PHILIPS

P850/P855/P860 Reference Data

Second draft

data systems

P850/P855/P860

Reference Data

Second Draft

P H I L I P S

-ORIGINAL-

P850/P855/P860

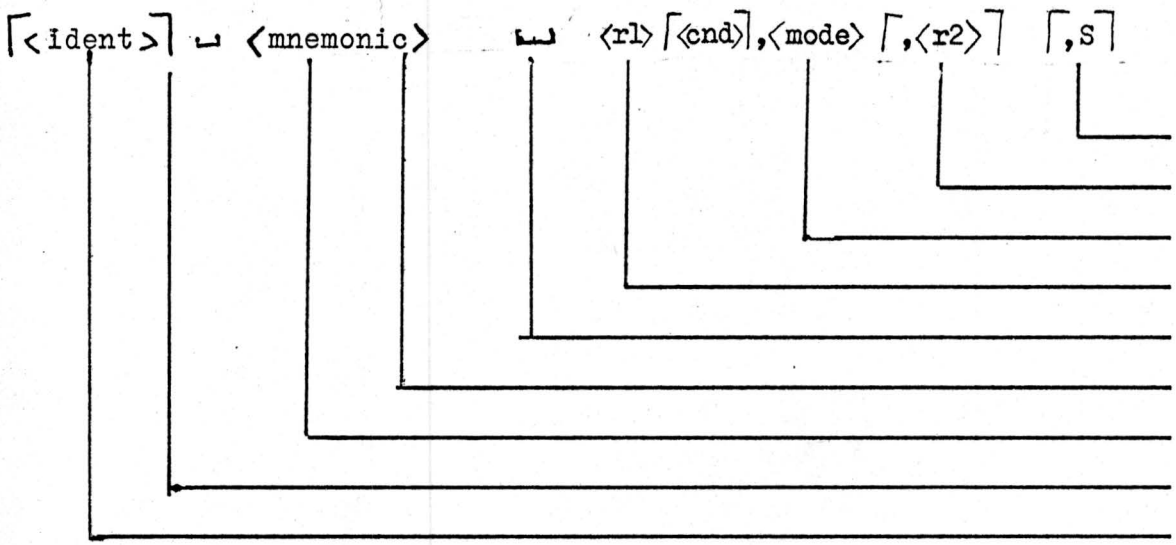
Reference Data

PHILIPS

data systems

Contents

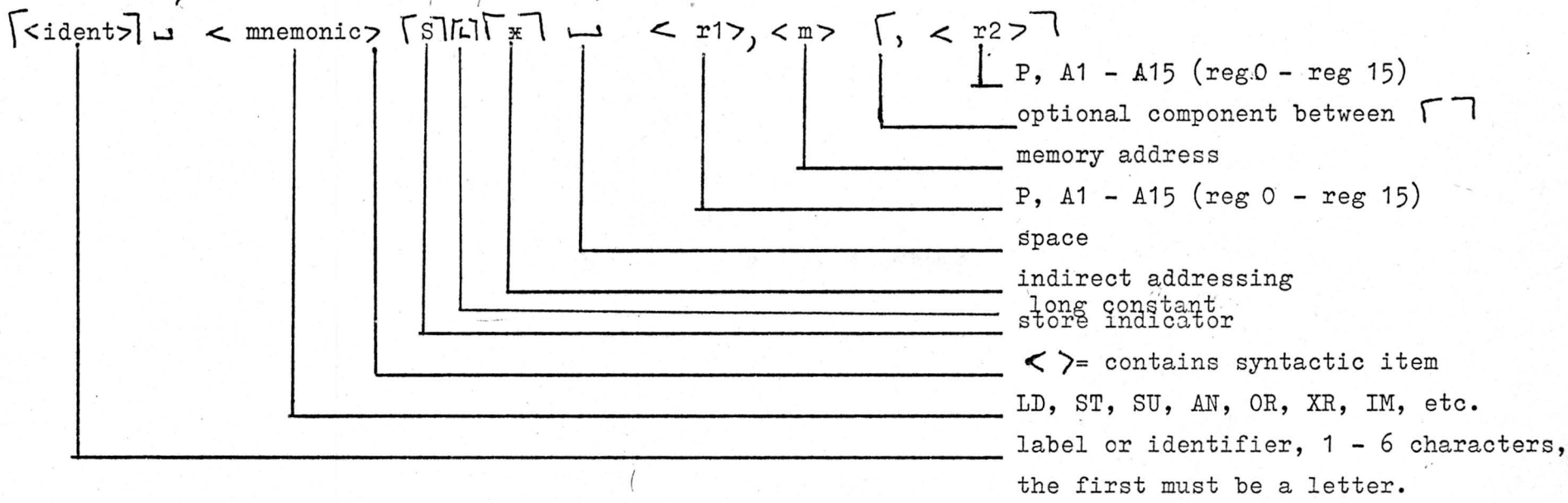
Assembly language P850 - general format - [1]



* in operand field represents current value of location counter

1) For details, see P850 User Manual (publication number 5122 991 1453X)

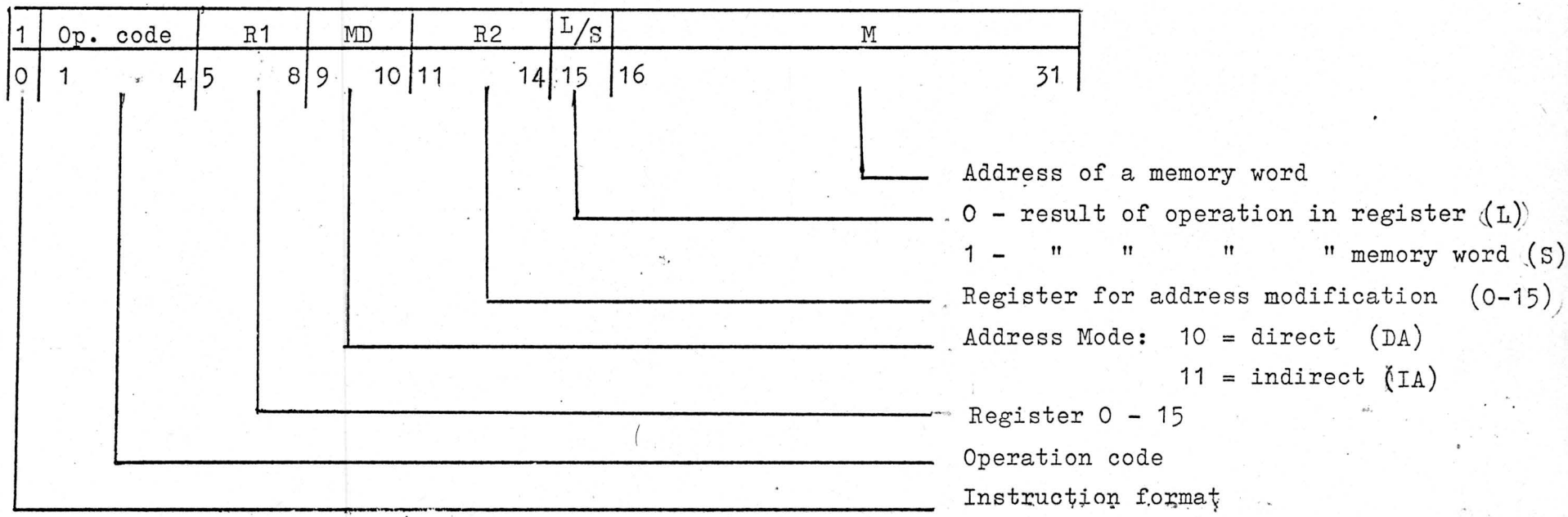
Assembly language P855/P860 - general format - [2]



* in operand field represents current value of location counter
(except when immediately after mnemonic)

[2] For details, see P855/P860 Reference Manual (publication number 5122 991 1835X)

