

Dick N

Course DA3020-1
CYBER CP COMPASS

STUDENT HANDOUT

PROPRIETARY NOTICE

The ideas and designs set forth in this document are the property of Control Data Corporation and are not to be disseminated, distributed, or otherwise conveyed to third persons without the express written permission of Control Data Corporation.

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

1982
©COPYRIGHT CONTROL DATA CORPORATION 1982
All Rights Reserved.

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.

1	INTRO AND PRETEST	REVIEW	DATA ITEMS	ERROR EXITS	CONDITIONAL ASSEMBLY
2	HARDWARE AND JOB OVERVIEW	FLOATING POINT ARITHMETIC			
3	SYMBOL DEFINITION	FLOATING POINT INSTRUCTIONS	PSEUDO OPERATIONS	MACRO'S	PARAMETER PASSING
4	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					

COURSE CHART - CP COMPASS

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:

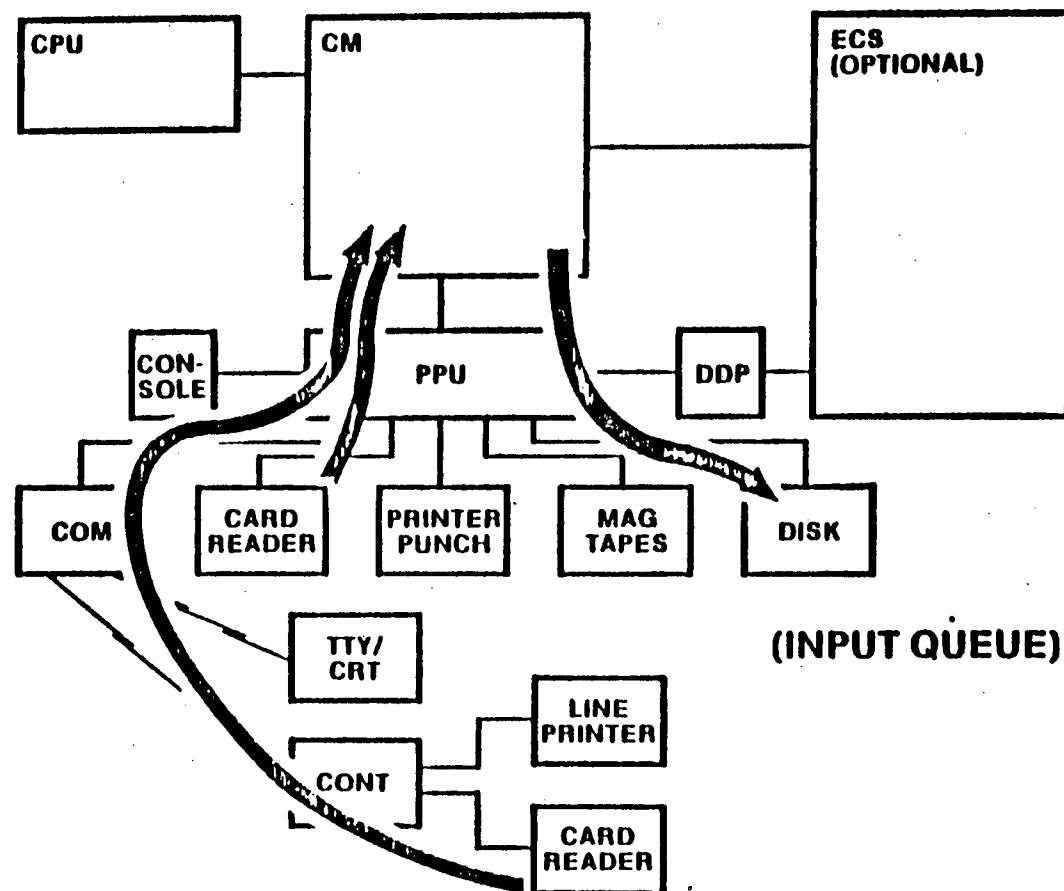
NONE

OBJECTIVES:

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

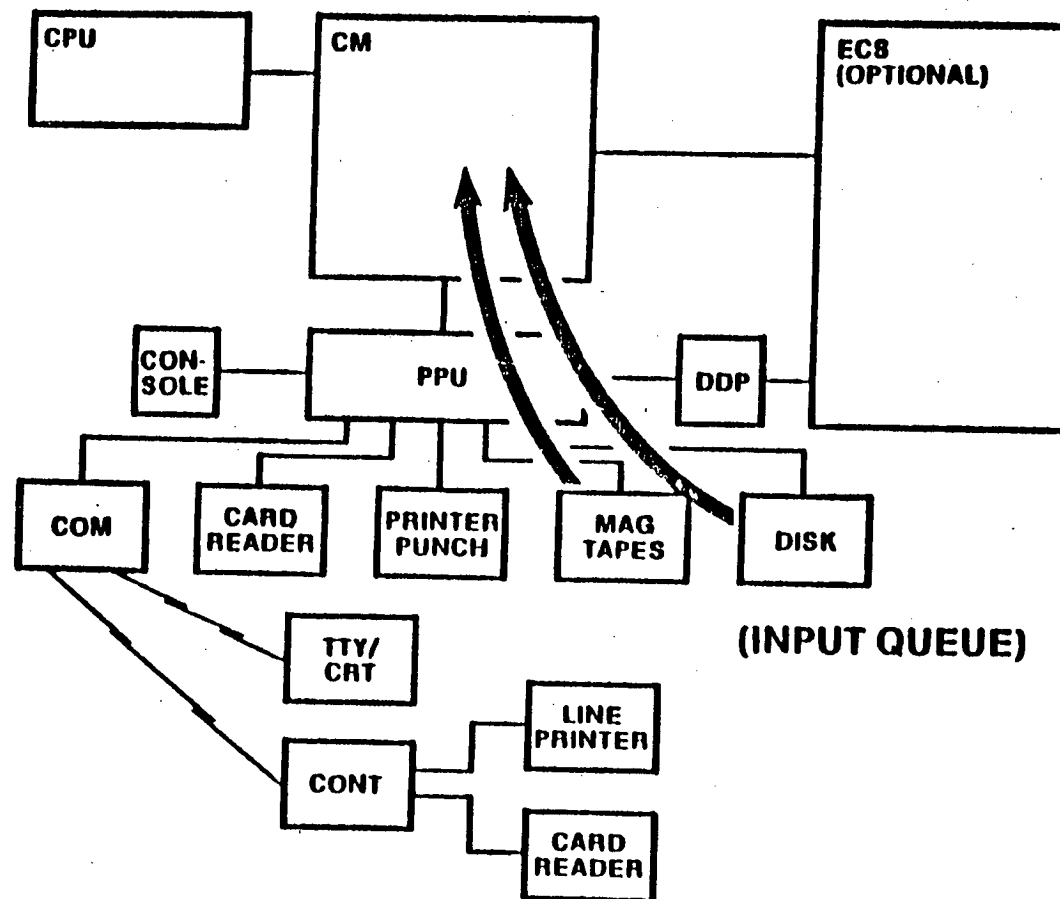
1. DESCRIBE THE FIVE PHASES OF A JOB IN THE SYSTEM.
2. EXPLAIN THE CONTROL POINT.
3. LIST THE COMPONENTS AND THEIR FUNCTIONS OF THE CENTRAL PROCESSOR.
4. EXPLAIN HOW AN INSTRUCTION IS OPERATED ON BY THE COMPONENTS OF THE CENTRAL PROCESSOR.

CDC CYBER 170 SERIES



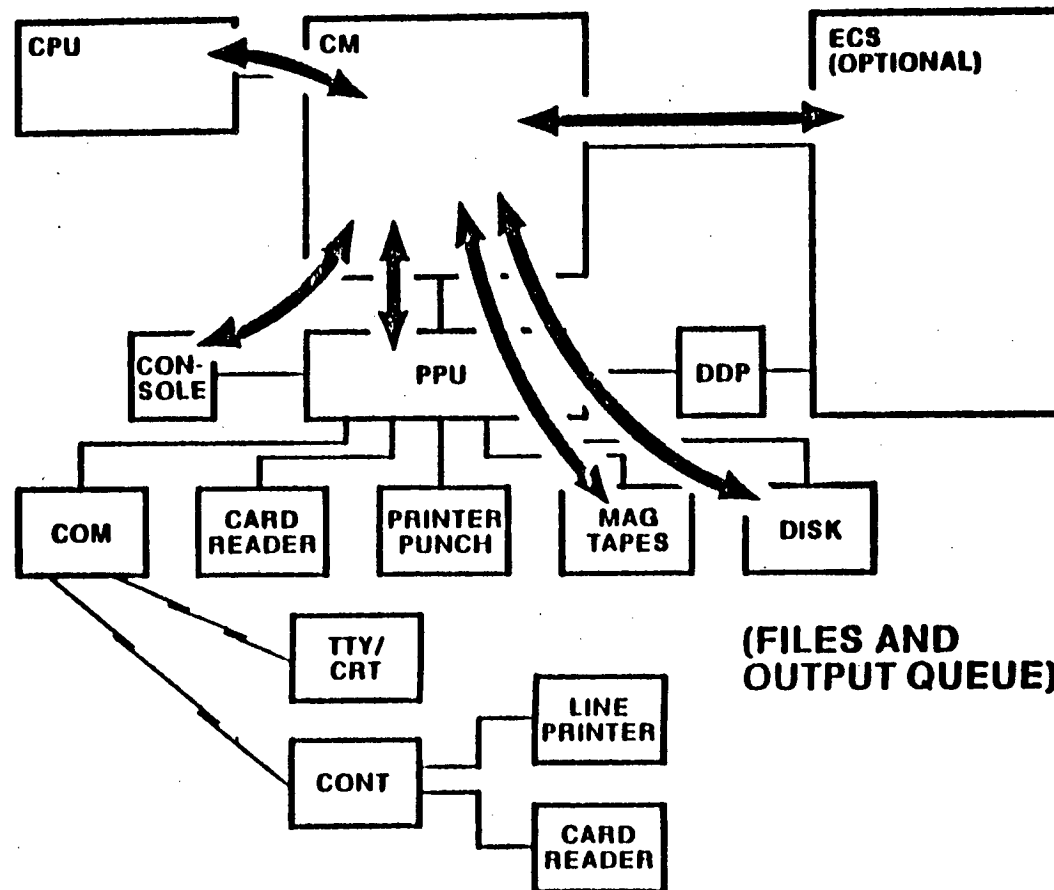
BATCH JOB INPUT

CDC CYBER 170 SERIES

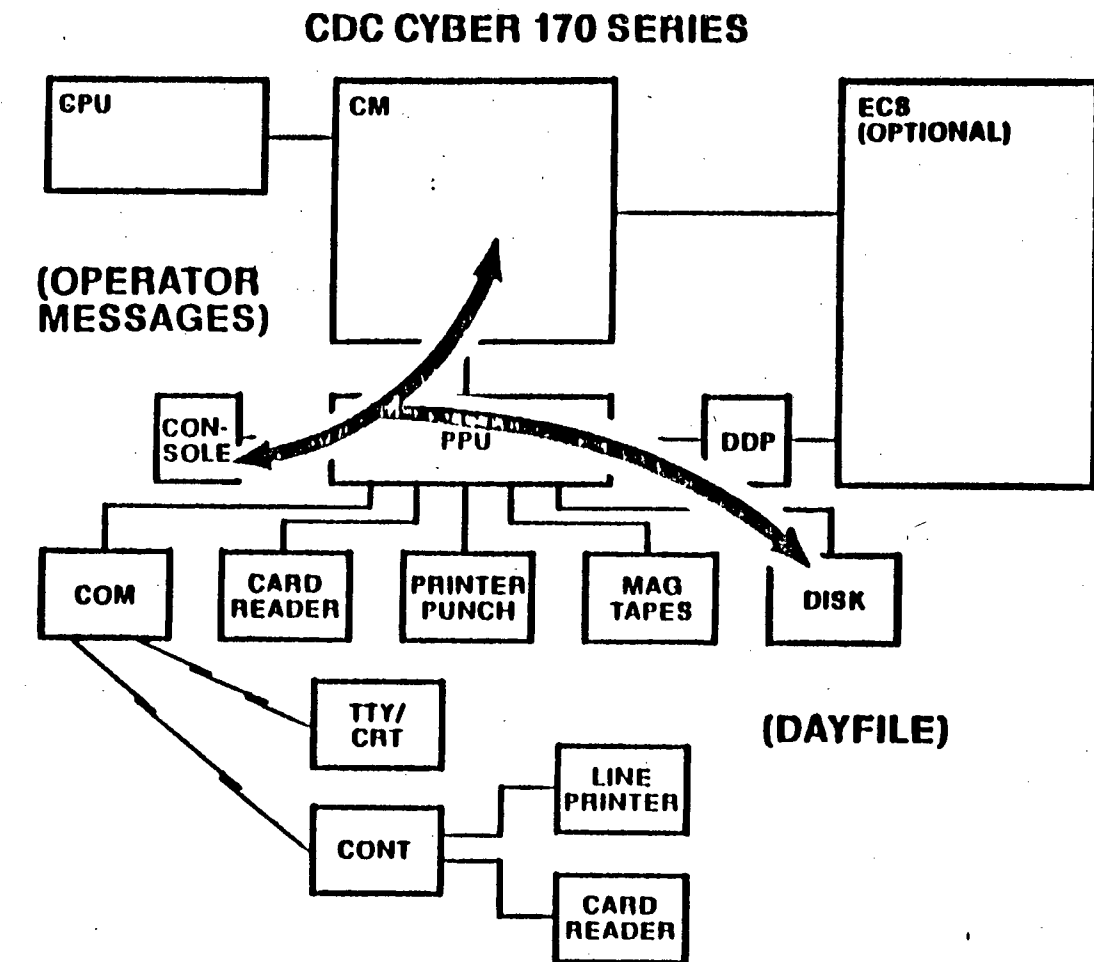


LOADING A JOB FOR EXECUTION

CDC CYBER 170 SERIES

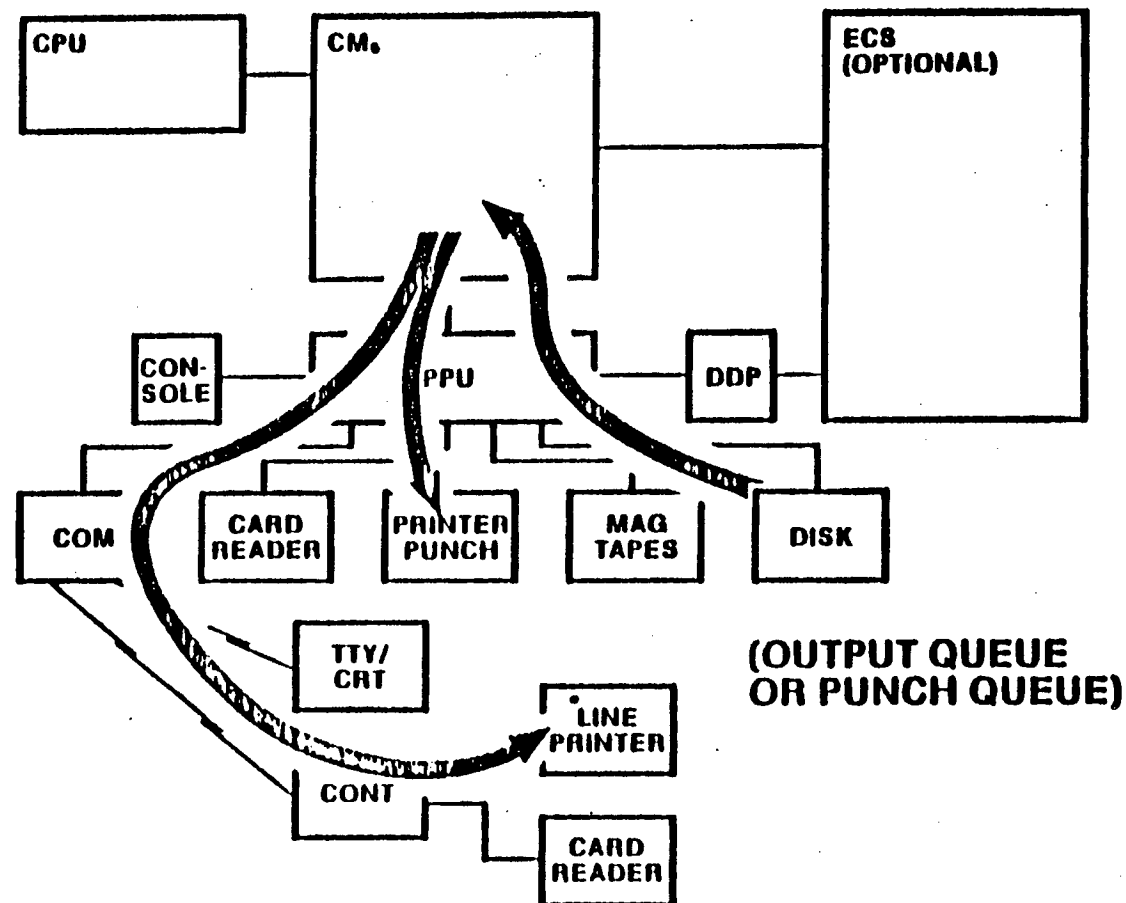


USER JOB EXECUTION

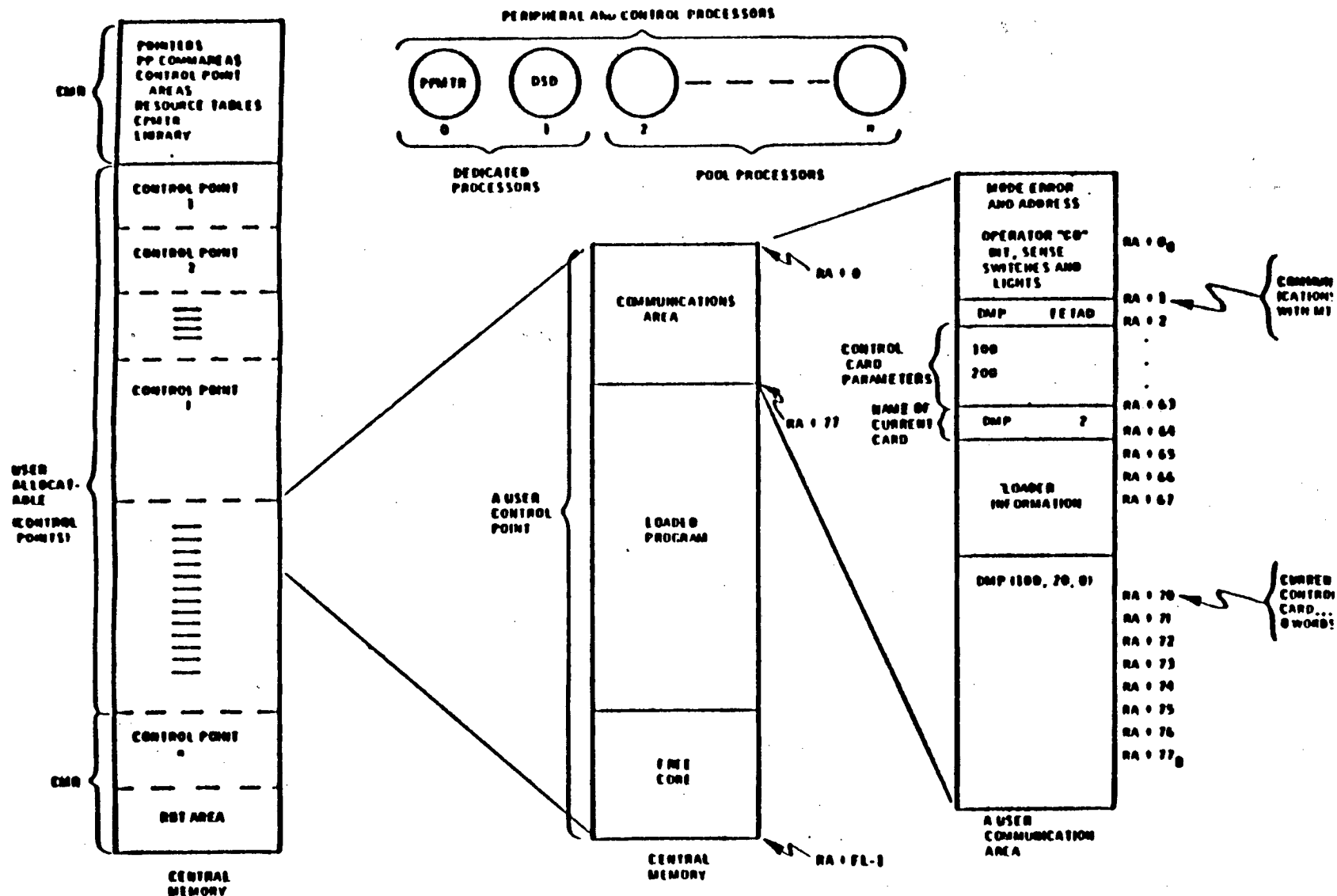


OPERATOR AND DAYFILE MESSAGES

CDC CYBER 170 SERIES



BATCH OUTPUT



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.

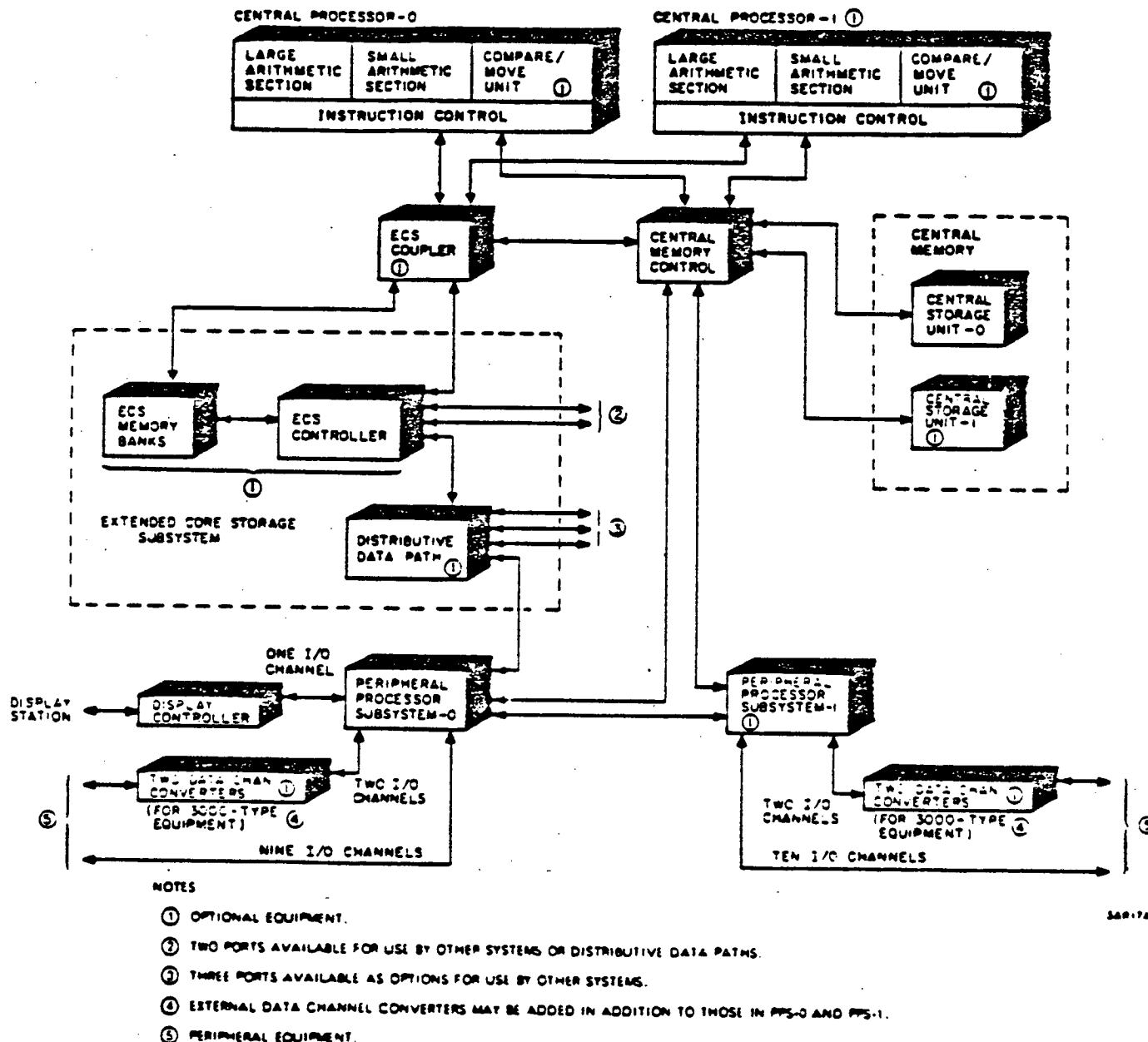


Figure 1-9. Model 171 Computer System

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.

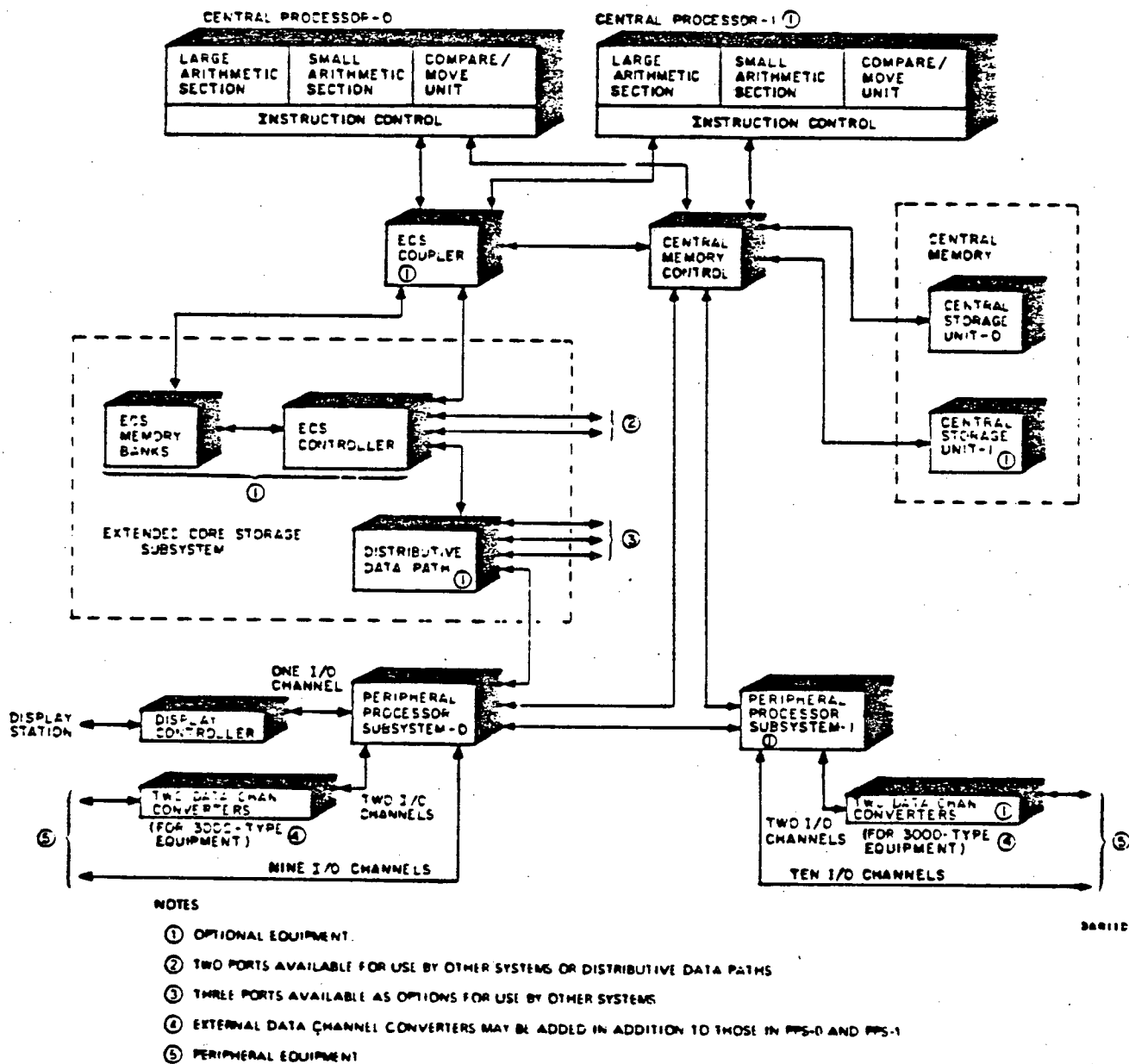


Figure 1-10. Model 172 Computer System

MODEL 173 SYSTEM

CP provides faster operation.

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

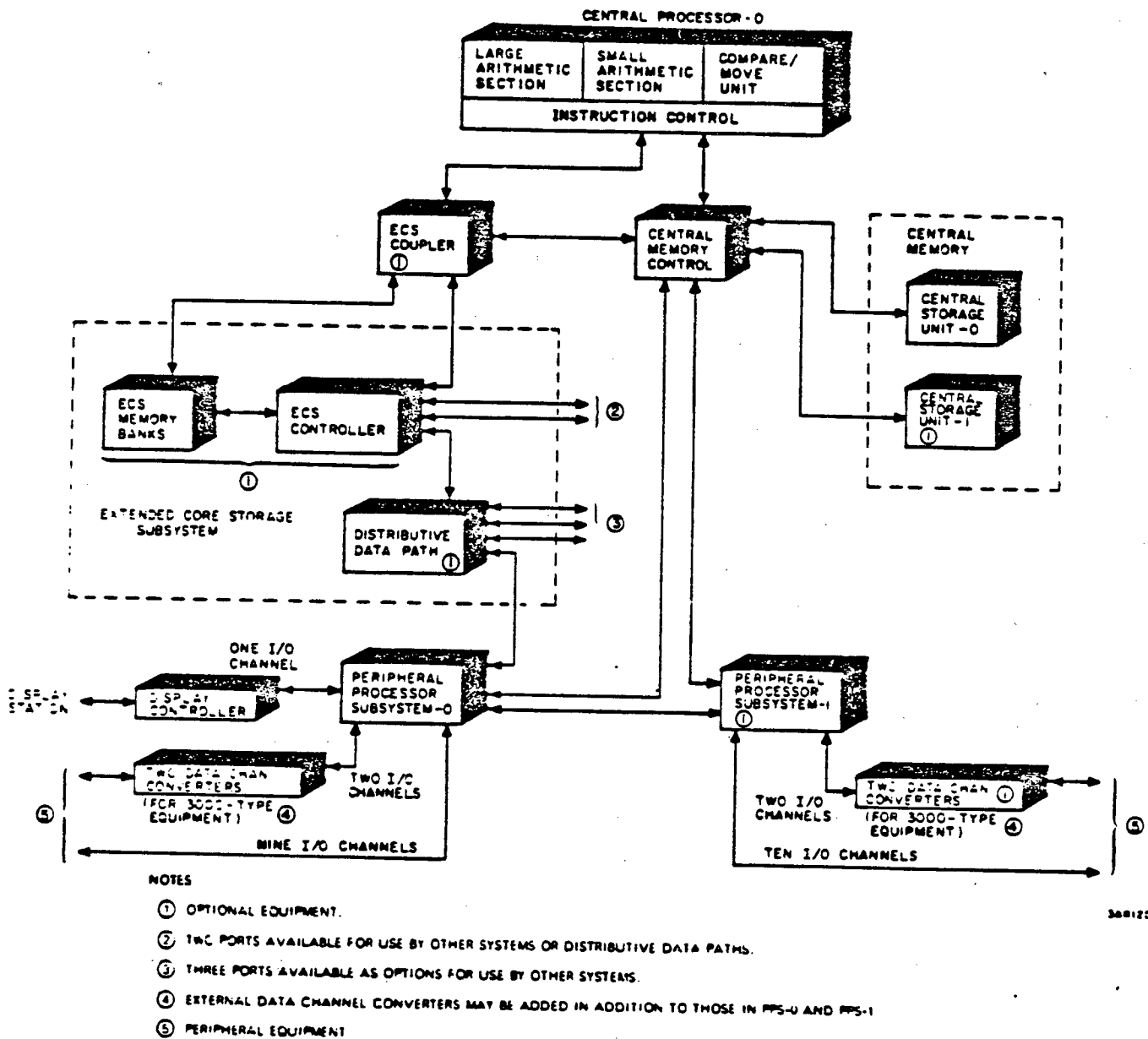


Figure 1-11. Model 173 Computer System

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.

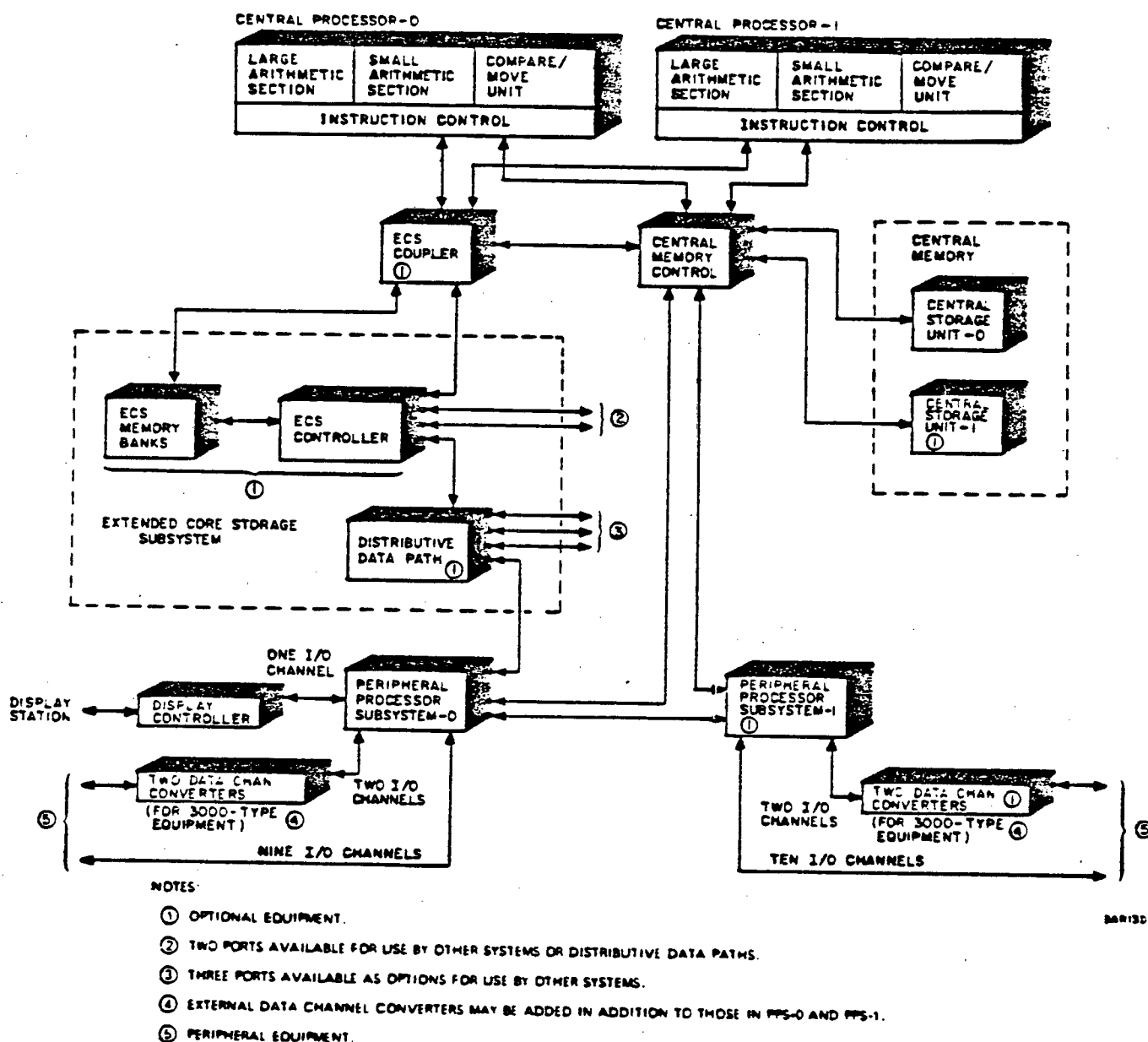


Figure 1-12. Model 174 Computer System

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 cen-

tral 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.

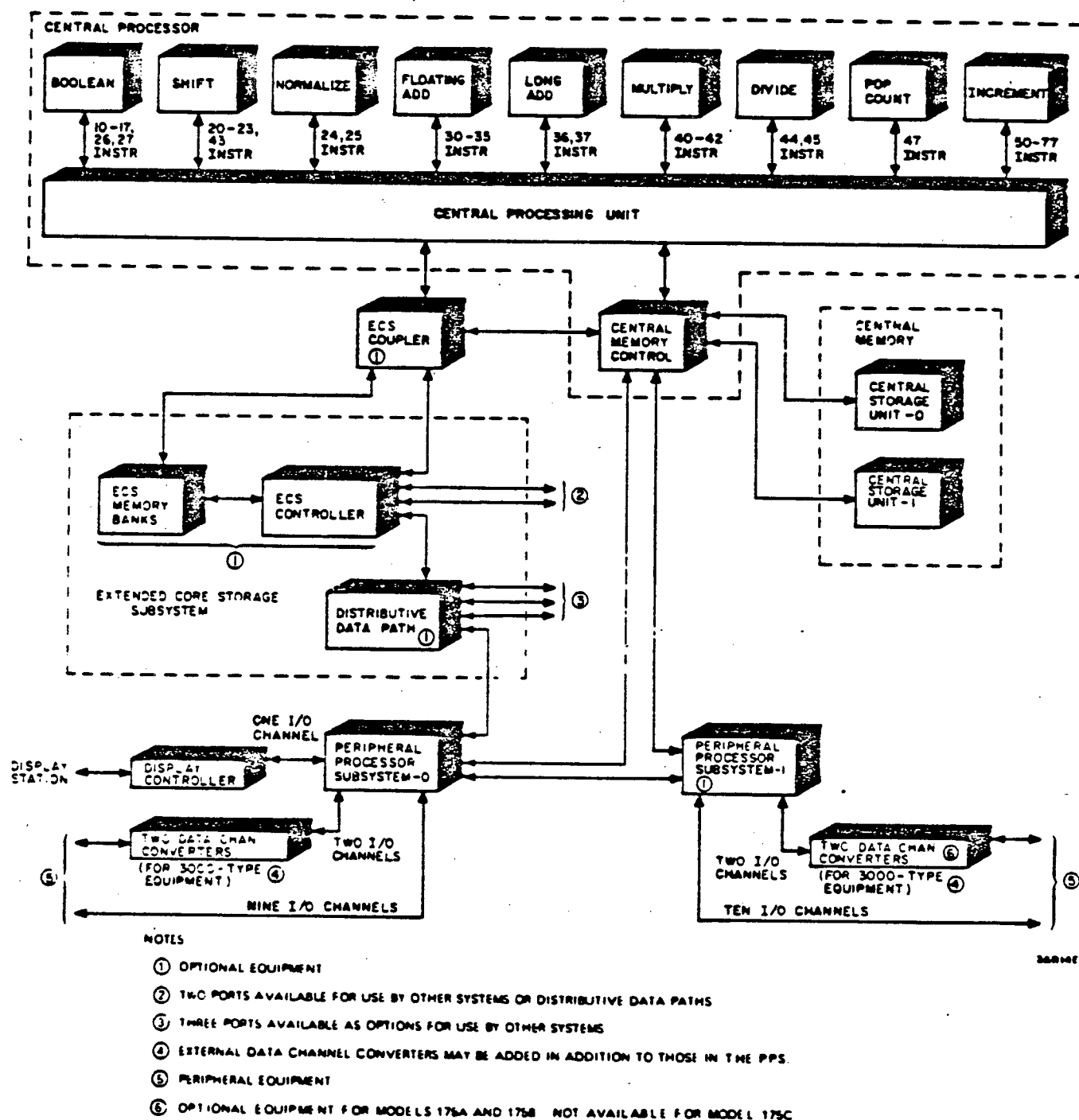


Figure 1-13. Model 175 Computer System

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

The diagram illustrates the architecture of the CDC 6600 computer. At the top, the **CENTRAL PROCESSOR** is shown as a row of nine functional blocks: **BOOLEAN**, **SHIFT**, **NORMALIZE**, **FLOATING ADD**, **LONG ADD**, **MULTIPLY**, **DIVIDE**, **POP COUNT**, and **INCREMENT**. Each block is associated with a specific instruction range: 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; and 50-57 INSTR, respectively. These are connected to the **CENTRAL PROCESSING UNIT**. Below this, the **LARGE CORE MEMORY EXTENSION AND CONTROL** (labeled 1) and **CENTRAL MEMORY AND CONTROL** (labeled 2) are connected to the CPU. A central **I/O MUX** (multiplexer) is connected to the CPU, the memory units, and the peripheral subsystems. **PERIPHERAL PROCESSOR SUBSYSTEM-0** (labeled 3) is connected to the I/O MUX and has a **DISPLAY STATION** and **DISPLAY CONTROLLER** connected to it. It also has a **ONE I/O CHANNEL** connection to the I/O MUX. **PERIPHERAL PROCESSOR SUBSYSTEM-1** (labeled 4) is connected to the I/O MUX and has a **LOGIC SCANNER** connected to it. It also has a **ONE I/O CHANNEL** connection to the I/O MUX. Both subsystems have **DATA CHANNEL CONVERTERS** (labeled 5) connected to them. Subsystem-0 has a **ONE I/O CHANNEL** connection to the I/O MUX, while Subsystem-1 has a **TWO I/O CHANNELS** connection. The diagram also shows **FOUR PERIPHERAL PROCESSOR UNITS** (labeled 6) connected to the I/O MUX, with **ONE I/O CHANNEL (PER PPU)** and **FOUR I/O CHANNELS PER PPU** connections. **LOGIC SCANNER** (labeled 7) is connected to the I/O MUX. **DATA CHANNEL CONVERTERS** (labeled 8) are connected to the I/O MUX, with **ONE I/O CHANNEL (FOR 300-TYPE EQUIPMENT)** and **TEN I/O CHANNELS** connections. **NOTES** at the bottom provide details on optional equipment and expandability.

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 LCM 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:

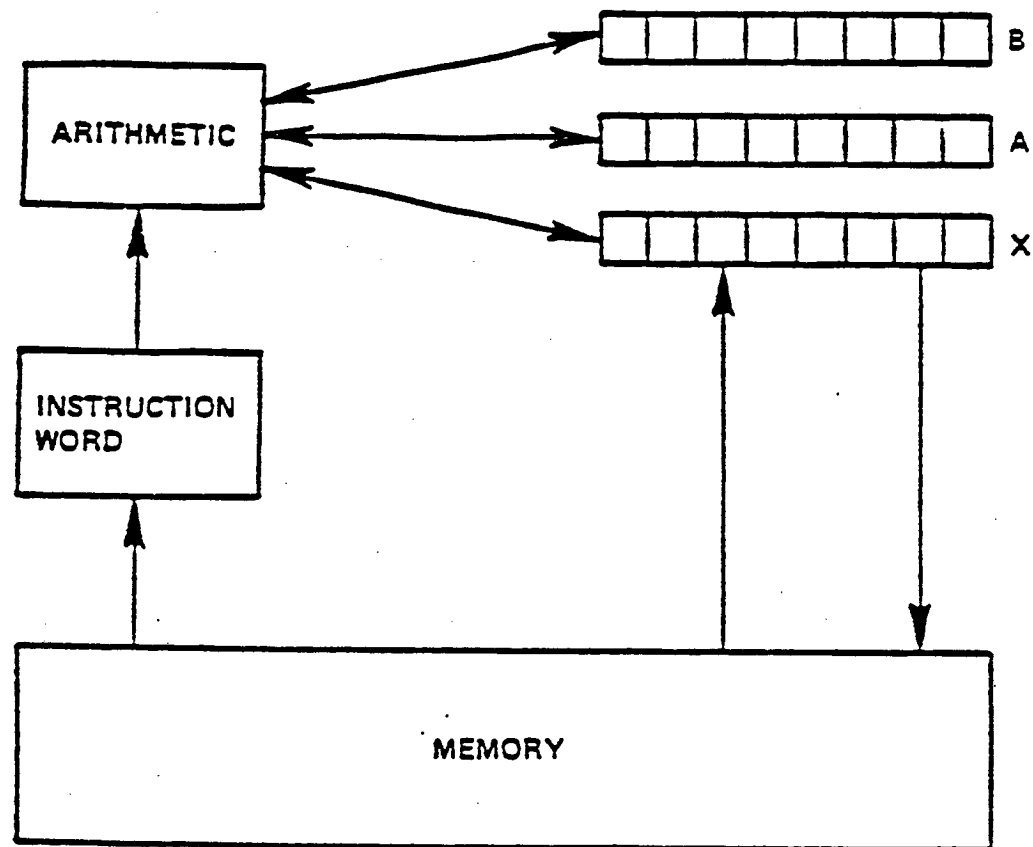
PROGRAMMING PROJECT #1

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.

