



INTERNATIONAL ATOMIC ENERGY AGENCY
UNITED NATIONS EDUCATIONAL, SCIENTIFIC AND CULTURAL ORGANIZATION



INTERNATIONAL CENTRE FOR THEORETICAL PHYSICS
34100 TRIESTE (ITALY) - P.O.B. 58 - MIRAMARE - STRADA COSTIERA 11 - TELEPHONES: 224281/2/3/4/5/6
CABLE: CENTRATON - TELEX 480392-1

SMR/90 - 24

COLLEGE ON MICROPROCESSORS:

TECHNOLOGY AND APPLICATIONS IN PHYSICS

September - 2 October 1981

1. SOLUTIONS TO EXERCISE ON MLP
COMPILER
2. SOLUTIONS TO EXERCISE ON THE
MLP COMPILER USING ASSEMBLER
SUBROUTINES

U. RAICH
EP Division
CERN
1211 Geneva 23
Switzerland

These are preliminary lecture notes, intended only for distribution to participants. Missing or extra copies are available from Room 230.

SOLUTION No. 7.1

ICTP COLLEGE ON MICROPROCESSORS

(TRIESTE, SEPTEMBER 1981)

PRACTICAL WORK SESSIONS

first calculate the determinant and stop the program if it is 0. If a unique solution exists then multiply equation 1 by $a(2,1)$ and equation 2 by $a(1,1)$ to get:

```
(4)  a(1,1)*a(2,1)*x(1)+a(1,2)*a(2,1)*x(2)=b(1)*a(2,1)
(5)  a(1,1)*a(2,1)*x(1)+a(1,1)*a(2,2)*x(2)=b(2)*a(1,1)
```

Subtract 5 from 4 and solve the resulting equation for $x(2)$:

```
(6)  x(2)=(b(1)*a(2,1)-b(2)*a(1,1))/(a(1,2)*a(2,1)-a(1,1)*a(2,2))
(7)  x(2)=(b(2)*a(1,1)-b(1)*a(2,1))/det
```

Put $x(2)$ in equation 1:

```
(8)  x(1)=(b(1)-a(1,2)*x(2))/a(1,1)
```

I declared all variables to be BINARY(2) even if the elements of the matrix, the vector and the solution would fit into 1 byte because the determinant can be too big for 1 byte and intermediate results may be too long for 1 byte.

```
!
! PROGRAM TO SOLVE A SET OF LINEAR EQUATIONS WITH 2 UNKNOWN
!
ORIGIN "0"
DECLARE
  MATR(2,2) BINARY(2),
  VECT(2) BINARY(2), SOL(2) BINARY(2),
  DET BINARY(2)
!
PROCEDURE OPTIONS(MAIN)
!
! CALCULATE DETERMINANT
!
DET=MATR(1,1)*MATR(2,2)-MATR(1,2)*MATR(2,1)
!
! IF NO UNIQUE SOLUTION, STOP!
!
IF DET EQ 0 THEN GOTO STOP
!
! CALCULATE SOLUTION
!
SOL(2)=(VECT(2)*MATR(1,1)-VECT(1)*MATR(2,1))/DET
SOL(1)=(VECT(1)*MATR(1,2)-SOL(2)*MATR(1,1))/MATR(1,1)
!
$STOP SWI !RETURN TO MONITOR
$ FCB 0
END
```

- 4 -

30	0092	BD	00C6		JSR	ZF18	
31	0095	43			COMA		
32	0096	53			COMB		
33	0097	99	09		ADCA	VECT+1	
34	0099	D9	08		ADCB	VECT	
35				*00024	!		
36	009B	DE	00		LDX	MATR	
37	009D	BD	0140		JSR	ZF19	
38	00A0	D7	0C		STAB	SOL	
39	00A2	97	0D		STAA	SOL+1	
40	00A4	3F		STOP	SWI		!RETURN TO MONITOR
41	00A5	00			FCB	0	
42				*00027	END		
43	00A6	0000		Z437	FDB	0	

32	00DB	36	
33	00DC	96	16
34	00DE	32	
35	00DF	2A	03
36	00E1	BD	00A8
37	00E4	39	
38	00E5	DF	16
39	00E7	DF	18
40	00E9	CE	0012
41	00EC	E7	05
42	00EE	2A	03
43	00F0	BD	00A8
44	00F3	E7	08
45	00F5	A7	09
46	00F7	A6	07
47	00F9	E6	06
48	00FB	2A	07
49	00FD	BD	00A8
50	0100	E7	06

ZF18A	PSHA		SIGN DOUBLE RESULT
	LDAA	T+4	
	PULA		
	BPL	ZF18B	ALIKE
	JSR	ZF07	COMPLEMENT DOUBLE
ZF18B	RTS		
ZF18D	STX	T+4	PREP DOUBLE
	STX	T+6	
	LDX	#T	
	STAB	5,X	
	BPL	ZF181	
	JSR	ZF07	
ZF181	STAB	8,X	
	STAA	9,X	
	LDAA	7,X	
	LDAB	6,X	
	BPL	ZF182	
	JSR	ZF07	
	STAB	6,X	

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
1	00A8	43		ZF07 COMA ** COMPLEMENT DOUBLE
2	00A9	53		COMB
3	00AA	89	00	ADCA #0
4	00AC	C9	00	ADCB #0
5	00AE	39		RTS
6	00AF	5F		ZF071 CLRB ONE BYTE MOVE
7	00B0	DE	14	ZF072 LDX T+2 *** FILL FROM LEFT **
8	00B2	A7	00	STAA 0,X
9	00B4	09		DEX
10	00B5	DF	14	STX T+2
11	00B7	5A		DECB
12	00B8	2E	F6	BGT ZF072
13	00BA	39		RTS
14	00BB	DE	14	ZF073 LDX T+2 *** FILL FROM RIGHT *
15	00BD	A7	00	STAA 0,X
16	00BF	08		INX
17	00C0	DF	14	STX T+2
18	00C2	5A		DECB
19	00C3	2E	F6	BGT ZF073
20	00C5	39		RTS
21				
22				
23	00C6	20	0D	ZF18 BRA ZF18F *** MULTIPLY DOUBLE *
24	00C8	7F	0012	ZF58 CLR T ** MIXED DIVIDE B2/B1
25	00C8	DF	13	STX T+1
26	00CD	2A	03	BPL ZF581
27	00CF	73	0012	COM T
28	00D2	DE	12	ZF581 LDX T
29	00D4	39		RTS SHIFTS X RIGHT & ARIT
30	00D5	BD	00E5	ZF18F JSR ZF180 PREPARE
31	00D8	BD	010B	JSR ZF184 MULTIPLY

- 5 -

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
1	0102	A7	07	STAA 7,X
2	0104	A6	05	ZF182 LDAA 5,X
3	0106	A8	04	EORA 4,X
4	0108	A7	04	STAA 4,X
5	010A	39		RTS
6	010B	86	1D	ZF184 LDAA #16 *** MULTIPLY ***
7	010D	A7	05	STAA 5,X
8	010F	E6	06	LDAB 6,X
9	0111	A6	07	LDAA 7,X
10	0113	48		ZF185 ASLA
11	0114	59		ROLB
12	0115	68	09	ASL 9,X
13	0117	69	08	ROL 8,X
14	0119	24	04	BCC ZF186
15	011B	AB	07	ADDA 7,X
16	011D	E9	06	ADCB 6,X
17	011F	6A	05	ZF186 DEC 5,X
18	0121	26	F0	BNE ZF185
19	0123	39		RTS
20	0124	CE	0012	ZF187 LDX #T PREP SINGLE
21	0127	6F	06	CLR 6,X
22	0129	6F	08	CLR 8,X
23	012B	A7	04	STAA 4,X
24	012D	2A	01	BPL ZF188
25	012F	40		NEGA
26	0130	A7	09	ZF188 STAA 9,X

- 6 -

27 0132 E7 05
 28 0134 2A 01
 29 0136 50
 30 0137 E7 07
 31 0139 A6 05
 32 0138 A8 04
 33 0130 A7 04
 34 013F 39
 35
 36
 37 0140 20 08
 38 0142 B0 0190
 39 0145 20 06
 40 0147 B0 0190
 41 014A B0 0190
 42 0140 B0 00E5
 43 0150 B0 015A
 44 0153 06 1A
 45 0155 96 18
 46 0157 7E 00DB
 47 015A 86 01
 48 015C 60 06
 49 015E 28 08
 50 0160 4C