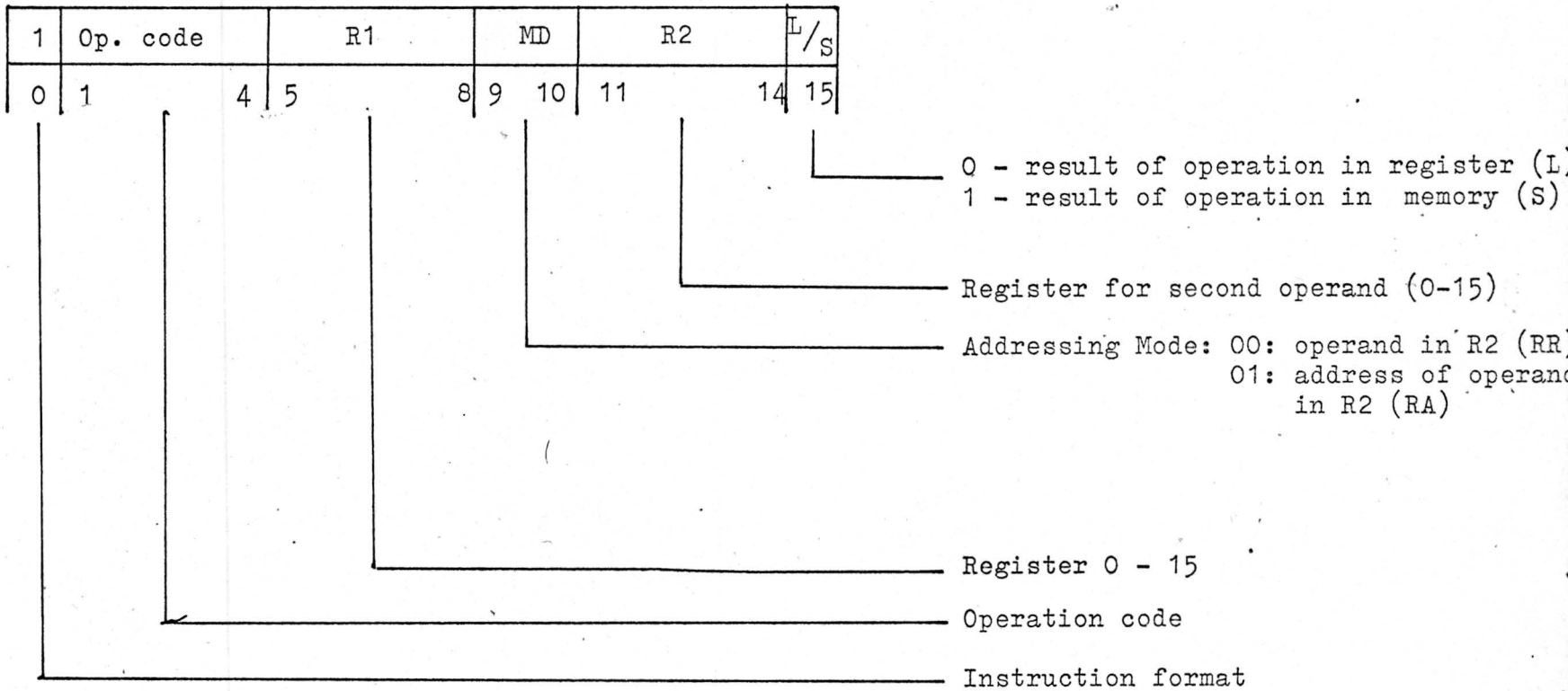
Memory Reference Instructions



P855/60-P850
ABI - AB
AD - AD
AN - I
C1 - C1
C2 - C2
CC - -
CFI - CF
CW - CW
DA - -
DS - -
DV - -
IM - IM

P855/60-P850
LC - TC
LD - TW
ML - -
MS - -
MU - -
OR - U
RB - RB
RF - RF
SC - TC
ST - TW
SU - SU
XR - X

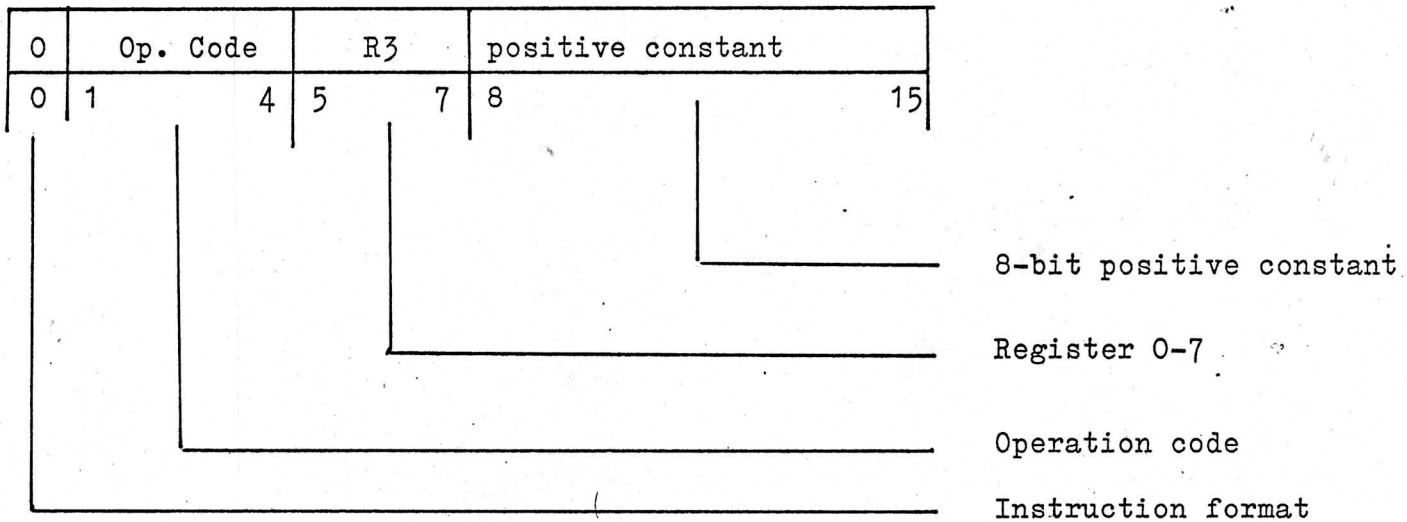
Register / Register Instructions



P855/60	P850
ABR	AB
ADR	AD
ANR	I
C1R	C1
C2R	C2
CCR	-
CFR	CF
CWR	CW
DAR	-
DSR	-
DVR	-
ECR	EC
IMR	IM

P855/60	P850
LCR	TC
LDR	TW
MLR	-
MSR	-
MUR	-
ORR	U
RTN	RT
SCR	TC
STR	TW
SUR	SU
TM	I
TNM	U
XRR	X

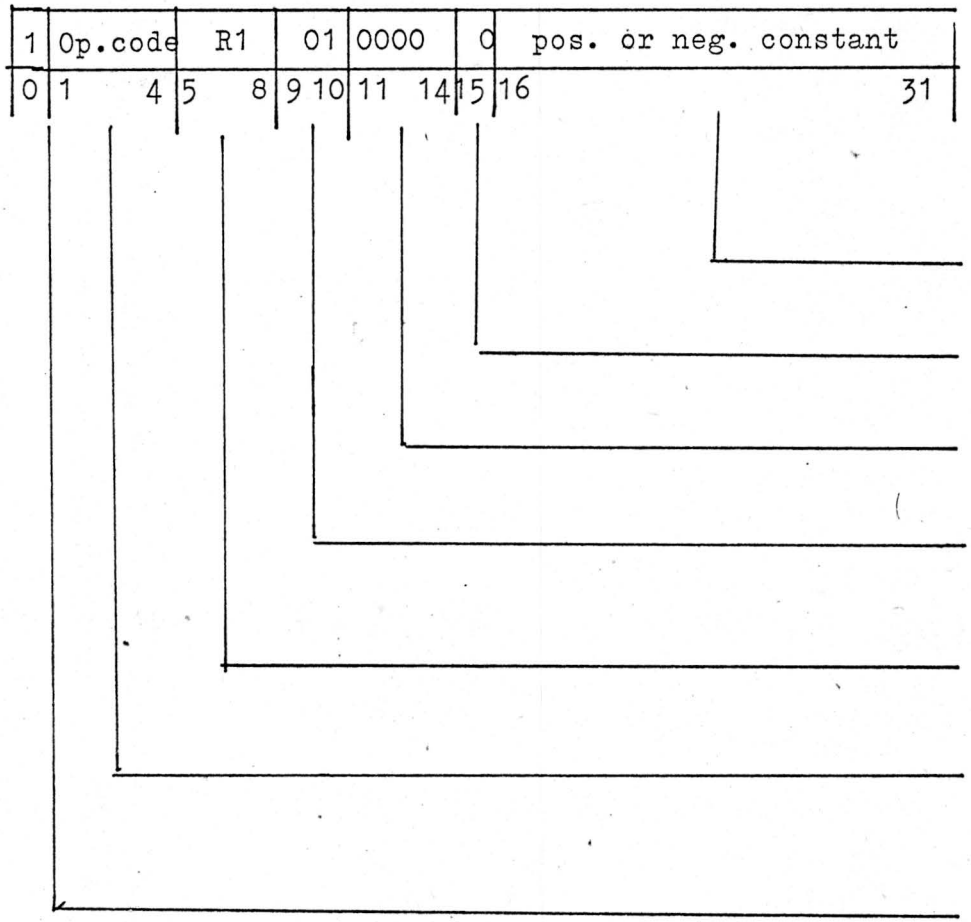
Constant instructions (short format)



P855/60 - P850

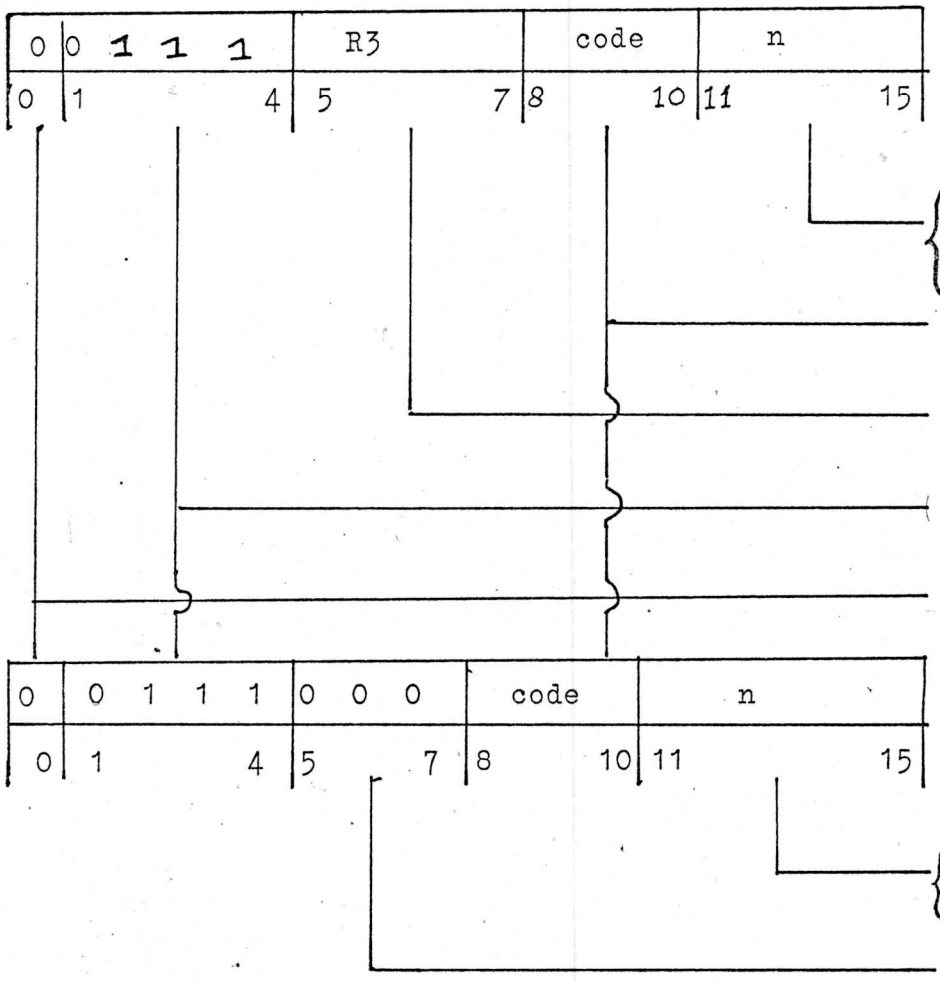
AB	BK
ADK	AK
ANK	IK
LDK	LK
ORK	UK
SUK	SK
XRK	XK

nstant Instructions (long format)

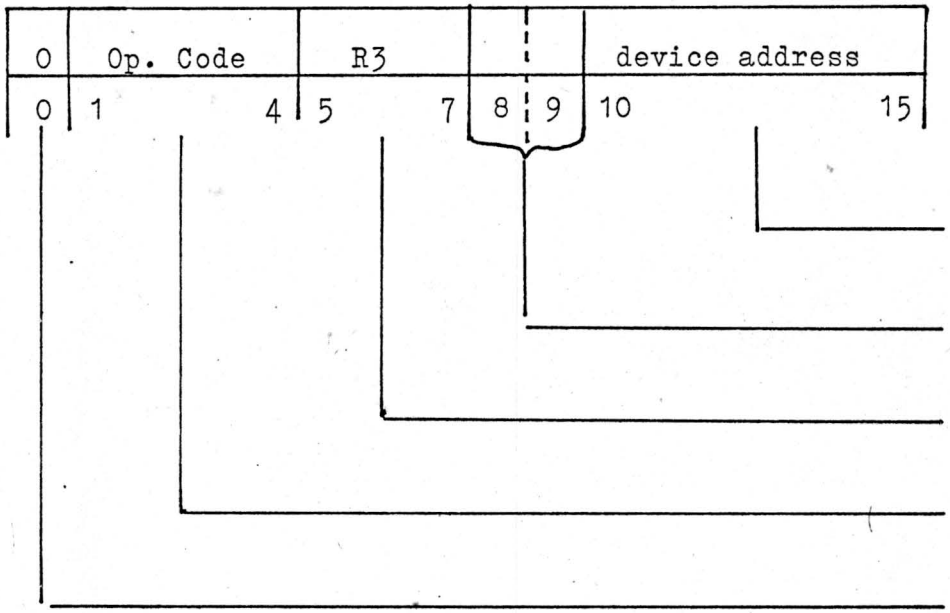


P855/60	P850
ABL	AB
ADKL	AD
ANKL	I
CCK	-
CF	CF
CWK	CW
DAK	-
DSK	-
DVK	-
LDKL	TW
LCK	TC
MLK	-
MUK	-
ORKL	U
SUKL	SU
XRKL	X

Shift Instructions



P855/60	P850
DLA	-
DLC	-
DLL	-
DLN	-
DRA	-
DRC	-
DRL	-
DRN	-
SLA	-
SLA1	LA
SLC	-
SLL	-
SLL1	LL
SLN	-
SRA	-
SRA1	RA
SRC	-
SRC1	RC
SRL	-
SRN	-



Device address (000000 for WMP, RIL, WIM, WM2 and RCA instructions)

Variable, see appropriate instruction for details

Register 0-7

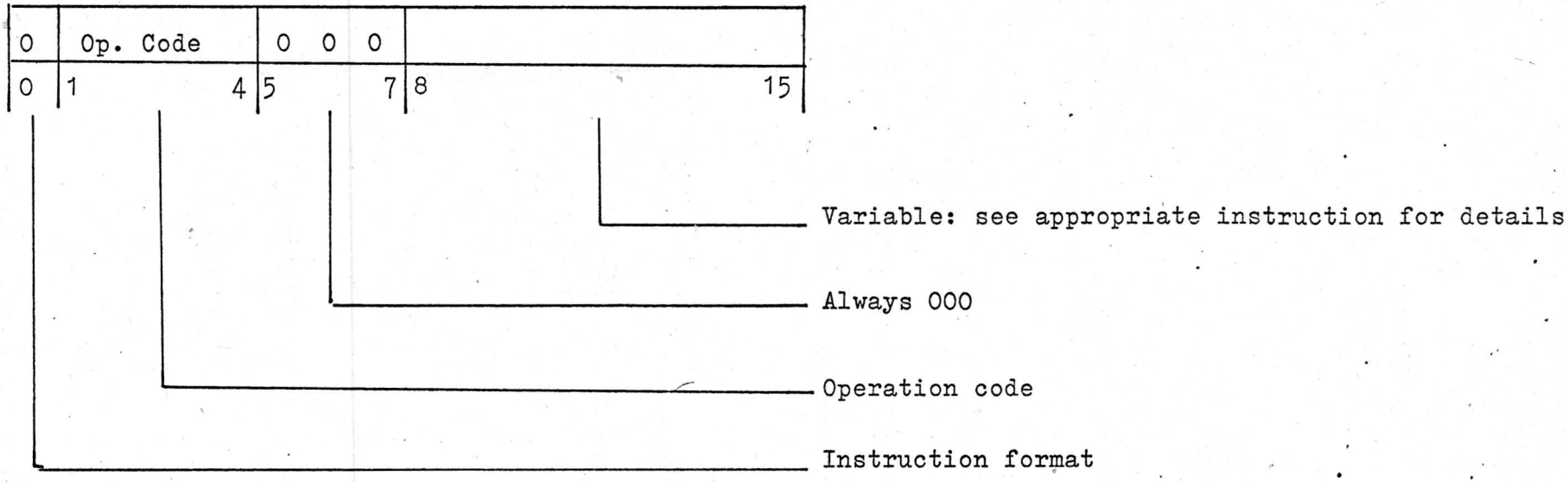
Operation code

Operation format

P855/60	- P850 *)	
CIO	CT	
INR	IN	IN
OTR	OT	
RCA	-	
RIL	RL	
SST	SS	
TST	TS	
WIM	WM	
WM2	-	
WMP	-	

*) Mnemonics are not recognized by the Assembler

Miscellaneous instructions



P855/60	P850 *)
ENB	EN
HLT	HT
INH	IH
LKM	LM
RIT	RI
SMD	-

*) Mnemonics are not recognized by the Assembler

Addressing Modes - Reference Types

Mode	MD 9	MD 10	R2	Effective Addresses	Meaning
RR	0	0	xxxx	R2	Register-to-register (2nd operand in R2)
RA	0	1	0000	(P)	Long constant. (word following the instruction is a 16-bit operand).
RA	0	1	≠0	(R2)	Address in register (effective address is in R2)
DA	1	0	0000	(q)	Address in next word
DA	1	0	≠0	(q) + (R2)	Indexed address (address in next word is modified by contents of R2).
IA	1	1	0000	[(q)]	Indirect address
IA	1	1	≠0	[(q) + (R2)]	Indirect indexed address

P = instruction counter
q = contents of P (i.e. word after instruction)
R2 = bits 11 -14 of instruction word
() = contents of
[] = indirect addressing.