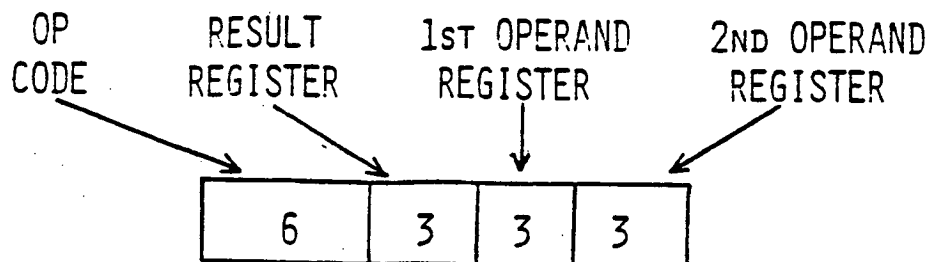


Instruction Flow

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

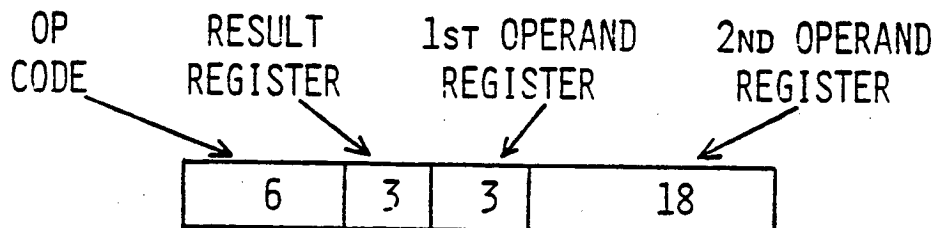
15 BIT INSTRUCTIONS



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

<u>MACHINE CODE</u>	<u>COMPASS</u>	
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:

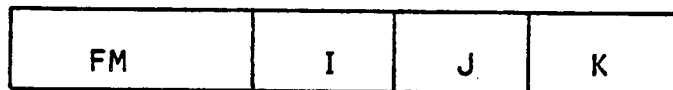
6111000001	SB1	B1+1
5160001000	SA6	B0+1000 ₈
0412000023	EQ	B1,B2,23 ₈

INSTRUCTION FORMATS

15-BIT INSTRUCTIONS

6-BIT
OP CODE

3-BIT
OPERAND



3-BIT OPERAND

3-BIT
RESULT

30-BIT INSTRUCTIONS

6-BIT
OP CODE

3-BIT
OPERAND



3-BIT
RESULT

18-BIT OPERAND
CONSTANT OR BRANCH ADDRESS

INSTRUCTION WORD FORMATS

15	15	15	15
----	----	----	----

15	15	30
----	----	----

15	30	15
----	----	----

30	30
----	----

30	15	15
----	----	----

60

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 0 2 5
	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 /

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.

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	<u>IDENT</u>	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	<u>ENTRY</u>	LINKAGE SYMBOLS SEPARATED BY COMMAS
- LINKAGE SYMBOLS ARE DECLARED TO THE LOADER AS ENTRY POINTS. - EACH LINKAGE SYMBOL MUST BE DEFINED IN THE ASSEMBLY AS A NON-EXTERNAL SYMBOL.		

SYMBOL OR BLANK	<u>END</u>	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	<u>BSS</u>	ABSOLUTE ADDRESS
OR		EXPRESSION
BLANK		

- 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 0 FORCES UPPER WITHOUT ALLOCATING STORAGE.

SYMBOL	<u>BSSZ</u>	ABSOLUTE ADDRESS
OR		EXPRESSION
BLANK		

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

BLANK	<u>DATA</u>	ABSOLUTE ADDRESS
OR		EXPRESSION
SYMBOL		

- 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
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/28/70 16.27.53.

PAGE 1

ADDRESS LENGTH

BINARY CONTROL CARDS.

0 12
12

IDENT ADD
END GO

START ASSEMBLY
TERMINATE COMPASS ASSEMBLY

ENTRY POINTS.

00 4.

EXTERNAL SYMBOLS.

SYS=

ADD START ASSEMBLY

COMPASS 3.5-470.

06/28/70 16.27.53.

PAGE 2

IDENT ADD
ENTRY GO
LIST 0

START ASSEMBLY
GO IS AN ENTRY POINT
SHOW DETAIL
DATA FOR PROGRAM

0 777777777777777750
1 000000000000000004
2 000000000000000025
3 777777777777777776
4 5110000000 *
5120000001 *
5 5130000002 *
5140000003 *
6 36612
36734
36667
7 5160000011 *
7160247021
11 1

LOCADDR

DATA -23.4.21.-1
00 SA1 LOCADDR
SA2 LOCADDR+1
SA3 LOCADDR+2
SA4 LOCADDR+3
IX6 X1+X2
IX7 X3+X4
IX8 X6+X7
SA6 RESULT
ENDRUN
RESULT RSSZ 1

EXECUTION BEGINS HERE
X2 = 2ND DATA WORD
X3 = 3RD DATA WORD
X4 = 4TH DATA WORD
X6 = 1ST SUM
X7 = 2ND SUM
X8 = FINAL RESULT
STORE ANSWER
TERMINATE PROGRAM EXECUTION
RESERVE WORD OF ZEROS FOR ANSWER

DEFAULT SYMBOLS DEFINED BY COMPASS.

0 X

SYS=

12

END 00

TERMINATE COMPASS ASSEMBLY

473000 SCM STORAGE USED
MODEL 174 ASSEMBLY

10 STATEMENTS
0.024 SECONDS

4 SYMBOLS
10 REFERENCES

3-15

ADD START ASSEMBLY
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/28/78 16.27.51.

PAGE 3

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

LOAD MAP - ADD

CYREN 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	PROCESSR VER	LEVEL	HARDWARE	COMMENTS
ADD	111	12	LGO	06/28/78	COMPASS	3.5	470	
SYS.RM	123	40	SL-SYSL10	05/16/78	COMPASS	3.5	470	PROCESS SYSTEM REQUEST.

.025 CP SECONDS

13200H 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 20452 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- CYH175-SN1 4LB7/R6H 04/15/78
 16.27.51.DON00FO FROM /SH
 16.27.51.IP 00000256 WORDS - FILE INPUT , DC 04
 16.27.51.DON. PSD.02/8.72CT011A.MILLER
 16.27.52.COMPASS.
 16.27.53. ASSEMBLY COMPLETE. 47300B SCH 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 .431 SEC. .611 ADJ.
 16.27.54.CH 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/Pass 1

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

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

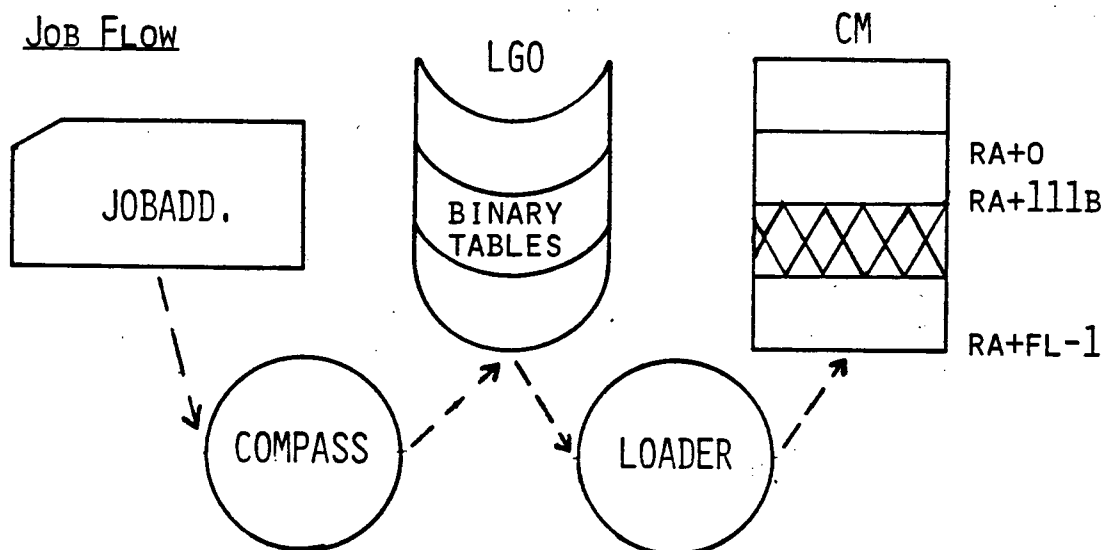
3-19

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
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 08.58.49.

PAGE 1

ADDRESS LENGTH

0 41
41

BINARY CONTROL CARDS.

IDENT	FROM	TRANSFER ADDRESS
END	START	

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM*	LOCAL	0	35
LITERALS*	LOCAL	35	4

ENTRY POINTS.

START	0*	BUF	7.

EXTERNAL SYMBOLS.

SUBJ	SYS*

IDENT ERROR
ENTRY START,RUF
EXT SUR1
LIST *

THIS IS AN EXAMPLE OF AN ERROR LISTING

0	46000		START	NO			
		7	FLAG	EOU	7		
	5110000034 *			SA1	TEMP		
	64243			SN2	R3.A.12		
1	7160000010			SK6	10R	DATA	
	5160000007 *			SA6	RUF	STORE IT	
	35 *	LITERAL	LIT	2.4.6		LITERAL	
2	5110000036 *			SA1	LITERAL.1	GET A LITERAL (2ND LITERAL)	
	18711			BR7	X1	MOVE IT OVER	
3	5170000010 *			SA7	RUF.1	STORE IT	
	5170000011 *			SA,FLAG	RUF.2		
4	0100000000 X			RJ	SUR1	GO TO SUR1	
				IF	DEF.FLAG.1		
5	5110000040 *			SA1	=100		
				ENDRUN			
	7160247021		SK6 JREND.4.1			ENDRUN	.1
6	20650		LX6 400			ENDRUN	.1
	0100000000 X		RJ =XSYS=			ENDRUN	.1
			ENDM			ENDRUN	.1
7		12	RUF	BSS2	10	BUFFER IN PROGRAM	
21	0100000000 X			RJ	SUR1	SHOW LINKAGE	
22	46000			NO			
	0100000000 X			RJ	SUR1		
				JUNK HERE			
23	5160000034 *			SA6	=STEMP	BSS AT END OF PROGRAM	
	20673			LX6	273R	NONFATAL ERROR (NUMBER)	
24	0000000000			LX6	X6.10	FATAL ERROR (LETTER)	
25		5	B4	BSS	5	B4 ILLEGAL SYMBOL	
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
34
SYS=
TEMP

CONTENT OF LITERALS BLOCK.

35 00000000000000000002
36 00000000000000000004
37 00000000000000000006
40 000000000000000000144

B
D
F
A9

41
END START TRANSFER ADDRESS

47300R SCH STORAGE USED 3R STATEMENTS 10 SYMBOLS
MODEL 174 ASSEMBLY 0.045 SECONDS 22 REFERENCES

2 ERRORS IN ERROR

3-22

7
0
1
0

ERROR
ERROR DIRECTORY.

COMPASS 3.5-470.

06/27/70 08.58.49.

PAGE 3

O TYPE ERROR
OCCURRED ON PAGES

OPERATION FIELD BAD.
2

U TYPE ERROR
OCCURRED ON PAGES

UNDEFINED SYMBOL. VALUE ASSUMED 0.
2

I TYPE ERROR
OCCURRED ON PAGES

LOCATION SYMBOL BAD. SYMBOL NOT DEFINED.
2

7 TYPE ERROR
OCCURRED ON PAGES

ADDRESS VALUE EXCEEDS FIELD SIZE, RESULT TRUNCATED
2

FRPDR
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 08.58.49.

PAGE 4

PUF	7	PROGRAM*	2/02 F	2/11 S	2/17 S	2/18 S	2/27 L
R4	25	PROGRAM*	2/35 L				
FLAG	7		2/09 D	2/14	2/20 F		
LGD.	33	PROGRAM*	2/37 L				
LITERAL	35	PROGRAM*	2/14 L	2/15			
START	0	PROGRAM*	2/02 F	2/08 L			
SURI	0	EXTERNAL*	2/03 X	2/19	2/28	2/30	
SYS*	0	EXTERNAL*	2/25				
TEMP	74	PROGRAM*	2/10	2/32 S			
UNDEF			2/36				

U

3-24

LOAD MAP -

CYBER LOADER 1.4-470

06/27/70 00.50.49.

PAGE 1

PWA OF THE LOAD
LVA-1 OF THE LOAD

111
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

132000 CM STORAGE USED

1 TABLE MOVE

OMPX.

P	000714	A0	000150	R0	000000
PA	124200	A1	000001	R1	000001
FL	030000	A2	013021	R2	777776
FA	700700	A3	000717	R3	006714
FE	000041	A4	000300	R4	006067
FE	000000	A5	011033	R5	007070
MA	000400	A6	000001	R6	005131
		A7	013021	R7	002440
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

CIA11	=	0102	2401	0000	0000	0000	CIA11	=	0102	2401	0000	0000	0000
CIA21	=	4000	0000	0002	0000	0111	CIA21	=	4000	0000	0002	0000	0111
CIA31	=	0000	0000	0000	0000	0000	CIA31	=	0000	0000	0000	0000	0000
CIA41	=	0000	0000	0000	0000	0000	CIA41	=	0000	0000	0000	0000	0000
CIA51	=	5100	0000	0001	0000	0014	CIA51	=	5100	0000	0001	0000	0014
CIA61	=	0102	2401	0000	0000	0000	CIA61	=	0102	2401	0000	0000	0000
CIA71	=	4000	0000	0002	0000	0111	CIA71	=	4000	0000	0002	0000	0111

00000	00000	00000	00000	00000
00054	56110	03110	00054	56120
00060	56154	63310	13415	21422
00064	16250	31405	25230	00000
00070	14071	75755	00000	00000
00100	51000	00000	01000	00014

01022	40100	00000	00000
63310	67410	13414	21422
61447	77776	03040	00056
00000	00000	00010	06067
00000	00000	00000	00000

00000	00000	00000	00000
03140	00057	02300	00000
07040	00060	51600	00001
40000	00000	02000	02440

13443	03040	00060	67405
04000	00063	00000	00021
00000	00000	40000	00000

00634	22611	43700	51700	00340
00640	53610	70217	77715	54710
00644	03040	00647	37472	46000
00650	20173	03310	00654	46000
00654	71700	00014	01000	00543
00660	51400	00657	04700	00657
00664	63321	05350	00665	63330
00670	73720	75231	01000	01160
00674	03360	00676	71700	00014
00700	03310	00703	03010	00703
00704	71700	00014	01000	00543
00710	51400	00707	52120	00004
00714	56160	04340	00677	22601
00720	71600	31117	51600	00554
00724	03020	00730	71600	00003
00730	67221	06200	00723	46000
00734	04000	01745	00000	00000
00740	51100	00066	03210	00742
00744	20652	01000	00734	46000
00750	71602	20314	04000	00744
00754	53160	20173	03310	00752
00760	20223	71603	24614	12662
00764	03260	00760	13221	76610
00770	00000	03646	00000	00001
00774	20150	74460	12616	46000
01000	00000	00000	61000	46000
01004	12662	10711	53769	20252
01010	03060	01000	43101	46000
01014	04000	01013	61000	46000
01020	01110	00000	04000	01023
01024	01009	01715	61000	46000
01030	10022	54060	61000	46000
01034	01000	01715	61000	46000

54130	73710	54711	76160
04000	02707	00000	00000
03140	00640	55311	54211
03010	00641	01000	00752
04000	00652	61000	46000
52120	00004	53321	66767
56160	04340	00677	66661
20173	54720	03210	00676
01000	00543	61000	46000
01000	00746	61000	46000
65241	02220	00000	46000
53321	43460	63510	46000
66661	15714	53620	76230
66211	43052	71200	00115
12616	71200	00115	46000
04000	00717	61000	46000
51100	00001	03110	00735
71600	00130	20660	46000
51100	00001	03110	00745
20150	36661	01000	00734
03010	00752	51100	00001
20651	01000	00734	46000
20650	12226	73610	20636
04004	00771	61000	46000
01000	00734	61000	46000
71306	42465	43652	15336
20736	01000	00734	46000
04000	01000	61000	46000
00000	00000	61000	46000
54100	10033	53040	46000
00000	00000	00000	00000
01210	00000	04000	01033

03260	00637	77101	46000
54311	73711	46000	37473
73730	37472	03140	00640
52120	00002	10644	54311
04000	00663	61000	46000
63510	46000	61000	46000
10611	53620	46000	76230
54330	77643	63230	22316
64240	02220	00000	46000
65241	02220	00000	46000
04000	00712	61000	46000
50130	00002	54231	63410
03170	00713	04000	00670
01000	00752	61000	46000
01000	00525	61000	46000
04000	00744	00000	00000
54610	04000	00733	46000
51600	00733	51100	00001
04000	00573	00000	00000
04000	00723	00000	00000
03110	00754	74110	46000
04000	03640	00000	00000
51600	00770	74160	12212
20136	20614	12616	46000
04000	00771	61000	46000
71600	01012	72220	00002
76310	11323	01000	01364
00000	00000	00000	00000
10300	74400	10022	46000
04000	01015	61000	46000
00000	00000	61000	46000
10033	53040	04000	01026

64240	54771	02220	00000
03340	00640	54331	73230
10466	50117	77775	74210
04000	00642	61000	46000
04004	00657	61000	46000
50130	00002	54231	63410
05670	00664	61000	46000
77752	21674	13467	37643
55131	73620	20173	54620
75231	01000	01160	46000
04000	02670	00000	00000
63321	05350	00714	63330
04000	01742	00000	00000
51120	00342	15210	11101
71707	77603	01000	00543
01300	00000	00000	00000
51100	00732	10611	46000
10611	01000	00732	46000
51100	00001	03110	00746
71602	20314	20652	36662
04000	00751	61000	46000
43266	15262	20222	12212
04000	00760	61000	46000
51600	00777	71102	06501
00000	00000	00000	00000
20350	20222	12663	20136
54170	21136	37617	10233
04000	02227	00000	00000
51000	01025	61000	46000
43101	71200	02406	46000
10300	74400	51600	01025
43101	71200	02410	46000

MFF NR2- CYB175-SN1 4LB7/R6B 05/15/78
 00.50.47.DON005G FROM /SM
 00.50.47.JP 000003R4 WORDS - FILE INPUT , DC 04
 00.50.47.DON. PSD-0270.72CT011A-MILLER
 00.50.48.COMPASS.
 00.50.49. 3 WARNING MESSAGES IN ERROR
 00.50.49. 2 ERRORS IN ERROR
 00.50.49. ASSEMBLY ERRORS. 473000 SCM USED.
 00.50.49. 0.091 CPU SECONDS ASSEMBLY TIME.
 00.50.49.LGO.
 00.50.49. FATAL LOADER ERROR - SEE MAP
 00.50.50.EXIT.
 00.50.50.OP 00001600 WORDS - FILE OUTPUT , DC 40
 00.50.50.WS 3504 WORDS (7160 MAX USED)
 00.50.50.CPA .120 SEC. .120 ADJ.
 00.50.50.I0 .624 SEC. .624 ADJ.
 00.50.50.CH 10.499 MVS. .640 ADJ.
 00.50.50.SS 1.393
 00.50.50.PP 2.715 SEC. DATE 06/27/78
 00.50.50.EJ END OF JOB, SH

SUBPROGRAM STRUCTURE

Main Program

EEEEEEE	IDENT	MAIN	ROUTINE IS NAMED MAIN
	ENTRY	EEEEEEE	ENTRY POINT
	EXT	XXXXXXX	XXXXXXX IS AN EXTERNAL SYMBOL
	*		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

XXXXXXX	IDENT	SUB	ROUTINE IS NAMED SUB
	ENTRY	XXXXXXX	DECLARES XXXXXX AS AN ENTRY POINT
	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	$\pm$ OPERAND	(50)
	B REGISTER	$\pm$ OPERAND	(51)
	X REGISTER	$\pm$ OPERAND	(52)
	X REGISTER	+ B REGISTER	(53)
	A REGISTER	$\pm$ B REGISTER	(54-55)
	B REGISTER	$\pm$ B REGISTER	(56-57)
SET B REGISTER TO	A REGISTER	$\pm$ OPERAND	(60)
	B REGISTER	$\pm$ OPERAND	(61)
	X REGISTER	$\pm$ OPERAND	(62)
	X REGISTER	+ B REGISTER	(63)
	A REGISTER	$\pm$ B REGISTER	(64-65)
	B REGISTER	$\pm$ B REGISTER	(66-67)
SET X REGISTER TO	A REGISTER	$\pm$ OPERAND	(70)
	B REGISTER	$\pm$ OPERAND	(71)
	X REGISTER	$\pm$ OPERAND	(72)
	X REGISTER	+ B REGISTER	(73)
	A REGISTER	$\pm$ B REGISTER	(74-75)
	B REGISTER	$\pm$ B REGISTER	(76-77)

SET INSTRUCTIONS

- * USE THE TWO INCREMENT UNITS

ALLOWABLE OPERATIONS

30-BIT INSTRUCTIONS

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

A CONSTANT USED AS AN
OPERAND OR
A TAG INSTEAD OF A SECOND
SOURCE REGISTER

15-BIT INSTRUCTIONS

$$\left. \begin{array}{l} X \\ A \\ B \end{array} \right\} \begin{array}{l} +B \\ \pm B \end{array}$$

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 $X \pm X$, $A \pm A$, $X-B$, OR $X \pm A$ ARE ILLEGAL.

ADDRESS LENGTH

0 6
6

BINARY CONTROL CARDS.

IDENT LOADNR
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 •
10611
1 5160000005 •
7160247021
3 00000000000001234567
4 2
6LOADNR SAI DAT LOAD THE NUMBER
BX6 X1 TRANSFER THE NUMBER
SA6 DAT*2 STORE NUMBER
ENDRUN TERMINATE JOB NORMALLY
DAT DATA 12345678 NUMBER
BSSZ 2 PLACE TO STORE NUMBER
END LOADNR TRANSFER ADDRESS47300R SCH STORAGE USED
MODEL 174 ASSEMBLY18 STATEMENTS
0.021 SECONDS3 SYMBOLS
6 REFERENCESLOADNR
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 16.17.43.

PAGE 3

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

132008 CM STORAGE USED

1 TABLE MOVE

4-5

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

D

D

D

D

D

D

D

D

WFF NR2- 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.027R.72CT011A.MILLER
 16.17.42.COMPASS.
 16.17.43. ASSEMBLY COMPLETE. 473008 SCH 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.CH 9.760 MVS. .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

```

1      PROGRAM TESTSET(OUTPUT)
      COMMON //ANSWER(4)
      CALL SETS
      PRINT 1,(ANSWER(1),1=1,4)
5      FORMAT(1X,4(5X,020))
      END

```

SETS
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.17.58.

PAGE 1

ADDRESS	LENGTH	PRIMARY CONTROL CARDS.
0	14	IDENT SETS
14		END

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM*	LOCAL	0	14
//	COMMON	0	4

ENTRY POINTS.

SETS 0*

4-6

SETS

COMPASS 3.5-470.

06/27/78 16.17.59.

PAGE 2

	ADDRESS	LENGTH	IDENT SETS	ENTRY SETS
0	46000		NO	
1	5110000010		SA1	HUF*0
	5120000011		SA2	HUF*1
2	5130000012		SA3	HUF*2
	5140000013		SA4	HUF*3
3	6110200000		SR1	200000B
	73611		SX6	91*X1
4	5160000000		SA6	ANS
	73721		SX7	91*X2
5	5170000001		SA7	ANS*1
	73631		SX6	91*X3
6	5160000002		SA6	ANS*2
	73741		SX7	91*X4
7	5170000003		SA7	ANS*3
	0400000000		EQ	SETS
10	00000000000006123456		DATA	6123456B
11	00000000000001423456		CON	1623456B
12	00000000000001312345		DATA	1312345B
13	00000000000006312345		CON	6312345B
			USE	//
0			BSS	4
14			END	

47300B SCH STORAGE USED
MODEL 174 ASSEMBLY24 STATEMENTS
0.029 SECONDS3 SYMBOLS
13 REFERENCES

```

MFF NB2- CYB175-SN1 4LB7/R68 05/15/78
16.17.55.DON00MA FROM /5H
16.17.55.1P 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.
16.18.03.10 1.582 SEC. 1.582 ADJ.
16.18.03.CH 29.518 KWS. 1.801 ADJ.
16.18.03.SS 3.656
16.18.03.PP 6.529 SEC. DATE 06/27/78
16.18.03.EJ END OF JOB, SH

```

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	FTM	4.7	470	767X 1	PROGRAM OPT=1
SETS	2210	14	LGO	06/27/78	COMPASS	3.5	470		
SYS=ID=	2224	1	SL-FORTRAN	05/02/78	COMPASS	3.5	470		LINK BETWEEN SYS=ID AND INITIALIZATION CODE.
/STP.END/	2225	1							
/FCL.C./	2226	25							
/OB.ID./	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		CMH V1.1 - ALLOCATE FIXED.
CMF.CSF	5313	6	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH V1.1 - CHANGE SPECS FIXED.
CMH.FFA	5321	14	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH V1.1 - FIXED FREE ALGORITHM.
CMF.FRF	5335	36	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH V1.1 - FREE FIXED.
CMH.R	5373	213	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH V1.1 - RESIDENT SUBROUTINES.
CMF.SLF	5606	22	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH V1.1 - SHRINK AT LWA FIXED.
CTLARM	5630	601	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CRM CONTROLLING ROUTINE.
ERRARM	6431	25	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CRM ERROR PROCESSOR ENTRY.
LISTARM	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.AM	7213	40	SL-SYSLIB	05/16/78	COMPASS	3.5	470		PROCESS SYSTEM REQUEST.
//	7253	4							

.117 CP SECONDS

23300R CM STORAGE USED

11 TABLE MOVES

ARITHMETIC IN THE INCREMENT UNIT

INSTRUCTION SX6 B1+X1

(X1) =	0000000000000006123456 ₈	123456
(B1) =	<u> 200000₈</u>	<u>200000</u>
(X6) =	000000000000000323456 ₈	323456

TRUNCATION AND SIGN EXTENSION

INSTRUCTION SX7 B1+X2

(X2) =	0000000000000001623456 ₈	623456
(B1) =	<u> 200000₈</u>	<u>200000</u>
(X7) =	000000000000000023457 ₈	1023456
	END-AROUND-CARRY	1

TRUNCATION AND SIGN EXTENSION 023457

INSTRUCTION SX6 B1+X3

(X3) =	0000000000000001312345 ₈	312345
(B1) =	<u> 200000₈</u>	<u>200000</u>
(X6) =	777777777777777512345 ₈	512345

SIGN EXTENSION AND TRUNCATION

INSTRUCTION SX7 B1+X4

(X4) =	0000000000000006312345 ₈	312345
(B1) =	<u> 200000₈</u>	<u>200000</u>
(X7) =	777777777777777512345 ₈	512345

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
NZ
PL
NG

TEST THE VALUE OF A "X" OR "B" REGISTER

IR
OR
DF
ID

TEST THE VALUE OF A "X" REGISTER

EQ
NE
GE
LT

COMPARE A "B" REGISTER TO A "B" REGISTER

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

IDENT JUMP
ENTRY ENTER

ILLUSTRATE VARIOUS USES OF JUMP INSTRUCTION

EXAMPLE 1 - JUMP TABLE - JUMPS TO ONE OF THREE ROUTINES
DEPENDING UPON THE VALUE PASSES IN R1

0
1 03110
2 0710000000 X
3 0211000004
4 0400000000 X
5 0400000000 X
6 0400000000 X

ENTER R557 1
SR1 X1
SR2 3
NG R1,=XERROR
GE R1,B2,=XERROR
JP TABLE+R1

TABLE

EQ =XROUTINO
EQ =XROUTINI NOTE THE FORCING UPPER
EQ =XROUTIN2 COULD BE JP =XROUTIN2

EXAMPLE 2 - JUMP TO MEMORY LOCATION HOLDADR

7 0110000000 C
0211000000

SR1 HOLDADR
JP R1 EQUIVALENT TO JP 0+R1

EXAMPLE 3 - NON INDEXED JUMP

10 0200000000
11 0400000000

JP ENTER EQUIVALENT TO JP B0+ENTER
EQ ENTER EQUIVALENT TO EQ B0+B0+ENTER

0
12

HOLDADR

USE /DON/
R55 1
END ENTER

47300B SCM STORAGE USED
MODEL 174 ASSEMBLY

39 STATEMENTS
0.034 SECONDS

7 SYMBOLS
13 REFERENCES

JUMP 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	2/3A L		
ROUTINO	0	EXTERNAL*	2/10			
ROUTINI	0	EXTERNAL*	2/19			
ROUTIN2	0	EXTERNAL*	2/20			
TABLE	4	PROGRAM*	2/16	2/1A L		

JUMP
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 09.00.32.

PAGE 1

ADDRESS LENGTH

0 12
12

BINARY CONTROL CARDS.

IDENT JUMP
END ENTER

BLOCKS TYPE ADDRESS LENGTH

PROGRAM LOCAL 0 12
COMMON 0 1

ENTRY POINTS.

ENTER 0*

EXTERNAL SYMBOLS.

ROUTIN2 ROUTINI ROUTINO ERROR

HFF NR2- CYB175-SN1 4LB7/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. 47300R SCH USED.
09.00.33. 0.085 CPU SECONDS ASSEMBLY TIME.
09.00.33.OP 00000640 WORDS - FILE OUTPUT , DC 40
09.00.33.HS 3584 WORDS (10752 MAX USED)
09.00.33.CPA .109 SEC. .108 ADJ.
09.00.33.I0 .474 SEC. .474 ADJ.
09.00.33.CH 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

ADDRESS LENGTH

0 15
15

BINARY CONTROL CARDS.

IDENT JUMPS
END JUMP

ENTRY POINTS.

JUMP 0.

EXTERNAL SYMBOLS.

SYS=

JUMPS

COMPASS 3.5-470.

06/27/70 08.56.05.

PAGE 2

IDENT JUMPS
ENTRY JUMP

CODE GENERATED BY VARIOUS JUMP INSTRUCTIONS

0	0200000014 .	JUMP	JP	TAG	UNCONDITIONAL JUMP
1	0400000014 .		FO	R0,R0.TAG	
	0400000014 .		FO	TAG	
2	0100000014 .		RJ	TAG	VOIDS STACK IN ALL SYSTEMS
3	0410000014 .		EO	R1,R0.TAG	CONDITIONAL JUMP B REGISTER EQUAL ZERO
	0410000014 .		ZR	R1.TAG	
4	0520000014 .		NE	R2,R0.TAG	CONDITIONAL JUMP B REGISTER UNEQUAL TO ZERO
	0520000014 .		NZ	R2.TAG	
5	0630000014 .		GE	R3,R0.TAG	POSITIVE B REGISTER TEST
	0630000014 .		LE	R0,R3.TAG	
6	0630000014 .		PL	R3.TAG	
7	0740000014 .		LT	R4,R0.TAG	NEGATIVE B REGISTER TEST
	0740000014 .		GT	R0,R4.TAG	
	0740000014 .		NG	R4.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	RSS	1	A LOCATION TO JUMP TO
15			END	JUMP	

47300M SCH STORAGE USED
MODEL 174 ASSEMBLY

37 STATEMENTS
0.041 SECONDS

3 SYMBOLS
22 REFERENCES

JUMPS
SYMBOLIC REFERENCE TABLE.

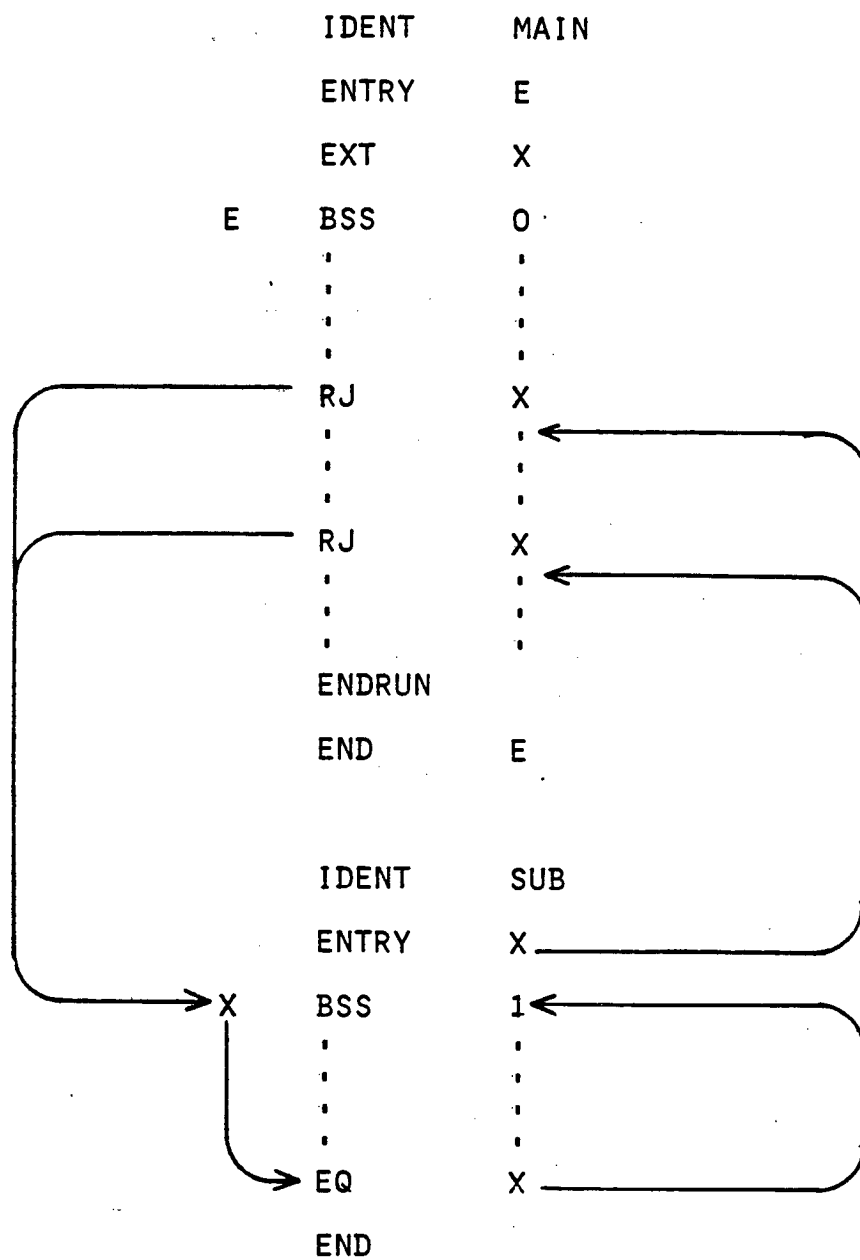
COMPASS 3.5-470.

06/27/78 08.56.05.

PAGE 3

JUMP	0	PROGRAM*	2/02 F	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 NR2- CYR175-SN1 4LB7/R6B 05/15/78
 00.56.01.DON0044 FROM /SH
 00.56.01.IP 00000256 WORDS - FILE INPUT , DC 04
 00.56.01.DON. PSD.027B.72CT011A.MILLER
 00.56.04.COMPASS.
 00.56.05. ASSEMBLY COMPLETE. 47300B SCH USED.
 00.56.05. 0.093 CPU SECONDS ASSEMBLY TIME.
 00.56.05.OP 00000576 WORDS - FILE OUTPUT , DC 40
 00.56.05.WS 3584 WORDS (7168 MAX USED)
 00.56.05.CPA .126 SEC. .126 ADJ.
 00.56.05.I0 .473 SEC. .473 ADJ.
 00.56.05.CM 0.531 KWS. .520 ADJ.
 00.56.05.SS 1.121
 00.56.05.PP 4.012 SEC. DATE 06/27/78
 00.56.05.EJ END OF JOB. SH



LINKAGE • ENTRY,EXT,RJ

IDENT RJTEST
ENTRY RJTEST

EXAMPLE OF CODE GENERATED BY A RETURN JUMP RJ INSTRUCTION

0 0100000002 *
1 0200000000

RJTEST RJ SUB
JP 0

2 7777777777777777
3 0400000002 *
4

SUB DATA -0
EQ SUB
END RJTEST

473000 SCM STORAGE USED
MODEL 174 ASSEMBLY

15 STATEMENTS
0.012 SECONDS

2 SYMBOLS
5 REFERENCES

LOAD MAP - RJTEST

CYBER LOADER 1.4-470

06/27/78 08.52.06.

PAGE 1

FVA OF THE LOAD 111
LVA-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	LOO	06/27/78	COMPASS 3.5 470		

.007 CP SECONDS

132000 CM STORAGE USED

1 TABLE MOVE

DUMP RELATIVE

DMP(111,117)

00111 01000 00113 61000 46000
00114 04000 00113 61000 46000

02000 00000 61000 46000
60000 00000 04004 00115

04000 00112 00000 00000
00117 60000 00000 04004 00117

RJTEST
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 08.52.06.

PAGE 1

ADDRESS LENGTH
0 4
4

BINARY CONTROL CARDS.

IDENT RJTEST
END RJTEST

ENTRY POINTS.

RJTEST 0*

DMPX.

P	000000	A0	000200	B0	000000
PA	125200	A1	000001	B1	000001
FL	000200	A2	000000	B2	000002
EM	700700	A3	000057	B3	012733
RE	000041	A4	000001	B4	000201
FE	000000	A5	000111	B5	000111
HA	000400	A6	000001	B6	000200
		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

CIA1)=	0000	0000	0000	0000	0000	CIR1)=	0000	0000	0000	0000	0000
CIA2)=	1505	1520	0000	0000	0061	CIR2)=	0000	0000	0000	0000	0000
CIA3)=	0000	0000	0000	0000	0000	CIR3)=					
CIA4)=	0000	0000	0000	0000	0000	CIR4)=					
CIA5)=	0100	0001	1361	0004	6000	CIR5)=	0100	0001	1361	0004	6000
CIA6)=	0000	0000	0000	0000	0000	CIR6)=					
CIA7)=	0000	0000	0000	0000	0000	CIR7)=					

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
00100	54000	00000	01000	00001

51100	00001	03110	00055
00000	00200	00000	00001
00000	00000	00000	00115
00000	00000	00000	00000

64550	02550	00000	46000
07040	00060	51600	00001
40000	00000	02000	00111

00000	00000	00000	00000
04000	00063	00000	00021
00000	00000	40000	00000

HFF NB2- CYB175-SN1 4LBT/R6B 05/15/78

08.52.04.D0M0040 FROM /SH

08.52.04.IP 00000192 WORDS - FILE INPUT * DC 04

08.52.04.D0M. PSD.0278.72CT011A-MILLER

08.52.06.COMPASS.

08.52.06.ASSEMBLY COMPLETE. 47300R SCM USED.

08.52.06. 0.062 CPU SECONDS ASSEMBLY TIME.

08.52.06.L00.

08.52.07.ERROR MODE =00. ADDRESS =000000

08.52.07.EXIT.

08.52.07.DMP(111.117)

08.52.07.OP 00000760 WORDS - FILE OUTPUT * DC 40

08.52.07.MS 3504 WORDS (10752 MAX USED)

08.52.07.CPA .100 SEC. .100 ADJ.

08.52.07.I0 .619 SEC. .619 ADJ.

08.52.07.CH 9.242 MWS. .563 ADJ.

08.52.07.SS 1.203

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.

<u>J = K → I</u>		
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)=~~7~~7707 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

[illegible]

2 SYMBOL:
13 REFERENCE:

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	<u>RIGHT, 3 BITS</u>
AX3	X1,B2	<u>LEFT, 3 BITS</u>

↑ INTO ↑ OUT OF ↙ SHIFT COUNT

4-34

0 400000000000000000000000

1 5110000000 *

10011

10211

2 10311

10411

10511

10611

3 10711

21100

20274

21312

4 20412

21562

20562

5 02000000000

6

IDENT	SHIFT
ENTRY	SHIFT
DATA	400000000000000000000000

SHIFT

SA1	0
AX0	X1
AX2	X1
AX3	X1
AX4	X1
AX5	X1
AX6	X1
AX7	X1

AX1	0
LX2	60
AX3	10
LX4	10
AX5	-10
LX6	-10
JP	0
END	SHIFT

.RIGHT SHIFT 0 IE. NO. CHANGE
 .LEFT SHIFT 60 BITS IE. NO CHANGE

 .LEFT SHIFT 10 BITS
 .RIGHT SHIFT (60-10) BITS
 .LEFT SHIFT (60-10) BITS END AROUND

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

0	5110000006	6	CONSTANT	IDENT	SHIFT	
	10011		SHIFT	ENTRY	SHIFT	
1	6110000006			111	40000000000000000000H	
	6120777771			SA1	CONSTANT	
2	6130777700			RX0	X1	
	6140777677			SR1	A	
3	6150777676			SR2	-6	
	22210			SR3	-63	-63 DECIMAL
	22320			SR4	-64	-64 DECIMAL
4	22430			SR5	-65	-65 DECIMAL
	22540			LX2	R1,X0	SHIFT LEFT 6 BITS
	22650			LX3	R2,X0	SHIFT RIGHT 6 BITS
5	0200000000			LX4	R3,X0	RIGHT SHIFT 63 PLACES
7				LX5	R4,X0	CLEAR DESTINATION REGISTER
				LX6	R5,X0	
				JP	0	
				END	SHIFT	

47300B SCM STORAGE USED

MODEL 174 ASSEMBLY

17 STATEMENTS

0.018 SECONDS

2 SYMBOLS

4 REFERENCES

4-35

DNPR.

P	000000	A0	000200	R0	000000
PA	220200	A1	000117	R1	000006
FL	000200	A2	000060	R2	777771
FM	700700	A3	000057	R3	777700
PE	000045	A4	000001	R4	777677
FE	000000	A5	000111	R5	777676
MA	000600	A6	000001	R6	000200
		A7	000001	R7	027756
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

C(A1)=	4000	0000	0000	0000	0000
C(A2)=	1505	1520	0000	0000	0061
C(A3)=	0000	0000	0000	0000	0000
C(A4)=	0000	0000	0000	0000	0000
C(A5)=	5110	0001	1710	0114	6000
C(A6)=	0000	0000	0000	0000	0000
C(A7)=	0000	0000	0000	0000	0000

C(R1)=	0000	0000	0000	0000	0000
C(R2)=					
C(R3)=					
C(R4)=					
C(R5)=					
C(R6)=					
C(R7)=					

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
00100	54000	00000	01000	00001

51100	00001	03110	00055
00000	00200	00000	00001
00000	00000	00000	00120
00000	00000	00000	00000

A4550	02550	00000	46000
07040	00060	51600	00001
40000	00000	02000	00111

00000	00000	00000	00000
04000	00063	00000	00021
00000	00000	40000	00000

SHIFTB
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.18.08.

PAGE 1

ADDRESS LENGTH

BINARY CONTROL CARDS.

0 7
7

IDENI SHIFTB
END SHIFTB

BLOCKS TYPE ADDRESS LENGTH

PROGRAM LOCAL 0 6
LITERALS LOCAL 6 1

ENTRY POINTS.

SHIFTB 0*

LOAD MAP - SHIFTB

CYBER LOADER 1.4-470

06/27/78 16.18.08.

