

PHILIPS

P855M

System Software Reference Data

November 1973

A publication of:

Philips-Electrologica B.V.
Main Marketing Group OEM
P.O. BOX 245 Apeldoorn, The Netherlands

Copyright © by Philips-Electrologica B.V.

5122 991 12351

Printed in The Netherlands



**Data
Systems**

CONTENTS

Page

- Bootstraps 2
- Standard file codes - basic 2
- Standard file codes - disc 2
- Standard device addresses 2
- Monitor operator messages 3
 - BOM - DOM 3
 - BRTM 3
 - DRTM 4
- CCI commands 4
 - Processor calls 4
 - Magnetic tape and cassette tape commands 6
- SCL commands 7
- Disc processor messages 7
 - Update 7
 - Debugging 8
- Basic processor messages 8
 - Assembler 8
 - Linkage editor 9
 - Update 9
 - Debugging 10
- Instructions 11
 - Memory reference instructions 13
 - Register-to-register instructions 14
 - Constant instructions 15
 - Shift instructions 16
 - I/O instructions 18
 - Miscellaneous instructions 18
- Directives 19
- Monitor requests 20
- I/O functions 21
- Tape handling orders 21
- Software status word 21
- Control unit status word configuration 22
- Data communication 23
 - I/O requests DATEM 23
 - I/O requests BSC 23
 - Line control table 23
 - OF Get line type 24
 - DATEM ECB 24
 - BSC ECB 25
- System messages 26
- Standard priority interrupt levels 35
- Standard connections interrupt line 35
- Program status word 36
- Memory protection 36
- Multiplex control words 37
- Fixed memory locations 37
- Powers of 16 37
- Powers of 2 37
- Hexadecimal-decimal conversion 39
- ASCII code 40

DISC BOOTSTRAP

0000	20BF	BBC0	0032	911C	921C	3F42	220F	1778
0010	BB3D	0503	8420	001A	8F1C	050C	8420	0022
0020	8F1C	8340	003C	9B20	0198	812C	5C2E	1304
0030	8F0C	45C0	4DC0	5C04	8F10	4E66	3E00	0002

P8 BOOTSTRAP FOR PTR

0000	20BF	43E0	4A20	5C04	227F	5808	0100	0402
0010	0602	E508	4A20	5C04	1E01	590A	9508	1C01
0020	5006	520C	8714	5F18	8314	1502	8094	5F20
0030	B114	1302	852D	1F01	5C2A	8F02	0000	0002

P4 BOOTSTRAP FOR ASR

0000	20BF	42D0	4A10	5C04	227F	5808	8320	FFFC
0010	0604	4A10	5C04	220F	3D41	3D41	3D41	3D41
0020	9508	1E01	5C14	1302	5006	5108	8717	5F20
0030	8314	5F24	B114	852D	1F01	5C2C	4590	207F

Standard file codes - basic

/01 = standard source input
 /02 = standard listing output
 /03 = standard punch output
 /04 = standard object input
 /05 = operator's keyboard

Standard file codes - disc

/01 = operator's typewriter
 /02 = listing output
 /03 = punch output
 /D0 = catalogued procedure input
 /D3 = reserved for system use
 /D4 = /S, or library source file (Update output)
 /D5 = /O (ASM output, LKE input)
 /D6 = /L (LKE output)
 /D7 = system object file (library)
 /D8 = user object file (library)
 /D9 - DA = reserved for system use
 /E0 = control command input
 /E1 = source input
 /E2 = object input
 /EE = catalogued procedure output
 /EF = system typewriter
 /F0 - /FF = disc units

Standard device addresses

ASR1	=	10		
PTR1	=	20		
PTP1	=	30		
FHD1	=	01	} one CU	(MX chan or DMAC)
FHD2	=	11		
FHD3	=	21		
FHD4	=	31		

MHD1	=	02	} one CU	(MX chan or DMAC)
MHD2	=	12		
MHD3	=	03		
MHD4	=	13		
PTR2	=	22	} one CU	(MX chan only)
PTP2	=	32		
MT1	=	04		
MT2	=	14		
MT3	=	24	} one CU	(MX chan only)
MT4	=	34		
CR	=	05		
ASR2	=	15		
TC1	=	06	} one CU	(MX chan only)
TC2	=	16		
TC3	=	26		
TC4	=	36		
LP	=	0D		
PL	=	0E		
Adapter	=	0F		
DMAC	=	1F		
MIDB	=	2F		
RTC	=	3F		

MONITOR OPERATOR MESSAGES**BOM - DOM**

Message	Meaning
AB	Abort a program
AS<u>file code</u><u>device name address</u>	Assign a file code
DM<u>address1</u><u>address2</u>	Dump memory
HD	Halt dump
LD<u>[<value>]</u><u>[<u>M</u></u>	Load a program (BOM only)
MC<u>file code</u><u>order</u><u>rept fact</u>	Manual device control
PS	Pause
RD<u>device address</u>	Release an operation from device
RS<u>[<new A7>]</u>	Restart a program
RY<u>device address</u>	Retry an I/O operation
ST	Start a program (BOM only)
WM<u>address</u><u>value1</u><u>[<u>value2</u>..<u>value n</u>]</u>	Write into memory
BRTM	
LD	Load a program
ST	Start a program
WM<u>address</u><u>value1</u><u>[<u>value2</u>..<u>value n</u>]</u>	Write into memory

Message	Meaning
DRTM	
CC	Activate SCL
CR<file code>	enter correction records from this file code. Format correction record: <address>,<value1>,<value2>.. ...<value n>]
DD<disc number>,<sector1>,<sector2>]	Dump disc
DM<address1>,<address2>]	Dump memory
HD	Halt dump
HT	Stop CPU activity
RD<device address>	Release I/O operation
RY<device address>	Retry I/O operation
WM<address>,<value1>,<value2>..<value n>]	Write memory

CCI COMMANDS

syntax:	ASG<file code1>,<file code2> <device name>][,<name>][,<userid>,<file code3>]] Assign a file code to a peripheral unit, a disc file or a temporary area on disc.
syntax:	BYE[BYE] This command terminates the session or changes the operational mode
syntax:	DCU<userid>,<disc number> Declare a new user (system session only)
syntax:	DEL<name>/OB /S /O /L] Delete a file or object module from a library
syntax:	DLU<userid>,<disc number> Delete a user (system session only)
syntax:	DUF<file code /O /L <name>][,<sect nb1>,<sect nb2>]] Dump file
syntax:	INC<OBJECT <name>][,<userid>,<disc file code>]] Include an object module
syntax:	JOB<userid>/FX,<userid>] Define a new session
syntax:	KPF<S /O /L <file code>][,<name>] Keep a file
syntax:	LIC<disc number> List catalogue on operator's typewriter (system session only)
syntax:	LSD</OB] List directory on operator's typewriter

syntax:	LST<file code> /S /S,<name> <name>][,<line nb1>,<line nb2>]] List file
syntax:	MES<message to operator> Command especially used in batch processing. It allows the specified message to be sent to the operator
syntax:	MOV<name>,<S /L <file code>][,<userid>,<disc file code>]] Move a file
syntax:	PCH<file code> /S <name> <name>,<S] Punch a file
syntax:	PLB<file code> Used to list the label of a tape on the operator's typewriter
syntax:	PLD<name>,<L] Punch a load file
syntax:	POB<name>] Punch an object module
syntax:	PRC<disc number> Print catalogue on print unit (system session only)
syntax:	PRD</OB] Print directory on print unit
syntax:	PRT<file code> /S /S,<name> <name>][,<line nb1>,<line nb2>]] Print file
syntax:	PSE<message to operator> Used to send a message to the operator and go in Pause state. The operator has to give an RS command to continue
syntax:	RDA<disc file code>,<input file code>] Read data
syntax:	RDO<file code>] Read object
syntax:	RDS<file code>] Read source
syntax:	REF<file code> This command positions the file to the first record
syntax:	RSU<disc number> Replace supervisor (system session only)
syntax:	RUN<name> <name>,<0 0] Run a program
syntax:	SCR[<S /O /L <file code>]] Scratch
syntax:	SDM<disc file code>,<mag tape file code> Save user file on magnetic tape
syntax:	SEG<name list> Define segments
syntax:	SVD<disc number 1>,<disc number 2> Save disc onto another disc (system session only)
syntax:	SVU<userid>,<disc number> Save user files