STAB 5,X
 BPL ZF189
 NEG8
 ZF189 STAB 7,X
 LDAA 5,X
 EORA 4,X
 STAA 4,X
 RTS
 ZF19 BRA ZF19F *** DIVIDE DOUBLE ***
 ZF59 JSR ZF19A *** MIXED DIVIDE B2/B
 BRA ZF19F
 ZF09 JSR ZF19A *** REV DBL MIXED DIV
 ZF99 JSR ZF19B *** REV DBL DIVIDE
 ZF19F JSR ZF180 PREPARE
 JSR ZF190 DIVIDE
 LDAB T+8
 LDAA T+9
 JMP ZF18A FIX DOUBLE SIGN
 LDAA #1
 ZF190 TST 6,X
 BMI ZF192
 ZF191 INCA

22 0189 66 07
 23 018B 6A 05
 24 018D 26 E6
 25 018F 39
 26 0190 7F 0012
 27 0193 DF 13
 28 0195 2A 03
 29 0197 73 0012
 30 019A DE 12
 31 019C 39
 32 019D DF 12
 33 019F 07 14
 34 01A1 97 15
 35 01A3 06 12
 36 01A5 96 13
 37 01A7 DE 14
 38 01A9 39
 39 01AA
 40 01AA

ROR 7,X
 DEC 5,X
 BNE ZF193
 RTS
 ZF19A CLR T ** MIXED DIVIDE B2/B1
 STX T+1
 BPL ZF591
 COM T
 ZF591 LDX T
 RTS
 ZF19B STX T ** REVERSE DOUBLE DIV
 STAB T+2
 STAA T+3
 LDAB T
 LDAA T+1
 LDX T+2
 RTS
 ZFFF EQU * SWAPS BA & X REGS
 END

1 IDENTIFIER CROSS REFERENCE.

CROSS ASSEMBLER FOR MOTOROLA 680
 VERS. 1.0, RUN AT CERN'S CDC COM

CROSS ASSEMBLER FOR MOTOROLA 680
 VERS. 1.0, RUN AT CERN'S CDC COM

1

LINE ADDR CODE EXTENSIONS SOURCE - STATEMENT

1 0161 68 07 ASL 7,X
 2 0163 69 06 ROL 6,X
 3 0165 28 04 BMI ZF192
 4 0167 81 11 CMPA #17
 5 0169 26 F5 BNE ZF191
 6 016B A7 05 ZF192 STAA 5,X
 7 016D A6 09 LDAA 9,X
 8 016F E6 08 LDAB 8,X
 9 0171 6F 08 CLR 8,X
 10 0173 6F 09 CLR 9,X
 11 0175 A0 07 ZF193 SUBA 7,X
 12 0177 E2 06 SBCB 6,X
 13 0179 24 07 BCC ZF194
 14 017B AB 07 ADDA 7,X
 15 017D E9 06 ADCB 6,X
 16 017F 0C CLC
 17 0180 20 01 SRA ZF195
 18 0182 0D ZF194 SEC
 19 0183 69 09 ZF195 ROL 9,X
 20 0185 69 08 ROL 8,X
 21 0187 64 06 LSR 6,X

DET	00000010 ABS	2/22	2/47	2/48	3/01	3/22	
MATR	00000000 ABS	2/16	2/30	2/31	2/32	2/39	2/40
		3/27	3/28	3/36			
MPLR10	PRG	2/01					
SOL	0000000C ABS	2/19	3/24	3/25	3/29	3/38	3/39
STOP	000000A4 ABS	3/03	3/40				
T	00000012 ABS	2/27	2/28	2/37	2/38	2/45	2/46
		3/21	4/07	4/10	4/14	4/17	4/24
		4/33	4/38	4/39	4/40	5/20	5/44
		6/29	6/30	6/32	6/33	6/34	6/35
VECT	00000008 ABS	2/17	3/08	3/09	3/14	3/15	3/33
ZF09	00000147 ABS	5/40					
ZFFF	000001AA EQU	6/39					
ZF071	000000AF ABS	4/06					
ZF072	00000080 ABS	4/07	4/12				
ZF073	0000008B ABS	4/14	4/19				
ZF07	000000A8 ABS	3/44	4/36	4/43	4/49		
ZF18A	000000CB ABS	4/32	5/46				
ZF18B	000000E4 ABS	4/35	4/37				
ZF18F	000000D5 ABS	4/23	4/30				
ZF180	000000E5 ABS	4/30	4/38	5/42			
ZF181	000000F3 ABS	4/42	4/44				
ZF182	00000104 ABS	4/48	5/02				
ZF184	0000010B ABS	4/31	5/06				
ZF185	00000113 ABS	5/10	5/18				
ZF186	0000011F ABS	5/14	5/17				
ZF187	00000124 ABS	5/20					
ZF188	00000130 ABS	5/24	5/26				

ZF189	00000137 ABS	5/28	5/30				
ZF18	00000006 ABS	2/33	2/42	3/11	3/17	3/30	4/23
ZF19A	00000190 ABS	5/38	5/40	6/26			
ZF19B	00000190 ABS	5/41	6/32				
ZF19F	00000140 ABS	5/37	5/39	5/42			
ZF190	0000015A ABS	5/43	5/47				
ZF191	00000160 ABS	5/50	6/05				
ZF192	0000016B ABS	5/49	6/03	6/06			
ZF193	00000175 ABS	6/11	6/24				
ZF194	00000182 ABS	6/13	6/18				
ZF195	00000183 ABS	6/17	6/19				
ZF19	00000140 ABS	3/23	3/37	5/37			
ZF581	00000002 ABS	4/26	4/28				
ZF58	00000008 ABS	4/24					
ZF591	0000019A ABS	6/28	6/30				
ZF59	00000142 ABS	5/38					
ZF99	0000014A ABS	5/41					
Z000	0000003A ABS	2/28					
Z002	00000067 ABS	3/02	3/08				
Z437	000000A6 ABS	2/50	3/43				

END OF ASM 6800- 3.724 CP SECONDS ASSEMBLY TIME.
1
*EOR

1PAGE 1 01/09/81 21.27.26. CUFOM -> MOTOROLA TRANSLATOR (1.1)

THE CONTROL CARD PARAMETERS ARE

OPTION INPUT FILE IS- OPTIONS
CUFOM INPUT FILE IS- CUFOM
BINARY OUTPUT FILE IS- LOADMOD
LIST OUTPUT FILE IS- LMATR

THE SECTION ORIGINS ARE SPECIFIED AS FOLLOWS

1 SECTION(1,ABSOLUTE)=0000.
1PAGE 2 01/09/81 21.27.26.

LOAD MAP FOR CUFOM -> MOTOROLA TRANSLATOR (1

MODULE	ENTRY	SECTION	ADDRESSES :	PROCESSOR COMMENTS
NAME	NAME	TYPE AND NAME	MEMORY ORIGIN LISTING	

MPLR10

CROSS ASSEMBLER FOR MOTOROLA 6800 PROC
RUN AT CERN'S CDC COMPUTER CENTRE 01

NO TRANSFER ADDRESS SPECIFIED FOR THIS MODULE

>>>>> 0 ERROR DETECTED IN CUFOM INPUT FILE <<<<<<<

ICTP COLLEGE ON MICROPROCESSORS

(TRIESTE, SEPTEMBER 1981)

PRACTICAL WORK SESSIONS

In the explanation of the exercise it was already shown how the program can be written. However I have to apply one trick: Since there are no I/O commands in MPL I have to use assembly language routines to output the letters "A",.....,"Z" and the ".",,"\$. These routines can be called from MPL using the CALL statement. Parameters are transferred in the accumulator A (CALL SUBNAM<VAR>). The necessary routines are collected in the file MPLUT (MPL utilities) which is inserted into the translated MPL program using the assembler directive INSERT. All the assembly language subroutines (GETDAT, IOOPEN, DISPLAY, TEXT, IOCLOSE) are translated separately and put together into a Cufom library DISLIB using the librarian. The command:

```
BEGIN,MICMPL,,S=NLET,B=BLET,L=LLET,ALIB=MPLUT,CLIB=DISLIB
```

first compiles the program NLET, then it inserts the file MPLUT into the assembly language code (output from the compiler) then it links the cufom library DISLIB to the output file from the assembler and in the end it translates the complete program into Motorola S-format. BLET is the program ready for down-line loading. The assembler listings of the programs:

GETDAT, IOOPEN, IOCLOSE, TEXT are given below as well as the listing LLET of the complete translation process of the MPL program NLET.

```
$ OPT REL,MEX
$ DSCT
$ XDEF HIST,LETTER
DECLARE
LETTER,HIST(29),T(40),STACK,I
$ PSCT
$ XREF GETDAT,IOOPEN,DSPLAY,IOCLOSE
PROCEDURE OPTIONS(MAIN,STACK)
CALL IOOPEN
DO I=1 TO 29 !CLEAR HISTOGRAM
HIST(I)=0
END
!
!
SORT: CALL GETDAT !GET DATA INTO LETTER
!
IF LETTER EQ "20" THEN GOTO SORT !SKIP BLANKS
DO I= 'A' TO 'Z'
IF LETTER EQ I THEN GOTO FILL !IS DATA A LETTER (A-Z)?
END
!
I=I+1
IF LETTER EQ '.' THEN GOTO FILL ! IS IT A '.'?
!
I=I+1
IF LETTER EQ ',' THEN GOTO FILL !IS IT A ','?
!
I=I+1
FILL: HIST(I-64)=HIST(I-64)+1 !FILL HISTOGRAM
```

```
! I CONTAINS THE ASCII VALUE FOR A LETTER (65-91 DECIMAL)
! OR 92 FOR '.', 93 FOR ',', 94 FOR ALL OTHER CHARACTERS.
!
```

```
IF HIST(I-64) LT 23 THEN GOTO SORT
CALL DSPLAY
```

```
! OUTPUT "A"....."Z", ".", ",", "$"
!
```

```
CALL OUTSP<25>
DO I='A' TO 'Z'
CALL OUTCH<I>
END
CALL OUTCH<','>
CALL OUTCH<','>
CALL OUTCH<'$>
CALL IOCLOSE
```

```
STOP: GOTO STOP
$ INSERT
END
```

```
*****
* MACRO TO PERFORM MONITOR CALLS *
* MON X WILL CALL MONITOR ROUTINE NR. X *
*****
MON MACR
SWI
FCB %0
ENDM
*****
* SUBROUTINE TO OUTPUT A CHARACTER *
* CAN BE CALLED FROM A MPL ROUTINE USING: *
* CALL OUTCH<ASCII> *
*****
OUTCH MON 6
RTS
*****
* SUBROUTINE TO OUTPUT A NUMBER OF BLANKS *
* CAN BE CALLED FROM A MPL ROUTINE USING: *
* CALL OUTSP<NUMBER> *
*****
OUTSP TAB
MON 10
RTS
```

1 CROSS ASSEMBLER FOR MOTOROLA 68
VERS. 1.0, RUN AT CERN'S CDC CC

CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS

0
0
0 NAME OF MODULE : GETDAT
0 PROGRAM TYPE : SUBPROGRAM
0 EXTERNAL REFER.: LETTER POINT

0 S E C T I O N S:

0 PROGRAM SECTION : 000C IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
GETDAT

1 CROSS ASSEMBLER FOR MOTOROLA 68
VERS. 1.0, RUN AT CERN'S CDC CC

LINE ADDR CODE EXTENSIONS . S O U R C E - S T A T E M E N T

```

1 0000      NAM      GETDAT
2 0000      OPT      REL
3          * THIS SUBROUTINE GETS ASCII CHARACTERS FROM
4          * A TEXTFILE AND STORES THEM IN MEMORY LOCATION LETTER
5          *
6 0000      PSCT
7          XREF      LETTER,POINT
8          XDEF      GETDAT
9 0000 FE 0000*  GETDAT LDX POINT
10 0003 A6 00      LDAA 0,X
11 0005 08          INX
12 0006 FF 0000*  STX POINT
13 0009 B7 0000*  STAA LETTER
14 000C 39          RTS
15 000D          END

```

1 IDENTIFIER CROSS REFERENCE. CROSS ASSEMBLER FOR MOTOROLA 68
VERS. 1.0, RUN AT CERN'S CDC CC

IDENTIFIER	ADDRESS	REL	2/01	2/08	2/09
GETDAT	00000000	REL	2/01	2/08	2/09
LETTER		EXT	2/07	2/13	
POINT		EXT	2/07	2/09	2/12

END OF ASM 6800- 0.353 CP SECONDS ASSEMBLY TIME.

13

1 CROSS ASSEMBLER FOR MOTOROLA 68
VERS. 1.0, RUN AT CERN'S CDC CC

CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS

0
0
0 NAME OF MODULE : IOOPEN
0 PROGRAM TYPE : SUBPROGRAM
0 EXTERNAL REFER.: TEXT

0 S E C T I O N S:

0 PROGRAM SECTION : 0006 IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
IOOPEN

0 DATA SECTION : 0001 IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
POINT

1 CROSS ASSEMBLER FOR MOTOROLA 68
VERS. 1.0, RUN AT CERN'S CDC CC

LINE ADDR CODE EXTENSIONS . S O U R C E - S T A T E M E N T

```

1          *
2 0000      NAM      IOOPEN
3 0000      OPT      REL
4 0000      PSCT
5 0000 02      POINT RMB 2
6          XDEF      POINT
7 0000      PSCT
8          XDEF      IOOPEN
9          XREF      TEXT
10 0000 CE 0000*  IOOPEN LDX #TEXT
11 0003 FF 0000*  STX POINT
12 0006 39      RTS
13 0007          END

```

1 IDENTIFIER CROSS REFERENCE. CROSS ASSEMBLER FOR MOTOROLA 68
VERS. 1.0, RUN AT CERN'S CDC CC

IDENTIFIER	ADDRESS	REL	2/02	2/08	2/10
IOOPEN	00000000	REL	2/02	2/08	2/10
POINT	00000000	REL	2/05	2/06	2/11
TEXT		EXT	2/09	2/10	

END OF ASM 6800- 0.320 CP SECONDS ASSEMBLY TIME.

1

14

1 CROSS ASSEMBLER FOR MOTOROLA 68
CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS
0 VERS. 1.0, RUN AT CERN'S CDC CC
0
0
0 NAME OF MODULE : TEXT
0 PROGRAM TYPE : SUBPROGRAM
0 EXTERNAL REFER.: -NONE-
0 S E C T I O N S:

0 DATA SECTION : 016C IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
TEXT

1 CROSS ASSEMBLER FOR MOTOROLA 68
VERS. 1.0, RUN AT CERN'S CDC CC
LINE ADDR CODE EXTENSIONS . SOURCE - STATEMENT
1 *
2 0000 NAM TEXT
3 0000 DSCT
4 XDEF TEXT
5 0000 54 48 49 53 20 50 TEXT FCC /THIS PROGRAM WILL SHOW YOU HOW A PHYSICS EXPERIMENT/
6 0033 43 41 4E 20 42 45 FCC /CAN BE PROGRAMMED.NORMALLY THE FOLLOWING TASKS HAVE/
7 0066 54 4F 20 42 45 20 FCC /TO BE PERFORMED: ALL ELECTRONICS HAS TO BE INITIALIZE
8 009D 45 2E 47 2E 20 43 FCC /E.G. CAMAC CLEAR (IOOPEN). THEN THE DATA HAVE TO BE/
9 0000 54 41 48 45 4E 20 FCC /TAKEN (GETDAT) AND ONLINE SPECTRA HAVE TO BE GENERATE
10 0107 54 48 45 53 45 20 FCC /THESE SPETRA HAVE TO BE DISPLAYED ON A SCREEN (DSPLY)
11 013D 4E 4F 52 4D 41 4C FCC /NORMALLY THE DATA ARE ALSO WRITTEN ONTO MAGTAPE./
12 016D END

1 CROSS ASSEMBLER FOR MOTOROLA 68
IDENTIFIER CROSS REFERENCE. VERS. 1.0, RUN AT CERN'S CDC CC
TEXT 00000000 REL 2/02 2/04 2/05
END OF ASM 6800- 0.743 CP SECONDS ASSEMBLY TIME.

1 CROSS ASSEMBLER FOR MOTOROLA 68
CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS VERS. 1.0, RUN AT CERN'S CDC CC
0
0
0 NAME OF MODULE : IOCLOS

0 PROGRAM TYPE : SUBPROGRAM
0 EXTERNAL REFER.: -NONE-
0 S E C T I O N S:

0 PROGRAM SECTION : 0000 IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
IOCLOS

1 CROSS ASSEMBLER FOR MOTOROLA 68
VERS. 1.0, RUN AT CERN'S CDC CC

LINE ADDR CODE EXTENSIONS . SOURCE - STATEMENT
1 *
2 0000 NAM IOCLOS
3 0000 PSCT
4 XDEF IOCLOS
5 0000 39 IOCLOS RTS
6 0001 END

1 CROSS ASSEMBLER FOR MOTOROLA 68
IDENTIFIER CROSS REFERENCE. VERS. 1.0, RUN AT CERN'S CDC CC

IOCLOS 00000000 REL 2/02 2/04 2/05
END OF ASM 6800- 0.209 CP SECONDS ASSEMBLY TIME.