File codes to be used with standard software

a. P850 ~~Xxfbxxxxx~~ File codes

01 = ASR Keyboard
02 = ASR Reader
03 = ASR Punch
04 = Punched Tape Reader
05 = Tape Punch

b. P855/P860 ~~XXXXXXXX~~ File codes- Basic

01 = Standard Source Input
02 = Standard Listing Output
03 = Standard Punch Output
04 = Standard Object Input
05 = Standard Operator Keyboard
06 = ASR Reader
07 = ASR Punch
08 = Punched Tape Reader
09 = Tape Punch
0A = Line Printer
0B = Card Reader
0C = Cassette Tape.

c. P855/P860 File codes - Disc

01 = Typewriter

02 = Print unit

03 = Punch unit

D4 = /S
D5 = /O } for DOS
D6 = /L

D7 = System object file

D8 = User object file

E0 = Control command input

E1 = Source input

E2 = Object input

EF = Typewriter (display only)

F0 - FF = Disc units

Directives

⌈<ident>⌉ ␣ DATA ␣ <data expression> ⌈,⟨data expression⟩, ...,⌉ (up to 16 words generated)

⌈<ident>⌉ ␣ EQU ␣ <predefined expression>

␣ IDENT ␣ <program name>

⌈<ident>⌉ ␣ END ␣ ⌈<predefined expression>⌉ ⌈,<symbol>⌉

⌈<ident>⌉ ␣ RES ␣ <predefined absolute expression> ⌈,⟨predefined absolute expression⟩⌉

␣ AØRG ␣ <predefined absolute expression⟩

␣ RØRG ␣ ⌈<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>⌉ ␣ CØMN ␣ <common field definition list>

<ident> ␣ FØRM ␣ <field definition> ⌈,<field definition> ...., <field definition>⌉

⌈/<field number list>⌉

<ident> ␣ XFØRM ␣ <FORM defined pseudo-mnenonic>, <field list>.

␣ GEN ␣ ⌈A⌉

␣ ALIT ␣ ⌈⟨predefined expression⟩⌉ ⌈,R⌉

⟨data expression⟩ ::= ⟨expression⟩ | ⟨character string⟩

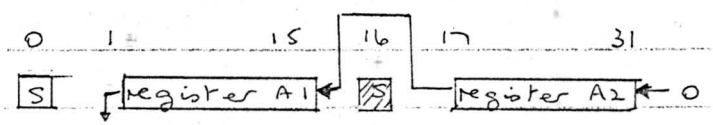
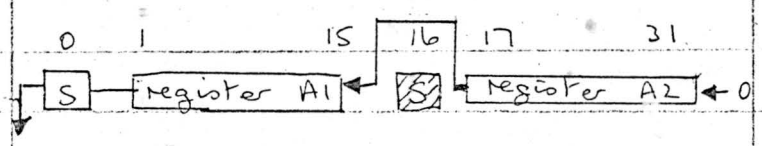
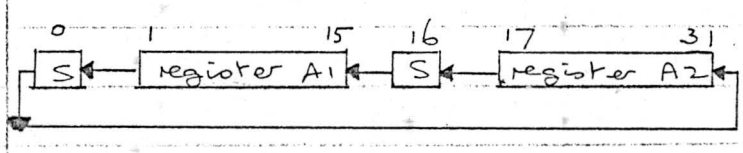
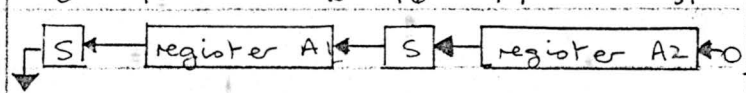
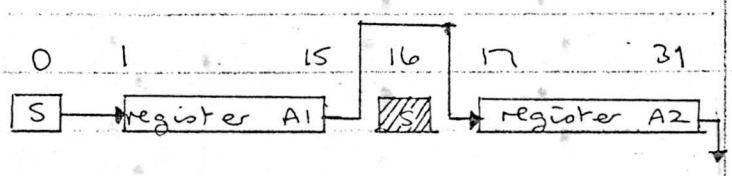
⟨field definition⟩ ::= ⟨field length definition⟩ ⌈=⟨field value definition⟩⌉

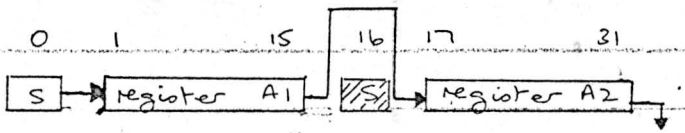
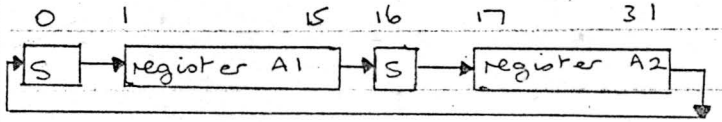
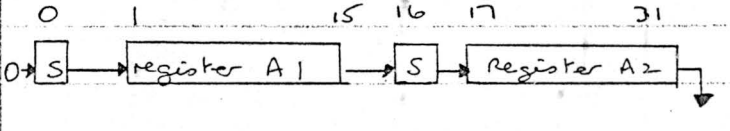
name (in alphabetical order)	mnemonic P850	mnemonic P855/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
Absolute branch	AB	ABI	1	0001	10	0	(M) → P		7		3	bits 5-7: condition
					10	0	(M + (R2)) → P		7		3	bit 8: n.s.
					11	0	((M)) → P		9		4	
					11	0	((M + (R2))) → P		9		4	
						0	no branch: (P) + 4 → P		3	1	1	
Absolute conditional branch (with constant)	BK	AB	0	0001	–	n.s.	K → P (q) → P	3)	4	1	1	short format
							no branch: (P) + 2 → P		3	1	1	
Absolute conditional branch (with constant)	AB	ABL	1	0001	01	0	KL → P		5	1	2	long format
							no branch: (P) + 2 → P		3	1	1	
Absolute conditional branch to register	AB	ABR	1	0001	00	n.s.	(R2) → P		4	1	1	bits 5-7: condition
					01	0	((R2)) → P		5		2	bit 8: n.s.
							no branch: (P) + 2 → P		3	1	1	
Add constant	AK	ADK	0	0010	–	–	(R3) + K → R3		4	1	1	short, R1 = 1111 system mode
	AD	ADKL	1	0010	01	0	(R1) + KL → R1		5		2	long
Addition	AD	AD	1	0010	10	0	(R1) + (M) → R1		7		3	R1 = 1111: system mo
					10	1	(R1) + (M) → M		8		4	when L/S bit = 1,
					10	0	(R1) + (M + (R2)) → R1		7		3	R1 must be ≠ 0,
					10	1	(R1) + (M + (R2)) → M		8		4	
					11	0	(R1) + ((M)) → R1	2)	9		4	
					11	1	(R1) + ((M)) → M		10		5	
					11	0	(R1) + ((M + (R2))) → R1		9		4	

name (in alphabetical order)	mnemonic P850	mnemonic P855/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
					11	1	(R1) + ((M+(R2))) → M		10		5	
Addition register/register	AD	ADR	1	0010	00	n.s.	(R1) + (R2) → R1	2)	4	1	1	when L/S bit = 1,
					01	0	(R1) + ((R2)) → R1		5		2	R1 must be ≠ 0;
					01	1	(R1) + ((R2)) → (R2)		6		3	R1 = 1111: system mo
Call function	CF	CFI	1	1110			(P) → (R1), (R1) − 2 → R1 (P855/P860)					R1 = 1111: System Mo
							(PSW) → (R1), (R1) − 2 → R1					
							(CR) → (R1), (R1) − 2 → R1 (P850)					
							(P) → (R1), (R1) − 2 → R1					
							then:					
					10	1	(M) → P		13	2	5	if reg. 15 = stack
					10	1	(M + (R2)) → P		13	2	5	pointer,
					11	1	((M)) → P		15	2	6	$>128_{10}$ → overflo
					11	1	((M + (R2))) → P		15	2	6	
Call function/constant	CF	CF	1	1110	01	1	(P) → (R1), (R1) − 2 → R1 (P855/60)		11	2	4	if reg. 15 = stac
							(PSW) → R1, (R1) − 2 → R1,					pointer
							KL → P	3)				$>128_{10}$ → overflow
							(CR) → R1, (R1) − 2 → R1 (P850)					
							(P) → R1, (R1) − 2 → R1					R1 = 1111: System Mo
							KL → P					
Call function/register	CF	CFR	1	1110			(P) → (R1), (R1) − 2 → R1 (P855/60)					if reg. 15 = stac
							(PSW) → (R1), (R1) − 2 → R1					pointer,
							(CR) → (R1), (R1) − 2 → R1 (P850)					128: stack overf

name (in alphabetical order)	mnemonic P850	mnemonic 55/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
							(P) → (R1), (Rt) -2 → R1 } P850					
							then:					R1 = 1111: System m
					00	1	(R2) → P	3)	10	3	3	
					01	1	((R2)) → P		11	2	4	
Compare characters	-	CC	1	1101	10	1	((R1)r ÷ (M) 1/r → CR				4	R1 = 1111 : System M
					10	1	(R1)r ÷ (M+(R2)) 1/r → CR				4	
					11	1	(R1)r ÷ ((M)) 1/r → CR				5	
					11	1	(R1)r ÷ ((M+(R2))) 1/r CR				5	
Compare characters register / register	-	CCR	1	1101	01	1	(R1)r ÷ ((R2)) 1/r → CR				3	R1 = 1111: System m
Compare character with constant	-	CCK	1	1101	01	1	(R1)r ÷ KL1 → CR	4)	-		3	R1 = 1111: System M
Compare words	CW	CW	1	1101	10	0	(R1) ÷ (M) → CR		7		3	R1 = 1111: System M
					10	0	(R1) ÷ (M+(R2)) → CR		7		3	
					11	0	(R1) ÷ ((M)) → CR		9		4	
					11	0	(R1) ÷ ((M+(R2))) → CR		9		4	
compare words register/register	CW	CWR	1	1101	00	n.s.	(R1) ÷ (R2) → CR		4	1	1	R1 = 1111 : System mo
					01	0	(R1) ÷ ((R2)) → CR		5		2	
Compare word with constant	CW	CWK	1	1101	01	0	(R1) ÷ KL → CR		5		2	R1 = 1111 : system mo