PROCESSOR CALLS

syntax: ASM_L[/S|<name>][,NL]
 Call the Assembler

syntax: LKE_L[/N|S|U][,M][,DE|DS][, <address>][, <start address>]
 Call the Linkage Editor

syntax: UPD_L<name>[, <file code1>]/S[, <file code2>]
 Call the Update Package

syntax: EDT_L<name>
 Call the Text Editor

syntax: DEB_L[<name>]
 Call the Debugging Package

syntax: FOR_L[/S|<name>][,NL]
 Call the Full FORTRAN Compiler

syntax: FRT_L[/S|<name>][,NL]
 Call the Basic FORTRAN Compiler

syntax: TCD
 Call the Full FORTRAN Transcoder

MAGNETIC TAPE AND CASSETTE TAPE COMMANDS

syntax: REW_L<file code>
 Rewind until load point

syntax: UL_L<file code>
 Unlock device

syntax: FFS_L<file code>[, <number>][ALL]
 File forward space

syntax: FBS_L<file code>[, <number>]
 File backward space

syntax: RFS_L<file code>[, <number>]
 Record forward space

syntax: RBS_L<file code>[, <number>]
 Record backward space

syntax: WEF_L<file code>[, <number>]
 Write End-Of-File

syntax: WES_L<file code>[, <number>]
 Write End-Of-Segment

syntax: REF_L<file code>
 This command positions the file to the first record

syntax: WLB_L<file code>[, <serial number>[, <security code>[, <owner>]
 This command is used to write the VOL 1 label on magnetic tape or cassette tape

syntax: WEV_L<file code>
 This command writes the End of Volume mark on the magnetic tape or cassette tape

SCL COMMANDS

LD_L<name>[, <disc number>[, <level>]/CM[, <level>][, SL, <number>]]
 Load a memory-resident program

RO_L<name>[, <disc number>][, SL, <number>]
 Declare a Read-Only program

CN_L<name>[, <level>]
 Connect a program to a software level

CT_L<name>[, <NTIM>[, <PR>[, <NC>][<HH>[, <MM>[, <SS>]]]
 Connect a program to the clock or to a timer

DN_L<name>
 Disconnect a program from a level

DT_L<name>
 Disconnect a program from a clock or timer

SD_L<DD>[, <MM>[, <YY>]
 Set date

SC_L[<HH>[, <MM>[, <SS>]]]
 Set clock

ST_L<name>[, <disc number>]
 Start a program

KF_L<file code>[, <file name>]
 Keep file

AS_L<file code1>[, <file code2>][DN|DA][DKFX[, <file name>[, <nr of granules>]]
 Assign a file code

DL_L<file code>
 Delete a file code

DF_L<file name>[, <disc number>]
 Delete a file

HC
 Halt clock

EN
 End of SCL commands

DISC PROCESSOR MESSAGES**Update**

!!DL_L<line number 1>[, <line number 2>]
 Delete the line, or lines, specified

!!!L_L[<line number>]
 Insert line(s) after the specified line number or after this statement if the parameter is not specified

!!EN
 End of Update

!!AB
 Abort the update and delete /S file

TERMINATE THE UPDATING WITH A KPF COMMAND

Debugging

AT<address>	Define a breakpoint
IF<[<register> <M<address>]>[< =>][</value> <register> <M<address>]>	Conditional execution of breakpoint
DM<address1>[-<address2>]	Dump memory
DR<[<number>] [-<number>]	Dump register
WM<address>=<value1>[-<value2>-...<value n>]	Write memory
WR<number>=<value1>[-<value2>-...<value n>]	Write register
TR<2 characters>	Trace
GO<address>	Restart program
CI	Change input
RE<file code>-<number> <address>	Read from external device
DB<address>	Delete a breakpoint
RT	Return

BASIC PROCESSOR MESSAGES**Assembler**

Option message:

[x x x x] [N] [R] [Q]	
	object output must be in 4x4 format
	possibility to correct recoverable errors
	no assembly listing
	object output file code. 0 if no object output wanted
	listing output file code
	0 (zero)
	source input file code

Other messages:

LF CR	1. No option message required. Default: 1023 2. Resume processing after :EOS 3. Concludes typed-in statement
:EOF	Terminate processing after A:

Linkage Editor

Option message

[[E L][:<xxx>],<name>[,4 8]][,/<address> ,<name>]	
	start address of produced module
	abs. start address of blank common
	program must be punched in 8+8
	program must be punched in 4x4
	name for load module
	object module input device
	listing output device
	object code output device (not used in link-load mode. Type not in 0)
	link-load mode (not in stand-alone L.E.)
	link-edit mode

Operator Messages

LF CR 1. No option message required. Default = L:21
2. Concludes typed-in message

A<address>	Define absolute address
E<entry points name list>	Define entry points (link-edit only)
L	Use library to solve unsat. external ref.
P	Process input file up to EOF
R<address>	Define relative base address
S<symbol>	Select named module
T	Terminate processing
U	List undefined external references
X<external reference name list>	Define external reference names (link-edit only)

Update

Option message

<xxxx>[,8],4]	
	object program to be updated is punched in 4x4
	object program to be updated is punched in 8+8
	file code onto which the updated program is punched or written program can be listed
	file code of the peripheral onto which the input or output program can be listed
	file code from which data can be added to master file
	file code from which the control messages are read
	master file code

Operator Messages

D:<name>	Delete object or source module of this name
D:<line no a>[<line no b>]	Delete line or lines (the ones specified included)
:EOF	Punch :EOF on Punch File
:EOS	Punch :EOS on Punch File
I<name>	Insert object or source module of this name
I:	Correction statements may be input
I:<line no>	Insert line or lines on this place
L	List all modules of the input file
M:<name>	Start updating at the module specified
S:<name>	All records deleted up to source module <name>
S	All records deleted up to :EOF mark
:	No updating until next :EOS
	1. No updating until next :EOF
	2. After U: terminate processing.
LF CR	Concludes typed-in statement or message

Debugging Messages

AT<address>.	Define a breakpoint as specified
Cl.	Read further control messages
DB<address>.	Delete breakpoint as specified
DM<address1>[<address2>].	Dump memory as specified
DR<[<number>][<number>].	Dump registers as specified
GO<address>.	Start program at specified address
IF<[<register>][M<address>][< = >][<value>][<register>][M<address>].	Conditional execution of breakpoint
RE<[file code]>[<number>]<address>.	Read from external device
RT.	Start execution of user program
TR<2 characters>.	Trace the execution of next breakpoint specified
WM<address>=<value1>[<value2>=<value3>...<value n>].	Write into memory
WR<number>=<value1>[<value2>=<value3>...<value n>]	Write into register

INSTRUCTIONS

In the instructions described below, each mnemonic (OPC) is followed by a letter (from a to g). This letter refers to the syntax description immediately following the instruction group heading (MEMORY REFERENCE INSTRUCTIONS etc.). Each instruction is followed by a number (from 1 to 15) referring to a condition register contents described hereafter.

- 1) CR = 0 if (<r1>) = 0
1 if (<r1>) > 0
2 if (<r1>) < 0
- 2) CR = unchanged
- 3) CR = 0 if result = 0
1 if result > 0
2 if result < 0
3 in case of overflow
- 4) CR = 0 if result = 0
1 if result > 0
2 if result < 0
- 5) CR = 0 if (A1) = 0
1 if (A1) > 0
2 if (A1) < 0
- 6) CR = 0 if (<r1>) = (2nd operand)
1 if (<r1>) > (2nd operand)
2 if (<r1>) < (2nd operand)
- 7) CR = 0 if (<r1>) = (2nd operand)l/r
1 if (<r1>) > (2nd operand)l/r
2 if (<r1>) < (2nd operand)l/r
- 8) CR = 0 if (A2) = 0
1 if (A2) > 0
2 if (A2) < 0
3 in case of overflow
- 9) CR = 0 if (<r1>)r = ((<r2>)l/r
1 if (<r1>)r > ((<r2>)l/r
2 if (<r1>)r < ((<r2>)l/r
- 10) Bits 0-5 of PSW loaded in PLR if <r2> = A15.
When reloaded from stack bits 6 and 7 of PSW are loaded in CR
- 11) CR = 0 if <lk> = 0
1 if <lk> > 0
2 if <lk> < 0
- 12) CR = 0 if (<r1>) = <lk>
1 if (<r1>) > <lk>
2 if (<r1>) < <lk>
- 13) CR = 0 if (<r1>)r = <lk>l
1 if (<r1>)r > <lk>l
2 if (<r1>)r < <lk>l
- 14) CR = 0 command accepted
1 command not accepted
3 device address unknown
- 15) CR = 0 command accepted
3 device address unknown

