#RAG7# DECIMAL CHARACTERS IN RAG7*
                           RAG9    MICRO  1,,*0042 DECIMAL CHARACTERS IN RAG7*
                           RAG10   MICRO  1,,*#RAG8# OCTAL CHARACTERS IN RAG8*
                           RAG10   MICRO  1,,*0052 OCTAL CHARACTERS IN RAG8*
                                   USE    /BLOCK1/
    0                14    STORE   BSS    12
                                   USE    //
    0                14    M       BSS    12
                                   USE    0
   23                              END

                474008 SCM STORAGE USED           101 STATEMENTS         7 SYMBOLS
                       MODEL 174 ASSEMBLY      0.118 SECONDS          31 REFERENCES


MICRO                                                      COMPASS 3.5-470.    06/27/78  16.32.05.      PAGE     4
SYMBOLIC REFERENCE TABLE.

DECLENG      52                       3/19 D     3/21
LASTWRD      23       PROGRAM*        3/11 L     3/12       3/13
M             0       //              2/30       2/32       2/47       2/49       3/06       3/07
                                      2/31       2/33       2/48       2/50       3/07       3/30 L
OCTLENG      52                       3/20 D     3/22
STORE         0       BLOCK1          2/26       2/44       3/04       3/28 L
TRAPOS        0       PROGRAM*        2/02 E     2/25 L     3/09       3/12       3/13
UPPRINT1      0       EXTERNAL*       2/03 X     3/08
```

```
LOAD MAP - SWITCH                                        CYBER LOADER 1.4-470        06/27/78  16.12.09.       PAGE      1


FWA OF THE LOAD              111
LWA+1 OF THE LOAD          12631

TRANSFER ADDRESS -- SWITCH           4250

PROGRAM ENTRY POINTS --        SWITCH          4250



PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK         ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

SWITCH            111     4171    LGO      06/27/78 FTN      4.7 470    767X 1     PROGRAM   OPT=1
/BLOCK1/         4302       14
UPRINT1          4316       15    LGO      06/27/78 FTN      4.7 470    767X 1     SUBROUTINEOPT=1
MICRO            4333       23    LGO      06/27/78 COMPASS 3.5 470
SYSAID=          4356        1 SL-FORTRAN  05/02/78 COMPASS 3.5 470                LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
/STP.END/        4357        1
/FCL.C./         4360       25
/OB.IO./         4405       77
Q2NTRY=          4504        0 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FCL INITIALIZATION ROUTINE.
/FCL=ENT/        4504       42
COMIO=           4546       14 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL=FDL          4562       40 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FCL CAPSULE LOADING
FLTIN=           4622      156 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON FLOATING INPUT CONVERTER.
FMTAP=           5000      377 SL-FORTRAN  05/02/78 COMPASS 3.5 470                CHACK APLIST AND FORMAT FOR KODER/KRAKER.
FORUTL=          5377       46 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FCL MISC. UTILITIES.
GETFIT=          5445       61 SL-FORTRAN  05/02/78 COMPASS 3.5 470                LOCATE AN FIT GIVEN A FILE NAME.
KRAKER=          5526      454 SL-FORTRAN  05/02/78 COMPASS 3.5 470                PROCESS FORMATTED FORTRAN INPUT.
OUTC=            6202      150 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FORMATTED WRITE FORTRAN RECORD.
FECMSK=          6352       41 SL-FORTRAN  05/02/78 COMPASS 3.5 470                INITIALIZE CONSTANTS.
FLTOUT=          6413      315 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON FLOATING OUTPUT CODE
FORSYS=          6730      302 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FORTRAN OBJECT LIBRARY UTILITIES.
INCOM=           7232      147 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON INPUT FORMATTING CODE
INPC=            7401      207 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FORMATTED READ FORTRAN RECORD.
KODER=           7610      461 SL-FORTRAN  05/02/78 COMPASS 3.5 470                OUTPUT FORMAT INTERPRETER.
OUTCOM=         10271      204 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON OUTPUT CODE
CMF.ALF         10475      160 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - ALLOCATE FIXED.
CMF.CSF         10655        6 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - CHANGE SPECS FIXED.
CMM.FFA         10663       14 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - FIXED FREE ALGORITHM.
CMF.FRF         10677       36 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - FREE FIXED.
CMM.R           10735      213 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF         11150       22 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - SHRINK AT LWA FIXED.
CTL$RM          11172      601 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM CONTROLLING ROUTINE.
ERR$RM          11773       25 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM ERROR PROCESSOR ENTRY.
LIST$RM         12020       66 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM - ALLOCATE SPACE FOR LIST OF FILES
/FDL.COM/       12106       14
FDL.RES         12122      211 SL-SYSLIB   05/16/78 COMPASS 3.5 470                FAST DYNAMIC LOADER RESIDENT.
FDL.MMI         12333      222 SL-SYSLIB   05/16/78 COMPASS 3.5 470                FDL MEMORY MANAGER INTERFACE.
SYS.RM          12555       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.
 //             12615       14


       .158 CP SECONDS                 27100B CM STORAGE USED                             32 TABLE MOVES
```

```
        1.                  2.                  3.                  4.
        5.                  6.                  7.                  8.
        9.                 10.                 11.                 12.
   1.0000              4.0000              7.0000             10.0000


   2.0000              5.0000              8.0000             11.0000


   3.0000              6.0000              9.0000             12.0000
```

```
  MFF   NB2-  CYB175-SN1    4L87/R6d  05/15/78
 16.32.01.DON00M0  FROM    /SH
 16.32.01.IP  00000448 WORDS - FILE INPUT  , DC 04
 16.32.01.DON.     PSD,0278,72CT011A,MILLER
 16.32.03.FTN,R=0.
 16.32.06.      .210 CP SECONDS COMPILATION TIME
 16.32.06.MAP,PART.
 16.32.06.LGO.
 16.32.10.     END  SWITCH
 16.32.10.      .036 CP SECONDS EXECUTION TIME
 16.32.11.OP  00002048 WORDS - FILE OUTPUT , DC 40
 16.32.11.MS    3584  WORDS  (   17920 MAX USED)
 16.32.11.CPA        .431 SEC.          .431 ADJ.
 16.32.11.IO        2.108 SEC.         2.108 ADJ.
 16.32.11.CM       43.356 KWS.         2.646 ADJ.
 16.32.11.SS                           5.186
 16.32.11.PP        7.892 SEC.     DATE 06/27/78
 16.32.11.EJ  END OF JOB, SH
```

```
IRPECHO                                                               COMPASS 3.6-476.     01/08/79  10.25.31.      PAGE      1

                                          IDENT  IRPECHO
                                          ENTRY  P
                                  **************************************************************************
                                  *                                                                        *
                                  *                                                                        *
                                  *       EXAMPLES OF IRP AND ECHO PSEUDO OPERATIONS                       *
                                  *                                                                        *
                                  *                                                                        *
                                  **************************************************************************

                                   IRPMO  MACRO  ARG,PARM     IRP MACRO
                                          IRP    ARG
                                          SA1    ARG
                                          SX6    X1+PARM
                                          SA6    ARG
                                          SA2    DATA
                                          BX7    X2
                                          SA7    DATA+1
                                          IRP                 END OF IRP RANGE
                                          ENDM
                                  ECHOMO  MACRO  P1,P2,P3     ECHO MACRO
                                          ECHO   3,ARG=(P1,P2,P3)
                                          SA1    ARG
                                          SX6    X1+1
                                          SA6    ARG
                                          SA2    DATA
                                          BX7    X2
                                          SA7    DATA+1
                                          ENDM
                                  ***
     0  46000                      P      NO
                                  *
                                          IRPMO  (J,K,L),1    CALL IRP MACRO
                                          IRP    J,K,L                                         IRPMO    .1
             5110000000 C                 SA1    J                                             IRPMO    .1
     1  7261000001                        SX6    X1+1                                          IRPMO    .1
                      5160000000 C        SA6    J                                             IRPMO    .1
     2  5120000003 C                      SA2    DATA                                          IRPMO    .1
                      10722               BX7    X2                                            IRPMO    .1
     3  5170000004 C                      SA7    DATA+1                                        IRPMO    .1
                      5110000001 C        SA1    K                                             IRPMO    .1
     4  7261000001                        SX6    X1+1                                          IRPMO    .1
                      5160000001 C        SA6    K                                             IRPMO    .1
     5  5120000003 C                      SA2    DATA                                          IRPMO    .1
                      10722               BX7    X2                                            IRPMO    .1
     6  5170000004 C                      SA7    DATA+1                                        IRPMO    .1
                      5110000002 C        SA1    L                                             IRPMO    .1
     7  7261000001                        SX6    X1+1                                          IRPMO    .1
                      5160000002 C        SA6    L                                             IRPMO    .1
    10  5120000003 C                      SA2    DATA                                          IRPMO    .1
                      10722               BX7    X2                                            IRPMO    .1
    11  5170000004 C                      SA7    DATA+1                                        IRPMO    .1
                                          IRP                 END OF IRP RANGE                 IRPMO    .1
                                          ENDM                                                 IRPMO    .1
```

```
IRPECHO                                                        COMPASS 3.6-476.      01/08/79  10.25.31.      PAGE     2

                                           ECHOMO J.K.L       CALL ECHO MACRO
                                           ECHO   3.ARG=(J,K.L)                                           ECHOMO   .1
                                           SA1    ARG                                                     ECHOMO   .1
                                           SX6    X1+1                                                    ECHOMO   .1
                                           SA6    ARG                                                     ECHOMO   .1
                  5110000000 C             SA1    J                                                       *ECHO*   .2
    12  7261000001                         SX6    X1+1                                                    *ECHO*   .2
                  5160000000 C             SA6    J                                                       *ECHO*   .2
    13  5110000001 C                       SA1    K                                                       *ECHO*   .2
                  7261000001               SX6    X1+1                                                    *ECHO*   .2
    14  5160000001 C                       SA6    K                                                       *ECHO*   .2
                  5110000002 C             SA1    L                                                       *ECHO*   .2
    15  7261000001                         SX6    X1+1                                                    *ECHO*   .2
                  5160000002 C             SA6    L                                                       *ECHO*   .2
    16  5120000003 C                       SA2    DATA                                                    ECHOMO   .1
                  10722                    BX7    X2                                                      ECHOMO   .1
    17  5170000004 C                       SA7    DATA+1                                                  ECHOMO   .1
                                           ENDM                                                           ECHOMO   .1
                                 *
                  7160247021               ENDRUN
                                 *
                                           USE    /IRPECHO/
     0  00000000000000000000      J        DATA   0
     1  00000000000000000000      K        DATA   0
     2  00000000000000000000      L        DATA   0
     3  00000000000000000000    DATA       DATA   +0,-0
    21                                     END    P

                     47600B CM  STORAGE USED                85 STATEMENTS        6 SYMBOLS
                                MODEL 73 ASSEMBLY        0.304 SECONDS           0 REFERENCES
```

```
  MFS  NB1-  CYB74-SN108    SC/R08    11/14/78
10.23.42.DON00K2  FROM    /OH
10.23.43.IP  00000320 WORDS - FILE INPUT  , DC 04
10.23.43.DON,TS.   001A,6883,1896,MILLER
10.23.53.REWIND,OUTPUT.
10.25.30.COMPASS,LO=BREM.
10.25.32. ASSEMBLY COMPLETE.  47600B  CM USED.
10.25.32.     0.492  CPU SECONDS  ASSEMBLY TIME.
10.25.32.OP  00000896 WORDS - FILE OUTPUT , DC 40
10.25.32.MS     3584  WORDS  (    7168 MAX USED)
10.25.32.CPA        .292 SEC.          .292 ADJ.
10.25.32.CPB        .327 SEC.          .327 ADJ.
10.25.32.IO         .311 SEC.          .311 ADJ.
10.25.32.CM       17.411 KWS.         1.062 ADJ.
10.25.32.SS                           1.994
10.25.32.PP        4.309 SEC.     DATE 01/08/79
10.25.32.EJ  END OF JOB,$OH
```

LESSON 10

CONDITIONAL ASSEMBLY

LESSON PREVIEW:

THIS LESSON COVERS THE DEFINITION AND PSEUDO OPERATIONS THAT ARE USED IN CONJUNCTION WITH CONDITIONAL ASSEMBLIES.

REFERENCES:

CHAPTER 4	COMPASS REFERENCE MANUAL #60492600

TRAINING AIDS:

VISUAL SET 10

PROJECTS:

PROGRAMMING PROJECT 6A AND 6B

OBJECTIVES:

AT THE COMPLETION OF THIS LESSON THE STUDENT WILL BE ABLE TO:

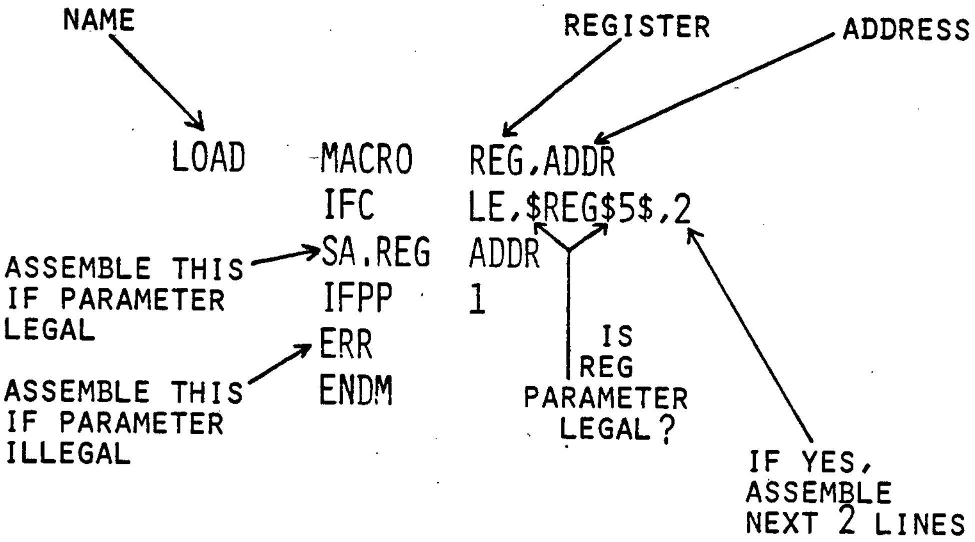
1. WRITE A PROGRAM THAT UTILIZES:

   A. ENDIF
   B. ELSE
   C. IFTYPE
   D. IFOPERATION
   E. IFCONDITION

2. USE CONDITIONAL ASSEMBLY

CONDITIONAL ASSEMBLY

I.E., TEST LEGALITY OF PARAMETERS PASSED TO MACRO



```
LOAD   MACRO  REG,ADDR
       IFC    LE,$REG$5$,2
       SA.REG ADDR
       IFPP   1
       ERR
       ENDM
```

NAME

REGISTER

ADDRESS

ASSEMBLE THIS IF PARAMETER LEGAL

ASSEMBLE THIS IF PARAMETER ILLEGAL

IS REG PARAMETER LEGAL?

IF YES, ASSEMBLE NEXT 2 LINES

THE CALLS:

```
LOAD   2,Y
SA.2   Y
```

A LEGAL CALL GENERATES THIS CODE

```
LOAD   7,Y
ERR
```

AN ILLEGAL CALL GENERATES THIS CODE

```
IFC/PP                                          COMPASS 3.5-470.     06/27/78  16.20.24.      PAGE      2

                                   IDENT  IFC/PP
                                   ENTRY  START
                                   LIST   *
                           ********************************************************************************
                           *                                                                              *
                           *                                                                              *
                           *       IFC PSEUDO OPERATION - COMPARES TWO CHARACTER STRING ACCORDING TO THE*
                           *       OPERATOR SPECIFIED AND ASSEMBLES INSTRUCTIONS IN THE IF RANGE IF THE *
                           *       COMPARISON IS SATISFIED.                                               *
                           *                                                                              *
                           *       IFTYPE PSEUDO INSTRUCTION TEST FOR THE TYPE OF PROCESSOR THAT WILL    *
                           *       EXECUTE THE OBJECT PROGRAM, AS DECLARED BY MACHINE, AND PERIPH OR PPU
                           *       PSEUDO INSTRUCTIONS.                                                   *
                           *                                                                              *
                           *       PP - ANY PERIPHERAL PROCESSOR UNIT                                     *
                           *                                                                              *
                           ********************************************************************************

                            LOAD     MACRO  REG,ADDR
                                     IFC    LE,$REG$0$,1
                                     ERR
                                     IFC    LE,$REG$5$,2           TEST PARAMETERS
                                     SA.REG ADDR
                                     IFPP   1
                                     ERR
                                     ENDM
                           *
      0  00000000000000000003       Y        DATA   3
      1  46000                      START    NO
                                     LOAD   2,Y          LEGAL CODE
                                     IFC    LE,$2$0$,1                                                LOAD    .1
                                     ERR                                                              LOAD    .1
                                     IFC    LE,$2$5$,2           TEST PARAMETERS                      LOAD    .1
               5120000000 +          SA.2 Y                                                           LOAD    .1
                                     IFPP   1                                                         LOAD    .1
                                     ERR                                                              LOAD    .1
                                     ENDM                                                             LOAD    .1

                                     LOAD   7,Y          ILLEGAL CALL
                                     IFC    LE,$7$0$,1                                                LOAD    .1
                                     ERR                                                              LOAD    .1
                                     IFC    LE,$7$5$,2           TEST PARAMETERS                      LOAD    .1
                                     SA.7 Y                                                           LOAD    .1
                                     IFPP   1                                                         LOAD    .1
                                     ERR                                                              LOAD    .1
                                     ENDM                                                             LOAD    .1
                                     ENDRUN
      2  7160247021                SX6 3REND*4+1                                                      ENDRUN  .1
                    20650          LX6 40D                                                            ENDRUN  .1
      3  0100000000 X              RJ =XSYS=                                                          ENDRUN  .1
                                   ENDM                                                               ENDRUN  .1

                           DEFAULT SYMBOLS DEFINED BY COMPASS.

      0 X                          SYS=

                                     END    START
```

P

```
IFC/PP                                                          COMPASS 3.5-470.     06/27/78  16.20.24.      PAGE      1
STORAGE ALLOCATION.

           ADDRESS   LENGTH             BINARY CONTROL CARDS.

                 0        4             IDENT   IFC/PP
                 4                      END     START

                                ENTRY POINTS.

                                START          1*

                                EXTERNAL SYMBOLS.

                                SYS=
```

```
IFC/PP                                                          COMPASS 3.5-470.     06/27/78  16.20.24.      PAGE      3

                    47300B SCM STORAGE USED               52 STATEMENTS          3 SYMBOLS
                           MODEL 174 ASSEMBLY          0.050 SECONDS             5 REFERENCES

                                      1 ERROR IN  IFC/PP
```

```
IFC/PP                                                          COMPASS 3.5-470.     06/27/78  16.20.24.      PAGE      4
ERROR DIRECTORY.

     P TYPE ERROR               CONSULT LISTINGS FOR REASON BEHIND P-ERROR.
           OCCURRED ON PAGES          2
```

```
IFC/PP                                                          COMPASS 3.5-470.     06/27/78  16.20.24.      PAGE      5
SYMBOLIC REFERENCE TABLE.

START           1      PROGRAM*        2/02 E     2/29 I
SYS=            0      EXTERNAL*       2/50
Y               0      PROGRAM*        2/28 L     2/34
```

```
 MFF   NB2-  CYA179-SN1    4LB7/R6B  05/15/78
16.20.22.DON00MM  FROM    /SH
16.20.22.IP  00000320 WORDS - FILE INPUT  , DC 04
16.20.22.DON.      PSD,0278,72CT011A,MILLER
16.20.24.COMPASS.
16.20.24.            1 ERROR IN  IFC/PP
16.20.24. ASSEMBLY ERRORS.     47300B SCM USED.
16.20.24.    0.100  CPU SECONDS  ASSEMBLY TIME.
16.20.24.OP  00000896 WORDS - FILE OUTPUT , DC 40
16.20.24.MS    3584  WORDS  (    7168 MAX USED)
16.20.24.CPA         .126 SEC.          .126 ADJ.
16.20.24.IO          .497 SEC.          .497 ADJ.
16.20.24.CM         8.848 KWS.          .540 ADJ.
16.20.24.SS                            1.164
16.20.24.PP         2.612 SEC.     DATE 06/27/78
16.20.24.EJ  END OF JOB, SH
```

```
                                                IDENT   CONDASM
                                                ENTRY   CONDASM
                                                LIST    D,G
                                        ****************************************************************
                                        *                                                              *
                                        *                                                              *
                                        *       IFLE AND IFGT TEST VALUE OF PARAMETER VAL TO DETERMINE WHETHER TO SET*
                                        *       AN X REGISTER WITH A CONSTANT (<18 BITS) OR SET AN A REGISTER TO A   *
                                        *       LITERAL (>18 BITS).  NOTE CODE GENERATED BY EACH CALL.  THE LIST G   *
                                        *       CAUSES ONLY THE CODE GENERATED TO BE LISTED, NOT ALL THE CODE.       *
                                        *                                                              *
                                        *                                                              *
                                        ****************************************************************

                                         SETX    MACRO   REG,VAL
                                                 IFLE    VAL,777777B,1
                                                 SX.REG  VAL          SET X IF A SMALL NUMBER
                                                 IFGT    VAL,777777B,1
                                                 SA.REG  =VAL         SET A IF LARGE NUMBER
                                                 ENDM

    0                          3         ANSWER  BSSZ    3
    3                                    CONDASM SETX    1,777776B
    3  7110777776                                SX.1 777776B           SET X IF A SMALL NUMBER            SETX    .1
                 10611                           BX6     X1
    4  5160000000 +                              SA6     ANSWER+0
                                        *
                                                 SETX    2,5477766B
              5120000011 +                       SA.2 =5477766B         SET A IF LARGE NUMBER              SETX    .1
    5  10622                                     BX6     X2
           5160000001 +                          SA6     ANSWER+1

                                                 SETX    3,1000000B
    6  5130000012 +                              SA.3 =1000000B         SET A IF LARGE NUMBER              SETX    .1
                 10633                           BX6     X3
    7  5160000002 +                              SA6     ANSWER+2
                                                 LIST    -G
              7160247021                         ENDRUN

                                        DEFAULT SYMBOLS DEFINED BY COMPASS.

    0 X                                 SYS=

                                        CONTENT OF LITERALS BLOCK.

   11  00000000000005477766                   E*1v
   12  00000000000001000000                   A

   13                                            END     CONDASM

                               47100B SCM STORAGE USED            45 STATEMENTS          3 SYMBOLS
                                      MODEL 174 ASSEMBLY       0.045 SECONDS             7 REFERENCES
```

```
LOAD MAP - CONDASM                                          CYBER LOADER 1.4-470       06/27/78  09.08.28.      PAGE      1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD          164

TRANSFER ADDRESS -- CONDASM          114

PROGRAM ENTRY POINTS --       CONDASM          114


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK         ADDRESS   LENGTH    FILE      DATE     PROCSSR VER LEVEL  HARDWARE   COMMENTS

CONDASM          111       13    LGO      06/27/78 COMPASS 3.5 470
SYS.RM           124       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                 PROCESS SYSTEM REQUEST.


     .025 CP SECONDS              132000 CM STORAGE USED                    1 TABLE MOVE


DUMP    RELATIVE                         DMP(111,127)

    00111   77777 77777 77777 77776        00000 00000 00054 77766        00000 00000 00010 00000
    00114   71107 77776 10611 46000        51600 00111 51200 00122        10622 51600 00112 46000        51300 00123 10633 46000

    00120   51600 00113 71602 47021        20650 01000 00126 46000        00000 00000 00054 77766        00000 00000 00010 00000

    00124   04000 00137 00000 00000        01300 00000 00000 00000        04000 00122 00000 00000        51100 00001 03110 00127
```

```
  MFF   NA2-  CY8175-SN1    4LB7/R6B  05/15/78
09.08.25.DON0059  FROM    /SM
09.08.25.IP  00000320 WORDS - FILE INPUT  , DC 04
09.08.25.DON.      PSO,0278,72CT011A,MILLER
09.08.26.COMPASS.
09.08.27. ASSEMBLY COMPLETE.   47300B SCM USED.
09.08.27.     0.097  CPU SECONDS  ASSEMBLY TIME.
09.08.27.LGO.
09.08.28.DMP(111,127)
09.08.28.OP  00000896 WORDS - FILE OUTPUT , DC 40
09.08.28.MS    3584  WORDS  (    10752 MAX USED)
09.08.28.CPA        .150 SEC.           .150 ADJ.
09.08.28.IO         .433 SEC.           .433 ADJ.
09.08.28.CM       10.637 KWS.           .649 ADJ.
09.08.28.SS                            1.433
09.08.28.PP        3.029 SEC.     DATE 06/27/78
09.08.28.EJ  END OF JOB, SM
```

Conditional Assembly
COMPASS Manual, page 4-63 ff

CONDITIONAL ASSEMBLY

Certain parts of a MACRO code sequence may not be needed every time the MACRO is called. The COMPASS assembler can check for conditions as it is assembling the code and assemble only the appropriate code. For instance, we might want to check for missing or incorrect variables. In the AVG MACRO that we created, we could put in these tests:

```
AVG       MACRO     LOC1,NUM,LOC2
          IFC       EQ,*LOC1**,1
A         ERR       ASSEMBLE IF LOC1 MISSING
          IF        -ABS,NUM,1
A         ERR       ASSEMBLE IF NUM NOT ABSOLUTE
          IFC       EQ,*LOC2**,1
A         ERR       ASSEMBLE IF LOC2 MISSING
          SA1       LOC1           PICK UP SUM
          SX2       NUM            NO. OF ITEMS
          IX6       X1/X2          AVE=SUM/NO.
          SA6       LOC2           STORE AVE
          ENDM
```

In the Extract OPDEF we could test for other conditions:

```
EXXQQ     OPDEF     R1,R2,FIRST,LAST
          IFGT      R1,7,1
A         ERR
          IF        -DEF,FIRST,1
FIRST     SET       0
CHECK2    IF        -DEF,LAST
LAST      SET       59
CHECK2    ENDIF
          IFGT      FIRST,LAST
          MX.R1     60-FIRST-1
          BX.R1     -X.R1*X.R2
          AX.R1     LAST
          ELSE
          MX.R1     60-LAST-1
          BX.R1     -X.R1*X.R2
          AX.R1     FIRST
          ENDIF
          ENDM
```

Conditional tests are line oriented. Testing can be limited by a line count as in:

```
IFC          EQ,*LOC1**,2
```

NOTE

2 = Line count so assemble next two lines of test if true.

Testing can also be limited by an ENDIF, when no line count is specified as in:

```
IFGT         FIRST,LAST
MX.R1        60-FIRST-1
BX.R1        -X.R1*X.R2
AX.R1        LAST
ELSE
MX.R1        60-LAST-1
BX.R1        -X.R1*X.R2
AX.R1        FIRST
ENDIF
```

IF statements without tags (as above) are terminated by an unlabeled ENDIF. IF statements with tags are terminated by an ENDIF with a matching tag, as in:

```
CHECK2       IF           -DEF,LAST
LAST         SET          59
CHECK2       ENDIF
```

The effect of an IF test can be reversed by the ELSE statement as in:

```
If test was true, assemble.        IFGT         FIRST,LAST
If test was false, skip.           MX.R1        60-FIRST-1
                                   BX.R1        -X.R1*X.R2
If test was true, skip.            AX.R1        LAST
If test was false, assemble.       ELSE
                                   MX.R1        60-LAST-1
                                   BX.R1        -X.R1*X.R2
                                   AX.R1        FIRST
                                   ENDIF
```

```
ANTIBUG                                                                     COMPASS 3.5-470.     06/27/78  09.05.14.       PAGE     1
STORAGE ALLOCATION.