1 CROSS ASSEMBLER FOR MOTOROLA 68
CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS VERS. 1.0, RUN AT CERN'S CDC CC
0
0
0 NAME OF MODULE : MPLR10
0 PROGRAM TYPE : SUBPROGRAM
0 EXTERNAL REFER.: DSPLY GETDAT IOOPEN IOCLOS
0 S E C T I O N S:

0 PROGRAM SECTION : 00CD IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
-NONE-
0 DATA SECTION : 0047 IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
HIST LETTER

sk 4
- 1

CROSS ASSEMBLER FOR MOTOROLA 68

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
1	0000			NAM MPLR10
2				*MOTOROLA SPD, INC. OWNS AND IS RESPONSIBLE FOR M68MPL
3				* COPYRIGHT 1975 AND 1976 BY MOTOROLA INC.
4				*
5				* MOTOROLA'S M6800 MPL COMPILER, RELEASE 1.2
6				*
7				*
8				*** COMPILED WITH MPL VERSION 1.2
9	0000			OPT REL,MEX
10	0000			DSCT
11				XDEF HIST,LETTER
12				*00004 DECLARE
13	0000	01		LETTER RMB 1
14	0001	10		HIST RMB 29
15	001E	28		T RMB 40
16	0046	01		STACK RMB 1
17				*00005 LETTER,HIST(29),T(40),STACK,I
18	0047	01		I RMB 1
19	0000			PSCT
20				XREF GETDAT,IOOPEN,DSPLAY,IOCLOS
21				*00008 PROCEDURE OPTIONS(MAIN,STACK)
22	0000	8E	0046+	Z000 LDS #STACK
23				*00009 CALL IOOPEN
24	0003	8D	0000*	JSR IOOPEN
25				*00010 DO\$1=1 TO 29
26	0006	86	01	LDAA #1 !CLEAR HISTO
27	0008	B7	0047+	Z001 STAA I
28				*00011 HIST(I)=0
29	000B	CE	0001+	LDX #HIST
30	000E	F6	0047+	LDAB I
31	0011	8D	00B7+	JSR ZF1F
32	0014	6F	00	CLR 0,X
33	0016	B6	0047+	LDAA I
34	0019	81	10	CMPA #29
35	001B	24	04	BCC Z002
36	001D	4C		INCA
37	001E	7E	0008+	JMP Z001
38				END
39				*00012
40				*00013 !
41		0021+		*00014 SORT: CALL GETDAT !GET DATA INTO LETTER
42				Z002 EQU *
43	0021	8D	0000*	*00015 !
44				SORT JSR GETDAT
45	0024	B6	0000+	*00016 IF LETTER EQ "20" THEN GOTO SORT !SKIP BLANKS
46	0027	81	20	LDAA LETTER
47	0029	26	03	CMPA #32
48	002B	7E	0021+	BNE Z004
49				JMP SORT
				*00017 DO\$I= 'A' TO 'Z'
50	002E	86	41	Z004 LDAA #A

17

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
1	0030	B7	0047+	Z005 STAA I
2				*00018 IF LETTER EQ I THEN GOTO FILL !IS DATA A I
3	0033	B6	0000+	LDAA LETTER
4	0036	B1	0047+	CMPA I
5	0039	26	03	BNE Z007
6	003B	7E	0063+	JMP FILL
7				*00019 END
8	003E	B6	0047+	Z007 LDAA I
9	0041	81	5A	CMPA #Z
10	0043	24	04	BCC Z008
11	0045	4C		INCA
12	0046	7E	0030+	JMP Z005
13				*00020 !
14				*00021 I=I+1
15	0049	7C	0047+	Z008 INC I
16				*00022 IF LETTER EQ ' ' THEN GOTO FILL ! IS IT A ' '
17	004C	B6	0000+	LDAA LETTER
18	004F	81	2E	CMPA #'
19	0051	26	03	BNE Z00A
20	0053	7E	0063+	JMP FILL
21				*00023 !
22				*00024 I=I+1
23	0056	7C	0047+	Z00A INC I
24				*00025 IF LETTER EQ ' ' THEN GOTO FILL !IS IT A ' '
25	0059	81	2C	CMPA #'
26	005B	26	03	BNE Z00C
27	005D	7E	0063+	JMP FILL
28				*00026 !
29				*00027 I=I+1
30	0060	7C	0047+	Z00C INC I
31				*00028 FILL: HIST(I-64)=HIST(I-64)+1 !FILL HISTOG
32				*00029 !
33				*00030 ! I CONTAINS THE ASCII VALUE FOR A LETTER (65-91 DEC
34				*00031 ! OR 92 FOR ' ' , 93 FOR ' ' , 94 FOR ALL OTHER CHARS
35				*00032 !
36	0063	CE	FFC1+	FILL LDX #HIST-64
37	0066	F6	0047+	LDAB I
38	0069	8D	00B7+	JSR ZF1F
39	006C	6C	00	INC 0,X
40				*00033 IF HIST(I-64) LT 23 THEN GOTO SORT
41	006E	CE	FFC1+	LDX #HIST-64
42	0071	F6	0047+	LDAB I
43	0074	8D	00B7+	JSR ZF1F
44	0077	A6	00	LDAA 0,X
45	0079	81	17	CMPA #23
46	007B	2C	03	BGE Z00E
47	007D	7E	0021+	JMP SORT
48				*00034 CALL DSPLAY
49				*00035 !

18

OSPLAY	EXT	2/20	4/02					
FILL	00000063 REL	3/06	3/20	3/27	3/36			
GETDAT	EXT	2/20	2/43					
HIST	00000001 REL	2/11	2/14	2/29	3/36	3/41		
IOCLOS	EXT	2/20	4/27					
IOOPEN	EXT	2/20	2/24					
I	00000047 REL	2/18	2/27	2/30	2/33	3/01	3/04	
		3/30	3/37	3/42	4/08	4/11		
LETTER	00000000 REL	2/11	2/13	2/45	3/03	3/17		
MON	MAC	4/35	4/44	5/05				
MPLR10	PRG	2/01						
OUTCH	00000080 REL	4/10	4/19	4/22	4/25	4/44		
OUTSP	00000083 REL	4/05	5/04					
SORT	00000021 REL	2/43	2/48	3/47				
STACK	00000046 REL	2/16	2/22					
STOP	000000AD REL	4/29	4/29					
T	0000001E REL	2/15	6/05	6/06	6/07	6/09	6/10	
ZFFF	000000CE EQU	6/12						
ZF1F1	000000CA REL	6/08	6/10					
ZF1F2	000000BB REL	5/12	6/04					
ZF1F	000000B7 REL	2/31	3/38	3/43	5/12			
ZF5F	0000008C REL	6/05						
Z00A	00000056 REL	3/19	3/23					
Z00C	00000060 REL	3/26	3/30					
Z00E	00000080 REL	3/46	4/02					
Z00F	0000008A REL	4/08	4/15					
Z000	00000000 REL	2/22						
Z001	00000008 REL	2/27	2/37					
Z002	00000021 EQU	2/35	2/41					
Z004	0000002E REL	2/47	2/50					
Z005	00000030 REL	3/01	3/12					
Z007	0000003E REL	3/05	3/08					
Z008	00000049 REL	3/10	3/15					
Z010	0000009B REL	4/13	4/18					

END OF ASM 6800- 2.649 CP SECONDS ASSEMBLY TIME.

1
*EOR

21

1
PAGE 1 03/09/81 14.29.33. CUFOM LINKAGE EDITOR (1.1)

==0==> "" LINKED (VERSION 1.1) ON CDC/NOS/BE (03/09/81).

----- FROM "CUFIN" -----

==1==> FROM MODULE : MPLR10

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 03/09/81 14.29.19.

RELOCATABLE SECTION 3 SIZE=0010B(267) OFFSET=00000 PROGRAM

RELOCATABLE SECTION 1 SIZE=001B7(439) OFFSET=00000 DATA

DEFINED IN SECTION 1 OFFSET=00001 HIST

DEFINED IN SECTION 1 OFFSET=00000 LETTER

----- FROM LIBRARIES -----

==1==> LIBRARY "DISPLAY" BUILD (VERSION 1.1) ON CDC/NOS/BE (30/08/81)

==1==> FROM MODULE : GETDAT

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 30/08/81 12.38.59.

DEFINED IN SECTION 3 OFFSET=000CE GETDAT

==1==> FROM MODULE : IOOPEN

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 30/08/81 12.39.00.

DEFINED IN SECTION 3 OFFSET=00008 IOOPEN

DEFINED IN SECTION 1 OFFSET=00048 POINT

==1==> FROM MODULE : TEXT

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 30/08/81 12.39.01.

DEFINED IN SECTION 1 OFFSET=0004A TEXT

==1==> FROM MODULE : IOCLOS

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 30/08/81 12.39.02.

22

OFFSET=000E2 IOCLOS

CROSS ASSEMBLER FOR MOTOROLA 6800 PRO
RUN AT CERN'S CDC COMPUTER CENTRE 3

```
==1==> FROM MODULE : DSPLAY
```

1

PAGE 2 03/09/81 14.29.37. CUFOM LINKAGE EDITOR (1.1)

==1==>

```
==1==) CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR,  VERS. 1.0
```

```
==1==> RUN AT CERN'S CDC COMPUTER CENTRE      30/08/81  12.40.03.
```

DEFINED IN SECTION 3

OFFSET=000E3 DISPLAY

>>>>> 0 ERROR DETECTED IN INPUT MODULES <<<<<<

*EQR

1PAGE 1 03/09/81 14.30.03.