- RF (e) $(P) + \langle D \rangle \rightarrow P$ (branch) (2)
 $(P) + 2 \rightarrow P$ (no branch)
- RB (e) $(P) + \langle D \rangle \rightarrow P$ (branch) (2)
 $(P) + 2 \rightarrow P$ (no branch)

Register-To-Register Instructions

- Syntax: (a) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle [*]_{\text{u}} \langle r1 \rangle, \langle r2 \rangle$
 (b) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle_{\text{u}} \langle r1 \rangle, \langle r2 \rangle$
 (c) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle_{\text{u}} \langle r2 \rangle$
 (d) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle_{\text{u}} \langle n \rangle, \langle r2 \rangle$
 (e) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle [(\langle \text{cnd} \rangle)] [*]_{\text{u}} \langle r2 \rangle$
 (f) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle [*]_{\text{u}} [\langle r1 \rangle], \langle r2 \rangle$
 (g) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle [*]_{\text{u}} \langle r2 \rangle$
- LDR (a) $\langle r2 \rangle \rightarrow \langle r1 \rangle$ (1)
- STR (b) $\langle r1 \rangle \rightarrow \langle r2 \rangle$ (2)
- ADR[S] (a) $\langle r1 \rangle + \langle r2 \rangle \rightarrow \langle r1 \rangle$ (3)
- SUR[S] (a) $\langle r1 \rangle - \langle r2 \rangle \rightarrow \langle r1 \rangle$ (3)
- ANR[S] (a) $\langle r1 \rangle \wedge \langle r2 \rangle \rightarrow \langle r1 \rangle$ (4)
- ORR[S] (a) $\langle r1 \rangle \vee \langle r2 \rangle \rightarrow \langle r1 \rangle$ (4)
- XRR[S] (a) $\langle r1 \rangle \nabla \langle r2 \rangle \rightarrow \langle r1 \rangle$ (4)
- IMR (c) $((\langle r2 \rangle) + 1) \rightarrow \langle r2 \rangle$ (3)
- C2R (c) $0 - ((\langle r2 \rangle)) \rightarrow \langle r2 \rangle$ (3)
- MLR (d) $((\langle r2 \rangle)) \rightarrow A1$
 $((\langle r2 \rangle) + 2) \rightarrow A2$
 $((\langle r2 \rangle) + 2n - 2) \rightarrow A_n$ (5)
- MSR (d) $(A1) \rightarrow \langle r2 \rangle$
 $(A2) \rightarrow \langle r2 \rangle + 2$
 $(A_n) \rightarrow \langle r2 \rangle + 2n - 2$ (2)
- ABR (e) $\langle r2 \rangle \rightarrow P$ (branch)
 $(P) + 2 \rightarrow P$ (no branch) (2)
- CFR (a) $(P) \rightarrow \langle r1 \rangle, \langle r1 \rangle - 2 \rightarrow \langle r1 \rangle$
 $(PSW) \rightarrow \langle r1 \rangle, \langle r1 \rangle - 2 \rightarrow \langle r1 \rangle$
 $\langle r2 \rangle \rightarrow P$ (2)
- C1R (f) $\overline{\langle r2 \rangle} \rightarrow \langle r1 \rangle$ (3)
- C1RS (c) $\overline{((\langle r2 \rangle))} \rightarrow \langle r2 \rangle$ (3)
- CWR (a) $\langle r1 \rangle \div \langle r2 \rangle \rightarrow CR$ (6)
- LCR (b) $((\langle r2 \rangle))_{/r} \rightarrow \langle r1 \rangle_r$ (2)

- CCR (b) $(\langle r1 \rangle)_r \div ((\langle r2 \rangle))_{/r} + CR$ (9)
- SCR (b) $(\langle r1 \rangle)_r \rightarrow (\langle r2 \rangle)_{r/l}$ (2)
- RTN (c) $\langle r2 \rangle + 4 \rightarrow \langle r2 \rangle$
 $((\langle r2 \rangle)) \rightarrow P$
 $((\langle r2 \rangle)) - 2 \rightarrow PSW$ (in system mode only) (10)
- ECR (b) $\langle r2 \rangle \rightarrow \langle r1 \rangle$ and $(\langle r2 \rangle)_r \rightarrow (\langle r1 \rangle)_l$ (2)
- MUR (g) $(A2) \times (\langle r2 \rangle) \rightarrow A1, A2$ (3)
- DVR (g) $(A1, A2) / (\langle r2 \rangle) \rightarrow A2(\text{quotient}) \quad A1(\text{remainder})$ (5)
- DAR (g) $\langle r2 \rangle, \langle r2 \rangle + 1 + (A1, A2) \rightarrow A1, A2$ (3)
- DSR (g) $(A1, A2) - (\langle r2 \rangle, \langle r2 \rangle + 1) \rightarrow A1, A2$ (3)
- TM (b) $[(\langle r1 \rangle) \wedge (\langle r2 \rangle)] \div 0 \rightarrow CR$ (4)
- TNM (b) $[(\langle r1 \rangle) \nabla (\langle r2 \rangle)] \div 0 \rightarrow CR$ (4)

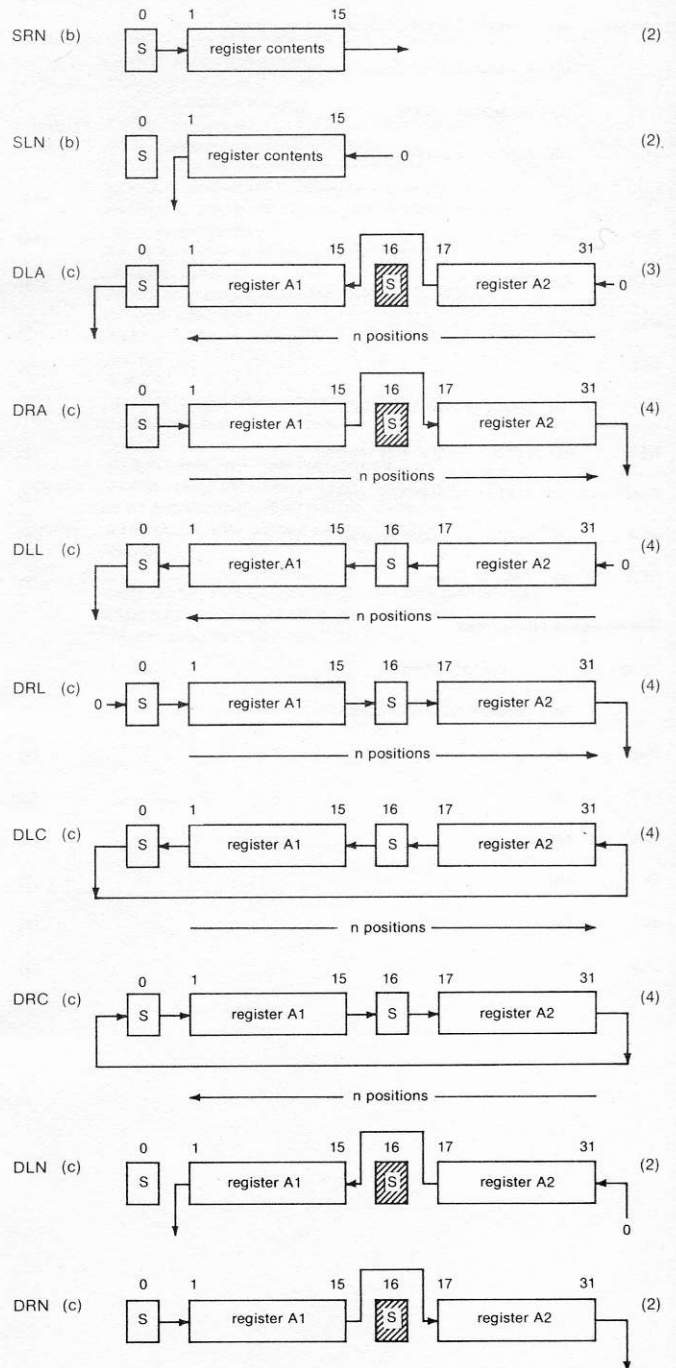
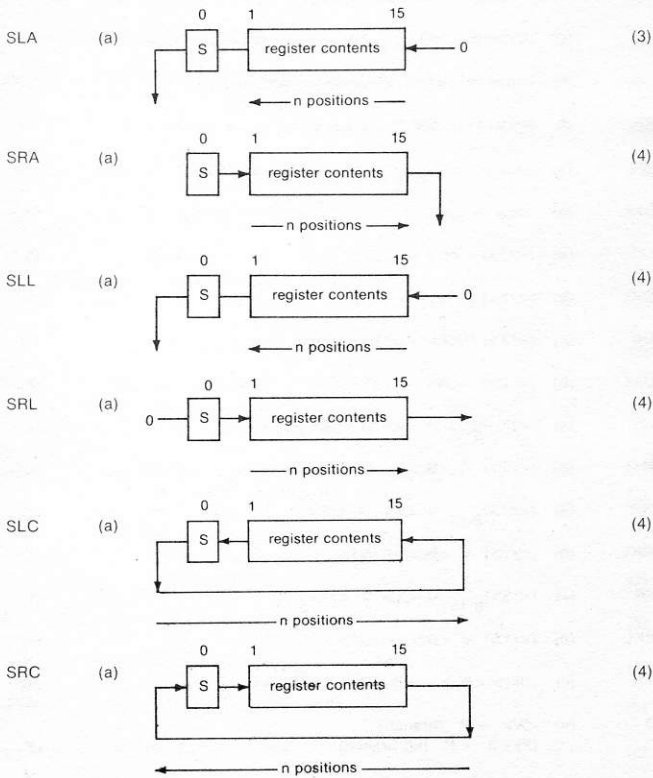
Constant Instructions