name (in alphabetical order)	mnemonic P850	mnemonic 55/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
control Input/Output	CT	CIO	0	1000	–	–	Start (bit 9=1) or stop (bit 9=0) any I/O operation	5)	4		3	bit 8 = 1 system mode
							quotient | remainder				
divide	–	DV	1	1001	10	0	(A1, A2) / (M) → A2 | A1		–	10.5	3	
					10	0	(A1, A2) / (M+(R2)) → A2 | A1		–	10.5	3	
					11	0	(A1, A2) / ((M)) → A2 | A1		–	10.5	4	
					11	0	(A1, A2) / ((M+(R2))) → A2 | A1		–	10.5	4	
							quotient | remainder				
divide by constant	–	DVK	1	1001	01	0	(A1, A2) / KL → A2 | A1		–	10.5	2	
							quotient | remainder				
divide registers/registers	–	DVR	1	1001	00	0	(A1, A2) / (R2) → A2 | A1		–	11.5	1	
					01	0	(A1, A2) / ((R2)) → A2 | A1		–	10.5	2	
double Add	–	DA	1	1010	10	0	(M, M+1) + (A1, A2) → A1, A2	2)	–	1	4	
					10	0	(M+(R2), M+(R2)+1) + (A1, A2) → A1, A2		–	1	4	
					11	0	((M), (M)+1) + (A1, A2) → A1, A2		–	1	5	
					11	0	((M+(R2)), ((M+(R2)+1) + (A1, A2) → A1, A2		–	1	5	
double add registers/registers	–	DAR	1	1010	00	0	(R2, R2+1) + (A1, A2) → A1, A2		–	3	1	
					01	0	((R2), (R2+1)) + (A1, A2) → A1, A2		–	1	3	
double add with constant	–	DAK	1	1010	01	0	KL + (A1, A2) → A1, A2		–	1	3	
double subtract	–	DS	1	1011	10	0	(A1, A2) - (M, M+1) → A1, A2		–	1	4	
					10	0	(A1, A2) - (M+(R2), M+(R2)+1) → A1, A2		–	1	4	

name (in alphabetical order)	mnemonic P850	mnemonic P855/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
					11	0	(A1,A2) − ((M), (M)+1) → A1, A2		–	1	5	
					11	0	(A1, A2) − ((M+(R2)), ((M+(R2))+1) → A1, A2		–	1	5	
Double subtract registers/ registers	–	DSR	1	1011	00	0	(A1, A2) − (R2, R2+1) → A1, A2	2)	–	3	1	
					01	0	(A1, A2) − ((R2), (R2+1)) → A1, A2		–	1	3	
Double subtract with constant	–	DSK	1	1011	01	0	(A1, A2) − KL → A1, A2		–	1	3	
Double left and Normalize shift	–	DLN	0	0111	–	n.s.		3)	–	$(1.5+\frac{n}{2})$	1	bits 8-10 : 100 bits 11-14 : R2; bit 15 : n.s.
Double left arithmetic shift	–	DLA	0	0111	–	–		2)	–	$(1.5+\frac{n}{2})$		bits 8-10 : 000
Double left circular shift	–	DLC	0	0111	–	–		1)	–	$(1.5+\frac{n}{2})$	1	bits 8-10 : 110
Double left logical shift	–	DLL	0	0111	–	–			–	$(1.5+\frac{n}{2})$	1	bits 8-10 : 010
Double right and normalize shift	–	DRN	0	0111	–	–		3)	–	$(1.5+\frac{n}{3}$	1	bits 8-10 : 101 bits 11-14 : R2, bit 15 : n.s.

name (in alphabetical order)	mnemonic P8505	mnemonic P850/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
Double right arithmetic shift	–	DRA	0	0111	–	–				$(1.5+\frac{n}{2})$	1	bits 8-10: 001
Double right circular shift	–	DRC	0	0111	–	–		1)	–	$(1.5+\frac{n}{2})$	1	bits 8-10: 111
Double right logical shift	–	DRL	0	0111	–	–			–	$(1.5+\frac{n}{2})$	1	bits 8-10: 011
Enable interrupt	EN 7)	ENB	0	0101	–	–	machine status = "permit interrupt"	3)	4	1	1	bits 8-15: 01000000
Exchange characters register/register	EC	ECR	1	1100	00	n.s.	$(R2) \rightarrow (R1)_{R}$; $(R2)_{R} \rightarrow (R1)$		5	1	1	R1 = 1111: System mode
Exclusive OR	X	XR		0110	10	0	$(R1) \neq (M) \rightarrow R1$		7		3	R1 = 1111: System mode
					10	1	$(R1) \neq (M) \rightarrow M$		8		4	
					10	0	$(R1) \neq (M+(R2)) \rightarrow R1$		7		3	
					10	1	$(R1) \neq (M+(R2)) \rightarrow M+(R2)$		8		4	
					11	0	$(R1) \neq ((M)) \rightarrow R1$		9		4	
					11	1	$(R1) \neq ((M)) \rightarrow (M)$	1)	10		5	
					11	0	$(R1) \neq ((M+(R2))) \rightarrow R1$		9		4	
					11	1	$(R1) \neq ((M+(R2))) \rightarrow (M+(R2))$		10		5	

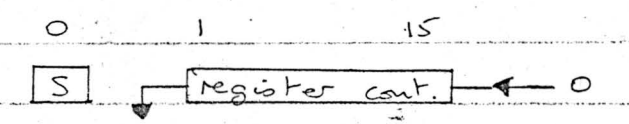
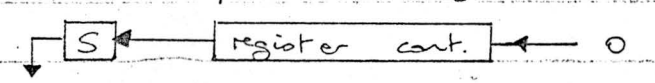
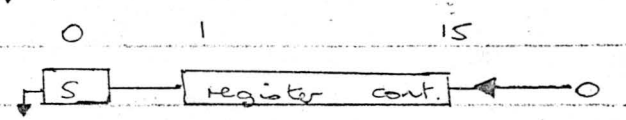
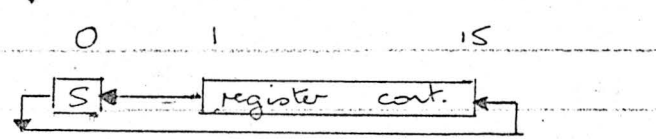
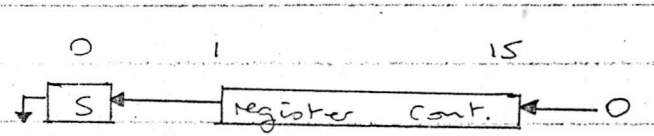
name (in alphabetical order)	mnemonic P850	mnemonic 55/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
Exclusive OR	X	XRR	1	0110	00	0	(R1) ∀ (R2) → R1		4	1	2	R1=1111: System mode
register / register					01	0	(R1) ∀ ((R2)) → R1		5		2	
					01	1	(R1) ∀ ((R2)) → R2	1)	6		3	
Exclusive OR	XH	XRH	0	0110	–	–	$(R3)_{8-15}$ ∀ H → $R3_{8-15}$		4	1	1	short; R1=1111: system mo
with constant	X	XRHL	1		01	0	(R1) ∀ HL → R1		5		2	long " " "
Halt	HT 7)	HLT	0	0100	–	–	machine → "halt" mode	3)	4	1	1	bits 8-15: 01111111; {system mo
Increment Memory	IM	IM	1	0010	10	1	(M)+1 → M		8		4	bits 5-8: 0000
					10	1	(M+(R2))+1 → M		8		4	
					11	1	((M+))+1 → M	2)	10		5	
					11	1	((M)+(R2)))+1 → (M+(R2))		10		5	
Increment Memory / register	IM	IMR	1	0010	01	1	((R2))+1 → R2		6		3	bits 5-8: 0000
Inhibit interrupt	IH 7)	INH	0	0100	–	–	machine status = "prohibit all interrupts except power failure"	3)	4	1	1	bits 8-15: 10111111 system mode
Input to register	IN 7)	INR	0	1001	–	–	word/character from device → R3	5)	4		3	bit 8=0; system mo
Link to monitor	LM 7)	LKM	0	0101	–	–	user mode → system mode		4	1	1	bits 8-15: 0000010
Load character	TC	LC	1	1100	10	0	(M) 4/R → $R3_R$		7		3	R1=1111: system mode
					10	0	(M + (R2)) 4/R → $R1_R$	3)	7		3	R1 must be ≠0
					11	0	((M)) 4/R → R1		9		4	
					11	0	((M + (R2))) 4/R → $R1_R$		9		4	

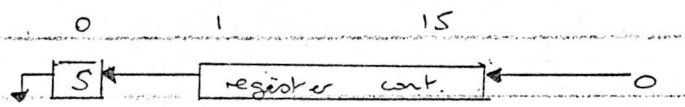
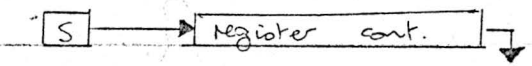
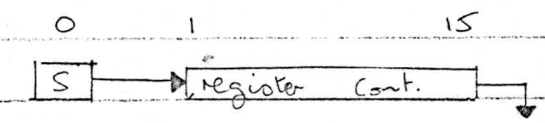
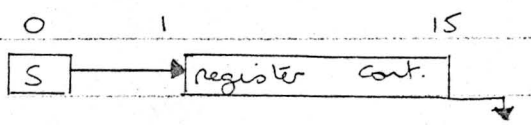
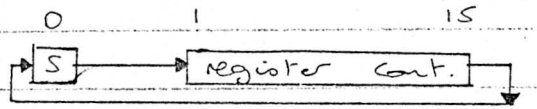
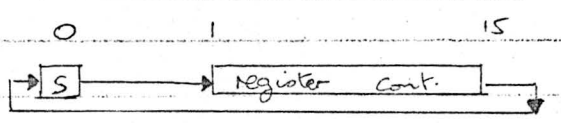
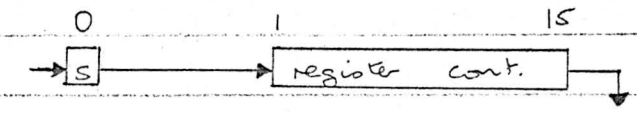
name (in alphabetical order)	mnemonic P850	mnemonic 55/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
Load character/constant	TC	LCK	1	1100	01	0	$KL_U \rightarrow R1_R$		5		2	R1=1111: system mode
Load character/register	TC	LCR	1	1100	01	0	$((R2))_{4/R} \rightarrow R1_R$	3)	5		2	R1=1111: system mode
Load constant	LK	LDK	0	0000	-	-	$K \rightarrow R3_{8-15}$, $0 \rightarrow R3_{0-7}$		4	1	1	short; R1=1111: system mode
	TW	LDKL	1	0000	01	0	$KL \rightarrow R1$		5		2	long; " "
Load register	TW	LD	1	0000	10	0	$(m) \rightarrow R1$		7		3	R1=1111 system mode
					10	0	$(m+(R2)) \rightarrow R1$		7		3	
					11	0	$((m)) \rightarrow R1$		9		4	
					11	0	$((m+(R2))) \rightarrow R1$		9		4	
Load register/register	TW	LDR	1	0000	00	n.s.	$(R2) \rightarrow R1$		4	1	1	R1=1111: system mode
					01	0	$((R2)) \rightarrow R1$		5		2	
					01	0	$(A15)+2 \rightarrow A15$, $((A15)) \rightarrow R1$		6	1	2	system mode
Logical AND	I	AN	1	0100	10	0	$(R1) \wedge (m) \rightarrow R1$	1)	7		3	R1=1111: system mode
					10	1	$(R1) \wedge (m) \rightarrow m$		8		4	
					10	0	$(R1) \wedge (m+(R2)) \rightarrow R1$		7		3	
					10	1	$(R1) \wedge (m+(R2)) \rightarrow m+(R2)$		8		4	
					11	0	$(R1) \wedge ((m)) \rightarrow R1$		9		4	
					11	1	$(R1) \wedge ((m)) \rightarrow (m)$		10		5	
					11	0	$(R1) \wedge ((m+(R2))) \rightarrow R1$		9		4	
					11	1	$(R1) \wedge ((m+(R2))) \rightarrow (m+(R2))$		10		5	
Logical AND register/register	I	ANR	1	0100	00	0	$(R1) \wedge (R2) \rightarrow R1$		4	1	1	R1=1111: system mode
					01	0	$(R1) \wedge ((R2)) \rightarrow R1$		5		2	