CUFOM -> MOTOROLA TRANSLATOR (1.1)

THE CONTROL CARD PARAMETERS ARE

OPTION	INPUT	FILE	IS-	OPTIONS
CUFOM	INPUT	FILE	IS-	CUFOM
BINARY	OUTPUT	FILE	IS-	BLETTER
LIST	OUTPUT	FILE	IS-	LLET

THE SECTION ORIGINS ARE SPECIFIED AS FOLLOWS

```
1 SECTION(3,RELOCATABLE,07PROGRAM)=0000.
```

```
2 SECTION(1,GENERAL,04DATA)=1000.
```

1PAGE 2 03/09/81 14.30.03.

LOAD MAP FOR CUFOM -> MOTOROLA TRANSLATOR (

MODULE	ENTRY	SECTION	ADDRESSES :		
NAME	NAME	TYPE AND NAME	MEMORY	ORIGIN	LISTING

PROCESSOR COMMENTS

MPLX 10

LINKED (VERSION 1.1) ON CDC/NOS/BE (

CROSS ASSEMBLER FOR MOTOROLA 6800 PRO
RUN AT CERN'S CDC COMPUTER CENTRE 0

HIST	G	DATA	001001	001000	000001
LETTER	G	DATA	001000	001000	000000

LIBRARY "DISPLAY" BUILD (VERSION 1.1)

CROSS ASSEMBLER FOR MOTOROLA 6800 PRO
RUN AT CERN'S CDC COMPUTER CENTRE 3

```
GETDAT      R  PROGRAM  0000CE  0000CE  000000
```

IDOPEN

TEXT

IOOPEN	R	PROGRAM	0000DB	0000DB	000000
POINT	G	DATA	001048	001048	000000

DOCLOS

TEXT G DATA 00104A 00104A 000000

DISPLAY

IOCLOS R PROGRAM 0000E2 0000E2 000000

```

DSPLAY      R  PROGRAM  0000E3  0000E3  000000

```

NO TRANSFER ADDRESS SPECIFIED FOR THIS MODULE

))))> 0 ERROR DETECTED IN CUFOM INPUT FILE <<<<<<<

ICTP COLLEGE ON MICROPROCESSORS

(TRIESTE, SEPTEMBER 1981)

PRACTICAL WORK SESSIONS

GOAL:

Exercise on the MPL Compiler
using Assembler subroutines.

PROBLEM DESCRIPTION:

Write a MPL program which fills a histogram ("HIST"). This mainprogram should make a statistical analysis of letters in a text. The occurrence of the letters "A....Z" the "." and the "," should be counted, "A" corresponding to HIST(1), "B" to HIST(2), ..., "." to HIST(27) and "," to HIST(28). If a blank is found in the text it should be simply skipped. All other special characters like: ";,%,?!,,: etc. should be counted in an extra channel HIST(29). Stop counting if a channel reaches the count of 23 (decimal!).

The MPL program should call the following 4 assembler subroutines:

- 1) "IOOPEN" Initializes a pointer "POINT" to the text.
This pointer will point to the first letter in
the text after IOOPEN has been called.
- 2) "GETDAT" Takes the letter pointed to by POINT and stores
it into memory location "LETTER" where it can be
found by the mainroutine. GETDAT then increments
the pointer.
- 3) "IOCLOS" Does nothing! This routine consists of only a RTS
- 4) "DSPLAY" This routine was written in a previous
exercise in assembly language.

You may collect all these programs in a library "DISLIB" which you link to your MPL main program.

The MPL program will call IOOPEN to initialize POINT and it will then fill the histogram using a loop. In this loop a call to GETDAT will transfer the next letter to the main routine. Do this using external references. LETTER will be defined in the MPL program. As soon as the histogram is full call DSPLAY to print it to the screen. Print the letters "A", ..., "Z" as well as "." and "," under the corresponding channels. A "\$" may be used for channel 29. IOCLOS will terminate our little "experiment".

ICTP COLLEGE ON MICROPROCESSORS

(TRIESTE, SEPTEMBER 1981)

PRACTICAL WORK SESSIONS

Use the following text for the analysis:

THIS PROGRAM WILL SHOW YOU HOW A PHYSICS EXPERIMENT CAN BE PROGRAMMED. NORMALLY THE FOLLOWING TASKS HAVE TO BE PERFORMED: ALL ELECTRONICS HAS TO BE INITIALIZED (IOOPEN). THEN THE DATA HAVE TO BE TAKEN (GETDAT) AND ON-LINE SPECTRA HAVE TO BE DISPLAYED ON A SCREEN (DSPLAY). NORMALLY THE INPUT RAW DATA ARE ALSO WRITTEN ONTO MAGNETIC TAPE. AT THE END OF THE EXPERIMENT THE ON-LINE SPECTRA MAY BE SAVED ON DISK OR MAGNETIC TAPE (IOCLOS).

Translate this text using the FCC assembler directive and store it into memory where your analysis program can find it.

ICTP COLLEGE ON MICROPROCESSORS

(T R I E S T E , S E P T E M B E R 1981)

PRACTICAL WORK SESSIONS

The whole spectrum will look like this:

[illegible]

This spectrum was produced by the solution program and transferred to the CDC using up-line loading

27

ICTP COLLEGE ON MICROPROCESSORS

(T R I E S T E , S E P T E M B E R 1981)

PRACTICAL WORK SESSIONS

First I give you the MPL source of the main program. In the MPL code I have to use assembly language statements to output the characters "A", "...", "Z" and "...". This I accomplish with macros collected in a small macro library MONMPL which is inserted into the MPL program using the assembler directive INSERT. All the assembly language subroutines (GETDAT, IOOPEN, DISPLAY, TEXT, IOCLOSE) are translated separately and put together into a Cufom library DISLIB using the librarian. The command:

```
BEGIN,MICMPL,,S=NLET,B=BLET,L=LLET,ALIB=MONMPL,CLIB=DISLIB
```

first compiles the program NLET, then it inserts the file MONMPL into the assembly language code (output from the compiler) then it links the cucom library DISLIB to the output file from the assembler and in the end it translates the complete program into Motorola S-format. BLET is the program ready for down-line loading. The assembler listings of the programs :

GETDAT, IOOPEN, IOCLOS, TEXT

are given below as well as the listing LLET of the complete translation process of the MPL program NLET.

```

$ OPT REL,MEX
$ DSCT
$ XDEF HIST,LETTER
DECLARE
LETTER,HIST(29),I(40),STACK,I
$ PSCT
$ INSERT
$ XREF GETDAT,IOOPEN,DSPLAY,IOCLOS
PROCEDURE OPTIONS(MAIN,STACK)
CALL IOOPEN
DO I=1 TO 29 !CLEAR HISTOGRAM
HIST(I)=0
END
!
! SORT: CALL GETDAT !GET DATA INTO LETTER
!
IF LETTER EQ "20" THEN GOTO SORT !SKIP BLANKS
DO I= 'A' TO 'Z'
IF LETTER EQ I THEN GOTO FILL !IS DATA A LETTER (A-Z)?
END
!
I=I+1
IF LETTER EQ '.' THEN GOTO FILL ! IS IT A '.'?
!
I=I+1
IF LETTER EQ ',' THEN GOTO FILL !IS IT A ','?
!
I=I+1
FILL: HIST(I-64)=HIST(I-64)+1 !FILL HISTOGRAM
!
! I CONTAINS THE ASCII VALUE FOR A LETTER (65-91 DECIMAL)
! OR 92 FOR '.', 93 FOR ',', 94 FOR ALL OTHER CHARACTERS.
```

28

```

!
    IF NIST(I-64) LT 23 THEN GOTO SORT
    CALL DISPLAY
!
!OUTPUT "A"....."Z", ".", ",", "$"
!
$    OUTSP 25                      !OUTPUT 25 BLANKS
    DO I='A' TO 'Z'
$    OUTCH I
    END
$    OUTCHI '
$    OUTCHI '
$    OUTCHI '$
    CALL IOCLOS
STOP: GOTO STOP
END

```

```

*****
*    MACRO TO PERFORM MONITOR CALLS    *
*    MON X WILL CALL MONITOR ROUTINE NR. X *
*****
MON    MACR
      SWI
      FCB  \0
      ENDM
*****
*    MACRO TO OUTPUT A CERTAIN NUMBER OF SPACES *
*    OUTSP X WILL OUTPUT X(DEC) SPACES *
*****
OUTSP  MACR
      LDAB  \0
      MON  10
      ENDM
*****
*    MACRO TO OUTPUT A CHARACTER *
*    OUTCH VAR WILL OUTPUT THE CHARACTER IN *
*    IN MEMORY LOCATION VAR *
*****
OUTCH  MACR
      LDAA  \0
      MON  6
      ENDM
*****
*    MACRO TO OUTPUT A CHARACTER *
*    OUTCHI CHAR WILL OUTPUT THE ASCII CHARACTER *
*    CHAR *
*    E.G. OUTCHI 'A' OUTPUTS ASCII "A" *
*****
OUTCHI MACR
      LDAA  \0
      MON  6
      ENDM