- Syntax: (a) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle_{\text{u}} \langle r3 \rangle, \langle k \rangle$
 (b) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle_{\text{u}} \langle r1 \rangle, \langle lk \rangle$
 (c) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle_{\text{u}} \langle n \rangle$
 (d) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle_{\text{u}} \langle lk \rangle$
 (e) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle (\langle \text{cnd} \rangle)_{\text{u}} [\langle k \rangle | \langle lk \rangle]$
 (f) $[\langle \text{ident} \rangle]_{\text{u}} \langle \text{OPC} \rangle_{\text{u}} \langle lk_1 \rangle, \langle lk_2 \rangle$
- LDK (a) $\langle k \rangle \rightarrow \langle r3 \rangle_{8-15} \quad 0 \rightarrow \langle r3 \rangle_{0-7}$ (11)
- LDKL (b) $\langle lk \rangle \rightarrow \langle r1 \rangle$ (11)
- ADK (a) $\langle r3 \rangle + \langle k \rangle \rightarrow \langle r3 \rangle$ (3)
- ADKL (b) $\langle r1 \rangle + \langle lk \rangle \rightarrow \langle r1 \rangle$ (3)
- SUK (a) $\langle r3 \rangle - \langle k \rangle \rightarrow \langle r3 \rangle$ (3)
- SUKL (b) $\langle r1 \rangle - \langle lk \rangle \rightarrow \langle r1 \rangle$ (3)
- ANK (a) $\langle r3 \rangle_{8-15} \wedge \langle k \rangle \rightarrow \langle r3 \rangle_{8-15}$ (4)
- ANKL (b) $\langle r1 \rangle \wedge \langle lk \rangle \rightarrow \langle r1 \rangle$ (4)
- ORK (a) $\langle r3 \rangle_{8-15} \vee \langle k \rangle \rightarrow \langle r3 \rangle_{8-15}$ (4)
- ORKL (b) $\langle r1 \rangle \vee \langle lk \rangle \rightarrow \langle r1 \rangle$ (4)
- XRK (a) $\langle r3 \rangle_{8-15} \nabla \langle k \rangle \rightarrow \langle r3 \rangle_{8-15}$ (4)
- XRKL (b) $\langle r1 \rangle \nabla \langle lk \rangle \rightarrow \langle r1 \rangle$ (4)
- MLK (c) $\langle lk_1 \rangle, \langle lk_2 \rangle, \dots, \langle lk_n \rangle \rightarrow A1, A2, \dots, A_n$ (5)
- AB (e) $\langle k \rangle \rightarrow P$ (branch)
 $(P) + 2 \rightarrow P$ (no branch) (2)

- ABL (e) $\langle lk \rangle \rightarrow P$ (branch) (2)
- (P) + 2 $\rightarrow P$ (no branch) (2)
- CF (b) (P) $\rightarrow \langle r1 \rangle$ ($\langle r1 \rangle - 2 \rightarrow r1$) (2)
- (PSW) $\rightarrow \langle r1 \rangle$ ($\langle r1 \rangle - 2 \rightarrow \langle r1 \rangle$) (2)
- CWK (b) $\langle r1 \rangle \div \langle lk \rangle \rightarrow CR$ (12)
- CCK (b) $\langle r1 \rangle_r \div \langle lk \rangle_l \rightarrow CR$ (13)
- LCK (b) $\langle lk \rangle_l \rightarrow \langle r1 \rangle_r$ (2)
- MUK (d) $(A2) \times \langle lk \rangle \rightarrow A1, A2$ (3)
- DVK (d) $(A1, A2) / \langle lk \rangle \rightarrow A1(\text{quotient}) \quad A2(\text{remainder})$ (8)
- DAK (f) $\langle lk_1 \rangle, \langle lk_2 \rangle + (A1, A2) \rightarrow A1, A2$ (3)
- DSK (f) $(A1, A2) - \langle lk_1 \rangle, \langle lk_2 \rangle \rightarrow A1, A2$ (3)

Shift Instructions

- Syntax:
- (a) $[\langle ident \rangle] \ll \langle OPC \rangle \ll \langle r3 \rangle, \langle n \rangle$
 - (b) $[\langle ident \rangle] \ll \langle OPC \rangle \ll \langle r3 \rangle, \langle r2 \rangle$
 - (c) $[\langle ident \rangle] \ll \langle OPC \rangle \ll \langle n \rangle$



I/O Instructions

- Syntax: (a) [`<ident>`] `└<OPC>` `└<r3>`, [0|1], `<dev>`
 (b) [`<ident>`] `└<OPC>` `└<r3>`, [0|1], `<dev>`
 (c) [`<ident>`] `└<OPC>` `└<r3>`, `<dev>`
 (d) [`<ident>`] `└<OPC>` `└<r3>`

CIO	(a)	(14)
INR	(b)	(14)
OTR	(b)	(14)
SST	(c)	(14)
TST	(c)	(15)
RIL	(d) Interrupt lines → <code><r3></code>	(2)
WMP	(d) <code><r3></code> → M.P. Key register	(2)
WM2	(d) <code><r3></code> → M.P. Key register	(2)
WIM	(d) <code><r3></code> → mask register	(2)
RCA	(d) CNL → <code><r3></code>	(2)

Miscellaneous Instructions

- Syntax: (a) [`<ident>`] `└<OPC>`
 (b) [`<ident>`] `└<OPC>` `└<k>`

ENB	(a)	(2)
HLT	(a)	(2)
SMD	(a)	(2)
RIT	(b)	(2)
INH	(a)	(2)
LKM	(a)	(2)

DIRECTIVES

- [`<ident>`] `└DATA` `└<data expression>` [, `<data expression>`, ...]
 (up to 16 words generated)
 [`<ident>`] `└EQU` `└<predefined expression>`
`└IDENT` `└<module name>`
 [`<ident>`] `└END` `└<predefined expression>` [, `<symbol>`]
 [`<ident>`] `└RES` `└<predefined absolute expression>`
`└AORG` `└<predefined absolute expression>`
`└RORG` `└<predefined relocatable expression>`
`└ENTRY` `└<entry point name>` [, `<entry point name>`, ...,
`<entry point name>`]
`└EXTRN` `└<external name>` [, `<external name>`, ...,
`<external name>`]
`└STAB` `└<internal symbol>` [, `<internal symbol>`, ...,
`<internal symbol>`]
`└NLIST`
`└LIST`
`└EJECT`
`└[IFT|IFF]` `└<predefined absolute expression>` =
`<predefined absolute expression>`
`└XIF`
 [`<ident>`] `└COMN` `└<common field definition list>`
 [`<ident>`] `└FORM` `└<field definition>` [, `<field definition>`, ...,
`<field definition>`] [`/` `<field number list>`]
 [`<ident>`] `└XFORM` `└<FORM defined pseudo-mnemonic>`, `<field list>`
`└GEN`