           ADDRESS   LENGTH                    BINARY CONTROL CARDS.

                 0        13                   IDENT   ANTIBUG
                13                             END     NO TRANSFER ADDRESS FOR SUBROUTINE

                                     BLOCKS     TYPE        ADDRESS    LENGTH

                                     PROGRAM*   LOCAL             0        13
                                     NUMBERS    COMMON            0         2

                                     ENTRY POINTS.

                                     ANTIBUG           0*
```

```
                                             IDENT  ANTIBUG
                                             ENTRY  ANTIBUG
                                             LIST   S
                                      ****************************************************************
                                      *                                                              *
                                      *                                                              *
                                      *       USE OF THE IF CONDITIONAL PSEUDO OPERATION             *
                                      *                                                              *
                                      *       PROGRAM SHOWS AN ANTIBUGGING TECHNIQUE FOR A COMPASS SUBROUTINE
                                      *                                                              *
                                      *       THIS EXAMPLE USES ANTIBUGGING CODE TO PREVENT UNEXPLAINED ERRORS.I.E.*
                                      *       A SUBROUTINE MAY INCLUDE THIS CODE TO TEST FOR NON-NUMERIC DISPLAY   *
                                      *       CODE CHARACTER AND ABORT THE JOB IF FOUND.  IF THE CHECK IS NOT DE-  *
                                      *       SIRED, THE STATEMENT #DEBUG EQU 1# IS REMOVED AND THE CHECK IS NOT   *
                                      *       MADE.                                                  *
                                      *                                                              *
                                      *                                                              *
                                      ****************************************************************
                                             USE    /NUMBERS/
 0                      2     DATAIN         BSS    2
                                             USE    0
                        1     DEBUG          EQU    1
 0  46000                     ANTIBUG        NO
 1  7100000077                *              SX0    77B          SET UP MASK
              5110000000 C                   SA1    DATAIN
 2  11201                                    BX2    X0*X1        ISOLATE LOW ORDER CHARACTER OF X1
                              *
                                             IF     DEF,DEBUG,5  TEST IF DEBUG IS SELECTED
        63620                                SB6    X2           TEST IF X2 IS BETWEEN 0 AND 9
              6170000033                     SB7    1R0
 3  0767777776                               LT     B6,B7,-1     ABORT JOB IF X2 LT DISPLAY CODE 0
              6170000044                     SB7    1R9
 4  0776777776                               GT     B6,B7,-1     ABORT JOB IF X2 GT DISPLAY CODE 9
                              *
                                             IF     DEF,DEBUG
              20106                          LX1    6
                   11201                     BX2    X0*X1
 5  63620                                    SB6    X2
        6170000033                           SB7    1R0
 6  0767777776                               LT     B6,B7,-1
              6170000044                     SB7    1R9
 7  0776777776                               GT     B6,B7,-1
                                             ENDIF
                                             IF     DEF,DEBUG,5
              63620                          SB6    X2
10  6170000033                               SB7    1R0
              0767777776                     LT     B6,B7,-1
                                             ELSE   2
                                             SB7    1R9
                                             GT     B6,B7,-1
                                             IF     -DEF,DEBUG,5
                                             SB6    X2
                                             SB7    1R0
                                             LT     B6,B7,-1
                                             ELSE
11  6170000044                               SB7    1R9
              0776777776                     GT     B6,B7,-1
```

```
ANTIBUG
                                                                COMPASS 3.5-470.     06/27/78  09.05.14.      PAGE     3

                                       .        CONTINUE PROGRAM
                                       .
   12  0400000000 +                             EQ      ANTIBUG     RETURN TO CALLER
   13                                           END     NO TRANSFER ADDRESS FOR SUBROUTINE

                        47100B SCM STORAGE USED               61 STATEMENTS          3 SYMBOLS
                                MODEL 176 ASSEMBLY        0.057 SECONDS           10 REFERENCES
```

```
ANTIBUG
SYMBOLIC REFERENCE TABLE.                                       COMPASS 3.5-470.     06/27/78  09.05.14.      PAGE     4

ANTIBUG         0     PROGRAM*         2/02 F    2/21 L    3/03
DATAIN          0     NUMBERS          2/20 L    2/25
DEBUG           1                      2/22 D    2/28 F    2/35 F    2/44 F    2/51 F
```

```
LOAD MAP - ANTIBUG
                                                           CYBER LOADER 1.4-470       06/27/78  09.05.16.      PAGE     1

FWA OF THE LOAD            111
LWA+1 OF THE LOAD          126

TRANSFER ADDRESS -- ANTIBUG           113

PROGRAM ENTRY POINTS --       ANTIBUG         113


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK           ADDRESS   LENGTH     FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

/NUMBERS/          111        2
ANTIBUG            113       13      LGO       06/27/78 COMPASS 3.5 470


     .014 CP SECONDS              13200B CM STORAGE USED                        1 TABLE MOVE
```

```
DMPX.

P    000000  A0  000200  B0  000000
RA   264600  A1  000111  B1  000001     C(A1)=  6000  0000  0004  0040  0111     C(B1)=  0000  0000  0000  0000  0000
FL   000200  A2  000060  B2  000002     C(A2)=  1505  1520  0000  0000  0061     C(B2)=  0000  0000  0000  0000  0000.
EM   700700  A3  000057  B3  012733     C(A3)=  0000  0000  0000  0000  0000     C(B3)=
RE   000055  A4  000001  B4  000201     C(A4)=  0000  0000  0000  0000  0000     C(B4)=
FE   000000  A5  000113  B5  000113     C(A5)=  4600  0460  0061  0004  6000     C(B5)=  4600  0460  0061  0004  6000
MA   001200  A6  000001  B6  000011     C(A6)=  0000  0000  0000  0000  0000     C(B6)=  0000  0000  0000  0000  0000
                 A7  000001  B7  000033     C(A7)=  0000  0000  0000  0000  0000     C(B7)=  0000  0000  0000  0000  0000
X0   0000  0000  0000  0000  0077
X1   6000  0000  0004  0040  0111
X2   0000  0000  0000  0000  0011
X3   0000  0000  0000  0000  0000
X4   0000  0000  0000  0000  0000
X5   6000  0000  0004  0040  0000
X6   1505  1520  0000  0000  0061
X7   0000  0000  0000  0000  0000

   00000    00017 77776 00000 00000        00000 00000 00000 00000
   00054   -56110 03110 00054 54710        51100 00001 03110 00055        64550 02550 00000 46000        00000 00000 00000 00000
   00060    15051 52000 00000 00061        00000 00200 00000 00001        07040 00060 51600 00001        04000 00063 00000 00021
   00064    14071 70000 00000 00000        00000 00000 00000 00126        40000 00000 02000 00111        00000 00000 40000 00000
   00070    05300 50325 24055 10116        24110 22507 52550 00000        00000 00000 00000 00000
   00100   -54000 00000 01000 00001

DUMP     RELATIVE                          DMP,111,177.

    00111    60000 00000 04004 00111   00113>46000 46000 61000 46000
    00114    71000 00077 51100 00111         11201 63620 61700 00033        07677 77776 61700 00044        07767 77776 20106 11201

    00120    63620 61700 00033 46000         07677 77776 61700 00044        07767 77776 63620 46000        61700 00033 07677 77776

    00124    61700 00044 07767 77776         04000 00113 61000 46000        60000 00000 04004 00126   00177>60000 00000 04004 00177


                                                      MFF   NR2-  CYB175-SN1    4LB7/R6B  05/15/78
                                                     09.05.10.DON005Y  FROM    /SH
                                                     09.05.10.IP  00000512 WORDS - FILE INPUT  , DC 04
                                                     09.05.10.DON.      PSC,027A,72CT011A,MILLER
                                                     09.05.13.COMPASS.
                                                     09.05.14. ASSEMBLY COMPLETE.   47300B SCM USED.
                                                     09.05.15.     0.108  CPU SECONDS  ASSEMBLY TIME.
                                                     09.05.15.LOAD(LGO)
                                                     09.05.15.EXECUTE(ANTIBUG)
                                                     09.05.16.ERROR MODE =01.  ADDRESS =777776
                                                     09.05.16.EXIT.
                                                     09.05.16.DMP,111,177.
                                                     09.05.16.OP  00001200 WORDS - FILE OUTPUT , DC 40
                                                     09.05.16.MS       3584  WORDS  (    10752 MAX USED)
                                                     09.05.16.CPA           .146 SEC.            .146 ADJ.
                                                     09.05.17.IO            .650 SEC.            .650 ADJ.
                                                     09.05.17.CM         10.612 KWS.             .647 ADJ.
                                                     09.05.17.SS                                1.444
                                                     09.05.17.PP           5.597 SEC.     DATE 06/27/78
                                                     09.05.17.EJ  END OF JOB, SH
```

```
      PROGRAM INITIAL     74/74    OPT=1                    FTN 4.6+444       02/25/77  16.09.59       PAGE     1

   1              PROGRAM INITIAL(OUTPUT)
                  COMMON /ANSWER/ BUFFER(16)
                  CALL START
                  PRINT 100,(BUFFER(I),I=1,16)
   5          100 FORMAT(4O22)
                  END
```

```
    SUBROUTINE ERRMSG1     74/74    OPT=1                   FTN 4.6+444       02/25/77  16.09.59       PAGE     1

   1              SUBROUTINE ERRMSG1
                  PRINT 100
              100 FORMAT(1H1,* ECS WRITE ERROR*)
                  RETURN
   5              END
```

```
    SUBROUTINE ERRMSG2     74/74    OPT=1                   FTN 4.6+444       02/25/77  16.09.59       PAGE     1

   1              SUBROUTINE ERRMSG2
                  PRINT 100
              100 FORMAT(1H1,* ECS READ ERROR*)
                  RETURN
   5              END
```

```
LOC7600                                                COMPASS 3.4-444.     02/25/77  16.10.08.       PAGE     1
STORAGE ALLOCATION.

             ADDRESS   LENGTH            BINARY CONTROL CARDS.

                   0       56            IDENT  LOC7600
                  56                     END

                                BLOCKS     TYPE       ADDRESS    LENGTH

                                PROGRAM*   LOCAL            0        56
                                ANSWER     COMMON           0        20

                                ENTRY POINTS.

                                START            10*

                                EXTERNAL SYMBOLS.

                                ERRMSG1    ERRMSG2
```

```
                                                IDENT  LOC7600
                                                MACHINE 7
                                                ENTHY  START
                                                EXT    ERRMSG1•ERRMSG2
                                                USE    /ANSWER/
      0                     20       ANSWER     BSSZ   16
                                                USE    0
      0  00000000000000000001        OPERAND    DATA   1•2•3•4•5•6•7•8
     10  00000000000000000000        START      DATA   0
     11  5100000014 •                           SA0    ADD
                   7100000000                   SX0    0
                                                IFCP6  4
                                                IFCP7  2
     12  0120000042                             WL     LBLOCKT
                   0110000012                   RL     LBLOCK1
     13  0400000014 •                           EQ     ADD
     14  5110000000 •                ADD        SA1    OPERAND•0
                   5120000001 •                 SA2    OPERAND•1
     15  5130000002 •                           SA3    OPERAND•2
                   5140000003 •                 SA4    OPERAND•3
     16  36612    .                             IX6    X1•X2
              36734                             IX7    X3•X4
                   5160000000 C                 SA6    ANSWER•0
     17  5170000001 C                           SA7    ANSWER•1
                   5110000004 •                 SA1    OPERAND•4
     20  5120000005 •                           SA2    OPERAND•5
                   5130000006 •                 SA3    OPERAND•6
     21  5140000007 •                           SA4    OPERAND•7
                   36612                        IX6    X1•X2
                        36734                   IX7    X3•X4
     22  5160000002 C                           SA6    ANSWER•2
                   5170000003 C                 SA7    ANSWER•3
     23  5100000014 •                           SA0    ADD
                   7100000012                   SX0    LBLOCK1
                                                IFCP6
                                                ENDIF
                                                IFCP7  1
     24  0110000007                             RL     LBLOCK2
     25  0100000014 •                END1BLK    RJ     ADD
L    14                                         LOC    ADD
L    14  00000000000000000000        SUBT       DATA   0
L    15  5110000000 •                           SA1    OPERAND•0
                   5120000001 •                 SA2    OPERAND•1
L    16  5130000002 •                           SA3    OPERAND•2
                   5140000003 •                 SA4    OPERAND•3
L    17  37621                                  IX6    X2-X1
             37734                              IX7    X3-X4
                   5160000004 C                 SA6    ANSWER•4
L    20  5170000005 C                           SA7    ANSWER•5
                   5100000014 •                 SA0    ADD
L    21  7100000021                             SX0    LBLOCK1•LBLOCK2
                                                IFCP6  2
                                                IFCP7  1
                   0110000007                   RL     LBLOCK3
L    22  0100000014 •                END2BLK    RJ     ADD
L     4                                         LOC    ADD
L        00000000000000000000        MULT       DATA   0
```

```
L     15  5110000000 •
                    5120000001 •                 SA1     OPERAND•0
                                                 SA2     OPERAND•1
L     16  5130000002 •                           SA3     OPERAND•2
                    5140000003 •                 SA4     OPERAND•3
L     17  42612                                  IX6     X1•X2
               42734                             IX7     X3•X4
                    5160000006 C                 SA6     ANSWER•6
L     20  5170000007 C                           SA7     ANSWER•7
                    5100000014 •                 SA0     ADD
L     21  7100000030                             SX0     LBLOCK1•LBLOCK2•LBLCCK3
                                                 IFCP6   2
                                                 IFCP7   1
                    0110000012                   RL      LHLOCK4
L     22  0100000014 •                  END3BLK  RJ      ADD
L     14                                         LOC     ADD
L     14  00000000000000000000          DIVIDE   DATA    0
L     15  5110000000 •                           SA1     OPERAND•0
                    5120000001 •                 SA2     OPERAND•1
L     16  5130000002 •                           SA3     OPERAND•2
                    5140000003 •                 SA4     OPERAND•3
L     17  27101                                  PX1     X1
               27202                             PX2     X2
                    24101                        NX1     X1
                         24202                   NX2     X2
L     20  44621                                  FX6     X2/X1
               27303                             PX3     X3
                    27404                        PX4     X4
                         24303                   NX3     X3
L     21  24404                                  NX4     X4
               44743                             FX7     X4/X3
                    5160000010 C                 SA6     ANSWER•8
L     22  5170000011 C                           SA7     ANSWER•9
                    26666                        UX6     X6•B5
                         26777                   UX7     X7•B7
L     23  22666                                  LX6     X6•B6
               22777                             LX7     X7•B7
                    5160000012 C                 SA6     ANSWER•10
L     24  5170000013 C                           SA7     ANSWER•11
L     25  0400000010 •                  END4BLK  EQ      START
                                12      LBLOCK1  EQU     END1BLK-ADD•1
                                 7      LBLOCK2  EQU     END2BLK-SUBT•1
                                 7      LBLOCK3  EQU     END3BLK-MULT•1
                                12      LBLOCK4  EQU     END4BLK-DIVIDE•1
                                42      LBLOCKT  EQU     LBLOCK1•LBLOCK2•LBLCCK3•LBLOCK4
                                12      MAXBLK   MAX     LBLOCK1,LBLOCK2,LBLCCK3,LBLOCK4
                                 7      MINBLK   MIN     LBLOCK1,LBLOCK2,LBLCCK3,LBLOCK4
      56                                                 END

                          47200B CM  STORAGE USED             114 STATEMENTS      20 SYMBOLS
                                     MODEL 74 ASSEMBLY      0.343 SECONDS         94 REFERENCES
```

```
LOC7600                                                                      COMPASS 3.4-444.    02/25/77  16.10.08.      PAGE     4
SYMBOLIC REFERENCE TABLE.

ADD          14      PROGRAM.        2/10       2/17 L     2/39       2/50       2/56       3/14       3/40
                                     2/16       2/33       2/40       2/55       3/09       3/15
ANSWER        0      ANSWER          2/06 L     2/24 S     2/32 S     2/49 S     3/08 S     3/32 S     3/38 S
                                     2/23 S     2/31 S     2/48 S     3/07 S     3/31 S     3/37 S
DIVIDE       14      PROGRAM.        3/16 L     3/43
END1BLK      25      PROGRAM.        2/39 L     3/40
END2BLK      22      PROGRAM.        2/55 L     3/41
END3BLK      22      PROGRAM.        3/14 L     3/42
END4BLK      25      PROGRAM.        3/39 L     3/43
ERRMSG1       0      EXTERNAL.       2/04 X
ERRMSG2       0      EXTERNAL.       2/04 X
LBLOCKT      42                      2/14       3/44 D
LBLOCK1      12                      2/15       2/34       2/51       3/10       3/40 D     3/44       3/45       3/46
LBLOCK2       7                      2/38       2/51       3/10       3/41 D     3/44       3/45       3/46
LBLOCK3       7                      2/54       3/10       3/42 D     3/44       3/45
LBLOCK4      12                      3/13       3/43 D     3/44       3/45
MAXBLK       12                      3/45 D
MINBLK        7                      3/46 D
MULT         14      PROGRAM.        2/57 L     3/42
OPERAND       0      PROGRAM.        2/08 L     2/19       2/26       2/42       2/45       3/03       3/18
                                     2/17       2/20       2/27       2/43       3/01       3/04       3/19
                                     2/18       2/25       2/28       2/44       3/02       3/17       3/20
START        10      PROGRAM.        2/03 E     2/09 L     3/39
SUBT         14      PROGRAM.        2/41 L     3/41
```

```
  MFS   NOS/BE 1.2          R443A/R3B  02/02/77
16.09.39.DON00MY  FROM    /LE
16.09.39.IP  00000832 WORDS - FILE INPUT  , DC 04
16.09.39.DON.          001A,0904,71GT100A,MILLER
16.09.59.FTN,R=0.
16.10.08.      .639 CP SECONDS COMPILATION TIME
16.10.08.MAP,PART.
16.10.09.EXIT.
16.10.09.OP  00001408 WORDS - FILE OUTPUT , DC 40
16.10.09.MS    3584  WORDS  (   14336 MAX USED)
16.10.09.CPA        .636 SEC.        .636 ADJ.
16.10.09.CPB        .146 SEC.        .146 ADJ.
16.10.09.IO        2.124 SEC.       2.124 ADJ.
16.10.10.CM       55.574 KWS.       3.391 ADJ.
16.10.10.SS                         6.299
16.10.10.PP       14.273 SEC.    DATE 02/25/77
16.10.10.EJ  END OF JOB, LE
```

```
   PROGRAM INITIAL     74/74    CPT=1                          FTN 4.6+444        02/22/77  06.31.07        PAGE      1

1                 PROGRAM INITIAL(OUTPUT)
                  COMMON /ANSWER/ BUFFER(16)
                  CALL START
                  PRINT 100,(BUFFER(I),I=1,16)
5             100 FORMAT(4O22)
                  END
```

```
  SUBROUTINE ERRMSG1     74/74    CPT=1                        FTN 4.6+444        02/22/77  06.31.07        PAGE      1

1                 SUBROUTINE ERRMSG1
                  PRINT 100
              100 FORMAT(1H1,* ECS WRITE ERROR*)
                  RETURN
5                 END
```

```
  SUBROUTINE ERRMSG2     74/74    CPT=1                        FTN 4.6+444        02/22/77  06.31.07        PAGE      1

1                 SUBROUTINE ERRMSG2
                  PRINT 100
              100 FORMAT(1H1,* ECS READ ERROR*)
                  RETURN
5                 END
```

```
LOC7600                                                      COMPASS 3.4-444.     02/22/77  06.31.12.       PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH            BINARY CONTROL CARDS.

                0       61            IDENT  LOC7600
               61                     END

                           BLOCKS     TYPE       ADDRESS    LENGTH

                           PROGRAM*   LOCAL            0        61
                           ANSWER     COMMON           0        20

                           ENTRY POINTS.

                           START           10+

                           EXTERNAL SYMBOLS.

                           ERRMSG1    ERRMSG2
```

```
                                           IDENT   LOC7600
                                           MACHINE 6
                                           ENTRY   START
                                           EXT     ERRMSG1,ERRMSG2
                                           USE     /ANSWER/
       0                    20    ANSWER   BSSZ    16
                                           USE     0
       0  00000000000000000001    OPERAND  DATA    1,2,3,4,5,6,7,8
      10  00000000000000000000    START    DATA    0
      11  5100000015 +                     SA0     ADD
                    7100000000             SX0     0
                                           IFCP6   4
      12  0120000044                       WE      LBLOCKT
                    0100000000 X           RJ      ERRMSG1
      13  0110000012                       RE      LBLOCK1
                    0100000000 X           RJ      ERRMSG2
                                           IFCP7   2
      14  0400000015 +                     EQ      ADD
      15  5110000000 +            ADD      SA1     OPERAND+0
                    5120000001 +           SA2     OPERAND+1
      16  5130000002 +                     SA3     OPERAND+2
                    5140000003 +           SA4     OPERANC+3
      17  36612                            IX6     X1+X2
               36734                       IX7     X3+X4
                    5160000000 C           SA6     ANSWER+0
      20  5170000001 C                     SA7     ANSWER+1
                    5110000004 +           SA1     OPERAND+4
      21  5120000005 +                     SA2     OPERAND+5
                    5130000006 +           SA3     OPERAND+6
      22  5140000007 +                     SA4     OPERAND+7
                    36612                  IX6     X1+X2
                         36734             IX7     X3+X4
      23  5160000002 C                     SA6     ANSWER+2
                    5170000003 C           SA7     ANSWER+3
      24  5100000015 +                     SA0     ADD
                    7100000012             SX0     LBLOCK1
                                           IFCP6
      25  0110000010                       RE      LBLOCK2
                    0100000000 X           RJ      ERRMSG2
                                           ENDIF
                                           IFCP7   1
      26  0100000015 +            ENDIBLK  RJ      ADD
L     15                                   LOC     ADD
L     15  00000000000000000000    SUBT     DATA    0
L     16  5110000000 +                     SA1     OPERAND+0
                    5120000001 +           SA2     OPERAND+1
L     17  5130000002 +                     SA3     OPERAND+2
                    5140000003 +           SA4     OPERAND+3
L     20  37621                            IX6     X2-X1
               37734                       IX7     X3-X4
                    5160000004 C           SA6     ANSWER+4
L     21  5170000005 C                     SA7     ANSWER+5
                    5100000015 +           SA0     ADD
L     22  7100000022                       SX0     LBLOCK1+LBLOCK2
                                           IFCP6   2
L     23  0110000010                       RE      LBLOCK3
                    0100000000 X           RJ      ERRMSG2
```

```
 L      24  0100000015 +                                 IFCP7   1
 L      15                                    END2BLK    RJ      ADD
 L      15  00000000000000000000                         LOC     ADD
 L      16  5110000000 +                      MULT       DATA    0
                                                         SA1     OPERAND+0
                       5120000001 +                      SA2     OPERAND+1
 L      17  5130000002 +                                 SA3     OPERAND+2
                       5140000003 +                      SA4     OPERAND+3
 L      20  42612                                        IX6     X1*X2
                 42734                                   IX7     X3*X4
                       5160000006 C                      SA6     ANSWER+6
 L      21  5170000007 C                                 SA7     ANSWER+7
                       5100000015 +                      SA0     ADD
 L      22  7100000032                                   SX0     LBLOCK1+LBLOCK2+LBLCCK3
 L      23  0110000012                                   IFCP6   2
                                                         RE      LBLOCK4
                       0100000000 X                      RJ      ERRMSG2
 L      24  0100000015 +                                 IFCP7   1
 L      15                                    END3BLK    RJ      ADD
 L      15  00000000000000000000                         LOC     ADD
 L      16  5110000000 +                      DIVIDE     DATA    0
                                                         SA1     OPERAND+0
                       5120000001 +                      SA2     OPERAND+1
 L      17  5130000002 +                                 SA3     OPERAND+2
                       5140000003 +                      SA4     OPERAND+3
 L      20  27101                                        PX1     X1
                 27202                                   PX2     X2
                       24101                             NX1     X1
                             24202                       NX2     X2
 L      21  44621                                        FX6     X2/X1
                 27303                                   PX3     X3
                       27404                             PX4     X4
                             24303                       NX3     X3
 L      22  24404                                        NX4     X4
                 44743                                   FX7     X4/X3
                       5160000010 C                      SA6     ANSWER+8
 L      23  5170000011 C                                 SA7     ANSWER+9
                       26666                             UX6     X6+B6
                             26777                       UX7     X7+B7
 L      24  22666                                        LX6     X6+B6
                 22777                                   LX7     X7+B7
                       5160000012 C                      SA6     ANSWER+10
 L      25  5170000013 C                                 SA7     ANSWER+11
 L      26  0400000010 +                      END4BLK    EQ      START
                                    12        LBLOCK1    EQU     END1BLK-ADD+1
                                    10        LBLOCK2    EQU     END2BLK-SUBT+1
                                    10        LBLOCK3    EQU     END3BLK-MULT+1
                                    12        LBLOCK4    EQU     END4BLK-DIVIDE+1
                                    44        LBLOCKT    EQU     LBLOCK1+LBLOCK2+LBLCCK3+LBLOCK4
                                    12        MAXBLK     MAX     LBLOCK1,LBLOCK2,LBLCCK3,LBLOCK4
                                    10        MINBLK     MIN     LBLOCK1,LBLOCK2,LBLCCK3,LBLOCK4
      61                                                 END

                         47200B CM  STORAGE USED             114 STATEMENTS        20 SYMBOLS
                                    MODEL 74 ASSEMBLY      0.357 SECONDS           99 REFERENCES
```

```
LOC7600                                                              COMPASS 3.4-444.     02/22/77  06.31.12.       PAGE      4
SYMBOLIC REFERENCE TABLE.