PAGE 1

FMA OF THE LOAD 111
LMA+1 OF THE LOAD 120

TRANSFER ADDRESS -- SHIFTB 111

PROGRAM ENTRY POINTS -- SHIFTB 111

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
SHIFTB	111	7	LGO	06/27/78	COMPASS 3.5 470		

.000 CP SECONDS

132008 CM STORAGE USED

1 TABLE MOVE

MFF M02- CYB175-SH1 4L87/R60 05/15/78
16.18.06.D0M00MR FROM /SH
16.18.06.1P 00000320 WORDS - FILE INPUT , DC 40
16.18.06.D0M. PS0.0278.72CT011A-MILLER
16.18.07.COMPASS.
16.18.08. ASSEMBLY COMPLETE. 473008 SCH 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.
16.18.09.10 .613 SEC. .613 ADJ.
16.18.09.CH 9.261 MWS. .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. SH

1STSHFT
STORAGE ALLOCATION.

COMPASS 3.5-410.

06/27/78 09.00.54.

PAGE 1

ADDRESS LENGTH

0 32
32

BINARY CONTROL CARDS.

IDENT 1STSHFT
END 1STSHFT

ENTRY POINTS.

1STSHFT 14.

EXTERNAL SYMBOLS.

SYS.

			IDENT	TSTSHFT	
			ENTRY	TSTSHFT	
		ANSWER	HSS?	12	
	0		HX0	1	
	14	43001	HX6	X0	
		10600	HX7	X0	
		10700	LX6	100	LEAST SIGNIFICANT 6 BITS USED 44(OCTAL)=36(DECIMAL)
7		20644	AX7	100	LEAST SIGNIFICANT 6 BITS USED 44(OCTAL)=36(DECIMAL)
7	15	21744	SA6	ANSWER+0	
		5160000000 *	SA7	ANSWER+1	
	16	5170000001 *	HX6	X0	
		10600	HX7	X0	
		10700	LX6	-100	LEAST SIGNIFICANT 6 BITS USED 33(OCTAL)=27(DECIMAL)
7	17	20633	AX7	-100	LEAST SIGNIFICANT 6 BITS USED 33(OCTAL)=27(DECIMAL)
7		21733	SA6	ANSWER+2	
		5160000002 *	SA7	ANSWER+3	
	20	5170000003 *	HX6	X0	
		10600	HX7	X0	
		10700	LX6	20-32	52(DECIMAL) = 64(OCTAL)
	21	20664	AX7	20-32	52(DECIMAL) = 64(OCTAL)
		21764	SA6	ANSWER+4	
		5160000004 *	SA7	ANSWER+5	
	22	5170000005 *	HX6	X0	
		10600	HX7	X0	
		10700	LX6	32-20	12(DECIMAL) = 14(OCTAL)
	23	20614	AX7	32-20	12(DECIMAL) = 14(OCTAL)
		21714	SA6	ANSWER+6	
		5160000006 *	SA7	ANSWER+7	
	24	5170000007 *	HX6	X0	
		10600	HX7	X0	
		10700	LX6	20-32	-12 = (60-12) = 48(DECIMAL) = 60(OCTAL)
	25	20660	AX7	20-32	-12 = (60-12) = 48(DECIMAL) = 60(OCTAL)
		21760	SA6	ANSWER+8	
		5160000010 *	SA7	ANSWER+9	
	26	5170000011 *	HX6	X0	
		10600	HX7	X0	
		10700	LX6	160-184	-24 = (60-24) = 36(DECIMAL) = 44(OCTAL)
	27	20644	AX7	160-184	-24 = (60-24) = 36(DECIMAL) = 44(OCTAL)
		21744	SA6	ANSWER+10	
		5160000012 *	SA7	ANSWER+11	
	30	5170000013 *	ENDRUN		
		7160247021	END	TSTSHFT	
	32				

47300B SCM STORAGE USED
MODEL 174 ASSEMBLY

45 STATEMENTS
0.055 SECONDS

3 SYMBOLS
16 REFERENCES

TSTSHFT
ERROR DIRECTORY.

COMPASS 3.5-470.

06/27/70 09.00.54.

PAGE

3

7 TYPE ERROR
OCCURRED ON PAGES

ADDRESS VALUE EXCEEDS FIELD SIZE. RESULT TRUNCATED
2

LOAD MAP - TSTSHFT

CYRER LOADER 1.4-470

06/27/78 09.00.54.

PAGE 1

FMA OF THE LOAD 111
LVA-1 OF THE LOAD 203

TRANSFER ADDRESS -- TSTSHFT 125

PROGRAM ENTRY POINTS -- TSTSHFT 125

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
TSTSHFT	111	32	L00	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

132000 CM STORAGE USED

1 TABLE MOVE

4-39

DUMP RELATIVE

DMP(111,147)

00111	00000 00040 00000 00000	77777 77777 77400 00000	00000 00000 04000 00000	00000 00000 00000 04000
00114	77777 77774 00000 00000	00100 00000 00000 00000	77777 77777 77777 77600	
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 00301 03110 00146	54610 04000 00144 46000

TSTSHFT
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 09.00.54.

PAGE 4

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- CYR175-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. 47300R SCH 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.I0 .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

```

1      PROGRAM WRITEIT(OUTPUT)
      COMMON /HLOCK/ ANSWER(25)
      CALL INSTRS
      PRINT 101
5      DO 10 I=1,25
        J=I-1
      10 PRINT 100,J,ANSWER(I)
      100 FORMAT(* ANS*12,023)
      101 FORMAT(1H1)
10     END

```

INSTRS
STORAGE ALLOCATION.

COMPASS 3.6-476.

01/08/79 10.29.06.

PAGE 1

ADDRESS	LENGTH
0	47
47	

BINARY CONTROL CARDS.

IDENT INSTRS
END

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM*	LOCAL	0	47
BLOCK	COMMON	0	31

ENTRY POINTS.

INSTRS 3.

IDENT INSTRS
ENTRY INSTRS
LIST -B,-H

COMMENT SET - SHIFT - MASK - BOOLEAN - INSTRUCTION PROBLEMS

0 000000000000000000004444 BUF LIST \$
1 000000000000000000005555 DATA 4444B,5555B,6666B
2 000000000000000000006666

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 + NUMBERS5 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 5160000012 C	NUMBER10	MX6 SA6	6 ANS+10
24	5110000000 + 21103 10611	NUMBER11	SA1 AX1 BX6 SA6	BUF 3 X1 ANS+11
25	5160000013 C			
26	21101 10711 5170000014 C	NUMBER12	AX1 BX7 SA7	1 X1 ANS+12
27	7110000463 6120000006	NUMBER13	SX1 SB2 LX6 SA6	463B 6 X1,B2 ANS+13
30	22621 5160000015 C			
31	6110777774 22711	NUMBER14	SB1 LX7 SA7	-3 X1,B1 ANS+14
32	5170000016 C			
33	23621 5160000017 C	NUMBER15	AX6 SA6	X1,B2 ANS+15
34	23711 5170000020 C	NUMBER16	AX7 SA7	X1,B1 ANS+16
35	6110000106 7110000007	NUMBER17	SH1 SX1 LX7 SA7	106B 7 X1,B1 ANS+17
36	22711 5170000021 C			
37	23711 5170000022 C	NUMBER18	AX7 SA7	X1,B1 ANS+18
40	43766 5170000023 C	NUMBER19	MX7 SA7	-6 ANS+19

*
*
*
EXAMPLE OF INCORRECT USAGE

41	5150000000 + 20560 10755	NUMBER20	SA5 LX5 BX7 SA7	BUF -12 X5 ANS+20
42	5170000024 C			
43	5150000000 + 21566 10655	NUMBER21	SA5 AX5 BX6 SA6	BUF -6 X5 ANS+21
44	5160000025 C			
45	5140000000 + 20442 10644	NUMBER22	SA4 LX4 BX6 SA6	BUF 162 X4 ANS+22
46	5160000026 C			

0400000003 *

0
47

31

ANS

EQ
USE
BSSZ
ENDINSTRS
/BLOCK/
2547500B CM STORAGE USED
MODEL 74 ASSEMBLY111 STATEMENTS
0.330 SECONDS25 SYMBOLS
58 REFERENCESINSTRS
ERROR DIRECTORY.

COMPASS 3.6-476.

01/08/79 10.29.06.

PAGE 5

7 TYPE ERROR
OCCURRED ON PAGESADDRESS VALUE EXCEEDS FIELD SIZE, RESULT TRUNCATED
3INSTRS
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
BUF	0	PROGRAM*	2/24 S	2/35 S	2/46 S	3/02 S	3/16 S	3/26 S	3/36 S	3/54 S
			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			
			2/26	2/37	3/04	3/46				
			2/02 E	2/16 L	4/02					
			2/22 L							
			3/01 L							
			3/04 L							
			3/09 L							
			3/13 L							
			3/18 L							
			3/22 L							
			3/25 L							
			3/28 L							
			3/32 L							
			3/35 L							
			2/26 L							
			3/41 L							
			3/46 L							
			3/51 L							
			2/30 L							
			2/33 L							
			2/37 L							
			2/42 L							
			2/45 L							
			2/48 L							
			2/51 L							

4-44

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							
/OB.IO./	2360	101							
Q2NTRY=	2461	0	SL-FORTRAN	11/16/78	COMPASS	3.6	485		FCL INITIALIZATION ROUTINE.
/FCL=ENT/	2461	40							
CUMIO=	2521	33	SL-FORTRAN	11/16/78	COMPASS	3.6	485		COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL=FUL	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
SYSALO=	3605	1	SL-FORTRAN	11/16/78	COMPASS	3.6	485		LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
FECHSK=	3606	41	SL-FORTRAN	11/16/78	COMPASS	3.6	485		INITIALIZE CONSTANTS.
FHTAP=	3647	357	SL-FORTRAN	11/16/78	COMPASS	3.6	485		CHACK 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 INTERPRETEN.
OUTC=	5024	150	SL-FORTRAN	11/16/78	COMPASS	3.6	485		FORMATTED WRITE FORTRAN RECORD.
/FOL.COM/	5174	14							
FOL.RES	5210	211	SL-SYSLIB	11/02/78	COMPASS	3.6	485		FAST DYNAMIC LOADER RESIDENT.
FOL.MHI	5421	222	SL-SYSLIB	11/02/78	COMPASS	3.6	485		FOL 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		CHM V1.1 - ALLOCATE FIXED.
CMF.CSF	6063	6	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CHM V1.1 - CHANGE SPECS FIXED.
CHM.FFA	6071	14	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CHM V1.1 - FIXED FREE ALGORITHM.
CMF.FHF	6105	36	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CHM V1.1 - FREE FIXED.
CHM.R	6143	214	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CHM V1.1 - RESIDENT SUBROUTINES.
CMF.SLF	6357	22	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CHM V1.1 - SHRINK AT LWA FIXED.
CTLKRM	6401	433	SL-SYSLIB	11/16/78	COMPASS	3.6	485		CRM CONTROLLING ROUTINE.
ERMSRM	7034	25	SL-SYSLIB	11/16/78	COMPASS	3.6	485		CRM ERROR PROCESSOR ENTRY.
LISTSRM	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

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

POSITIVE COEFFICIENTS	{	000	}	NEGATIVE EXPONENTS	1717	}	
		0 0 0 0					
		001	}	NEGATIVE EXPONENTS	1717		
		1 7 7 7					
		-					
		010	}	POSTIVE EXPONENTS	2014		
2 0 0 0							
011	}	POSTIVE EXPONENTS	2014				
3 7 7 7							
POSITIVE COEFFICIENTS	{	100	}	POSITIVE EXPONENTS	5763	}	
		4 0 0 0					
		101	}	POSITIVE EXPONENTS	5763		
		5 7 7 7					
		-					
		110	}	NEGATIVE EXPONENTS	6060		
6 0 0 0							
111	}	NEGATIVE EXPONENTS	6060				
7 7 7 7							

= 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 $1 \cdot 2^0$. WE NEED THE FOLLOWING INSTRUCTIONS:

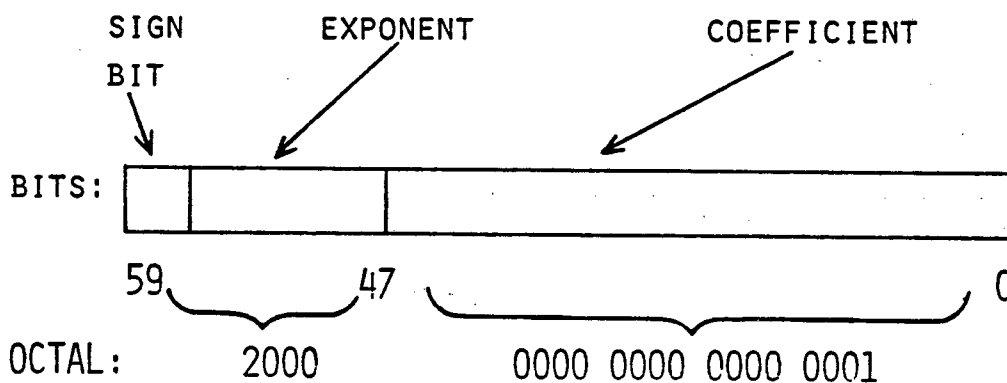
SX1	1	(X1)=0000 0000 0000 0000 0001	(COEFFICIENT)
SB1	0	(B1)=000000	(EXPONENT)
PX2	X1,B1	(X2)=2000 0000 0000 0000 000 1	

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:

$1 \cdot 2^0 = 2000 \ 0000 \ 0000 \ 0000 \ 0001$

$1 \cdot 2^1 = 2001 \ 0000 \ 0000 \ 0000 \ 0001$

$1 \cdot 2^5 = 2005 \ 0000 \ 0000 \ 0000 \ 0001$

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:

$$\text{SX1}\ 1\quad (X1)=0000\ 0000\ 0000\ 0000\ 0001$$

$$\text{SB1}\ -5\quad (B1)=777772$$

$$\text{PX2}\ X1.B1\quad (X2)=1772\ 0000\ 0000\ 0000\ 0001$$

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 RE-
MAINING 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:

$$\text{SX1}\ -1\quad (X1)=7777\ 7777\ 7777\ 7777\ 7776$$

$$\text{SB1}\ 2\quad (B1)=000002$$

$$\text{PX2}\ X1.B1\quad (X2)=5775\ 7777\ 7777\ 7777\ 7776$$

$$\text{SX1}\ -1\quad (X1)=7777\ 7777\ 7777\ 7777\ 7776$$

$$\text{SB1}\ -2\quad (B1)=777775$$

$$\text{PX2}\ X1.B1\quad (X2)=6002\ 7777\ 7777\ 7777\ 7776$$

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 INFINITE OR OUT OF RANGE. 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 INDEFINITE ON A 6000 MACHINE AND OUT OF RANGE 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

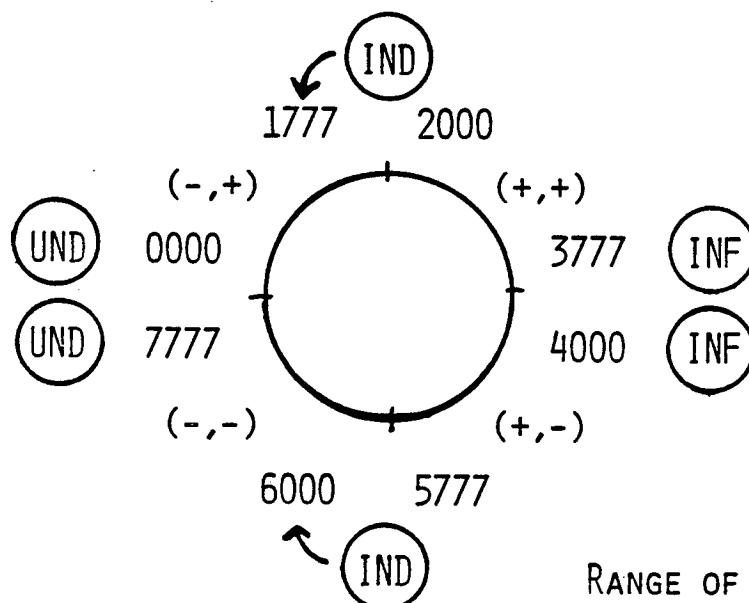
THE PROGRAMMER IS RESPONSIBLE FOR DEVELOPING HIS OWN TEST FOR UNDERFLOW.

FLOATING POINT ARITHMETIC EXERCISES - SPECIAL EXPONENTS

EXERCISE 1

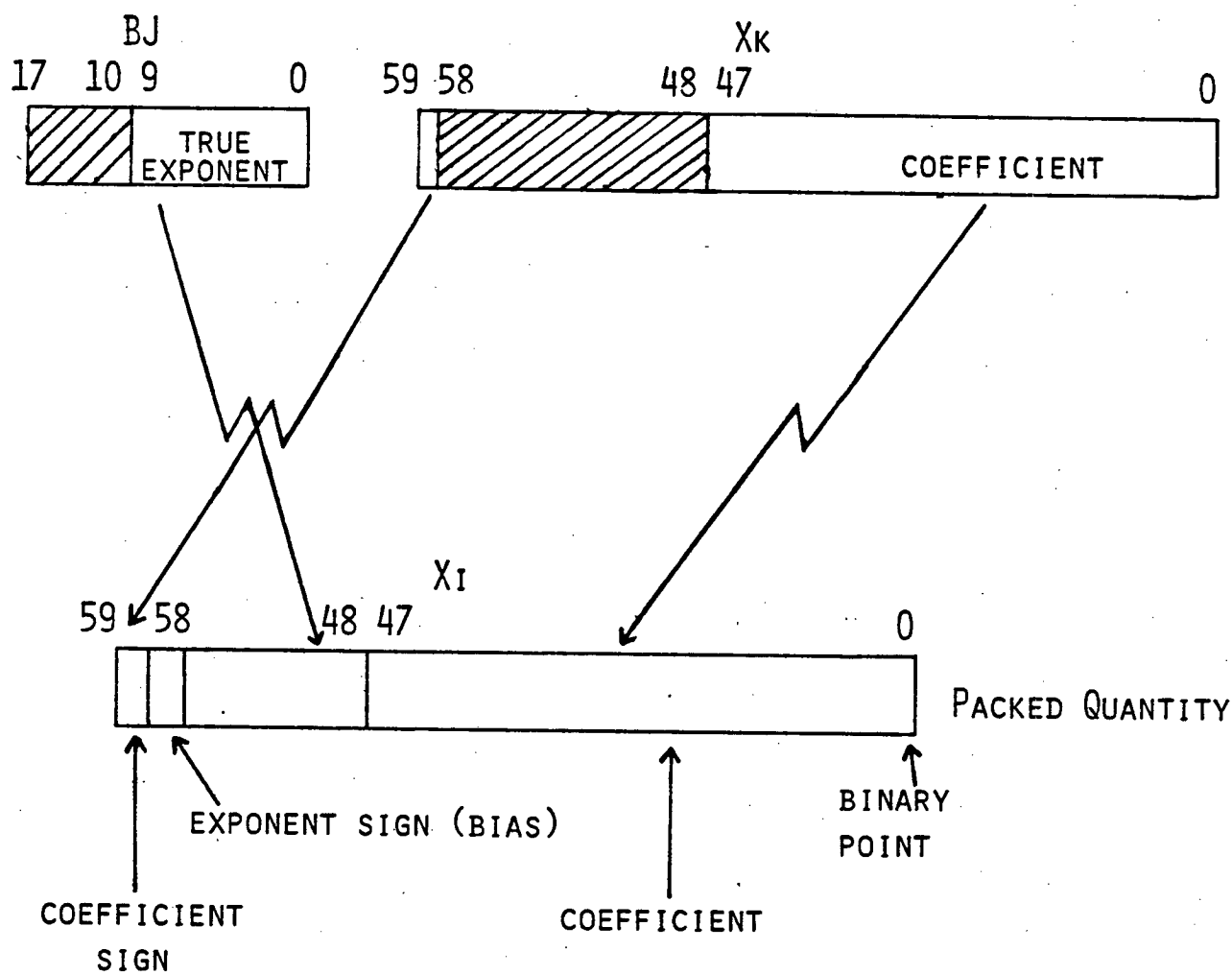
DIRECTIONS: FILL IN THE MISSING ITEMS ON THIS CHART
(REFER TO CHART AT BOTTOM OF PAGE FOR RANGES).

<u>UPPER 12 BITS</u>	<u>EXPONENTS (+ OR -)</u>	<u>COEFFICIENT (+ OR -)</u>
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: PX_I BJ, X_K



HOW TO PACK A FLOATING POINT NUMBER:

$$(X1) = 00000000000000000000_8$$

PX2 X1, B₀

$$(X2) = 20000000000000000000_8$$

EXP
BIASED

TAKEN FROM
LOWER 48 BITS
OF X1 COEFFICIENT

0000 ← 10-BIT EXP
TAKEN FROM B₀

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

2000₈

010 000 000 000 (BINARY)

SIGN OF EXP.

BIAS A POSITIVE EXPONENT BY ADDING 2000₈ 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

$$(X2) = 200000000000000000004_8$$

NX3 X2.B2

$$(X3) = \underbrace{17224}_{\text{ADJUSTED EXPONENT}} \underbrace{000000000000000000}_{\text{COEFFICIENT IS SHIFTED LEFT UNTIL ITS MOST SIGNIFICANT ONE BIT IS IN BIT POS. 47}}_8 \quad (B2) = 000055_8$$

ADJUSTED
EXPONENT

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

1777_8

-55_8

1722_8

BIAS MAKES EXP. SIGN BIT 0 AND
COMPLIMENTS EXP.
EXP. (NO. OF BITS SHIFTED LEFT)

001 111 010 010 ← (BINARY)

SIGN
OF
EXPONENT

10-BIT
COMPLEMENTED
EXPONENT

THE TRUE VALUE OF THE NUMBER IS:

$$4000000000000000000 \times 2^{-55_8}$$

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) = 200000000000000000004₈

NORMALIZE IT:

(X3) = 17224000000000000000₈

COMPLEMENT IT:

(X3) = 6055377777777777777₈

LOGICALLY
THIS IS THE
OPERATION
ACCOMPLISHED

THE BINARY EXPONENT FIELD IS:

6 0 5 5

SIGN
OF
COEF. → 110 000 101 101

SIGN 10-BIT
OF EXP. EXPONENT

THE COEFFICIENT IS COMPLEMENTED:

3777777777777777₈ (48 BITS)

EXAMPLES OF PACKED AND NORMALIZED NUMBERS

NOTE: ALL NUMBERS ARE OCTAL

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

20000000000000000040₈
17254000000000000000₈

PACK IT
NORM. IT

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

17660000000000000004₈
17114000000000000000₈

PACK IT
NORM. IT

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

20000000000000000040₈
17254000000000000000₈
6052377777777777777₈

PACK IT
NORM. IT
COMPLEMENT IT

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

17660000000000000004₈
17114000000000000000₈
6066377777777777777₈

PACK IT
NORM. IT
COMPLEMENT IT

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

177400000000000000014₈
17206000000000000000₈

PACK IT
NORM. IT

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

177400000000000000014₈
17206000000000000000₈
6057177777777777777₈

PACK IT
NORM. IT
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 EXPO-
NENT IS DECREMENTED ACCORDINGLY. THE ORIGINAL AND
NORMALIZED VALUE ARE THE SAME BUT THEIR INTERNAL REP-
RESENTATION 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)

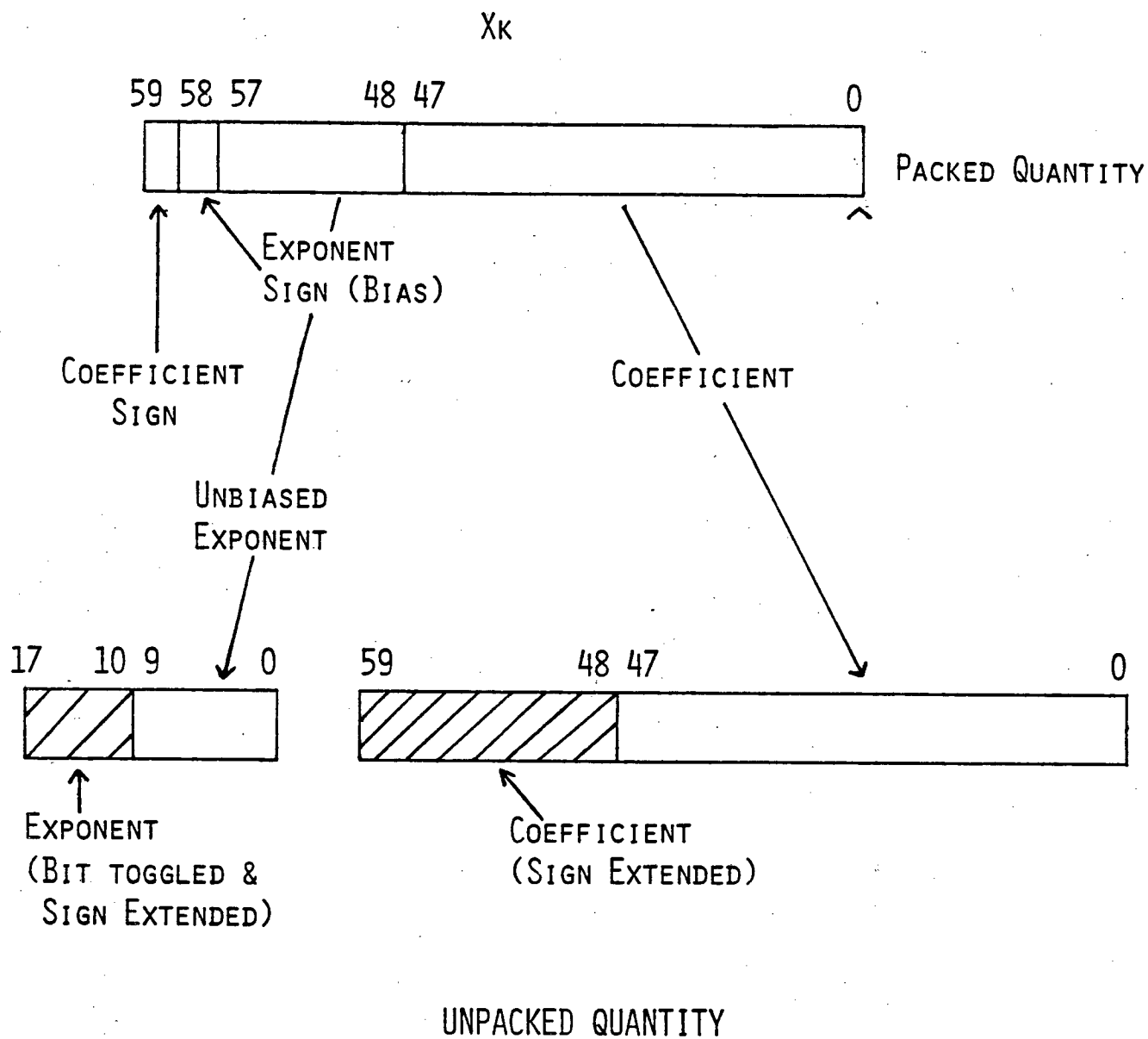
UX2 B2,X3 WOULD PRODUCE:

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

LX1 X2,B2 WOULD PRODUCE:

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

UNPACKING: UX_I BJ, X_k



HOW TO UNPACK A NUMBER

$$(X3) = 17224000000000000000_8$$

UX4 X3,B4

$$(X4) = \underbrace{00004}_{\text{SIGN OF COEF. EXTENDED}} \underbrace{000000000000000000}_{\text{COEF. FROM LOWER 48 BITS OF X3}}_8$$

SIGN OF	COEF. FROM
COEF.	LOWER 48
EXTENDED	BITS OF X3

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

TO UNNORMALIZE IT: (THERE IS NO UNNORMALIZE INSTRUCTION)

LX7 X4,B4

$$(X7) = 00000000000000000000 \rightarrow_8$$

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 EXPONENT 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

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 SIGNIFICANT BITS - UPPER 48 BITS.)

DX4 $X1+X2$ (X4 WILL RECEIVE THE LEAST SIGNIFICANT 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
STORAGE ALLOCATION.

ADDRESS LENGTH

0 11
11

BINARY CONTROL CARDS.

IDENT SAMPLE
END FLOP

ENTRY POINTS.

FLOP 00

LOAD MAP - SAMPLE

CYRER LOADER 1.4-470

06/27/70 09.01.04.

PAGE 1

FMA OF THE LOAD 111
LMA+1 OF THE LOAD 122

TRANSFER ADDRESS -- FLOP 111

PROGRAM ENTRY POINTS -- SAMPLE 111

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
SAMPLE	111	11	LGO	06/27/70	COMPASS 3.5	470	

.010 CP SECONDS

13200B CM STORAGE USED

1 TABLE MOVE

MFF NH2- CYH175-SN1 4LUT/H40 05/15/70
 09.01.02.DON005L FROM /SH
 09.01.02.IP 00000320 WORDS - FILE INPUT , DC 04
 09.01.02.DON. PSD.0270.72CT011A.MILLEN
 09.01.03.COMPASS.
 09.01.04. ASSEMBLY COMPLETE. 47300B SCH 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 00000496 WORDS - FILE OUTPUT , DC 40
 09.01.04.HS 3504 WORDS 1 716B MAX USED1
 09.01.04.CPA .121 SEC. .121 ADJ.
 09.01.04.ID .422 SEC. .622 ADJ.
 09.01.04.CH 9.539 MMS. .382 ADJ.
 09.01.04.SS 1.325
 09.01.04.PP 2.763 SEC. DATE 06/27/70
 09.01.04.F1 END OF ICH. CH

INCNVRT
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.19.06.

PAGE 1

ADDRESS LENGTH

0 2
2

BINARY CONTROL CARDS.

IDENT INCNVRT
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
27100
24211
1 26322
22423

INCNVRT SX0 123456R PLACE INTEGER IN X0
PX1 X0 PACK X0 INTO X1, B0 ASSUMED
HX2 X1.B1 NORMALIZE X1 INTO X2, SHIFT COUNT IN B1
UX3 X2.B2 UNPACK X2 INTO X3, SHIFT COUNT IN B2
LX4 X3.B2 LEFT SHIFT X3 INTO X4 USING COUNT IN B2

0200000000

JP 0 ABNORMAL TERMINATION
END INCNVRT TRANSFER ADDRESS

47306R SCH STORAGE USED
MODEL 174 ASSEMBLY

15 STATEMENTS
0.016 SECONDS

1 SYMBOLS
2 REFERENCES

DMPX.

P 000000 A0 000200 B0 000000
RA 220200 A1 000001 B1 000040
FL 000200 A2 000060 B2 777737
EM 700700 A3 000057 B3 012733
RE 000045 A4 000001 B4 000201
FE 000000 A5 000111 B5 000111
MA 000600 A6 000001 B6 000200
A7 000001 B7 027756
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 0000 0000 0004 0040 0000
X6 1505 1520 0000 0000 0061
X7 0000 0000 0000 0000 0000

C(A1)= 0000 0000 0000 0000 0000 C(A1)= 0000 0000 0000 0000 0000
C(A2)= 1505 1520 0000 0000 0061 C(A2)=
C(A3)= 0000 0000 0000 0000 0000 C(A3)=
C(A4)= 0000 0000 0000 0000 0000 C(A4)=
C(A5)= 7100 1234 5627 1002 4211 C(A5)= 7100 1234 5627 1002 4211
C(A6)= 0000 0000 0000 0000 0000 C(A6)=
C(A7)= 0000 0000 0000 0000 0000 C(A7)=

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
00100 54000 00000 01000 00001

51100 00001 03110 00055
00000 00200 00000 00001
00000 00000 00000 00113
00000 00000 00000 00000

44550 02550 00000 46000
07040 00060 51600 00001
40000 00000 02000 00111

00000 00000 00000 00000
04000 00063 00000 00021
00000 00000 40000 00000

5-26

INCNVRT
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 16.19.06.

PAGE 3

INCNVRT 0 PROGRAM 2/02 E 2/00 L

LOAD MAP - INCNVRT

CYRER 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	PROCESSR VER LEVEL	HARDWARE	COMMENTS
INCNVRT	111	2	LGO	06/27/78	COMPASS 3.5 470		

.007 CP SECONDS

132000 CM STORAGE USED

1 TABLE MOVE

MFF NR2- CYR175-SN1 4LH7/R6B 05/15/78
16.19.04.DONOME FROM /SH
16.19.04.1P 00000256 WORDS - FILE INPUT , DC 04
16.19.04.DON. PSQ.027A.72CT011A.MILLER
16.19.05.COMPASS.
16.19.06. ASSEMBLY COMPLETE. 47300R SCH USED.
16.19.06. 0.068 CPU SECONDS ASSEMBLY TIME.
16.19.06.LGO.
16.19.07.ERROR NONE =00. ADDRESS =000000
16.19.07.OP 00000704 WORDS - FILE OUTPUT , DC 40
16.19.07.HS 3584 WORDS (10752 MAX USED)
16.19.07.CPA .103 SEC. .103 ADJ.
16.19.07.10 .612 SEC. .612 ADJ.
16.19.08.CM 9.307 KMS. .567 ADJ.
16.19.08.SS 1.203
16.19.08.PP 3.295 SEC. DATE 06/27/78
16.19.08.EJ END OF JOB, SH

5-27

RIGINT
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 08.50.56.

PAGE 1

ADDRESS LENGTH

0 5
5

BINARY CONTROL CARDS.

IDENT RIGINT
END RIGINT TRANSFER ADDRESS

BLOCKS TYPE ADDRESS LENGTH

PROGRAM* LOCAL
LITERALS* LOCAL

0 4
4 1

ENTRY POINTS.

RIGINT 0*

LOAD MAP - RIGINT

CYBER LOADER 1.4-470

06/27/78 08.50.57.

PAGE 1

FMA OF THE LOAD 111
LMA+1 OF THE LOAD 116

TRANSFER ADDRESS -- RIGINT 111

PROGRAM ENTRY POINTS -- RIGINT 111

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK ADDRESS LENGTH FILE DATE PROCESSOR VER LEVEL HARDWARE COMMENTS

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

.008 CP SECONDS

132008 CM STORAGE USED

D MFF NB2- CYB175-SN1 4L07/R08 05/15/78
08.50.54.DON004J FROM /SH
D 08.50.54.1P 00000320 WORDS - FILE INPUT , DC 04
08.50.54.DON. PSD-0270.72CT011A-MILLER
D 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.
D 08.50.57.ERROR MODE =00. ADDRESS =000000
08.50.58.OP 00000896 WORDS - FILE OUTPUT , DC 40
D 08.50.58.MS 3504 WORDS (10752 MAX USED)
08.50.58.CPA .130 SEC. .130 ADJ.
08.50.58.10 .616 SEC. .616 ADJ.
D 08.50.58.CM 9.770 MWS. .596 ADJ.
08.50.58.SS 1.343
08.50.58.PP 3.526 SEC. DATE 06/27/78
D 08.50.58.EJ END OF JOB, SH

.....

CONVERSION OF A LARGE INTEGER TO FLOATING POINT TO INTEGER

.....

IDENT RIGINT
ENTRY RIGINT

CONVERSION OF A LARGE INTEGER (I.E. GREATER THAN 48 BITS) TO A
FLOATING POINT NUMBER AND BACK TO AN INTEGER

0 46000

RIGINT NO

LOAD NUMBER INTO X1

5110000004
1 6110000006

23211

27312

2 24423

9A1 1234567123456712348

SN1 6 PLACE SHIFT COUNT IN B1

AX2 X1.B1 SHIFT BEFORE PACKING

PX3 X2.B1 PACK USING SHIFT COUNT IN B1

MX4 X3.B2 NORMALIZE X3 INTO X4, SHIFT COUNT TO B2

PACKED-NORMALIZED NUMBER NOW IN X4, SHIFT COUNT IN B2

26534

22635

27701

3 0200000000

UX5 X4.B3 UNPACK X4 INTO X5, SHIFT COUNT TO B3

LX6 X5.B3 LEFT SHIFT I.E. FINISH UNPACKING

PX7 X1 PACKING WITHOUT SHIFTING

JP 0 ABNORMAL TERMINATION

END RIGINT TRANSFER ADDRESS

473008 SCM STORAGE USED
MODEL 174 ASSEMBLY

29 STATEMENTS
0.032 SECONDS

1 SYMBOLS
2 REFERENCES

DMPX.

P	000000	A0	000200	R0	000000
RA	240200	A1	000115	R1	000006
FL	000200	A2	000060	R2	000002
EM	700700	A3	000057	R3	000004
RE	000051	A4	000001	R4	000201
FE	000000	A5	000111	R5	000111
MA	001000	A6	000001	R6	000200
		A7	000001	R7	027756
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

C(A1)=	0012	3456	7123	4567	1234
C(A2)=	1505	1520	0000	0000	0061
C(A3)=	0000	0000	0000	0000	0000
C(A4)=	0000	0000	0000	0000	0000
C(A5)=	4600	0511	0000	1154	6000
C(A6)=	0000	0000	0000	0000	0000
C(A7)=	0000	0000	0000	0000	0000

C(R1)=	0000	0000	0000	0000	0000
C(R2)=	0000	0000	0000	0000	0000
C(R3)=	0000	0000	0000	0000	0000
C(R4)=					
C(R5)=	4600	0511	0000	1154	6000
C(R6)=					
C(R7)=					

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

51100	00001	03110	00055
00000	00200	00000	00001
00000	00000	00000	00116
00000	00000	00000	00000

A4550	02550	00000	46000
07040	00060	51600	00001
40000	00000	02000	00111

00000	00000	00000	00000
04000	00063	00000	00021
00000	00000	40000	00000

IDENT EXPONENT
ENTRY EXPONENT
LIST 1

RANGE OF EXPONENTS FOR VALID UNPACKING

UX1 RJ.XK
LX1 RJ.XK

IF TRUE EXPONENT, AND THUS RJ<-4R 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 $-47 \leq \text{TRUE EXPONENT} \leq +11$ (DECIMAL)

SHOW EXPONENTS OUT OF RANGE

0 5110000003 .
26221
22322

1 5140000004 .
26554
22655

2 0200000000

EXPONENT SA1 =20147654321000000000B LOAD VALUE INTO X1
UX2 X1.B2 UNPACK X1 INTO X2, SHIFT COUNT TO R2
LX3 X2.B2 LEFT SHIFT X2 INTO X3, R2 PLACES

SA4 =17177654321000000000B LOAD VALUE INTO X4
UX5 X4.B5 UNPACK X4 INTO X5, SHIFT COUNT TO R5
LX6 X5.B5 LEFT SHIFT X5 INTO X6, BY R5 PLACES

JP 0 ABNORMAL TERMINATION DMPX.

CONTENT OF LITERALS BLOCK.

3 20147654321000000000
4 17177654321000000000

PLN=ZH
OON=ZH

5

END EXPONENT TRANSFER ADDRESS

47300B SCM STORAGE USED
MODEL 174 ASSEMBLY

36 STATEMENTS
0.036 SECONDS

1 SYMBOL
2 REFERENCE

P	000000	A0	000200	R0	000000
RA	125200	A1	000114	R1	000001
FL	000200	A2	000060	R2	000014
EM	700700	A3	000057	R3	012733
RE	000041	A4	000115	R4	000201
FE	000000	A5	000111	R5	777717
MA	000400	A6	000001	R6	000200
		A7	000001	R7	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
00000	00000	00000	00000	00000

EXPONET
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 08.52.31.

PAGE 1

ADDRESS LENGTH

0 5
5

BINARY CONTROL CARDS.

IDENT EXPONET
END EXPONET TRANSFER ADDRESS

BLOCKS TYPE ADDRESS LENGTH

PROGRAM* LOCAL 0 3
LITERALS* LOCAL 3 2

ENTRY POINTS.

EXPONET 0*

EXPONET
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 08.52.31.

PAGE 3

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	PROCESSOR VER LEVEL
EXPONET	111	5	LGO	06/27/78	COMPASS 3.5 470

.007 CP SECONDS

13200B CM STORAGE USED

MFF NB2- CYR175-SNI 4L87/R6B 05/15/78
08.52.29.DON0045 FROM /SM
08.52.29.1P 00000384 WORDS - FILE INPUT , DC 04
08.52.29.DON. PSD-0278.72CT011A-MILLER
08.52.31.COMPASS.
08.52.31.ASSEMBLY COMPLETE. 47300B SCH 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.HS 3584 WORDS (10752 MAX USED)
08.52.32.CPA .131 SEC. .130 ADJ.
08.52.32.10 .619 SEC. .619 ADJ.
08.52.32.CH 9.982 MWS. .609 ADJ.

5-31

IDENT TESTFPO
ENTRY TESTFPO
LIST *

TEST FLOATING POINT ZERO

0 5110000005 *
10011
1 5110000006 *
31211
40321
2 44421
44512
30651
30761

TESTFPO SA1 =0.0
RX0 X1
SA1 =3.5
FX2 X1-X1
FX3 X2*X1
FX4 X2/X1
FX5 X1/X2
FX6 X5*X1
FX7 X6*X1

PRODUCE 0.0 IN X2
ZERO TIMES A NUMBER
ZERO DIVIDED BY A NUMBER
NUMBER DIVIDED BY ZERO
USE PREVIOUS ANSWER
USE PREVIOUS ANSWER

ENDRUN

3
3 7160247021
20650
4 0100000000 X

SK6 JREND*4.1
LX6 400
PJ =XSYS=
ENDM

ENDRUN .1
ENDRUN .1
ENDRUN .1
ENDRUN .1

DEFAULT SYMBOLS OFFINED BY COMPASS.

0 X

SYS=

CONTENT OF LITERALS BLOCK.

5 00000000000000000000
6 17217000000000000000

00*

7

END TESTFPO TRANSFER ADDRESS

473008 SCM STORAGE USED
MODEL 174 ASSEMBLY

21 STATEMENTS
0.023 SECONDS

2 SYMBOLS
3 REFERENCES

DMPX.

P	000000	A0	000200	R0	000000	C(A1)=	1721	7000	0000	0000	0000	C(A1)=	0000	0000	0000	0000	0000
RA	133400	A1	000117	R1	000001	C(A2)=	1505	1520	0000	0000	0061	C(A2)=	0000	0000	0000	0000	0000
FL	000200	A2	000060	R2	000002	C(A3)=	0000	0000	0000	0000	0000	C(A3)=					
EM	700700	A3	000057	R3	012733	C(A4)=	0000	0000	0000	0000	0000	C(A4)=					
RE	000045	A4	000001	R4	000201	C(A5)=	0000	0000	0000	0000	0000	C(A5)=					
FE	000000	A5	000111	R5	000111	C(A6)=	5110	0001	1610	0114	6000	C(A6)=	5110	0001	1610	0114	6000
MA	000400	A6	000001	R6	000200	C(A7)=	0000	0000	0000	0000	0000	C(A7)=					
		A7	000001	R7	027756	C(A7)=	0000	0000	0000	0000	0000	C(A7)=					
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

00000 00000 00000 00000

00000 00000 00000 00000

00000 00000 00000 00000

00000 00000 00000 00000

00000 00000 00000 00000

00000 00000 00000 00000

00000 00000 00000 00000

00000 00000 00000 00000

TESTFPO
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 08.51.01.

PAGE 1

ADDRESS LENGTH

0 7
7

BINARY CONTROL CARDS.

IDENT TESTFPO
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

FMA 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	PROCESSR	VER	LEVEL	HARDWARE
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 4LB7/R6B 05/15/78
08.50.58.DON004L FROM /SM
08.50.58.IP 00000256 WORDS - FILE INPUT , DC 04
08.50.58.DON. PSD.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.I0 .676 SEC. .676 ADJ.
08.51.02.CH 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

IDENT ROUND
ENTRY ROUND
LIST \$

EXAMPLE SHOWING A ROUND NORMALIZE INSTRUCTION

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

47500B CM

STORAGE USED
MODEL 74 ASSEMBLY

23 STATEMENTS
0.070 SECONDS

3 SYMBOLS
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					

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 SC/R08 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.I0 .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
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 08.57.39.

PAGE 1

ADDRESS LENGTH

BINARY CONTROL CARDS.