`<data expression>` ::= `<expression>` | `<character string>`
`<field definition>` ::= `<field length definition>`
 [= `<field value definition>`]

LKM	Monitor requests	BOM	DOM	SRTM	BRTM	DRTM
1	I/O	X	X	X	X	X
2	Wait for an event	X	X	X	X	X
3	Exit	X	X	X	X	X
4	Get Buffer	X	X		X	X
5	Release Buffer	X	X		X	X
6	Pause	X	X			
7	Keep control on abort condition	X	X			
8	DATM	X	X		X	X
9	Load a segment		X			
10	Connect a program to a timer			X	X	X
11	Disconnect a program from a timer			X	X	X
12	Activate program			X	X	X
13	Switch inside a software level				X	X
14	Attach a device to a program			X	X	X
15	Detach a device from a program			X	X	X
17	Get Time				X	X
18	Reset an event				X	X
20	Connect a program to a level				X	X
21	Disconnect a program from a level				X	X
22	Wait for a given time				X	X
23	Assign file code					X
24	Delete file code					X
25	Read unsolicited operator message					X

I/O FUNCTIONS

- /01 Basic Read
- /02 Standard Read
- /05 Basic Write
- /06 Standard Write
- /07 Object Write 4 x 4
- /08 Object Write 8+8
- /0A Random Read (only for disc)
- /0B Random Write (only for disc)
- /30 Get Information about a file code

TAPE HANDLING ORDERS

- /14 Skip Forward to next EOS mark
- /16 Skip Forward to next EOF mark
- /22 Write EOS mark
- /24 Write EOV (End of Volume) Mark
- /26 Write EOF mark
- /31 Rewind to Load Point
- /33 Backspace one block
- /34 Skip backwards to EOF mark
- /37 Lock
- /38 Unlock

SOFTWARE STATUS WORD

Request accepted

For Basic orders: status word gives the hardware status
 For Standard orders: Status 8000: no operation (after RD message)
 Status 8XXX: bit 8 End Of Volume mark
 10 beginning of tape
 11 end of disc/tape
 12 incorrect length requested
 13 illegal character code or checksum error (object)
 14 EOS
 15 EOF
 Status 0000: normal I/O completion

Request rejected

For Basic and Standard orders:
 Status CXXX: bit 5 disc overflow (no more granules available)
 6 no disc buffer available (Dynamic allocation area overflow)
 7 disc queue overflow
 11 function unknown or not compatible with device
 BOM 12 illegal buffer size
 DOM 13 illegal buffer address
 BRTM 14 device attached to another program
 15 illegal file code or non-existing

C.U. STATUS WORD CONFIGURATION

Bit	Description	CU										
		ASR	CR	MHD	LP	TP	PTR	CASS Tape	PLOT	FHD	MT	
0	-											
1	ready			x				x				x
2	rewind											x
3	tape mark has been read							x				x
4	-											
5	on cylinder load point			x								x
6	seek error			x								
	write unable							x				x
7	A or B side							x				
8	Device Address							x		x	x	
9	Device Address			x				x		x	x	
10	EOT						x	x				
	tape low					x						
11	Program error			x				x	x	x	x	
12	Incorrect length		x	x				x		x	x	
	Y limit overpass									x		
13	Parity error							x				
	Data fault		x	x						x	x	
14	throughput error	x	x	x			x	x		x	x	
15	not operable	x	x	x	x	x	x	x	x	x	x	

ASR I/O typewriter TP tape punch PLOT plotter
 CR card reader PTR punched tape reader FHD fixed head disc
 MHD moving head disc CASS cassette tape MT magnetic tape
 LP line printer

DATA COMMUNICATION

LKM ± 8 (- 8, < sched lab >)

I/O requests DITEM

- /01 Read with Echo
- /02 Read without Echo
- /03 Read with Echo and without Time-out
- /04 Read with Echo and with Time-out
- /06 Write a Transmission Block
- /0F Get Line Type
- /10 Stop the Exchange
- /11 Disconnect the Line
- /12 Search Line Address
- /13 Accept Automatic Call
- /14 Accept Data
- /15 Set Time-out

I/O requests BSC

- 01 Read Initial } Point-to-point lines
- 02 Write Initial }
- 03 Read Initial } Multipoint lines
- 04 Write Initial }
- 05 Search
- 06 Read Continue
- 07 Read Repeat
- 08 Write Continue
- 09 Write EOT
- 10 Write Forward Abort
- 11 Request for Line Disconnection

Line Control Table

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
WORD 0																
WORD 1																
WORD 2	TIME OUT VALUE															
WORD 3	ECB ADDRESS															
WORD 4	CURRENT CHARACTER ADDRESS															
WORD 5	NUMBER OF CHARACTERS REQUESTED															
WORD 6	NUMBER OF RECEIVED / SENT CHARACTERS															
WORD 7	WAIT FOR DATA BUFFER OR SYN PATTERN (IN AND OUT) (MELCU) (SYLCU)															
WORD 8	SCHEDULED LABEL ADDRESS															

- word 0 bit 0 = 1 busy
 1 = 1 write
 2 = 1 read
 3 = 1 echo
 4 = 1 wait for data/ENQ
 5 = 1 wait for call
 6 = 1 time-out
 8 = 1 multiplex channel
 0 programmed channel
 9 = 1 SYLCU
 0 ASYLCU
 10 = 1 MELCU
 11 = 1 full duplex
 0 half duplex

word 1 bit 0 = 1 connected
 1 = 1 start bit received
 2 = 1 echo refused
 3 = 1 character received
 4 = 1 character sent
 5 = 1 escape received
 6 = 1 check character
 7 = 1 wait for data character received
 8 = 1 refuse this character
 9 = 1 two more characters requested
 12-15 line number (0 to 7 or /F)

DATM

OF GET LINE TYPE

bit 15 } 00 = ASYLCU
 01 = MELCU
 14 } 10 = SYLCU
 13 1 = DERCO (SYLCU) or MIU (MELCU)
 12 1 = connected
 11 not used (zero)
 10 1 = Multiplex Channel
 9 1 = full-duplex
 8 1 = CRC computer (DERCO)
 7 1 = parity computed
 6 1 = odd parity
 0 = even parity
 5-1 not used (zeroes)
 0 line busy

DATM ECB

The ECB address must be loaded in register A8.
 The structure of the ECB is:

WORD 0	EVENT CHARACTER	LOGICAL LINE NO
WORD 1	TRANSMISSION BLOCK ADDRESS	
WORD 2	REQUESTED LENGTH	
WORD 3	TRANSMITTED LENGTH	
WORD 4	STATUS WORD	
WORD 5	SPECIAL CHARACTER TABLE ADDRESS	
WORD 6	TIME-OUT VALUE	

The status word has the following format:

bit 15 = 1 modem not operable
 14 = 1 throughput error
 13 = 1 parity error
 12 = 1 I/O error
 11 = 1 Escape received (interrupt in transmission block)
 10 = 1 time over
 9 = 1 not used
 8 = 1 transmission stopped
 7 = 1 buffer overflow
 6 = 1 time-out request may not be served
 5 = 1 non-positive message length
 4 = 1 characters lost (wait for data)
 3 = 1 non-compatible request, e.g. Wait for Data for SYLCU or illegal request
 2 = 1 illegal file code
 1 = 1 non-connected line
 0 = 1 line is busy

BSC ECB

	0	8	9	11	12	13	14	15		
WORD 0	+		*	+		*	*	+	+	+
WORD 1	RESERVED			LINE CODE						
WORD 2	BUFFER AREA ADDRESS									
WORD 3	BUFFER LENGTH									
WORD 4	RESERVED									
WORD 5	STATUS WORD									
WORD 6	RESERVED									
WORD 7	POLLING/SELECTING TABLE ADDRESS OR TIME-OUT VALUE									
WORD 8	LINE AREA ADDRESS									

* BIT SET/RESET BY PROGRAM + BIT SET/RESET BY USER

Where:

Word 0 bit 0 = 1 the operation is completed
 8 = 0 slave/tributary station
 1 master/control station
 9 = 1 ITB control character received
 11 = 0 ASCII-code
 1 EBCDIC-code
 12 = 0 LRC
 1 CRC
 13 = 0/1 user is sending a header (reset by BSC)
 user has received a header (to be reset by user)

Word 5 Status word

The status word is divided as follows:

bits 0 - 3 Only one of four bits is set to give the kind of status:
 bit 0 Acknowledgement status
 bit 1 Software error status
 bit 2 Transmission error status
 bit 3 Procedure error status
 bits 4 - 15 give, per bit, a description of the status.