ADD            15      PROGRAM.       2/10       2/19 L    2/42      2/53      3/03      3/19      3/45
                                      2/18       2/35      2/43      3/02      3/13      3/20
ANSWER          0      ANSWER         2/06 L     2/26 S    2/34 S    2/52 S    3/12 S    3/37 S    3/43 S
                                      2/25 S     2/33 S    2/51 S    3/11 S    3/36 S    3/42 S
DIVIDE         15      PROGRAM.       3/21 L     3/48
END1BLK        26      PROGRAM.       2/42 L     3/45
END2BLK        24      PROGRAM.       3/02 L     3/46
END3BLK        24      PROGRAM.       3/19 L     3/47
END4BLK        26      PROGRAM.       3/44 L     3/48
ERRMSG1         0      EXTERNAL.      2/04 X     2/14
ERRMSG2         0      EXTERNAL.      2/04 X     2/16      2/39      2/57      3/17
LBLOCKT        44                     2/13       3/49 D
LBLOCK1        12                     2/15       2/36      2/54      3/14      3/45 D    3/49      3/50      3/51
LBLOCK2        10                     2/38       2/54      3/14      3/46 D    3/49      3/50      3/51
LBLOCK3        10                     2/56       3/14      3/47 D    3/49      3/50
LBLOCK4        12                     3/16       3/48 D    3/49      3/50
MAXBLK         12                     3/50 D
MINBLK         10                     3/51 D
MULT           15      PROGRAM.       3/04 L     3/47
OPERAND         0      PROGRAM.       2/08 L     2/21      2/28      2/45      2/48      3/07      3/23
                                      2/19       2/22      2/29      2/46      3/05      3/08      3/24.
                                      2/20       2/27      2/30      2/47      3/06      3/22      3/25
START          10      PROGRAM.       2/03 E     2/09 L    3/44
SUBT           15      PROGRAM.       2/44 L     3/46
```

```
 MFS   NCS/BE 1.2          R443A/R38  02/02/77
06.31.06.CON003E  FROM    /LE
06.31.06.IP  00000832 WORDS - FILE INPUT  , DC 04
06.31.06.CON.            U01A,0904,71GT100A,MILLER
06.31.07.FTN,R=0.
06.31.12.       .647 CP SECONDS COMPILATION TIME
06.31.12.MAP,PART.
06.31.12.EXIT.
06.31.12.CP  00001408 WORDS - FILE OUTPUT , DC 40
06.31.12.MS      3584  WORDS  (   14336 MAX USED)
06.31.12.CPA         .717 SEC.           .717 ADJ.
06.31.12.CPB         .001 SEC.           .001 ADJ.
06.31.12.IO         1.970 SEC.          1.970 ADJ.
06.31.12.CM        54.249 KWS.          3.311 ADJ.
06.31.12.SS                             6.000
06.31.12.PP         7.119 SEC.      DATE 02/22/77
06.31.12.EJ  END OF JOB, LE
```

LESSON 11

SUBROUTINES

LESSON PREVIEW:

THIS SECTION COVERS SUBROUTINE STRUCTURE AND THE VARIOUS WAYS OF PASSING INFORMATION OR PARAMETERS BETWEEN MAIN PROGRAM AND THE SUBROUTINE, INCLUDING BLANK AND LABELLED COMMON.

REFERENCES:

CHAPTER 3	COMPASS REFERENCE MANUAL #60492600

TRAINING AIDS:

VISUAL SET V11
PROGRAM LISTINGS FROM EXAMPLE DECKS 9A, 9B

PROJECT:

PROGRAMMING PROJECT 5

OBJECTIVES:

AT THE COMPLETION OF THIS LESSON THE STUDENT WILL BE ABLE TO:

1. WRITE A MAIN PROGRAM THAT CALLS A SUBROUTINE USING RETURN JUMP INSTRUCTION.
2. CORRECTLY SET UP COMMON BLOCK FOR USE BY THE VARIOUS ROUTINES.
3. CALL A COMPASS ROUTINE FROM FORTRAN EXTENDED.
4. CALL A FORTRAN EXTENDED ROUTINE FROM COMPASS.

PARAMETERS

3 WAYS TO PASS PARAMETERS:

IN REGISTERS

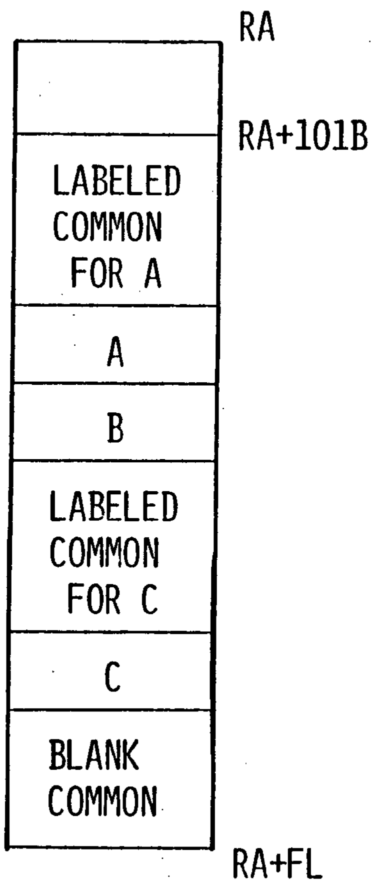
IN COMMON BLOCKS

BY PARAMETER PICKUP METHOD

(THIS METHOD IS NORMALLY USED FOR LIBRARY SUBROUTINES)

COMMON

COMMON	BLANK COMMON	LABELED COMMON
HISTORY	ORIGINALLY BLANK COMMON WAS DESIGNED AS <u>THE</u> ONE AREA FOR COMMUNICATION BETWEEN SUBPROGRAMS....	AS MANY SUBROUTINES WERE DEVELOPED BY DIFFERENT PROGRAMMERS. IT BECAME NECESSARY TO SET UP SEVERAL DIFFERENT BLOCKS FOR COMMUNICATION BETWEEN THE DIFFERENT SUBPROGRAMS. THESE BLOCKS WERE GIVEN <u>LABELS</u> TO IDENTIFY WHICH BLOCK WAS TO BE USED BY WHICH SUBROUTINE. *
LOAD LOCATION	BLANK COMMON IS TYPICALLY LOCATED AT THE END OF THE LOAD AND HENCE	LABELED COMMON BLOCKS COME AT THE BEGINNING OF THE LOAD AND THUS
PRESETTING	NO PRESETTING, I.E., ALL INFORMATION MUST BE PUT INTO BLANK COMMON AT <u>EXECUTION TIME</u>.	CAN BE PRESET, I.E., THE PROGRAMMER CAN STORE INSTRUCTIONS AND DATA IN LABELED COMMON BLOCKS AND THESE AREAS WILL BE SET UP AT <u>LOAD TIME</u>.



```
                                                   COMPASS 3.5-470.    07/14/78  12.43.36.      PAGE      1
LINKAGE
STORAGE ALLOCATION.

         ADDRESS   LENGTH              BINARY CONTROL CARDS.

               0       77              IDENT   LINKAGE
              77                       END     START


                                 BLOCKS     TYPE       ADDRESS    LENGTH

                                 PROGRAM*  LOCAL             0        36
                                 LITERALS* LOCAL            36         3
                                 DATA      LOCAL            41        36
                                 DATA     +COMMON            0        24
                                 //       +COMMON            0        12


                                 ENTRY POINTS.

                                 START             0+            BUF            24+           ERRMSG1        15+


                                 EXTERNAL SYMBOLS.

                                 SUB1      MSG=        SYS=
                                                  IDENT   LINKAGE
                                                  ENTRY   START,BUF,ERRMSG1
                                                  LIST    $
                                                  COMMENT THIS EXAMPLE SHOWS THE LINKAGE BETWEEN PROGRAM BLOCKS
                                                  EXT     SUB1
     0  46000                           START     NO
     1  7160000010                      +         SX6     10B          DATA
                 5160000024 +                     SA6     BUF          STORE IT
                         36 +   LTAG              LIT     2,4,6        LITERAL
     2  5110000037 +                              SA1     LTAG+1       GET 2ND LITERAL
                 10711                            BX7     X1           MOVE TO X7
     3  5170000025 +                              SA7     BUF+1        STORE LITERAL
                 5100000053 +                     SA0     DATA03       STARTING S.C.M. ADDRESS
     4  7100000000 C                              SX0     DATA10       L.C.M. STARTING ADDRESS
     5  0120000012                                WE      10           MOVE DATA FROM S.C.M. LOCAL TO L.C.M. LABELLED COMM
                 0100000015 +                     RJ        ERRMSG1
     6  0100000000 X                    +         RJ      SUB1         GO TO SUB1
     7  5100000041 +                              SA0     DATA02       S.C.M. STARTING ADDRESS
                 7100000012 C                     SX0     DATA01       L.C.M. STARTING ADDRESS
    10  0120000012                                WE      10           MOVE DATA FROM S.C.M. LOCAL BLOCK TO L.C.M. LOCAL
                 0100000015 +                     RJ        ERRMSG1
    11  5100000065 +                    +         SA0     DATA04       S.C.M. STARTING ADDRESS
                 7100000000 C                     SX0     DATA6        L.C.M. STARTING ADDRESS
    12  0120000012                                WE      10           MOVE DATA FROM S.C.M. LOCAL TO L.C.M. BLANK COMMON
                 0100000015 +                     RJ        ERRMSG1
    13                                  +         ENDRUN
    13  7160247021                      SX6 3REND*4+1                                                   ENDRUN  .1
                 20650                  LX6 40D                                                         ENDRUN  .1
    14  0100000000 X                    RJ =XSYS=                                                       ENDRUN  .1
                                        ENDM                                                            ENDRUN  .1
    15  00000000000000000000          ERRMSG1  DATA    0            GIVE ERROR MESSAGE
    16                                            MESSAGE MESSAGE,,RECALL
    16  7110000021 +                    SX1 MESSAGE                                                     MESSAGE .1
                 43652                  MX6 42D                                                         MESSAGE .1
                      15116             BX1 -X6*X1                                                      MESSAGE .1
                          0           ++SCP≡RC SET 0                                                    MESSAGE .1
                                        IFC NE, RECALL  ,1                                              MESSAGE .1
                     200000           ++SCP≡RC SET 200000B                                              MESSAGE .1
                                        IF -REG,,7                                                      MESSAGE .1
                          1           ++SCP≡OP SET 1                                                    MESSAGE .1
                                         IF DEF,,1                                                      MESSAGE .1
                          0           ++SCP≡OP SET                                                      MESSAGE .1
                                        IFC EQ,$$LOCAL$,1                                               MESSAGE .1
                                      ++SCP≡OP SET 3                                                    MESSAGE .1
                                        R= X6,++SCP≡RC+++SCP≡OP                                         MESSAGE .1
    17  7160200000                      SX6    ++SCP≡RC+++SCP≡OP                                        MESSAGE .1
                                        SKIP 7                                                          MESSAGE .1
                                        IFC EQ,$$X1$,1                                                  MESSAGE .1
                                        SKIP 1                                                          MESSAGE .1
                                        IFC EQ,$$X6$,1                                                  MESSAGE .1
                                      4 ERR OPTION IN X1 OR X6 IGNORED                                  MESSAGE .1
                                        R= X6,++SCP≡RC                                                  MESSAGE .1
                                        SKIP 1                                                          MESSAGE .1
                                        R= X6,++SCP≡RC+                                                 MESSAGE .1
                 0100000000 X           RJ =XMSG=                                                       MESSAGE .1
                                        ENDM                                                            MESSAGE .1
```

```
                                       USE     DATA
 41  00000000000000000001      DATA02  DATA 1,1.0,2,2.0,3,3.0,4,4.0,5,5.0
 42  17204000000000000000
 43  00000000000000000002
 44  17214000000000000000
 45  00000000000000000003
 46  17216000000000000000
 47  00000000000000000004
 50  17224000000000000000
 51  00000000000000000005
 52  17225000000000000000
 53  00000000000000000144      DATA03  CON     100,200,300,400,500,600,700,800,900,1000
 54  00000000000000000310
 55  00000000000000000454
 56  00000000000000000620
 57  00000000000000000764
 60  00000000000000001130
 61  00000000000000001274
 62  00000000000000001440
 63  00000000000000001604
 64  00000000000000001750
 65  00000000000000000001      DATA04  CON     1RA,1RB,1RC,1RD,1RE,1RF,1RG,1RH,1RI,1RJ
 66  00000000000000000002
 67  00000000000000000003
 70  00000000000000000004
 71  00000000000000000005
 72  00000000000000000006
 73  00000000000000000007
 74  00000000000000000010
 75  00000000000000000011
 76  00000000000000000012
                                       USELCM /DATA/
  0                    12      DATA10  BSS     10
 12                    12      DATA01  BSS     10
                                       USELCM //
  0                    12      DATA6   BSS     10
                                       USE     0
 21  55055703572357552722      MESSAGE DIS     ,/ E.C.S. WRITE ERROR OCCURRED/
 22  11240555052222172255
 23  17030325222205040000
 24                    12      BUF     BSSZ    10             BUFFER FOR PROGRAM
                              DEFAULT SYMBOLS DEFINED BY COMPASS.

  0 X                         SYS=
  0 X                         MSG=

                              CONTENT OF LITERALS BLOCK.

 36  00000000000000000002             A
 37  00000000000000000004             D
 40  00000000000000000006             F

 77                                    END     START
```

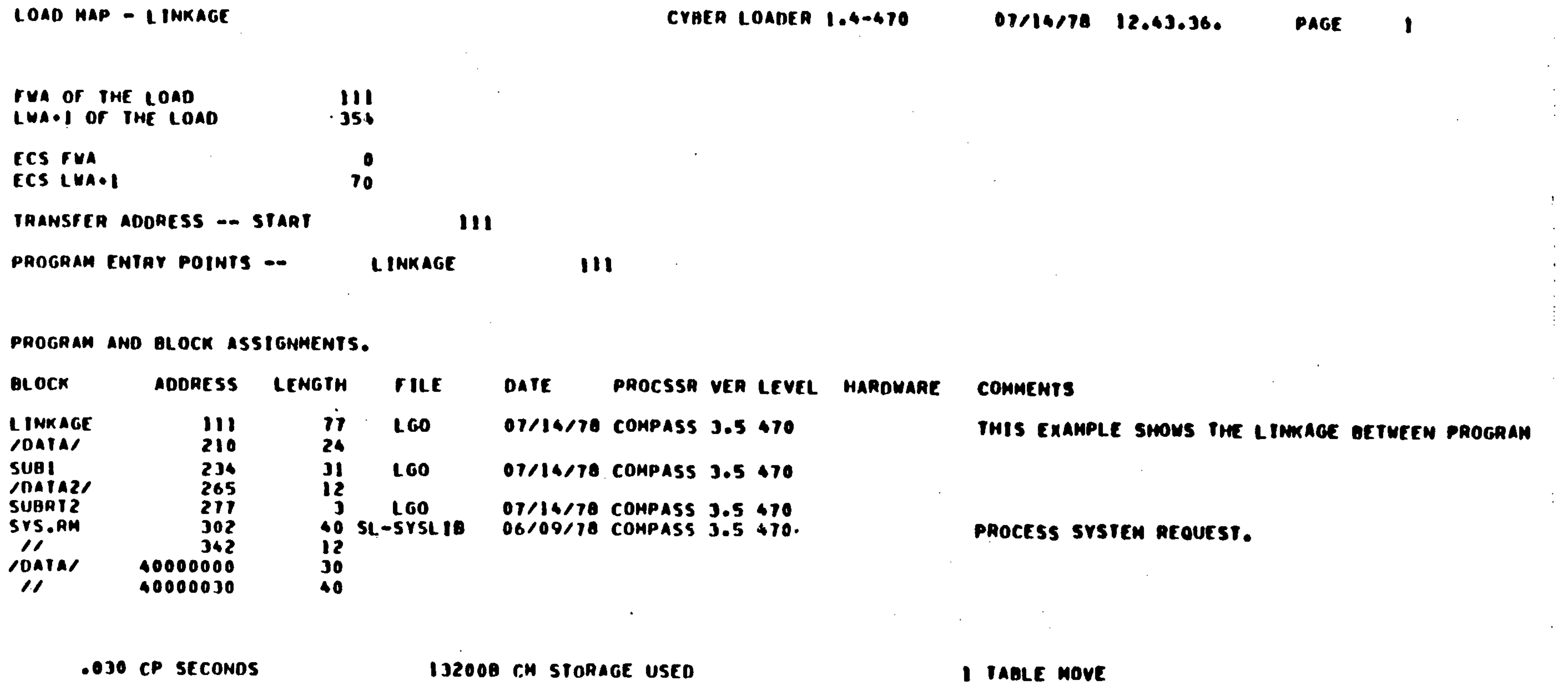
```
LINKAGE                                                                    COMPASS 3.5-470.     07/14/78  12.43.36.      PAGE      4
SYMBOLIC REFERENCE TABLE.

BUF             24      PROGRAM*          2/02 E     2/08 S     2/12 S     3/41 L
DATA01          12      DATA              2/19       3/34 L
DATA02          41      PROGRAM*          2/18       3/02 L
DATA03          53      PROGRAM*          2/13       3/12 L
DATA04          65      PROGRAM*          2/22       3/22 L
DATA10           0      DATA              2/14       3/31 L
DATA6            0      //                2/23       3/36 L
ERRMSG1         15      PROGRAM*          2/02 E     2/16       2/21       2/25       2/31 L     2/57
LTAG            16      PROGRAM*          2/09 L     2/10
MESSAGE         21      PROGRAM*          2/33       3/38 L
MSG=             0      EXTERNAL*         2/55
START            0      PROGRAM*          2/02 E     2/06 L
SUB1             0      EXTERNAL*         2/05 X     2/17
SYS=             0      EXTERNAL*         2/29
```

```
SUB1                                                                 COMPASS 3.5-470.     07/14/78  12.43.36.      PAGE      1
STORAGE ALLOCATION.

                ADDRESS   LENGTH              BINARY CONTROL CARDS.

                      0        31             IDENT  SUB1
                     31                       END

                                    BLOCKS      TYPE       ADDRESS    LENGTH

                                    PROGRAM*   LOCAL             0        31
                                    //         COMMON            0         5
                                    DATA       COMMON            0        24
                                    DATA      *COMMON            0        12
                                    //        *COMMON            0        36

                                    ENTRY POINTS.

                                    SUB1              0*

                                    EXTERNAL SYMBOLS.

                                    BUF        MSG*       SUB2       ERRMSG1
```

```
SUB1                                                                 COMPASS 3.5-470.     07/14/78  12.43.36.      PAGE      2

                                                  IDENT   SUB1
                                                  EXT     BUF
                                                  ENTRY   SUB1
  0  00000000000000000000         SUB1      DATA    0          RETURN ADDRESS STORED HERE.
  1  7160000006                             SX6     6          DATA
                5160000002 X                SA6     BUF+2      STORE IT
  2  5100000017 +                           SA0     BUFFER     S.C.M. STARTING ADDRESS
                7100000000 C                SX0     DATA9      L.C.M. STARTING ADDRESS
  3  0110000012                             RE      10         TRANSFER 10 WORDS FROM LCM TO SCM
                0100000010 +                RJ      ERRMSG2
  4  7100000012 C               +           SX0     DATA8      L.C.M. STARTING ADDRESS
  5  0120000012                             WE      10         TRANSFER 10 WORDS FROM S.C.M. TO L.C.M. BLANK COMMN
                0100000000 X                RJ      =XERRMSG1  GIVE ERROR MESSAGE
  6  0100000000 X               +           RJ      =XSUB2     GO TO SUB2
  7  0400000000 +                           EQ      SUB1       EXIT FROM THIS SUBROUTINE
 10  00000000000000000000       ERRMSG2     DATA    0
 11  7110000014 +                           MESSAGE READMSG,,RECALL  GIVE ERROR MESSAGE
 13  0400000010 +                           EQ      ERRMSG2    RETURN TO CALLER
 14  55055703572357552205       READMSG     DIS     ,/ E.C.S. READ ERROR OCCURRED/
 17                    12       BUFFER      BSS     10
                                            USE     //         USE BLANK COMMON
  0                     5       BF4         BSS     5
                                            USE     /DATA/     USE LABELED COMMON
  0                     5        DATA       BSS     5
  5                    17       BUFF2       BSS     15
                                            USELCM /DATA/
  0                    12        DATA9      BSS     10
                                            USELCM //
  0                    12        DATA7      BSS     10
 12                    24        DATA8      BSS     20
 31                                         END

      1100B LCM  47600B SCM STORAGE USED            49 STATEMENTS         16 SYMBOLS
                            MODEL 174 ASSEMBLY      0.061 SECONDS         21 REFERENCES
```

```
SUBRT2                                                     COMPASS 3.5-470.     07/14/78  12.43.36.      PAGE     1
STORAGE ALLOCATION.

          ADDRESS   LENGTH              BINARY CONTROL CARDS.

                0        3              IDENT  SUBRT2
                3                       END

                              BLOCKS     TYPE       ADDRESS    LENGTH

                              PROGRAM*   LOCAL            0         3
                              //         COMMON           0        12
                              DATA       COMMON           0        24
                              DATA2      COMMON           0        12

                              ENTRY POINTS.

                              SUB2            0*

                              EXTERNAL SYMBOLS.

                              BUF
```

```
SUBRT2                                                     COMPASS 3.5-470.     07/14/78  12.43.36.      PAGE     2

                                                 IDENT  SUBRT2
                                                 ENTRY  SUB2
                                                 USE    //          USE BLANK COMMON
   0                      12      BUFF4          BSS    10
                                                 USE    0           GO BACK TO ZERO BLOCK
   0  00000000000000000000        SUB2           DATA   0
   1  7160000004                                 SX6    4           DATA
                 5160000003 X                    SA6    =XBUF+3     STORE DATA
   2  0400000000 +                               EQ     SUB2        EXIT FROM THIS SUBROUTINE
                                                 USE    /DATA/      USE LABELED COMMON
   0                       5      DUMMY          BSS    5           DUMMY OVER
   5                       5      BUFF2          BSS    5           USED IN SUBRT2
  12                      12      BUFF5          BSS    10
                                                 USE    /DATA2/     USE LABELED COMMON
   0                      12      BUFF3          BSS    10
   3                                             END

      1100B LCM  46500B SCM STORAGE USED              16 STATEMENTS          7 SYMBOLS
                          MODEL 174 ASSEMBLY      0.017 SECONDS              9 REFERENCES
```



```
LOAD MAP - LINKAGE                                          CYBER LOADER 1.4-470       07/14/78  12.43.36.      PAGE      1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD          354

ECS FWA                      0
ECS LWA+1                   70

TRANSFER ADDRESS -- START            111

PROGRAM ENTRY POINTS --       LINKAGE          111



PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS   LENGTH     FILE      DATE     PROCSSR VER LEVEL  HARDWARE   COMMENTS

LINKAGE          111       77     LGO      07/14/78 COMPASS 3.5 470                THIS EXAMPLE SHOWS THE LINKAGE BETWEEN PROGRAM
/DATA/           210       24
SUB1             234       31     LGO      07/14/78 COMPASS 3.5 470
/DATA2/          265       12
SUBRT2           277        3     LGO      07/14/78 COMPASS 3.5 470
SYS.RM           302       40 SL-SYSLIB    06/09/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.
 //              342       12
/DATA/      40000000       30
 //         40000030       40


     .030 CP SECONDS                 13200B CM STORAGE USED                      1 TABLE MOVE
```

```
DUMP     RELATIVE                        DMP,111,337.