```

29

```

1
CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS
0
0
0    NAME OF MODULE : GETDAT
0    PROGRAM TYPE   : SUBPROGRAM
0    EXTERNAL REFER.: LETTER  POINT

```

```

0    S E C T I O N S:
-----
0    PROGRAM SECTION      : 000C IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
                           GETDAT

```

1 CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

LINE	ADDR	CODE	EXTENSIONS	SOURCE	STATEMENT
1	0000			NAM	GETDAT
2	0000			OPT	REL
3				*	THIS SUBROUTINE GETS ASCII CHARACTERS FROM
4				*	A TEXTFILE AND STORES THEM IN MEMORY LOCATION LETTER
5				*	
6	0000			PSCT	
7				XREF	LETTER, POINT
8				XDEF	GETDAT
9	0000	FE	0000*	GETDAT LD	POINT
10	0003	A6	00	LDAA	0,X
11	0005	08		INX	
12	0006	FF	0000*	STX	POINT
13	0009	B7	0000*	STAA	LETTER
14	000C	39		RTS	
15	000D			END	

1 IDENTIFIER CROSS REFERENCE. CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

IDENTIFIER	ADDRESS	CODE	EXTENSIONS	2/01	2/08	2/09
GETDAT	00000000	REL		2/01	2/08	2/09
LETTER		EXT		2/07	2/13	
POINT		EXT		2/07	2/09	2/12

END OF ASM 6800- 0.353 CP SECONDS ASSEMBLY TIME.

30

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

```
0 DATA SECTION : 0001 IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
                  POINT
```

CROSS ASSEMBLER FOR MOTOROLA 68000
VERS. 1.0, RUN AT CERN'S CDC COM

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
1	0000	0000		
2	0001	0000		
3	0002	0000		
4	0003	0000		
5	0004	0000		
6	0005	0000		
7	0006	0000		
8	0007	0000		
9	0008	0000		
10	0009	0000		
11	0010	0000		
12	0011	0000		
13	0012	0000		
14	0013	0000		
15	0014	0000		
16	0015	0000		
17	0016	0000		
18	0017	0000		
19	0018	0000		
20	0019	0000		
21	0020	0000		
22	0021	0000		
23	0022	0000		
24	0023	0000		
25	0024	0000		
26	0025	0000		
27	0026	0000		
28	0027	0000		
29	0028	0000		
30	0029	0000		
31	0030	0000		
32	0031	0000		
33	0032	0000		
34	0033	0000		
35	0034	0000		
36	0035	0000		
37	0036	0000		
38	0037	0000		
39	0038	0000		
40	0039	0000		
41	0040	0000		
42	0041	0000		
43	0042	0000		
44	0043	0000		
45	0044	0000		
46	0045	0000		
47	0046	0000		
48	0047	0000		
49	0048	0000		
50	0049	0000		
51	0050	0000		
52	0051	0000		
53	0052	0000		
54	0053	0000		
55	0054	0000		
56	0055	0000		
57	0056	0000		
58	0057	0000		
59	0058	0000		
60	0059	0000		
61	0060	0000		
62	0061	0000		
63	0062	0000		
64	0063	0000		
65	0064	0000		
66	0065	0000		
67	0066	0000		
68	0067	0000		
69	0068	0000		
70	0069	0000		
71	0070	0000		
72	0071	0000		
73	0072	0000		
74	0073	0000		
75	0074	0000		
76	0075	0000		
77	0076	0000		
78	0077	0000		
79	0078	0000		
80	0079	0000		
81	0080	0000		
82	0081	0000		
83	0082	0000		
84	0083	0000		
85	0084	0000		
86	0085	0000		
87	0086	0000		
88	0087	0000		
89	0088	0000		
90	0089	0000		</

1					*		
2	0000					NAM	100PEN
3	0000					OPT	REL
4	0000					DSCT	
5	0000	02			POINT	RMB	2
6						XDEF	POINT
7	0000					PSCT	
8						XDEF	100PEN
9						XREF	TEXT
10	0000	CE	0000*		100PEN	L0X	*TEXT
11	0003	FF	0000+			STX	POINT
12	0006	39				RTS	
13	0007					END	

1
IDENTIFIER CROSS REFERENCE.

100PEN	00000000	REL	2/02	2/08	2/10
POINT	00000000	REL	2/05	2/06	2/11
TEXT		EXT	2/09	2/10	

0END OF ASM 6800- 0.320 CP SECONDS ASSEMBLY TIME.

1

CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS

```
0      DATA SECTION      : 016C IS HIGHEST ADDRESS USED,  ENTRY POINTS ARE:
                                TEXT
```

1 CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
000000	000000	000000	000000	000000
000001	000001	000001	000001	000001
000002	000002	000002	000002	000002
000003	000003	000003	000003	000003
000004	000004	000004	000004	000004
000005	000005	000005	000005	000005
000006	000006	000006	000006	000006
000007	000007	000007	000007	000007
000008	000008	000008	000008	000008
000009	000009	000009	000009	000009
000010	000010	000010	000010	000010
000011	000011	000011	000011	000011
000012	000012	000012	000012	000012
000013	000013	000013	000013	000013
000014	000014	000014	000014	000014
000015	000015	000015	000015	000015
000016	000016	000016	000016	000016
000017	000017	000017	000017	000017
000018	000018	000018	000018	000018
000019	000019	000019	000019	000019
000020	000020	000020	000020	000020
000021	000021	000021	000021	000021
000022	000022	000022	000022	000022
000023	000023	000023	000023	000023
000024	000024	000024	000024	000024
000025	000025	000025	000025	000025
000026	000026	000026	000026	000026
000027	000027	000027	000027	000027
000028	000028	000028	000028	000028
000029	000029	000029	000029	000029
000030	000030	000030	000030	000030
000031	000031	000031	000031	000031
000032	000032	000032	000032	000032
000033	000033	000033	000033	000033
000034	000034	000034	000034	000034
000035	000035	000035	000035	000035
000036	000036	000036	000036	000036
000037	000037	000037	000037	000037
000038	000038	000038	000038	000038
000039	000039	000039	000039	000039
000040	000040	000040	000040	000040
000041	000041	000041	000041	000041
000042	000042	000042	000042	000042
000043	000043	000043	000043	000043
000044	000044	000044	000044	000044
000045	000045	000045	000045	000045
000046	000046	000046	000046	000046
000047	000047	000047	000047	000047
000048	000048	000048	000048	000048
000049	000049	000049	000049	000049
000050	000050	000050	000050	000050
000051	000051	000051	000051	000051
000052	000052	000052	000052	000052
000053	000053	000053	000053	000053
000054	000054	000054	000054	000054
000055	000055	000055	000055	000055
000056	000056	000056	000056	000056
000057	000057	000057	000057	000057
000058	000058	000058	000058	000058
000059	000059	000059	000059	000059
000060				

```

1          *
2 0000          NAM TEXT
3 0000          DSCT
4          XDEF      TEXT
5 0000 54 48 49 53 20 50  TEXT FCC /THIS PROGRAM WILL SHOW YOU HOW A PHYSICS EXPERIMENT/
6 0033 43 41 4E 20 42 45  FCC /CAN BE PROGRAMMED.NORMALLY THE FOLLOWING TASKS HAVE/
7 0066 54 4F 20 42 45 20  FCC /TO BE PERFORMED: ALL ELECTRONICS HAS TO BE INITIALIZED
8 009D 45 2E 47 2E 20 43  FCC /E.G. CAMAC CLEAR (IOOPEN). THEN THE DATA HAVE TO BE/
9 0000 54 41 4B 45 4E 20  FCC /TAKEN (GETBAT) AND ONLINE SPECTRA HAVE TO BE GENERATED
10 0107 54 48 45 53 45 20  FCC /THESE SPETRA HAVE TO BE DISPLAYED ON A SCREEN (DSPLY).
11 013D 4E 4F 52 4D 41 4C  FCC /NORMALLY THE DATA ARE ALSO WRITTEN ONTO MAGTAPE./
12 016D          END

```

1 IDENTIFIER CROSS REFERENCE. CROSS ASSEMBLER FOR MOTOROLA 68000
VERS. 1.0, RUN AT CERN'S CDC COM

TEXT 00000000 REL 2/02 2/04 2/05

END OF ASM 6800- 0.743 CP SECONDS ASSEMBLY TIME.