Acknowledgement status (bit 0)

bit 15 Conversational mode received
 14 ENQ received (contention: local station is slave)
 13 RVI received
 12 EOT received (normal condition)
 11 Forward abort, STXENQ or EOT received
 10 ENQ received or write EOT

Software error status (bit 1)

bit 15 Write conversational refused (NAK pending)
 14 Non-positive text length (Write request)
 13 Pending BSC request on this line
 12 Header transmission refused
 11 Illegal request
 10 Illegal file code
 9 Non-connected line } detected by DATM
 8 Line busy }
 7 System error (stack error)
 6 System error (time-out not serviced)

Transmission error status (bit 2)

bit 15 Modem not operable
 14 Throughput error
 13 Parity error
 12 BCC error (software computation)

SYSTEM MESSAGES

Message	Program	Meaning
A:	ASM	Request for option/control message. See page 8
A:	IPLRT	Request for activate message. Answer with A or LF CR
ABORT <code> <address>	BOM	Program aborted
ABS.ADR.	LKE	Disk Linkage Editor does not accept absolute addresses
ABS.STR.	LKE	Absolute start address (ignored)
ASS.ERR. <number>	ASM	Number of assembly errors
ASSIGN ERROR	CCI	KPF erroneous assignment
BATCH PROCESSING?	CCI	Type in Y [ES] or N [O]
BLK.COM	LKE	Erroneous optional blank common address
BLK DAT <name>	LKE	Unknown common block name used in a Block Data subprogram
BLK DATA ER	LKE	Data error encountered
BOOT CK ER	IPL	Checksum error encountered
C:	UPD	Request for control message. See page 9
C?	UPD	Parameter error. Request for correct control message
*C	ASM	Illegal constant
CATALOG OVERFLOW	CCI	Too many userids catalogued
COMMAND NOT ALLOWED	CCI	This is not a system userid
COMMAND UNKNOWN	CCI	Erroneous CCI command
C ER	LKE	Labeled common error or error in base address of blank common
CORE OVERFLOW	LKE	Insufficient core available for user program
CORE RESIDENT AREA LENGTH:	DRTM	Type in the length (4 hexa char) of this area
D:	DBG	Request for control message. See page 10
DATE:	DOM/ DRTM	Disk system asking for date. DD MM YY or YY MM DD
D.D.	LKE	Double definition error

Message	Program	Meaning
DBL.DEF <name>	LKE	Name is defined more than once as an entry point or in the name of a common block
DEBUG OPTION REDUNDANT	CCI	Debug option redundant
DEVICE ADDRESS ERROR	CCI	Erroneous device address specified
DEVICE NAME ERROR	CCI	Erroneous device name specified
DEVICE NAME MISSING	CCI	2nd parameter missing in command
DEVICE UNKNOWN	CCI	Unknown device address
DIRECTORY OVERFLOW	CCI	Directory overflow
DIRECTORY OVERFLOW ON XXXXXFT	CCI	Directory overflow. File FT is catalogued
DISK ADDRESS MISSING	CCI	Disk address not specified
DISK <address> UNKNOWN	DRTM	Disk unknown by CPU
DISK ASSIGN ERROR	CCI	System cannot assign a temporary work file
DISK FILE CODE ABSENT	CCI	Disk file code not specified in command
DISK FILE CODE ERROR	CCI	2nd parameter not numeric
DISK FILE CODE UNKNOWN	CCI	File code not declared at sysgen
DISK FILE CODE MISSING	CCI	No file code specified
DISK NOT OPERATIONAL	CCI	Disk unit not ready
DISK I/O ERROR	CCI	I/O error on disk
DISK OVERFLOW	CCI	No free granule available to allocate to temporary disk file
DISK UNKNOWN	CCI/DRTM	Disk not specified at sysgen
DKER┐<address><sect number> <status>	DOM/ DTRM	Disk not ready to be used
DSK INPUT ERR, UPD ABORTED	CCI	Erroneous input for Disk Update
DSK INIT ERR	DOM/ DRTM	Disk not ready to be used
DSK OUTPUT ERR, UPD ABORTED	CCI	Erroneous output from Disk Update
DYN AREA LENGTH:	DRTM	Length of dynamic area requested (4 hexa char.)
** DRTM ** yy **	DRTM	yy = release number
* E	ASM	Address is not even

Message	Program	Meaning
E = <absolute address>	LKE	Address is the highest absolute address in the generated module
***** E	ASM	END directive missing
EC	BOM/ BRTM IPLRT	Erroneous cluster or input error encountered during loading
EC TYPE	LKE	Erroneous cluster encountered
END MIS	LKE	END cluster missing
:EOF	BOM/ASM	End-of-file mark encountered
EOF, UPD TERMINATED	CCI	EOF encountered before reaching specified line
:EOS <address>	BOM/ASM	End of segment encountered. Address is the first free location
EOV ON INPUT FILE, MOUNT NEW TAPE THEN RESTART	DOM	EOV mark detected before EOF. Place new reel and restart.
EOV ON OUTPUT FILE, MOUNT NEW TAPE THEN RESTART	DOM	The EOV mark detected on output device (magnetic tape or cassette tape). Mount a new reel and restart
ER	BOM/DOM	Operator message error
ER 00	SCL	Command unknown
01	SCL	System error
02	SCL	Disc not operational
03	SCL	File code unknown
04	SCL	No PCT available
05	SCL	Read-only save area overflow
06	SCL	Memory-resident area overflow
07	SCL	Level error
08	SCL	Level already connected
09	SCL	Program unknown
10	SCL	Too many scheduled labels
11	SCL	I/O error or too many scheduled labels
12	SCL	Program already declared previously
13	SCL	Program too long
14	SCL	Program has not been connected

Message	Program	Meaning
15	SCL	Parameter error
16	SCL	The specified timer has not been assigned
21	SCL	Unknown file name
22	SCL	Non-disc file
23	SCL	File has already been catalogued.
24	SCL	No entry available in the library directory
51	SCL	I/O error on disk
52	SCL	No spare entry available in FCT
53	SCL	No disk file description table free
54	SCL	Device unknown or disc file code unknown
55	SCL	Disc overflow or too many granules requested
56	SCL	File unknown
57	SCL	File code 2 unknown
58	SCL	More than 7 file codes assigned to the same disc file
ER.MOD	LKE	Erroneous input module
ERR.MOD.	CCI	Error in assembly or compilation
ERR.LKE	LKE	A non-fatal error has occurred during this link-edit run
ERROR ASSIGN	CCI	Erroneous file code specified
EXIT	BOM	User program run completed
* F	ASM	Illegal FORM or XFORM directive
FATAL ERROR HAS OCCURRED. NO OBJECT CODE PRODUCED	ASM	A fatal error has occurred during assembly
FCT OVERFLOW	CCI	File code table overflow
FILE ALREADY CATALOGUED	CCI	This file was already kept
FILE CODE ABSENT	CCI	File code not specified in command
FILE CODE ERROR	CCI	Erroneous file code specified
FILE CODE MISSING	CCI	Parameter is not specified
FILE CODE NOT ASSIGNED	CCI	File code assigned to NO device or not yet assigned