  00111   46000 46000 61000 46000        71600 00010 51600 00135        51100 00150 10711 46000
  00114   51700 00136 51000 00164        71000 00000 61000 46000        01200 00012 01000 00126        01000 00234 61000 46000

  00120   51000 00152 71000 00012        01200 00012 01000 00126        51000 00176 71000 00030        01200 00012 01000 00126

  00124   71602 47021 20650 46000        01000 00304 61000 46000        00000 00000 00000 00000        71100 00132 43652 15116

  00130   71602 00000 01000 00333        04000 00126 61000 46000        55055 70357 23575 52722        11240 55505 22221 72255

  00134   17030 32522 22050 40000        00000 00000 00000 00010        00000 00000 00000 00004        00000 00000 00000 00006

  00140   00000 00000 00000 00004        00000 00000 00000 00000  00147>00000 00000 00000 00002
  00150   00000 00000 00000 00004        00000 00000 00000 00006        00000 00000 00000 00001        17204 00000 00000 00000

  00154   00000 00000 00000 00002        17214 00000 00000 00000        00000 00000 00000 00003        17216 00000 00000 00000

  00160   00000 00000 00000 00004        17224 00000 00000 00000        00000 00000 00000 00005        17225 00000 00000 00000

  00164   00000 00000 00000 00144        00000 00000 00000 00310        00000 00000 00000 00454        00000 00000 00000 00620

  00170   00000 00000 00000 00764        00000 00000 00000 01130        00000 00000 00000 01274        00000 00000 00000 01440

  00174   00000 00000 00000 01604        00000 00000 00000 01750        00000 00000 00000 00001        00000 00000 00000 00002

  00200   00000 00000 00000 00003        00000 00000 00000 00004        00000 00000 00000 00005        00000 00000 00000 00006

  00204   00000 00000 00000 00007        00000 00000 00000 00010        00000 00000 00000 00011        00000 00000 00000 00012

  00210   60000 00000 04004 00210  00234>04000 00120 00000 00000        71600 00006 51600 00137        51000 00253 71000 00000

  00237   01100 00012 01000 00244
  00240   71000 00042 61000 46000        01200 00012 01000 00126        01000 00277 61000 46000        04000 00234 61000 46000

  00244   00000 00000 00000 00000        71100 00250 43652 15116        71602 00000 01000 00333        04000 00244 61000 46000

  00250   55055 70357 23575 52205        01045 50522 22172 25517        03032 52222 05040 00000        00000 00000 00000 00144

  00254   00000 00000 00000 00310        00000 00000 00000 00454        00000 00000 00000 00620        00000 00000 00000 00764

  00260   00000 00000 00000 01130        00000 00000 00000 01274        00000 00000 00000 01440        00000 00000 00000 01604

  00264   00000 00000 00000 01750        60000 00000 04004 00265  00277>04000 00243 00000 00000
  00300   71600 00004 51600 00140        04000 00277 61000 46000        04000 00315 00000 00000        01300 00000 00000 00000

  00304   04000 00126 00000 00000        51100 00001 03110 00305        54610 04000 00303 46000        51100 00066 03310 00311

  00310   51100 00302 04000 00312        71100 00130 20160 46000        13661 13161 13661 46000        51600 00303 10611 46000

  00314   51100 00001 01000 00302        20652 01000 00304 46000        51100 00001 03110 00316        04004 00317 61000 46000

  00320   51100 00001 03110 00317        71602 20314 04000 00315        20150 36661 01000 00304        04004 00323 61000 46000

  00324   71602 20314 20652 36662        53160 20173 03310 00323        03010 00323 51100 00001        03110 00325 71100 00001

  00330   04000 00322 61000 46000        71603 24616 12661 20651        01000 00304 61000 46000        04004 00333 61000 46000

  00334   73660 20630 12161 73610        20173 03310 00333 20151        [illegible]                    [illegible]
```

```
ECS DUMP ARGUMENTS ARE   FWA=0000000B  LWA=0000100B  TYPE=00B     ECS FL=0001000B  CM FL=016500B                    PAGE NO.       1
0000000 00000000000000000144 00000000000000000310 00000000000000000454 00000000000000000620               A9           CH           D=           FP
0000004 00000000000000000764 00000000000000001130 00000000000000001274 00000000000000001440               G≠           1X           J≤           L5
0000010 00000000000000001604 00000000000000001750 00000000000000000001 17204000000000000000               ND           O/           1A OP51
0000014 00000000000000000002 17214000000000000000 00000000000000000003 17216000000000000000               1B OQ51                   1C OQ≡1
0000020 00000000000000000004 17224000000000000000 00000000000000000005 17225000000000000000               1D OR51                   1E OR/1
0000024 60000000000400000024 60800000000400000025 60000000000400000026 60000000000400000027 ≡1   1D   1T ≡1   1D   1U ≡1   1D   1V ≡1   1D   1W
0000030 00000000000000000001 00000000000000000002 00000000000000000003 00000000000000000004               1A           1B           1C           1D
0000034 00000000000000000005 00000000000000000006 00000000000000000007 00000000000000000010               1E           1F           1G           1H
0000040 00000000000000000011 00000000000000000012 00000000000000000144 00000000000000000310               11           1J           A9           CH
0000044 00000000000000000454 00000000000000000620 00000000000000000764 00000000000000001130               D=           FP           G≠           1X
0000050 00000000000000001274 00000000000000001440 00000000000000001604 00000000000000001750               J≤           L5           ND           O/
0000054 60000000000400000054 60000000000400000055 60000000000400000056 60000000000400000057 ≡1   1D   1= ≡1   1D   1  ≡1   1D   1, ≡1   1D   1.
0000060 60000000000400000060 60000000000400000061 60000000000400000062 60000000000400000063 ≡1   1D   1≡ ≡1   1D   1[ ≡1   1D   1] ≡1   1D   1%
0000064 60000000000400000064 60000000000400000065 60000000000400000066 60000000000400000067 ≡1   1D   1≠ ≡1   1D   1→ ≡1   1D   1∨ ≡1   1D   1∧
0000070 60000000000400000070 60000000000400000071 60000000000400000072 60000000000400000073 ≡1   1D   1↑ ≡1   1D   1↓ ≡1   1D   1< ≡1   1D   1>
0000074 60000000000400000074 60000000000400000075 60000000000400000076 60000000000400000077 ≡1   1D   1≤ ≡1   1D   1≥ ≡1   1D   1¬ ≡1   1D   1;
0000100 60000000000400000100 60000000000400000101 60000000000400000102 60000000000400000103 ≡1   1D   A1 ≡1   1D   AA ≡1   1D   AB ≡1   1D   AC
0000104 60000000000400000104 60080000000400000105 60000000000400000106 60000000000400000107 ≡1   1D   AD ≡1   1D   AE ≡1   1D   AF ≡1   1D   AG
```

```
  MFF   NA2-  CYB175-SN1    5201/R6B  06/07/78
12.43.34.DON00BU  FROM    /SH
12.43.34.IP  00000640 WORDS - FILE INPUT  , DC 04
12.43.34.DON,EC1. PSD,0278,77CT011A,MILLER
12.43.35.COMPASS.
12.43.36. ASSEMBLY COMPLETE.   47600B SCM USED.
12.43.36.     0.244  CPU SEC.    1100B LCM USED.
12.43.36.LGO.
12.43.36.DMP,111,337.
12.43.37.DMPECS,0,100.
12.43.37.     STOP  01
12.43.37.EXIT.
12.43.37.OP  00003200 WORDS - FILE OUTPUT , DC 40
12.43.37.MS     3584  WORDS  (    10752 MAX USED)
12.43.37.CPA         .330 SEC.            .330 ADJ.
12.43.37.IO          .785 SEC.            .785 ADJ.
12.43.37.CM       15.997 KWS.             .976 ADJ.
12.43.37.FC          .565 KWS.            .017 ADJ.
12.43.37.SS                              2.108
12.43.37.PP        2.914 SEC.      DATE 07/14/78
12.43.37.EJ  END OF JOB, SH
```

```
TRUNCOM                                                      COMPASS 3.5-470.     06/27/78  16.20.41.       PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH             BINARY CONTROL CARDS.

                0        2             IDENT   TRUNCOM
                2                      END     TRUNCOM


                              BLOCKS     TYPE       ADDRESS     LENGTH

                              PROGRAM*   LOCAL            0          2
                              BLOCK      COMMON           0         12


                              ENTRY POINTS.

                              TRUNCOM          0*


                              EXTERNAL SYMBOLS.

                              SYS=
```

```
TRUNCOM                                                      COMPASS 3.5-470.     06/27/78  16.20.41.       PAGE      2


                                             IDENT   TRUNCOM
                                             ENTRY   TRUNCOM
                                       *
                                       *     FIRST PROGRAM DECLARES 10 WORDS IN /BLOCK/
                                       *
                                             USE     /BLOCK/      USE LABELLED COMMON
  0                      12        BUF       BSSZ    10           RESERVE 10 WORDS OF ZEROS
                                             USE     0
  0  46000                         TRUNCOM   NO
          7160247021                         ENDRUN
  2                                          END     TRUNCOM

                  50400B SCM STORAGE USED             14 STATEMENTS          3 SYMBOLS
                         MODEL 174 ASSEMBLY      0.025 SECONDS               4 REFERENCES
```

```
TRUNCOM                                                      COMPASS 3.5-470.     06/27/78  16.20.41.       PAGE      3
SYMBOLIC REFERENCE TABLE.

BUF             0      BLOCK           2/07 L
SYS=            0      EXTERNAL*       2/11
TRUNCOM         0      PROGRAM*        2/02 E    2/09 L
```

```
TRUNCIT                                                    COMPASS 3.5-470.     06/27/78  16.20.42.      PAGE      1
STORAGE ALLOCATION.

            ADDRESS   LENGTH              BINARY CONTROL CARDS.

                  0        0              IDENT  TRUNCIT
                  0                       END

                                 BLOCKS     TYPE       ADDRESS    LENGTH

                                 BLOCK      COMMON           0        24


                                 ENTRY POINTS.

                                 BUF                0*/BLOCK/
```

```
TRUNCIT                                                    COMPASS 3.5-470.     06/27/78  16.20.42.      PAGE      2

                                              IDENT  TRUNCIT
                                              ENTRY  BUF
                                        .
                                        .
                                        .     SECOND PROGRAM DECLARES 20 WORDS IN /BLOCK/
                                              USE    /BLOCK/      USE LABELED COMMON
      0                    24     BUF         BSSZ   20
      0                                       END

                          47300B SCM STORAGE USED              8 STATEMENTS        1 SYMBOLS
                                 MODEL 174 ASSEMBLY      0.010 SECONDS          2 REFERENCES
```

```
TRUNCIT                                                    COMPASS 3.5-470.     06/27/78  16.20.42.      PAGE      3
SYMBOLIC REFERENCE TABLE.

BUF              0      BLOCK             2/02 E     2/07 L
```

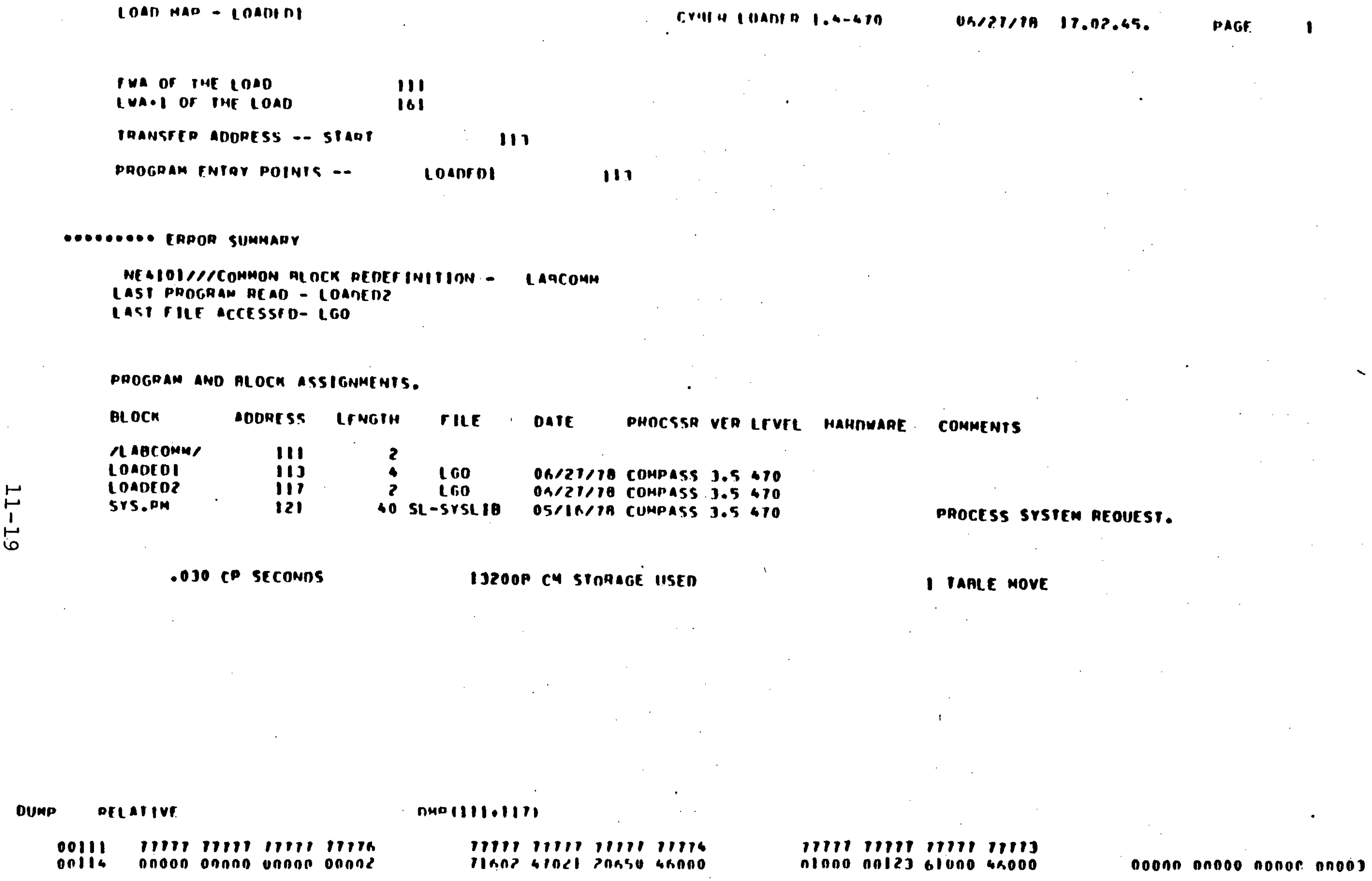
```
LOAD MAP - TRUNCOM                                          CYBER LOADER 1.4-470       06/27/78  16.20.43.      PAGE     1


      FWA OF THE LOAD              111
      LWA+1 OF THE LOAD            175

      TRANSFER ADDRESS -- TRUNCOM           123

      PROGRAM ENTRY POINTS --       TRUNCOM          123


 ********** ERROR SUMMARY

       NE4101///COMMON BLOCK REDEFINITION -    BLOCK
      LAST PROGRAM READ - TRUNCIT
      LAST FILE ACCESSED- LGO


      PROGRAM AND BLOCK ASSIGNMENTS.

      BLOCK        ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

      /BLOCK/          111       12
      TRUNCOM          123        2    LGO       06/27/78 COMPASS 3.5 470
      TRUNCIT          125       10    LGO       06/27/78 COMPASS 3.5 470
      SYS.RM           135       40 SL-SYSLIB    05/16/78 COMPASS 3.5 470                      PROCESS SYSTEM REQUEST.
DMPX.

 P    000000  A0  000100  B0  000000
 RA   220200  A1  000001  B1  000001     C(A1)=  0000  0000  0000  0000  0000     C(B1)=  0000  0000  0000  0000  0000
 FL   000100  A2  000060  B2  000002     C(A2)=  1505  1520  0000  0000  0061     C(B2)=  0000  0000  0000  0000  0000
 FM   700700  A3  000057  B3  012713     C(A3)=  0000  0000  0000  0000  0000     C(B3)=
 RE   000045  A4  000001  B4  000101     C(A4)=  0000  0000  0000  0000  0000     C(B4)=
 FE   000000  A5  000123  B5  000123     C(A5)=  0000  0000  0000  0000  0000     C(B5)=  0000  0000  0000  0000  0000
 MA   000600  A6  000001  B6  000300     C(A6)=  0000  0000  0000  0000  0000     C(B6)=
              A7  000001  B7  027756     C(A7)=  0000  0000  0000  0000  0000     C(B7)=
 X0   0000  0000  0000  0000  0000
 X1   0000  0000  0000  0000  0000
 X2   1505  1520  0000  0000  0061
 X3   0000  0000  0000  0000  0000
 X4   0000  0000  0000  0000  0000
 X5   6000  0000  0004  0040  0000
 X6   1505  1520  0000  0000  0061
 X7   0000  0000  0000  0000  0000

    00000    00000 00123 00000 00000         00000 00000 00000 00000
    00054   *56110 03110 00054 54710         51100 00001 03110 00055         64550 02550 00000 46000         00000 00000 00000 00000
    00060    15051 52000 00000 00061         00000 00300 00000 00001         07040 00060 51600 00001         04000 00063 00000 00021
    00064    14071 70000 00000 00000         00000 00000 00000 00175         40000 00000 02000 00111         00000 00000 40000 00000
    00070    14071 75755 00000 00000         00000 00000 00000 00000
    00100   *54000 00000 01000 00001

    00100    54000 00000 01000 00001         00006 40000 00000 00175         00000 00000 00000 00000
    00104   *00000 00000 00000 00175         00000 00000 00000 00000
    00110   *24222 51603 17150 00123         00000 00000 00000 00000
    00124   *00000 00137 00000 00000         00000 00000 00000 00000     00135*54110 20123 03310 00136         04000 00142 61000 46000
    00137    04004 00137 61000 46000
```

```
  MFF   NR2-  CYB175-SN1    4LA7/M68  05/15/78
16.20.40.DON00MJ  FROM    /SM
16.20.40.IP  00000256 WORDS - FILE INPUT  , DC 04
16.20.40.DON.       PSD,0278,72CT011A,MILLER
16.20.41.COMPASS.
16.20.42. ASSEMBLY COMPLETE.   50400B SCM USED.
16.20.42.     0.093  CPU SECONDS  ASSEMBLY TIME.
16.20.42.LGO.
16.20.43.        NON-FATAL LOADER ERRORS - SEE MAP
16.20.43.ERROR MODE =00.  ADDRESS =000123
16.20.43.OP  00001216 WORDS - FILE OUTPUT , DC 40
16.20.43.MS    3584  WORDS  (     7168 MAX USED)
16.20.43.CPA        .153 SEC.          .153 ADJ.
16.20.43.IO         .645 SEC.          .645 ADJ.
16.20.43.CM      10.589 KWS.           .646 ADJ.
16.20.43.SS                           1.445
16.20.43.PP         2.915 SEC.     DATE 06/27/78
16.20.43.EJ  END OF JOB, SM
```

```
LOADED1                                                               COMPASS 3.5-470.     06/27/78  17.02.43.       PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH                BINARY CONTROL CARDS.

                0        4            IDENT        LOADED1
                4                         END      START

                                    BLOCKS     TYPE        ADDRESS     LENGTH

                                    PROGRAM*   LOCAL             0          4
                                    LABCOMM    COMMON            0          2


                                    ENTRY POINTS.

                                    START              0*


                                    EXTERNAL SYMBOLS.

                                    SYS=
```

```
LOADED1                                                               COMPASS 3.5-470.     06/27/78  17.02.43.       PAGE      2

                                         IDENT      LOADED1
                                            ENTRY   START
                                   ************************************************************************
                                   *                                                                      *
                                   *       IMPROPER USE OF LABELED COMMON                                 *
                                   *                                                                      *
                                   ************************************************************************
     0  00000000000000000001       START     DATA    1,2
     1  00000000000000000002
     2                                       ENDRUN
     2  7160247021                 SX6 3REND*4+1                                                 ENDRUN  .1
                    20650          LX6 40D                                                       ENDRUN  .1
     3  0100000000 X               RJ =XSYS=                                                     ENDRUN  .1
                                   ENDM                                                          ENDRUN  .1
                                             USE     /LABCOMM/
     0  77777777777777777776                 DATA    -1,-2
     1  77777777777777777775

                                             USE     0

                                  DEFAULT SYMBOLS DEFINED BY COMPASS.

     0 X                          SYS=

     4                                       END     START

                         504008 SCM STORAGE USED            17 STATEMENTS        2 SYMBOLS
                                   MODEL 174 ASSEMBLY       0.035 SECONDS        3 REFERENCES
```

```
LOADED2                                                      COMPASS 3.5-470.     06/27/78  17.02.43.      PAGE      1
STORAGE ALLOCATION.

              ADDRESS   LENGTH                BINARY CONTROL CARDS.

                    0        2                IDENT  LOADED2
                    2                         END

                                   BLOCKS     TYPE       ADDRESS    LENGTH

                                   PROGRAM*   LOCAL            0         2
                                   LABCOMM    COMMON           0         3




LOADED2                                                      COMPASS 3.5-470.     06/27/78  17.02.43.      PAGE      2


                                              IDENT   LOADED2
      0  00000000000000000003                 DATA    3,4
      1  00000000000000000004
                                              USE     /LABCOMM/
      0                     1                 BSS     1
      1  77777777777777777774                 DATA    -3,-4
      2  77777777777777777773
      2                                       END

                    47300B SCM STORAGE USED             6 STATEMENTS          0 SYMBOLS
                           MODEL 174 ASSEMBLY       0.011 SECONDS             0 REFERENCES
```



```
LOAD MAP - LOADED1                                          CYBER LOADER 1.4-470       06/27/78  17.02.45.      PAGE      1


FWA OF THE LOAD              111
LWA+1 OF THE LOAD            161

TRANSFER ADDRESS -- START             113

PROGRAM ENTRY POINTS --       LOADED1         113


********** ERROR SUMMARY

 NE4101///COMMON BLOCK REDEFINITION -   LABCOMM
LAST PROGRAM READ - LOADED2
LAST FILE ACCESSED- LGO


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK          ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

/LABCOMM/         111        2
LOADED1           113        4    LGO      06/27/78 COMPASS 3.5 470
LOADED2           117        2    LGO      06/27/78 COMPASS 3.5 470
SYS.PM            121       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.


     .030 CP SECONDS              13200B CM STORAGE USED                      1 TABLE MOVE




DUMP    RELATIVE                        DMP(111,117)

   00111   77777 77777 77777 77776       77777 77777 77777 77774       77777 77777 77777 77773
   00114   00000 00000 00000 00002       71602 41021 20650 46000       01000 00123 61000 46000       00000 00000 00000 00003
```

```
DMPX.