name (in alphabetical order)	mnemonic P850	mnemonic 55/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
					01	1	$(R1) \wedge ((R2)) \rightarrow (R2)$		6		3	
Logical • AND	IK	ANK	0	0100	–	–	$(R3)_{8-15} \wedge K \rightarrow R3_{8-15}$		4	1	1	short; R1=1111: system mod-
with constant	I	ANKL	1		01	0	$(R1) \wedge KL \rightarrow R1$		5		2	long; " "
Logical OR	U	OR	1	0101	10	0	$(R1) \vee (m) \rightarrow R1$		7		3	R1 = 1111: system mod
					10	1	$(R1) \vee (m) \rightarrow m$		8		4	
					10	0	$(R1) \vee (m+(R2)) \rightarrow R1$		7		3	
					10	1	$(R1) \vee (m+(R2)) \rightarrow m+(R2)$		8		4	
					11	0	$(R1) \vee ((m)) \rightarrow R1$		9		4	
					11	1	$(R1) \vee ((m)) \rightarrow (m)$		10		5	
					11	0	$(R1) \vee ((m+(R2))) \rightarrow R1$	1)	9		4	
					11	1	$(R1) \vee ((m+(R2))) \rightarrow (m+(R2))$		10		5	
Logical OR	U	ORR	1	0101	00	0	$(R1) \vee (R2) \rightarrow R1$		4	1	1	R1=1111: system mod
register / register					01	0	$(R1) \vee ((R2)) \rightarrow R1$		5		2	
					01	1	$(R1) \vee ((R2)) \rightarrow (R2)$		6		3	
Logical OR	UK	ORK	0	0101	–	–	$(R3)_{8-15} \vee K \rightarrow R3_{8-15}$		4	1	1	short; R1=1111: system mode
with constant	U	ORKL	1		01	0	$(R1) \vee KL \rightarrow R1$		5		2	long; " "
Multiple load	–	ML	1	0111	10	0	$(m) \dots (m+n) \rightarrow A1 \dots An$		–		*	* 3+(n-1).[bits 5-8
					10	0	$(m+(R2)) \dots (m+(R2)+n) \rightarrow A1 \dots An$		–			
					11	0	$((m)) \dots ((m)+n) \rightarrow A1 \dots An$		–		**	** 4+(n-1).]
					11	0	$((m+(R2))) \dots ((m+(R2))+n) \rightarrow A1 \dots An$		–			n = 1111: system mod

name (in alphabetical order)	mnemonic P8505	mnemonic 55/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	execution time in cycles 6) P850 m.c.	P855/60 l.c.	P855/60 m.c.	remarks
Multiple load / constant	-	MLK	1	0111	01	0	KL1, KL2, --- KLn → A1, A2, --- An		-		n+1	n = 1111: system mode
Multiple load / register	-	MLR	1	0111	01	0	((R2)) → A1; ((R2)+2) → A2; ----- ;		-		n+1	n = 1111: system mode
							((R2) + 2n-2) → An					bits 5-8; n.
					-		(A15) + 2n → A15; ((A15)) → A1;		-	1	n+1	system mode;
							((A15)-2) → A2; ----- ;					bits 5-8: n
							((A15) - 2n+2 → An					
Multiple store	-	MS	1	0111	10	1	A1 --- An → M ... M+n		-	n	2+n	n = 1111: system mode
					10	1	A1 --- An → M+(R2) --- M+(R2)+n		-	n	2+n	bits 5-8: n
					11	1	A1 --- An → (M) --- (M)+n		-	n	2+n	
					11	1	A1 --- An → (M+(R2)) --- (M+(R2))+n		-	n	2+n	
Multiple store / register	-	MSR	1	0111	01	1	(A1) → (R2); (A2) → (R2)+2;	3)	-	n	n+1	n = 1111: system mode
							----; (An) → (R2) + 2n - 2					bits 5-8: n
							(A1) → (A15); (A2) → (A15) - 2n+2;			n	n+1	pointer 128$_{10}$: stack overf
							----; (An) → (A15) - 2n+2;					bits 5-8: n;
							(A15) - n → (A15)					system mode
Multiply	-	MU	1	1000	10	0	(A2) x (M) → A1, A2		-	8.5	3	
					10	0	(A2) x (M+(R2)) → A1, A2	-	-	8.5	3	
					11	0	(A2) x ((M)) → A1, A2			8.5	4	
					11	0	(A2) x ((M+(R2)))	2)		8.5	4	
Multiply registers / registers	-	MUR	1	1000	00	0	(A2) x (R2) → A1, A2		-	9.5	1	
					01	0	(A2) x ((R2)) → A1, A2		-	8.5	2	

name (in alphabetical order)	mnemonic P8505	mnemonic P855/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	time in cycles 6) P850 m.c.	time in cycles 6) P855/60 l.c.	time in cycles 6) P855/60 m.c.	remarks
Multiply with Constant	-	MUH	1	1000	01	0	$(A2) \times KL \rightarrow A1, A2$	2)	-	8-5	2	
One's complement	C1	C1	1	1111	10	0	$(\overline{M}) \rightarrow R1$		7		3	R1 = 1111: system mo
					10	1	$(\overline{M}) \rightarrow M$		8		4	
					10	0	$\overline{(M+(R2))} \rightarrow R1$		7		3	
					10	1	$\overline{(M+(R2))} \rightarrow M+(R2)$		8		4	
					11	0	$\overline{((M))} \rightarrow R1$	1)	9		4	
					11	1	$\overline{((M))} \rightarrow (M)$		10		5	
					11	0	$\overline{((M+(R2)))} \rightarrow R1$		9		4	
					11	1	$\overline{((M+(R2)))} \rightarrow (M+(R2))$		10		5	
One's Complement register/register	C1	C1R	1	1111	00	n.s.	$(\overline{R2}) \rightarrow R1$		4	1	1	when L/S bit = 0,
					01	0	$(\overline{(R2)}) \rightarrow R1$		5		2	R1 must be $\neq 0$;
					01	1	$(\overline{(R2)}) \rightarrow (R2)$		6		3	R1 = 1111: system mo
Output from register	OT	OTR	0	1000	-	-	word/character from R3 $\rightarrow$ device	5)	4		3	bit 8 = 0; system mo
Read channel address	-	RCA	0	1001	-	-	Channel number causing MIDB interrupt request $\rightarrow$ R3		-		3	bits 8-15: 0100000 system mode
Read interrupt lines	RL	RIL	0	1001	-	-	State interrupt lines $\rightarrow$ R3		4	2	1	bits 8-15: 00000000 system mode
Relative backwards conditional branch	RB	RB	0	1010	-	0	$(P) - displ. \rightarrow P$ (branch effective)	3)	4	1	1	bits 5-7: condition
							$(P) \rightarrow P$ (no branch)		3	1	1	bits 8-14: displ.; bits 15: n.s.

name (in alphabetical order)	mnemonic P850	mnemonic P855/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	time in cycles 6) P850 m.c.	time in cycles 6) P855/60 l.c.	time in cycles 6) P855/60 m.c.	remarks
Relative forward Conditional branch	RF	RF	0	1010	—	0	(P) + displ. ⇒ P (branch effective)		4	1	1	bits 5-7: condition
						—	(P) → P (no branch)	3)	3	1	1	bits 8-14: displ; bit 15: n.s.
Reset internal interrupt	RI 7)	RIT	0	0100	—	1	to clear internal interrupt bits		4	1	1	bits 8,9,15: 1, bits 10-14; D.A; system mod
Return from Function	RT	RTN	1	1110	01	0	(R2) + 4 → R2; ((R2)) → P; ((R2)-2) → PSW	*	9	1	3	*reloaded from stack; R2 = 1111: System mod
Send Status	SS 7)	SST	0	1001	-	-	Status character/word from device → R3	5)	4		3	bits 8-9: 11; system m
Set mode	-	SMD	0	0101	-	-	system mode ⇒ user mode	3)	-	1	1	bits 8-15: 0000000
Single left and normalize shift	-	SLN	0	0111	-	—			-	(1.5+n/2)	1	bits 8-10: 100; bits 11-14: R2; bit 15: n.
Single left arithmetic shift	-	SLA	0	0111	-	—		2)	-	(1.5+n/2)	1	bits 8-10: 000
Single left arithmetic shift (1 position)	LA	SLAI	0	0111	-	-			4	1	1	bits 8-10: 000; bits 11-15: 00001
Single left circular shift	—	SLC	0	0111	—	—		1)	-		*	bits 8-10: 110 * (2+½n+½)
Single left logical shift	—	SLL	0	0111	—	—			-	(1.5+n/2)	1	bits 8-10: 010

name (alphabetical order)	mnemonic P850	mnemonic 55/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	time in cycles 6) P850 m.c.	time in cycles 6) P855/60 l.c.	time in cycles 6) P855/60 m.c.	remarks
Single left logical shift (1 position)	LL	SLL1	0	0111	–	–		1)	4	1	1	bits 8-10: 010; bits 11-15: 00001
Single right and normalize shift	–	SRN	0	0111	–	–		3)	–	$(1.5+\frac{n}{2})$	1	bits 8-10: 101 bits 11-14: R2; bit 15: n.s
Single right arithmetic shift	–	SRA	0	0111	–	–			–	$(1.5+\frac{n}{2})$	1	bits 8-10: 001
Single right arithmetic shift (1 position)	RA	SRA1	0	0111	–	–		1)	4	1	1	bits 8-10: 001; bits 11-15: 00001
Single right circular shift	–	SRC	0	0111	–	–			–	$(1.5+\frac{n}{2})$	1	bits 8-10: 111
Single right circular shift (1 position)	RC	SRC1	0	0111	–	–			4	1	1	bits 8-10: 111; bits 11-15: 00001
Single right logical shift	–	SRL	0	0111	–	–			–	$(1.5+\frac{n}{2})$	1	bits 8-10: 011
one character	TC	SC	1	1100	10	1	$(R1)_{r} \rightarrow (M)_{r/l}$		8		4	R1 = 1111 : system mode
					10	1	$(R1)_{r} \rightarrow (M+(R2))_{r/l}$	3)	8		4	R1 must be ≠ 0
					11	1	$(R1)_{r} \rightarrow ((M))_{r/l}$		10		5	
					11	1	$(R1)_{r} \rightarrow ((M+(R2)))_{r/l}$		10		5	

name (alphabetical order)	mnemonic P850	mnemonic P855/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	time in cycles 6) P850 m.c.	time in cycles 6) P855/60 l.c.	time in cycles 6) P855/60 m.c.	remarks
Store character/register	TC	SCR	1	1100	01	1	$(R1)_r \rightarrow (R2)_{rL}$		6		3	R1=1111: system mod
Store register	TW	ST	1	0000	10	1	$(R1) \rightarrow M$		8	1	3	R1 = 1111: System mo
					10	1	$(R1) \rightarrow M + (R2)$		8	1	3	
					11	1	$(R1) \rightarrow ((M))$	3)	10	1	4	
					11	1	$(R1) \rightarrow ((M+(R2)))$		10	1	4	
Store register/register	TW	STR	1	0000	01	1	$(R1) \rightarrow (R2)$		6	1	2	R1 = 1111: system mod
					01	1	$(R1) \rightarrow (A15); (A15)-2 \rightarrow A15$		6	1	2	If pointer 128_{10}: stack over
Subtract constant	SK	SUK	0	0011	–	–	$(R3) - K \rightarrow R3$		4	1	1	short; R1=1111: system mo
	SU	SUKL	1		01	0	$(R1) - KL \rightarrow R1$		5		2	long; " "
Subtract register/register	SU	SUR	1	0011	00	n.s.	$(R1) - (R2) \rightarrow R1$		4	1	1	when L/S bit = 1, R1 mu
					01	0	$(R1) - ((R2)) \rightarrow R1$		5		2	be ≠ 0; R1 = 1111:
					01	1	$(R1) - ((R2)) \rightarrow (R2)$		6		3	System mode
Subtract word	SU	SU	1	0011	10	0	$(R1) - (M) \rightarrow R1$	2)	7		3	R1 = 1111: system mode
					10	1	$(R1) - (M) \rightarrow M$		8		4	when L/S bit = 1,
					10	0	$(R1) - (M+(R2)) \rightarrow R1$		7		3	R3 must be ≠ 0
					10	1	$(R1) - (M+(R2)) \rightarrow M+(R2)$		8		4	
					11	0	$(R1) - ((M)) \rightarrow R1$		9		4	
					11	1	$(R1) - ((M)) \rightarrow (M)$		10		5	
					11	0	$(R1) - ((M+(R2))) \rightarrow R1$		9		4	
					11	1	$(R1 - ((M+(R2))) \rightarrow (M+(R2))$		10		5	

name (in alphabetical order)	mnemonic P8050	mnemonic P8055/60	format	OP-code	mode	L/S (0/1) bit	function	condition register	time in cycles 6) P850 m.c.	time in cycles 6) P855/60 l.c.	time in cycles 6) P855/60 m.c.	remarks
Test mask	I	TM	1	0100	00	1	[(R1) ∧ (R2) ÷ 0 → CR	1)	4	1	1	R1=1111: system mod
Test not mask	U	TNM	1	0110	00	1	[(R1) ∧ (R2) ÷ 0 → CR	1)	4	1	1	" " "
Test status	TS 7)	TST	0	1001	–	–	test DCU "ready state"	5)	4		3	bits 8-9: 10; system m
Two's complement	C2	C2	1	0011	10	1	0 − (M) → M		8		4	bits 5-8: 0000
					10	1	0 − (M + (R2)) → M + (R2)		8		4	
					11	1	0 − ((M)) → (M)	2)	10		5	
					11	1	0 − ((M + (R2))) → (M + (R2))		10		5	
Two's complement/ register	CR	C2R	1	0011	01	1	0 − ((R2)) → (R2)		6		3	bits 5-8: 0000
Write interrupt mask	WM 7)	WIM	0	1000	–	–	(R3) → mask register	3)	4	2	1	bits 8-15: 000000 system mode
Write mask protection	–	WMP	0	1000	–	–	(R3) → M.P. key register		–	2	1	bits 8-15: 01000000 system mode
Write mask protection nº 2		WM2	0	1000	–	–	(R3) → M.P. key register		–	2	1	bits 8-15: 11000000 system mode

Condition register

1)

CR=0 if result = 0
1 if result > 0
2 if result < 0

2)

CR=0 if result = 0
1 if result > 0
2 if result < 0
3 if overflow

3)

CR unchanged

4)

CR=0 if a = b
1 if a > b
2 if a < b

5)

CR=0 if command accepted
1 if command not accepted
3 if device address unknown

6)

Execution times in microseconds

	P850	P855	P860
mc (memory cycle)	1.6	1.2	0.84
lc (logic cycle)	-	0.72	0.72

7)

Mnemonics are not recognized by the Assembler.