Message	Program	Meaning
FILE CODE UNKNOWN	CCI	Wrong file code specified
FILE NAME ERROR	CCI	First parameter is neither /S, nor a file code nor a character string
FILE NAME	CCI	CCI dit not recognise this file name
FILE NOT CATALOGUED	CCI	File to be deleted not catalogued
FILE OVERFLOW	CCI	File cannot accept more modules
FILE TYPE MISSING	CCI	Parameter missing
FILE TYPE ERROR	CCI	Parameter following <name> is not /S.
FIRST FILE CODE ERROR	CCI	Erroneous file code
FIRST FILE CODE MISSING	CCI	File code of disk to be copied not specified in command
FIRST FILE CODE UNKNOWN	CCI	File code not known by system
FOR O/R <address>	ASM	Forward reference contained error: - value > 255 for 8 least sign. bits - value specified was not absolute
I:	IPLRT	Initialization complete request for control message. Answer with WM or LD or ST
*I	ASM	Illegal identifier
*****I	ASM	IDENT directive missing
IDENT <prog id><address>	BOM/IPL	Name and first address of loaded program
IDENT MISSING	CCI	IDENT record missing
IDENT TOO LONG	LKE	IDENT name too long
IDT.MIS	LKE	IDENT record missing
ILLEGAL EOS IN INPUT FILE	CCI	First record of the module is EOS
INPUT I/O ERROR	CCI	Input I/O error
INPUT COMMAND I/O ERROR	DOM	I/O error in user identification
INPUT DISK I/O ERROR	CII	Error from the specified disc
INPUT FILE ASSIGN ERROR	CCI	Wrong assignment
INPUT FILE I/O ERROR	CCI	Input file I/O error
INPUT FILE CANNOT BE ASSIGNED	CCI	Temporary work file cannot be assigned. The message is followed by the reason.
INVALID DISK FILE CODE	CCI	Second parameter not in range /F0 to /FF.

Message	Program	Meaning
INVALID DISK ADDRESS	CCI	Wrong address specified
INVALID DISK TYPE	CCI	Disc not supported by system
INVALID FILE CODE	CCI	Wrong file code specified
INVALID NAME	CCI	Module name not accepted
INVALID PARAM	CCI	Parameter not in range /01 to /EF.
INVALID USERID	CCI	User identification does not begin with letter
INV.LGH name	CCI	Common block of this name too long
INV.IDT	LKE	Invalid IDENT record
I/O ER	LKE	I/O error encountered
I/O ERROR	CCI	I/O error. This message may follow DKER.
I/O ERROR <file><status>	ASM/LKE	I/O error encountered
I/O ERROR IN CATALOG	CCI	I/O error during this operation
I/O ERROR ON LAST RECORD. [TRY AGAIN]	CCI	Type a new command from /01
L:	IPLRT	Request for level to be used. Answer with LF CR or 2 digit hexa level number.
*L	ASM	Illegal label
L:	LKE	Request for option/control message. See page 9
L?	LKE	Erroneous option/control message
L= <hexa value>	LKE	Length of relocatable program section
/L ASSIGN ERROR	CCI	/L cannot be assigned
/L EMPTY	CCI	/L file empty
LFT OVERFLOW	CCI	Disc locigal file table overflow
LIBRARY OPTION REDUNDANT	CCI	Library option is redundant
LINE NUMBER ERROR	CCI	Wrong line number specified
*M	ASM	Unknown mnemonic
M:	MON	Request control message after INT button pressed
MAP OPTION REDUNDANT	CCI	MAP option is redundant
MISSING PARAMETER	CCI	Parameter not specified

Message	Program	Meaning
MODULE UNKNOWN	CCI	Object module unknown
NS	BOM	No start address defined
NL OPTION ERROR	CCI	NL more than once declared
NO LABEL	CCI	Label on tape absent
NO LOAD MODULE	CCI	No load module in file
NO OBJECT LIBRARY	CCI	Object library not found
NO STRT.	LKE	Invalid IDENT record
*O	ASM	Erroneous displacement value
*****O	ASM	Core overflow
/O ASSIGN ERROR	ASM	/O file cannot be assigned
/O CLOSE ERROR	CCI	Error during writing of EOF or re-winding of /O file
/O EMPTY	CCI	/O file empty
/O INPUT ERROR	CCI	Error during reading of /O
OBJECT LIBRARY ASSIGN ERROR	CCI	Work area for user object library cannot be assigned
OBJECT MODULE NAME ERROR	CCI	Wrong module name specified
OBJECT MODULE NOT CATALOGUED	CCI	Object module not catalogued
OBJECT TAPE ON READER THINK OF BASE	IPL	Place object tape to be loaded on the paper tape reader. Change program's base address if necessary
OUTPUT DISK I/O ERROR	CCI	I/O error on specified disk
OUTPUT FILE I/O ERROR	CCI	Output file I/O error
OUTPUT I/O ERROR	CCI	Output I/O error
OUTPUT NOT ASSIGNED	CCI	The /O2 file is assigned to NO device or not assigned
OV	BOM	Insufficient memory available
OVL	IPLRT	Insufficient memory available
OVT	IPLRT	Insufficient table area remaining
*P	ASM	Illegal parameter
PARAM ABSENT	CCI	Parameter not specified
PARAM ERROR	CCI	Parameter error or error in sector number
PCT POOL SIZE ?	IPLRT	Request for program control table size (type in 4 digit hexa number)

Message	Program	Meaning
PARAM MISSING	CCI	Request for program control table size (type in 4 digit hexa number)
PRG.OVL	CCI	Generated load module exceeds 32k
PROCESSOR NOT CATALOGUED	CCI	A segment of processor or compiler not catalogued
PROG ABORTED AT <address>	DOM	Program aborted at this address. The message is followed by the reason of the abort, the contents of PSW and the contents of registers. The reason may be: POWER FAILURE NOT WIRED INSTRUCTION MEMORY PROTECT BUFFER AREA DESTROYED TOO MANY SCHEDULED LABELS OPERATOR ABORTED BUFFER ALLOCATION OVERFLOW DISK OVERFLOW DISK QUEUE OVERFLOW MEMORY OVERFLOW DURING LOADING PHASE
PROGRAM NAME ERROR	CCI	Erroneous name specified
PROGRAM NOT CATALOGUED	CCI	Program not catalogued
PU.<device name and address> <hardware status>.[RY]	MON	Peripheral error. For hardware status see control unit status word configuration
*R	ASM	Illegal relocation
R:	ASM	Request correct assembly statement
R	DBG	Request input from external device
READ ONLY LENGTH:	DRTM	Define (4 hexa char.) Read-Only area length. Min. /800.
READ ONLY SAVE AREA:	DRTM	Define (4 hexa char.) length of Read-Only save area
*S	ASM	Illegal statement
S:	CCI	Request control command
S= <address>	LKE	Start address of module
/S CANNOT BE ASSIGNED	CCI	/S cannot be assigned
/S ASSIGN ERROR:	CCI	/D4 cannot be assigned to source file
/S EMPTY	CCI	The file to be punched is empty
SC:	IPLRT	Request time for initialization. HH MM SS or LF CR

Message	Program	Meaning
SD:	IPLRT	Request date for initialization. DD MM YY or LF CR
SECTOR DELETED	CCI	The sector was deleted previously
SECOND FILE CODE ERROR	CCI	Erroneous second file code specified
SECOND FILE CODE MISSING	CCI	File code of disc onto which is to be copied is missing
SECOND FILE CODE UNKNOWN	CCI	File code not known by system
SEGMENT NBR.01 MISSING	CCI	This parameter not specified
SEGMENT NBR NOT CATALOGUED	CCI	This segment was not catalogued or it is declared more than once
SEGMENT NBR ERROR	CCI	Erroneous segment
SEQUENCE ERR, [TRY AGAIN]	CCI	Update the lines in ascending order
START ADDR. REDUNDANT	CCI	Start address is redundant
SYNTAX ERR, [TRY AGAIN]	CCI	Error in Update command
SYSTEM DISK I/O ERROR	CCI	I/O error
SYSTEM LIB ASSIGN ERROR	CCI	Assign error in system library
SYSTEM TAPE ON READER, PLEASE	IPL	Load request for system program
SYSTEM SESSION COMMAND	CCI	LIC (List Catalogue) may only be used in system session
TBL.OVL	LKE	Not enough space to link-edit modules
TIME:	DOM/ DRTM	Disc system asking for time (h, m, s, or LF CR)
T.O.	LKE	Insufficient table area remaining
TOO MANY FILE CODE EQU	CCI	More than 7 file codes assigned to one disc file
TOO MANY MODULES TO BE ASSIGNED	CCI	More than 18 modules to be assigned
TOO MANY PARAM	CCI	Too many parameters specified
U:	UPD	Request control message. See page 10
U?	UPD	Erroneous control message
UND.ENT <number>	ASM	Number of undefined entry points
UND.LAB <references>	ASM	Missing labels in module
UNKNOWN COMMAND [TRY AGAIN]	CCI	Erroneous update command given