P    000000   A0   000200   B0   000000
RA   313100   A1   000001   B1   000001     C(A1)=   0000   0000   0000   0000   0000      C(B1)=   0000   0000   0000   0000   0000
FL   000200   A2   000060   B2   000007     C(A2)=   1505   1520   0000   0000   0061      C(B2)=   0000   0000   0000   0000   0000
EM   700700   A3   000057   B3   012713     C(A3)=   0000   0000   0000   0000   0000      C(B3)=
RE   000051   A4   000001   B4   000201     C(A4)=   0000   0000   0000   0000   0000      C(B4)=
FE   000000   A5   000113   B5   000113     C(A5)=   7777   7777   7777   7777   7773      C(B5)=   7777   7777   7777   7777   7773
MA   001000   A6   000001   B6   000200     C(A6)=   0000   0000   0000   0000   0000      C(B6)=
              A7   000001   B7   027756     C(A7)=   0000   0009   0000   0000   0000      C(B7)=
X0   0000   0000   0000   0000   0000
X1   0000   0000   0000   0000   0000
X2   1505   1520   0000   0000   0061
X3   0000   0000   0000   0000   0000
X4   0000   0000   0009   0000   0000
X5   6000   0000   0004   0040   0000
X6   1505   1520   0000   0000   0061
X7   0000   0000   0000   0001   5023

   00000     00000 00114 00000 00000        00000 00000 00000 00000
   00054    *56110 03110 00054 54710        51100 00001 03110 00055        64550 02550 00000 46000        00000 00000 00000 00000
   00060     15051 52000 00000 00061        00000 00200 00000 00001        07040 00060 51600 00001        04000 00063 00000 00021
   00064     14071 70000 00000 00000        00000 00000 00000 00161        40000 00000 02000 00111        00000 00000 40000 00000
   00070     05300 50125 24055 12324        01222 45255 00000 00000        00000 00000 00000 00000
   00100    *54000 00000 01000 00001

   00100     54000 00000 01000 00001        00005 00000 00000 00161        00000 00000 00000 00000
   00104    *00000 00000 00000 00161        00000 00000 00000 00000
   00110    *14170 10405 04340 00113        77777 77777 77777 77776        77777 77777 77777 77774        77777 77777 77777 77773
   00114     00000 00000 00000 00002        71602 47021 20650 46000        01000 00123 61000 46000        00000 00000 00000 00003
   00120     00000 00000 00000 00004        54110 20123 03310 00122        04000 00126 61000 46000        04004 00123 61000 46000
   00124     51100 00001 03110 00124        54610 04000 00122 46000        51100 00066 03310 00130        51100 00121 04000 00131
   00130     71100 00130 20160 46000        13661 13161 13661 46000        51600 00122 10611 46000        51100 00001 01000 00121
   00134     20652 01000 00123 46000        51100 00001 03110 00135        04004 00136 61000 46000        51100 00001 03110 00136
   00140     71602 20314 04000 00134        20150 36661 01000 00123        04004 00142 61000 46000        71602 20314 20652 36662
   00144     53160 20173 03310 00142        03010 00142 51100 00001        03110 00144 71100 00001        04000 00141 61000 46000
   00150     71603 24616 12661 20651        01000 00123 61000 46000        04004 00152 61000 46000        73660 20630 12161 73610
   00154     20123 03210 00150 20151        13116 20636 51600 00160        74660 36116 20123 46000        04000 00150 61000 46000
   00160     00000 00000 00000 00000        60000 00000 04004 00161        60000 0000(
   00164     60000 00000 04004 00164        60000 00000 04004 00165        60000 0000(
   00170     60000 00000 04004 00170        60000 00000 04004 00171        60000 0000(
   00174     60000 00000 04004 00174        60000 00000 04004 00175        60000 0000(
   00200     00000 00000 00000 00000
```

```
   MFF   NB2-  CYB175-SN1     4LB7/R6B  05/15/78
17.02.40.00N00NZ  FROM     /SM
17.02.40.IP  00000320 WORDS - FILE INPUT  , DC 04
17.02.40.00N.      PSD,0278,72CT011A,MILLER
17.02.42.COMPASS(LO=$$$$)
17.02.43. ASSEMBLY COMPLETE.    50400B SCM USED.
17.02.43.     0.107  CPU SECONDS  ASSEMBLY TIME.
17.02.43.LOAD(LGO)
17.02.44.EXECUTE(START)
17.02.45.         NON-FATAL LOADER ERRORS - SEE MAP
17.02.45.ERROR MODE =00.  ADDRESS =000114
17.02.45.EXIT.
17.02.46.DMP(111,117)
17.02.46.OP  00001408 WORDS - FILE OUTPUT , DC 40
17.02.46.MS     3584  WORDS  (    10752 MAX USED)
17.02.46.CPA          .165 SEC.              .165 ADJ.
17.02.46.IO           .675 SEC.              .675 ADJ.
17.02.46.CM         11.048 KWS.              .674 ADJ.
17.02.46.SS                                 1.515
17.02.46.PP          4.588 SEC.      DATE 06/27/78
17.02.46.EJ  END OF JOB, SM
```

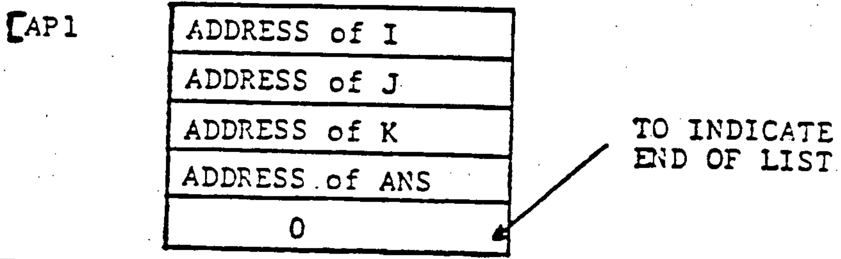
PARAMETERS
FORTRAN EXTENDED

THE CALL:

$$\text{CALL SUB } (\overbrace{\text{I , J , K , ANS}}^{\text{ACTUAL PARAMETERS}})$$

THE CODE GENERATED:

- AN AP List is Created: (ADDRESSES OF PARAMETERS)



- ADDRESS OF AP List is set into A1
- RJ is executed to Subroutine

Therefore, when the Subroutine is Entered:

- (A1) = AP list address
- (X1) = Address of First Parameter

EXAMPLE:

The VALUE of I is 1

The VALUE of J is 2

The VALUE of K is 3

The ADDRESS of I is 2060

The ADDRESS of J is 2061

The ADDRESS of K is 2062

The ADDRESS of ANS is 2057

2057	ANS	xxx
2060	I	1
2061	J	2
2062	K	3

The ADDRESS of the AP LIST is 2050

2050	AP1	2060	Address of I
2051		2061	Address of J
2052		2062	Address of K
2053		2057	Address of ANS
2054		0	

The CODE to enter the SUBROUTINE would be:

```
SA1   AP1
RJ    SUB
```

(A1) = 2050   (X1) = 2060

```
SUB                                                          COMPASS 3.6-476.     01/05/79  10.14.17.      PAGE      1

                                               IDENT   SUB
                                               ENTRY   SUB
                                               LIST    -B,-R
                                        ***************************************************************************
                                        *                                                                         *
                                        *      COMPASS SUBROUTINE PICKING UP PARAMETERS FROM FORTRAN MAIN PROGRAM *
                                        *      THE COMPASS SUBROUTINE MUST USE THE ADDRESS PASSED TO IT IN A1 TO  *
                                        *      LOCATE THE PARAMETERS AND RETURN THE ANSWER.                       *
                                        *                                                                         *
                                        *                                                                         *
                                        ***************************************************************************
  0  00000000000000000000       SUB     DATA    0             ADDRESS PARAMETER LIST IS IN A1 ON ENTRY
  1  53210                             SA2     X1            1ST PARAMETER IN X2 (I)
          5011000001                   SA1     A1+1          2ND PARAMETER ADDRESS IN X1
                    53310              SA3     X1            2ND PARAMETER IN X3 (J)
  2  5011000001                        SA1     A1+1          3RD PARAMETER ADDRESS IN X1 (J)
                 53410                 SA4     X1            3RD PARAMETER IN X4 (K)
                    36723              IX7     X2+X3         COMPUTE SUM
  3  36774                             IX7     X7+X4         COMPUTE SUM
          5011000001                   SA1     A1+1          4TH PARAMETER ADDRESS IN X1
                    53710              SA7     X1            STORE ANSWER (ANSWER)
  4  040000C000 +                      EQ      SUB           RETURN TO CALLER
  5                                    END

                  50600B CM  STORAGE USED               23 STATEMENTS        1 SYMBOLS
                             MODEL 74 ASSEMBLY       0.112 SECONDS           0 REFERENCES




                                            MFS  NHI-  CYB74-SN108    5C/R0B    11/14/78
                                          10.14.12.DON0090  FROM    /OH
                                          10.14.12.IP  00000320 WORDS - FILE INPUT  , DC 04
                                          10.14.12.DON,T5.   001A,6883,1896,MILLER
                                          10.14.15.FTN(OL,R=0)
                                          10.16.34.       .387 CP SECONDS COMPILATION TIME
                                          10.16.34.MAP,PART.
                                          10.16.34.LGO.
                                          10.16.43.      END  PARAMS
                                          10.16.43.       .062 CP SECONDS EXECUTION TIME
                                          10.16.43.OP  00001792 WORDS - FILE OUTPUT , DC 40
                                          10.16.43.MS    3584  WORDS  (   14336 MAX USED)
                                          10.16.43.CPA         .592 SEC.          .592 ADJ.
                                          10.16.43.CPB         .296 SEC.          .296 ADJ.
                                          10.16.43.IO         1.218 SEC.         1.218 ADJ.
                                          10.16.43.CM        36.786 KWS.         2.245 ADJ.
                                          10.16.43.SS                            4.353
                                          10.16.43.PP         7.116 SEC.     DATE 01/05/79
                                          10.16.43.EJ  END OF JOB, OH
```

```
     PROGRAM PARAMS      74/74   OPT=1                                        FTN 4.7+485          01/05/79  10.14.15        PAGE      1

 1                   PROGRAM PARAMS(OUTPUT)
                     INTEGER ANSWER
                     ANSWER=0
                     I=1
 5                   J=2
                     K=3
                     CALL SUB(I,J,K,ANSWER)
               C     SUB IS ENTERED WITH ADDRESS OF PARAMETER ADDRESSES IN A1.
                     PRINT 101
10                   PRINT 100,I,J,K,ANSWER
                 100 FORMAT(4I10)
                 101 FORMAT(1H1)
                     END




                                                        IDENT      PARAMS
                                                  USEBLK
                                                  LDSET   LIB=FORTRAN
                                                  USE START.


                                   000000  002062  START.    LOCAL
                                   002062  000000  VARDIM.   LOCAL
                                   002062  000000  ENTRY.    LOCAL
                                   002062  000021  CODE.     LOCAL
                                   002103  000022  DATA.     LOCAL
                                   002125  000000  DATA..    LOCAL
                                   002125  000000  HOL.      LOCAL

                                      EXTERNALS
                                 END.        OUTCI.     SUB         Q2NTRY.

        000000 START.     17252420252400000001            FILCRM  OUTPUT#,20038,0B
        000001 START.     00000000000000000051
        000002 START.     00000000000000000000
        000012 START.     00000000000060000000
        000013 START.     00000000220000000000
        000014 START.     00000000000000000000
        000015 START.     14000000000000000000
        000016 START.     00000000000000000000
        000017 START.     00000000060000000000
        000020 START.     00000000000000000000
        000021 START.     00000000000000002003

        002054 START.     20010000000000002057            LIBLNK. BSS 0B
        002055 START.     00000000000000141520              LIBLNK  1B,141520B
        002056 START.     00000000000000000000

        002057 START.     17252420252400000000            FILES. BSS 0B
        002060 START.     00000000000000000000              FLINK   OUTPUT#
        002061 START.     20012201152355002062              DATA 0
                                                            TRACE   PARAMS,PARAMS
                                                            USE CODE.
```

```
                                                 PENTRY  PARAMS,,,0
002062 CODE.     5110002054           START.     SA1 LIBLNK.
                           0100000000 <EXT>      RJ Q2NTRY.
                                                 USE DATA.
                                                 USE DATA..
                                                 USE DATA.
002103 DATA.                                   ]101      BSS 0B
002103 DATA.     00000000000000000000 START.     IOM     OUTPUT#
002104 DATA.     00000000000000002117 DATA.      IOM     .101
002105 DATA.     00000000000000000000            EIO     0B
002106 DATA.                                   ]102      BSS 0B
002106 DATA.     00000000000000000000 START.     IOM     OUTPUT#
002107 DATA.     00000000000000002115 DATA.      IOM     .100
002110 DATA.     00020000000100002122 DATA.      IOM     I,0B,2B,1B
002111 DATA.     00020000000100002123 DATA.      IOM     J,0B,2B,1B
002112 DATA.     00020000000100002124 DATA.      IOM     K,0B,2B,1B
002113 DATA.     00020000000100002121 DATA.      IOM     ANSWER,0B,2B,1B
002114 DATA.     00000000000000000000            EIO     0B
002115 DATA.     55343333555500000000            FMT     .100
002116 DATA.     51371134335255555555            DIS   1,(4I10)
002117 DATA.     55343334555500000000            FMT     .101
002120 DATA.     51341034525555555555            DIS   1,(1H1)
                                                 EXT     END.
                                                 EXT     OUTCI.
                                                 EXT     SUB
                                                 EXT     Q2NTRY.
002121 DATA.                                   ANSWER  BSS 1B
002122 DATA.                                   I       BSS 1B
002123 DATA.                                   J       BSS 1B
002124 DATA.                                   K       BSS 1B
                                                         USE    CODE.
                                               *                         LINE    3
002063 CODE.     43700                                   MX7    0B
                      7160000001                         SX6    1B
002064 CODE.     5170002121           DATA.              SA7    ANSWER
                           7170000002                    SX7    2B
002065 CODE.     5160002122           DATA.              SA6    I
                           7160000003                    SX6    3B
002066 CODE.     5170002123           DATA.              SA7    J
                           5160002124 DATA.              SA6    K
002067 CODE.     5110002076           CODE.              SA1    [AP]
002070 CODE.     0100000000           <EXT>    +         RJT    SUB,7B
                           0007002061
002071 CODE.     5110002103           DATA.              SA1    ]101
002072 CODE.     0100000000           <EXT>    +         RJT    OUTCI.,11B
                           0011002061
002073 CODE.     5110002106           DATA.              SA1    ]102
002074 CODE.     0100000000           <EXT>    +         RJT    OUTCI.,12B
                           0012002061
002075 CODE.     5110002061           START.             SA1    TRACE.
                           0400000000 <EXT>              EQ     END.
002076 CODE.                                    [AP]     BSS    0B
002076 CODE.     00000000000000002122 DATA.              APL    I
002077 CODE.     00000000000000002123 DATA.              APL    J
002100 CODE.     00000000000000002124 DATA.              APL    K
002101 CODE.     00000000000000002121 DATA.              APL    ANSWER
002102 CODE.     00000000000000000000                    APL
```

```
LOAD MAP - SUB                                          CYBER LOADER 1.4-485      01/05/79  10.16.40.     PAGE      1


FWA OF THE LOAD              111
LWA+1 OF THE LOAD           7062

TRANSFER ADDRESS -- PARAMS           2200

PROGRAM ENTRY POINTS --        SUB             2200


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS   LENGTH     FILE       DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

SUB              111       5      LGO        01/05/79 COMPASS 3.6 476
PARAMS           116    2125      LGO        01/05/79 FTN     4.7 485     666X I     PROGRAM   OPT=1
/STP.END/       2243       1
/FCL.C./        2244      26
/QB.IO./        2272     101
Q2NTRY=         2373       0 SL-FORTRAN  11/16/78 COMPASS 3.6 485                FCL INITIALIZATION ROUTINE.
/FCL=ENT/       2373      40
COMIO=          2433      33 SL-FORTRAN  11/16/78 COMPASS 3.6 485                COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL=FDL         2466      40 SL-FORTRAN  11/16/78 COMPASS 3.6 485                FCL CAPSULE LOADING
FEIFST=         2526       3 SL-FORTRAN  11/16/78 COMPASS 3.6 485                 CONVERTED DATA STORAGE
FLTOUT=         2531     311 SL-FORTRAN  11/16/78 COMPASS 3.6 485                COMMON FLOATING OUTPUT CODE
FORSYS=         3042     301 SL-FORTRAN  11/16/78 COMPASS 3.6 485                FORTRAN OBJECT LIBRARY UTILITIES.
OUTCOM=         3343     154 SL-FORTRAN  11/16/78 COMPASS 3.6 485                COMMON OUTPUT CODE
SYSAID=         3517       1 SL-FORTRAN  11/16/78 COMPASS 3.6 485                LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
FECMSK=         3520      41 SL-FORTRAN  11/16/78 COMPASS 3.6 485                INITIALIZE CONSTANTS.
FMTAP=          3561     357 SL-FORTRAN  11/16/78 COMPASS 3.6 485                CRACK APLIST AND FORMAT FOR KODER/KRAKER.
FORUTL=         4140      46 SL-FORTRAN  11/16/78 COMPASS 3.6 485                FCL MISC. UTILITIES.
GETFIT=         4206      57 SL-FORTRAN  11/16/78 COMPASS 3.6 485                LOCATE AN FIT GIVEN A FILE NAME.
KODER=          4265     451 SL-FORTRAN  11/16/78 COMPASS 3.6 485                OUTPUT FORMAT INTERPRETER.
OUTC=           4736     150 SL-FORTRAN  11/16/78 COMPASS 3.6 485                FORMATTED WRITE FORTRAN RECORD.
/FDL.COM/       5106      14
FDL.RES         5122     211 SL-SYSLIB   11/02/78 COMPASS 3.6 485                FAST DYNAMIC LOADER RESIDENT.
FDL.MMI         5333     222 SL-SYSLIB   11/02/78 COMPASS 3.6 485                FDL MEMORY MANAGER INTERFACE.
CPU.SYS         5555      40 SL-SYSLIB   11/15/78 COMPASS 3.6 476                PROCESS SYSTEM REQUEST.
CMF.ALF         5615     160 SL-SYSLIB   11/15/78 COMPASS 3.6 485                CMM V1.1 - ALLOCATE FIXED.
CMF.CSF         5775       6 SL-SYSLIB   11/15/78 COMPASS 3.6 485                CMM V1.1 - CHANGE SPECS FIXED.
CMM.FFA         6003      14 SL-SYSLIB   11/15/78 COMPASS 3.6 485                CMM V1.1 - FIXED FREE ALGORITHM.
CMF.FRF         6017      36 SL-SYSLIB   11/15/78 COMPASS 3.6 485                CMM V1.1 - FREE FIXED.
CMM.R           6055     214 SL-SYSLIB   11/15/78 COMPASS 3.6 485                CMM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF         6271      22 SL-SYSLIB   11/15/78 COMPASS 3.6 485                CMM V1.1 - SHRINK AT LWA FIXED.
CTL$RM          6313     433 SL-SYSLIB   11/16/78 COMPASS 3.6 485                CRM CONTROLLING ROUTINE.
ERR$RM          6746      25 SL-SYSLIB   11/16/78 COMPASS 3.6 485                CRM ERROR PROCESSOR ENTRY.
LIST$RM         6773      67 SL-SYSLIB   11/16/78 COMPASS 3.6 485                CRM - ALLOCATE SPACE FOR LIST OF FILES


     .336 CP SECONDS               23500B CM STORAGE USED                12 TABLE MOVES

                                   1           2           3           6
```

```
    PROGRAM DUMMY       73/74   OPT=1                             FTN 4.7+485         01/05/79  10.13.38       PAGE      1

                  PROGRAM DUMMY(OUTPUT)
             ************************************************************************
             *                                                                      *
             *                                                                      *
             *    A DUMMY FORTRAN PROGRAM IS SET UP TO DECLARE THE FILES AND INITIALIZE *
             *    AND DUMP THE BUFFERS CORRECTLY.  THIS IS DONE BY Q8ENTRY.          *
             *                                                                      *
             *                                                                      *
             ************************************************************************
                  CALL MAINPGM
                  END
```

```
MAINPGM                                                       COMPASS 3.6-476.     01/05/79  10.13.42.      PAGE      1

                                                 IDENT   MAINPGM
                                                 ENTRY   MAINPGM
                                                 EXT     OUTSUB
                                                 LIST    -B,-R
                                          ************************************************************************
                                          *                                                                      *
                                          *                                                                      *
                                          *        COMPASS MAIN PROGRAM CALLING A FORTRAN SUBROUTINE THAT DOES I/O *
                                          *                                                                      *
                                          *                                                                      *
                                          ************************************************************************
        0  00000000000000000000      MAINPGM  DATA    0
        1  7160000010                         SX6     8
                    5160000000 C              SA6     BUF
        2  5160000001 C                       SA6     BUF+1
                    5160000002 C              SA6     BUF+2
        3  0100000000 X                       RJ      OUTSUB
        4  0400000000 +                       EQ      MAINPGM
                                              USE     /LAB/
        0                 3          BUF      BSSZ    3
        5                                     END

                   50600B CM  STORAGE USED            21 STATEMENTS        3 SYMBOLS
                              MODEL 74 ASSEMBLY    0.094 SECONDS           0 REFERENCES
```

```
  SUBROUTINE OUTSUB      73/74    OPT=1                           FTN 4.7+485         01/05/79  10.13.38       PAGE      1

                     SUBROUTINE OUTSUB
                   COMMON /LAB/ IBUF(3)
                   PRINT 101
                   PRINT 100,(IBUF(I),I=1,3)
               100 FORMAT(3O24)
               101 FORMAT(1H1)
                   END
```

```
LOAD MAP - DUMMY                                      CYBER LOADER 1.4-485       01/05/79  10.13.53.      PAGE      1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD         7050

TRANSFER ADDRESS -- DUMMY           2173

PROGRAM ENTRY POINTS --       DUMMY           2173


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK       ADDRESS   LENGTH    FILE     DATE     PROCSSR VER LEVEL  HARDWARE   COMMENTS

DUMMY            111    2065    LGO      01/05/79 FTN     4.7 485    666X I     PROGRAM   OPT=1
/LAB/           2176       3
MAINPGM         2201       5    LGO      01/05/79 COMPASS 3.6 476
OUTSUB          2206      23    LGO      01/05/79 FTN     4.7 485    666X I     SUBROUTINEOPT=1
/STP.END/       2231       1
/FCL.C./        2232      26
/Q8.IO./        2260     101
Q2NTRY=         2361       0 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FCL INITIALIZATION ROUTINE.
/FCL=ENT/       2361      40
COMIO=          2421      33 SL-FORTRAN  11/16/78 COMPASS 3.6 485               COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL=FDL         2454      40 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FCL CAPSULE LOADING
FEIFST=         2514       3 SL-FORTRAN  11/16/78 COMPASS 3.6 485                CONVERTED DATA STORAGE
FLTOUT=         2517     311 SL-FORTRAN  11/16/78 COMPASS 3.6 485               COMMON FLOATING OUTPUT CODE
FORSYS=         3030     301 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FORTRAN OBJECT LIBRARY UTILITIES.
OUTCOM=         3331     154 SL-FORTRAN  11/16/78 COMPASS 3.6 485               COMMON OUTPUT CODE
SYSAID=         3505       1 SL-FORTRAN  11/16/78 COMPASS 3.6 485               LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
FECMSK=         3506      41 SL-FORTRAN  11/16/78 COMPASS 3.6 485               INITIALIZE CONSTANTS.
FMTAP=          3547     357 SL-FORTRAN  11/16/78 COMPASS 3.6 485               CRACK APLIST AND FORMAT FOR KODER/KRAKER.
FORUTL=         4126      46 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FCL MISC. UTILITIES.
GETFIT=         4174      57 SL-FORTRAN  11/16/78 COMPASS 3.6 485               LOCATE AN FIT GIVEN A FILE NAME.
KODER=          4253     451 SL-FORTRAN  11/16/78 COMPASS 3.6 485               OUTPUT FORMAT INTERPRETER.
OUTC=           4724     150 SL-FORTRAN  11/16/78 COMPASS 3.6 485               FORMATTED WRITE FORTRAN RECORD.
/FDL.COM/       5074      14
FDL.RES         5110     211 SL-SYSLIB   11/02/78 COMPASS 3.6 485               FAST DYNAMIC LOADER RESIDENT.
FDL.MMI         5321     222 SL-SYSLIB   11/02/78 COMPASS 3.6 485               FDL MEMORY MANAGER INTERFACE.
CPU.SYS         5543      40 SL-SYSLIB   11/15/78 COMPASS 3.6 476               PROCESS SYSTEM REQUEST.
CMF.ALF         5603     160 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - ALLOCATE FIXED.
CMF.CSF         5763       6 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - CHANGE SPECS FIXED.
CMM.FFA         5771      14 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - FIXED FREE ALGORITHM.
CMF.FRF         6005      36 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - FREE FIXED.
CMM.R           6043     214 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF         6257      22 SL-SYSLIB   11/15/78 COMPASS 3.6 485               CMM V1.1 - SHRINK AT LWA FIXED.
CTL$RM          6301     433 SL-SYSLIB   11/16/78 COMPASS 3.6 485               CRM CONTROLLING ROUTINE.
ERR$RM          6734      25 SL-SYSLIB   11/16/78 COMPASS 3.6 485               CRM ERROR PROCESSOR ENTRY.
LIST$RM         6761      67 SL-SYSLIB   11/16/78 COMPASS 3.6 485               CRM - ALLOCATE SPACE FOR LIST OF FILES


      .350 CP SECONDS              23500B CM STORAGE USED                     12 TABLE MOVES
```

```
00000000000000000010    00000000000000000010    00000000000000000010
```

```
  MFS  NB1-  CYB74-SN108    SC/R08    11/14/78
10.13.36.DON009L  FROM    /OH
10.13.36.IP  00000384 WORDS - FILE INPUT  , DC 04
10.13.36.DON,TS.   001A,6883,1896,MILLER
10.13.38.FTN,R=0.
10.13.45.        .366 CP SECONDS COMPILATION TIME
10.13.45.MAP,PART.
10.13.45.LGO.
10.13.56.       END  DUMMY
10.13.56.        .071 CP SECONDS EXECUTION TIME
10.13.56.OP  00001088 WORDS - FILE OUTPUT , DC 40
10.13.56.MS    3584  WORDS  (   14336 MAX USED)
10.13.56.CPA         .623 SEC.          .622 ADJ.
10.13.56.CPB         .260 SEC.          .260 ADJ.
10.13.56.IO         1.663 SEC.         1.663 ADJ.
10.13.56.CM        46.133 KWS.         2.815 ADJ.
10.13.56.SS                            5.361 .
10.13.56.PP        11.208 SEC.     DATE 01/05/79
10.13.56.EJ  END OF JOB, OH
```

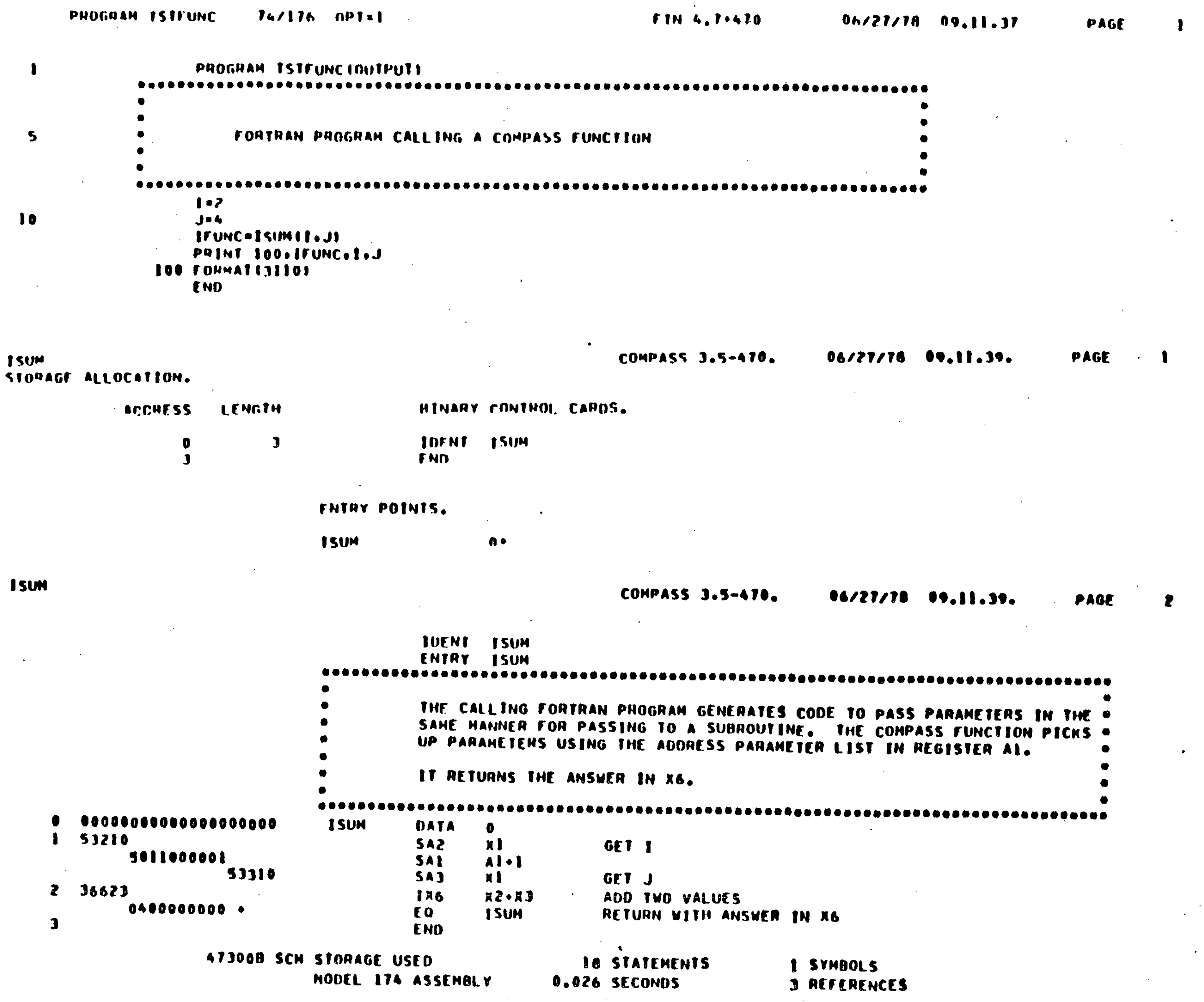
```
MAINPRG                                                 COMPASS 3.5-470.     06/27/78  09.11.51.      PAGE      1
STORAGE ALLOCATION.