Memory protection

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

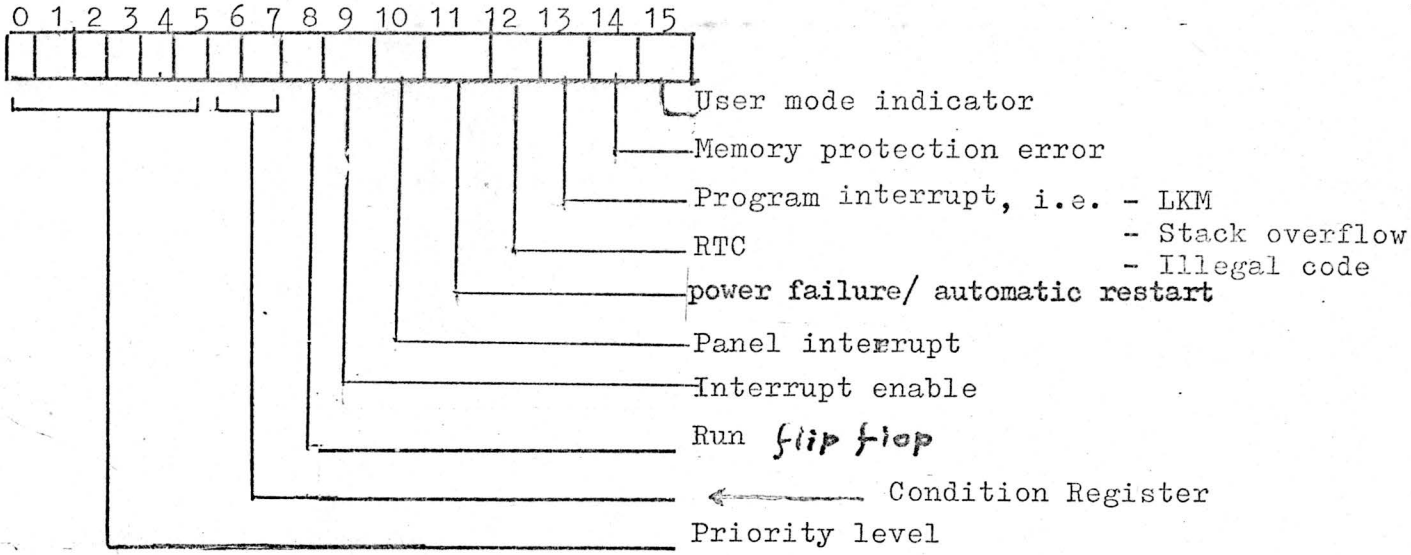
MK1

bit	memory area
15	0000 - 07FE
14	0800 - 0FFE
13	1000 - 17FE
12	1800 - 1FFE
11	2000 - 27FE
10	2800 - 2FFE
9	3000 - 37FE
8	3800 - 3FFE
7	4000 - 47FE
6	4800 - 4FFE
5	5000 - 57FE
4	5800 - 5FFE
3	6000 - 67FE
2	6800 - 6FFE
1	7000 - 77FE
0	7800 - 7FFE

MK2

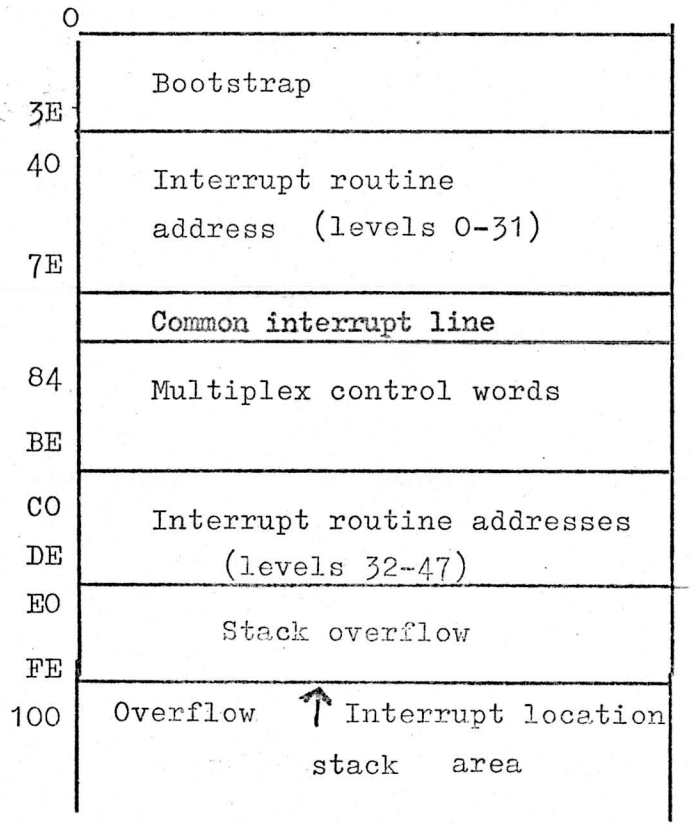
bit	memory area
15	8000 - 87FE
14	8800 - 8FFE
13	9000 - 97FE
12	9800 - 9FFE
11	A000 - A7FE
10	A800 - AFFE
9	B000 - B7FE
8	B800 - BFFE
7	C000 - C7FE
6	C800 - CFFE
5	D000 - D7FE
4	D800 - DFFE
3	E000 - E7FE
2	E800 - EFFE
1	F000 - F7FE
0	F800 - FFFE

Program Status Word P850-55-60



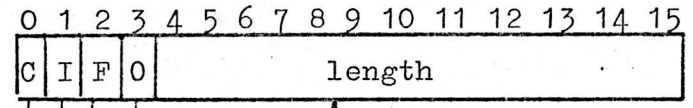
Dedicated Memory Locations

Hexa address



MUX Control Words

1st word



Length = given in negative value (2's complement

0: not used

F: Function bit

I: Input/Output bit

1 = input

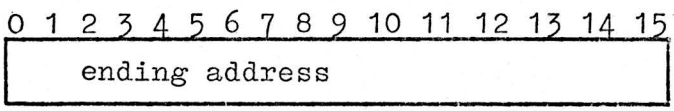
0 = output

C: Character bit

1 = character exchange

0 = word exchange

2nd word



If word transfer, bit 15 must be zero.

Memory Clear Program

This program puts the contents of A2 in the memory location X'40' upto the address given in A3.

P850

```
0   824D  TW   A2,DA,A3,S
2   0000
4   1B02  SK   A3,2
6   5C08  RB   4,8
8   207F  HALT
```

P855/P860

```
0   L1    DATA  0
2         ST    A2,L1,A3
6         SUK   A3,2
8         RB(4) L1+2
A         HLT
```

This program puts the contents of A1 in the memory locations /4C upto the address given in A3.

P850

```
40  024C  LK A2,/4C
42  8129  TW A1,RA,A2,S
44  1202  AK A2,2
46  EA0C  CW A2,RR,A3
48  5A08  RB 2,8
4A  207F  DATA /207F
4C
```

P855/P860

```
        LDK   A2,/4C
LAB     STR   A1,A2
        ADK   A2,2
        CWR   A2,A3
        RB(2) LAB
        HLT
```

Bootstrap P860 (8x8)

ca- on	Hexadecimal code	Source statement			Comments
			IDENT	BOOT	
		*		BOOTSTRAP FOR 8+8 FORMAT FROM PTR(/20)	
		*		BOOTSTRAP FOR P850/855/860	
		*		BEFORE STARTING SET REGISTERS : A3 = 1 (ASR ONLY)	
		*		A1 = Ø	
		*		AØ = Ø	
		*		USING IPL LOCATION /3A MUST BE 8F02 (=ABR A8)	
		PTR .	EQU	/20	
		S	EQU	1	
00	20BF		INH		
02	43E0		CIO	A3,S,PTR	
04	4A20		INR	A2,0,PTR	
	5C04		RB(4)	*-2	
08	227F		ANK	A2,/7F	
0A	5808		RB(0)	*-6	LOOP TILL RUB OUT
0C	0100		LDK	A1,0	
0E	0402		LDK	A4,2	A4 : FLAG FOR LENGTH, BASE OR CODE
10	0602		LDK	A6,2	
12	E508		ECR	A5,A2	
14	4A20	INR	INR	A2,0,PTR	
16	5C04		RB(4)	*-2	
18	1E01		SUK	A6,1	
1A	590A		RB(1)	INR-2	
1C	9508		ADR	A5,A2	
1E	1C01		SUK	A4,1	
20	5006		RF(0)	*+8	ADDR
22	520C		RF(2)	*+14	CODE
24	8714		LDR	A7,A5	A7 : LENGTH
26	5F18		RB	INR-4	
28	8314	ADDR	LDR	A3,A5	A3 : BASEADDR
2A	1502		ADK	A5,2	
2	8094		LDR	A8,A5	A8 : STARTADDRESS IPL
2E	5F20		RB	INR-4	
30	B114	CODE	XRR	A1,A5	
32	1302		ADK	A3,2	
34	8520		STR	A5,A3	
36	1F01		SUK	A7,1	
38	5C2A		RB(4)	INR-4	
3A	207F		HLT		
			END		

Ø = cijfer nul
O = hoofdletter o

Bootstraps P850 ASR, PTR (4x4)

Location	Hexadecimal code	Source Statements		Comments
0000	20BF	[IH]		INHIBIT INTERRUPTS
	42C1*	[CT]		START DEVICE
	4A01*	[IN]		READ TAPEFEED OR FIRST CHARACTER
	5C04	RB	4,4	IF NOT ACCEPTED TRY AGAIN
0008	227F	IK	A2,/7F	TAKE ONLY SEVEN BITS
	5808	RB	0,8	IF ZERO, SKIP TAPEFEED
	8320	TW	A3,RA,P	SET INDICATOR .NEXT
	FFFC	DATA	-4	WORD IS LENGTH
0010	0604	LK	A6,4	NEW WORD: CHARACTER COUNT
	4A01*	[IN]		READ NEXT CHARACTER
	5C04	RB	4,4	IF NOT ACCEPTED, TRY AGAIN
	220F	IK	A2,15	TAKE ONLY LAST 4 BITS.
0018	3D41	LL	A5	MAKE ROOM IN A5
	3D41	LL	A5	
	3D41	LL	A5	
	3D41	LL	A5	
0020	9508	AD	A5,RR,A2	INSERT NEW TETRAD
	1E01	SK	A6,1	DECREASE WORD COUNT
	5C14	RB	4,/14	IF NOT LAST WORD, READ NEXT
	1302	AK	A3,2	ELSE INCREASE INDICATOR
0028	5006	RF	0,6	IF INDICATOR ZERO, WORD IS START ADDRESS
	5108	RF	1,8	IF POSITIVE, STORE CODE
	8714	TW	A7,RR,A5	LENGTH IN A7
	5F20	RB	7,/20	READ NEXT WORD
0030	8314	TW	A3,RR,A5	REPLACE INDICATOR BY CURRENT WORD-ADDRESS
	5F24	RB	7,/24	READ NEXT WORD
	B114	X	A1,RR,A5	SUMCHECK
	852D	TW	A5,RR,A3,S	STORE NEXT CODE WORD
0038	1F01	SK	A7,1	DECREASE LENGTH
	5C2C	RB	4,/2C	IF NOT ZERO, GO ON.
	4281*	CT		CT STOP
	207F	HT		HALT
		END		END

In this example the input device address is 1, and the absolute code is input via register A2.