0 10
10

IDENT INTMULT
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
7110000024
1 27200
27301
42423
26504
2 42601
4072J
3 46000
4 24402
2450J
40645
42745
5 46000
6
6 7160247021
20650
7 0100000000 X

INTMULT SX0 13H
SX1 24H
PX2 X0
PX3 X1
DX4 X2*X3
UX5 X4
DX6 X0*X1
FX7 X2*X3
IRKONE NO
NX4 X2
NX5 X3
FX6 X4*X5
DX7 X4*X5
IRKTWO NO
ENDRUN
SX6 3REND*4+1
LX6 400
RJ =XSYS=
ENDM

PLACE 13 OCTAL IN X0
PLACE 24 OCTAL IN X1
CONVERT 13 OCTAL TO A PACKED FLOATING POINT NUMBER
CONVERT 24 OCTAL TO A PACKED FLOATING POINT NUMBER
DOUBLE PRECISION MULTIPLY -
LEAST SIGNIFICANT DIGITS IN X4
UNPACK X4 CONTENTS INTO X5 -
SHIFT COUNT GOING TO 80
DOUBLE PRECISION MULTIPLY OF 13H AND 24H
SINGLE PRECISION MULTIPLY OF X2 AND X3 CONTENTS

ENDRUN .1
ENDRUN .1
ENDRUN .1
ENDRUN .1

DEFAULT SYMBOLS DEFINED BY COMPASS.

0 X

SYS=

10

END INTMULT

473008 SCM STORAGE USED

MODEL 174 ASSEMBLY

30 STATEMENTS

0.035 SECONDS

4 SYMBOLS

5 REFERENCES

TRAP DIRECTIVES

DERUG 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

DERUG 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                                (A1)=11162415251424005202    R1=000001    (R1)=00000000000000000000    X0=00000000000000000013
*   A1=000305                                (A2)=00000000000000000000    R2=005202    (R2)=71000000137110000024    X1=00000000000000000024
*   A2=000303                                (A3)=00000000000000001234    R3=001234    (R3)=34555555555555555555    X2=20000000000000000013
*   A3=000300                                (A4)=04000005010000000000    R4=001234    (R4)=34555555555555555555    X3=20000000000000000024
*   A4=000170                                (A5)=61100000017160241115    R5=002212    (R5)=00000000000000000001    X4=20000000000000000034
*   A5=000434                                (A6)=00000000000000001234    R6=000333    (R6)=00000000000000000001    X5=00000000000000000034
*   A6=000317                                (A7)=32323232323543000175    R7=000334    (R7)=00000050000000000000    X6=00000000000000000034
*   A7=000326                                (A7)=32323232323543000175    R7=000334    (R7)=00000050000000000000    X7=20600000000000000000
*****
```

005200 24220103135504251520 55031716241116250504 71000000137110000024 27200273014242324504 TRACK DUMP CONTINUED* K+M TMRB>A77ZAD
005204 42A01407236100046000 01000023664000001224 24402245034064542745 01000023664000001230 7ELGSI DE A SvS JTTSR+C5#W+A SvS JK

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                                (A1)=11162415251424005202    R1=000001    (R1)=00000000000000000000    X0=00000000000000000013
*   A1=000305                                (A2)=00000000000000000000    R2=005202    (R2)=71000000137110000024    X1=00000000000000000024
*   A2=000303                                (A3)=00000000000000001234    R3=001234    (R3)=34555555555555555555    X2=20000000000000000013
*   A3=000300                                (A4)=04000005010000000000    R4=001234    (R4)=34555555555555555555    X3=20000000000000000024
*   A4=000170                                (A5)=61100000017160241115    R5=002212    (R5)=00000000000000000001    X4=17235400000000000000
*   A5=000434                                (A6)=00000000000000001562    R6=000333    (R6)=00000000000000000001    X5=17245000000000000000
*   A6=000317                                (A7)=32323232323543000175    R7=000334    (R7)=00000050000000000000    X6=17276700000000000000
*   A7=000326                                (A7)=32323232323543000175    R7=000334    (R7)=00000050000000000000    X7=16470000000000000000
*****
```

5-37

INTMULT
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 08.57.39.

PAGE 3

INTMULT	0	PROGRAM	2/02 I	2/03 L
SYS	0	EXTERNAL	2/2A	
TRAPONE	3	PROGRAM	2/19 L	
TRAPTWO	5	PROGRAM	2/24 L	

LOAD MAP - TRAPPER

CYBER LOADER 1.4-470

06/27/78 08.57.42.

PAGE 1

FMA OF THE LOAD 111
LMA-1 OF THE LOAD 5252

TRANSFER ADDRESS -- INTMULT 5202
DEBUG ENTRY USED -- TRAPSETH 436

PROGRAM ENTRY POINTS -- TRAPPER 5202

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSOR 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

17600H CM STORAGE USED

1 TABLE MOVE

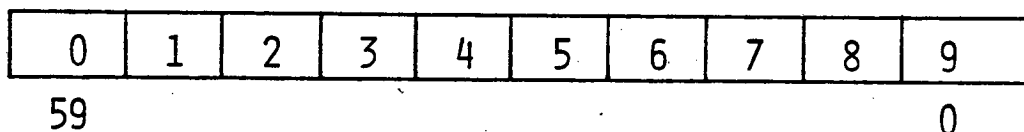
MPF NR2- CYB175-SN1 4L07/R68 05/19/78
08.57.36.DON0049 FROM /SM
08.57.36.JP 00000320 WORDS - FILE INPUT - DC 04
08.57.36.DON. PSN.0278.72CT011A.MILLER
08.57.39.TRAPIL=PUTOUTI
08.57.39.COMPASS.
08.57.40. ASSEMBLY COMPLETE. 47300H SCM USED.
08.57.40. 0.080 CPU SECONDS ASSEMBLY TIME.
08.57.40.LGO.
08.57.42.REWIND(PUTOUTI)
08.57.42.COPYRF. 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 1 17920 MAX USED)
08.57.43.CPA .211 SEC. .211 ADJ.
08.57.43.10 .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

5-138

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.

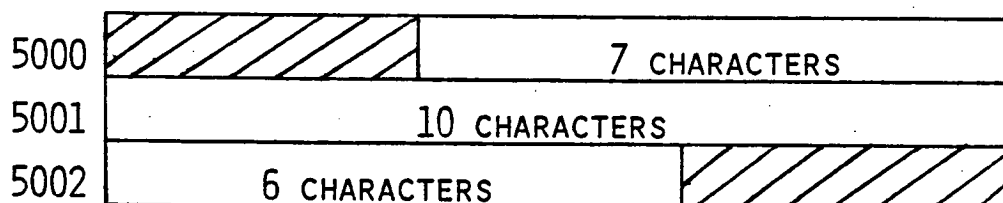
THERE ARE TEN 6-BIT CHARACTER POSITIONS IN EACH 60-BIT WORD. THESE POSITIONS ARE NUMBERED 0 THROUGH 9 FROM LEFT TO RIGHT RESPECTIVELY. THE 4-BIT CHARACTER ADDRESSES OF THESE POSITIONS ARE (IN BINARY) 0000, 0001,, 1000, AND 1001. CHARACTER ADDRESSES 1010 THROUGH 1111 ARE ILLEGAL AND CAUSE THE INSTRUCTIONS 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 IS 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 $(BJ) \pm K$. 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 $0 \leq L \leq 8191$, 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 $0 \leq L \leq 127$, 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.

USE P SCALING FOR UNNORMALIZED NUMBERS

24 17250000000000000000
 25 200000000000000000050
 26 200300000000000000005
 27 177400000000000000000
 30 000000000000000000000
 31 1771000000000000000002

DATA 40.0 NORMALIZED FLOATING POINT NUMBER
 DATA 40.P0 UN-NORMALIZED FLOATING POINT NUMBER (POSITIONED)
 DATA 40.P-3 SHIFTED RIGHT
 DATA 40.P3 SHIFTED LEFT
 DATA 0.04P0 NO SIGNIFICANT BITS IN REGISTER
 DATA 0.04P6 LEFT SHIFT

EVALUATION UNDER IS F THEN S THEN P

32 1776000000000000000002
 33 60017777777777777775
 34 000000000000000000000
 35 000000000000000000000
 36

DATA 7.2P15-5E1
 DATA -7.2P15-5E1
 DATA DEES

END

47300R SCH STORAGE USED
 MODEL 174 ASSEMBLY

77 STATEMENTS
 0.066 SECONDS

0 SYMBOLS
 0 REFERENCES

1 ERROR IN NUMDATA

NUMDATA
 ERROR DIRECTORY.

A TYPE ERROR
 OCCURRED ON PAGES

ADDRESS FIELD HAD.
 3

MFF NR2- CYR175-SN1 4LR7/R6R 05/15/78
 07.34.57.000001W FROM /SCH
 07.34.57.1P 00000640 WORDS - FILE INPUT - DC 04
 07.34.57.00N. PSC-027P.72CT011A.MILLER
 07.34.59.COMPASS.
 07.34.59. 1 ERROR IN NUMDATA
 07.34.59. ASSEMBLY ERRORS. 47300R SCH USED.
 07.34.59. 0.116 CPU SECONDS ASSEMBLY TIME.
 07.34.59.0P 00001000 WORDS - FILE OUTPUT - DC 40
 07.34.59.45 1584 WORDS 1 10752 MAX USED)
 07.34.59.CPA .143 SEC. .143 ADJ.
 07.34.59.10 .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

IDENT DATA
ENTRY DATA
LIST 8

USE OF DATA PSFIND OP *****CHARACTER DATA*****

0 01020304050600000000
1 01020304050607100000
2 00000000000000000000

TYPEC

DATA ACARCOFF LEFT JUSTIFIED WITH ZERO FILL. FOR DATA ITEMS OR
DATA ACARCOFFGHI LITERALS, 12 ZERO BITS ARE GUARANTEED AT THE END OF
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
4 01020304050607101155

TYPEH

DATA AHARCOFF LEFT JUSTIFIED WITH BLANK FILL
DATA AHARCOFFGHI LEFT JUSTIFIED WITH BLANK FILL
SPACE 2

5 55555555010203040506
6 55010203040506071011

TYPEA

DATA AAARCOFF RIGHT JUSTIFIED WITH BLANK FILL
DATA AAARCOFFGHI RIGHT JUSTIFIED WITH BLANK FILL
SPACE 2

7 00000000010203040506
10 00010203040506071011

TYPER

DATA ARARCOFF RIGHT JUSTIFIED WITH ZERO FILL
DATA ARARCOFFGHI RIGHT JUSTIFIED WITH ZERO FILL
SPACE 2

11 01020304050600000000
12 01020304050607101100

TYPEL

DATA ALARCOFF LEFT JUSTIFIED WITH ZERO FILL
DATA ALARCOFFGHI LEFT JUSTIFIED WITH ZERO FILL
SPACE 2

13 01020304050600000000
14 01020304050607101100
15 01020304050607101112
16 00000000000000000000

TYPEZ

DATA AZARCOFF LEFT JUSTIFIED WITH ZERO FILL. FOR DATA ITEMS OR
DATA AZARCOFFGHI LITERALS, SIX ZERO BITS ARE GUARANTEED AT THE END
DATA AZARCOFFGHIJ OF THE STRING EVEN IF ANOTHER WORD MUST BE ALLO-

17 01020304050600000000
20 76757473727177777777
21 01020304050600000000

DATA -6CAHCOFF SIGN,N,TYPE,STRING
DATA -6CAHCOFF SIGN,N,TYPE,STRING
DATA -6CAHCOFF SIGN,N,TYPE,STRING
SPACE 1

22 04050601020355555555
23 73727176757422222222
24 04050601020355555555

DATA -H*DEFARCO SIGN,TYPE,DELIMITER,STRING,DELIMITER
DATA -H*DEFARCO SIGN,TYPE,DELIMITER,STRING,DELIMITER
DATA -H*DEFARCO SIGN,TYPE,DELIMITER,STRING,DELIMITER
SPACE 1

25 5110010703

SAI 7HARC N,TYPE STRING
SPACE 1

26 5110000067 .
5110000070 .
27 5110000067 .

DATA

SAI -4AUEFG -SIGN,N,TYPE,STRING
SAI -4AUEFG -SIGN,N,TYPE,STRING
SAI -4AUEFG -SIGN,N,TYPE,STRING

EJECT

USE OF DATA PSEUDO OP *****NUMERIC DATA*****

30 0000000000000000125
 31 77777777777777777774
 32 17204000000000000000
 33 17225000000000000000
 34 17245046314631463146
 35 17245046314631463146
 36 60532731463146314631
 37 17244060000000000000
 40 60533717777777777777
 41 17244060000000000000
 42 17136314631463146315
 43 17307640000000000000
 44 17307640000000000000
 45 16500000000000000000
 46 17307640000000000000
 47 16500000000000000000
 50 60470137777777777777
 51 61277777777777777777
 52 20000000000000000001
 53 57777777777777777776
 54 177400000000000000400
 55 17200400000000000000

DATA R5 VALUE
 DATA -3 SIGN.VALUE
 DATA 1.0 SIGN.VALUE
 DATA 5.0 SIGN.VALUE
 DATA 020.3 SIGN.PREHADIX.VALUE
 DATA +020.3 SIGN.PREHADIX.VALUE
 DATA -020.3 SIGN.PREHADIX.VALUE
 DATA 020.3 SIGN.PREHADIX.VALUE
 DATA -020.3 SIGN.PREHADIX.VALUE
 DATA +020.3 SIGN.PREHADIX.VALUE
 DATA 5.0E-2 SIGN.VALUE.MODIFIER
 DATA 5.0E2 SIGN.VALUE.MODIFIER
 DATA 5.0EE2 SIGN.VALUE.MODIFIER
 DATA -5.0EE2 SIGN.VALUE.MODIFIER
 DATA 1.0P0 SIGN.VALUE.MODIFIER
 DATA -1.0P0 SIGN.VALUE.MODIFIER
 DATA 3.2E1P3 SIGN.VALUE.MODIFIER
 DATA 0.1BP47 SIGN.VALUE.MODIFIER

N IS OMITTED

56 555555555555111605
 57
 57 05222217225511165520
 60 04210000000000000000

DATA A*MINE*
 DATA H**
 DATA L*ERROR IN PDQ*

N=0

61 55555555555512171016
 62 12171016000000000000
 63
 63 7110021714
 7120010203
 64 7130000060
 65 00000000000000000000
 66 00000000000000000123

DATA 0AJ0HN
 DATA 0LJOHN DOE SPACE ACTS AS A DELIMITOR
 DATA 0L JOHN DOE SPACE ACTS AS A DELIMITOR
 SX1 0RSYMBOL
 SX2 0HAB0DEFGHIJ
 SX3 1R.01
 DATA 1R.01
 DATA 123B

CONTENT OF LITERALS BLOCK.

67 55555555555504050607
 70 22222222222271721170

DEFG
 RRRRRR>44+

71

END

DATA
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 08.49.25.

PAGE 1

ADDRESS LENGTH

0 71
71

BINARY CONTROL CARDS.

IDENT DATA
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
ERROR DIRECTORY.

COMPASS 3.5-470.

06/27/78 08.49.25.

PAGE 5

A TYPE ERROR
OCCURRED ON PAGES

ADDRESS FIELD BAD.
3

T TYPE ERROR
OCCURRED ON PAGES

ADDRESS VALUE EXCEEDS FIELD SIZE, RESULT TRUNCATED
3

DATA
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 08.49.25.

PAGE 6

DATA	26	PROGRAM*	2/02 F	2/54 L
MINF	46	PROGRAM*	1/46 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	
TYPEP	7	PROGRAM*	2/26 L	
TYPEZ	13	PROGRAM*	2/36 L	

MFF NP2- CYH175-SN1 4LH7/RAH 05/15/70
 00.49.23.DONOR040 FROM /SH
 00.49.23.IP 00000576 WORDS - FILE INPUT , DC 04
 00.49.23.DON. PSD-027R.72CTR11A.MILLER
 00.49.25.COMPASSILO=11111
 00.49.25. 2 WARNING MESSAGES IN DATA
 00.49.25. 1 ERROR IN DATA
 00.49.26. ASSEMBLY ERRORS. 47300R SCH USED.
 00.49.26. 0.139 CPU SECONDS ASSEMBLY TIME.
 00.49.26.OP 00001216 WORDS - FILE OUTPUT , DC 40
 00.49.26.WS 3584 WORDS 1 716R MAX USED
 00.49.26.CPA .166 SEC. .166 ADJ.
 00.49.26.I0 .491 SEC. .491 ADJ.
 00.49.26.CH 9.933 KWS. .606 ADJ.
 00.49.26.SS 1.264
 00.49.26.PP 2.639 SEC. DATE 06/27/70
 00.49.26.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
1 00000000000000000000
2 00000000000000000000
3 00000000000000000024
4 00000000000000000055
5 00000000000000000033
6 00000000000000000064
7 00000000000000000033
10 33000000000000000000
11 33555555555555555555
12 55555555555555555533
13 00000000000000000060
14 00000000000000000000
15 7160247021
17

CON CON 0
CON CON 4
CON CON 1
CON CON 20
CON CON 1R
CON CON 1H0
CON CON 1R0
CON CON 1H0,1L0,1H0,1L0

CON 1R.01
DATA 1R.01
ENDRUN
END CON

47300R SCM STORAGE USED
MODEL 174 ASSEMBLY

40 STATEMENTS
0.044 SECONDS

2 SYMBOLS
1 REFERENCES

1 ERROR IN CON

CON
ERROR DIRECTORY.

A TYPE ERROR
OCCURRED ON PAGES

ADDRESS FIELD HAD.
2

CON
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.29.23.

PAGE 1

ADDRESS	LENGTH	BINARY CONTROL CARDS.
0	17	IDENT CON
17		END CON

ENTRY POINTS.

CON 0*

EXTERNAL SYMBOLS.

SYS*

CON
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 16.29.23.

PAGE 4

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

6-9

MFF NR2- CY8175-SN1 4L87/R6B 05/15/78
 16.29.20.DON00MU FROM /SH
 16.29.20.1P 00000384 WORDS - FILE INPUT , DC 04
 16.29.20.DON. PSD.02/8.72CT011A.MILLER
 16.29.22.COMPASS.
 16.29.23. 1 ERROR IN CON
 16.29.23. ASSEMBLY ERRORS. 47300R SCH 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.
 16.29.23.10 .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

DIS
STORAGE ALLOCATION.

ADDRESS LENGTH

0 22
22

BINARY CONTROL CARDS.

IDENT DIS
END FIRST

ENTRY POINTS.

FIRST 0*

EXTERNAL SYMBOLS.

SYS=

DIS

IDENT DIS
ENTRY FIRST

USE OF DIS PSEUDO INSTRUCTION

•
•
•
•
•

0	01020304050607101112	FIRST	LIST 0
1	13010203040506071011		DIS 2:ARCEDEFGHIJKABCEFGHIJ
2	01141405160114140516		DIS 2:ALLENALLENALLEN
3	01141405165555555555		DIS 0:HOWDY
4	10172704310000000000		DIS 3/COMPASS EXAMPLE GOOD/
5	03171520012323550530		
6	01152014055507171704		
7	00000000000000000000		

USE OF ASSZ PSEUDO OPERATION

•
•
•

10	20	20	71A0247021	SAVE ASSZ A
				SAVEONE ASSZ
				ENDRUN

DEFAULT SYMBOLS DEFINED BY COMPASS.

SYS=

END FIRST

22

47300R SCH STORAGE USED
MODEL 174 ASSEMBLY22 STATEMENTS
0.026 SECONDS4 SYMBOLS
5 REFERENCES

*DIS for
Display Code*

01-9

D15
SYMBOLIC REFERENCE TABLE.

COMPASS 1.5-470.

06/27/70 16.29.15.

PAGE 1

FIRST	0	PROGRAM	2/02 F	2/09 L
SAVE	10	PROGRAM	2/20 L	
SAVEONE	20	PROGRAM	2/21 L	
SYS	0	INTERNAL	2/23	

MFT NR2- CYR175-SNI ALB7/R60 05/15/70
 16.29.32.00N00MV FROM /SH
 16.29.32.1P 00000256 WORDS - FILE INPUT + DC 04
 16.29.32.00N. PSD.0270.72C1011A.MILLER
 16.29.34.COMPASS.
 16.29.35. ASSEMBLY COMPLETE. 47300B SCH USED.
 16.29.35. 0.071 CPU SECONDS ASSEMBLY TIME.
 16.29.35.0P 00000440 WORDS - FILE OUTPUT + DC 40
 16.29.35.M5 3504 WORDS 1 10752 MAX USED
 16.29.35.CPA .095 SEC. .095 ADJ.
 16.29.35.10 .470 SEC. .470 ADJ.
 16.29.35.CH 7.947 MW. .484 ADJ.
 16.29.35.SS 1.051
 16.29.35.PP 3.140 SEC. DATE 06/27/70
 16.29.35.EJ END OF JOB. SH

IDENT LITERAL
ENTRY ST

USE OF LITERAL PSEUDO OPERATION

0 5130000020 . ST
5140000021 .
1 30734 24707 5170000017 .
2 5110000022 . 31711
3 5170000011 . 24707
4 5170000012 . 5120000023 .
5 30621 24606 5160000014 .
6 40712 5170000015 . 44612
7 5160000016 .
10 20650 7160247071
0100000000 X
11 777777777777777777
12 6

LIST L+R+G+D
SA1 =1.5
SA4 =1.0
FX7 X1-X4
NX7 X7
SA7 SAVE+6
SA1 =0.0
FX7 X1-X1
SA7 SAVE
NX7 X7
SA7 SAVE+1
SA2 =1.5
FX6 X2-X1
NX6 X6
SA6 SAVE+3
FX7 X1-X2
SA7 SAVE+4
FX6 X1/X2
SA6 SAVE+5
ENDRUN

SX6 JREND+4+1
LX6 400
RJ =XSYS=
SAVE DATA -0
RSSZ 6

DEFAULT SYMBOLS DEFINED BY COMPASS.

0 X

SYS=

CONTENT OF LITERALS BLOCK.

20 17206000000000000000
21 17204000000000000000
22 00000000000000000000
23 17217000000000000000

00+

24

END ST

47300R SCH STORAGE USED
MODEL 174 ASSEMBLY

31 STATEMENTS
0.036 SECONDS

3 SYMBOLS
10 REFERENCES

HFF NR2- CYB175-SN1 4L07/R6R 05/15/78
16.30.56.DON000R FROM /SH
16.30.56.IP 00000320 WORDS - FILE INPUT - DC 0
16.30.56.DON. PS0.0278.72CT011A.MILLER
16.30.58.COMPASS.
16.31.00. ASSEMBLY COMPLETE. 47300R SCH USED.
16.31.00. 0.081 CPU SECONDS ASSEMBLY TIME.
16.31.00.L60.
16.31.01.DMP(111,137)
16.31.01.EXIT.
16.31.01.OP 00000096 WORDS - FILE OUTPUT - DC 4
16.31.01.MS 3584 WORDS 1 7168 MAX USED)
16.31.01.CPA .127 SEC. .127 ADJ.
16.31.01.10 .632 SEC. .632 ADJ.
16.31.01.CH 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
SYMBOLIC REFERENCE TABLE.

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

LOAD MAP - LITERAL

CYBER LOADER 1.4-470

06/27/78 16.31.01.

PAGE 1

LWA 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	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
LITERAL	111	24	LGO	06/27/78	COMPASS 3.5 470		
SYS.PM	135	40	SL-SYSLIB	05/16/78	COMPASS 3.5 470		PROCESS SYSTEM REQUEST.

.023 CP SECONDS

13200R CM STORAGE USED

1 TABLE MOVE

DUMP RELATIVE

DMP(1111.137)

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	71607	47021	20650	01000	00137	46000	00000	00000	00000	00000	00125	17217	00000	00000
00124	00000	00000	00000	00000	00130	17215	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.

ADDRESS	LENGTH
0	24
24	

BINARY CONTROL CARDS.

IDENT	LITERAL
END	ST

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	000000000000000000000000	SAM	DATA	1000
	25	CHECK1	FOU	250
1	07537	TAG	VFD	15/7537H
	10711		RR7	X1
	0000000000		VFD	42/CHECK1
2	0025		VFD	42/OLMYFILE
	15310611140500		VFD	1A/1001H
	00			
3	1001		SA2	CHKLIT1
	5120000032		VFD	42/OLMYFILE,1A/1001H
	15310			
4	611140500001001		SA2	CHKLIT+1
5	5120000033		VFD	27/1.10/2
	0000000010			
6	010		VFD	40/CHECK1
	00000000000052		VFD	10/3.30/4
	00030			
7	000000004			
10	5130000001	TAG1	SA1	TAG
	0102032301		VFD	1A/3HARC,24/4HSAM,1A/0
11	1555000000		VFD	17/3HABC
12	02040A		VFD	1A/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	7.5HABCD,2
	43		LIT	7.5HABC
	45		LIT	4.5HABC,2
	47		LIT	7.5HABD,2
	5130000032		SA1	=7
15	5140000031		SA4	=10.5E4
	5150000042		SA5	=2

EXAMPLE OF CONSTANTS

16	5211000055	SA1	X1.1H
	5121010203	SA2	01.3HABC
17	5132010255	SA3	02.3LAB
	5143010203	SA4	03.3HABC
20	5154020304	SA5	04.4HABCD
	5120517252	SA2	0L171
21	6110010455	SN1	3LAB
	6120010203	SN2	3HABC
22	6130020304	SN3	4HABCD

VFD

COMPASS 3.5-470.

06/27/76 09.03.48.

PAGE 1

102030 TWO FOUR 1LARC
 102030 THREE FOUR 4LARC
 104000 FOUR SET 1LAD
 102000 FIVE SET 2LAD
 SIX FOUR 1.00
 7 SEVEN SET 7
 0 EIGHT SET 0
 0 NINE SET 0
 12 TEN SET 10
 13 ELEVEN SET 11
 -0 LARGE = 777777H

EXAMPLE OF ADDRESS EXPRESSIONS

5110000006 *
 23 5110000010 *
 5110777774 *
 24 6120000000
 6120000011
 25 6140777773
 5110000052 *
 26 5110000052 *
 5110000053 *
 27 6110000000
 71A0247021
 30 20650
 0100000000 X

ENDRUN

SX6 JREND*4.1
 LX6 400
 RJ *XSYS*
 ENDM

ENDRUN .1
 ENDRUN .1
 ENDRUN .1
 ENDRUN .1

DEFAULT SYMBOLS DEFINED BY COMPASS.

SYS=

CONTENT OF LITERALS BLOCK.

0 X
 11 17406321200000000000
 32 00000000000000000007
 33 00000000000000000001
 34 00000000000000000000
 35 0000000000000000003720
 36 00000000000000000007
 37 01020356355555555555
 40 00000000000000000007
 41 01020304065555555555
 42 00000000000000000007
 43 00000000000000000007
 44 01020355555555555555
 45 00000000000000000004
 46 01020356355555555555
 47 00000000000000000007
 50 01020456355555555555
 51 17216220771740156067
 52 16413641652363756556
 53 24051603100122032423
 54 00000000000000000000

0580P

G
A

4P

G

ABC.2

G

ABCD

H

G

ABC

D

ABC.2

G

ABD.2

001P105MEX

N616-SX2A.

TENCHARCT5

VFD
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 09.03.48.

PAGE 1

ADDRESS	LENGTH	BINARY CONTROL CARDS.
0	55	IDENT VFD
55		END VFD

HLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM*	LOCAL	0	31
LITERALS*	LOCAL	31	24

ENTRY POINTS.

VFD 17.

EXTERNAL SYMBOLS.

SYS=

VFD

COMPASS 3.5-470.

06/27/78 09.03.48.

PAGE 4

55

END VFD

47300R SCH STORAGE USED
MODEL 174 ASSEMBLY

81 STATEMENTS
0.091 SECONDS

19 SYMBOLS
40 REFERENCES

VFD
ERROR DIRECTORY.

COMPASS 3.5-470.

06/27/78 09.03.48.

PAGE 5

2 TYPE ERROR
OCCURRED ON PAGES

ADDRESS ERROR ON SYMBOL DEFINITION.
1

7 TYPE ERROR
OCCURRED ON PAGES

ADDRESS VALUE EXCEEDS FIELD SIZE. RESULT TRUNCATED
2. 3

MFF NR2- CYR175-SNI 4LB7/R6B 05/15/78
09.03.41.DON005R FROM /SH
09.03.42.IP 00000384 WORDS - FILE INPUT , DC (,
09.03.42.DON. PSD.0278.72CT011A.MILLER
09.03.47.COMPASSILO=8888)
09.03.48. 4 WARNING MESSAGES IN VFD
09.03.48. ASSEMBLY COMPLETE. 47300H SCH 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 (,
09.03.50.HS 3584 WORDS (7168 MAX USED)
09.03.50.CPA .254 SEC. .254 ADJ.
09.03.50.10 .659 SEC. .659 ADJ.
09.03.50.CH 12.164 MWS. .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

97-9

VFD
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 09.03.49.

PAGE 6

CHECK1	25		2/07 D	2/10	2/21	
CHKL11	12	PROGRAM*	2/18	2/31 L	2/34	2/35
CHKL111	32	PROGRAM*	2/15	2/34 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	3/14	3/19	3/20
ONE	102030		2/57 D			
SAM	0	PROGRAM*	2/06 L			
SEVEN	7		3/06 D	3/14	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

PWA 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	PROCESSOR VEN 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

13200R CM STORAGE USED

1 TABLE MOVE

DMPK.

P	000000	A0	000100	R0	000000
PA	123200	A1	000055	R1	000001
FL	000100	A2	010204	R2	000002
EM	700700	A3	000143	R3	012733
PE	000041	A4	000142	R4	000301
FE	000000	A5	000153	R5	000124
MA	000400	A6	000001	R6	000100
		A7	000001	R7	027756
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

C1A1)=	5110	0000	0101	1100	0055
C1A2)=					
C1A3)=	0000	0000	0000	0000	0007
C1A4)=	1740	6321	2000	0000	0000
C1A5)=	0000	0000	0000	0000	0002
C1A6)=	0000	0000	0000	0000	0000
C1A7)=	0000	0000	0000	0000	0000

C1R1)=	0000	0000	0000	0000	0000
C1R2)=	0000	0000	0000	0000	0000
C1R3)=					
C1R4)=					
C1R5)=	5110	0001	4251	2000	0143
C1R6)=					
C1R7)=					

00000	00010	00130	00000	00000
00054	56110	01110	00054	54710
00060	15051	52000	00000	00061
00064	14071	70000	00000	00000
00070	14071	75755	00000	00000
00100	54000	00000	01000	00001

00000	00000	00000	00000
51100	00001	03110	00055
00000	00300	00000	00001
00000	00000	00000	00226
00000	00000	00000	00000

64550	02550	00000	46000
07040	00060	51600	00001
40000	00000	02000	00111

00000	00000	00000	00000
04000	00063	00000	00021
00000	00000	40000	00000

00100	54000	00000	01000	00001
00104	50000	00000	00000	00226
00110	26060	40000	00000	00124
00114	10010	51200	00143	15310
00120	00000	00040	61000	46000
00124	51100	00142	51200	00143
00130	51320	10255	51430	10203
00134	51100	00121	51100	00106
00140	61100	00000	71602	47021
00144	00000	00000	00000	00001
00150	01020	35635	55555	55555
00154	00000	00000	00000	00007
00160	00000	00000	00000	00007
00164	24051	60310	01220	32423
00170	04004	00170	61000	46000
00174	51100	00166	04000	00176
00200	51100	00001	01000	00166
00204	51100	00001	03110	00203
00210	71602	20314	20652	36462
00214	04000	00206	61000	46000
00220	73660	20630	12161	73610
00224	04000	00215	61000	46000
00230	60000	00000	04004	00230

00011	50000	00000	00226
00000	00000	00000	00000
00000	00000	00000	00100
61114	05000	01001	46000
51300	00112	01020	32301
51300	00145	51300	00143
51540	20304	51205	13252
61200	00000	61200	00011
20650	01000	00170	46000
00000	00000	00000	00000
00000	00000	00000	00007
01020	35555	55555	55555
01020	45435	55555	55555
00000	00000	00000	00000
51100	00001	03110	00171
71100	00130	20160	46000
20652	01000	00170	46000
71602	20314	04000	00201
53160	20173	03110	00207
71603	24616	12661	20651
20173	03210	00215	20151
00000	00000	00000	00000

00000	00000	00000	00000
07537	10711	00000	00000
51200	00144	00000	00010
15550	00000	61000	46000
51400	00142	51500	00153
61100	10455	61200	10203
61407	77773	51100	00163
17406	32120	00000	00000
00000	00000	00000	03720
01020	30406	55555	55555
00000	00000	00000	00004
17216	22077	17401	56063
54110	20123	03310	00167
54610	04000	00167	46000
13661	13161	13661	46000
51100	00001	03110	00202
20150	36661	01000	00170
03010	00207	51100	00001
01000	00170	61000	46000
13116	20636	51600	00225
60000	00000	04004	00226

00251	53106	11140	50000
01000	00000	00005	20030
02040	60000	23011	50000
52110	00055	51210	10203
61300	20304	51100	00117
51100	00163	51100	00164
00000	00000	00000	00007
00000	00000	00000	00007
00000	00000	00000	00002
01020	35635	55555	55555
16413	44165	23633	56556
04000	00173	61000	46000
51100	00066	03310	00175
51600	00167	10611	46000
04004	00203	61000	46000
04004	00207	61000	46000
03110	00211	71100	00001
04004	00217	61000	46000
74660	36116	20123	46000
60000	00000	04004	00227

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

0
2 2

BINARY CONTROL CARDS.

IDENT TITLE
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 TERMINATES THE OPERATION FIELD. IF IT DOES NOT FIND A NONBLANK CHARACTER BEFORE THE DEFAULT COMMENTS COLUMN (SEE COL INSTRUCTION), IT TAKES THE CHARACTERS STARTING WITH THE DEFAULT COMMENTS COLUMN MINUS ONE UP TO THE END OF THE STATEMENT. 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 CHARACTERS BEYOND THE 62ND ARE LOST. A BLANK STRING PRODUCES AN UNTITLED LISTING.

0 46000

7160247021

20650

1 0100000000 X

TITLE NO
ENDRUN
SX6 3REND*4*1
LX6 400
RJ =XSYS=
ENDM

ENDRUN .1
ENDRUN .1
ENDRUN .1
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 EXPRESSION 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.

7-5

PSEUDO INSTRUCTION
SPACE PSEUDO INSTRUCTION.

COMPASS 3.6-476.
A123456789

01/09/79 08.34.39.

PAGE 5

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.

PSEUDO INSTRUCTION
EJECT PSEUDO INSTRUCTION

COMPASS 3.6-476.
TEJECT

01/09/79 08.34.39.

PAGE 7

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 TERMINATES 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 DEFAULT 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

SYM1 ONE OR MORE SYMBOLS DEFINED IN THE SUBPROGRAM. IF A SYMBOL QUALIFIER IS IN EFFECT WHEN THE "NOREF" IS ENCOUNTERED, THE SYMBOLS ARE ASSUMED TO BE QUALIFIED BY THE QUALIFIER IN USE. ALTERNATIVELY, SYM1 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 THAT 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, USELCH, ORG, OR
ORGC.

A LOCATION SYMBOL, IF PRESENT, IS IGNORED.

2

USELESS BSS 0
USELESS NOREF USELESS
SPACE 20

THIS SYMBOL SHOULD NOT OCCUR IN THE SYMBOLIC
REFERENCE TABLE

7-9

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 INDICATES 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 ACCORDING 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
NOTE THIS IS A NEW TITLE

COMPASS 3.6-476.

01/09/79 08.34.39.

PAGE 11

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.

SYS=

END TITLE

47500B CM STORAGE USED
MODEL 74 ASSEMBLY

432 STATEMENTS
0.949 SECONDS

4 SYMBOLS
4 REFERENCES

7-11

0 X

2

TITLE

COMMENT-PREFIX TABLE COMMENT

THE COMMENT PSEUDO INSTRUCTION INSERTS THE CHARACTER STRING SPECIFIED IN THE VARIABLE FIELD INTO THE EIGHTH THROUGH FOURTEENTH 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 INSTRUCTION IS MEANINGLESS. COMMENT INSTRUCTIONS FOLLOWING SEG AND BLANK IDENT PSEUDO INSTRUCTIONS ARE IGNORED WITHOUT NOTIFICATION.

LOCATION OPERATION VARIABLE-SUBFIELDS

COMMENT STRING

STRING COMPASS SEARCHES THE COLUMNS FOLLOWING THE BLANK THAT TERMINATES 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 DEFAULT 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 STATEMENT. 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 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" INSTRUCTION GENERATES A SUBTITLE AND CAUSES A PAGE EJECT. IF X IS

TTL PSEUDO INSTRUCTION
SYMBOLIC REFERENCE TABLE.

COMPASS 3.6-476.

01/09/79 08.34.39.

PAGE 14

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
LVA+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

135008 CM STORAGE USED

1 TABLE MOVE

MFS N01- CYB74-SN108 5C/R08 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. 475008 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.CH 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
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.20.34.

PAGE 1

ADDRESS LENGTH

0 4
4

BINARY CONTROL CARDS.

IDENT SET/EQU
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
CHECK EQU 200B
6160000144 SB6 CHECK
1000000 SETCHK1 SET 262144
100000 SETCHK2 = 100000B
COMMENT
50 COL 40
0 2 0000000000
7160247021
4 ENDRUN
END CHECK

THIS IS AN EXAMPLE OF A COMMENT
THIS IS AN EXAMPLE OF A COMMENT
THIS IS AN EXAMPLE OF A COMMENT BLANK PSEUDO OP
THIS IS AN EXAMPLE OF THE COL PSEUDO OPERATION
THIS IS AN EXAMPLE OF THE COL PSEUDO OPERATION

47300B SCH STORAGE USED
MODEL 174 ASSEMBLY

28 STATEMENTS
0.028 SECONDS

6 SYMBOLS
12 REFERENCES

2 ERRORS IN SET/EQU

SET/EQU
ERROR DIRECTORY.

COMPASS 3.5-470.

06/27/78 16.20.34.

PAGE 3

0 TYPE ERROR
OCCURRED ON PAGES

OPERATION FIELD BAD.
?

0 TYPE ERROR
OCCURRED ON PAGES

DOUBLY DEFINED SYMBOL. THE FIRST DEFINITION HOLDS
2

7-14

D

0



SE1/EQU
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 16.20.34.

PAGE 4

CHECK 144
CHECK1 144
SE1CHK 20
SE1CHK1 1000000
SE1CHK2 100000
SYS# 0

EXTERNAL

2/02 E 2/09 D 2/10 2/16
2/09 D
2/11 D 2/12 2/13 D 2/16
2/17 D
2/18 D
2/25

MFF NR2- CYR175-SN1 4LB7/R6B 05/15/78
16.20.32.DON00M1 FROM /SH
16.20.32.IP 00000320 WORDS - FILE INPUT , DC 04
16.20.32.DON. PSD.027A.72CT011A.MILLER
16.20.33.COMPASS.
16.20.34. 2 ERRORS IN SE1/EQU
16.20.34. ASSEMBLY ERRORS. 47300R SCH 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.I0 .474 SEC. .474 ADJ.
16.20.34.CM 0.480 MVS. .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

0	00000000000000000001	SAM	IDENT	REPEAT	
1	00000000000000000002		LIST	H.M.S.X	
2	00000000000000000003		ENTRY	RETTY	
3	00000000000000000004		REP	S/SAM, D/TOM, B/S.C/2	RFP PSEUDO OP
4	00000000000000000005		DATA	1.2.1.4.5	
5	5120000064	RETTY	SA2	44P	
	74620		SA6	A2	
	10722		SA7	X2	
6	5160000000 C		SA4	ANS	
	5170000001 C		SA7	ANS+1	
7	5130000012 .	LOOP	SA7	WORD	
	10733		BA7	X3	
10	5170000001		SA7	1	
	5140000001		SA4	1	
11	0314000007 .		NZ	X4, LOOP	
12	05160460000000000000	WORD	VFN	1R/3LEND, 3/6, 39/0	
13	01020304050607101112		DATA	10LAHCDEFGHIJ	
	0-0		HASE	0	
14	12	TOM	RSSZ	10	
			USE	/ANS/	USE LARELED COMMON
0	10	ANS	RSSZ	R	
			USE	//	USE BLANK COMMON
0	10		BSS	R	
26			END	RETTY	

RET

473008 SCM STORAGE USED 24 STATEMENTS 6 SYMBOLS
 MODEL 174 ASSEMBLY 0.029 SECONDS 13 REFERENCES

DUMP	RELATIVE	DMP(1111,177)
00111	00000 00000 00000 00064	14071 70000 00000 00000
00122	00000 00000 00000 00002	00000 00000 00000 00003
00124	00000 00000 00000 00004	00000 00000 00000 00005
00130	51300 00133 10733 44000	51700 00001 51400 00001
00134	01020 30405 04071 01112	00000 00000 00000 00001
00140	00000 00000 00000 00004	00000 00000 00000 00005
00144	00000 00000 00000 00003	00000 00000 00000 00004
00177	>60000 00000 04004 00177	

REPEAT
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.28.10.

PAGE 1

ADDRESS	LENGTH	BINARY CONTROL CARDS.	
0	26	IDENT	REPEAT
26		END	RETRY

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM*	LOCAL	0	26
ANS	COMMON	0	10
//	COMMON	0	10

ENTRY POINTS.

RETRY 5*

LOAD MAP - REPEAT

CYBER LOADER 1.4-470

06/27/78 16.28.10.

PAGE 1

FMA OF THE LOAD 111
LMA+1 OF THE LOAD 157

TRANSFER ADDRESS -- RETRY 126

PROGRAM ENTRY POINTS -- REPEAT 126

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
/ANS/	111	10					
REPEAT	121	26	LGO	06/27/78	COMPASS 3.5 470		
//	147	10					

.012 CP SECONDS

132000 CM STORAGE USED

MFP NR2- CYR175-SN1 4L07/R68 05/15/78
 16.28.08.DON00MR FROM /SM
 16.28.08.1P 00000320 WORDS - FILE INPUT , DC 04
 16.28.08.DON. PSD-02/R.72C10J1A-MILLER
 16.28.10.COMPASS.
 16.28.10. ASSEMBLY COMPLETE. 47300R SCH USED.
 16.28.10. 0.083 CPU SECONDS ASSEMBLY TIME.
 16.28.10.LOAD(LGO)
 16.28.10.EXECUTE(RETRY)
 16.28.11.DMP(111,177)
 16.28.11.OP 0000076R WORDS - FILE OUTPUT , DC 40
 16.28.11.MS 3584 WORDS 1 10752 MAX USED)
 16.28.11.CPA .120 SEC. .120 ADJ.
 16.28.11.10 .609 SEC. .609 ADJ.
 16.28.11.CM 9.840 KMS. .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

7-17

LOC

COMPASS 1.5-470.

06/27/78 09.04.34.

PAGE 2

	0		12			
	12	00000000000000000001		GAB	IDENT	100
	16	46000		SAM	ENTRY	GAM
L	12				BSS7	10
L	12	5110000000			DATA	1.2.3.4
		5120000001			NO	
L	13	5130000013			LOC	GAB
		5140000015			SA1	0
					SA2	1
					SA3	0
					SA4	0.2
					USE	0
21	10611				PK6	X1
	74630				SK6	A3
	74740				SK7	A4
22	5160000033				SA6	ANS.6
	5170000034				SA7	ANS.7
23	7160247021				ENDRUN	
25			12	ANS	BSS7	10
37					FND	GAM

473009 SCM STORAGE USED
MODEL 174 ASSEMBLY

22 STATEMENTS
0.024 SECONDS

4 SYMBOLS
A REFERENCES

7-18

DUMP	RELATIVE	DMP(111,147)
00111	00000 00000 00000 00000	00123-00000 00000 00000 00001
00124	00000 00000 00000 00002	00000 00000 00000 00003
00130	51100 00000 51200 00001	51300 00124 51400 00126
00134	71602 47021 20650 46000	01000 00152 61000 46000
00145	00000 00000 00000 00126	00000 00000 00000 00000
		00147-00000 00000 00000 00000
		00000 00000 00000 00004
		46000 46000 61000 46000
		10611 74630 74740 46000
		51600 00144 51700 00145
		00000 00000 00000 00000
		00144-00000 00000 00000 00124

LOC
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 09.04.34.