          ADDRESS   LENGTH              BINARY CONTROL CARDS.

                0        5              IDENT   MAINPRG
                5                       END     MAINPRG

                              BLOCKS     TYPE       ADDRESS    LENGTH

                              PROGRAM*   LOCAL            0         5
                              LAB        COMMON           0         3

                              ENTRY POINTS.

                              MAINPRG            0*

                              EXTERNAL SYMBOLS.

                              OUTSUB     SYS=

MAINPRG                                                 COMPASS 3.5-470.     06/27/78  09.11.51.      PAGE      2


                                        IDENT   MAINPRG
                                        ENTRY   MAINPRG
                                        EXT     OUTSUB
                              **************************************************************************
                              *                                                                        *
                              *                                                                        *
                              *       INCORRECT SETUP FOR A COMPASS MAIN PROGRAM AND FORTRAN SUBROUTINE *
                              *                                                                        *
                              *                                                                        *
                              **************************************************************************
     0  7160000010            MAINPRG   SX6     8
                 5160000000 C           SA6     BUF
     1  5160000001 C                    SA6     BUF+1
                 5160000002 C           SA6     BUF+2
     2  0100000000 X                    RJ      OUTSUB
     3  7160247021                      ENDRUN
                                        USE     /LAB/
     0                     3  BUF       BSSZ    3
     5                                  END     MAINPRG

                 50400B SCM STORAGE USED            22 STATEMENTS         4 SYMBOLS
                        MODEL 174 ASSEMBLY       0.042 SECONDS            9 REFERENCES

MAINPRG                                                 COMPASS 3.5-470.     06/27/78  09.11.51.      PAGE      3
SYMBOLIC REFERENCE TABLE.

BUF            0      LAB            2/12 S     2/13 S     2/14 S     2/18 L
MAINPRG        0      PROGRAM*       2/02 E     2/11 L
OUTSUB         0      EXTERNAL*      2/03 X     2/15
SYS=           0      EXTERNAL*      2/17
```

```
SUBROUTINE OUTSUB      74/176  OPT=1                              FTN 4.7+470          06/27/78  09.11.49         PAGE      1

              SUBROUTINE OUTSUB
       ****************************************************************************
       *                                                                          *
       *                                                                          *
       *      THE SUBROUTINE CAN NOT OUTPUT CORRECTLY BECAUSE THE FILE BUFFERS HAVE NOT*
       *      BEEN INITIALIZED.  A FORTRAN MAIN PROGRAM IS REQUIRED.              *
       *                                                                          *
       *                                                                          *
       ****************************************************************************
              COMMON /LAB/ IBUF(3)
                 PRINT 100,(IBUF(I),I=1,3)
          100 FORMAT(3O20)
              END
```

```
   MFF   NH2-  CYB175-SN1    4LB7/R6B  05/15/78
 09.11.47.DON006N  FROM    /SH
 09.11.47.IP  00000320 WORDS - FILE INPUT  , DC 04
 09.11.47.DON.      PSD,0278,72CT011A,MILLER
 09.11.49.FTN,R=0.
 09.11.52.       .122 CP SECONDS COMPILATION TIME
 09.11.52.MAP,PART.
 09.11.52.LGO.
 09.11.54.      NON-FATAL LOADER ERRORS - SEE MAP
 09.11.54.FTN - NO OUTPUT FILE FOUND - EXECUTION E
 09.11.54.RRORS
 09.11.54.ERROR MODE =01.  ADDRESS =405373
 09.11.54.OP  00001792 WORDS - FILE OUTPUT , DC 40
 09.11.54.MS    3584  WORDS  (   14336 MAX USED)
 09.11.54.CPA        .279 SEC.         .279 ADJ.
 09.11.54.IO        1.629 SEC.        1.629 ADJ.
 09.11.54.CM       31.296 KWS.        1.910 ADJ.
 09.11.54.SS                          3.819
 09.11.54.PP        6.997 SEC.     DATE 06/27/78
 09.11.54.EJ  END OF JOB, SH
```

```
LOAD MAP - MAINPRG                                           CYBER LOADER 1.4-470       06/27/78  09.11.54.      PAGE      1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD         5162

TRANSFER ADDRESS -- MAINPRG          114

PROGRAM ENTRY POINTS --        MAINPRG          114


********** ERROR SUMMARY

  NE4100///  UNSATISFIED EXTERNAL REF -- OUTPUT=


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK         ADDRESS   LENGTH    FILE      DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

/LAB/            111        3
MAINPRG          114        5     LGO      06/27/78 COMPASS 3.5 470
OUTSUB           121       14     LGO      06/27/78 FTN     4.7 470     767X I     SUBROUTINEOPT=1
/Q8.IO./         135       77
/FCL=ENT/        214       42
/FCL.C./         276       25
COMIO=           323       14 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL=FDL          337       40 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FCL CAPSULE LOADING
FMTAP=           377      377 SL-FORTRAN  05/02/78 COMPASS 3.5 470                CRACK APLIST AND FORMAT FOR KODER/KRAKER.
FORUTL=          776       46 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FCL MISC. UTILITIES.
GETFIT=         1044       61 SL-FORTRAN  05/02/78 COMPASS 3.5 470                LOCATE AN FIT GIVEN A FILE NAME.
KODER=          1125      461 SL-FORTRAN  05/02/78 COMPASS 3.5 470                OUTPUT FORMAT INTERPRETER.
OUTC=           1606      150 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FORMATTED WRITE FORTRAN RECORD.
FECMSK=         1756       41 SL-FORTRAN  05/02/78 COMPASS 3.5 470                INITIALIZE CONSTANTS.
FLTOUT=         2017      315 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON FLOATING OUTPUT CODE
FORSYS=         2334      302 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FORTRAN OBJECT LIBRARY UTILITIES.
OUTCOM=         2636      204 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON OUTPUT CODE
CMF.ALF         3042      160 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - ALLOCATE FIXED.
CMF.CSF         3222        6 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - CHANGE SPECS FIXED.
CMF.FFA         3230       14 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - FIXED FREE ALGORITHM.
CMF.FRF         3244       36 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - FREE FIXED.
CMM.R           3302      213 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF         3515       22 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - SHRINK AT LWA FIXED.
CTL$RM          3537      601 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CRM CONTROLLING ROUTINE.
ERR$RM          4340       25 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CRM ERROR PROCESSOR ENTRY.
LIST$RM         4365       66 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CRM - ALLOCATE SPACE FOR LIST OF FILES
/FDL.COM/       4451       14
FDL.RES         4467      211 SL-SYSLIB   05/16/78 COMPASS 3.5 470                FAST DYNAMIC LOADER RESIDENT.
FDL.MMI         4700      222 SL-SYSLIB   05/16/78 COMPASS 3.5 470                FDL MEMORY MANAGER INTERFACE.
SYS.RM          5122       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470                PROCESS SYSTEM REQUEST.


     .120 CP SECONDS              21200B CM STORAGE USED    ,                  6 TABLE MOVES
```

```
DMPX.

 P    000000   A0   000001   B0   000000
 PA   203600   A1   000111   B1   000001      C(A1)=  4135  5555  5555  5555  5555      C(B1)=  0000  0000  0000  0000  0000
 FL   007200   A2   000305   B2   000001      C(A2)=  0000  0000  0000  0000  0002      C(B2)=  0000  0000  0000  0000  0000
 FM   700700   A3   000313   B3   000074      C(A3)=  0000  0000  0000  0000  0000      C(B3)=  0000  0000  0000  0000  0000
 RE   000045   A4   005507   B4   000001      C(A4)=  7777  7777  7777  7777  7700      C(B4)=  0000  0000  0000  0000  0000
 FF   000000   A5   005424   B5   000001      C(A5)=  4000  0000  0000  0000  0000      C(B5)=  0000  0000  0000  0000  0000
 MA   000600   A6   001015   B6   000000      C(A6)=  4135  5555  5555  5555  5555      C(B6)=  0001  4053  7300  0000  0000
               A7   005205   B7   000014      C(A7)=  0000  0000  0000  0000  0076      C(B7)=  0000  0000  0000  0000  0000
 X0   7777  7777  7777  7700  0000
 X1   4135  5555  5555  5555  5555
 X2   0000  0000  0000  0000  0002
 X3   1725  2420  2524  0000  0000
 X4   7777  7777  7777  7777  7601
 X5   0000  0000  0000  0000  0000
 X6   4135  5555  5555  5555  5555
 X7   2000  0000  0000  0000  0006

    00000     00014 05373 00000 00000      00000 00000 00000 00000
                                                                     64550 02550 00000 46000      00000 00000 00000 00000
    00054    -56110 03110 00054 54710      51100 00001 03110 00055
                                                                     07040 00060 51600 00001      04000 00063 00000 00021
    00060     15051 52000 00000 00061      00000 05200 00000 00001
                                                                     40000 00000 02000 00111      00000 00000 40000 00000
    00064     14071 70000 00000 00000      00000 00000 00007 72615
    00070     14071 75755 00000 00000      00000 00000 00000 00000
    00100    -54000 00000 01000 00001

    05273     15557 71600 00151 11565
    05274     03050 05277 61000 46000      71200 03220 61000 46000      01000 02512 00000 05214      52510 00017 43746 15657
    05300     51600 05505 04000 05304      52530 00015 71600 05430      43752 15667 11775 36667      54650 46000 61000 46000
    05304     51100 05205 51200 00307      61500 00001 61300 00203      63210 53010 10711 46000      06320 05311 51000 00064
    05310     74600 51600 05205 46000      04000 05312 61000 46000      51200 00313 51100 05205      03020 02471 51200 00307
    05314     21122 03110 05420 43060      51504 05373 53220 55655      13225 61400 00012 77304      15420 22602 54225 54665
    05320     73334 03140 05317 46000      71204 05377 03060 05324      72330 00012 01000 01006      54660 46000 61000 46000
    05324     01000 05453 00000 05241      51500 05515 51400 00311      10755 51704 05404 54355      43044 20430 15640 46000
    05330     51500 05517 12736 54775      51300 00314 10655 63330      51200 02471 54675 21236      52427 77776 61000 46000
    05334     10544 21560 62157 77677      43752 43014 05100 05364      43536 51300 05521 13143      15615 03060 05364 21336
    05340     13143 15615 03060 05364      63740 04700 05364 54355      53140 11271 20252 12623      54665 20036 11504 63210
    05344     03050 05374 51200 05522      21522 10722 62157 70000      54765 05100 05347 54545      73450 66420 01000 01001
    05350     66240 54675 61000 46000      61404 05426 75364 76130      76520 20122 12615 36033      51600 05511 71204 05431
    05354     20303 36303 72337 77775      01000 05453 00000 05241      51100 05511 63210 21122      63310 51604 05434 56220
    05360     73120 04200 05364 46000      03110 05364 67335 20236      52427 77776 51500 05523      10655 54665 07030 05334
    05364     51400 00310 03140 05444      51400 05205 61300 00074      61500 00001 67405 63244      67223 66445 06200 05367
    05370     51440 05506 66223 22424      03240 05247 61000 46000      51100 00311 10611 46000      51600 01015 04004 05373
    05374     53340 64440 63740 77347      51400 00276 43260 11624      54665 03330 05351 37666      43271 20352 61100 00006
    05400     61400 00033 61000 46000      20303 15432 73544 20606      36665 67115 05100 05401      51200 05525 10722 20614
    05404     71300 02355 20360 12636      54760 54675 04000 05351      06241 65546 55161 75517      25242 02524 55061 11405
    05410     55061 72516 04554 65505      30050 32524 11171 65505      22221 72223 55000 00000      51500 05424 03150 05416
    05414     71100 05406 43652 15116      71602 00000 01000 05153      43601 51600 05424 46000      04000 05365 61000 46000
    05420     26212 76310 36033 20303      36303 46000 61000 46000      01000 05453 00000 05241      04000 05247 61000 46000
    05424     40000 00000 00000 00000      06241 65546 55172 52420      25245 50611 14055 52516      25230 10214 05000 00000
    05430     00000 00000 00000 00000      61500 00001 71100 05425      43652 15116 71602 00000      01000 05153 61000 46000
    05434     51100 00313 52510 00015      76600 43752 15667 11775      36667 54650 51100 05205      21122 03010 05417 46000
    05440     51500 00307 73150 43652      15116 71602 00000 46000      01000 05153 61000 46000      04000 05247 61000 46000
    05444     13777 54740 04000 02476      61500 00001 51300 02476      10633 51600 02471 46000      51600 00310 04000 02472
    05450     23312 30522 22550 05453      77777 77777 77777 77777      61500 00001 61000 46000      04004 05453 61000 46000
    05454     51100 00313 61100 00001      53010 64500 61600 05457      04000 04273 43000 00000      51100 00313 51200 05505
    05460     51300 05205 52510 00017      43746 15627 11775 36667      54650 62437 77654 46000      04400 05452 53210 43452
    05464     51300 00301 11242 37423      03140 05452 51200 00306      52510 00622 43736 15357      37432 03140 05452 46000
    05470     51200 00312 51300 02471      10622 22703 51600 05502      51700 05503 10644 20636      51600 05504 71100 00123
```



```
    PROGRAM TSTFUNC    74/176  OPT=1                        FTN 4.7+470       06/27/78  09.11.37       PAGE      1

                 PROGRAM TSTFUNC(OUTPUT)
          *******************************************************************
          *                                                                 *
          *                                                                 *
          *      FORTRAN PROGRAM CALLING A COMPASS FUNCTION                 *
          *                                                                 *
          *                                                                 *
          *******************************************************************
                 I=2
                 J=4
                 IFUNC=ISUM(I,J)
                 PRINT 100,IFUNC,I,J
             100 FORMAT(3I10)
                 END


ISUM                                                      COMPASS 3.5-470.     06/27/78  09.11.39.      PAGE    1
STORAGE ALLOCATION.

             ADDRESS   LENGTH              BINARY CONTROL CARDS.

                   0        3              IDENT  ISUM
                   3                       END

                                ENTRY POINTS.

                                ISUM            0+

ISUM
                                                          COMPASS 3.5-470.     06/27/78  09.11.39.      PAGE    2

                                           IDENT  ISUM
                                           ENTRY  ISUM
                                ********************************************************************
                                *                                                                  *
                                *      THE CALLING FORTRAN PROGRAM GENERATES CODE TO PASS PARAMETERS IN THE *
                                *      SAME MANNER FOR PASSING TO A SUBROUTINE.  THE COMPASS FUNCTION PICKS *
                                *      UP PARAMETERS USING THE ADDRESS PARAMETER LIST IN REGISTER A1.      *
                                *                                                                  *
                                *      IT RETURNS THE ANSWER IN X6.                                *
                                *                                                                  *
                                ********************************************************************
      0  00000000000000000000    ISUM     DATA    0
      1  53210                            SA2     X1          GET I
              5011000001                  SA1     A1+1
                         53310            SA3     X1          GET J
      2  36623                            IX6     X2+X3       ADD TWO VALUES
              0400000000 +                EQ      ISUM        RETURN WITH ANSWER IN X6
      3                                   END

                 47300B SCM STORAGE USED                18 STATEMENTS          1 SYMBOLS
                           MODEL 174 ASSEMBLY        0.026 SECONDS             3 REFERENCES
```

```
LOAD MAP - TSTFUNC                                  CYBER LOADER 1.4-470        06/27/78  09.11.41.       PAGE      1


LWA OF THE LOAD              111
LWA+1 OF THE LOAD           7251

TRANSFER ADDRESS -- TSTFUNC            2173

PROGRAM ENTRY POINTS --        TSTFUNC           2173



PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK         ADDRESS   LENGTH    FILE      DATE       PROCSSR VER LEVEL  HARDWARE   COMMENTS

TSTFUNC          111     2110     LGO     06/27/78 FTN     4.7 470     767X I     PROGRAM   OPT=1
ISUM            2221        3     LGO     06/27/78 COMPASS 3.5 470
SYSAID=         2224        1 SL-FORTRAN  05/02/78 COMPASS 3.5 470                LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
/STP.END/       2225        1
/FCL.C./        2226       25
/QQ.IO./        2253       77
Q2NTRY=         2352        0 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FCL INITIALIZATION ROUTINE.
/FCL=ENT/       2352       42
COMIO=          2414       14 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL=FDL         2430       40 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FCL CAPSULE LOADING
FMTAP=          2470      377 SL-FORTRAN  05/02/78 COMPASS 3.5 470                CRACK APLIST AND FORMAT FOR KODER/KRAKER.
FORUTL=         3067       46 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FCL MISC. UTILITIES.
GETFIT=         3135       61 SL-FORTRAN  05/02/78 COMPASS 3.5 470                LOCATE AN FIT GIVEN A FILE NAME.
KODER=          3216      461 SL-FORTRAN  05/02/78 COMPASS 3.5 470                OUTPUT FORMAT INTERPRETER.
OUTC=           3677      150 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FORMATTED WRITE FORTRAN RECORD.
FECHSK=         4047       41 SL-FORTRAN  05/02/78 COMPASS 3.5 470                INITIALIZE CONSTANTS.
FLTOUT=         4110      315 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON FLOATING OUTPUT CODE
FORSYS=         4425      302 SL-FORTRAN  05/02/78 COMPASS 3.5 470                FORTRAN OBJECT LIBRARY UTILITIES.
OUTCOM=         4727      204 SL-FORTRAN  05/02/78 COMPASS 3.5 470                COMMON OUTPUT CODE
CMF.ALF         5133      160 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - ALLOCATE FIXED.
CMF.CSF         5313        6 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - CHANGE SPECS FIXED.
CMM.FFA         5321       14 SL-SYSLIB   05/02/78 COMPASS 3.5 470                CMM V1.1 - FIXED FREE ALGORITHM.
CMF.FRF         5335       36 SL-SYSLIB   05/02/78 COMPASS 3.5 470
CMM.R           5373      213 SL-SYSLIB   05/02/78 COMPASS 3.5 470
CMF.SLF         5606       22 SL-SYSLIB   05/02/78 COMPASS 3.5 470
CTL$RM          5630      601 SL-SYSLIB   05/02/78 COMPASS 3.5 470
ERR$RM          6431       25 SL-SYSLIB   05/02/78 COMPASS 3.5 470
LIST$RM         6456       66 SL-SYSLIB   05/02/78 COMPASS 3.5 470
/FDL.COM/       6544       14
FDL.RES         6560      211 SL-SYSLIB   05/16/78 COMPASS 3.5 470
FDL.MMI         6771      222 SL-SYSLIB   05/16/78 COMPASS 3.5 470
SYS.RM          7213       40 SL-SYSLIB   05/16/78 COMPASS 3.5 470


     .114 CP SECONDS                 233000B CM STORAGE USED
```

```
  MFF   NR2-  CYR175-SN1    4LB7/R6B  05/15/78
09.11.35.DON006M  FROM    /SH
09.11.35.IP  0000034 WORDS - FILE INPUT  , DC 04
09.11.35.DON.      PSR.02/R.72CT011A.MILLER
09.11.37.FTN,R=0.
09.11.39.       .106 CP SECONDS COMPILATION TIME
09.11.39.MAP,PART.
09.11.39.LGO.
09.11.42.      END  TSTFUNC
09.11.42.       .017 CP SECONDS EXECUTION TIME
09.11.42.OP  00001216 WORDS - FILE OUTPUT , DC 40
09.11.42.MS     3504  WORDS  (   17920 MAX USED)
09.11.42.CPA         .259 SEC.           .259 ADJ.
09.11.42.IO         1.573 SEC.          1.573 ADJ.
09.11.42.CM        29.455 KWS.          1.809 ADJ.
09.11.42.SS                             3.642
09.11.42.PP         6.800 SEC.     DATE 06/27/78
09.11.42.EJ  END OF JOB, SH
```

6   2   4

PARAMETERS

COBOL

THE CALL:

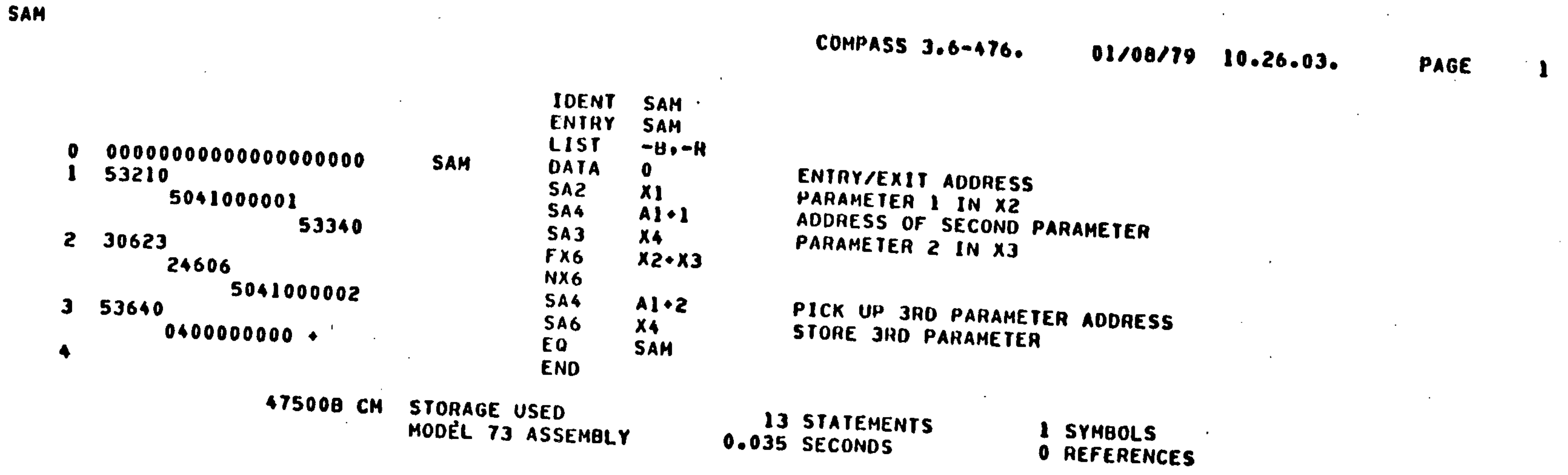
ENTER SUB, PARM1, PARM2, PARM3

THE CODE GENERATED:

* IDENTICAL TO FORTRAN EXTENDED

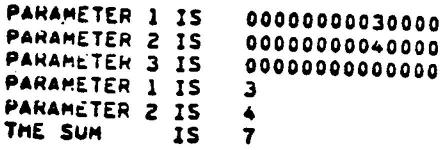
```
               AO 0112                                    COBOL    V4.7 L485  01/08/79  10.25.58.      PAGE      1

               IDENTIFICATION DIVISION.

               PROGRAM-ID. PAPAM.

               ENVIRONMENT DIVISION.
               CONFIGURATION SECTION.
               SOURCE-COMPUTER. 6600.
               OBJECT-COMPUTER. 6600.
               SPECIAL-NAMES.
                   OUTPUT-C IS PRINT.

               DATA DIVISION.
               WORKING-STORAGE SECTION.
               77  PARM1 USAGE IS COMPUTATIONAL-2 VALUE IS 3.
               77  PARM2 USAGE IS COMPUTATIONAL-2 VALUE IS 4.
               77  PARM3 USAGE IS COMPUTATIONAL-2 VALUE IS 0.
               77  PARM4 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
               77  PARM5 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
               77  PARM6 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
```

```
     PAPAM        AO 0112                                    COBOL    V4.7 L485  01/08/79  10.25.58.      PAGE      2

               PROCEDURE DIVISION.

               BEGIN.
                   DISPLAY "1PARAMETER 1 IS   " PARM1 UPON PRINT.
                   DISPLAY " PARAMETER 2 IS   " PARM2 UPON PRINT.
                   DISPLAY " PARAMETER 3 IS   " PARM3 UPON PRINT.