Message	Program	Meaning
UNKNOWN USERID	CCI	Specified userid not recognised by system
UNS.EXT.	LKE	One or more unsatisfied external references
USER DISK I/O ERROR	CCI	I/O error
USER LIB ASSIGN ERROR	CCI	Erroneous assignment
USERID:	DOM	Request for identification. Reply with <disc no>,<userid> or <userid>.
USERID ABSENT	CCI	No userid given in command
USERID ALREADY CATALOGUED	CCI	CCI specified userid was not catalogued
USERID ERROR	CCI	Error in userid or the first parameter is not a userid
USERID MISSING	CCI	No parameter given
USERID NOT CATALOGUED	CCI	The userid was not catalogued
USERID UNKNOWN	CCI/DOM	userid not found on the disc
*X	ASM	Illegal expression
XN	LKE	Unsatisfied external reference

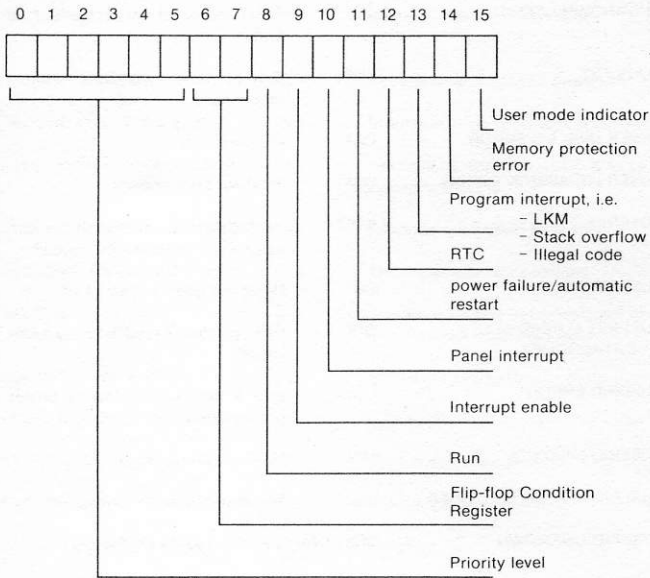
STANDARD PRIORITY INTERRUPT LEVELS

0	Power failure	interrupt location /40
1	LKM - stack overflow - illegal code	interrupt location /42
2	Real time clock	interrupt location /44
3	Memory Increment Data Break	interrupt location /46
4	Reserved	interrupt location /48
5	Common interrupt line	interrupt location /4A
6	Memory protect	interrupt location /4C
7	Control panel	interrupt location /4E

STANDARD CONNECTIONS COMMON INTERRUPT LINE

0	Reserved	(lowest priority)
1	Plotter	
2	Line printer	
3	Moving head disc 2 (not serviced by BOM)	
4	Moving head disc 1 (not serviced by BOM)	
5	Fixed head disc (not serviced by BOM)	
6	Magnetic tape	
7	(Operator's typewriter 2)	
8	Operator's typewriter 1	
9	(Tape punch 2)	
10	Tape punch 1	
11	Cassette tape	
12	(Punched tape reader 2)	
13	Punched tape reader 1	
14	Card reader	
15	Reserved	(highest priority)

PROGRAM STATUS WORD



bit 8, 9 and 13 are not copied into the stack but they are displayed on the control panel

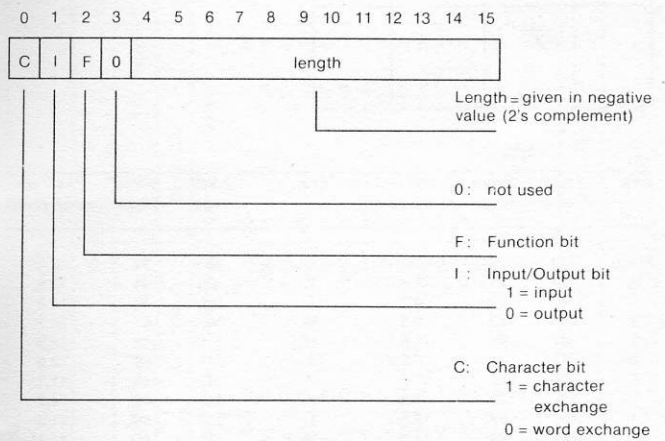
MEMORY PROTECTION

If a certain bit is set, the corresponding memory area is protected.

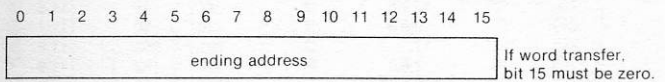
MK1			MK2		
bit	memory area	module	bit	memory area	module
15	0000 - 07FE	MM0	15	8000 - 87FE	MM4
14	0800 - 0FFE		14	8800 - 8FFE	
13	1000 - 17FE		13	9000 - 97FE	
12	1800 - 1FFE		12	9800 - 9FFE	
11	2000 - 27FE	MM1	11	A000 - A7FE	MM5
10	2800 - 2FFE		10	A800 - AFFE	
9	3000 - 37FE		9	B000 - B7FE	
8	3800 - 3FFE		8	B800 - BFFE	
7	4000 - 47FE	MM2	7	C000 - C7FE	MM6
6	4800 - 4FFE		6	C800 - CFFE	
5	5000 - 57FE		5	D000 - D7FE	
4	5800 - 5FFE		4	D800 - DFFE	
3	6000 - 67FE	MM3	3	E000 - E7FE	MM7
2	6800 - 6FFE		2	E800 - EFFE	
1	7000 - 77FE		1	F000 - F7FE	
0	7800 - 7FFE		0	F800 - FFFE	

MULTIPLEX CONTROL WORDS

1st word



2nd word



The control unit number = multiplex control word couple number

FIXED MEMORY LOCATIONS

Bootstrap	:	/0 thru /3E
Hardware interrupt locations	:	/40 thru /7E
Interrupt Table address	:	/80
Communication Vector Table address	:	/82
Multiplex control words	:	/84 thru /BE
Hardware interrupt locations	:	/C0 thru /DE
Stack	:	/100

Powers of 16

Powers of 16		Powers of 2	
16^n	n	2^n	n
1	0	256	8
16	1	512	9
256	2	1 024	10
4 096	3	2 048	11
65 536	4	4 096	12
1 048 576	5	8 192	13
16 777 216	6	16 384	14
268 435 456	7	32 768	15
4 294 967 296	8	65 536	16
68 719 476 736	9	131 072	17
1 099 511 627 776	10	262 144	18
17 592 186 044 416	11	524 288	19
281 474 976 710 656	12	1 048 576	20
4 503 599 627 370 496	13	2 097 152	21
72 057 594 037 927 936	14	4 194 304	22
1 152 921 504 606 846 976	15	8 388 608	23
18 446 744 073 709 551 616	16	16 777 216	24

ASCII CODE

space	• • •	D	• • •
!	• • •	E	• • •
"	• • •	F	• • •
#	• • •	G	• • •
\$	• • •	H	• • •
%	• • •	I	• • •
&	• • •	J	• • •
'	• • •	K	• • •
(• • •	L	• • •
)	• • •	M	• • •
*	• • •	N	• • •
+	• • •	O	• • •
,	• • •	P	• • •
-	• • •	Q	• • •
.	• • •	R	• • •
/	• • •	S	• • •
0	• • •	T	• • •
1	• • •	U	• • •
2	• • •	V	• • •
3	• • •	W	• • •
4	• • •	X	• • •
5	• • •	Y	• • •
6	• • •	Z	• • •
7	• • •	[• • •
8	• • •	*/	• • •
9	• • •]	• • •
:	• • •	↑	• • •
;	• • •	**←	• • •
<	• • •	CR	• • •
=	• • •	Line feed	• • •
>	• • •	X on	• • •
?	• • •	Bell	• • •
@	• • •	X off	• • •
A	• • •		
B	• • •		
C	• • •		

* delete record
 ** delete character (EOR)