PAGE 1

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

PWA 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	PROCESSOR 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- CYR175-SN1 4L87/R6B 05/15/78
09.04.32.DON005U FROM /SM
09.04.32.1P 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.L00.
09.04.35.DMP(111.147)
09.04.35.OP 00000704 WORDS - FILE OUTPUT . DC 40
09.04.35.M5 3584 WORDS 1 716B MAX USED)
09.04.35.CPA .132 SEC. .132 ADJ.
09.04.35.10 .628 SEC. .628 ADJ.
09.04.35.CM 9.987 MVS. .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

7-19

CHECK1
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.28.34.

PAGE 1

ADDRESS LENGTH

0 7
7

BINARY CONTROL CARDS.

IDEN: CHECK1
END START1

ENTRY POINTS.

START1 2* TAG 0.

EXTERNAL SYMBOLS.

SYS* STAR12

CHECK1
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 16.28.34.

PAGE 3

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

7-20

IDENT CHECK 1
ENTRY START1, TAG
LIST 1

.....
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 RELOCATABLE ASSEMBLY 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
1 77777777777777777776
2 5110000000 *
10711
3 5170000000 *
0200000000 X
4
4 7160247021
20650
5 0100000000 X

TAG DATA -0.-1
START1 SA1 TAG
RX7 X1 TRANSMIT (X1) TO X7
SA7 =SHOLD
JP =XSTART2
ENDRUN
SX6 JREND*4+1
LX6 400
RJ =XSYS=
ENDM

ENDRUN .1
ENDRUN .1
ENDRUN .1
ENDRUN .1

DEFAULT SYMBOLS DEFINED BY COMPASS.

6
0 X
0 X
7

HOLD
START2
SYS=

END START1

504008 SCH STORAGE USED
MODEL 174 ASSEMBLY

40 STATEMENTS
0.058 SECONDS

5 SYMBOLS
8 REFERENCES

CHECK2
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.28.34.

PAGE 1

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

ADDRESS	LENGTH	IDENT	CHECK2	ENTRY	START2	SA1	XTAG+1	RT7	X1	SA7	SAVE	ENNRUN	RSS7	I	END
0	5110000001	X													
		10711													
1	5170900003	*													
		7160247021													
3		1													
4															

473008 SCM STORAGE USED
MODEL 174 ASSEMBLY

11 STATEMENTS
0.015 SECONDS

4 SYMBOLS
6 REFERENCES

CHECK2
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 16.28.34.

PAGE 3.

SAVE	J	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	

7-22

LOAD MAP - CHECK1

CYBER LOADER 1.4-470

06/27/78 16.20.35.

PAGE 1

LWA 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

13200R CM STORAGE USED

1 TABLE MOVE

7-23

DUMP RELATIVE

DMP(111,127)

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

HFF NO2- CYR175-SN1 4L07/R6R 05/15/78
 16.20.32.DON00MS FROM /SH
 16.20.32.IP 00000512 WORDS - FILE INPUT . DC 04
 16.20.32.DON. PSD-0270.72CT011A-MILLER
 16.20.33.COMPASS.
 16.20.34. ASSEMBLY COMPLETE. 50400H SCM USED.
 16.20.34. 0.130 CPU SECONDS ASSEMBLY TIME.
 16.20.34.LGO.
 16.20.35.DMP(111,127)
 16.20.35.OP 00001216 WORDS - FILE OUTPUT . DC 40
 16.20.36.MS 3504 WORDS (716R MAX USED)
 16.20.36.CPA .102 SEC. .101 ADJ.
 16.20.36.I0 .642 SEC. .642 ADJ.
 16.20.36.CM 11.400 KMS. .695 ADJ.
 16.20.36.SS 1.520
 16.20.36.FP 3.093 SEC. DATE 06/27/78
 16.20.36.EJ END OF JOB. SH

```
1      PROGRAM INITIAL(OUTPUT)
      COMMON /ANSWER/ BUFFER(16)
      CALL START
      PRINT 101
5      PRINT 100,(BUFFER(I),I=1,16)
      100 FORMAT(4022)
      101 FORMAT(11H)
      END
```

```
1      SUBROUTINE ERRMSG1
      PRINT 100
      100 FORMAT(11H,* ECS WRITE ERROR*)
      RETURN
5      END
```

```
1      SUBROUTINE ERRMSG2
      PRINT 100
      100 FORMAT(11H,* ECS READ ERROR*)
      RETURN
5      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	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 +	RJ	ADD
L	15		LOC	ADD
L	15	00000000000000000000	DATA	0
		MULT		

L 16 5110000000 *
 5120000001 *
 L 17 5130000002 *
 5140000003 *
 L 20 42612
 42734
 5160000006 C
 L 21 5170000007 C
 5100000015 *
 L 22 7100000032
 L 23 0110000012
 0100000000 X
 L 24 0100000015 *
 L 15
 L 15 00000000000000000000
 L 16 5110000000 *
 5120000001 *
 L 17 5130000002 *
 5140000003 *
 L 20 27101
 27202
 24101
 24202
 L 21 44621
 27303
 27404
 24303
 L 22 24404
 44743
 5160000010 C
 L 23 5170000011 C
 26666
 26777
 L 24 22666
 22777
 5160000012 C
 L 25 5170000013 C
 L 26 0400000010 *
 12
 10
 10
 12
 44
 12
 10
 61

END3BLK

DIVIDE

END4BLK

LBLOCK1 EQU
 LBLOCK2 EQU
 LBLOCK3 EQU
 LBLOCK4 EQU
 LBLOCKT EQU
 MAXBLK MAX
 MINBLK MIN
 END

SA1 OPERAND+0
 SA2 OPERAND+1
 SA3 OPERAND+2
 SA4 OPERAND+3
 IX6 X1*X2
 IX7 X3*X4
 SA6 ANSWER+6
 SA7 ANSWER+7
 SA0 ADD
 SX0 LBLOCK1+LBLOCK2+LBLOCK3
 RE LBLOCK4
 RJ ERMMSG2
 RJ ADD
 LOC ADD
 DATA 0
 SA1 OPERAND+0
 SA2 OPERAND+1
 SA3 OPERAND+2
 SA4 OPERAND+3
 PX1 X1
 PX2 X2
 NX1 X1
 NX2 X2
 FX6 X2/X1
 PX3 X3
 PX4 X4
 NX3 X3
 NX4 X4
 FX7 X4/X3
 SA6 ANSWER+8
 SA7 ANSWER+9
 UX6 X6,B6
 UX7 X7,B7
 LX6 X6,B6
 LX7 X7,B7
 SA6 ANSWER+10
 SA7 ANSWER+11
 EQ START
 EQU END1BLK-ADD+1
 EQU END2BLK-SUBT+1
 EQU END3BLK-MULT+1
 EQU END4BLK-DIVIDE+1
 EQU LBLOCK1+LBLOCK2+LBLOCK3+LBLOCK4
 MAX LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4
 MIN LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4
 END

1100B ECS 46700B CM

STORAGE USED
 MODEL 73 ASSEMBLY

100 STATEMENTS
 0.347 SECONDS

20 SYMBOLS
 0 REFERENCES

FMA 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.10./	2376	101							
Q2NTRY=	2477	0	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
/FCL=ENT/	2477	40							
COMIO=	2537	33	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FCL=FOL	2572	40	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FEIFST=	2632	3	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FLTOUT=	2635	311	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FORSYS=	3146	301	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
OUTCOM=	3447	154	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
SYSID=	3623	1	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FECHSK=	3624	41	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FMTAP=	3665	357	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FORUIL=	4244	46	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
GETFIT=	4312	57	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
KODER=	4371	451	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
OUTC=	5042	150	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
/FOL.COM/	5212	14							
FOL.RES	5226	211	SL-SYSLIB	11/02/78	COMPASS	3.6	405		
FOL.MMI	5437	222	SL-SYSLIB	11/02/78	COMPASS	3.6	485		
CPU.SYS	5661	40	SL-SYSLIB	11/15/78	COMPASS	3.6	476		
CMF.ALF	5721	160	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CMF.CSF	6101	6	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CMF.FFA	6107	14	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CMF.FHF	6123	36	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CMH.R	6161	214	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CMF.SLF	6375	22	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CTLSRM	6417	433	SL-SYSLIB	11/16/78	COMPASS	3.6	485		
ERRSRM	7052	25	SL-SYSLIB	11/16/78	COMPASS	3.6	485		
LISTSRM	7077	67	SL-SYSLIB	11/16/78	COMPASS	3.6	485		

FCL INITIALIZATION ROUTINE.

COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL CAPSULE LOADING
CONVERTED DATA STORAGE
COMMON FLOATING OUTPUT CODE
FORTRAN OBJECT LIBRARY UTILITIES.
COMMON OUTPUT CODE
LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
INITIALIZE CONSTANTS.
CRACK APLIST AND FORMAT FOR KODER/KRAKER.
FCL MISC. UTILITIES.
LOCATE AN FIT GIVEN A FILE NAME.
OUTPUT FORMAT INTERPRETER.
FORMATTED WRITE FORTRAN RECORD.

FAST DYNAMIC LOADER RESIDENT.
FOL MEMORY MANAGER INTERFACE.
PROCESS SYSTEM REQUEST.
CMM V1.1 - ALLOCATE FIXED.
CMM V1.1 - CHANGE SPECS FIXED.
CMM V1.1 - FIXED FREE ALGORITHM.
CMM V1.1 - FREE FIXED.
CMM V1.1 - RESIDENT SUBROUTINES.
CMM V1.1 - SHRINK AT LWA FIXED.
CRM CONTROLLING ROUTINE.
CRM ERROR PROCESSOR ENTRY.
CRM - ALLOCATE SPACE FOR LIST OF FILES

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

MFS NH1- CY874-SN108 5C/R08 11/14/78
 12.26.16.DON00HT FROM /OH
 12.26.16.1P 00000768 WORDS - FILE INPUT , DC 04
 12.26.16.DON,EC1. 001A,6883,1896,MILLER
 12.26.38.REWIND,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.
 12.31.20.10 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 CONDITION 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 REFERENCES TO BLANK COMMON. THE TEST IS MADE IN PASS TWO OF THE ASSEMBLER. 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
8	EQU	1
-1	C	= -1
0	D	= 0
-1	Z	ERRNG C
-1	Y	ERRMI C
1	X	ERRNZ B
1	W	ERRPL B
0	6	ERRZR D
27	Q	ERRPL BUFF-ERRXX-120B
	ENDRUN	
144	BUF	BSS 100
	BUFF	BSS 0
	END	

1 7160247021

3
147
147

ERRXX

COMPASS 3.6-476.

01/09/79 12.27.56.

PAGE 2

5 ERRORS IN ERRXX

ERRXX
ERROR DIRECTORY.

COMPASS 3.6-476.

01/09/79 12.27.56.

PAGE 3

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 N01- CYB74-SN108 SC/R0B 11/14/78
12.27.53.DON00FL FROM /OH
12.27.53.IP 00000512 WORDS - FILE INPUT , DC 04
12.27.53.DON,T5. 001A,6083,1096,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. 0.305 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.I0 .305 SEC. .305 ADJ.
12.27.57.CH 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

7-30

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

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.)

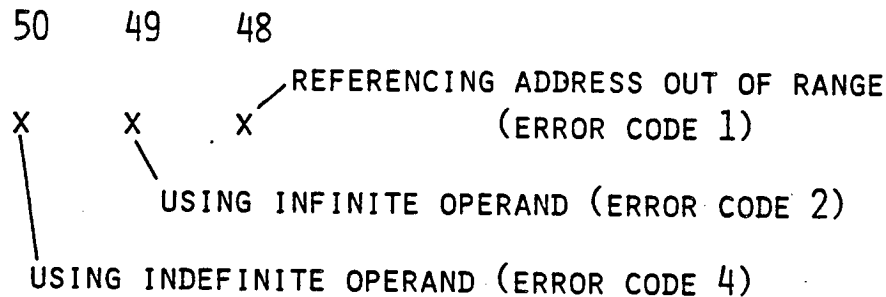
OR

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:



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.

```

IDEN1  INDEF IN
ENTRY  INDEF IN
VFD    12/1/77R.48/0  INDEFINITE OPERAND
DATA   2.0
SA1    INDEF
SA2    NUMBER
FX1    1/1X2          USE INDEFINITE OPERAND IN DIVIDE
NO
NO
NO
NO
ENDRUN  NORMAL TERMINATION
END     INDEF IN      TRANSFER ADDRESS

```

47300R SCM STORAGE USED	16 STATEMENTS	4 SYMBOLS
MODEL 174 ASSEMBLY	0.019 SECONDS	7 REFERENCES

DMPH.

P	000000	A0	000200	R0	000000
RA	510400	A1	000111	R1	000001
FL	000200	A2	000112	R2	000002
EW	700700	A3	000057	R3	012733
RE	000055	A4	000001	R4	000201
FE	000000	A5	000113	R5	000113
MA	001200	A6	000001	R6	000200
		A7	000001	R7	027756
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

C(A1)= 1777 0000 0000 0000 0000 C(A1)= 0000 0000 0000 0000 0000
 C(A2)= 1721 4000 0000 0000 0000 C(A2)= 0000 0000 0000 0000 0000
 C(A3)= 0000 0000 0000 0000 0000 C(A3)=
 C(A4)= 0000 0000 0000 0000 0000 C(A4)=
 C(A5)= 5110 0001 1151 2000 0112 C(A5)= 5110 0001 1151 2000 0112
 C(A6)=
 C(A7)=

[illegible]

INDEFIN
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 08.55.10.

PAGE 1

ADDRESS	LENGTH	BINARY CONTROL CARDS.		
0	11	IDENT	INDEFIN	
11		END	INDEFIN	TRANSFER ADDRESS

ENTRY POINTS.

INDEFIN 2*

EXTERNAL SYMBOLS.

SYS=

LOAD MAP - INDEFIN

CYREN LOADER 1.4-470

06/27/78 08.55.11.

PAGE 1

PWA 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	PROCESSOR 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

13200R CM STORAGE USED

MFF NO2- CYB175-SNI 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-02/8.72CT011A.MILLER
08.55.10.COMPASS.
08.55.10. ASSEMBLY COMPLETE. 47300R SCH 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.
08.55.12.10 .639 SEC. .639 ADJ.
08.55.12.CH 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
1 511000001
2 46000
3 46000
4 46000
6 7160247021

TSTADDR NO.
SRI -500
SAI 01..
NO
NO
NO
NO
ENDRUN
END TSTADDR

NORMAL TERMINATION
TRANSFER ADDRESS

4730000 SCM STORAGE USED
MODEL 174 ASSEMBLY

25 STATEMENTS
0.025 SECONDS

2 SYMBOLS
3 REFERENCES

8-7

OMPR.

P 000000 A0 000200 R0 000000
RA 206600 A1 777125 H1 777013
FL 000200 A2 000060 R2 000002
FM 700700 A3 000057 R3 012733
RE 000055 A4 000001 R4 000201
FE 000000 A5 000111 R5 000111
NA 001200 A6 000001 R6 000200
A7 000001 R7 027756
R0 0000 0000 0000 0000 0000
R1 0000 0000 0000 0000 0000
R2 1505 1520 0000 0000 0061
R3 0000 0000 0000 0000 0000
R4 0000 0000 0000 0000 0000
R5 0000 0000 0000 0000 0000
R6 1505 1520 0000 0000 0061
R7 0000 0000 0000 0000 0000

CIA11= 1505 1520 0000 0000 0061
CIA21= 0000 0000 0000 0000 0000
CIA31= 0000 0000 0000 0000 0000
CIA41= 4600 0611 0777 0134 6000
CIA51= 0000 0000 0000 0000 0000
CIA61= 0000 0000 0000 0000 0000
CIA71= 0000 0000 0000 0000 0000

CIR11= 0000 0000 0000 0000 0000
CIR21= 0000 0000 0000 0000 0000
CIR31= 4600 0611 0777 0134 6000
CIR41= 0000 0000 0000 0000 0000
CIR51= 0000 0000 0000 0000 0000
CIR61= 0000 0000 0000 0000 0000
CIR71= 0000 0000 0000 0000 0000

00000 00010 00113 00000 00000
00054 56110 03110 00054 54710
00060 15051 52700 00000 00061
00064 14071 70000 00000 00000
00070 14071 75755 00000 00000
00100 54000 00000 01000 00001

00000 00000 00000 00000
51100 00001 03110 00055
00000 00200 00000 00001
00000 00000 00000 00157
00000 00000 00000 00000

44550 02550 00000 46000
07040 00060 51600 00001
40000 00000 02000 00111

00000 00000 00000 00001
04000 00063 00000 00021
00000 00000 40000 00001

00100 54000 00000 01000 00001
00104 00000 00000 00000 00157
00110 24232 40104 04220 00111
00115 44000 71602 47021 20450

00004 60000 00000 00157
00000 00000 00000 00000
46000 61107 77013 46000
01000 00111 21602 20450

00000 00000 00000 00000
51110 00112 46000 46000
01000 00111 21602 20450

46000 46000 61000 46001

ISTADDP
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 09.03.50.

PAGE 1

ADDRESS LENGTH

BINARY CONTROL CARDS.

0 6
6

IDENT ISTADDR
END ISTADDR TRANSFER ADDRESS

ENTRY POINTS.

ISTADDR 0.

EXTERNAL SYMBOLS.

SYS.

LOAD MAP - ISTADDR

CYBER LOADER 1.4-470

06/27/78 09.03.51.

PAGE 1

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

TRANSFER ADDRESS -- ISTADDR 111

PROGRAM ENTRY POINTS -- ISTADDR 111

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
ISTADDR	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.

.021 CP SECONDS

13200B CM STORAGE USED

MFF NR2- CYR175-SN1 4L87/R68 05/15/78
09.03.47.DON005T FROM /SH
09.03.47.1P 00000320 WORDS - FILE INPUT , DC 04
09.03.47.DON. PSD.0278.72CT011A.MILLER
09.03.50.COMPASS.
09.03.51. ASSEMBLY COMPLETE. 47300R SCH 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 0000100R WORDS - FILE OUTPUT , DC 40
09.03.52.MS 3584 WORDS 1 7168 MAX USED
09.03.52.CPA .126 SEC. .126 ADJ.
09.03.52.10 .643 SEC. .643 ADJ.
09.03.52.CM 9.440 MWS. .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

IDENT OUTHANG
ENTRY OUTHANG

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
5170000000
1 7150001111
5150377776
2 7170006666
5170377776
3 46000
4 46000
7130003333
5 46000
6 46000
7160247021
10

OUTRANG SX7 7
SA7 0
SX5 11110
SA5 377776B
SX7 6666B
SA7 377776B
NO
NO
SX3 3333B
NO
NO
ENDRUN
END OUTHANG

47500B CM STORAGE USED
MODEL 74 ASSEMBLY

31 STATEMENTS
0.095 SECONDS

2 SYMBOLS
0 REFERENCES

LOAD MAP - OUTHANG

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 -- OUTHANG 111

PROGRAM ENTRY POINTS -- OUTHANG 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

P 000123 A0 000200 B0 000000
 RA 175600 A1 000001 B1 000001
 FL 000200 A2 000060 B2 000002
 EM 000600 A3 000057 B3 013274
 RE 000037 A4 000001 B4 000201
 FE 000000 A5 377776 B5 000111
 MA 001000 A6 000122 B6 000200
 A7 377776 B7 027756
 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
 00056 64550 02550 00000 46000
 00060 15051 52000 00000 00061
 00064 04152 00000 00000 00000
 00070 14071 75700 00000 00000
 00100 54000 00000 01000 00001
 00105 00000 00000 00000 00000
 00113 71700 06666 51703 77776
 00114 46000 46000 61000 46000
 00120 01000 00123 61000 46000
 00124 51100 00001 03110 00124
 00130 71100 00130 20160 46000
 00134 20652 01000 00123 46000
 00140 71602 20314 04000 00134
 00144 53160 20173 03310 00142
 00150 71603 24616 12661 20651
 00154 20123 03210 00150 20151
 00160 00000 00000 00000 00000

C(A1)= 0000 0000 0000 0000 0000
 C(A2)= 1505 1520 0000 0000 0061
 C(A3)= 0000 0000 0000 0000 0000
 C(A4)= 0000 0000 0000 0000 0000
 C(A5)= * * OUT OF RANGE * *
 C(A6)= 0130 0000 0000 0000 0000
 C(A7)= * * OUT OF RANGE * *

C(B1)= 0000 0000 0000 0000 0000
 C(B2)= 0000 0000 0000 0000 0000
 C(B3)= * * OUT OF RANGE * *
 C(B4)= * * OUT OF RANGE * *
 C(B5)= 7170 0000 0751 7000 0000
 C(B6)= * * OUT OF RANGE * *
 C(B7)= * * OUT OF RANGE * *

00000 00000 00000 00000 00054_56110 03110 00054 54710
 00000 00000 00000 00000
 00000 00200 00000 00001 07040 00060 51600 00001
 00000 00000 00000 00161 40000 00000 02000 00111
 00000 00000 00000 00000 00100_54000 00000 01000 00001

51100 00001 03110 00055
 04000 00063 00000 00000
 00000 00000 40000 00000

00005 00000 00000 00161
 00110_17252 42201 16070 00111

00000 00000 00000 00000
 71700 00007 51700 00000

00104_00000 00000 00000 00161
 71500 01111 51503 77776

46000 71300 03333 46000
 04000 00134 00000 00000
 54610 04000 00122 46000
 13661 13161 13661 46000
 51100 00001 03110 00135
 20150 36661 01000 00123
 03010 00142 51100 00001
 01000 00123 61000 46000
 13116 20636 51600 00160
 60000 00000 04004 00161

46000 46000 61000 46000
 01300 00000 00000 00000
 51100 00066 03310 00130
 51600 00122 10611 46000
 04004 00136 61000 46000
 04004 00142 61000 46000
 03110 00144 71100 00001
 04004 00152 61000 46000
 74660 36116 20123 46000
 00177>60000 00000 04004 00177

46000 71602 47021 20650
 04000 00121 00000 00000
 51100 00121 04000 00131
 51100 00001 01000 00121
 51100 00001 03110 00136
 71602 20314 20652 36662
 04000 00141 61000 46000
 73660 20630 12161 73610
 04000 00150 61000 46000

8-10

MFS NBI- CYB74-SN108 5C/R0B 11/14/78
 10.25.10.DON00K1 FROM /OH
 10.25.10.IP 00000384 WORDS - FILE INPUT , DC 04
 10.25.10.DON.75. 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.IU .316 SEC. .316 ADJ.
 10.29.06.CH 12.493 KWS. .762 ADJ.
 10.29.06.SS 1.806
 10.29.06.PP 10.161 SEC.
 10.29.06.EJ END OF JOB, OH DATE 01/08/79

TESTRAO
STORAGE ALLOCATION.

ADDRESS	LENGTH	BINARY CONTROL CARDS.
0	10	IDENT TESTRAO
10		END TESTRAO

ENTRY POINTS.

TESTRAO 1.

EXTERNAL SYMBOLS.

SYS=

LOAD MAP - TESTRAO

CYBER LOADER 1.4-470

06/27/78 08.38.21.

PAGE 1

FMA OF THE LOAD 111
LMA+1 OF THE LOAD 161

TRANSFER ADDRESS -- TESTRAO 112
PROGRAM ENTRY POINTS -- TESTRAO 112

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
TESTRAO	111	10	LGO	06/27/78	COMPASS 3.5 470		
SYS.PH	121	40	SL-SYSLIB	05/16/78	COMPASS 3.5 470		

.022 CP SECONDS

132008 CM STORAGE USED

MFF NB2- CYB175-SN1 4L07/R60 05/15/78
08.38.17.DON0031 FROM /SM
08.38.17.IP 00000192 WORDS - FILE INPUT , DC 04
08.38.17.DON. PSD.0278.72CT011A.MILLER
08.38.19.COMPASS.
08.38.20. ASSEMBLY COMPLETE. 473008 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.
08.38.21.10 .631 SEC. .631 ADJ.
08.38.21.CH 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

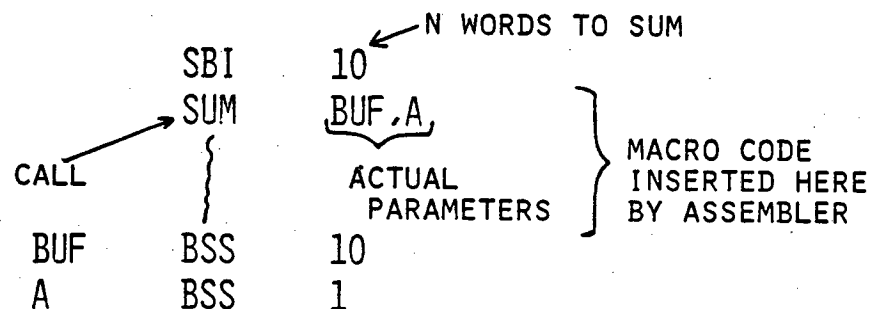
[illegible]

MACRO

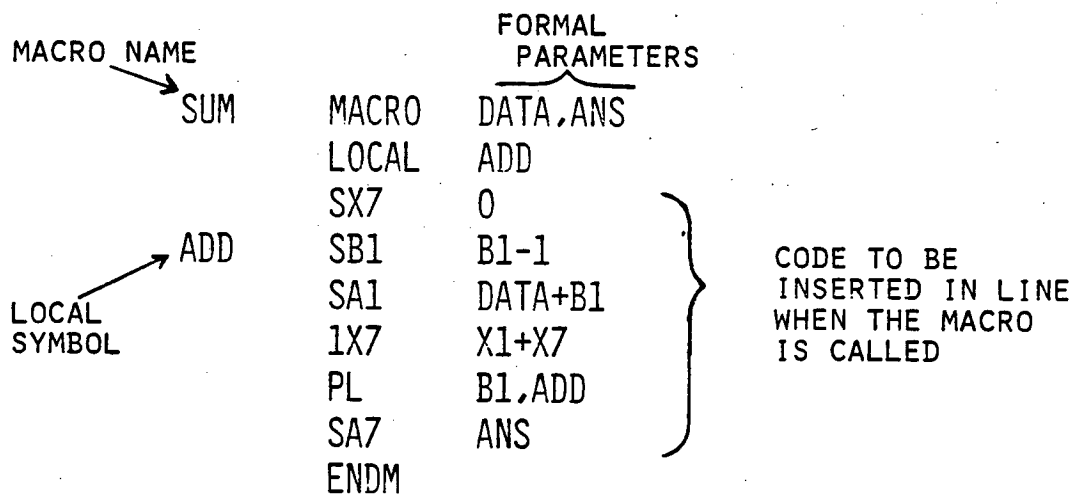
- * TYPE 1 FORMAT
- * SINGLE "INSTRUCTION" TO CALL A SEQUENCE OF CODE
- * LIKE AN IN-LINE SUBROUTINE

I.E.
SUM MACRO

THE CALL:



THE MACRO DEFINITION:



MOV7600
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 09.13.58.

PAGE 1

ADDRESS LENGTH

0 100
100

BINARY CONTROL CARDS.

IDENT MOV7600
END ADD

ENTRY POINTS.

ADD 30*

EXTERNAL SYMBOLS.

SYS*

MFF NR2- CYR175-SN1 4LR7/H6R 05/15/78
09.13.56.DON006U FROM /SM
09.13.56.IP 00000512 WORDS - FILE INPUT . DC 04
09.13.56.DON. PSC.027A.72CT011A.MILLER
09.13.57.COMPASS.
09.13.58. ASSEMBLY COMPLETE. 47300R SCM USED.
09.13.58. 0.179 CPU SECONDS ASSEMBLY TIME.
09.13.58.LGO.
09.13.59.DMP(111.177)
09.13.59.FXIT.
09.13.59.OP 00001792 WORDS - FILE OUTPUT . DC 40
09.13.59.WS 3544 WORDS 1 10752 MAX USED)
09.13.59.CPA .229 SEC. .229 ADJ.
09.13.59.10 .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. SM

9-4

MOV7600
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 09.13.58.

PAGE 4

ADD	70	PROGRAM*	2/03 F	2/37 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				
END1PLK	43	PROGRAM*	2/41 L	3/44					
END2PLK	54	PROGRAM*	3/02 L	3/45					
END3PLK	65	PROGRAM*	3/20 L	3/46					
END4PLK	76	PROGRAM*	3/43 L	3/47					
LPLOCK1	50		3/48 D						
LPLOCK1	14		3/44 D	3/49	3/49	3/50			
LPLOCK2	11		2/34	3/45 D	3/48	3/50			
LPLOCK3	11		2/52	3/46 D	3/48				
LPLOCK4	12		3/13	3/47 D	3/48				
MAXPLK	14		3/49 D						
MINPLK	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
SURT	44	PROGRAM*	2/35	2/42 L	3/45				
SYS*	0	EXTERNAL*	3/44						

```

0          20
20 000000000000000000000001
30 5110000020 .
      5120000021 .
31 5130000022 .
      5140000023 .
32 36612
      36734
      5160000000 .
33 5170000001 .
      5110000024 .
34 5120000025 .
      5130000026 .
35 5140000027 .
      36612
      36734
36 5160000002 .
      5170000003 .
37
37 6110000000
      6120000011
40 5151000044 .
      10655
41 5161000030 .
      6111000001
42 0512000040 .
43 0400000030 .
44 5110000020 .
      5120000021 .
45 5130000022 .
      5140000023 .
46 37621
      37734
      5160000004 .
47 5170000005 .
      6110000000
50 6120000011
51 5151000055 .
      10655
52 5161000030 .
      6111000001
53 0512000051 .

```

```

IDENT MOV7600
LIST M
ENTRY ADD
MOVE MACRO PARM1,PARM2,PARM3
LOCAL MOVE1
SR1 0
SR2 PARM1
MOVE1 SA5 PARM2+01
      RX6 X5
      SA6 PARM3+01
      SR1 01+1
      NE 01,02,MOVE1
      ENDM
ANSWER BSSZ 16
OPERAND DATA 1,2,3,4,5,6,7,8
ADD SA1 OPERAND+0
      SA2 OPERAND+1
      SA3 OPERAND+2
      SA4 OPERAND+3
      IX6 X1+X2
      IX7 X3+X4
      SA6 ANSWER+0 STORE RESULT
      SA7 ANSWER+1 STORE RESULT
      SA1 OPERAND+4
      SA2 OPERAND+5
      SA3 OPERAND+6
      SA4 OPERAND+7
      IX6 X1+X2
      IX7 X3+X4
      SA6 ANSWER+2
      SA7 ANSWER+3
MOVE LBLOCK2,SUBT,ADD
SR1 0
SR2 LBLOCK2
      SA5 SUBT+01
      RX6 X5
      SA6 ADD+01
      SR1 01+1
      NE 01,02,++000001
      ENDM
ENDBLK EQ ADD
SUBT SA1 OPERAND+0
      SA2 OPERAND+1
      SA3 OPERAND+2
      SA4 OPERAND+3
      IX6 X2-X1
      IX7 X3-X4
      SA6 ANSWER+4 STORE RESULT
      SA7 ANSWER+5 STORE RESULT
MOVE LBLOCK3,MULT,ADD
SR1 0
SR2 LBLOCK3
      SA5 MULT+01
      RX6 X5
      SA6 ADD+01
      SR1 01+1
      NE 01,02,++000002

```

```

MOVE .1
MOVE .1
MOVE .1
MOVE .1
MOVE .1
MOVE .1
MOVE .1

```

```

MOVE .1
MOVE .1
MOVE .1
MOVE .1
MOVE .1
MOVE .1

```

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		
			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		MX1	X1		
	24202		MX2	X2		
71	27303		PX3	X3		
	27404		PX4	X4		
	24303		MX3	X3		
	24404		MX4	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	EQ	END1BLK-ADD*1		
	11	LBLOCK2	EQ	END2BLK-SUBT*1		
	11	LBLOCK3	EQ	END3BLK-MULT*1		
	12	LBLOCK4	EQ	END4BLK-DIVIDE*2		
	50	LBLOCKT	EQ	LBLOCK1*LBLOCK2*LBLOCK3*LBLOCK4		
	14	MAXBLK	MAX	LBLOCK1*LBLOCK2*LBLOCK3*LBLOCK4		
	11	MINBLK	MIN	LBLOCK1*LBLOCK2*LBLOCK3*LBLOCK4		
			END	ADD		

100

473008 SCM STORAGE USED
MODEL 174 ASSEMBLY

111 STATEMENTS
0.122 SECONDS

21 SYMBOLS
78 REFERENCES

000003 INVENTED SYMBOLS

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	PROCESSOR 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

13200B CH STORAGE USED

1 TABLE MOVE

9-7

DUMP	RELATIVE	DUMP(111,177)
00111	00000 00000 00000 00003	00000 00000 00000 00007
00114	00000 00000 00000 00017	00000 00000 00000 00001
00120	00000 00000 00000 00014	17214 00000 00000 00000
00124	00000 00000 00000 00001	00000 00000 00000 00000
00131	00000 00000 00000 00003	00000 00000 00000 00005
00134	00000 00000 00000 00004	00000 00000 00000 00005
00140	00000 00000 00000 00010	51100 00131 51200 00132
00144	27303 27404 24703 24404	44621 44743 51600 00121
00150	51700 00124 61000 46000	71602 47021 20650 46000
00154	04000 00141 61000 46000	51100 00131 51200 00132
00160	51700 00116 61100 00000	61200 00011 61900 46000
00164	05120 00162 61000 46000	04000 00141 61000 46000
00170	42612 42734 51600 00117	51700 00120 61100 00000
00174	51610 00141 61110 00001	05120 00173 61000 46000

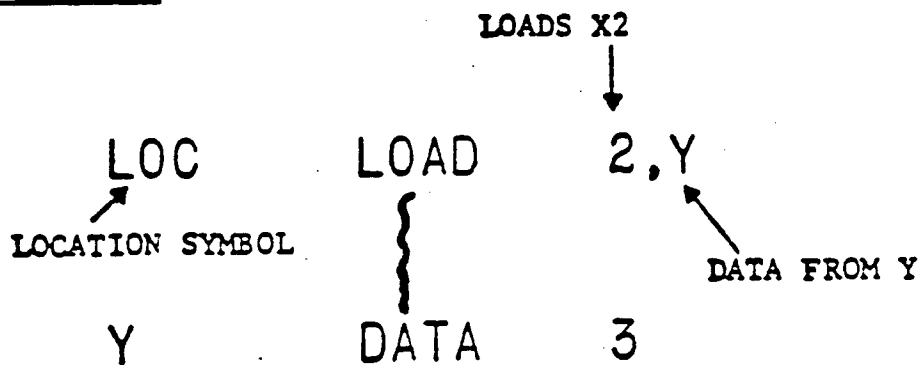
MACRO

Type 2 format:

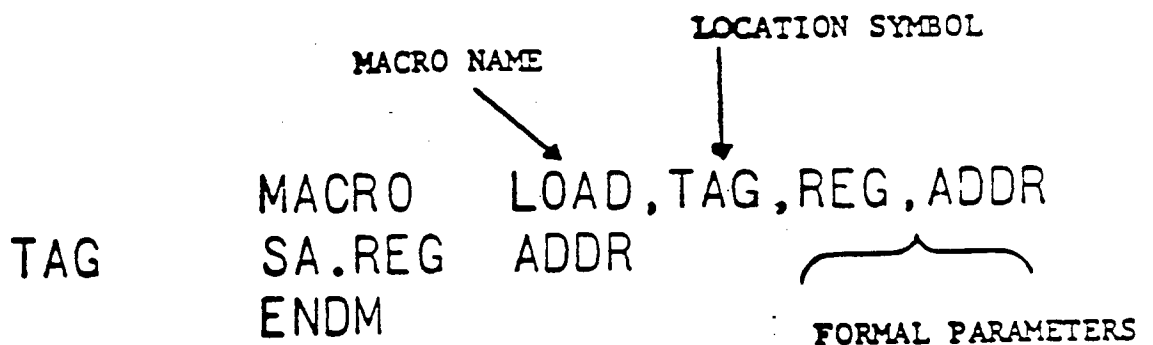
i.e.

LOAD macro

The Call:



The Macro Definition:



FORMAT II

SAM
BEGIN
P+U

PLACE TAG ON NEXT LOCATION

[illegible]

61 STATEMENTS
0.265 SECONDS

```

MFS NH-      CYH74-SN108      5C/HOH      11/14/74
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. 47500H 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 00
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.ID .317 SEC. .317 ADJ.
16.41.46.CH 16.162 KWS. .986 ADJ.
16.41.47.SS 1.861
16.41.47.PP 2.942 SEC.
16.41.47.EJ END OF JOB, OH DATE 01/08/74

```

FWA OF THE LOAD 111
LWA+1 OF THE LOAD 213

TRANSFER ADDRESS -- BEGIN 111

PROGRAM ENTRY POINTS -- MACHU 111

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
MACHO	111	42	L60	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

9-10

DUMP	RELATIVE	DUMP(111,200)
00111	51100 00134 51200 00133	42512 51300 00131 46000
00114	51600 00147 51100 00137	51200 00136 42512 46000
00120	51100 00142 51200 00141	42512 51300 00137 46000
00124	51200 00145 42512 46000	51300 00143 51400 00144
00130	01000 00155 61000 46000	00000 00000 00000 00001
00134	00000 00000 00000 00004	00000 00000 00000 00005
00140	00000 00000 00000 00010	00000 00000 00000 00011
00144	00000 00000 00000 00002	00000 00000 00000 00003
00150	00000 00000 00000 01510	00000 00000 00000 00000
00154	01300 00000 00000 00000	04000 00131 00000 00000
00160	51100 00066 03310 00162	51100 00153 04000 00163
00164	51600 00154 10611 46000	51100 00001 01000 00153
00170	04004 00170 61000 46000	51100 00001 03110 00170
00174	04004 00174 61000 46000	71602 20314 20652 36662
00200	03110 00176 71100 00001	

```

1      PROGRAM MACDPH(OUTPUT)
      COMMON ANSWER(24)
      PRINT 100
100    FORMAT(1H1)
5      CALL BEGIN
      PRINT 20,(ANSWER(I),I=1,24)
20    FORMAT(4024)
      END
  
```

17214000000000000000	17216000000000000000	17214000000000000000	17216000000000000000
17176000000000000000	6055677777777777777	17176000000000000000	6057377777777777777
17247400000000000000	17235400000000000000	17247400000000000000	17235400000000000000
17177000000000000000	6054737777777777777	17177000000000000000	6057377777777777777
17265500000000000000	17244600000000000000	17265500000000000000	17244600000000000000
17177252525252525252	6054737777777777777	17177252525252525252	6057377777777777777

9-11

```

MFS NB1- CY874-SN108 5C/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 MACDPH
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.
12.56.25.10 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
  
```

9-12

		IDENT	MACHO		
		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			
1		DATA	-0		
		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		
1	5110000032 *	SA1	M	SAM	.1
	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
	24606	ENDM		STORE	.2
	24707	NX6	X6	SAM	.1
10		NX7	X7	SAM	.1
10	5160000006 C	STORE	ANS+6,ANS+7	SAM	.1
	5170000007 C	SA6	ANS+6	STORE	.2
		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	51100000036 *	SA1	M+4	LOAD	.2
	51200000037 *	SA2	M+5	LOAD	.2
12	51300000040 *	SA3	M+6	LOAD	.2
	51400000041 *	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
	51600000010 C	SA6	ANS+8	STORE	.2
14	51700000011 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	51600000012 C	SA6	ANS+10	STORE	.2
	51700000013 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
	51600000014 C	SA6	ANS+12	STORE	.2
17	51700000015 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	51600000016 C	SA6	ANS+14	STORE	.2
	51700000017 C	SA7	ANS+15	STORE	.2
		ENDM		STORE	.2
		ENDM		SAM	.1
21	51100000042 *	LIST	-M		
		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	04000000000 *	EQ	BEGIN		
32	1720400000000000000000	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		BSS	24		
	30	USE	0		
46		END			

47600B CM STORAGE USED
MODEL 73 ASSEMBLY

121 STATEMENTS
0.454 SECONDS

3 SYMBOLS
0 REFERENCES

3 TYPE ERROR
OCCURRED ON PAGESDUPLICATE MACRO DEFINITION. NEW ONE OVERRIDES.
1

LOAD MAP - MACDPH

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 -- MACDPH 2173

PROGRAM ENTRY POINTS -- MACDPH 2173

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCSSR	VER	LEVEL	HARDWARE	COMMENTS
MACDPH	111	2105	LGO	01/09/79	FTN	4.7	485	666X 1	PROGRAM OPT=1
MACRO	2216	46	LGO	01/09/79	COMPASS	3.6	476		
/STP.END/	2264	1							
/FCL.C./	2265	26							
/08.10./	2313	101							
Q2NTRY=	2414	0	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
/FCL=ENT/	2414	40							
COMIO=	2454	33	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FCL=FOL	2507	40	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FEIFST=	2547	3	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FLTOUT=	2552	311	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FORSYS=	3063	301	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
OUTCOM=	3364	154	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
YSAID=	3540	1	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
ECMSK=	3541	41	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
ENTAP=	3602	357	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
FORUTL=	4161	46	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
GETFIT=	4227	57	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
KODER=	4306	451	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
OUTC=	4757	150	SL-FORTRAN	11/16/78	COMPASS	3.6	485		
/FDL.COM/	5127	14							
FDL.RES	5143	211	SL-SYSLIB	11/02/78	COMPASS	3.6	485		
FDL.MHI	5354	222	SL-SYSLIB	11/02/78	COMPASS	3.6	485		
CPU.SYS	5576	40	SL-SYSLIB	11/15/78	COMPASS	3.6	476		
CHF.ALF	5636	160	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHF.CSF	6016	6	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHM.FFA	6024	14	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHF.FRF	6040	36	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHM.R	6076	214	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHF.SLF	6312	22	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CTL3RM	6334	433	SL-SYSLIB	11/16/78	COMPASS	3.6	485		
ERR3RM	6767	25	SL-SYSLIB	11/16/78	COMPASS	3.6	485		
LIST3RM	7014	67	SL-SYSLIB	11/16/78	COMPASS	3.6	485		
//	7103	30							

FCL INITIALIZATION ROUTINE.

COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL CAPSULE LOADING
CONVERTED DATA STORAGE
COMMON FLOATING OUTPUT CODE
FORTRAN OBJECT LIBRARY UTILITIES.
COMMON OUTPUT CODE
LINK BETWEEN SYS=AID AND INITIALIZATION CODE.
INITIALIZE CONSTANTS.
CRACK APLIST AND FORMAT FOR KODER/KRAKER.
FCL MISC. UTILITIES.
LOCATE AN FIT GIVEN A FILE NAME.
OUTPUT FORMAT INTERPRETER.
FORMATTED WRITE FORTRAN RECORD.

FAST DYNAMIC LOADER RESIDENT.
FDL MEMORY MANAGER INTERFACE.
PROCESS SYSTEM REQUEST.
CMM VI.1 - ALLOCATE FIXED.
CMM VI.1 - CHANGE SPECS FIXED.
CMM VI.1 - FIXED FREE ALGORITHM.
CMM VI.1 - FREE FIXED.
CMM VI.1 - RESIDENT SUBROUTINES.
CMM VI.1 - SHRINK AT LWA FIXED.
CRM CONTROLLING ROUTINE.
CRM ERROR PROCESSOR ENTRY.
CRM - ALLOCATE SPACE FOR LIST OF FILES

9-15

0	20	IDENT	MACROE		
20	00000000000000000001	ENTRY	ADD		
30	5110000020 *	LIST	M,-B,-R		
	5120000021 *	MOVE	MACROE LENGTH, FROM, TO		
31	5130000022 *	LOCAL	MOVE1		
	5140000023 *	SH1	0		
32	36612	SH2	LENGTH		
	36734	SA5	FROM+B1		
	5160000000 *	BX6	X5		
33	5170000001 *	SA6	TO+B1		
	5110000024 *	SH1	B1+1		
34	5120000025 *	NE	B1,B2,MOVE1		
	5130000026 *	ENDM			
35	5140000027 *	ANSWER	BSSZ 16		
	36612	OPERAND	DATA 1,2,3,4,5,6,7,8		
	36734	ADD	SA1 OPERAND+0		
36	5160000002 *		SA2 OPERAND+1		
	5170000003 *		SA3 OPERAND+2		
37	6110000000		SA4 OPERAND+3		
	6120000011		IX6 X1+X2		
40	5151000044 *		IX7 X3+X4		
	10655		SA6 ANSWER+0	STORE RESULT	
41	5161000030 *		SA7 ANSWER+1	STORE RESULT	
	6111000001		SA1 OPERAND+4		
42	0512000040 *		SA2 OPERAND+5		
	0400000030 *		SA3 OPERAND+6		
43	0400000030 *		SA4 OPERAND+7		
44	5110000020 *		IX6 X1+X2		
	5120000021 *		IX7 X3+X4		
45	5130000022 *		SA6 ANSWER+2		
	5140000023 *		SA7 ANSWER+3		
46	37621	MOVE	LENGTH=LBLOCK2, FROM=SUBT, TO=ADD		
	37734	SH1	0		
47	5170000005 *	SH2	LBLOCK2		
	6110000000	SA5	SUBT+B1		
50	6120000011	BX6	X5		
51	5151000055 *	SA6	ADD+B1		
	10655	SH1	B1+1		
52	5161000030 *	NE	B1,B2,*7000001		
	6111000001	ENDM			
53	0512000051 *	ENDBLK			
		SUBT	EQ ADD		
			SA1 OPERAND+0		
			SA2 OPERAND+1		
			SA3 OPERAND+2		
			SA4 OPERAND+3		
			IX6 X2-X1		
			IX7 X3-X4		
			SA6 ANSWER+4	STORE RESULT	
			SA7 ANSWER+5	STORE RESULT	
		MOVE	FROM=MULT, TO=ADD, LENGTH=LBLOCK3		
		SH1	0		
		SH2	LBLOCK3		
		SA5	MULT+B1		
		BX6	X5		
		SA6	ADD+B1		
		SH1	B1+1		
		NE	B1,B2,*7000002		

54	0400000030 *		END2BLK	ENDM				
55	5110000020 *		MULT	EQ	ADD		MOVE	.1
		5120000021 *		SA1	OPERAND+0			
56	5130000022 *			SA2	OPERAND+1			
		5140000023 *		SA3	OPERAND+2			
57	42612			SA4	OPERAND+3			
	42734			IX6	X1*X2			
		5160000006 *		IX7	X3*X4			
60	5170000007 *			SA6	ANSWER+6	STORE RESULTS		
				SA7	ANSWER+7	STORE RESULTS		
		6110000000		LIST	-M			
65	0400000030 *		END3BLK	MOVE	TO=ADD,LENGTH=LBLOCK4,FROM=DIVIDE			
66	5110000020 *		DIVIDE	EQ	ADD			
		5120000021 *		SA1	OPERAND+0			
67	5130000022 *			SA2	OPERAND+1			
		5140000023 *		SA3	OPERAND+2			
70	27101			SA4	OPERAND+3			
	27202			PX1	X1			
	24101			PX2	X2			
		24202		NX1	X1			
71	27303			NX2	X2			
	27404			PX3	X3			
		24303		PX4	X4			
		24404		NX3	X3			
72	44621			NX4	X4			
	44743			FX6	X2/X1			
		5160000010 *		FX7	X4/X3			
73	5170000011 *			SA6	ANSWER+8	STORE RESULT(FLOATING POINT)		
	26676			SA7	ANSWER+9	STORE RESULT(FLOATING POINT)		
		26767		UX6	X6,B7			
74	22676			UX7	X7,B6			
	22767			LX6	X6,B7			
		5160000012 *		LX7	X7,B6			
75	5170000013 *			SA6	ANSWER+10	STORE RESULT(INTEGER)		
76	7160247021			SA7	ANSWER+11	STORE RESULT(INTEGER)		
		14	END4BLK	ENDRUN				
		11	LBLOCK1	END1BLK-ADD+1				
		11	LBLOCK2	END2BLK-SUBT+1				
		11	LBLOCK3	END3BLK-MULT+1				
		12	LBLOCK4	END4BLK-DIVIDE+2				
		50	LBLOCK7	LBLOCK1+LBLOCK2+LBLOCK3+LBLOCK4				
		14	MAXBLK	LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4				
		11	MINBLK	LBLOCK1,LBLOCK2,LBLOCK3,LBLOCK4				
				ADD				

47500B CM STORAGE USED
MODEL 73 ASSEMBLY

111 STATEMENTS
0.331 SECONDS

21 SYMBOLS
0 REFERENCES

000003 INVENTED SYMBOLS

MACROE
ERROR DIRECTORY.

COMPASS 3.6-476.

01/08/79 10.27.31.

PAGE 3

3 TYPE ERROR
OCCURRED ON PAGES

DUPLICATE MACRO DEFINITION. NEW ONE OVERRIDES.
1

LOAD MAP - MACROE

CYBER LOADER 1.4-485

01/08/79 10.27.33.

PAGE 1

FWA OF THE LOAD 111
LVA+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

9-17

DUMP RELATIVE

DMP(111.177)

00111	00000	00000	00000	00003	00000	00000	00000	00007	00000	00000	00000	00013	00000	00000	00000	00002
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	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	61700	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.DON00K9 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

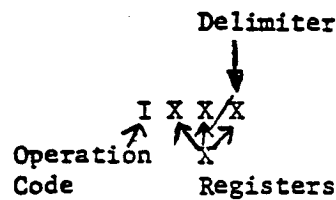
MACROS are sequences of code identified by a name: that is, whenever the name is used, the assembler brings in the associated code.

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.

MACROS
- OPDEF Syntax

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 DÍVIDE

The Call:

ACTUAL REGISTER PARAMETERS

$$1 \times 6^j \quad x_1^j / x_2^k$$

The Macro Definition:

FORMAL REGISTER PARAMETERS

IXX/X

IX1 INSTRUCTION

Xj/Xk ADDRESS FIELD

OPDEF

ENDM

P1ⁱ, P2^j, P3^k

CODE TO PERFORM INTEGER DIVIDE

IDENT	OPREF
ENTRY	INDIVID
LIST	M

1111/11

```
OPDEF EXAMPLE - INTEGER DIVIDE
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
    5110000011 .
1 5120000012 .