               ENTER SAM, PARM2, PARM1, PARM3.
                   MOVE PARM1 TO PARM4.
                   MOVE PARM2 TO PARM5.
                   MOVE PARM3 TO PARM6.
                   DISPLAY " PARAMETER 1 IS   " PARM4 UPON PRINT.
                   DISPLAY " PARAMETER 2 IS   " PARM5 UPON PRINT.
                   DISPLAY " THE SUM     IS   " PARM6 UPON PRINT.
               STOP RUN.
PAPAM   LENGTH IS  000151
 061000B SCM USED
```

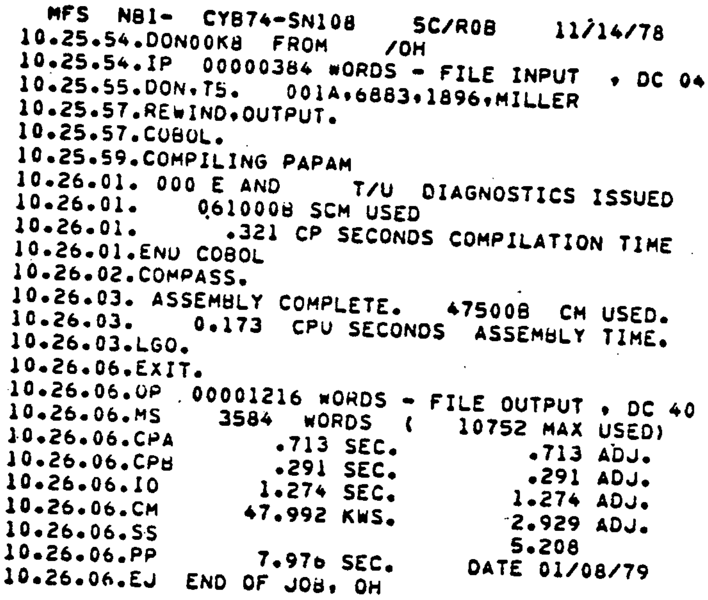


```
SAM
                                                      COMPASS 3.6-476.     01/08/79  10.26.03.       PAGE      1

                                      IDENT   SAM
                                      ENTRY   SAM
                                      LIST    -B,-R
  0  00000000000000000000     SAM     DATA    0                ENTRY/EXIT ADDRESS
  1  53210                            SA2     X1               PARAMETER 1 IN X2
           5041000001                 SA4     A1+1             ADDRESS OF SECOND PARAMETER
                     53340            SA3     X4               PARAMETER 2 IN X3
  2  30623                            FX6     X2+X3
           24606                      NX6
               5041000002             SA4     A1+2             PICK UP 3RD PARAMETER ADDRESS
  3  53640                            SA6     X4               STORE 3RD PARAMETER
           0400000000 +               EQ      SAM
  4                                   END

                    47500B CM  STORAGE USED                     13 STATEMENTS        1 SYMBOLS
                               MODEL 73 ASSEMBLY             0.035 SECONDS           0 REFERENCES
```

```
LOAD MAP - OUTPUT#                                        CYBER LOADER 1.4-485      01/08/79  10.26.05.     PAGE     1

FWA OF THE LOAD           111
LWA+1 OF THE LOAD        5335

TRANSFER ADDRESS -- CENT.00          412

PROGRAM ENTRY POINTS --       OUTPUT#          412


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK        ADDRESS   LENGTH    FILE       DATE     PROCSSR VER LEVEL  HARDWARE   COMMENTS

OUTPUT#         111      116    LGO       01/08/79 COBOL    4.7 L485       I
INPUT#          227      116    LGO       01/08/79 COBOL    4.7 L485       I
/D.COMON/       345       24
D.COMON         371        0    LGO       01/08/79 COBOL    4.7 L485       I
/PAPAM#/        371       21
PAPAM           412      151    LGO       01/08/79 COBOL    4.7 L485       I
SAM             563        4    LGO       01/08/79 COMPASS 3.6 476
C=COBIO         567      751 SL-COBOL     11/16/78 COMPASS 3.6 485
D=CMM15        1540       10 SL-COBOL     11/16/78 COMPASS 3.6 485
D=USE          1550      211 SL-COBOL     11/16/78 COMPASS 3.6 485
D=DSPLY        1761      437 SL-COBOL     11/16/78 COMPASS 3.6 485
/STP.END/      2420        1
C=NSTAN        2421      153 SL-COBOL     11/16/78 COMPASS 3.6 485
D=TENS         2574       25 SL-COBOL     11/16/78 COMPASS 3.6 485
D=CVBD         2621      136 SL-COBOL     11/16/78 COMPASS 3.6 485
D=BN           2757       21 SL-COBOL     11/16/78 COMPASS 3.6 485
D=MOVE         3000      133 SL-COBOL     11/16/78 COMPASS 3.6 485
D=TRUBL        3133      121 SL-COBOL     11/16/78 COMPASS 3.6 485
D=ZNTAB        3254       12 SL-COBOL     11/16/78 COMPASS 3.6 485
D=TENDP        3266       73 SL-COBOL     11/16/78 COMPASS 3.6 485
/FDL.COM/      3361       14
FDL.RES        3375      211 SL-SYSLIB    11/02/78 COMPASS 3.6 485                 FAST DYNAMIC LOADER RESIDENT.
FDL.MMI        3606      222 SL-SYSLIB    11/02/78 COMPASS 3.6 485                 FDL MEMORY MANAGER INTERFACE.
CPU.SYS        4030       40 SL-SYSLIB    11/15/78 COMPASS 3.6 476                 PROCESS SYSTEM REQUEST.
CMF.ALF        4070      160 SL-SYSLIB    11/15/78 COMPASS 3.6 485                 CMM V1.1 - ALLOCATE FIXED.
CMF.CSF        4250        6 SL-SYSLIB    11/15/78 COMPASS 3.6 485                 CMM V1.1 - CHANGE SPECS FIXED.
CMM.FFA        4256       14 SL-SYSLIB    11/15/78 COMPASS 3.6 485                 CMM V1.1 - FIXED FREE ALGORITHM.
CMF.FRF        4272       36 SL-SYSLIB    11/15/78 COMPASS 3.6 485                 CMM V1.1 - FREE FIXED.
CMM.R          4330      214 SL-SYSLIB    11/15/78 COMPASS 3.6 485                 CMM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF        4544       22 SL-SYSLIB    11/15/78 COMPASS 3.6 485                 CMM V1.1 - SHRINK AT LWA FIXED.
CTL$RM         4566      433 SL-SYSLIB    11/16/78 COMPASS 3.6 485                 CRM CONTROLLING ROUTINE.
ERR$RM         5221       25 SL-SYSLIB    11/16/78 COMPASS 3.6 485                 CRM ERROR PROCESSOR ENTRY.
LIST$RM        5246       67 SL-SYSLIB    11/16/78 COMPASS 3.6 485                 CRM - ALLOCATE SPACE FOR LIST OF FILES


      .316 CP SECONDS               21600B CM STORAGE USED                     9 TABLE MOVES
```



```
PARAMETER 1 IS   00000000030000
PARAMETER 2 IS   00000000040000
PARAMETER 3 IS   00000000000000
PARAMETER 1 IS   3
PARAMETER 2 IS   4
THE SUM     IS   7
```



```
  MFS  NB1-  CYB74-SN108    SC/R0B    11/14/78
10.25.54.DON00K8  FROM    /OH
10.25.54.IP  00000384 WORDS - FILE INPUT  , DC 04
10.25.55.DON,T5.   001A,6883,1896,MILLER
10.25.57.REWIND,OUTPUT.
10.25.57.COBOL.
10.25.59.COMPILING PAPAM
10.26.01. 000 E AND     T/U  DIAGNOSTICS ISSUED
10.26.01.    061000B SCM USED
10.26.01.      .321 CP SECONDS COMPILATION TIME
10.26.01.ENU COBOL
10.26.02.COMPASS.
10.26.03. ASSEMBLY COMPLETE.   47500B  CM USED.
10.26.03.    0.173  CPU SECONDS  ASSEMBLY TIME.
10.26.03.LGO.
10.26.06.EXIT.
10.26.06.OP  00001216 WORDS - FILE OUTPUT , DC 40
10.26.06.MS    3584  WORDS  (   10752 MAX USED)
10.26.06.CPA         .713 SEC.          .713 ADJ.
10.26.06.CPB         .291 SEC.          .291 ADJ.
10.26.06.IO         1.274 SEC.         1.274 ADJ.
10.26.06.CM        47.992 KWS.         2.929 ADJ.
10.26.06.SS                            5.208
10.26.06.PP         7.976 SEC.    DATE 01/08/79
10.26.06.EJ  END OF JOB, OH
```

```
IDENTIFICATION DIVISION.

PROGRAM-ID. ASTERISK.

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.
SOURCE-COMPUTER. 6600.
OBJECT-COMPUTER. 6600.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT OUTFILE ASSIGN TO OUTPUT.

DATA DIVISION.

FILE SECTION.
FD OUTFILE
    LABEL RECORDS ARE OMITTED
    DATA RECORD IS XXXX.
01 XXXX PICTURE X(136).
WORKING-STORAGE SECTION.
77 ONE PICTURE 9(10) USAGE IS COMPUTATIONAL-1 VALUE IS 1.
77 FOUR PICTURE 9(10) USAGE IS COMPUTATIONAL-1 VALUE IS 4.
77 TEN PICTURE 9(10) USAGE IS COMPUTATIONAL-1 VALUE IS 10.
77 XWORD PICTURE 9(10) USAGE IS COMPUTATIONAL-1.
77 XPOS PICTURE 9(10) USAGE IS COMPUTATIONAL-1.
01 HDR.
    02 FILLER PICTURE X(22) VALUE ≠ THE STAR WAS IN WORD ≠.
    02 WORD PICTURE ZZZZZZZZ9.
    02 FILLER PICTURE X(20) VALUE ≠ CHARACTER POSITION ≠.
    02 POS PICTURE ZZZZZZZZ9.
01 A PICTURE X(20) VALUE ≠ FOR TABLE A≠.
01 B PICTURE X(20) VALUE ≠ FOR TABLE B≠.
01 C PICTURE X(20) VALUE ≠ FOR TABLE C≠.
01 BUFF.
    02 AA PICTURE X(10) OCCURS 10 TIMES.
    02 BB PICTURE X(10) OCCURS 10 TIMES.
    02 CC PICTURE X(10) OCCURS 10 TIMES.
    ASTERIS         AO 0113                              COBOL    V4.7 L470* 06/27/78  09.07.03.      PAGE       2


       00039             PROCEDURE DIVISION.
       00040             ROLLOUT.
       00041                 OPEN OUTPUT OUTFILE.
       00042                 MOVE #*AB12# TO AA(10).
       00043                 MOVE #AB*# TO CC(1).
       00044                 ENTER SEARCH, AA(1),TEN,XWORD,XPOS.
       00045                 MOVE XWORD TO WORD.
       00046                 MOVE XPOS TO POS.
       00047                 WRITE XXXX FROM A.
       00048                 WRITE XXXX FROM HDR.
       00049                 ENTER SEARCH, BB(1),ONE,XWORD,XPOS.
       00050                 MOVE XWORD TO WORD.
       00051                 MOVE XPOS TO POS.
       00052                 WRITE XXXX FROM B.
       00053                 WRITE XXXX FROM HDR.
       00054                 ENTER SEARCH, CC(1),FOUR,,XWORD,XPOS.
       00055                 MOVE XWORD TO WORD.
       00056                 MOVE XPOS TO POS.
       00057                 WRITE XXXX FROM C.
       00058                 WRITE XXXX FROM HDR.
       00059                 CLOSE OUTFILE.
       00060                 STOP RUN.
  ASTERIS LENGTH IS  000342
   061200B SCM USED
```

```
                                                         COMPASS 3.5-470.     06/27/78  09.07.06.      PAGE       1
      SEARCH
      STORAGE ALLOCATION.

              ADDRESS   LENGTH             BINARY CONTROL CARDS.

                    0       22              IDENT SEARCH
                   22                  END

                          BLOCKS    TYPE       ADDRESS    LENGTH

                          PROGRAM*  LOCAL            0        21
                          LITERALS* LOCAL           21         1

                          ENTRY POINTS.

                          SEARCH            0*
```

```
SEARCH                                                          COMPASS 3.5-470.     06/27/78  09.07.06.       PAGE       2

                                                 IDENT SEARCH
                                                 ENTRY SEARCH
                                          *      CALL SEARCH(ARRAY NAME,ARRAY LENGTH,WORD NUMBER,CHARACTER POSITION)
                              47         STAR    EQU    47B
    0  77777777777777777777              SEARCH  DATA   -0              ENTRY/EXIT
    1  53210                                     SA2    X1              ADDRESS OF ARRAY IN A2, 1ST WORD IN X2
            5011000001                           SA1    A1+1            INDEX TO NEXT PARAMETER(2ND PARAMETER)
                      53310                      SA3    X1              ADDRESS OF ARRAY LENGTH IN A3, VALUE IN X3
    2  5011000001                                SA1    A1+1            INDEX TO NEXT PARAMETER (3RD PARAMETER)
                 53410                           SA4    X1              ADDRESS OF WORD POSITION IN A4
    3  5011000001                                SA1    A1+1            INDEX TO NEXT PARAMETER (4TH PARAMETER)
                 53510                           SA5    X1              ADDRESS OF CHARACTER POSITION IN A5
    4  6130000000                                SB3    0               CURRENT WORD INDEX IN B3
    5  6120000011                        OUTLOOP SB2    9               REVERSE CHARACTER POSITION INDEX TO B2
                 54123                           SA1    A2+B3           LOAD INDEXED ELEMENT OF ARRAY
    6  7170000006                        INLOOP  SX7    6               X7 = NUMBER OF BITS PER CHARACTER
                 76020                           SX0    B2              CHARACTER INDEX TO X0
                      42007                      IX0    X0*X7           ACTUAL SHIFT COUNT TO X0
    7  63400                                     SB4    X0              NOW TO B4
          7170000077                             SX7    77B            .LOWER 6 BIT MASK TO X7
                      23641                      AX6    B4,X1           SHIFT X1 RIGHT B4, PUT IT IN X6
   10  11667                                     BX6    X6*X7           LOGICAL PRODUCT ALL BUT LOWER SIX BITS
          7266777730                             SX6    X6-STAR         SUBTRACT CODE FOR CHARACTER BEING LOOKED FOR
   11  0306000016 +                              ZR     X6,FOUND        GO TO FOUND IF EQUAL TO ZERO
                 6122777776                      SB2    B2-1            REDUCE SHIFT INDEX
   12  0620000006 +                              GE     B2,INLOOP       IF NOT NEGATIVE, CONTINUE INNER LOOP
                 6133000001                      SB3    B3+1            INDEX TO NEXT WORD
   13  63430                                     SB4    X3              ARRAY LENGTH FOR COMPARISON
            0534000005 +                         NE     B3,B4,OUTLOOP  IF NOT EQUAL, RESET AND CONTINUE OUTER LOOP
   14  5110000021 +                              SA1    =-0             GET -0 FOR NOFIND FLAG ***  NO FIND  ***
                 10611                           BX6    X1              TRANSFER X6 TO X1
                      54640                      SA6    A4              STORE INTO ARRAY WORD NUMBER, SHOW NOT FOUND
   15  54650                                     SA6    A5              SET CHARACTER POSITION EQUAL TO MINUS ZERO
            0400000000 +                         EQ     SEARCH          EXIT---NO MATCH
   16  7163000001                        FOUND   SX6    B3+1            WORD NUMBER, (1TO MAXIMUM) TO X6
                 54640                           SA6    A4              SET IN PARAMETER (WORD POSITION)
   17  6160000012                                SB6    10              CHARACTER POSITION 1 TO 10
                 77662                           SX6    B6-B2           B6 - SHIFT INDEX = CHARACTER POSITION
                      54650                      SA6    A5              STORE PARAMETER (CHARACTER POSITION)
   20  0400000000 +                              EQ     SEARCH          EXIT----FOUND
   22                                        END

                  47300B SCM STORAGE USED                41 STATEMENTS          5 SYMBOLS
                             MODEL 174 ASSEMBLY       0.055 SECONDS            12 REFERENCES


SEARCH                                                          COMPASS 3.5-470.     06/27/78  09.07.06.       PAGE       3
SYMBOLIC REFERENCE TABLE.

FOUND           16      PROGRAM*          2/24        2/35 L
INLOOP           6      PROGRAM*          2/16 L      2/26
OUTLOOP          5      PROGRAM*          2/14 L      2/29
SEARCH           0      PROGRAM*          2/02 E      2/05 L      2/34        2/40
STAR            47                        2/04 D      2/23
```

```
LOAD MAP - OUTPUT≡                                        CYBER LOADER 1.4-470       06/27/78  09.07.10.      PAGE      1


FWA OF THE LOAD            111
LWA+1 OF THE LOAD         7033

TRANSFER ADDRESS -- CENT.00          376

PROGRAM ENTRY POINTS --      OUTPUT≡          376


PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK         ADDRESS   LENGTH    FILE       DATE      PROCSSR VER LEVEL  HARDWARE   COMMENTS

OUTPUT≡           111      116    LGO        06/27/78 COBOL   4.7 L470*       I
INPUT≡            227      116    LGO        06/27/78 COBOL   4.7 L470*       I
/D.COMON/         345       24
D.COMON           371        0    LGO        06/27/78 COBOL   4.7 L470*       I
/ASTERI≡/         371        5
ASTERIS           376      342    LGO        06/27/78 COBOL   4.7 L470*       I
SEARCH            740       22    LGO        06/27/78 COMPASS 3.5 470
DDSVR             762      110 SL-COBOL      05/07/78 COMPASS 3.5 470
D=CKETL          1072       15 SL-COBOL      05/07/78 COMPASS 3.5 470
C=COBIO          1107      751 SL-COBOL      05/07/78 COMPASS 3.5 470
D=CRMIS          2060       10 SL-COBOL      05/07/78 COMPASS 3.5 470
D=SOIO           2070      445 SL-COBOL      05/07/78 COMPASS 3.5 470
D=USE            2535      211 SL-COBOL      05/07/78 COMPASS 3.5 470
/STP.END/        2746        1
C=NSTAN          2747      155 SL-COBOL      05/07/78 COMPASS 3.5 470
D=TENS           3124       25 SL-COBOL      05/07/78 COMPASS 3.5 470
D=CVRD           3151      137 SL-COBOL      05/07/78 COMPASS 3.5 470
D=BN             3310       22 SL-COBOL      05/07/78 COMPASS 3.5 470
D=ED             3332      456 SL-COBOL      05/07/78 COMPASS 3.5 470
D=COLSQ          4010      114 SL-COBOL      05/07/78 COMPASS 3.5 470
D=RCDCM          4124      116 SL-COBOL      05/07/78 COMPASS 3.5 470
C=NVI            4242      100 SL-COBOL      05/07/78 COMPASS 3.5 470
D=TRUBL          4342      121 SL-COBOL      05/07/78 COMPASS 3.5 470
D=TENDP          4463       73 SL-COBOL      05/07/78 COMPASS 3.5 470
CMF.ALF          4556      160 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CMM V1.1 - ALLOCATE FIXED.
CMF.CSF          4736        6 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CMM V1.1 - CHANGE SPECS FIXED.
CMM.FFA          4744       14 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CMM V1.1 - FIXED FREE ALGORITHM.
CMF.FRF          4760       36 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CMM V1.1 - FREE FIXED.
CMM.R            5016      213 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CMM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF          5231       22 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CMM V1.1 - SHRINK AT LWA FIXED.
CTL$LBL          5253       25 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CRM CONTROLLER - LABEL PROCESSING.
CTL$RM           5300      601 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CRM CONTROLLING ROUTINE.
CTL$WR           6101       31 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CRM CONTROLLER - WEOX, REWIND
ERR$RM           6132       25 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CRM ERROR PROCESSOR ENTRY.
LIST$RM          6157       66 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CRM - ALLOCATE SPACE FOR LIST OF FILES
/FDL.COM/        6245       14
FDL.RES          6261      211 SL-SYSLIB     05/16/78 COMPASS 3.5 470                   FAST DYNAMIC LOADER RESIDENT.
FDL.MMI          6472      222 SL-SYSLIB     05/16/78 COMPASS 3.5 470                   FDL MEMORY MANAGER INTERFACE.
SYS.RM           6714       40 SL-SYSLIB     05/16/78 COMPASS 3.5 470                   PROCESS SYSTEM REQUEST.
CTL$SKP          6754       57 SL-SYSLIB     05/02/78 COMPASS 3.5 470                   CRM CONTROLLER - SKIP PHYSICAL/FILE.
```

```
      LOAD MAP - OUTPUT=                                  CYBER LOADER 1.4-470       06/27/78  09.07.10.       PAGE      2

         .149 CP SECONDS               23200B CM STORAGE USED                          10 TABLE MOVES



 FOR TABLE A
 THE STAR WAS IN WORD          10 CHARACTER POSITION          1
 FOR TABLE B
 THE STAR WAS IN WORD           0 CHARACTER POSITION          0
 FOR TABLE C
 THE STAR WAS IN WORD           1 CHARACTER POSITION          3
```