Before the bootstrap is loaded the following values must be placed in the first three registers:

P-register	0
A1	0
A2	1

Key

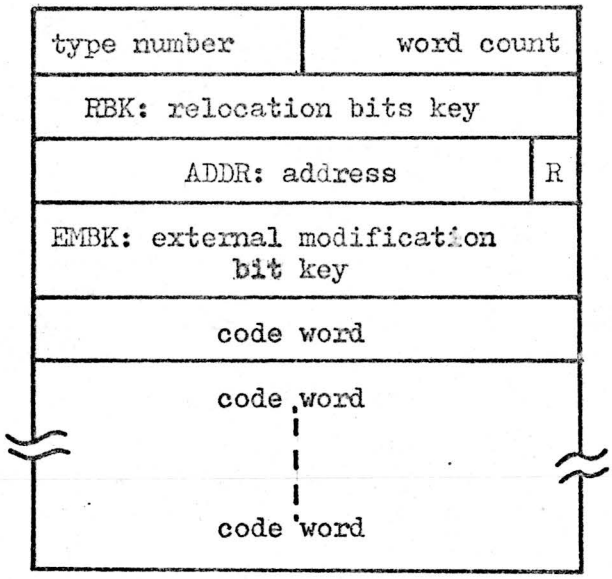
[ ] Pseudo-mnemonic.

*) code word includes a device address, which may have to be altered for different installations.

Program identification

IDENT - program name - line feed - X off - carriage return.

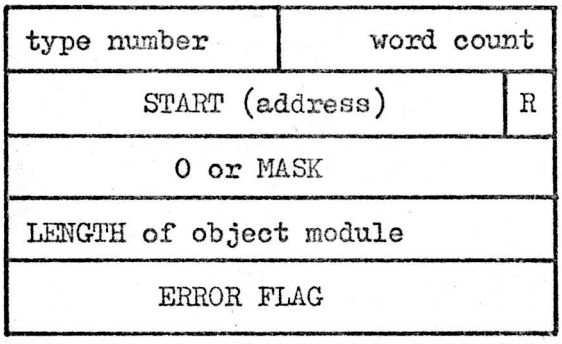
Code cluster



= 0: address is absolute
= 1: address is relative

P855/P860 only

END/START cluster

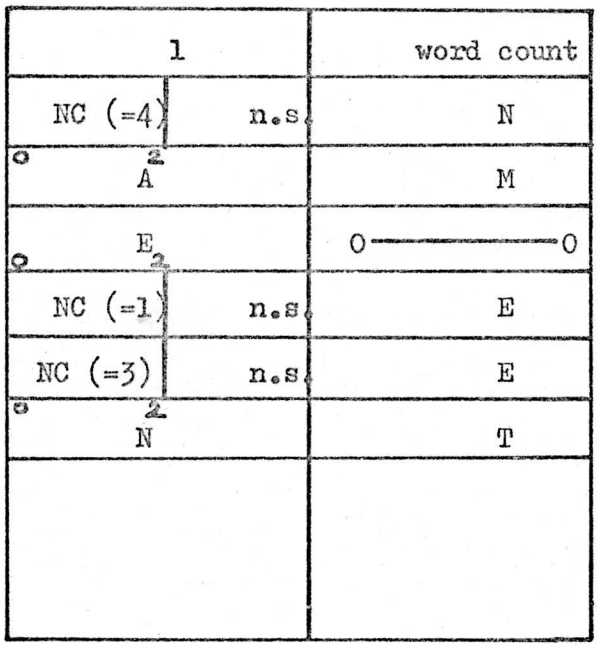


= 0: start address absolute
= 1: start address relative

even number of characters

indicates number of errors in binary

Entry point names cluster



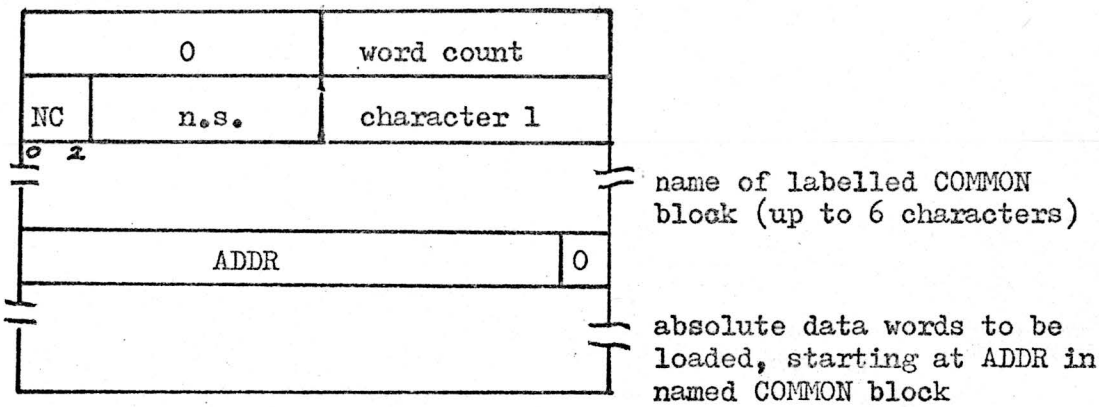
NC = number of characters of entry point name

n.s. = not significant

External Reference names cluster

Same format as "Entry point names cluster".

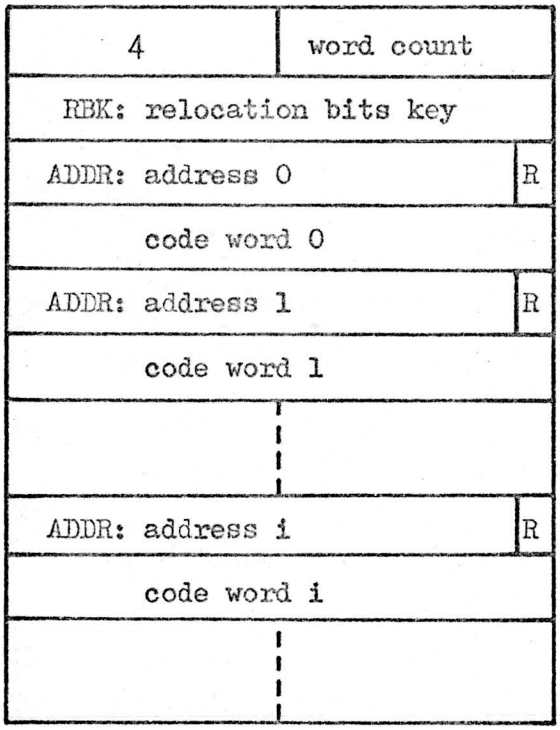
COMMON data cluster



NC = number of characters of the symbol (3 bits)

n.s. = not significant.

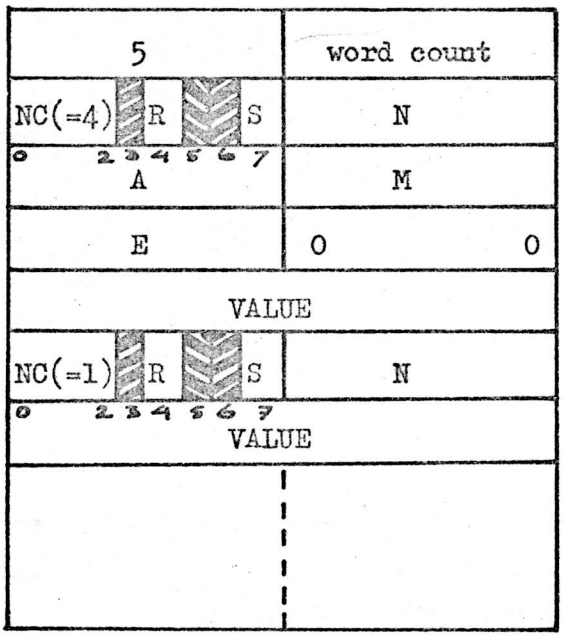
Internal modification cluster



= 0: address absolute
= 1: address relative

maximum number
of code words: 16

Entry point definition cluster

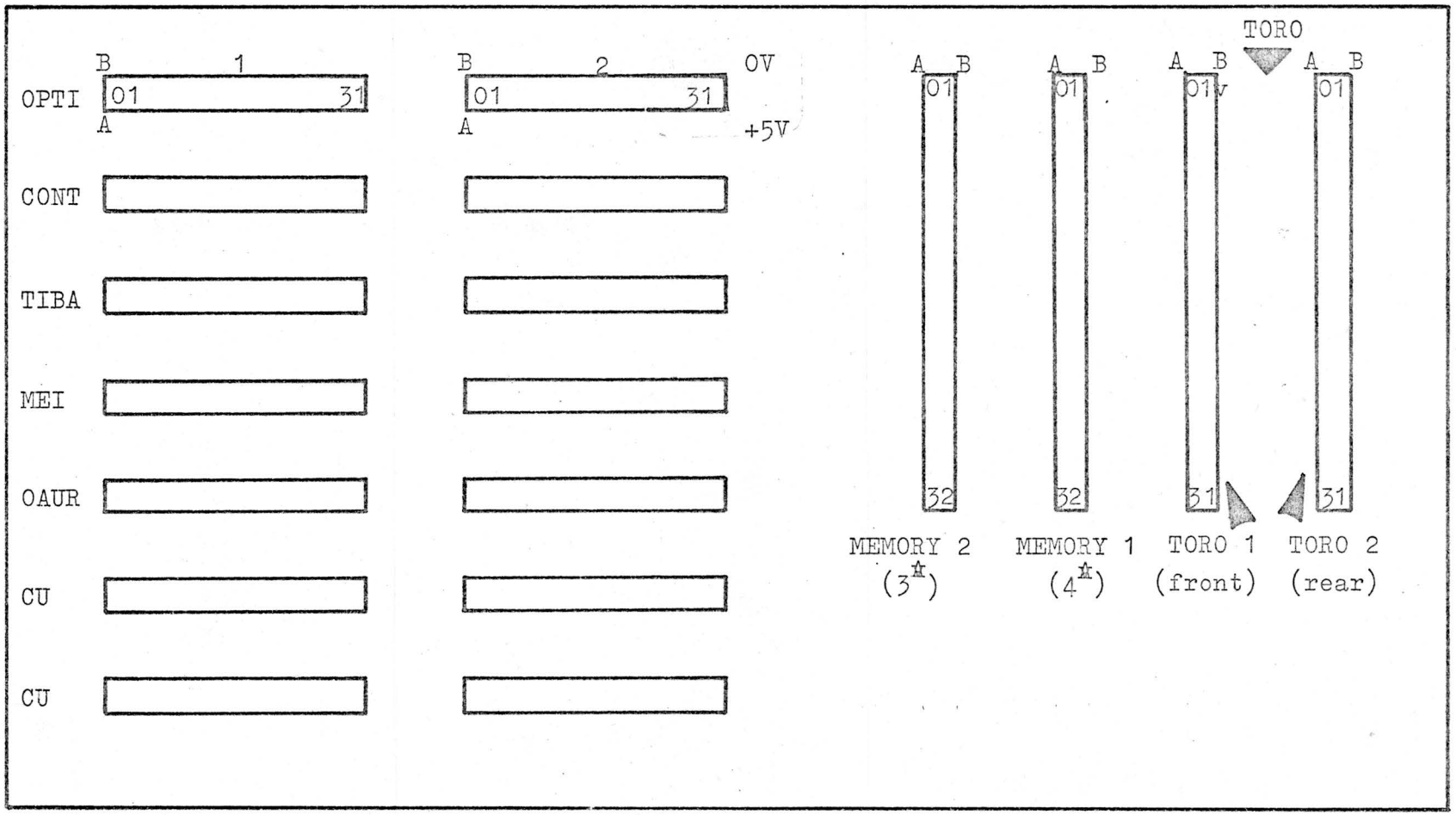


NC = number of characters of entry point name
R = 0 : absolute entry point
  = 1 : relative entry point
S = 1 if entry point name = internal symbol table name.

Common length definition cluster

Same format as "Entry point definition cluster".