    27101
    24141
2 27202
    24252
    45612
    26666
3 22666

    5160000000 C
4 5130000013 .
    5150000014 .
5
5 27303
    24343
    27505
    24555
6 45735
    26767
    22767

7 5170000001 C
    7160247021
1 00000000000000000010

0
5
    2

```

[illegible]

47300B SCM STORAGE USED
MODEL 174 ASSEMBLY

49 STATEMENTS
0.049 SECONDS

4 SYMBOLS

11 REFERENCES

LOAD MAP - OPDEF

CYREN LOADER 1.4-470

06/27/78 16.31.15.

PAGE 1

FMA OF THE LOAD 111
LMA-1 OF THE LOAD 170

TRANSFER ADDRESS -- INTDIVD 113

PROGRAM ENTRY POINTS -- OPDEF 113

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSR VER LEVEL	HARDWARE	COMMENTS
/ANSWERS/	111	2					
OPDEF	113	15	LGN	06/27/78	COMPASS 3.5 470		
SYS.RM	170	40	SL-SYSLIB	05/16/78	COMPASS 3.5 470		PROCESS SYSTEM REQUEST.

.023 CM SECONDS

13200H CM STORAGE USED

1 TABLE MOVE

9-23

DUMP RELATIVE

DMP(111,177)

00111	00000 00000 00000 00004	77777 77777 77777 77776	46000 51100 00124 46000	
00114	51200 00125 27101 24141	27207 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 00171 46000	51100 00066 03110 00137	51100 00130 04000 00140	71100 00130 20160 46000
00140	13661 13161 13661 46000	51600 00111 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 00172 61000 46000	04004 00161 61000 46000	73660 20630 12161 73610	20123 03210 00157 20:51
00164	13116 20576 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
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.31.13.

PAGE 1

ADDRESS LENGTH

0 15
15

BINARY CONTROL CARDS.

IDENT OPDEF
END INTDIVD

BLOCKS TYPE ADDRESS LENGTH

PROGRAM* LOCAL 0 15
ANSWERS COMMON 0 7

ENTRY POINTS.

INTDIVD 0*

EXTERNAL SYMBOLS.

SYS*

9-24

OPDEF
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				

COMPASS 3.5-470.

06/27/78 16.31.13.

PAGE 4

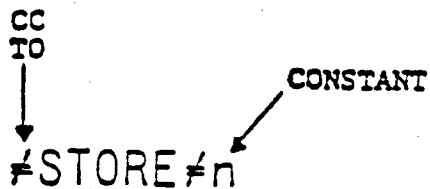
MTT NR2- CYR175-SNI 4LH7/H6R 05/15/78
16.31.11.DONDDMY FROM /SM
16.31.11.IP 00000254 WORDS - FILE INPUT , DC 04
16.31.11.DON. PS0.02/A.72C1011A.MILLER
16.31.12.COMPASS.
16.31.13. 1 WARNING MESSAGE IN OPDEF
16.31.14. ASSEMBLY COMPLETE. 473008 SEC USED.
16.31.14. 0.102 CPU SECONDS ASSEMBLY TIME.
16.31.14.LGO.
16.31.15.DMP(111.177)
16.31.15.OP 00001000 WORDS - FILE OUTPUT , DC 40
16.31.15.MS 3504 WORDS 1 10752 MAX USED)
16.31.15.CPA .152 SEC. .152 ADJ.
16.31.15.10 .635 SEC. .635 ADJ.
16.31.15.CH 10.503 KWS. .645 ADJ.
16.31.15.SS 1.433
16.31.15.PP 3.032 SEC. DATE 06/27/78
16.31.15.FJ END OF JOB. SM

MICRO

- USE A SYMBOL NAME
- TO GET A STRING OF CHARACTERS

i.e.
STORE

The Call:

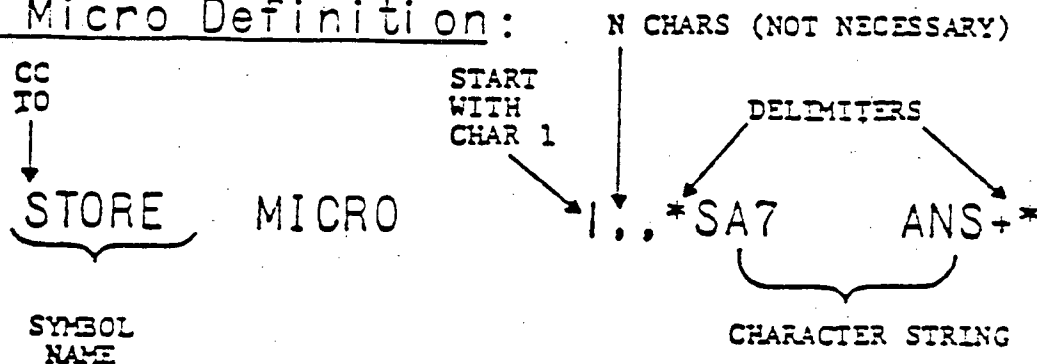


GENERATES THE CODE:

SA7

ANS+n

The Micro Definition:



IDENT MICRO

THE CHARACTER STRING OF THE MICRO IS INSERTED WHERE IT IS CALLED
BY THE #NNNNNNNNNNNN#. THE LIST A LISTS THE MICRO EXPANSION

		LOAD	ENTRY	START	
		STORE	LIST	A.0	
0	4110000001		MICRO	10.0SA1	A1.0*
	0120000002		MICRO	10.0SA7	ANSWER.0
	12 *	START	SB1	1	
			SB2	2	
1	5110000012 *	DATA	LIT	2,4,5,6	
	10711		SA1	DATA	
			BX7	X1	
2	5170000006 *		#STORE#0		
			SA7	ANSWER.0	
			#LOAD#1		
	54111		SA1	A1.01	
	10711		BX7	X1	
			#STORE#1		
3	5170000007 *		SA7	ANSWER.1	
			#LOAD#2		
	54112		SA1	A1.02	
	10711		BX7	X1	
			#STORE#2		
4	5170000010 *		SA7	ANSWER.2	
	7160247021		ENDRUN		
6	4	ANSWER	BSS7	4	

DEFAULT SYMBOLS DEFINED BY COMPASS.

0 A

SVS=

CONTENT OF LITERALS BLOCK.

12 00000000000000000002
13 00000000000000000004
14 00000000000000000005
15 00000000000000000006

B
D
E
F

16

END START

477000 SCM STORAGE USED

41 STATEMENTS

4 SYMBOLS

LOAD MAP - MICRO

TYPE: LOADER 1.4-470

06/27/70 16.31.28.

PAGE 1

LWA 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	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
MICRO	111	16	LGO	06/27/70	COMPASS 1.5 470		
SYS.RM	127	40	SL-SVSLIB	05/16/70	COMPASS 1.5 470		PROCESS SYSTEM REQUEST.

.023 CP SECONDS

13200R CM STORAGE USED

1 TABLE MOVE

9-27

DUMP RELATIVE

DMP(111,127)

00111	61100	00001	61200	00002	51100	00123	10711	46000	51700	00117	54111	10711		
00114	51700	00120	54117	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- CY0175-SN1 AL01/H00 05/15/70
 16.31.25.00000007 FROM /SM
 16.31.25.1P 00000256 WORDS - FILE INPUT . DC 04
 16.31.25.000. PS0.0270.72CT011A.MILLEN
 16.31.27.COMPASS.
 16.31.27. ASSEMBLY COMPLETE. 47300R SCH USED.
 16.31.27. 0.094 CPU SECONDS ASSEMBLY TIME.
 16.31.27.LGO.
 16.31.28.DMP(111,127)
 16.31.28.OP 00000096 WORDS - FILE OUTPUT . DC 40
 16.31.28.MS 3504 WORDS 1 10752 MAX USED
 16.31.28.CPA .146 SEC. .146 ADJ.
 16.31.28.10 .632 SEC. .632 ADJ.
 16.31.28.CM 10.416 KMS. .641 ADJ.
 16.31.28.SS 1.421
 16.31.28.PP 3.074 SEC. DATE 06/27/70
 16.31.28.FJ END OF JOB. SM

IDENT CONVERT

MICRO AND DUP PSEUDO OPERATIONS

CREATE A EXTERNAL RCD TO DISPLAY CODE CONVERSION TABLE

0		TABLE	LIST	0.0	
	D-D		HSS	0	
	1		CODE	DISPLAY CODE	
	35	D	SET	1	
			DUP	35R	
		CHAR	MICRO	0.1,811234567890-101 /STUVWXYZ).18	
			DATA	1R/CHAR#	
		D	SET	0.1	
			FNDD		
0	00000000000000000000		DATA	1R1	*DUP* .1
1	000000000000000000034		DATA	1R1	*DUP* .1
2	000000000000000000035		DATA	1R2	*DUP* .1
3	000000000000000000036		DATA	1R3	*DUP* .1
4	000000000000000000037		DATA	1R4	*DUP* .1
5	000000000000000000040		DATA	1R5	*DUP* .1
6	000000000000000000041		DATA	1R6	*DUP* .1
7	000000000000000000042		DATA	1R7	*DUP* .1
10	000000000000000000043		DATA	1R8	*DUP* .1
11	000000000000000000044		DATA	1R9	*DUP* .1
12	000000000000000000033		DATA	1R0	*DUP* .1
13	000000000000000000054		DATA	1R-	*DUP* .1
14	000000000000000000074		DATA	1R1	*DUP* .1
15	000000000000000000063		DATA	1R0	*DUP* .1
16	000000000000000000061		DATA	1R1	*DUP* .1
17	000000000000000000055		DATA	1R	*DUP* .1
20	000000000000000000050		DATA	1R/	*DUP* .1
21	000000000000000000023		DATA	1R5	*DUP* .1
22	000000000000000000024		DATA	1R7	*DUP* .1
23	000000000000000000025		DATA	1R0	*DUP* .1
24	000000000000000000026		DATA	1R0	*DUP* .1
25	000000000000000000027		DATA	1R0	*DUP* .1
26	000000000000000000030		DATA	1R1	*DUP* .1
27	000000000000000000031		DATA	1R1	*DUP* .1
30	000000000000000000032		DATA	1R2	*DUP* .1
31	000000000000000000062		DATA	1R1	*DUP* .1
32	000000000000000000056		DATA	1R1	*DUP* .1
33	000000000000000000051		DATA	1R1	*DUP* .1
34	000000000000000000055		DATA	1R	*DUP* .1
35	000000000000000000065		DATA	65R	
	1	D	SET	1	
	42		DUP	42R	
		CHAR	MICRO	0.1,73A-JKLMNOPQRS+*+*+ARCEFGHIS.12 1/	
			DATA	1R/CHAR#	
		D	SET	0.1	
			FNDD		
36	000000000000000000060		DATA	1R3	*DUP* .1
37	000000000000000000067		DATA	1R4	*DUP* .1
40	000000000000000000046		DATA	1R-	*DUP* .1
41	000000000000000000012		DATA	1R1	*DUP* .1

PAGE 1

[illegible]

CODE	OTHER
CHAR	0.33H
CHAR	33H.0
SET	1
DUP	35H
MICRO	0.1.3
DATA	1H/CHA
SET	0.1

```

ENDD
DATA 1R1
DATA 1R1
DATA 1R2
DATA 1R3
DATA 1R4
DATA 1R5
DATA 1R6
DATA 1R7
DATA 1R8
DATA 1R9
DATA 1R0
DATA 1R=
DATA 1R<
DATA 1R#
DATA 1R!
DATA 1R
DATA 1R/

```

[illegible]

9-29

CONVERT

COMPASS 3.5-470.

06/27/78 08.40.57.

PAGE 4

122	000000000000000000024	DATA	IRI		
123	000000000000000000025	DATA	IRU		*DUP* .1
124	000000000000000000026	DATA	IRV		*DUP* .1
125	000000000000000000027	DATA	IRW		*DUP* .1
126	000000000000000000030	DATA	IRX		*DUP* .1
127	000000000000000000031	DATA	IRY		*DUP* .1
130	000000000000000000032	DATA	IRZ		*DUP* .1
131	000000000000000000062	DATA	IRJ		*DUP* .1
132	000000000000000000056	DATA	IR.		*DUP* .1
133	000000000000000000051	DATA	IRI		*DUP* .1
134	000000000000000000055	DATA	IR		*DUP* .1
	0=0	CODE	*		
	1	SET	1		
	35	DUP	75R		
		CHAR	MICRO	0.1.811234567890=IRI /STUVWXYZ).18	
			DATA	IR/CHAR/	
		D	SET	0=1	
			FNDD		
135	000000000000000000000	DATA	IRI		*DUP* .1
136	00000000000000000000034	DATA	IRI		*DUP* .1
137	00000000000000000000035	DATA	IR2		*DUP* .1
140	00000000000000000000036	DATA	IR3		*DUP* .1
141	00000000000000000000037	DATA	IR4		*DUP* .1
142	00000000000000000000040	DATA	IR5		*DUP* .1
143	00000000000000000000041	DATA	IR6		*DUP* .1
144	00000000000000000000042	DATA	IR7		*DUP* .1
145	00000000000000000000043	DATA	IR8		*DUP* .1
146	00000000000000000000044	DATA	IR9		*DUP* .1
147	00000000000000000000033	DATA	IR0		*DUP* .1
150	00000000000000000000054	DATA	IR.		*DUP* .1
151	00000000000000000000074	DATA	IR4		*DUP* .1
152	00000000000000000000063	DATA	IR4		*DUP* .1
153	00000000000000000000061	DATA	IRI		*DUP* .1
154	00000000000000000000055	DATA	IR		*DUP* .1
155	00000000000000000000050	DATA	IR/		*DUP* .1
156	00000000000000000000023	DATA	IRS		*DUP* .1
157	00000000000000000000024	DATA	IRI		*DUP* .1
160	00000000000000000000025	DATA	IRU		*DUP* .1
161	00000000000000000000026	DATA	IRV		*DUP* .1
162	00000000000000000000027	DATA	IRW		*DUP* .1
163	00000000000000000000030	DATA	IRX		*DUP* .1
164	00000000000000000000031	DATA	IRY		*DUP* .1
165	00000000000000000000032	DATA	IRZ		*DUP* .1
166	00000000000000000000062	DATA	IRJ		*DUP* .1
167	00000000000000000000056	DATA	IR.		*DUP* .1
170	00000000000000000000051	DATA	IRI		*DUP* .1
171	00000000000000000000055	DATA	IR		*DUP* .1
172		END			

47300R SCM STORAGE USED
MODEL 174 ASSEMBLY

284 STATEMENTS
0.356 SECONDS

2 SYMBOLS
247 REFERENCES

9-30

CONVERT
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 08.40.57.

PAGE 1

ADDRESS LENGTH

BINARY CONTROL CARDS.

0 172
172

IDFNT CONVERT
END

CONVERT
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 08.40.57.

PAGE 5

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	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	
2/10 L							

TABLE 0 PROGRAM*

MFF NR2- CYR175-SNI 4LB7/R6B 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. 47300R SCH USED.

08.40.58. 0.41R 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.I0 .506 SEC. .506 ADJ.

08.40.58.CH 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

FIN 4.7+470

06/27/78 16.32.03

PAGE 1

```

1      PROGRAM SWITCH(INPUT,OUTPUT)
      COMMON A(12)
      PRINT 100
100    FORMAT('H')
5      READ 7,(A(I)),I=1,12)
      PRINT 7,(A(I)),I=1,12)
      CALL TRAPOS
      FORMAT(4F20.0)
      END
  
```

SUBROUTINE UPRINT 74/176 OPT=1

FIN 4.7+470

06/27/78 16.32.03

PAGE 1

```

1      SUBROUTINE UPRINT
      COMMON /BLOCK)/STORE(12)
      PRINT 5,(STORE(I)),I=1,12)
5      FORMAT(1X,4F20.4,///)
      END
  
```

9-32

MICRO
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.32.05.

PAGE 1

ADDRESS LENGTH

0 23
23

HIGHWAY CONTROL CARDS.

IDENT MICRO
END

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM*	LOCAL	0	23
BLOCK1	COMMON	0	14
//	COMMON	0	14

ENTRY POINTS.

TRAPOS 0*

EXTERNAL SYMBOLS.

UPRINT1

10FNT MICRO
 ENTRY TRAPOS
 EXT UPPRINT
 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 R1
 SA7 R1.1
 BX6 X3
 BX7 X4
 SA6 R1.2
 SA7 R1.3
 ENDM

0 000000000000000000
 1 0110000000 C

TRAPOS DATA 0
 SB1 STORE
 SAM #ROW1#
 SAM M.M.3,M.6,M.9

2 5110000000 C
 3 5120000003 C
 5130000006 C
 5140000011 C
 10611 10722
 4 56610 5171000001 10633
 5 10744 5161000002
 6 5171000003

SA1 M
 SA2 M.3
 SA3 M.6
 SA4 M.9
 BX6 X1
 BX7 X2
 SA6 R1
 SA7 R1.1
 BX6 X3
 BX7 X4
 SA6 R1.2
 SA7 R1.3
 ENDM

SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1

6110000004 C

SB1 STORE.4
 LIST -A
 SAM #ROW2#
 SA1 M.1
 SA2 M.4
 SA3 M.7
 SA4 M.10
 BX6 X1
 BX7 X2
 SA6 R1
 SA7 R1.1
 BX6 X3
 BX7 X4
 SA6 R1.2

7 5110000001 C
 5120000004 C
 10 5130000007 C
 5140000012 C
 11 10611 10722 56610
 12 5171000001 10633 10744
 13 5161000002

SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1
 SAM .1

```

5171000003          SA7  RI.3          SAM .1
                     ENOM                SAM .1

14 6110000010 C      SH1  STORE.8
                     LIST  -M
                     SAM   #HOWJ#

                     5110000002 C
                     0100000000 X
22 0400000000 .      RJ   UPPINT1
                     FO   TRAPUS
                     LIST  *
23                     RSS  0          GET ADDRESS OF NEXT WORD
                     23  AA  DECMIC LASTWD-TRAPUS.4 GET DECIMAL LENGTH OF PROGRAM
                     23  AB  OCTMIC LASTWD-TRAPUS.6 GET OCTAL LENGTH OF PROGRAM
                     RAG5  MICRO 1..*PAA# IS THE DECIMAL LENGTH OF THIS PROGRAM*
                     RAG5  MICRO 1..*0019 IS THE DECIMAL LENGTH OF THIS PROGRAM*
                     RAG6  MICRO 1..*PAH# IS THE OCTAL LENGTH OF THIS PROGRAM*
                     RAG6  MICRO 1..*00023 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  RSS  12
                     USE  //
0                     14  M      RSS  12
                     USE  0
23                     END

```

474008 SCH STORAGE USED
MODEL 174 ASSEMBLY

101 STATEMENTS
0.118 SECONDS

7 SYMBOLS
31 REFERENCES

MICRO SYMBOLIC REFERENCE TABLE.

DECLENG	52		3/19 D	3/21				
LASTWD	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		
TRAPUS	0	PROGRAM*	2/02 F	2/25 L	3/09	3/12	3/13	
UPPINT1	0	EXTERNAL*	2/03 X	3/08				

LWA 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	PROCESSOR VER LEVEL	HARDWARE	COMMENTS
SWITCH	111	4171	LGO	06/27/78	FIN 4.7 470	767X 1	PROGRAM OPT=1
/BLOCK1/	4302	14					
UPRINT1	4314	15	LGO	06/27/78	FIN 4.7 470	767X 1	SUBROUTINEOPT=1
MICRO	4333	23	LGO	06/27/78	COMPASS 3.5 470		
SYS=ID=	4354	1	SL-FORTRAN	05/02/78	COMPASS 3.5 470		LINK BETWEEN SYS=ID AND INITIALIZATION CODE.
/SIP.END/	4357	1					
/FCL.C./	4360	25					
/OR.IO./	4405	77					
QENTRY=	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=FUL	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		CHECK 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.
FECHSK=	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 HEAD 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.AL	10475	160	SL-SYSLIB	05/02/78	COMPASS 3.5 470		CMF V1.1 - ALLOCATE FIXED.
CMF.CSF	10655	6	SL-SYSLIB	05/02/78	COMPASS 3.5 470		CMF V1.1 - CHANGE SPECS FIXED.
CMF.FFA	10663	14	SL-SYSLIB	05/02/78	COMPASS 3.5 470		CMF V1.1 - FIXED FREE ALGORITHM.
CMF.FHF	10677	36	SL-SYSLIB	05/02/78	COMPASS 3.5 470		CMF V1.1 - FREE FIXED.
CMF.R	10735	213	SL-SYSLIB	05/02/78	COMPASS 3.5 470		CMF V1.1 - RESIDENT SUBROUTINES.
CMF.SLF	11150	27	SL-SYSLIB	05/02/78	COMPASS 3.5 470		CMF V1.1 - SHRINK AT LWA FIXED.
CTLBRM	11172	401	SL-SYSLIB	05/02/78	COMPASS 3.5 470		CMF CONTROLLING ROUTINE.
ERRBRM	11773	25	SL-SYSLIB	05/02/78	COMPASS 3.5 470		CMF ERROR PROCESSOR ENTRY.
LSTBRM	12020	66	SL-SYSLIB	05/02/78	COMPASS 3.5 470		CMF - ALLOCATE SPACE FOR LIST OF FILES
/FOL.COM/	12106	14					
FOL.PES	12127	211	SL-SYSLIB	05/16/78	COMPASS 3.5 470		FAST DYNAMIC LOADER RESIDENT.
FOL.MMI	12133	222	SL-SYSLIB	05/16/78	COMPASS 3.5 470		FOL MEMORY MANAGER INTERFACE.
SYS.PM	12555	40	SL-SYSLIB	05/16/78	COMPASS 3.5 470		PROCESS SYSTEM REQUEST.
//	12615	14					

9-35

1. 5. 9. 1.0000	2. 6. 10. 4.0000	3. 7. 11. 7.0000	4. 8. 12. 10.0000
2.0000	5.0000	8.0000	11.0000
3.0000	6.0000	9.0000	12.0000

MFF NR2- CYR175-SNI 4L87/R6B 05/15/78
 16.32.01.DON00M0 FROM /SM
 16.32.01.JP 00000448 WORDS - FILE INPUT , DC 04
 16.32.01.DON. PSN: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.
 16.32.11.10 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

IDENT IRPECHO
ENTHY 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
ENDM
ENDM      END OF IRP RANGE

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
.

```
IRPMO      (J,K,L),1      CALL IRP MACRO
IRP        J,K,L
SA1        J
SX6        X1+1
SA6        J
SA2        DATA
BX7        X2
SA7        DATA+1
SA1        K
SX6        X1+1
SA6        K
SA2        DATA
BX7        X2
SA7        DATA+1
SA1        L
SX6        X1+1
SA6        L
SA2        DATA
BX7        X2
SA7        DATA+1
IRP
ENDM      END OF IRP RANGE
```

```
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
IRPMO      .1
```

```
5110000000 C
1 7261000001 5160000000 C
2 5120000003 C 10722
3 5170000004 C 5110000001 C
4 7261000001 5160000001 C
5 5120000003 C 10722
6 5170000004 C 5110000002 C
7 7261000001 5160000002 C
10 5120000003 C 10722
11 5170000004 C
```

5110000000 C
 12 7261000001 5160000000 C
 13 5110000001 C 7261000001
 14 5160000001 C 5110000002 C
 15 7261000001 5160000002 C
 16 5120000003 C 10722
 17 5170000004 C

7160247021

0 00000000000000000000
 1 00000000000000000000
 2 00000000000000000000
 3 00000000000000000000
 21

J
 K
 L
 DATA

ECHOMO J,K,L CALL ECHO MACRO
 ECHO 3,ARG=(J,K,L)
 SA1 ARG
 SX6 X1+1
 SA6 ARG
 SA1 J
 SX6 X1+1
 SA6 J
 SA1 K
 SX6 X1+1
 SA6 K
 SA1 L
 SX6 X1+1
 SA6 L
 SA2 DATA
 DX7 X2
 SA7 DATA+1
 ENDM

ENDRUN

USE /IRPECHO/
 DATA 0
 DATA 0
 DATA 0
 DATA +0,-0
 END P

ECHOMO .1
 ECHOMO .1
 ECHOMO .1
 ECHOMO .1
 ECHO .2
 ECHO .2
 ECHO .2
 ECHO .2
 ECHO .2
 ECHO .2
 ECHO .2
 ECHO .2
 ECHO .2
 ECHOMO .1
 ECHOMO .1
 ECHOMO .1
 ECHOMO .1

47600B CM STORAGE USED
 MODEL 73 ASSEMBLY

85 STATEMENTS
 0.304 SECONDS

6 SYMBOLS
 0 REFERENCES

MFS N01- CY874-SN108 5C/R08 11/14/78
 10.23.42.DUN00K2 FROM /OH
 10.23.43.1P 00000320 WORDS - FILE INPUT , DC 04
 10.23.43.DON.75. 001A,6803,1896,MILLER
 10.23.53.REWIND,OUTPUT.
 10.25.30.COMPASS,LO=UREM.
 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.HS 3584 WORDS (7168 MAX USED)
 10.25.32.CPA .292 SEC. .292 ADJ.
 10.25.32.CPB .327 SEC. .327 ADJ.
 10.25.32.10 .311 SEC. .311 ADJ.
 10.25.32.CH 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,10H

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

[illegible]

```

graph TD
    NAME --> LOAD
    REGISTER --> REG_ADDR[REG, ADDR]
    ADDRESS --> REG_ADDR
    REG_ADDR --> LE[LE, $REG$5$, 2]
    LE --> ADDR[ADDR]
    ADDR --> IS_REG[IS REG  
PARAMETER  
LEGAL?]
    IS_REG --> IF_YES[IF YES,  
ASSEMBLE  
NEXT 2 LINES]
    IF_YES --> LE
    IF_NO[IF NO,  
ASSEMBLE  
ERR  
ENDM]
    IF_NO --> IFPP[IFPP]
    IFPP --> IF_YES
  
```

NAME

REGISTER

ADDRESS

LOAD

MACRO

IFC

IFPP

ERR

ENDM

ASSEMBLE THIS
IF PARAMETER
LEGAL

ASSEMBLE THIS
IF PARAMETER
ILLEGAL

SA,REG

1

REG, ADDR

LE, \$REG\$5\$, 2

ADDR

IS
REG
PARAMETER
LEGAL?

IF YES,
ASSEMBLE
NEXT 2 LINES

THE CALLS:

SA.2 Y

ERR

AN ILLEGAL CALL
GENERATES
THIS CODE

10-3

IFC/PP
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.20.24.

PAGE 1

ADDRESS LENGTH

BINARY CONTROL CARDS.

0 4
4

IDENI IFC/PP
END START

ENTRY POINTS.

START 1.

EXTERNAL SYMBOLS.

SYS=

IFC/PP

COMPASS 3.5-470.

06/27/78 16.20.24.

PAGE 3

47300R SCH STORAGE USED
MODEL 174 ASSEMBLY

52 STATEMENTS
0.050 SECONDS

7 SYMBOLS
5 REFERENCES

1 ERROR IN IFC/PP

IFC/PP
ERROR DIRECTORY.

COMPASS 3.5-470.

06/27/78 16.20.24.

PAGE 4

P TYPE FROM
OCCURRED ON PAGES

CONSULT LISTINGS FOR REASON BEHIND P-FRONT.
2

IFC/PP
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 16.20.24.

PAGE 5

START	1	PROGRAM*	2/02 E	2/29 I
SYS=	0	EXTERNAL*	2/50	
Y	0	PROGRAM*	2/28 L	2/34

HFF HB2- CYR179-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. 47300R SCH 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.
16.20.24.10 .497 SEC. .497 ADJ.
16.20.24.CH 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 CONDASH
ENTRY CONDASH
LIST D:G

IFLE AND IFGT TEST VALUE OF PARAMETER VAL TO DETERMINE WHETHER TO SET
AN X REGISTER WITH A CONSTANT (410 BITS) OR SET AN A REGISTER TO A
LITERAL (110 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,777777D,1
SX,REG VAL SET X IF A SMALL NUMBER
IFGT VAL,777777D,1
SA,REG =VAL SET A IF LARGE NUMBER
ENDM

0 3 ANSWER BSSZ 3
CONDASH SETX 1,777777H SETX .1
3 7110777776 SX,1 777776H
10411 RXA X1
4 5160000000 SA, ANSWER+0
SETX 2,54777766H SETX .1
5 10622 5120000011 SA,2 =54777766H SET A IF LARGE NUMBER
5160000001 RXA X2
SA, ANSWER+1
6 5130000012 SETX 3,10000000H SETX .1
10633 SA,3 =10000000H SET A IF LARGE NUMBER
7 5160000002 RXA X3
SA, ANSWER+2
LIST -G
7160247021 ENDRUN

DEFAULT SYMBOLS DEFINED BY COMPASS.

0 R

SYS=

CONTENT OF LITERALS BLOCK.

11 00000000000005477766
12 00000000000001000000

E+1v
A

13

END CONDASH

47100R SCH STORAGE USED
MODEL 174 ASSEMBLY

45 STATEMENTS
0.045 SECONDS

3 SYMBOLS
7 REFERENCES

LOAD MAP - CONDASH

CYBER LOADER 1.4-470

06/27/70 09.00.20.

PAGE 1

FMA OF THE LOAD 111
LVA-1 OF THE LOAD 164

TRANSFER ADDRESS -- CONDASH 114

PROGRAM ENTRY POINTS -- CONDASH 114

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSR VER LEVEL	HARDWARE	COMMENTS
CONDASH	111	13	LGO	06/27/70	CUMPASS 3.5	470	
SYS.RM	124	40	SL-SYSLIB	05/16/70	CUMPASS 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
00114	71107 77776 10611 46000	51600 00111 51200 00122
00120	51600 00113 71607 47021	20650 01000 00126 46000
00124	04000 00137 00000 00000	01300 00000 00000 00000

HFF NR2- CYR175-SM1 4L07/R60 05/15/70
09.00.25.DON0059 FROM /SH
09.00.25.IP 00000320 WORDS - FILE INPUT , DC 04
09.00.25.DON, PSD.0278.72CT011A,MILLER
09.00.26.COMPASS.
09.00.27. ASSEMBLY COMPLETE. 47300R SCH USED.
09.00.27. 0.097 CPU SECONDS ASSEMBLY TIME.
09.00.27.LGO.
09.00.28.DMP(111,127)
09.00.28.OP 00000096 WORDS - FILE OUTPUT , DC 40
09.00.28.MS 3504 WORDS (10752 MAX USED)
09.00.28.CPA .150 SEC. .150 ADJ.
09.00.28.10 .433 SEC. .433 ADJ.
09.00.28.CH 10.637 HWS. .649 ADJ.
09.00.28.SS 1.433
09.00.28.PP 3.029 SEC. DATE 06/27/70
09.00.28.EJ END OF JOB, SH

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
	DX6	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
	EX.R1	-X.R1*X.R2
	AX.R1	LAST
	ELSE	
	MX.R1	60-LAST-1
	EX.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
EX.R1     -X.R1*X.R2
AX.R1     LAST
ELSE
MX.R1     60-LAST-1
EX.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
	EX.R1	-X.R1*X.R2
	AX.R1	LAST
If test was true, skip.	ELSE	
If test was false, assemble.	MX.R1	60-LAST-1
	EX.R1	-X.R1*X.R2
	AX.R1	FIRST
	ENDIF	

ANTIBUG
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/70 09.05.16.

PAGE 1

ADDRESS LENGTH

0 13
13

PRIMARY CONTROL CARDS.

IDENT ANTIBUG
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 8

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 DESIRED, THE STATEMENT #DEBUG EQU 1 IS REMOVED AND THE CHECK IS NOT MADE.

0	2	DATAIN	USE	/NUMBERS/	
			BSS	2	
	1	DEBUG	USE	0	
0		ANTIBUG	EQU	1	
1			NO		
1	46000		SK0	770	SET UP MASK
1	7100000077		SA1	DATAIN	
2	11201	5110000000 C	BK2	X0=X1	ISOLATE LOW ORDER CHARACTER OF X1
	63620		IF	DEF,DEBUG,5	TEST IF DEBUG IS SELECTED
	6170000033		SB6	X2	TEST IF X2 IS BETWEEN 0 AND 9
3	0767777776		SB7	1R0	
	6170000044		LT	R6,B7,-1	ABORT JOB IF X2 LT DISPLAY CODE 0
4	0776777776		SB7	1R9	
			GT	R6,B7,-1	ABORT JOB IF X2 GT DISPLAY CODE 9
	20106		IF	DEF,DEBUG	
	11201		LX1	4	
5	63620		BK2	X0=X1	
	6170000033		SB6	X2	
6	0767777776		SB7	1R0	
	6170000044		LT	R6,B7,-1	
7	0776777776		SB7	1R9	
			GT	R6,B7,-1	
			ENDIF		
	63620		IF	DEF,DEBUG,5	
10	6170000033		SB6	X2	
	0767777776		SB7	1R0	
			LT	R6,B7,-1	
			ELSE	2	
			SB7	1R9	
			GT	R6,B7,-1	
			IF	-DEF,DEBUG,5	
			SB6	X2	
			SB7	1R0	
			LT	R6,B7,-1	
			ELSE		
11	6170000044		SB7	1R9	
	0776777776		GT	R6,B7,-1	

10-10

ANTIBUG

COMPASS 3.5-470.

06/27/78 09.05.14.

PAGE 3

CONTINUE PROGRAM

12 0400000000

13

TO ANTIHUG RETURN TO CALLER
END NO TRANSFER ADDRESS FOR SUBROUTINE

47100H CM STORAGE USED

MODEL 174 ASSEMBLY

61 STATEMENTS

0.057 SECONDS

3 SYMBOLS

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

DERUG

1

2/22 D

2/24 F

2/35 F

2/44 F

2/51 F

10-11

LOAD MAP - ANTIHUG

CYBER LOADER 1.4-470

06/27/78 09.05.14.

PAGE 1

FVA OF THE LOAD

111

LVA-1 OF THE LOAD

126

TRANSFER ADDRESS -- ANTIHUG

113

PROGRAM ENTRY POINTS --

ANTIBUG

113

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK

ADDRESS

LENGTH

FILE

DATE

PROCESSOR VER LEVEL

HARDWARE

COMMENTS

/NUMBERS/

111

2

ANTIBUG

113

13

LGO

06/27/78 COMPASS 3.5 470

.014 CP SECONDS

13200H CM STORAGE USED

1 TABLE MOVE

DMP.

P 000000 A0 000200 R0 000000
 RA 264000 A1 000111 R1 000001
 FL 000200 A2 000060 R2 000002
 FM 700700 A3 000057 R3 012733
 RE 000055 A4 000001 R4 000201
 FE 000000 A5 000113 R5 000113
 MA 001200 A6 000001 R6 000011
 A7 000001 R7 000033
 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

CIA1)= 6000 0000 0004 0040 0111
 CIA2)= 1505 1520 0000 0000 0061
 CIA3)= 0000 0000 0000 0000 0000
 CIA4)= 0000 0000 0000 0000 0000
 CIA5)= 4600 0460 0061 0004 6000
 CIA6)= 0000 0000 0000 0000 0000
 CIA7)= 0000 0000 0000 0000 0000

CIR1)= 0000 0000 0000 0000 0000
 CIR2)= 0000 0000 0000 0000 0000
 CIR3)=
 CIR4)=
 CIR5)= 4600 0460 0061 0004 6000
 CIR6)= 0000 0000 0000 0000 0000
 CIR7)= 0000 0000 0000 0000 0000

00000 00017 77776 00000 00000
 00054 56110 03110 00054 54710
 00060 15051 52000 00000 00061
 00064 14071 70000 00000 00000
 00070 05300 50325 24055 10116
 00100 54000 00000 01000 00001

00000 00000 00000 00000
 51100 00001 03110 00055
 00000 00200 00000 00001
 00000 00000 00000 00126
 24110 22507 52550 00000

64550 02550 00000 46000
 07040 00060 51600 00001
 40000 00000 02000 00111
 00000 00000 00000 00000

00000 00000 00000 00000
 04000 00063 00000 00021
 00000 00000 40000 00000

DUMP RELATIVE

00111 60000 00000 04004 00111
 00114 71000 00077 51100 00111
 00120 63620 61700 00033 46000
 00124 61700 00044 07747 77776

DMP.111.177.

00113>46000 46000 61000 46000
 11201 63620 61700 00033
 07677 77776 61700 00044
 04000 00113 61000 46000

07677 77776 61700 00044

07767 77776 20106 11201

07767 77776 63620 46000

61700 00033 07677 77776

40000 00000 04004 00126

00177>60000 00000 04004 00177

10-12

MFF NR2- CYN175-SN1 4L87/R68 05/15/78
 09.05.10.DON005Y FROM /SM
 09.05.10.1P 00000512 WORDS - FILE INPUT , DC 04
 09.05.10.DON. PSC.027A.72CT011A,MILLER
 09.05.13.COMPASS.
 09.05.14. ASSEMBLY COMPLETE. 47300R SCH USED.
 09.05.15. 0.108 CPU SECONDS ASSEMBLY TIME.
 09.05.15.LOAD(ILGO)
 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 1 10752 MAX USED1
 09.05.16.CPA .146 SEC. .146 ADJ.
 09.05.17.10 .650 SEC. .650 ADJ.
 09.05.17.CH 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 CPT=1

FIN 4.6-444

02/25/77 16.09.59

PAGE 1

```

1      PROGRAM INITIAL(OUTPUT)
        COMMON /ANSWER/ BUFFER(16)
        CALL START
5      PRINT 100,(BUFFER(I),I=1,16)
        FORMAT(14022)
        END

```

SUBROUTINE ERRMSG1 74/74 CPT=1

FIN 4.6-444

02/25/77 16.09.59

PAGE 1

```

1      SUBROUTINE ERRMSG1
        PRINT 100
        100 FORMAT(1H),* ECS WHITE ERROR*
5      RETURN
        END

```

SUBROUTINE ERRMSG2 74/74 OPT=1

FIN 4.6-444

02/25/77 16.09.59

PAGE 1

```

1      SUBROUTINE ERRMSG2
        PRINT 100
        100 FORMAT(1H),* ECS READ ERROR*
5      RETURN
        END

```

LOC7600
STORAGE ALLOCATION.

COMPASS 3.4-444.

02/25/77 16.10.08.

PAGE 1

ADDRESS LENGTH

0 56
56

BINARY CONTROL CARDS.

IDENT LOC7600
END

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM*	LOCAL	0	56
ANSWER	COMMON	0	20

ENTRY POINTS.

START 10*

EXTERNAL SYMBOLS.

ERRMSG1 ERRMSG2

10-13

			IDENT LOC7600
			MACHINE 7
			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	5100000014 *		SA0 ADD
	7100000000		SX0 0
			IFCP6 4
12	0120000042		IFCP7 2
	0110000012		WL LBLOCKT
13	0400000014 *		RL LBLOCK1
14	5110000000 *	ADD	EQ ADD
	5120000001 *		SA1 OPERAND*0
15	5130000002 *		SA2 OPERAND*1
	5140000003 *		SA3 OPERAND*2
16	36612 *		SA4 OPERAND*3
	36734		IX6 X1-X2
	5160000000 C		IX7 X3-X4
17	5170000001 C		SA6 ANSWER*0
	5110000004 *		SA7 ANSWER*1
20	5120000005 *		SA1 OPERAND*4
	5130000006 *		SA2 OPERAND*5
21	5140000007 *		SA3 OPERAND*6
	36612		SA4 OPERAND*7
	36734		IX6 X1-X2
22	5160000002 C		IX7 X3-X4
	5170000003 C		SA6 ANSWER*2
23	5100000014 *		SA7 ANSWER*3
	7100000012		SA0 ADD
			SX0 LBLOCK1
			IFCP6
			ENDIF
24	0110000007		IFCP7 1
25	0100000014 *	END1BLK	RL LBLOCK2
			RJ ADD
L	14		LOC ADD
L	14	00000000000000000000	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
L	22	0110000007	RL LBLOCK3
L	4	0100000014 *	RJ ADD
L		00000000000000000000	LOC ADD
		MULT	DATA 0

10-14

L	15	5110000000 *		SA1	OPERAND*0
		5120000001 *		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		SA5	ANSWER*6
L	20	5170000007 C		SA7	ANSWER*7
		5100000014 *		SA0	ADD
L	21	7100000030		SX0	LBLOCK1*LBLOCK2*LBLOCK3
				IFCP6	2
				IFCP7	1
L	22	0100000014 *	END3BLK	RL	LHLOCK4
L	14			RJ	ADD
L	14	000000000000000000		LOC	ADD
L	15	5110000000 *	DIVIDE	DATA	0
		5120000001 *		SA1	OPERAND*0
L	16	5130000002 *		SA2	OPERAND*1
		5140000003 *		SA3	OPERAND*2
L	17	27101		SA4	OPERAND*3
		27202		PX1	X1
		24101		PX2	X2
		24202		NX1	X1
L	20	44621		NX2	X2
		27303		FX6	X2/X1
		27404		PX3	X3
		24303		PX4	X4
L	21	24404		NX3	X3
		44743		NX4	X4
		5160000010 C		FX7	X4/X3
L	22	5170000011 C		SA6	ANSWER*8
		26666		SA7	ANSWER*9
		26777		UX6	X6*U5
L	23	22666		UX7	X7*U7
		22777		LX6	X6*U6
		5160000012 C		LX7	X7*U7
L	24	5170000013 C		SA6	ANSWER*10
L	25	0400000010 *		SA7	ANSWER*11
			END4BLK	EO	START
	12		LBLOCK1	EOU	END1BLK-ADD*1
	7		LBLOCK2	EOU	END2BLK-SURT*1
	7		LBLOCK3	EOU	END3BLK-MULT*1
	12		LBLOCK4	EOU	END4BLK-DIVIDE*1
	42		LBLOCK7	EOU	LBLOCK1*LBLOCK2*LBLOCK3*LBLOCK4
	12		MAXBLK	MAX	LBLOCK1*LBLOCK2*LBLOCK3*LBLOCK4
	7		MINBLK	MIN	LBLOCK1*LBLOCK2*LBLOCK3*LBLOCK4
			END	END	

56

47200B CM STORAGE USED
MODEL 74 ASSEMBLY

114 STATEMENTS
0.343 SECONDS

20 SYMBOLS
94 REFERENCES

10-15

LOC7600
SYMBOLIC REFERENCE TABLE.

COMPASS 3.4-444.

02/25/77 16.10.08.

PAGE 4

ADD	14	PROGRAM	2/10	2/17 L	2/39	2/50	2/56	3/14	3/40	
ANSWER	0	ANSWER	2/16	2/33	2/40	2/55	3/09	3/15		
			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		
CIVIDE	14	PROGRAM	3/16 L	3/43						
ENC1BLK	25	PROGRAM	2/39 L	3/40						
END2HLK	22	PROGRAM	2/55 L	3/41						
ENC3HLK	22	PROGRAM	3/14 L	3/42						
END4BLK	25	PROGRAM	3/39 L	3/43						
EPMSU1	0	EXTERNAL	2/04 X							
EPMSU2	0	EXTERNAL	2/04 X							
LPLCKT	42		2/14	3/44 D						
LRLCK1	12		2/15	2/34	2/51	3/10	3/40 D	3/44	3/45	3/46
LPLCK2	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			
LRLCK4	12		3/13	3/43 D	3/44	3/45				
PAXBLK	12		3/45 D							
WLABLK	7		3/46 D							
MULT	14	PROGRAM	2/57 L	3/42						
CPERAND	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	
			2/03 E	2/09 L	3/39					
START	10	PROGRAM	2/41 L	3/41						
SUBT	14	PROGRAM								

MFS NOS/BE 1.2 R443A/R3B 02/02/77
 16.09.39.DON00MY FROM /LE
 16.09.39.1P 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.PS 3584 WORDS 1 14336 MAX USED!
 16.10.09.CPA .636 SEC. .636 ADJ.
 16.10.09.CPB .146 SEC. .146 ADJ.
 16.10.09.10 2.124 SEC. 2.124 ADJ.
 16.10.10.CH 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

FIN 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 FCRMAT(4022)
        END
  
```

SUBROUTINE ERRMSG1 74/74 CPT=1

FIN 4.6-444

02/22/77 06.31.07

PAGE 1

```

1      SUBROUTINE ERRMSG1
        PRINT 100
        100 FORMAT(1H1,0 ECS WRITE ERROR=)
        RETURN
5      END
  
```

SUBROUTINE ERRMSG2 74/74 CPT=1

FIN 4.6-444

02/22/77 06.31.07

PAGE 1

```

1      SUBROUTINE ERRMSG2
        PRINT 100
        100 FCRMAT(1H1,0 ECS READ ERROR=)
        RETURN
5      END
  
```

LOC7600
STORAGE ALLOCATION.