1

```

1
CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS
CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

```

CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS

```
0
0      NAME OF MODULE : IOCLOS
```

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

0 PROGRAM TYPE : SUBPROGRAM
0 EXTERNAL REFER.: -NONE-
0 S E C T I O N S:

CROSS ASSEMBLER FOR MOTOROLA 6
VERS. 1.0, RUN AT CERN'S CDC C

0 PROGRAM SECTION : 0000 IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
IOCLOS

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC C

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
1				*
2	0000			NAM IOCLOS
3	0000			PSCT
4				XDEF IOCLOS
5	0000	39		IOCLOS RTS
6	0001			END

1 IDENTIFIER CROSS REFERENCE.

IOCLOS	00000000	REL	2/02	2/04	2/05

0 END OF ASM 6800- 0.209 CP SECONDS ASSEMBLY TIME.

1 CUFOM MODULE NAME, ENTRY POINTS, EXTERNAL SYMBOLS

0 NAME OF MODULE : MPLR10
0 PROGRAM TYPE : SUBPROGRAM
0 EXTERNAL REFER.: DSPLAY GETBAT IOOPEN IOCLOS

0 S E C T I O N S:

0 PROGRAM SECTION : 0004 IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
-NONE-

0 DATA SECTION : 0047 IS HIGHEST ADDRESS USED, ENTRY POINTS ARE:
HIST LETTER

33

LINE ADDR CODE EXTENSIONS SOURCE - STATEMENT

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
1	0000			NAM MPLR10
2				*MOTOROLA SPD, INC. OWNS AND IS RESPONSIBLE FOR M68MPL
3				* COPYRIGHT 1975 AND 1976 BY MOTOROLA INC.
4				*
5				* MOTOROLA'S M6800 MPL COMPILER, RELEASE 1.2
6				*
7				*** COMPILED WITH MPL VERSION 1.2
8				OPT REL,MEX
9	0000			DSCT
10	0000			XDEF HIST,LETTER
11				*00004 DECLARE
12				LETTER RMB 1
13	0000	01		HIST RMB 29
14	0001	10		T RMB 40
15	001E	28		STACK RMB 1
16	0046	01		*00005 LETTER,HIST(29),T(40),STACK,1
17				I RMB 1
18	0047	01		PSCT
19	0000			INSERT
20				*****
21				* MACRO TO PERFORM MONITOR CALLS *
22				* MON X WILL CALL MONITOR ROUTINE NR. X *
23				*****
24				MON MACR
25				SWI
26				FCB ~0
27				ENDM
28				*****
29				* MACRO TO OUTPUT A CERTAIN NUMBER OF SPACES *
30				* OUTSP X WILL OUTPUT X(DEC) SPACES *
31				*****
32				OUTSP MACR
33				LDAB ~0
34				MON 10
35				ENDM
36				*****
37				* MACRO TO OUTPUT A CHARACTER *
38				* OUTCH VAR WILL OUTPUT THE CHARACTER IN *
39				* IN MEMORY LOCATION VAR *
40				*****
41				OUTCH MACR
42				LDAA ~0
43				MON 6
44				ENDM
45				*****
46				* MACRO TO OUTPUT A CHARACTER *
47				* OUTCHI CHAR WILL OUTPUT THE ASCII CHARACTER *
48				CHAR
49				*

34

* E.G. OUTCHI 'A' OUTPUTS ASCII "A" *

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC CO:

LINE ADDR CODE EXTENSIONS

SOURCE - STATEMENT

```

1
2
3
4
5
6
7
8
9 0000 8E 0046+
10
11 0003 8D 0000*
12
13 0006 86 01
14 0008 87 0047+
15
16 000B CE 0001+
17 000E F6 0047+
18 0011 BD 00AE+
19 0014 6F 00
20 0016 B6 0047+
21 0019 81 10
22 001B 24 04
23 001D 4C
24 001E 7E 0008+
25
26
27
28 0021+
29
30 0021 BD 0000*
31
32 0024 B6 0000+
33 0027 81 20
34 0029 26 03
35 002B 7E 0021+
36
37 002E 86 41
38 0030 87 0047+
39
40 0033 B6 0000+
41 0036 B1 0047+
42 0039 26 03
43 003B 7E 0063+
44

*****
OUTCHI MACR
  LDAA #0
  MON 6
  ENDM

XREF GETDAT, IOOPEN, DSPLAY, IOCLOS
*00009 PROCEDURE OPTIONS(MAIN, STACK)
Z000 LDS #STACK
*00010 CALL IOOPEN
JSR IOOPEN
*00011 DO$1=1 TO 29
  LDAA #1
  STAA I
  *00012 HIST(I)=0
  LDX #HIST
  LDAB I
  JSR ZF1F
  CLR 0,X
  LDAA I
  CMPA #29
  BCC Z002
  INCA
  JMP Z001
*00013 END
*00014 !
*00015 SORT: CALL GETDAT
Z002 EQU *
*00016 !
SORT JSR GETDAT
*00017 IF LETTER EQ "20" THEN GOTO SORT
  LDAA LETTER
  CMPA #32
  BNE Z004
  JMP SORT
*00018 DO$I= 'A' TO 'Z'
Z004 LDAA #'A
Z005 STAA I
*00019 IF LETTER EQ I THEN GOTO FILL
  LDAA LETTER
  CMPA I
  BNE Z007
  JMP FILL
*00020 END
  
```

35

```

45 003E 86 0047+
46 0041 81 5A
47 0043 24 04
48 0045 4C
49 0046 7E 0030+
50
  
```

```

Z007 LDAA I
      CMPA #Z
      BCC Z008
      INCA
      JMP Z005
*00021 !
  
```

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC CO:

LINE ADDR CODE EXTENSIONS

SOURCE - STATEMENT

```

1
2 0049 7C 0047+
3
4 004C B6 0000+
5 004F 81 2E
6 0051 26 03
7 0053 7E 0063+
8
9
10 0056 7C 0047+
11
12 0059 81 2C
13 005B 26 03
14 005D 7E 0063+
15
16
17 0060 7C 0047+
18
19
20
21
22
23 0063 CE FFC1+
24 0066 F6 0047+
25 0069 BD 00AE+
26 006C 6C 00
27
28 006E CE FFC1+
29 0071 F6 0047+
30 0074 BD 00AE+
31 0077 A6 00
32 0079 81 17
33 007B 2C 03
34 007D 7E 0021+
35
36
37
38
39 0080 BD 0000*

*00022 I=I+1
Z008 INC I
*00023 IF LETTER EQ '.' THEN GOTO FILL ! IS IT A '.'
  LDAA LETTER
  CMPA #'.
  BNE Z00A
  JMP FILL
*00024 !
*00025 I=I+1
Z00A INC I
*00026 IF LETTER EQ ',' THEN GOTO FILL ! IS IT A ','
  CMPA #','
  BNE Z00C
  JMP FILL
*00027 !
*00028 I=I+1
Z00C INC I
*00029 FILL: HIST(I-64)=HIST(I-64)+1 ! FILL HISTOGR
*00030 !
*00031 ! I CONTAINS THE ASCII VALUE FOR A LETTER (65-91 DEC)
*00032 ! OR 92 FOR '.', 93 FOR ',', 94 FOR ALL OTHER CHARAC
*00033 !
FILL LDX #HIST-64
      LDAB I
      JSR ZF1F
      INC 0,X
*00034 IF HIST(I-64) LT 23 THEN GOTO SORT
      LDX #HIST-64
      LDAB I
      JSR ZF1F
      LDAA 0,X
      CMPA #23
      BGE Z00E
      JMP SORT
*00035 CALL DSPLAY
*00036 !
*00037 ! OUTPUT "A"....."Z", ".", ",", "5"
*00038 !
Z00E JSR DSPLAY
  
```

36

```

40      OUTSP 25
41 0083 C6 19      LDAB #25
42      MON 10
43 0085 3F      SWI
44 0086 0A      FCB 10
45      ENDM
46      ENDM
47      *00040      DO$1='A' TO 'Z'
48 0087 86 41      LDAA #'A
49 0089 B7 0047+    ZOOF STAA I
50      OUTCH I

```

!OUTPUT 25 BLANKS

```

35      *00046      CALL IOCLOS
36 00A8 B0 0000*    JSR IOCLOS
37      *00047 STOP: GOTO STOP
38 00AB 7E 00AB+    STOP JMP STOP
39      *00048      END