```
   MFF   NB2-  CYB175-SN1    4LB7/R6B  05/15/78
 09.06.59.DON0052  FROM    /SH
 09.06.59.IP  00000704 WORDS - FILE INPUT  , DC 04
 09.06.59.DON.     PSO,027B,72CT011A,MILLER
 09.07.02.COBOL.
 09.07.04.COMPILING ASTERIS
 09.07.05. 000 E AND     I/U  DIAGNOSTICS ISSUED
 09.07.05.    061200B SCM USED
 09.07.05.       .182 CP SECONDS COMPILATION TIME
 09.07.05.END COBOL
 09.07.05.COMPASS.
 09.07.07. ASSEMBLY COMPLETE.   47300B SCM USED.
 09.07.07.     0.109  CPU SECONDS  ASSEMBLY TIME.
 09.07.07.LGO.
 09.07.12.EXIT.
 09.07.12.OP  0000204B WORDS - FILE OUTPUT , DC 40
 09.07.12.MS    3584  WORDS  (   14336 MAX USED)
 09.07.12.CPA        .500 SEC.         .500 ADJ.
 09.07.12.IO        1.816 SEC.        1.816 ADJ.
 09.07.12.CM       45.907 KWS.        2.801 ADJ.
 09.07.12.SS                          5.119
 09.07.12.PP       10.551 SEC.     DATE 06/27/78
 09.07.12.EJ  END OF JOB, SH
```

QUIZZES

1. PLACE THE FOLLOWING NUMBERS IN FLOATING POINT PRESENTATION. ALL NUMBERS ARE NOW DECIMAL.

   5×2^{6}

   5×2^{-6}

   $-(5 \times 2^{6})$

   $-(5 \times 2^{-6})$

   15

   -10

   $-(1 \times 2^{64})$

   1×2^{-1}

2. CONVERT THE FOLLOWING NUMBERS FROM FLOATING POINT FORM TO $(C \times 2^{E})$ FORM.

   3066 0000 0000 0000 0032

   1512 0000 0000 0000 0260

   5710 7777 7777 7777 7755

   7733 7777 7777 7777 7707

3. CHANGE THE FOLLOWING FLOATING POINT NUMBERS TO OCTAL (INTEGER FRACTION FORM).

   1722 5000 0000 0000 0000

   1730 5006 0000 0000 0000

   1721 6400 0000 0000 0000

   1675 7000 0000 0000 0000

   1720 6550 0000 0000 0000

   1747 4000 0007 0000 0000

   6055 1777 7777 7777 7777

   6055 1477 7777 7777 7777

   6071 1777 7777 7777 7777

   1736 5123 4567 7700 0000

1. WRITE AN INSTRUCTION THAT WILL CHANGE THE PROGRAM SEQUENCE TO ONE OF FIVE SUBROUTINES, DEPENDING UPON THE CONTENTS OF THE B1 INDEX REGISTER. THE INITIAL ADDRESS OF ALL THE ROUTINES IS CALLED "BASEADDR".

2. WRITE AN INSTRUCTION THAT TESTS THE CONTENTS OF X0 FOR ZERO, AND IF THE CONDITION IS MET THE PROGRAM BRANCHES TO "DONE".

3. WRITE AN INSTRUCTION THAT CHECKS B1 AGAINST B1 TO SEE IF THEY ARE EQUAL. IF THEY ARE EQUAL, BRANCH TO A LOCATION CALLED "MATCH".

4. WRITE AN INSTRUCTION TO PERFORM THE LOGICAL PRODUCT OF THE OPERANDS IN X6 AND X4, AND SEND THE RESULT TO X6.

5. WRITE AN INSTRUCTION THAT SHIFTS THE CONTENTS OF THE X1 REGISTER LEFT 12_{10} (14_8) PLACES.

6. WRITE AN INSTRUCTION THAT UNPACKS THE FLOATING POINT QUANTITY IN X1, PLACING THE EXPONENT IN B1, AND THE COEFFICIENT IN X1.

7. WRITE AN INSTRUCTION WHICH PERFORMS THE FLOATING POINT ADDITION WITH THE OPERANDS IN X4 AND X5, PLACING THE NEW RESULT BACK IN X6. THE OPERANDS ARE IN FLOATING POINT FORMAT.

8. WRITE AN INSTRUCTION TO PERFORM THE INTEGER DIFFERENCE OF THE 60-BIT OPERANDS IN X4 AND X5. (X5 - X4). THE RESULT IS TO BE PLACED IN X6.

9. WRITE AN INSTRUCTION THAT FORMS A MASK IN THE UPPER 30-BITS OF THE X0 REGISTER.

10. WRITE AN INSTRUCTION TO LOAD THE X1 REGISTER WITH A 60-BIT WORD LOCATED IN MEMORY. A1 CONTAINS THE REQUIRED ADDRESS $+200_8$.

11. WRITE AN INSTRUCTION TO STORE THE VALUE IN X7 IN THE LOCATION IN MEMORY CALLED "SYMTAB" INDEXED BY NO B-REGISTER.

12. WRITE AN INSTRUCTION TO LOAD FROM MEMORY THE X2 REGISTER WITH THE 18-BIT ADDRESS IN X1 MODIFIED BY AN INDEX REGISTER B7.

13. WRITE AN INSTRUCTION TO BUMP THE INDEX REGISTER B4 BY A VALUE OF 120_8.

14. WRITE AN INSTRUCTION TO SET THE INDEX REGISTER B1 TO THE VALUE IN A0 DECREMENTED BY THE VALUE IN B4.

15. WRITE AN INSTRUCTION WHICH WILL COMPUTE THE DIFFERENCE OF B1 AND B2 AND PLACE THE RESULT IN X0.

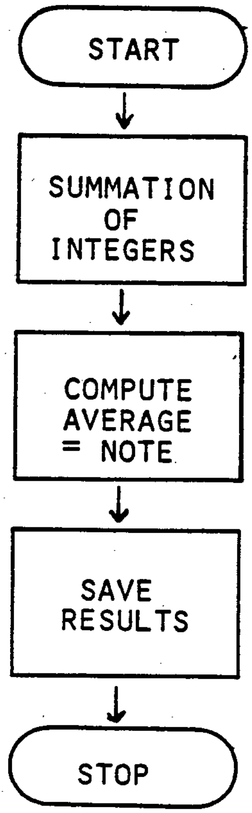
PROJECTS

Problem Set 1A

Average Integers

Write a program to find the average of 10 integers.

Use the following flowchart.

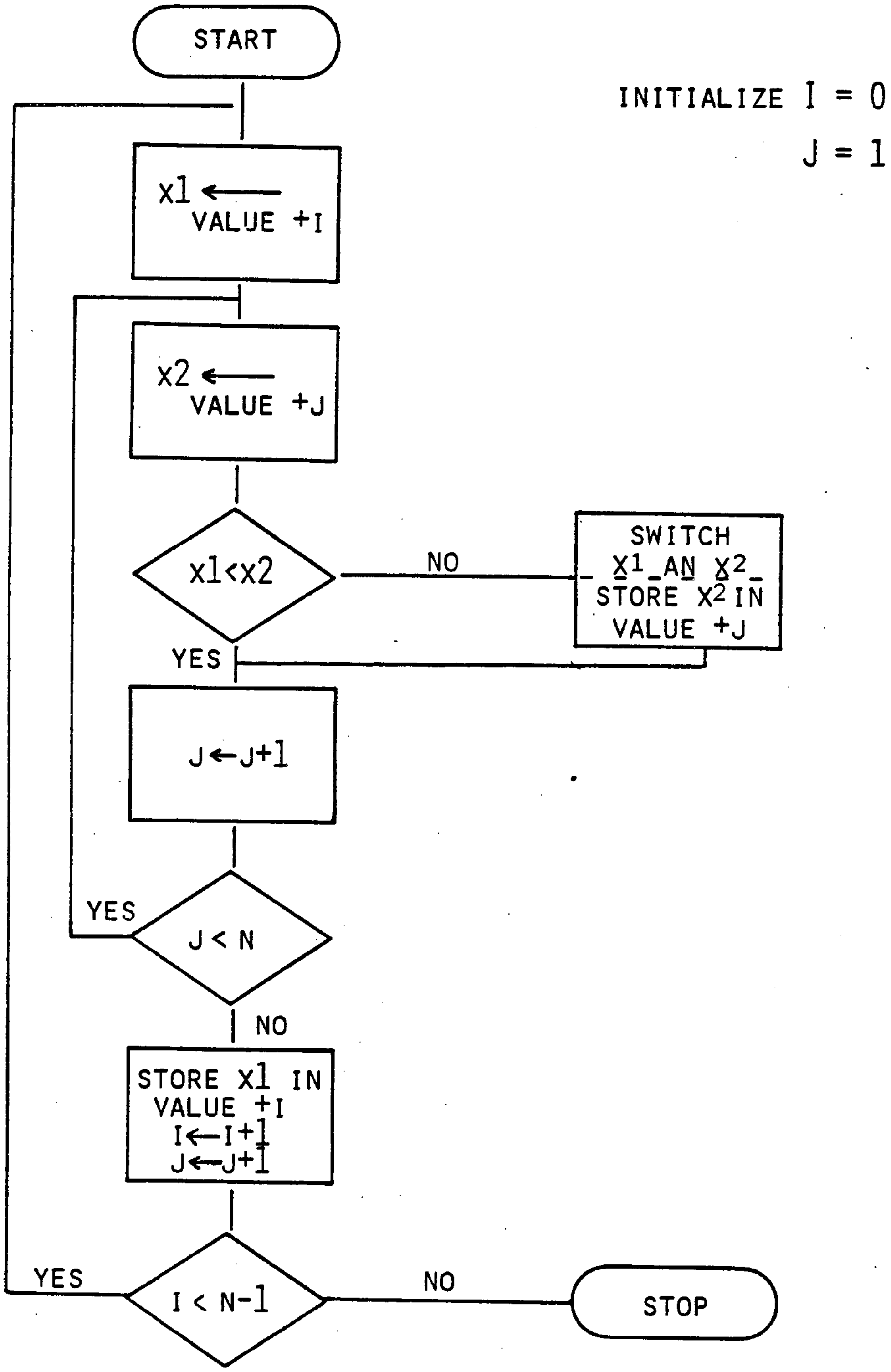


NOTE: IF A AND B ARE INTEGER VALUES, THEN B/A = Q WITH A REMAINDER OF B - (Q=A).

PROCEDURE: PACK AND NORMALIZE A AND B.
COMPUTE B/A = Q.
UNPACK AND SHIFT Q; NOW Q IS INTEGER QUOTIENT.
PACK AND NORMALIZE Q AND A.
COMPUTE Q = A.
UNPACK AND SHIFT THE PRODUCT AND SUBTRACT FROM B.
THIS IS REMAINDER.

Problem 1B

Write a program to sort N integers into ascending order. The following is a flow chart to accomplish this using the selection sort method.



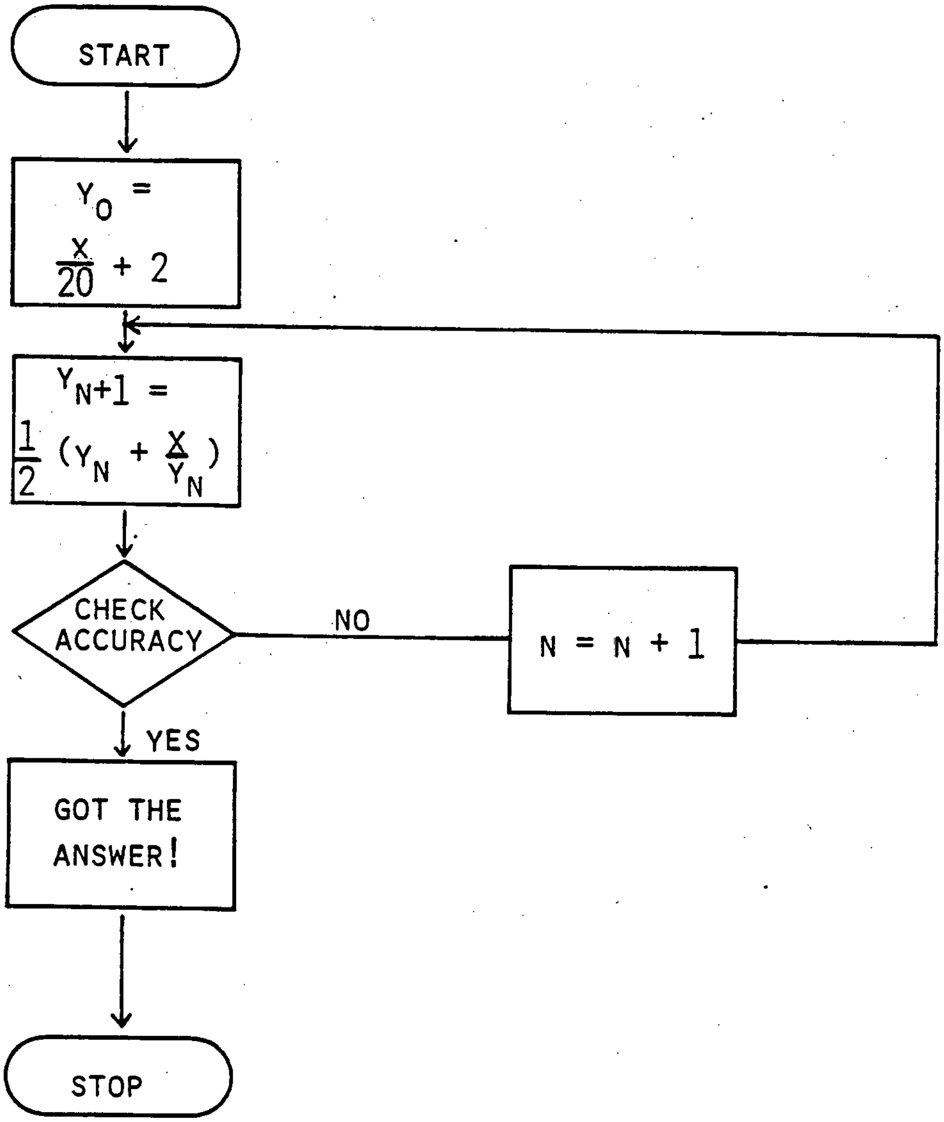
PROBLEM 1C SQUARE ROOT

WRITE YOUR OWN SQUARE ROOT SUBROUTINE USING THE NEWTON ITERATION....

$$Y_{N+1} = \frac{1}{2}\left(Y_N + \frac{X}{Y_N}\right) \text{ WITH } Y_0 = \frac{X}{20.0} + 2.0$$

THE INPUT VALUE IS X WHILE THE RETURNED VALUE IS YOUR APPROXIMATION TO THE SQUARE ROOT OF X. OBTAIN A RELATIVE ACCURACY OF....

$$\frac{1}{2}\,(10^{-9}) \qquad (Y_{N+1} - Y_N)/Y_{N+1}$$



Problem Set 2

CHARACTER DATA, DEFERRED SYMBOLS, SUBROUTINE.

WRITE A SUBROUTINE TO CHECK IF THE FIRST CHARACTER OF A WORD IS A DOLLAR SIGN OR NOT.

THE WORD TO CHECK CAN BE PLACED IN A X REGISTER.

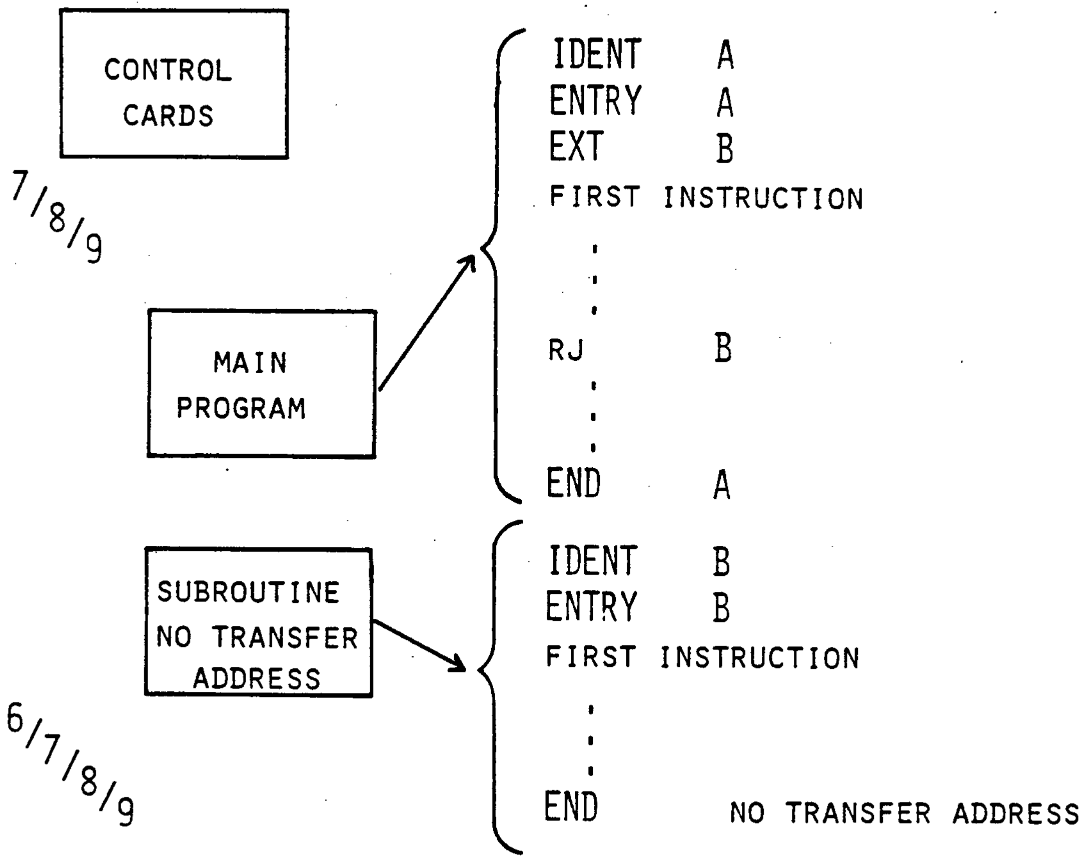
THE X REGISTER SOULD BE SET TO ZERO IF CHARACTER 1 IS A $.

IF NOT, PUT A SHORT MESSAGE IN THE X REGISTER.

THE MAIN PROGRAM SHOULD PLACE THE WORD TO BE CHECKED IN AN X REGISTER, CALL THE SUBROUTINE AND PLACE THE ERROR MESSAGE IN STORAGE WHEN CONTROL IS RETURNED.

USE THE FOLLOWING COMPASS FEATURES SOMEWHERE IN YOUR PROGRAM: CHARACTER DATA, DEFERRED SYMBOLS (=S, =X) AND RETURN JUMP.

DECK STRUCTURE



CONTROL CARDS

7/8/9

MAIN PROGRAM

```
IDENT    A
ENTRY    A
EXT      B
FIRST INSTRUCTION
   .
   .
   .
RJ       B
   .
   .
   .
END      A
```

SUBROUTINE NO TRANSFER ADDRESS

6/7/8/9

```
IDENT    B
ENTRY    B
FIRST INSTRUCTION
   .
   .
   .
END            NO TRANSFER ADDRESS
```

Problem Set 3 COMMON STORAGE

Set up a blank common block of 12 locations.

Set up a common block labeled TABLE which contains display code for the number 0-7.

List 12 octal one-digit integers in your main program.

Transfer the 12 numbers to blank common.

Go to a subroutine.

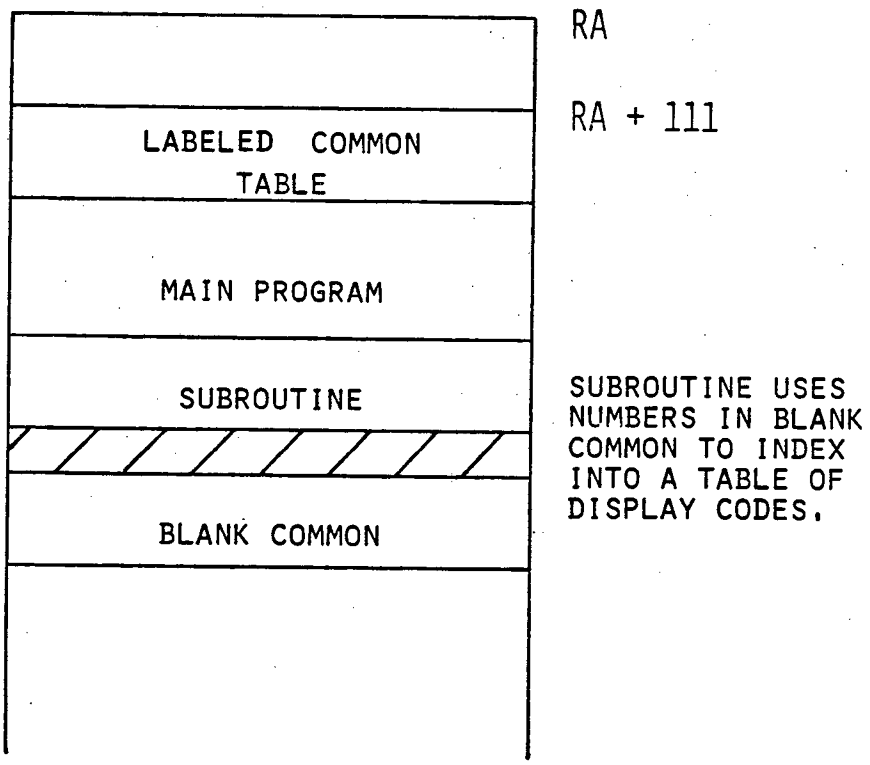
In the subroutine, fetch the number from blank common, look up the display code in TABLE.

Store the display code back in blank common.

Return to the main program and stop.

For added challenge, use display code for 0-7.

List 12 decimal integers (2 or more digits) and find the display code.


RA
RA + 111
LABELED COMMON
TABLE
MAIN PROGRAM
SUBROUTINE
BLANK COMMON
SUBROUTINE USES NUMBERS IN BLANK COMMON TO INDEX INTO A TABLE OF DISPLAY CODES.

Problem Set 4A MACRO

Write the following macro:

Purpose: To compute n factorial exactly (integer).
n = n(n-l)(n-2)···(2)(1)

Parameters: P1 = n

P2 = result register

Procedure: Define n with a SET (1 < n < 18)

Call the macro, storing the result at the specified x register.

Test Program: SET n

Call the macro

reSET n

Call the macro again

Store the result

Stop

PROBLEM SET 4B

PURPOSE: WRITE A MACRO TO MOVE A MESSAGE FROM A USER BUFFER TO A COMMON BLOCK BUFFER.

PARAMETERS: P1 = BEGINNING OF USER BUFFER.

P2 = LENGTH OF USER BUFFER.

P3 = BEGINNING OF BUFFER IN COMMON STORAGE.

PROCEDURE: COMPUTE THE LENGTH OF THE BUFFER.

CALL THE MACRO.

PROBLEM 5A

REWRITE THE INTEGER MULTIPLY MACRO AS AN OPDEF.

CALL IT.

PROBLEM 5B

WRITE AN OPDEF WHICH WILL COMPARE TWO CORE LOCATIONS FOR EQUALITY.

OPDEF CALL:

```
        TAG        CE        Q1, Q2, Q3
```

WRITE A TEST PROGRAM WHICH CALLS THE MACRO.

PROBLEM 6A CONDITIONAL OPERATORS, MACROS

WRITE THE FOLLOWING MACRO:

PURPOSE: COUNT THE NUMBER OF BITS SET IN EACH WORD OF A TABLE. SAVE AND RESTORE ALL X REGISTERS IF REQUESTED.

PARAMETERS: P1 = LENGTH OF THE TABLE

P2 = STARTING LOCATION OF THE TABLE

P3 = SAVE REGISTER FLAG

= S SAVE AND RESTORE ALL X REGISTERS

= BLANK UNNECESSARY TO SAVE REGISTERS

PROCEDURE: CHECK P3. IF P3 = S SAVE THE REGISTERS.

COUNT THE BITS IN THE TABLE WORDS, RETURNING THE COUNT TO THE TABLE. (LET THE TABLE BE LOCATED IN COMMON STORAGE IF YOU PREFER).

RETURN THE REGISTERS IF NECESSARY AND RETURN TO THE USER.

OPTIONAL COMPLICATION: SAVE ALL REGISTERS ON REQUEST (NOT JUST THE X REGISTERS). THIS IS VERY DIFFICULT.

PROBLEM 6B

MACRO, CONDITIONAL (MICRO USAGE OPTIONAL)

WRITE THE FOLLOWING MACRO:

PURPOSE: TO DOUBLE, SQUARE, OR HALVE A NUMBER. (FLOATING POINT)

PARAMETER: P1 = X REGISTER CONTAINING THE NUMBER TO BE OPERATED ON.

P2 = CORE LOCATION WHERE RESULT IS TO BE PLACED.

P3 = OPERATION FLAG

P3 = D DOUBLE CONTENTS OF X REGISTER

P3 = S SQUARE CONTENTS OF X REGISTER

P3 = H DIVIDE CONTENTS OF X REGISTER BY 2.

PROCEDURE: CHECK P3 USING CONDITIONALS.

GENERATE THE APPROPRIATE RESULT AND STORE IT.

TEST PROGRAM: THE TEST PROGRAM CAN BE VERY SIMPLE. SET UP A RESULT LOCATION, LOAD THE X REGISTER, CALL THE MACRO AND STOP.

OPTIONAL COMPLICATION: PERMIT P3 TO BE D, S, H, OR ANY COMBINATION. THE RESULTS CAN BE PLACED IN P2, P2 + 1 AND P2 + 2 IF NECESSARY. XK, XK + 1 AND XK + 2 CAN CONTAIN OPERANDS. MICROS MUST BE USED TO CHECK P3.

EVALUATION FORM

Course/Seminar Name ____________________ **Date of Attendance From** ________ **To** ________

Instructor ______________________________ **Location** ______________________________

Please place a rating in the box fcr each area and then add comments explaining your rating.

Rating Key

Excellent	*5*
Very Good	*4*
Good	*3*
Fair	*2*
Poor	*1*

The Course/Seminar

* How well did the course/seminar cover the stated objectives? □

* To what degree will the course/seminar be helpful in improving on-the-job performance? □

* To what extent were the handout materials and visuals helpful in aiding your understanding of the topic? □

* What is your overall rating of the organization and content of the course/seminar? □

The Instructor

* How do you rate the instructor's knowledge of the material and ability to answer questions? □

* How effective was the instructor in presenting the material in an understandable manner? □

* How effective was the instructor in generating and sustaining interest in the course/seminar? □

* How do you rate the instructor's responsiveness to the needs of participants? □

* What is your overall rating of the instructor? □

The Facilities

* How do you rate the appropriateness of the facilities to the topic and means of presentation? □

* To what extent were the facilities comfortable, well-lighted and heated or cooled? □

* How convenient was the location of the facility? □

EVALUATION FORM

Page 2

General Comments

* What changes in the course/seminar would you make if you were the instructor?

* Would you recommend this course/seminar to others in your company or department? Why?

* Please list colleagues or associates who should receive advance notices of similar courses/seminars.

1) Name ______________________

Organization ______________________

Address ______________________

____

Bus. Tel. No. ______________________

2) Name ______________________

Organization ______________________

Address ______________________

____

Bus. Tel. No. ______________________

3) Name ______________________

Organization ______________________

Address ______________________

____

Bus. Tel. No. ______________________

4) Name ______________________

Organization ______________________

Address ______________________

____

Bus. Tel. No. ______________________

* Should this course be offered at your company site? If so, who should be contacted to manage it?

* If we may use your comments in future descriptions of the course/seminar, please sign below.

Signature ______________________

(Optional)

PARTICIPANT INFORMATION FORM

In order for our seminars/courses to be most effective, they need to take into account the characteristics, needs and objectives of the people who attend them. The information asked for below will assist us in keeping our presentations relevant to the participants and in developing and scheduling new presentations that will meet participant needs. Please complete this form and leave it with the presenter at the next break.

Seminar/Course Title ____________________

Name ____________________

Title ____________________

Business Address ____________________

____

____

Date of Presentation ____________________

Field or Type of Business ____________________

Years of Experience ____________________

Supervisor's Title ____________________

Last professional degree ____________________

List your three primary objectives in attending this seminar.

1. ____________________
2. ____________________
3. ____________________

Will this course/seminar be credited toward certification/training requirements? ____________

Rank in order of importance in your choice of this seminar session.

Instructor ______ **Date** ______ **Location** ______ **Employer's Preference** ______

Previous courses/seminars attended relating to this topic.

1. ____________________
2. ____________________
3. ____________________

Topics for additional courses/seminars in which you would be interested.

1. ____________________
2. ____________________
3. ____________________

PARTICIPANT INFORMATION FORM

Page 2

What trade journals/magazines do you regularly read or subscribe to in order to keep abreast in your profession?

1. ______________________________________

2. ______________________________________

3. ______________________________________

How did you become aware of this course/seminar?

Schedule/Catalogue ________,

Direct Mail Brochure ________,

Recommendations of Supervisor ________,

Recommendation of Colleague ________,

Corporate Training Department ________,

Other ______________________________________ .

COMMENT SHEET

MANUAL TITLE: CYBER CP COMPASS

PUBLICATION NO.: DA3020-1 REVISION: D

NAME: ____________________

COMPANY: ____________________

STREET ADDRESS: ____________________

CITY: ____________ STATE: ____________ ZIP CODE: ____________

This form is not intended to be used as an order blank. Control Data Corporation welcomes your evaluation of this manual. Please indicate any errors, suggested additions or deletions, or general comments below (please include page number references).

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND STAPLE

CUT ALONG LINE

AA3419 REV. 4/79 PRINTED IN U.S.A.

STAPLE

STAPLE

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 8241 MINNEAPOLIS, MINN.

POSTAGE WILL BE PAID BY

CONTROL DATA CORPORATION
National Coordinator
Bloomington Facility (MNA02B)
5001 West 80th Street
Bloomington, Minnesota 55437
Attn: Curtis Vicha

CUT ALONG LINE

FOLD

FOLD

B instructions move 60 bits

CONTROL DATA SEMINARS

an education service of
CONTROL DATA CORPORATION

CORPORATE HEADQUARTERS
P.O. BOX 0
MINNEAPOLIS, MINNESOTA 55440