COMPASS 3.4-444.

02/22/77 06.31.12.

PAGE 1

ADDRESS LENGTH

BINARY CONTROL CARDS.

0 61
61

IDENT LOC7600
END

ELOCKS TYPE ADDRESS LENGTH

PROGRAM* LOCAL 0 61
ANSWER COMMON 0 20

ENTRY POINTS.

START 10*

EXTERNAL SYMBOLS.

ERRMSG1 ERRMSG2

10-17

			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		SA0 0
12	0120000044		IFCP6 4
	0100000000 X		WE LBLOCK1
13	0110000012		RJ ERRMSG1
	0100000000 X		RE LBLOCK1
			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 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		SA0 LBLOCK1
25	0110000010		IFCP6
	0100000000 X		RE LBLOCK2
			RJ ERRMSG2
			ENDIF
			IFCP7 1
26	0100000015 *	END18LK	RJ ADD
L 15	00000000000000000000	SUBT	LOC ADD
L 16	5110000000 *		DATA 0
	5120000001 *		SA1 OPERAND*0
L 17	5130000002 *		SA2 OPERAND*1
	5140000003 *		SA3 OPERAND*2
L 20	37621		SA4 OPERAND*3
	37734		IX6 X2-X1
	5160000004 C		IX7 X3-X4
L 21	5170000005 C		SA6 ANSWER*4
	5100000015 *		SA7 ANSWER*5
L 22	7100000022		SA0 ADD
			SA0 LBLOCK1+LBLOCK2
L 23	0110000010		IFCP6 2
	0100000000 X		RE LBLOCK3
			RJ ERRMSG2

10-18

L	24	0100000015 *	END2BLK	IFCP7	1
L	15		RJ	ADD	
L	15	00000000000000000000	MULT	LOC	ADD
L	16	5110000000 *		DATA	0
		5120000001 *		SA1	OPERAND*0
L	17	5130000002 *		SA2	OPERAND*1
		5140000003 *		SA3	OPERAND*2
L	20	42612		SA4	OPERAND*3
		42734		IX6	X1-X2
		5160000006 C		IX7	X3-X4
L	21	5170000007 C		SA6	ANSWER*6
		5100000015 *		SA7	ANSWER*7
L	22	7100000032		SA0	ADD
				SA0	ADD
L	23	0110000012		SA0	LBLOCK1+LBLOCK2+LBLOCK3
		0100000000 X		IFCP6	2
				RE	LBLOCK4
L	24	0100000015 *	END3BLK	RJ	ERRMSG2
L	15			IFCP7	1
L	15	00000000000000000000		RJ	ADD
L	16	5110000000 *	DIVIDE	LOC	ADD
		5120000001 *		DATA	0
L	17	5130000002 *		SA1	OPERAND*0
		5140000003 *		SA2	OPERAND*1
L	20	27101		SA3	OPERAND*2
		27202		SA4	OPERAND*3
		24101		PX1	X1
		24202		PX2	X2
L	21	44621		NX1	X1
		27303		NX2	X2
		27404		FX6	X2/X1
		24303		PX3	X3
L	22	24404		PX4	X4
		44743		NX3	X3
		5160000010 C		NX4	X4
L	23	5170000011 C		FX7	X4/X3
		26666		SA6	ANSWER*8
		26777		SA7	ANSWER*9
L	24	22666		UX6	X6+H6
		22777		UX7	X7+H7
		5160000012 C		LX6	X6+H6
L	25	5170000013 C		LX7	X7+H7
L	26	0400000010 *	END4BLK	SA6	ANSWER*10
			EO	SA7	ANSWER*11
					START
		12	LBLOCK1	EOU	END1BLK-ADD*1
		10	LBLOCK2	EOU	END2BLK-SUBT*1
		10	LBLOCK3	EOU	END3BLK-MULT*1
		12	LBLOCK4	EOU	END4BLK-DIVIDE*1
		44	LBLOCKT	EOU	LBLOCK1+LBLOCK2+LBLOCK3+LBLOCK4
		12	MAXBLK	MAX	LBLOCK1+LBLOCK2+LBLOCK3+LBLOCK4
		10	MINBLK	MIN	LBLOCK1+LBLOCK2+LBLOCK3+LBLOCK4
				END	

61

472008 CM STORAGE USED
MODEL 74 ASSEMBLY

114 STATEMENTS
0.357 SECONDS

20 SYMBOLS
99 REFERENCES

10-19

LOC7600
SYMBOLIC REFERENCE TABLE.

COMPASS 3.4-444.

02/22/77 06.31.12.

PAGE 4

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		
CIVIDE	15	PROGRAM	3/21 L	3/48						
ENC1BLK	26	PROGRAM	2/42 L	3/45						
ENC2BLK	24	PROGRAM	3/02 L	3/46						
ENC3BLK	24	PROGRAM	3/19 L	3/47						
ENC4BLK	26	PROGRAM	3/44 L	3/48						
EPHMSG1	0	EXTERNAL	2/04 X	2/14						
EPHMSG2	0	EXTERNAL	2/04 X	2/16	2/39	2/57	3/17			
LBLOCK1	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				
PAFLK	12		3/50 D							
PIARLK	10		3/51 D							
MULT	15	PROGRAM	3/04 L	3/47						
CREFERAND	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 NGS/BE 1.2 R443A/R3B 02/02/77
06.31.06.CON003E FROM /LE
06.31.06.IP 00000832 WORDS - FILE INPUT , DC 04
06.31.06.CON. 001A.0904.71GT100A.MILLER
06.31.07.FIN.R=0.
06.31.12. .647 CP SECONDS COMPIATION TIME
06.31.12.PAP.PART.
06.31.12.EXIT.
06.31.12.CP 00001408 WORDS - FILE OUTPLY , DC 40
06.31.12.PS 3584 WORDS (14336 MAX USED)
06.31.12.CPA .717 SEC. .717 ADJ.
06.31.12.CPB .001 SEC. .001 ADJ.
06.31.12.10 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

HISTORY

BLANK COMMON

ORIGINALLY BLANK COMMON WAS DESIGNED AS THE ONE AREA FOR COMMUNICATION BETWEEN SUBPROGRAMS....

LOAD LOCATION

BLANK COMMON IS TYPICALLY LOCATED AT THE END OF THE LOAD AND HENCE

PRESETTING

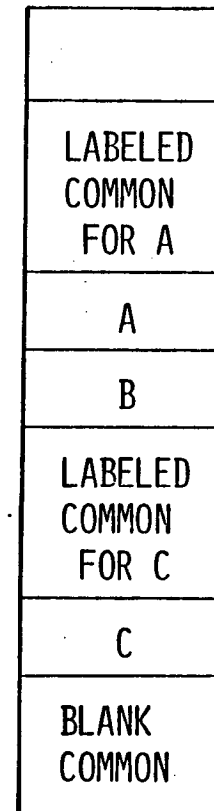
NO PRESETTING, I.E., ALL INFORMATION MUST BE PUT INTO BLANK COMMON AT EXECUTION TIME.

LABELED COMMON

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 LABELS TO IDENTIFY WHICH BLOCK WAS TO BE USED BY WHICH SUBROUTINE.

LABELED COMMON BLOCKS COME AT THE BEGINNING OF THE LOAD AND THUS

CAN BE PRESET, I.E., THE PROGRAMMER CAN STORE INSTRUCTIONS AND DATA IN LABELED COMMON BLOCKS AND THESE AREAS WILL BE SET UP AT LOAD TIME.



RA

RA+101B

RA+FL

LINKAGE
STORAGE ALLOCATION.

COMPASS 3.5-470.

07/14/78 12.43.36.

PAGE 1

ADDRESS LENGTH

0 77
77

BINARY CONTROL CARDS.

IDENT LINKAGE
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, RUF, ERRMSG1	LIST	COMMENT	THIS EXAMPLE SHOWS THE LINKAGE BETWEEN PROGRAM BLOCKS
		EXT	SURI					
0	46000	START	NO					
1	7160000010	•	SK6	10R	DATA			
	5160000024 •		SA6	RUF	STORE IT			
	36 •	LTAG	L11	2,4,6	LITERAL			
2	5110000037 •		SA1	LTAG+1	GET 2ND LITERAL			
	10711		RK7	X1	MOVE TO X7			
3	5170000025 •		SA7	RUF+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	SURI	GO TO SUR1			
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		SK6	3REND+4+1	ENDRUN	.1		
	20650		LX6	400	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	420	MESSAGE	.1		
	15116		RX1	-X6*X1	MESSAGE	.1		
	0	•	•	SCPERC SET 0	MESSAGE	.1		
				IFC NE, RECALL .1	MESSAGE	.1		
	200000	•	•	SCPERC SET 200000	MESSAGE	.1		
				IF -REG,,7	MESSAGE	.1		
	1	•	•	SCPEOP SET 1	MESSAGE	.1		
				IF DEF,,1	MESSAGE	.1		
	0	•	•	SCPEOP SET	MESSAGE	.1		
				IFC EO,,BLLOCALS,1	MESSAGE	.1		
		•	•	SCPEOP SET 1	MESSAGE	.1		
			R=	X6,,+SCPERC++SCPEOP	MESSAGE	.1		
			SK6	++SCPERC++SCPEOP	MESSAGE	.1		
			SKIP	7	MESSAGE	.1		
			IFC	EO,,SX1S,1	MESSAGE	.1		
			SKIP	1	MESSAGE	.1		
			IFC	EO,,SX6S,1	MESSAGE	.1		
		4	ERR	OPTION IN X1 ON X6 IGNORED	MESSAGE	.1		
			R=	X6,,+SCPERC	MESSAGE	.1		
			SKIP	1	MESSAGE	.1		
			R=	X6,,+SCPERC	MESSAGE	.1		
			RJ	=XMSG=	MESSAGE	.1		
	0100000000 X		ENDM			MESSAGE .1		

```

DATA02  USF  DATA  1.1.0,2.2.0,3.3.0,4.4.0,5.5.0

```

DATA03 CON 100,200,300,400,500,600,700,800,900,1000

DATA04 CON 1RA,1RB,1RC,1RD,1RE,1RF,1RG,1RH,1RI,1RJ

0	12
12	12
0	12
21	55055703572357552722
22	11240555052222172255
23	17030325222205040000
24	12

```

      USELCM /DATA/
DATA10  BSS      10
DATA01  BSS      10
      USELCM //
DATA6   BSS      10
      USE       0
MESSAGE DIS      ,/ E.C.S. WRITE ERROR OCCURRED/

BUF      BSSZ     10          BUFFER FOR PROGRAM

```

DEFAULT SYMBOLS DEFINED BY COMPASS.

0 X
0 X

SVS-
HSG-

CONTENT OF LITERALS BLOCK.

```

36 000000000000000000000002
37 000000000000000000000004
40 000000000000000000000006

```

**A
N
D**

77

END START

LINKAGE
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

07/14/78 12.43.36.

PAGE 4

RUF	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	3/34 L				
FRMSG1	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					

ADDRESS LENGTH

0 31
31

BINARY CONTROL CARDS.

IDENT SUB1
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

0	00000000000000000000	SUB1	IDENT SUB1	
1	7160000006		EXT BUF	
	5160000002 X		ENTRY SUB1	
2	5100000017		DATA 0	RETURN ADDRESS STORED HERE.
	7100000000 C		SA6 6	DATA
3	0110000012		SA6 BUF+2	STORE IT
	0100000010		SA0 BUFFER	S.C.M. STARTING ADDRESS
4	7100000012 C		SA0 DATA9	L.C.M. STARTING ADDRESS
5	0120000012		RE 10	TRANSFER 10 WORDS FROM LCM TO SCM
	0100000000 X		RJ ERRMSG2	
6	0100000000 X		SA0 DATA8	L.C.M. STARTING ADDRESS
7	0400000000		WE 10	TRANSFER 10 WORDS FROM S.C.M. TO L.C.M. BLANK COMM
	0100000000 X		RJ XERRMSG1	GIVE ERROR MESSAGE
10	00000000000000000000	ERRMSG2	RJ XSUB2	GO TO SUB2
11	7110000014		EQ SUB1	EXIT FROM THIS SUBROUTINE
13	0400000010		DATA 0	
14	55055703572357552205	READMSG	MESSAGE READMSG, RECALL	GIVE ERROR MESSAGE
17	12	BUFFER	EQ ERRMSG2	RETURN TO CALLER
0	5	BF4	DIS // E.C.S. READ ERROR OCCURRED/	
0	5	DATA	USE //	USE BLANK COMMON
5	17	BUFF2	USE /DATA/	USE LABELED COMMON
0	12	DATA9	USE LCM /DATA/	
0	12	DATA7	USE LCM //	
12	24	DATA8	USE LCM //	
31			END	

11008 LCM 476008 SCM STORAGE USED
MODFL 174 ASSEMBLY

49 STATEMENTS
0.061 SECONDS

16 SYMBOLS
21 REFERENCES

SUBRT2
STORAGE ALLOCATION.

COMPASS 3.5-470.

07/14/78 12.43.36.

PAGE 1

ADDRESS LENGTH

0 3
3

BINARY CONTROL CARDS.

IDENT SUBRT2
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

6-11

SUBRT2

COMPASS 3.5-470.

07/14/78 12.43.36.

PAGE 2

ADDRESS	LENGTH	IDENT	SUBRT2	ENTRY	SUB2	USE	COMMENT
0	12	BUFF4	0	USE	//	USE BLANK COMMON	
0	00000000000000000000	SUB2	0	USE	10	GO BACK TO ZERO BLOCK	
1	7160000004	5160000003 X	0	DATA	0	DATA	
2	0400000000	0	4	SA6	*XRUF*3	STORE DATA	
			ED	SUB2		EXIT FROM THIS SUBROUTINE	
			USE	/DATA/		USE LABELED COMMON	
0	5	DUMMY	5	USE	5	DUMMY OVER	
5	5	BUFF2	5	USE	5	USED IN SUBRT2	
12	12	BUFF5	10	USE	10	USE LABELED COMMON	
0	12	BUFF3	10	USE	10	USE LABELED COMMON	
3				END			

1100R LCM 46500R SCH STORAGE USED
MODEL 174 ASSEMBLY

16 STATEMENTS
0.017 SECONDS

7 SYMBOLS
9 REFERENCES

FMA OF THE LOAD 111
LWA-1 OF THE LOAD 354

ECS FMA 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

11-10

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 46000	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 20430 12161 73610	20173 03310 00331 20173	03310 20276 01000 00331	

PAGE NO. 1

[illegible]

11-12

```

MFF NR2- CYR175-SN1 5201/R6B 06/07/70
12.43.34.00N00RU FROM /SH
12.43.34.1P 00000640 WORDS - FILE INPUT * DC 04
12.43.34.NON,EC1. PSD,0278,72CT011A,MILLER
12.43.35.COMPASS.
12.43.36. ASSEMBLY COMPLETE. 47600B SCH 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. .370 ADJ.
12.43.37.10 .785 SEC. .785 ADJ.
12.43.37.CH 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 07/14/70
12.43.37.FJ END OF JOB. SH

```

TRUNCOM
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.20.41.

PAGE 1

ADDRESS LENGTH

0 2
2

BINARY CONTROL CARDS.

IDENT TRUNCOM
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/

0 12 BUF
0 46000 TRUNCOM
2 7160247021

USE /BLOCK/ USE LABELLED COMMON
RSSZ 10 RESERVE 10 WORDS OF ZEROS
USE 0
NO
ENDRUN
END TRUNCOM

50400B SCM STORAGE USED
MODEL 174 ASSEMBLY

14 STATEMENTS
0.025 SECONDS

1 SYMBOLS
4 REFERENCES

TRUNCOM
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 16.20.41.

PAGE 3

BUF	0	BLOCK	2/07 L
SYS*	0	EXTERNAL*	2/11
TRUNCOM	0	PROGRAM*	2/02 E 2/03 L

11-13

TRUNCIT
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 16.20.42.

PAGE 1

ADDRESS LENGTH

0 0
0

HINARY CONTROL CARDS.

IDENT TRUNCIT
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
BSS 20
END

0
0

24 BUF

47300R SCH STORAGE USED
MODEL 174 ASSEMBLY

9 STATEMENTS
0.010 SECONDS

1 SYMBOLS
2 REFERENCES

11-14

TRUNCIT
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 16.20.42.

PAGE 3

BUF 0 BLOCK 2/02 F 2/07 L

FMA OF THE LOAD 111
LMA-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.PM	135	40	SL-SYSLIB	05/16/78	COMPASS	3.5	470		

PROCESS SYSTEM REQUEST.

DMPX.

P	000000	A0	000100	R0	000000														
RA	220200	A1	000001	R1	000001	C(A1)=	0000	0000	0000	0000	0000	C(R1)=	0000	0000	0000	0000	0000		
FL	000100	A2	000060	R2	000002	C(A2)=	1505	1520	0000	0000	0061	C(R2)=	0000	0000	0000	0000	0000		
FM	700700	A3	000057	R3	012733	C(A3)=	0000	0000	0000	0000	0000	C(R3)=							
RE	000045	A4	000001	R4	000301	C(A4)=	0000	0000	0000	0000	0000	C(R4)=							
FE	000000	A5	000123	R5	000123	C(A5)=	0000	0000	0000	0000	0000	C(R5)=	0000	0000	0000	0000	0000		
MA	000600	A6	000001	R6	000300	C(A6)=	0000	0000	0000	0000	0000	C(R6)=							
		A7	000001	R7	027756	C(A7)=	0000	0000	0000	0000	0000	C(R7)=							
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	00000										
00054	56110	03110	00054	54710	51100	00001	03110	00055	64550	02550	00000	46000	00000	00000	00000	00000	00000	00000	00000
00060	15051	52000	00000	00061	00000	00300	00000	00001	07040	00060	51600	00001	04000	00063	00000	00021	00000	00000	00000
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	00006	40000	00000	00175	00000	00000	00000	00000	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															

11-15

MFF NR2- CYR175-SNI 4LR7/H6B 05/15/78
 16.20.40.DON00MJ FROM /SM
 16.20.40.1P 00000256 WORDS - FILE INPUT , DC 04
 16.20.40.DON. PSD.027B.72CT011A.MILLER
 16.20.41.COMPASS.
 16.20.42. ASSEMBLY COMPLETE. 50400B SCH 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.ID .645 SEC. .645 ADJ.
 16.20.43.CH 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, SH

LOADED1
STORAGE ALLOCATION.

COMPASS 1.5-470.

06/27/78 17.02.43.

PAGE 1

ADDRESS LENGTH

BINARY CONTROL CARDS.

0
4

4

IDENT
END

LOADED1
START

BLOCKS TYPE ADDRESS LENGTH

PROGRAM* LOCAL 0 4
LARCOMM COMMON 0 2

ENTRY POINTS.

START 0*

EXTERNAL SYMBOLS.

SYS*

11-17

LOADED1

COMPASS 3.5-470.

06/27/78 17.02.43.

PAGE 2

IDENT LOADED1
ENTRY START

.....
*
* IMPROPER USE OF LABELED COMMON *
*
.....

0 00000000000000000001
1 00000000000000000002
2
2 7160247021
20650
3 0100000000 X

START DATA 1.2

ENDRUN

SX6 JREND*4.1

LX6 40D

RJ *XSYS*

ENDM

ENDRUN .1
ENDRUN .1
ENDRUN .1
ENDRUN .1

0 777777777777777777
1 777777777777777775

USE /LARCOMM/
DATA -1.-2

USE 0

DEFAULT SYMBOLS DEFINED BY COMPASS.

0 X

SYS*

4

END START

50400B SCH STORAGE USED
MODEL 174 ASSEMBLY

17 STATEMENTS
0.035 SECONDS

2 SYMBOLS
3 REFERENCES

LOADED?
STORAGE ALLOCATION.

COMPASS 1.5-470.

06/27/74 17.02.43.

PAGE 1

ADDRESS LENGTH

0 2
2

BINARY CONTROL CARDS.

IDENT LOADED?
END

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM	LOCAL	0	2
LARCOMM	COMMON	0	3

LOADED?

COMPASS 1.5-470.

06/27/74 17.02.43.

PAGE 2

0 0000010000000000000003
1 0000000000000000000004

0
1 1111111111111111111114
2 1111111111111111111113
2

IDENT LOADED?
DATA 1.4
USE /LARCOMM/
HSS 1
DATA -3.-4
END

473008 SCH STORAGE USED
MODEL 174 ASSEMBLY

6 STATEMENTS
0.011 SECONDS

0 SYMBOLS
0 REFERENCES

81-11

LOAD MAP - LOADED1

CYCLE LOADER 1.4-470

06/27/78 17.02.45.

PAGE 1

PWA OF THE LOAD 111
LWA+1 OF THE LOAD 161

TRANSFER ADDRESS -- START 111

PROGRAM ENTRY POINTS -- LOADED1 111

***** ERROR SUMMARY

NE4101///COMMON BLOCK REDEFINITION - LABCOMM
LAST PROGRAM READ - LOADED2
LAST FILE ACCESSFD- LGO

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSR	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

13200P CM STORAGE USED

1 TABLE MOVE

11-19

DUMP RELATIVE

DMP(111+117)

00111 77777 77777 77777 77776
00114 00000 00000 00000 00002

77777 77777 77777 77776
71602 47021 20450 46000

77777 77777 77777 77773
01000 00123 61000 46000

00000 00000 00000 00000

DMPX.

P 000000 A0 000200 H0 000000
 RA 313100 A1 000000 H1 000001
 FL 000200 A2 000060 H2 000002
 EM 700700 A3 000057 H3 012713
 RE 000051 A4 000001 H4 000201
 FE 000000 A5 000113 H5 000113
 MA 001000 A6 000001 H6 000200
 A7 000001 H7 027756
 H0 0000 0000 0000 0000 0000
 H1 0000 0000 0000 0000 0000
 H2 1505 1520 0000 0000 0061
 H3 0000 0000 0000 0000 0000
 H4 0000 0000 0000 0000 0000
 H5 6000 0000 0004 0040 0000
 H6 1505 1520 0000 0000 0061
 H7 0000 0000 0000 0001 5023

C1A11= 0000 0000 0000 0000 0000
 C1A21= 1505 1520 0000 0000 0061
 C1A11= 0000 0000 0000 0000 0000
 C1A41= 0000 0000 0000 0000 0000
 C1A51= 7777 7777 7777 7777 7777
 C1A61= 0000 0000 0000 0000 0000
 C1A71= 0000 0000 0000 0000 0000

C(H1)= 0000 0000 0000 0000 0000
 C(H2)= 0000 0000 0000 0000 0000
 C(H3)=
 C(H4)=
 C(H5)= 7777 7777 7777 7777 7777
 C(H6)=
 C(H7)=

00000 00000 00114 00000 00000
 00054 56110 03110 00054 54710
 00060 15051 52000 00000 00061
 00064 14071 70000 00000 00000
 00070 05100 50125 24055 12324
 00100 54000 00000 01000 00001

00000 00000 00000 00000
 51100 00001 03110 00055
 00000 00200 00000 00001
 00000 00000 00000 00161
 01222 45255 00000 00000

64550 02550 00000 46000
 07040 00060 51600 00001
 40000 00000 02000 00111
 00000 00000 00000 00000

00000 00000 00000 00000
 04000 00063 00000 00021
 00000 00000 40000 00000

00100 54000 00000 01000 00001
 00104 00000 00000 00000 00161
 00110 14170 10405 04340 00113
 00114 00000 00000 00000 00002
 00120 00000 00000 00000 00004
 00124 51100 00001 03110 00124
 00130 71100 00130 20160 46000
 00134 20652 01000 00123 46000
 00140 71602 20314 04000 00134
 00144 53160 20173 03310 00142
 00150 71603 24616 12661 20651
 00154 20123 03210 00150 20151
 00160 00000 00000 00000 00000
 00164 60000 00000 04004 00164
 00170 60000 00000 04004 00170
 00174 60000 00000 04004 00174
 00200 00000 00000 00000 00000

00005 00000 00000 00161
 00000 00000 00000 00000
 77777 77777 77777 77776
 71602 47021 20650 46000
 54110 20123 03310 00122
 54610 04000 00122 46000
 13661 13161 13661 46000
 51100 00001 03110 00135
 20150 36661 01000 00123
 03010 00142 51100 00001
 01000 00123 61000 46000
 13116 20636 51600 00160
 60000 00000 04004 00161
 60000 00000 04004 00165
 60000 00000 04004 00171
 60000 00000 04004 00175

00000 00000 00000 00000
 77777 77777 77777 77774
 01000 00123 61000 46000
 04000 00126 61000 46000
 51100 00066 03310 00130
 51600 00122 10611 46000
 04004 00136 61000 46000
 04004 00142 61000 46000
 03110 00144 71100 00001
 04004 00152 61000 46000
 74660 36116 20123 46000
 60000 00001
 60000 00001
 60000 00001
 60000 00001

77777 77777 77777 77773
 00000 00000 00000 00003
 04004 00123 61000 46000
 51100 00121 04000 00131
 51100 00001 01000 00121
 51100 00001 03110 00136
 71602 20314 20652 36662
 04000 00141 61000 46000
 73660 20630 12161 73610
 04000 00150 61000 46000

MFF NB2- CYR175-SN1 4LB7/R6H 05/15/78
 17.02.40.DON00NZ FROM /SM
 17.02.40.1P 00000320 WORDS - FILE INPUT , DC 04
 17.02.40.DON. PSD.0278.72CT011A.MILLER
 17.02.42.COMPASS(LO=1888)
 17.02.43. ASSEMBLY COMPLETE. 504008 SCH USED.
 17.02.43. 0.107 CPU SECONDS ASSEMBLY TIME.
 17.02.43.LOADILGO)
 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.
 17.02.46.10 .675 SEC. .675 ADJ.
 17.02.46.CH 11.048 MWS. .674 ADJ.
 17.02.46.SS 1.515
 17.02.46.PP 4.588 SEC. DATE 06/27/78
 17.02.46.FI END OF JOB. CH

PARAMETERS FORTRAN EXTENDED

THE CALL:

CALL SUB (I , J , K , ANS)

ACTUAL PARAMETERS

THE CODE GENERATED:

. AN AP List is Created:

(ADDRESSES
OF
PARAMETERS)

[AP1

ADDRESS of I
ADDRESS of J
ADDRESS of K
ADDRESS of ANS
0

TO INDICATE
END OF LIST

. 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.-H

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
1 53210 5011000001 53310
2 5011000001 53410 36723
3 36774 5011000001 53710
4 040000C000 +
5

SUB DATA 0 ADDRESS PARAMETER LIST IS IN A1 ON ENTRY
SA2 X1 1ST PARAMETER IN X2 (I)
SA1 A1+1 2ND PARAMETER ADDRESS IN X1
SA3 X1 2ND PARAMETER IN X3 (J)
SA1 A1+1 3RD PARAMETER ADDRESS IN X1 (J)
SA4 X1 3RD PARAMETER IN X4 (K)
IX7 X2+X3 COMPUTE SUM
IX7 X7+X4 COMPUTE SUM
SA1 A1+1 4TH PARAMETER ADDRESS IN X1
SA7 X1 STORE ANSWER (ANSWER)
EQ SUB RETURN TO CALLER
END

50600B CM STORAGE USED
MODEL 74 ASSEMBLY

23 STATEMENTS
0.112 SECONDS

1 SYMBOLS
0 REFERENCES

MFS NHI- CYR74-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.
10.16.43.10 1.218 SEC. 1.218 ADJ.
10.16.43.CH 36.786 KWS. 2.245 ADJ.
10.16.43.SS 4.353
10.16.43.PP 7.116 SEC.
10.16.43.EJ END OF JOB, OH DATE 01/05/79

11-23

```

1      PROGRAM PARAMS(OUTPUT)
      INTEGER ANSWER
      ANSWER=0
      I=1
5      J=2
      K=3
      CALL SUB(I,J,K,ANSWER)
      SUB IS ENTERED WITH ADDRESS OF PARAMETER ADDRESSES IN A1.
10     PRINT 101
      PRINT 100,I,J,K,ANSWER
      100 FORMAT(4I10)
      101 FORMAT(1H1)
      END

```

```

      IDENT  PARAMS
      USEBLK
      LOSET  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
000001 START. 00000000000000000051
000002 START. 00000000000000000000
000012 START. 00000000000000000000
000013 START. 00000000000000000000
000014 START. 00000000000000000000
000015 START. 1+00000000000000000000
000016 START. 00000000000000000000
000017 START. 00000000000000000000
000020 START. 00000000000000000000
000021 START. 00000000000000000000

```

```

002054 START. 2001000000000000002057
002055 START. 000000000000000000141520
002056 START. 0000000000000000000000

```

```

002057 START. 17252420252400000000
002060 START. 0000000000000000000000
002061 START. 20012201152355002062

```

FILCRM OUTPUT#.20038.0B

```

LIBLNK. BSS 0B
LIBLNK 18.141520B

```

```

FILES. BSS 0B
FLINK  OUTPUT#
DATA 0
TRACE PARAMS,PARAMS
USE CODE.

```

002062 CODE. 5110002054 START.
0100000000 <EXT>

002103 DATA. 00000000000000000000 START.
002103 DATA. 00000000000000000000 DATA.
002104 DATA. 00000000000000000000 DATA.
002105 DATA. 00000000000000000000 DATA.
002106 DATA. 00000000000000000000 DATA.
002106 DATA. 00000000000000000000 START.
002107 DATA. 00000000000000000000 DATA.
002110 DATA. 00020000000100002122 DATA.
002111 DATA. 00020000000100002123 DATA.
002112 DATA. 00020000000100002124 DATA.
002113 DATA. 00020000000100002121 DATA.
002114 DATA. 00000000000000000000 DATA.
002115 DATA. 55343333555500000000 DATA.
002116 DATA. 51371134335255555555 DATA.
002117 DATA. 55343334555500000000 DATA.
002120 DATA. 51341034525555555555 DATA.

PENTRY PARAMS...0
SA1 LI0LNK.
RJ Q2NTRY.
USE DATA.
USE DATA..
USE DATA.
J101 BSS 08
IOM OUTPUT#
IOM .101
EIO 08
J102 BSS 08
IOM OUTPUT#
IOM .100
IOM 1,08,28,18
IOM J,08,28,18
IOM K,08,28,18
IOM ANSWER,08,28,18
EIO 08
FMT .100
DIS 1,(4110)
FMT .101
DIS 1,(111)
EXT END.
EXT OUTCI.
EXT SUB
EXT Q2NTRY.

ANSWER BSS 18
I BSS 18
J BSS 18
K BSS 18

USE CODE.

LINE 3

002063 CODE. 43700
7160000001
002064 CODE. 5170002121 DATA.
7170000002
002065 CODE. 5160002122 DATA.
7160000003
002066 CODE. 5170002123 DATA.
5160002124 DATA.
002067 CODE. 5110002076 CODE.
002070 CODE. 0100000000 <EXT>
0007002061
002071 CODE. 5110002103 DATA.
002072 CODE. 0100000000 <EXT>
0011002061
002073 CODE. 5110002106 DATA.
002074 CODE. 0100000000 <EXT>
0012002061
002075 CODE. 5110002061 START.
0400000000 <EXT>

*
MX7 08
SX6 18
SA7 ANSWER
SX7 28
SA6 I
SX6 38
SA7 J
SA6 K
SA1 [AP1
RJT SUB,7B
SA1 J101
RJT OUTCI.,11B
SA1 J102
RJT OUTCI.,12B
SA1 TRACE.
EQ END.
BSS 08
APL I
APL J
APL K
APL ANSWER
APL

002076 CODE. 00000000000000000000 DATA.
002076 CODE. 00000000000000000000 DATA.
002077 CODE. 00000000000000000000 DATA.
002100 CODE. 00000000000000000000 DATA.
002101 CODE. 00000000000000000000 DATA.
002102 CODE. 00000000000000000000 DATA.