```

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

1

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
1	008C	B6	0047+	LDAA I
2				MON 6
3	008F	3F		SWI
4	0090	06		FCB 6
5				ENDM
6				ENDM
7				*00042 END
8	0091	86	0047+	LDAA I
9	0094	81	5A	CMPI #'Z
10	0096	24	04	BCC Z010
11	0098	4C		INCA
12	0099	7E	0089+	JMP ZOOF
13	009C+			Z010 EQU *
14				OUTCHI
15	009C	86	2E	LDAA #'.
16				MON 6
17	009E	3F		SWI
18	009F	06		FCB 6
19				ENDM
20				ENDM
21				OUTCHI
22	00A0	86	2C	LDAA #'.
23				MON 6
24	00A2	3F		SWI
25	00A3	06		FCB 6
26				ENDM
27				ENDM
28				OUTCHI
29	00A4	86	24	LDAA #'\$
30				MON 6
31	00A6	3F		SWI
32	00A7	06		FCB 6
33				ENDM
34				ENDM

37

LINE	ADDR	CODE	EXTENSIONS	SOURCE - STATEMENT
1	00AE	26	02	ZF1F BNE ZF1F2 *** ADD TO INDEX ***
2	00B0	09		DEX
3	00B1	39		RTS
4	00B2	5A		ZF1F2 DECB
5	00B3	FF	0024+	ZF5F STX T+6
6	00B6	FB	0025+	ADDB T+7
7	00B9	F7	0025+	STAB T+7
8	00BC	24	03	BCC ZF1F1
9	00BE	7C	0024+	INC T+6
10	00C1	FE	0024+	ZF1F1 LDX T+6
11	00C4	39		RTS
12		00C5+		ZFFF EQU *
13	00C5			END

1

CROSS ASSEMBLER FOR MOTOROLA 680
VERS. 1.0, RUN AT CERN'S CDC COM

IDENTIFIER CROSS REFERENCE.

IDENTIFIER	TYPE	ADDRESS	EXTENSION	STATEMENT	ADDRESS	EXTENSION	STATEMENT	ADDRESS	EXTENSION	STATEMENT
DISPLAY	EXT	3/07		4/39						
FILL	00000063 REL	3/43		4/07	4/14		4/23			
GETDAT	EXT	3/07		3/30						
HIST	00000001 REL	2/11		2/14	3/16		4/23	4/28		
IOCLOS	EXT	3/07		5/36						
IOOPEN	EXT	3/07		3/11						
I	00000047 REL	2/18		3/14	3/17		3/20	3/38	3/41	
		4/17		4/24	4/29		4/49	4/50	5/01	
LETTER	00000000 REL	2/11		2/13	3/32		3/40	4/04		
MON	MAC	2/25		4/42	5/02		5/16	5/23	5/30	
MPLR10	PRG	2/01								
OUTCHI	MAC	3/02		5/14	5/21		5/28			
OUTCH	MAC	2/42		4/50						
OUTSP	MAC	2/33		4/40						
SORT	00000021 REL	3/30		3/35	4/34					
STACK	00000046 REL	2/16		3/09						
STOP	000000AB REL	5/38		5/38						
T	0000001E REL	2/15		6/05	6/06		6/07	6/09	6/10	
ZFFF	000000C5 EQU	6/12								
ZF1F1	000000C1 REL	6/08		6/10						

38

ZF1F2	000000B2	REL	5/40	6/04		
ZF1F	000000AE	REL	3/18	4/25	4/30	5/40
ZF5F	000000B3	REL	6/05			
Z00A	00000056	REL	4/06	4/10		
Z00C	00000060	REL	4/13	4/17		
Z00E	00000080	REL	4/33	4/39		
Z00F	00000089	REL	4/49	5/12		
Z000	00000000	REL	3/09			
Z001	00000008	REL	3/14	3/24		
Z002	00000021	EQU	3/22	3/28		
Z004	0000002E	REL	3/34	3/37		
Z005	00000030	REL	3/38	3/49		
Z007	0000003E	REL	3/42	3/45		
Z008	00000049	REL	3/47	4/02		
Z010	0000009C	EQU	5/10	5/13		

END OF ASM 6800- 2.949 CP SECONDS ASSEMBLY TIME.

↑

*EOR

39

1
PAGE 1 01/09/81 09.43.58. CUFOM LINKAGE EDITOR (1.1)

==0==> "" LINKED (VERSION 1.1) ON CDC/NOS/BE (01/09/81).

----- FROM "CUFIN" -----

==1==> FROM MODULE : MPLR10

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 01/09/81 09.43.49.

RELOCATABLE SECTION 3 SIZE=00102(258) OFFSET=00000 PROGRAM

RELOCATABLE SECTION 1 SIZE=001B7(439) OFFSET=00000 DATA

DEFINED IN SECTION 1 OFFSET=00001 HIST

DEFINED IN SECTION 1 OFFSET=00000 LETTER

----- FROM LIBRARIES -----

==1==> LIBRARY "DISPLAY" BUILD (VERSION 1.1) ON CDC/NOS/BE (30/08/81)

==1==> FROM MODULE : GETDAT

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 30/08/81 12.38.59.

DEFINED IN SECTION 3 OFFSET=00005 GETDAT

==1==> FROM MODULE : IOOPEN

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 30/08/81 12.39.00.

DEFINED IN SECTION 3 OFFSET=00002 IOOPEN

DEFINED IN SECTION 1 OFFSET=00048 POINT

==1==> FROM MODULE : TEXT

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 30/08/81 12.39.01.

DEFINED IN SECTION 1 OFFSET=0004A TEXT

==1==> FROM MODULE : IOCLOS

==1==>

==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0

==1==> RUN AT CERN'S CDC COMPUTER CENTRE 30/08/81 12.39.02.

DEFINED IN SECTION 3 OFFSET=00009 IOCLOS

40

==1==> FROM MODULE : DSPLAY

1
PAGE 2 01/09/81 09.44.05. CUFOM LINKAGE EDITOR (1.1)

==1==>
==1==> CROSS ASSEMBLER FOR MOTOROLA 6800 PROCESSOR, VERS. 1.0
==1==> RUN AT CERN'S CDC COMPUTER CENTRE 30/08/81 12.40.03.
DEFINED IN SECTION 3 OFFSET=000DA DSPLAY

>>>>> 0 ERROR DETECTED IN INPUT MODULES <<<<<<<<
*EOR

1PAGE 1 01/09/81 09.44.33.

CUFOM -> MOTOROLA TRANSLATOR (1.1)

THE CONTROL CARD PARAMETERS ARE

OPTION INPUT FILE IS- OPTIONS
CUFOM INPUT FILE IS- CUFOM
BINARY OUTPUT FILE IS- BLET
LIST OUTPUT FILE IS- LDIS

THE SECTION ORIGINS ARE SPECIFIED AS FOLLOWS

1 SECTION(3,RELOCATABLE,07PROGRAM)=0000.
2 SECTION(1,GENERAL,04DATA)=1000.
1PAGE 2 01/09/81 09.44.33.

LOAD MAP FOR CUFOM -> MOTOROLA TRANSLATOR (1

MODULE NAME	ENTRY NAME	SECTION TYPE AND NAME	MEMORY	ADDRESSES : ORIGIN LISTING	PROCESSOR COMMENTS
MPLR10					LINKED (VERSION 1.1) ON CDC/NOS/BE (0
					CROSS ASSEMBLER FOR MOTOROLA 6800 PROC
					RUN AT CERN'S CDC COMPUTER CENTRE 01
GETDAT	HIST LETTER	G DATA G DATA	001001 001000	001000 001000	000001 000000
					LIBRARY "DISPLAY" BUILD (VERSION 1.1)
					CROSS ASSEMBLER FOR MOTOROLA 6800 PROC
					RUN AT CERN'S CDC COMPUTER CENTRE 30
IOOPEN	GETDAT	R PROGRAM	0000C5	0000C5	000000
					CROSS ASSEMBLER FOR MOTOROLA 6800 PROC
					RUN AT CERN'S CDC COMPUTER CENTRE 30
					CROSS ASSEMBLER FOR MOTOROLA 6800 PROC
					RUN AT CERN'S CDC COMPUTER CENTRE 30
TEXT	IOOPEN POINT	R PROGRAM G DATA	0000D2 001048	0000D2 001048	000000 000000
					CROSS ASSEMBLER FOR MOTOROLA 6800 PROC
					RUN AT CERN'S CDC COMPUTER CENTRE 30
IOCLOS	TEXT	G DATA	00104A	00104A	000000
					CROSS ASSEMBLER FOR MOTOROLA 6800 PROC
					RUN AT CERN'S CDC COMPUTER CENTRE 30
DSPLAY	IOCLOS	R PROGRAM	0000D9	0000D9	000000
					CROSS ASSEMBLER FOR MOTOROLA 6800 PROC
					RUN AT CERN'S CDC COMPUTER CENTRE 30
					CROSS ASSEMBLER FOR MOTOROLA 6800 PROC
					RUN AT CERN'S CDC COMPUTER CENTRE 30
	DSPLAY	R PROGRAM	0000DA	0000DA	000000

NO TRANSFER ADDRESS SPECIFIED FOR THIS MODULE

>>>>> 0 ERROR DETECTED IN CUFOM INPUT FILE <<<<<<<