11-25

FMA 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	L60	01/05/79	COMPASS 3.6 476		
PARAMS	116	2125	L60	01/05/79	FIN 4.7 485	666X 1	PROGRAM OPT=1
/STP.END/	2243	1					
/FCL.C./	2244	26					
/OOL.O./	2272	101					
OZNTY=	2373	0	SL-FORTRAN	11/16/78	COMPASS 3.6 485		
/FCL=ENT/	2373	40					FCL INITIALIZATION ROUTINE.
COMIO=	2433	33	SL-FORTRAN	11/16/78	COMPASS 3.6 485		
FCL=FDL	2466	40	SL-FORTRAN	11/16/78	COMPASS 3.6 485		COMMON CODED I/O ROUTINES AND CONSTANTS.
FEIFST=	2526	3	SL-FORTRAN	11/16/78	COMPASS 3.6 485		FCL CAPSULE LOADING
FLTOUT=	2531	311	SL-FORTRAN	11/16/78	COMPASS 3.6 485		CONVERTED DATA STORAGE
FOKSYS=	3042	301	SL-FORTRAN	11/16/78	COMPASS 3.6 485		COMMON FLOATING OUTPUT CODE
OUTCOM=	3343	154	SL-FORTRAN	11/16/78	COMPASS 3.6 485		FORTAN OBJECT LIBRARY UTILITIES.
SYS=ID=	3517	1	SL-FORTRAN	11/16/78	COMPASS 3.6 485		COMMON OUTPUT CODE
FECHSK=	3520	41	SL-FORTRAN	11/16/78	COMPASS 3.6 485		LINK BETWEEN SYS=ID AND INITIALIZATION CODE.
FMTAP=	3561	357	SL-FORTRAN	11/16/78	COMPASS 3.6 485		INITIALIZE CONSTANTS.
FORUTL=	4140	46	SL-FORTRAN	11/16/78	COMPASS 3.6 485		CRACK APLIST AND FORMAT FOR KODER/KRAKER.
GETFIT=	4206	57	SL-FORTRAN	11/16/78	COMPASS 3.6 485		FCL MISC. UTILITIES.
KODER=	4265	451	SL-FORTRAN	11/16/78	COMPASS 3.6 485		LOCATE AN FIT GIVEN A FILE NAME.
OUTC=	4736	150	SL-FORTRAN	11/16/78	COMPASS 3.6 485		OUTPUT FORMAT INTERPRETER.
/FDL.COM/	5106	14					FORMATTED WRITE FORTRAN RECORD.
FDL.RES	5122	211	SL-SYSLIB	11/02/78	COMPASS 3.6 485		
FDL.MHI	5333	222	SL-SYSLIB	11/02/78	COMPASS 3.6 485		FAST DYNAMIC LOADER RESIDENT.
CPU.SYS	5555	40	SL-SYSLIB	11/15/78	COMPASS 3.6 476		FDL MEMORY MANAGER INTERFACE.
CMF.ALF	5615	160	SL-SYSLIB	11/15/78	COMPASS 3.6 485		PROCESS SYSTEM REQUEST.
CMF.CSF	5775	6	SL-SYSLIB	11/15/78	COMPASS 3.6 485		CMH VI.1 - ALLOCATE FIXED.
CMH.FFA	6003	14	SL-SYSLIB	11/15/78	COMPASS 3.6 485		CMH VI.1 - CHANGE SPECS FIXED.
CMF.FHF	6017	36	SL-SYSLIB	11/15/78	COMPASS 3.6 485		CMH VI.1 - FIXED FREE ALGORITHM.
CMH.H	6055	214	SL-SYSLIB	11/15/78	COMPASS 3.6 485		CMH VI.1 - FREE FIXED.
CMF.SLF	6271	22	SL-SYSLIB	11/15/78	COMPASS 3.6 485		CMH VI.1 - RESIDENT SUBROUTINES.
CTL.SRM	6313	433	SL-SYSLIB	11/16/78	COMPASS 3.6 485		CMH VI.1 - SHRINK AT LWA FIXED.
ERR.SRM	6746	25	SL-SYSLIB	11/16/78	COMPASS 3.6 485		CRM CONTROLLING ROUTINE.
LIST.SRM	6773	67	SL-SYSLIB	11/16/78	COMPASS 3.6 485		CRM ERROR PROCESSOR ENTRY.
							CRM - ALLOCATE SPACE FOR LIST OF FILES

.336 CP SECONDS

235008 CH STORAGE USED

12 TABLE MOVES

1

2

3

6

```

1      PROGRAM DUMMY(OUTPUT)
      .....
      *
      *
5     *   A DUMMY FORTRAN PROGRAM IS SET UP TO DECLARE THE FILES AND INITIALIZE
      *   AND DUMP THE BUFFERS CORRECTLY.  THIS IS DONE BY QBENTRY.
      *
      *
      .....
10     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		5X6	0
	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
5				3
			END	

506008 CM STORAGE USED
MODEL 74 ASSEMBLY

21 STATEMENTS
0.094 SECONDS

3 SYMBOLS
0 REFERENCES

SUBROUTINE OUTSUB 73/74 OPT=1

FTN 4.7-485

01/05/79 10.13.38

PAGE 1

```

1          SUBROUTINE OUTSUB
          COMMON /LAB/ IBUF(3)
          PRINT 101
          PRINT 100,(IBUF(I),I=1,3)
5          100 FORMAT(3O24)
          101 FORMAT(1H1)
          END

```

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							
/QB.IQ./	2260	101							
QZNTY=	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=FCL	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
SYSID=	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.
FURUTL=	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.
KUDER=	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.
/FOL.COM/	5074	14							
FOL.RES	5110	211	SL-SYSLIB	11/02/78	COMPASS	3.6	485		FAST DYNAMIC LOADER RESIDENT.
FOL.MMI	5321	222	SL-SYSLIB	11/02/78	COMPASS	3.6	485		FOL 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		CMH V1.1 - ALLOCATE FIXED.
CMF.CSF	5763	6	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CMH V1.1 - CHANGE SPECS FIXED.
CMH.FFA	5771	14	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CMH V1.1 - FIXED FREE ALGORITHM.
CMF.FHF	6005	36	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CMH V1.1 - FREE FIXED.
CMH.R	6043	214	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CMH V1.1 - RESIDENT SUBROUTINES.
CMF.SLF	6257	22	SL-SYSLIB	11/15/78	COMPASS	3.6	485		CMH V1.1 - SHRINK AT LWA FIXED.
CTL3RM	6301	433	SL-SYSLIB	11/16/78	COMPASS	3.6	485		CRM CONTROLLING ROUTINE.
ERR3RM	6734	25	SL-SYSLIB	11/16/78	COMPASS	3.6	485		CRM ERROR PROCESSOR ENTRY.
LIST3RM	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

00000000000000000000000010

00000000000000000000000010

00000000000000000000000010

MFS NBI- 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.T5. 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
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 09.11.51.

PAGE 1

ADDRESS LENGTH

0 5
5

BINARY CONTROL CARDS.

IDENT MAINPRG
END MAINPRG

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM	LOCAL	0	5
LAB	COMMON	0	3

ENTRY POINTS.

MAINPRG 0.

EXTERNAL SYMBOLS.

OUTSUR SYS=

MAINPRG

COMPASS 3.5-470.

06/27/78 09.11.51.

PAGE 2

IDENT MAINPRG
ENTRY MAINPRG
EXT OUTSUR

INCONRECT SETUP FOR A COMPASS MAIN PROGRAM AND FORTRAN SUBROUTINE

0 7160000010
1 5160000001 C
2 0100000000 X
3 7160247021

0 3
5

MAINPRG SAK A
SAK BUF
SAK BUF+1
SAK BUF+2
RJ OUTSUR
ENDRUN
USE /LAB/
HSSZ 7
END MAINPRG

50400B SCH STORAGE USED
MODEL 174 ASSEMBLY

22 STATEMENTS
0.042 SECONDS

4 SYMBOLS
9 REFERENCES

MAINPRG
SYMBOLIC REFERENCE TABLE.

COMPASS 3.5-470.

06/27/78 09.11.51.

PAGE 3

BUF	0	LAB	2/12 S	2/13 S	2/14 S	2/15 L
MAINPRG	0	PROGRAM	2/02 F	2/11 L		
OUTSUR	0	EXTERNAL	2/03 X	2/15		
SYS=	0	EXTERNAL	2/17			

11-30

1

SUBROUTINE OUTSUP

5

THE SUBROUTINE CAN NOT OUTPUT CORRECTLY BECAUSE THE FILE BUFFERS HAVE NOT
BEEN INITIALIZED. A FORTRAN MAIN PROGRAM IS REQUIRED.

10

COMMON /LAB/ IBUF(3)
PRINT 100,(IBUF(I),I=1,3)
100 FORMAT(3020)
END

MFF NH2- CYR175-SN1 4LB7/R68 05/15/78
09.11.47.DONORON FROM /SH
09.11.47.IP 00000320 WORDS - FILE INPUT , DC 04
09.11.47.DON. PSD.0278.72CT011A.MILLER
09.11.49.FTN,H=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.ERRORS
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.I0 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

11-31

LWA 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	PROCESSOR	VEN	LEVEL	HARDWARE	COMMENTS
/LAH/	111	3							
MAINPRG	114	5	LGO	06/27/78	COMPASS	1.5	470		
OUTSUB	121	14	LGO	06/27/78	FIN	4.7	470	767X 1	SUBROUTINEOPT=1
/OBLD./	135	17							
/FCL=ENI/	214	42							
/FCL.C./	276	25							
COMIO=	323	14	SL-FORTRAN	05/02/78	COMPASS	1.5	470		COMMON CODED I/O ROUTINES AND CONSTANTS.
FCL=FOL	337	40	SL-FORTRAN	05/02/78	COMPASS	1.5	470		FCL CAPSULE LOADING
ENIAP=	377	377	SL-FORTRAN	05/02/78	COMPASS	1.5	470		CHECK APLIST AND FORMAT FOR KODER/KRAKER.
FORUIL=	716	46	SL-FORTRAN	05/02/78	COMPASS	1.5	470		FCL MISC. UTILITIES.
GETFIT=	1044	61	SL-FORTRAN	05/02/78	COMPASS	1.5	470		LOCATE AN FIT GIVEN A FILE NAME.
KODER=	1125	461	SL-FORTRAN	05/02/78	COMPASS	1.5	470		OUTPUT FORMAT INTERPRETER.
OUTC=	1606	150	SL-FORTRAN	05/02/78	COMPASS	1.5	470		FORMATTED WRITE FORTRAN RECORD.
FECHSK=	1756	41	SL-FORTRAN	05/02/78	COMPASS	1.5	470		INITIALIZE CONSTANTS.
FLIOUT=	2017	315	SL-FORTRAN	05/02/78	COMPASS	1.5	470		COMMON FLOATING OUTPUT CODE
FORSYS=	2374	302	SL-FORTRAN	05/02/78	COMPASS	1.5	470		FORTRAN OBJECT LIBRARY UTILITIES.
OUTCOM=	2676	204	SL-FORTRAN	05/02/78	COMPASS	1.5	470		COMMON OUTPUT CODE
CMF.ALF	3042	160	SL-SYSLIB	05/02/78	COMPASS	1.5	470		CMH VI.1 - ALLOCATE FIXED.
CMF.CSF	3222	6	SL-SYSLIB	05/02/78	COMPASS	1.5	470		CMH VI.1 - CHANGE SPECS FIXED.
CMF.FFA	3230	14	SL-SYSLIB	05/02/78	COMPASS	1.5	470		CMH VI.1 - FIXED FREE ALGORITHM.
CMF.FHF	3244	36	SL-SYSLIB	05/02/78	COMPASS	1.5	470		CMH VI.1 - FREE FIXED.
CMH.P	3702	213	SL-SYSLIB	05/02/78	COMPASS	1.5	470		CMH VI.1 - RESIDENT SUBROUTINES.
CMF.SLF	3515	22	SL-SYSLIB	05/02/78	COMPASS	1.5	470		CMH VI.1 - SHRINK AT LWA FIXED.
CTLSPH	3537	601	SL-SYSLIB	05/02/78	COMPASS	1.5	470		CMH CONTROLLING ROUTINE.
FORBOM	4740	25	SL-SYSLIB	05/02/78	COMPASS	1.5	470		CMH ERROR PROCESSOR ENTRY.
LISTBOM	4745	66	SL-SYSLIB	05/02/78	COMPASS	1.5	470		CMH - ALLOCATE SPACE FOR LIST OF FILES
/FOL.COM/	4451	14							
FOL.MES	4467	211	SL-SYSLIB	05/16/78	COMPASS	1.5	470		FAST DYNAMIC LOADER RESIDENT.
FOL.MHI	4700	222	SL-SYSLIB	05/16/78	COMPASS	1.5	470		FOL MEMORY MANAGER INTERFACE.
SYS.PH	5122	40	SL-SYSLIB	05/16/78	COMPASS	1.5	470		PROCESS SYSTEM REQUEST.

DMPH.

P	000000	A0	000001	R0	000000
PA	201000	A1	000111	R1	000001
FL	007200	A2	000105	R2	000001
FM	700700	A3	000113	R3	000074
RE	000045	A4	005507	R4	000001
FE	000000	A5	005474	R5	000001
MA	000600	A6	001015	R6	000000
		A7	005205	P7	000014
X0	7777	7777	7777	7700	0000
X1	4135	5555	5555	5555	5555
X2	0000	0000	0000	0000	0002
X3	1725	7420	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	000A

CIA11	=	4135	5555	5555	5555	5555
CIA21	=	0000	0000	0000	0000	0002
CIA31	=	0000	0000	0000	0000	0000
CIA41	=	7777	7777	7777	7777	7700
CIA51	=	4000	0000	0000	0000	0000
CIA61	=	4135	5555	5555	5555	5555
CIA71	=	0000	0000	0000	0000	0076

CIA11	=	0000	0000	0000	0000	0000
CIA21	=	0000	0000	0000	0000	0000
CIA31	=	0000	0000	0000	0000	0000
CIA41	=	0000	0000	0000	0000	0000
CIA51	=	0000	0000	0000	0000	0000
CIA61	=	0001	4053	7300	0000	0000
CIA71	=	0000	0000	0000	0000	0000

00000	00014	05173	00000	00000
00054	54110	03110	00054	54710
00060	15051	52000	00000	00061
00064	14071	70000	00000	00000
00070	14071	75755	00000	00000
00100	54000	00000	01000	00001

00000	00000	00000	00000
51100	00001	03110	00055
00000	05200	00000	00001
00000	00000	00007	72615
00000	00000	00000	00000

44550	02550	00000	46000
07040	00060	51600	00001
40000	00000	02000	00111

00000	00000	00000	00000
04000	00063	00000	00021
00000	00000	40000	00000

05273	15557	71600	00151	11565
05274	03050	05277	61000	46000
05300	51600	05505	04000	05704
05304	51100	05205	51200	00307
05310	74600	51600	05205	46000
05314	21122	03110	05420	47060
05320	73334	03140	05317	46000
05324	01000	05453	00000	05241
05330	51500	05517	12736	54775
05334	10544	21560	62157	77677
05340	13143	15615	03060	05364
05344	03050	05374	51200	05522
05350	66240	54675	61000	46000
05354	20303	36303	72337	77775
05360	73120	04200	05364	46000
05364	51400	00310	03140	05444
05370	51440	05506	66223	22424
05374	53340	64440	63740	77347
05400	61400	00033	61000	46000
05404	71300	02355	20360	12636
05410	55061	72516	04554	65505
05414	71100	05406	43652	15116
05420	26212	74310	36033	20303
05424	40000	00000	00000	00000
05430	00000	00000	00000	00000
05434	51100	00313	52510	00015
05440	51500	00307	73150	43652
05444	13777	54740	04000	02476
05450	23312	30522	22550	05451
05454	51100	00313	61100	00001
05460	51300	05205	52510	00017
05464	51300	00301	11242	37423
05470	51200	00312	51300	02471

71200	03220	41000	46000
52530	00015	71600	05430
61500	00001	61300	00203
04000	05312	61000	46000
51504	05373	53220	55655
71204	05377	03060	05324
51500	05515	51400	00311
51300	00314	10655	63330
43752	43014	05100	05364
63740	04700	05364	54355
21522	10722	62157	70000
61404	05426	75364	76130
01000	05453	00000	05241
03110	05364	67335	20236
51400	05205	61300	00074
03240	05247	61000	46000
51400	00276	43260	11624
20303	15432	73544	20606
54760	54675	04000	05351
30050	32524	11171	65505
71602	00000	01000	05153
36303	46000	61000	46000
06241	65546	55172	52420
61500	00001	71100	05425
76600	43752	15667	11775
15116	71602	00000	46000
61500	00001	51300	02474
77777	77777	77777	77777
53010	64500	61600	05457
43746	15627	11775	36667
03140	05452	51200	00706
10622	22703	51600	05502

01000	02512	00000	05214
43752	15667	11775	36667
63210	53010	10711	46000
51200	00313	51100	05205
13225	61400	00012	77304
72330	00012	01000	01006
10755	51704	05404	54355
51200	02471	54675	21236
43536	51300	05521	13143
53140	11271	20252	12623
54765	05100	05347	54545
76520	20122	12615	36033
51100	05511	63210	21122
52427	77776	51500	05523
61500	00001	67405	63244
51100	00311	10611	46000
54665	03330	05351	37666
36665	67119	05100	05401
06241	65546	55161	75517
22221	72223	55000	00000
43601	51600	05424	46000
01000	05453	00000	05241
25245	50611	14055	52516
43652	15116	71602	00000
36667	54650	51100	05205
01000	05153	61000	46000
10633	51600	02471	46000
61500	00001	61000	46000
04000	04273	43000	00000
54650	62437	77654	46000
52510	00022	43736	15357
51700	05503	10644	20636

52510	00017	43746	15657
54650	46000	61000	46000
06320	05311	51000	00064
03020	02471	51200	00307
15420	22602	54225	54665
54660	46000	61000	46000
43044	20430	15640	46000
52427	77776	61000	46000
15615	03060	05364	21336
54665	20036	11504	63210
73450	66420	01000	01001
51600	05511	71204	05431
63310	51604	05434	56220
10655	54665	07030	05334
67223	66445	06200	05367
51600	01015	04004	05373
43271	20352	61100	00006
51200	05525	10722	20614
25242	02524	55061	11405
51500	05424	03150	05416
04000	05365	61000	46000
04000	05247	61000	46000
25230	10214	05000	00000
01000	05153	61000	46000
21122	03010	05417	46000
04000	05247	61000	46000
51600	00310	04000	02472
04004	05453	61000	46000
51100	00313	51200	05505
04400	05452	53210	47452
37432	03360	05452	46000
51600	05504	71100	00123

```

1      PROGRAM TSTFUNC(OUTPUT)
.....
5      *
      *      FORTRAN PROGRAM CALLING A COMPASS FUNCTION
      *
.....
10     I=2
      J=4
      IFUNC=ISUM(I,J)
      PRINT 100,IFUNC,I,J
100    FORMAT(1110)
      END

```

ISUM
STORAGE ALLOCATION.

COMPASS 3.5-470.

06/27/78 09.11.39.

PAGE 1

ADDRESS	LENGTH	MINARY 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 000000000000000000000000
1 53210
   5011000001
   53310
2 36623
   0400000000
3

```

```

ISUM DATA 0
SA2 X1 GET I
SA1 A1+1
SA3 X1 GET J
IX6 X2+X3 ADD TWO VALUES
EQ ISUM RETURN WITH ANSWER IN X6
END

```

473008 SCH STORAGE USED
MODEL 174 ASSEMBLY

10 STATEMENTS
0.026 SECONDS

1 SYMBOLS
3 REFERENCES

LWA OF THE LOAD 111
LWA-1 OF THE LOAD 1251

TRANSFER ADDRESS -- TSTFUNC 2173

PROGRAM ENTRY POINTS -- TSTFUNC 2173

PROGRAM AND BLOCK ASSIGNMENTS.

BLOCK	ADDRESS	LENGTH	FILE	DATE	PROCESSR	VER	LEVEL	HARDWARE	COMMENTS
TSTFUNC	111	2110	LGO	06/27/78	FIN	4.7	470	767X 1	PROGRAM OPT=1
ISUM	2221	3	LGO	06/27/78	COMPASS	3.5	470		
SYSID=	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							
/ON.10./	2253	77							
OZNTRY=	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=IDL	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,ALT	5133	160	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH V1.1 - ALLOCATE FIXED.
CMF,CSF	5313	6	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH V1.1 - CHANGE SPECS FIXED.
CMH,FFA	5321	14	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH V1.1 - FIXED FREE ALGORITHM.
CMF,FAF	5335	36	SL-SYSLIB	05/02/78	COMPASS	3.5	470		
CMH,H	5373	213	SL-SYSLIB	05/02/78	COMPASS	3.5	470		
CMF,SLF	5606	22	SL-SYSLIB	05/02/78	COMPASS	3.5	470		
CLBRN	5630	601	SL-SYSLIB	05/02/78	COMPASS	3.5	470		
ERBRN	6431	25	SL-SYSLIB	05/02/78	COMPASS	3.5	470		
LSTRN	6456	66	SL-SYSLIB	05/02/78	COMPASS	3.5	470		
/FOL.COM/	6544	14							
FOL,RES	6560	211	SL-SYSLIB	05/16/78	COMPASS	3.5	470		
FOL,MHI	6771	222	SL-SYSLIB	05/16/78	COMPASS	3.5	470		
SYS,PH	7213	40	SL-SYSLIB	05/16/78	COMPASS	3.5	470		

.114 CP SECONDS

23300R CM STORAGE USED

MFF NR2- CYR175-SH1 4LB7/R6B 05/15/78
09.11.35.DON006M FROM /SH
09.11.35.IP 00000304 WORDS - FILE INPUT , DC 04
09.11.35.DON. PSN-02/R.72CT011A.MILLER
09.11.37.FIN,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.HS 3504 WORDS 1 17920 MAX USED)
09.11.42.CPA .259 SEC. .259 ADJ.
09.11.42.IJ 1.573 SEC. 1.573 ADJ.
09.11.42.CM 29.455 KWS. 1.809 ADJ.
09.11.42.SS .3642
09.11.42.PP 6.800 SEC. DATE 06/27/78
09.11.42.FI END OF INR. SH

PARAMETERS

COBOL .

THE CALL:

ENTER SUB, PARM1, PARM2, PARM3

THE CODE GENERATED:

* IDENTICAL TO FORTRAN EXTENDED

```

00001 IDENTIFICATION DIVISION.
00002
00003 PROGRAM-ID. PAPAN.
00004
00005 ENVIRONMENT DIVISION.
00006 CONFIGURATION SECTION.
00007 SOURCE-COMPUTER. 6600.
00008 OBJECT-COMPUTER. 6600.
00009 SPECIAL-NAMES.
00010     OUTPUT-C IS PRINT.
00011
00012 DATA DIVISION.
00013 WORKING-STORAGE SECTION.
00014 77 PARM1 USAGE IS COMPUTATIONAL-2 VALUE IS 3.
00015 77 PARM2 USAGE IS COMPUTATIONAL-2 VALUE IS 4.
00016 77 PARM3 USAGE IS COMPUTATIONAL-2 VALUE IS 0.
00017 77 PARM4 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
00018 77 PARM5 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.
00019 77 PARM6 USAGE IS COMPUTATIONAL SIZE IS 1 DIGITS.

```

PAPAN

AO 0112

COBOL V4.7 L485 01/08/79 10.25.50.

PAGE 2

```

00020 PROCEDURE DIVISION.
00021
00022 BEGIN.
00023     DISPLAY "PARAMETER 1 IS " PARM1 UPON PRINT.
00024     DISPLAY " PARAMETER 2 IS " PARM2 UPON PRINT.
00025     DISPLAY " PARAMETER 3 IS " PARM3 UPON PRINT.
00026
00027     ENTER SAM, PARM2, PARM1, PARM3.
00028     MOVE PARM1 TO PARM4.
00029     MOVE PARM2 TO PARM5.
00030     MOVE PARM3 TO PARM6.
00031     DISPLAY " PARAMETER 1 IS " PARM4 UPON PRINT.
00032     DISPLAY " PARAMETER 2 IS " PARM5 UPON PRINT.
00033     DISPLAY " THE SUM IS " PARM6 UPON PRINT.
00034     STOP RUN.

```

PAPAN LENGTH IS 000151
061000B SCH USED

SAM

COMPASS 3.6-476.

01/08/79 10.26.03.

PAGE 1

0 00000000000000000000
1 53210 5041000001 53340
2 30623 24606 5041000002
3 53640 0400000000 +
4

IDENT SAM
ENTRY SAM
LIST -B,-H
DATA 0
SA2 X1
SA4 A1+1
SA3 X4
FX6 X2+X3
NX6
SA4 A1+2
SA6 X4
EQ SAM
END

ENTRY/EXIT ADDRESS
PARAMETER 1 IN X2
ADDRESS OF SECOND PARAMETER
PARAMETER 2 IN X3

PICK UP 3RD PARAMETER ADDRESS
STORE 3RD PARAMETER

47500B CM STORAGE USED
MODEL 73 ASSEMBLY

13 STATEMENTS
0.035 SECONDS

1 SYMBOLS
0 REFERENCES

11-38

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=COB10	567	751	SL-COBOL	11/16/78	COMPASS	3.6	485		
D=CRM15	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=USPLY	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=TRUHL	3133	121	SL-COBOL	11/16/78	COMPASS	3.6	485		
D=ZNTAU	3254	12	SL-COBOL	11/16/78	COMPASS	3.6	485		
D=LENDP	3266	73	SL-COBOL	11/16/78	COMPASS	3.6	485		
/FOL.COM/	3361	14							
FOL.RES	3375	211	SL-SYSLIB	11/02/78	COMPASS	3.6	485		
FOL.HMI	3606	222	SL-SYSLIB	11/02/78	COMPASS	3.6	485		
CPU.SYS	4030	40	SL-SYSLIB	11/15/78	COMPASS	3.6	476		
CHF.ALF	4070	160	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHF.CSF	4250	6	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHM.FFA	4256	14	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHF.FHF	4272	36	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHM.H	4330	214	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CHF.SLF	4544	22	SL-SYSLIB	11/15/78	COMPASS	3.6	485		
CTLSRM	4566	433	SL-SYSLIB	11/16/78	COMPASS	3.6	485		
EHRSHM	5221	25	SL-SYSLIB	11/16/78	COMPASS	3.6	485		
LISTSRM	5246	67	SL-SYSLIB	11/16/78	COMPASS	3.6	485		

FAST DYNAMIC LOADER RESIDENT.
FOL MEMORY MANAGER INTERFACE.
PROCESS SYSTEM REQUEST.
CHM VI.1 - ALLOCATE FIXED.
CHM VI.1 - CHANGE SPECS FIXED.
CHM VI.1 - FIXED FREE ALGORITHM.
CHM VI.1 - FREE FIXED.
CHM VI.1 - RESIDENT SUBROUTINES.
CHM VI.1 - SHRINK AT LWA FIXED.
CRM CONTROLLING ROUTINE.
CRM ENHOR PROCESSOR ENTRY.
CRM - ALLOCATE SPACE FOR LIST OF FILES

.316 CP SECONDS

21600B CH 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 NBI- CY874-SN108 SC/R08 11/14/78
 10.25.54.DON00KB FROM /OH
 10.25.54.IP 00000384 WORDS - FILE INPUT , DC 04
 10.25.55.DON.TS. 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.END 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.
 10.26.06.EJ END OF JOB, OH DATE 01/08/79

00001 IDENTIFICATION DIVISION.
00002
00003 PROGRAM-ID. ASTERISK.
00004
00005 ENVIRONMENT DIVISION.
00006
00007 CONFIGURATION SECTION.
00008 SOURCE-COMPUTER. 6600.
00009 OBJECT-COMPUTER. 6600.
00010 INPUT-OUTPUT SECTION.
00011 FILE-CONTROL.
00012 SELECT OUTFILE ASSIGN TO OUTPUT.
00013
00014 DATA DIVISION.
00015
00016 FILE SECTION.
00017 FD OUTFILE
00018 LABEL RECORDS ARE OMITTED
00019 DATA RECORD IS XXXX.
00020 01 XXXX PICTURE X(136).
00021 WORKING-STORAGE SECTION.
00022 77 ONE PICTURE 9(10) USAGE IS COMPUTATIONAL-1 VALUE IS 1.
00023 77 FOUR PICTURE 9(10) USAGE IS COMPUTATIONAL-1 VALUE IS 4.
00024 77 TEN PICTURE 9(10) USAGE IS COMPUTATIONAL-1 VALUE IS 10.
00025 77 XWORD PICTURE 9(10) USAGE IS COMPUTATIONAL-1.
00026 77 XPOS PICTURE 9(10) USAGE IS COMPUTATIONAL-1.
00027 01 HDR.
00028 02 FILLER PICTURE X(22) VALUE # THE STAR WAS IN WORD #.
00029 02 WORD PICTURE ZZZZZZZZ9.
00030 02 FILLER PICTURE X(20) VALUE # CHARACTER POSITION #.
00031 02 POS PICTURE ZZZZZZZZ9.
00032 01 A PICTURE X(20) VALUE # FOR TABLE A#.
00033 01 B PICTURE X(20) VALUE # FOR TABLE B#.
00034 01 C PICTURE X(20) VALUE # FOR TABLE C#.
00035 01 BUFF.
00036 02 AA PICTURE X(10) OCCURS 10 TIMES.
00037 02 BB PICTURE X(10) OCCURS 10 TIMES.
00038 02 CC PICTURE X(10) OCCURS 10 TIMES.

ASTERIS

AD 0113

CONOL V4.7 L470 06/27/78 09.07.03.

PAGE 2

```

00039  PROCEDURE DIVISION.
00040  ROLL OUT.
00041  OPEN OUTPUT OUTFILE.
00042  MOVE *AB120 TO AA(10).
00043  MOVE *AB*4 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
0612000 SCM USED

11-42

COMPASS 3.5-470.

06/27/78 09.07.06.

PAGE 1

SEARCH
STORAGE ALLOCATION.

ADDRESS LENGTH

0 22
22

BINARY CONTROL CARDS.

IDENT SEARCH

END

BLOCKS	TYPE	ADDRESS	LENGTH
PROGRAM*	LOCAL	0	21
LITERALS*	LOCAL	21	1

ENTRY POINTS.

SEARCH 0*

IDENT SEARCH
ENTRY SEARCH
CALL SEARCH (ARRAY NAME, ARRAY LENGTH, WORD NUMBER, CHARACTER POSITION)

0	7777777777777777	47	STAR	FOR	47R	ENTRY/EXIT
1	53210		SEARCH	DATA	-0	ADDRESS OF ARRAY IN A2, 1ST WORD IN X2
	5011000001			SA2	X1	INDEX TO NEXT PARAMETER (2ND PARAMETER)
	53110			SA3	X1	ADDRESS OF ARRAY LENGTH IN A3, VALUE IN X3
2	5011000001			SA1	A1.1	INDEX TO NEXT PARAMETER (3RD PARAMETER)
	57410			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			SB1	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	X	X7 = NUMBER OF BITS PER CHARACTER
	76020			SX0	B2	CHARACTER INDEX TO X0
	42007			IX0	X0*X7	ACTUAL SHIFT COUNT TO X0
7	67400			SX4	X0	NOW TO B4
	1170000077			SX7	77R	LOWER 6 BIT MASK TO X7
	23A41			AX6	B4.X1	SHIFT X1 RIGHT B4, PUT IT IN X6
10	11667			BX6	X6*X7	LOGICAL PRODUCT ALL BUT LOWER SIX BITS
	72667777710			SX6	X6-STAR	SUBTRACT CODE FOR CHARACTER BEING LOOKED FOR
11	0706000016			ZR	X6, FOUND	GO TO FOUND IF EQUAL TO ZERO
	6122777776			SB2	B2-1	REDUCE SHIFT INDEX
12	0420000006			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 NOT FOUND FLAG *** NO FIND ***
	10A11			RX6	X1	TRANSFER X6 TO X1
	54A40			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, 1 TO 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			

47300R SCH STORAGE USED

MODEL 174 ASSEMBLY

41 STATEMENTS

0.055 SECONDS

5 SYMBOLS

12 REFERENCES

SEARCH
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 F	2/05 L	2/34	2/40
STAR	47		2/04 D	2/27		

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	PROCESSOR	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	
/ASTERIS/	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=CKEYL	1072	15	SL-COBOL	05/07/78	COMPASS	3.5	470		
C=COBOL	1107	751	SL-COBOL	05/07/78	COMPASS	3.5	470		
D=CHM15	2060	10	SL-COBOL	05/07/78	COMPASS	3.5	470		
D=SOLO	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=COLSO	4010	114	SL-COBOL	05/07/78	COMPASS	3.5	470		
D=PCDCM	4124	116	SL-COBOL	05/07/78	COMPASS	3.5	470		
C=NV1	4242	100	SL-COBOL	05/07/78	COMPASS	3.5	470		
D=TRUHL	4362	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		CMH VI.1 - ALLOCATE FIXED.
CMF,CSF	4736	6	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH VI.1 - CHANGE SPECS FIXED.
CMH,FFA	4744	14	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH VI.1 - FIXED FREE ALGORITHM.
CMF,FRF	4760	36	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH VI.1 - FREE FIXED.
CMH,P	5016	213	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH VI.1 - RESIDENT SUBROUTINES.
CMF,SLF	5231	22	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CMH VI.1 - SHRINK AT LWA FIXED.
CTL,SLAL	5253	25	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CRM CONTROLLER - LABEL PROCESSING.
CTL,SRM	5300	601	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CRM CONTROLLING ROUTINE.
CTL,SWR	6101	31	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CRM CONTROLLER - WEOX, REWIND
ERR,SRM	6132	25	SL-SYSLIB	05/02/78	COMPASS	3.5	470		CRM ERROR PROCESSOR ENTRY.
LIST,SRM	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,MHI	6472	222	SL-SYSLIB	05/16/78	COMPASS	3.5	470		FDL MEMORY MANAGER INTERFACE.
SYS,SRM	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

23200R 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 NR2- CYB175-SN1 4L87/R6B 05/15/78
09.06.59.DON0052 FROM /SH
09.06.59.IP 00000704 WORDS - FILE INPUT , DC 04
09.06.59.DON. PSD.027R.72CT011A,MILLER
09.07.02.COROL.
09.07.04.COMPILEING ASTERIS
09.07.05. 000 E AND I/O DIAGNOSTICS ISSUED
09.07.05. 061200R SCM USED
09.07.05. .142 CP SECONDS COMPILATION TIME
09.07.05.END COROL
09.07.05.COMPASS.
09.07.07. ASSEMBLY COMPLETE. 47300R SCM USED.
09.07.07. 0.109 CPU SECONDS ASSEMBLY TIME.
09.07.07.LGO.
09.07.12.EXIT.
09.07.12.OP 0000204R 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.I0 1.816 SEC. 1.816 ADJ.
09.07.12.CH 45.907 KRS. 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

11-45

QUIZZES

1. PLACE THE FOLLOWING NUMBERS IN FLOATING POINT PRESENTATION. ALL NUMBERS ARE NOW DECIMAL.

$$5 \times 2^6$$

$$5 \times 2^{-6}$$

$$-(5 \times 2^6)$$

$$-(5 \times 2^{-6})$$

$$15$$

$$-10$$

$$-(1 \times 2^{64})$$

$$1 \times 2^{-1}$$

2. CONVERT THE FOLLOWING NUMBERS FROM FLOATING POINT FORM TO (C X 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.

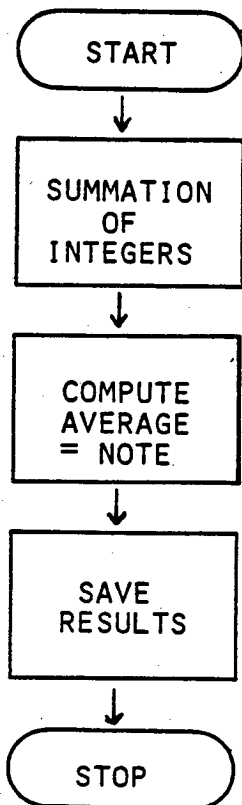
PROJECTS

PROBLEM SET 1A

AVERAGE INTEGERS

WRITE A PROGRAM TO FIND THE AVERAGE OF 10 INTEGERS.

USE THE FOLLOWING FLOWCHART.



NOTE:

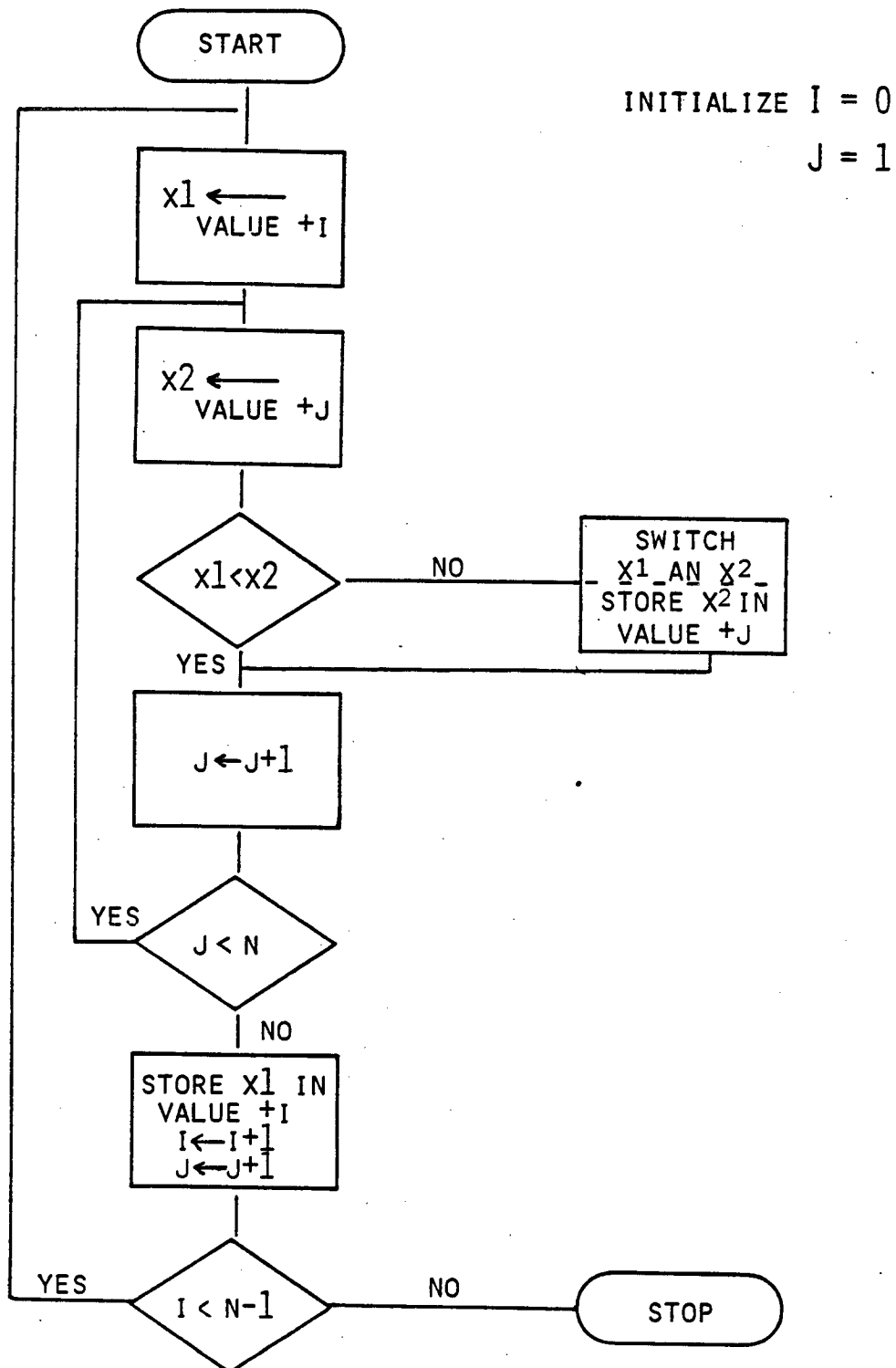
IF A AND B ARE INTEGER VALUES, THEN $B/A = Q$ WITH A REMAINDER OF $B - (Q \times A)$.

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 $Q \times A$.
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

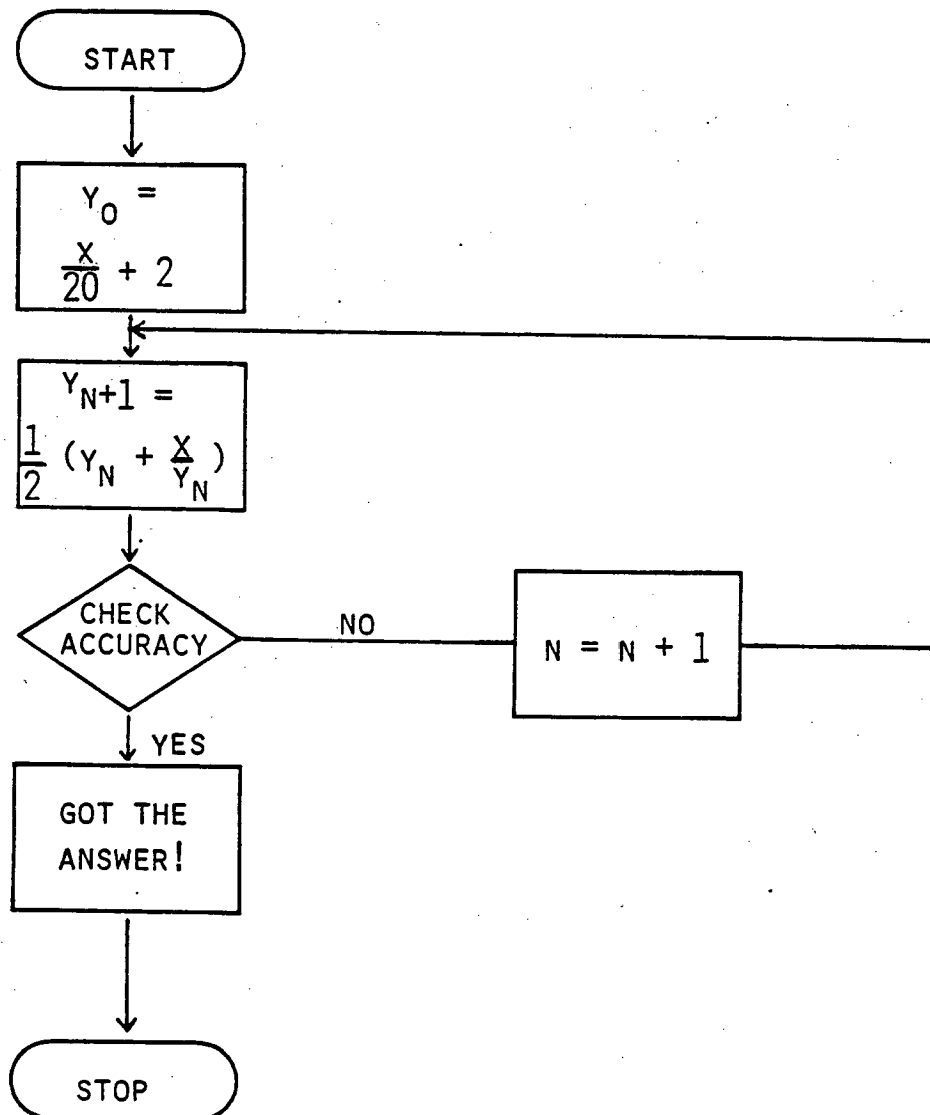
SQUARE ROOT

WRITE YOUR OWN SQUARE ROOT SUBROUTINE USING THE NEWTON ITERATION....

$$Y_{N+1} = \frac{1}{2} (Y_N + \frac{X}{Y_N}) \text{ WITH } Y_0 = \frac{X}{20.0} + 2.0$$

THE INPUT VALUE IS X WHILE THE RETURNED VALUE IS YOUR APPROXIMATION TO THE SQUARE ROOT OF X. OBTAIN A RELATIVE ACCURACY OF....

$$\frac{1}{2} (10^{-9}) \quad (Y_{N+1} - Y_N) / Y_{N+1}$$



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.

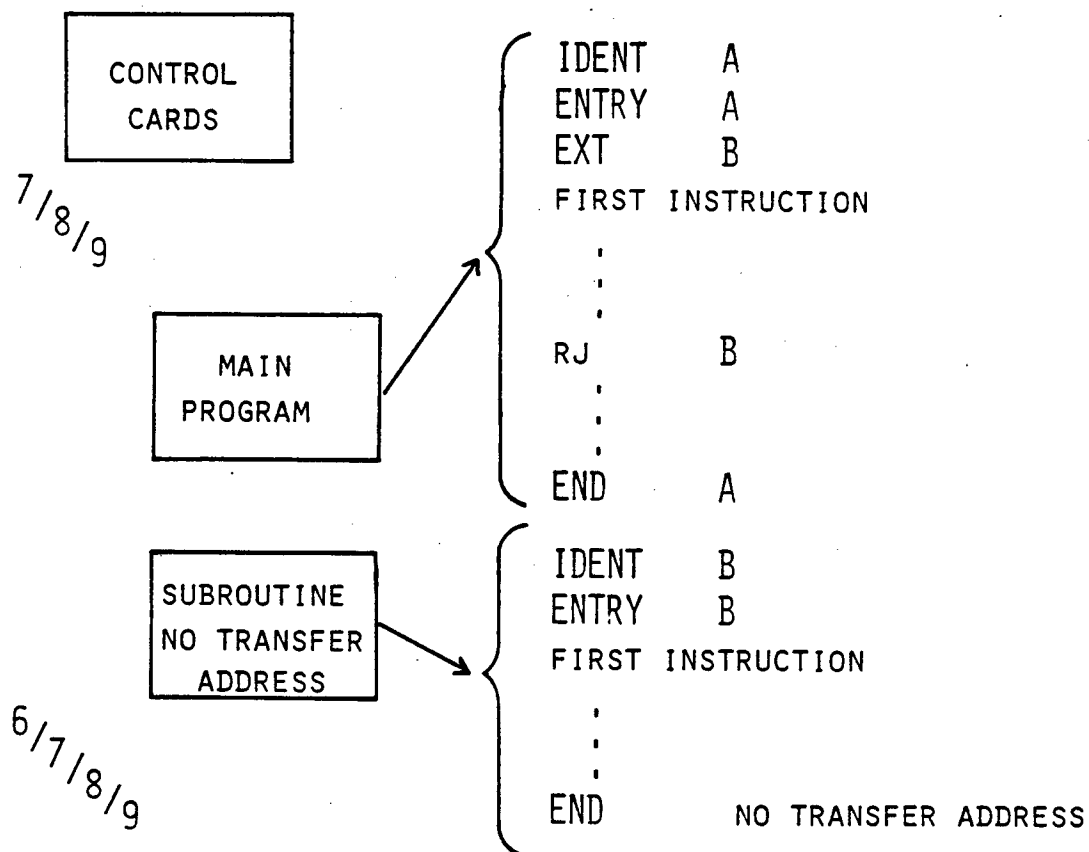
THE X REGISTER SHOULD BE SET TO ZERO IF CHARACTER 1 IS A \$.

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



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.

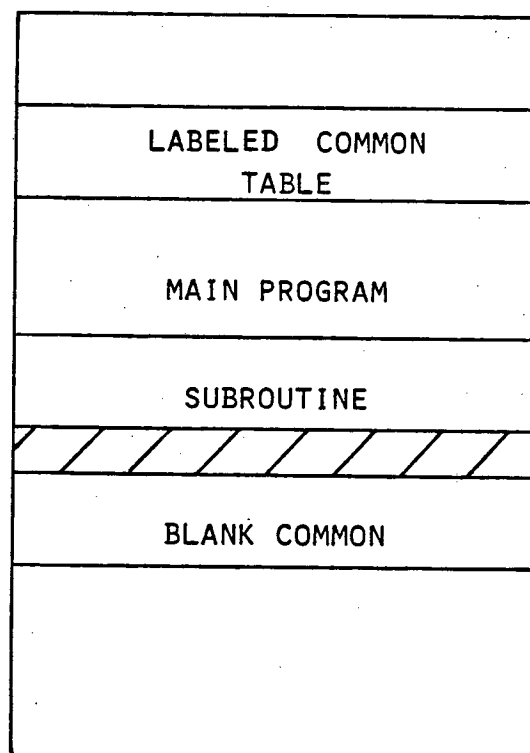
IN THE SUBROUTINE, FETCH THE NUMBER FROM BLANK COMMON,
LOOK UP THE DISPLAY CODE IN TABLE.

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

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).
 $N = N(N-1)(N-2)\cdots(2)(1)$

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 Q_1, Q_2, Q_3

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, RE-
TURNING 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 DIFFI-
CULT.

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 COMBI-
NATION. THE RESULTS CAN BE PLACED IN
P2, P2 + 1 AND P2 + 2 IF NECESSARY.
Xk, Xk + 1 AND Xk + 2 CAN CONTAIN OP-
ERANDS. 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 for each area and then add comments explaining your rating.

Rating Key

<i>Excellent</i>	<i>5</i>
<i>Very Good</i>	<i>4</i>
<i>Good</i>	<i>3</i>
<i>Fair</i>	<i>2</i>
<i>Poor</i>	<i>1</i>

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 _____ Date of Presentation _____
Name _____ Field or Type of Business _____
Title _____ Years of Experience _____
Business Address _____ 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).

CUT ALONG LINE

AA3419 REV. 4/79 PRINTED IN U.S.A.

NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

FOLD ON DOTTED LINES AND STAPLE

STAPLE

STAPLE

FOLD

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS

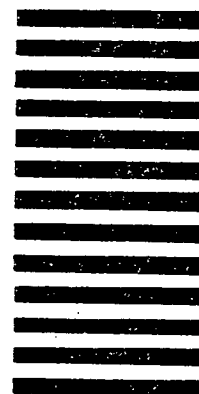
PERMIT NO. 8241

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

3 instructions move 60 bits

CONTROL DATA SEMINARS

 an education service of
CONTROL DATA CORPORATION

CORPORATE HEADQUARTERS
P.O. BOX 0
MINNEAPOLIS, MINNESOTA 55440