Backpanel P850 basic cabinet



M E I 1 A 2 5

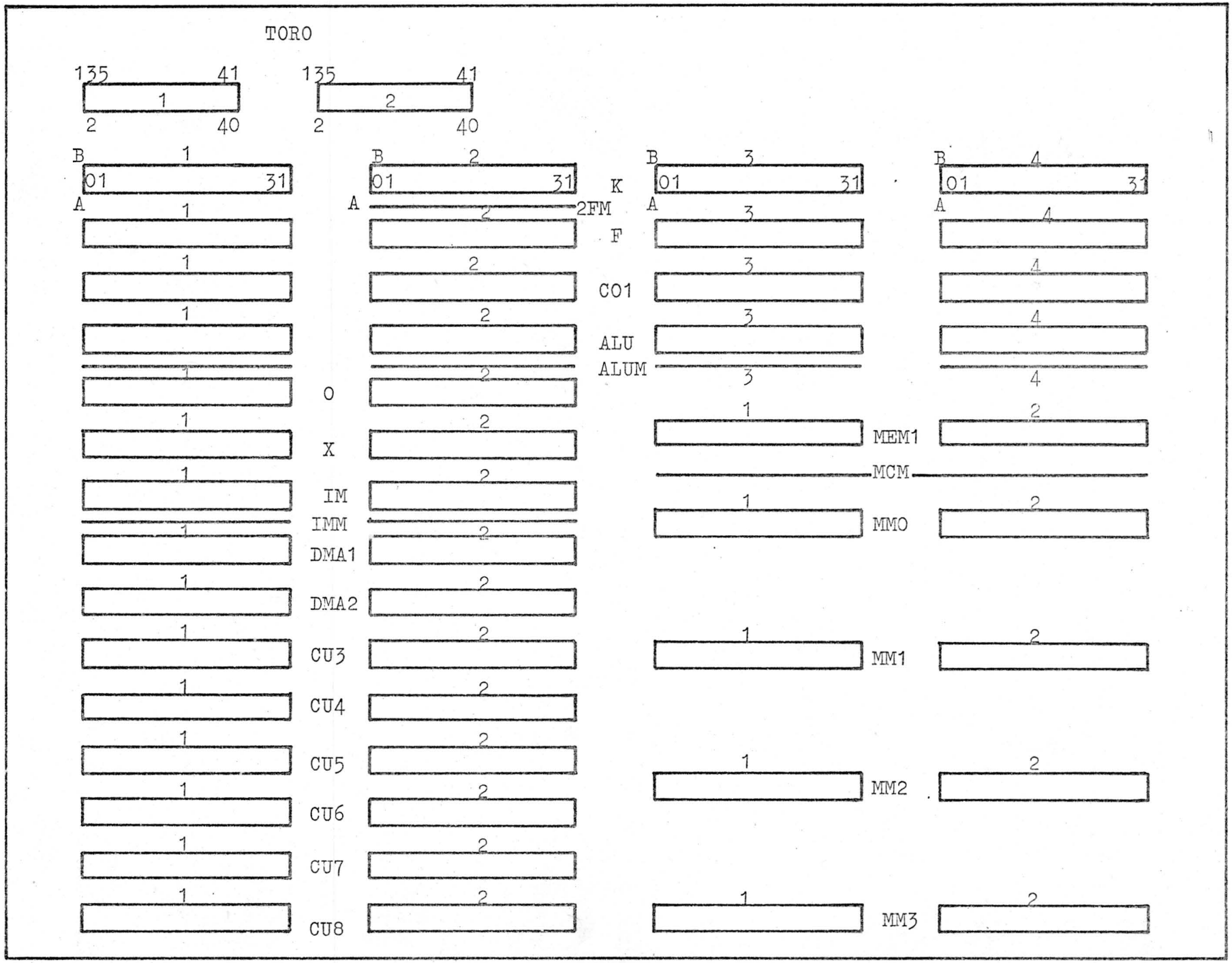
- pin number (1 - 31)
- row indicator: A = up, B = down
- 1 = left, 2 = right
- card type

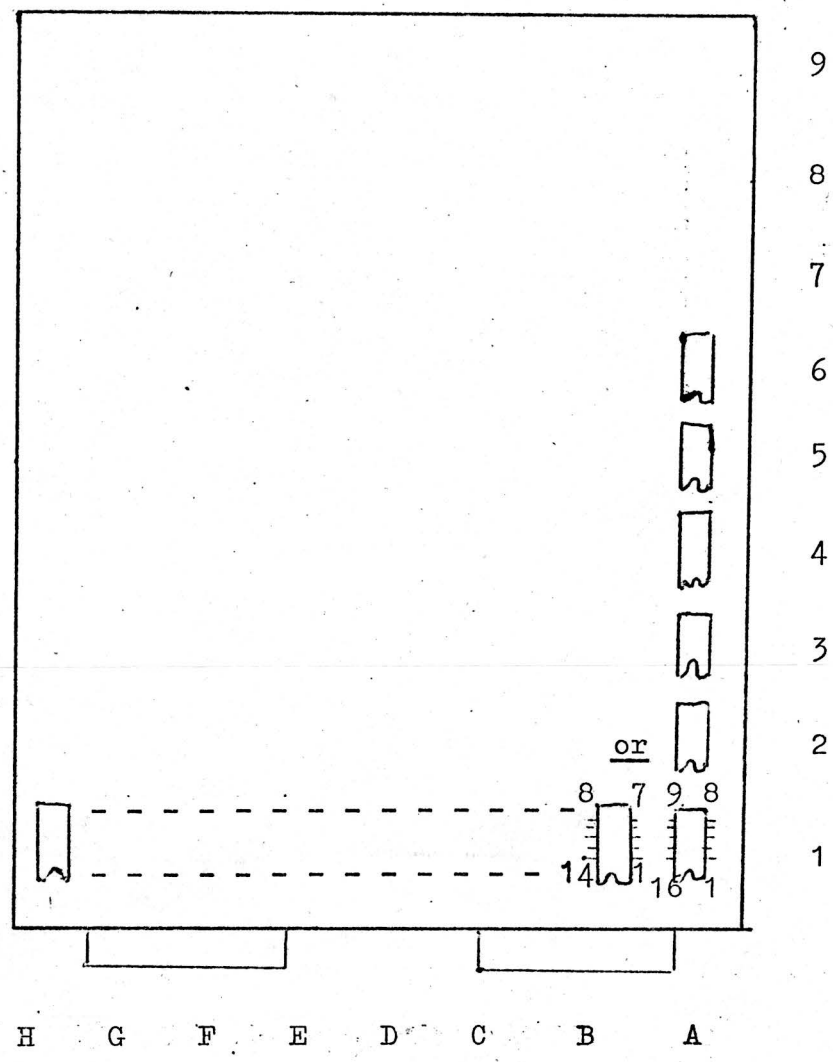
M E M 2 1 A 1 0

- pin number (1 - 31)
- connector 1A or 1B
- memory number

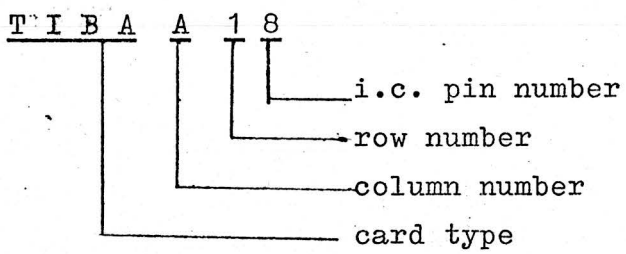
* In Extended cabinet

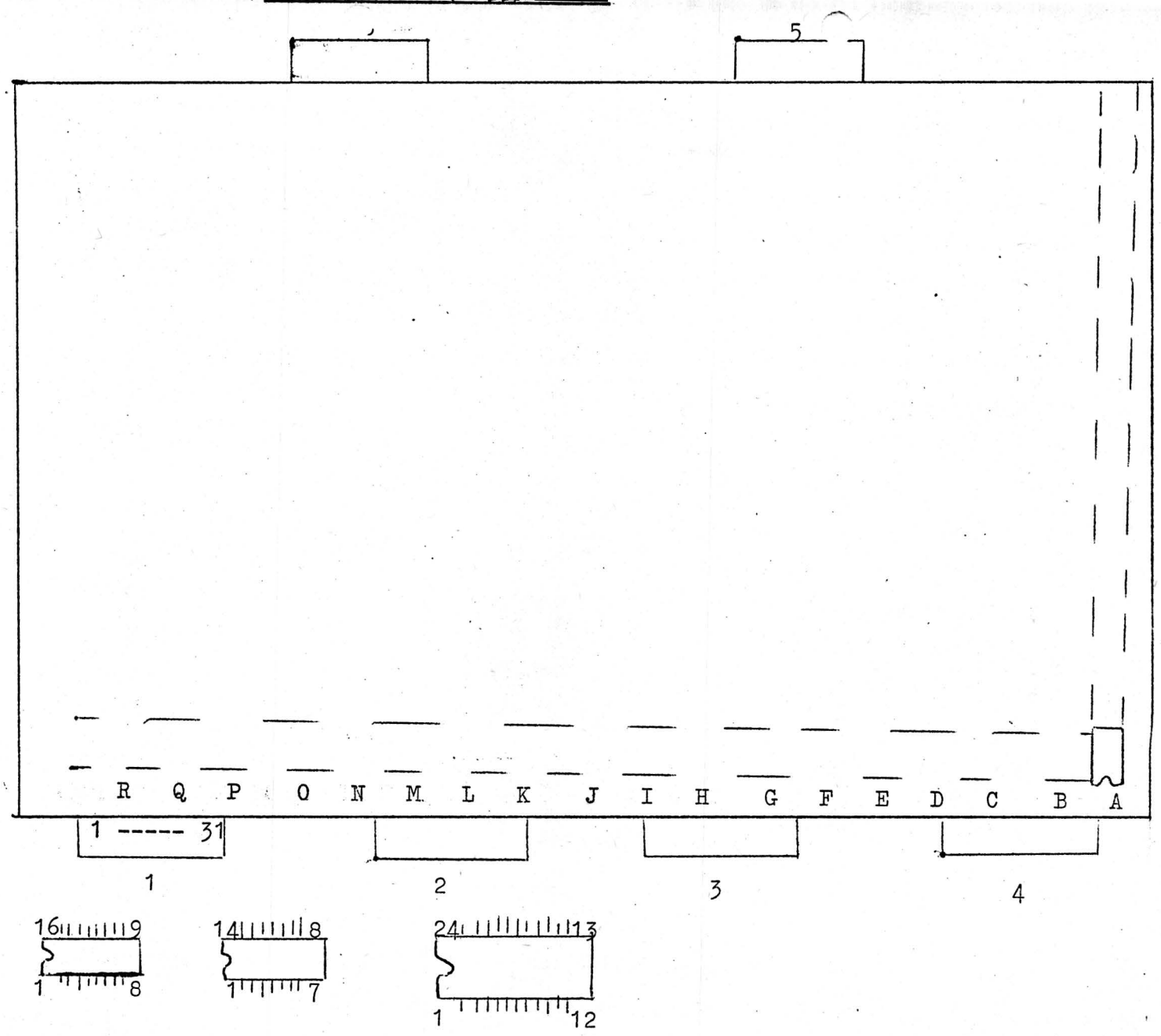
Backpanel P860 basic cabinet





indication of location:




5
9
8
7
6
5
4
3
2
1
R Q P O N M L K J I H G F E D C B A
1 ----- 31
1
2
3
4
16 9
1 8
14 8
1 7
24 13
1 12

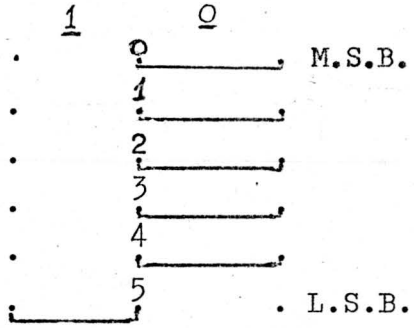
Control Units

. Status flip-flops

	F0	F1
INCT	0	0
EX	0	1
EXCH	1	1
WST	1	0

Changing the device address

The device address on a C.U. card may be altered by means of jumpers.



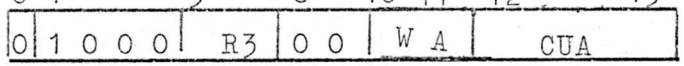
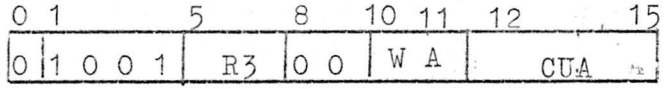
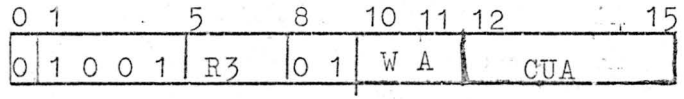
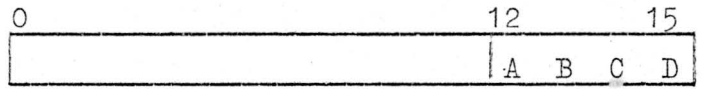
Example for device address 0001.

C.U. Status Word Configuration

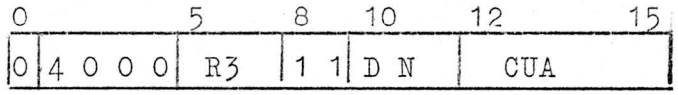
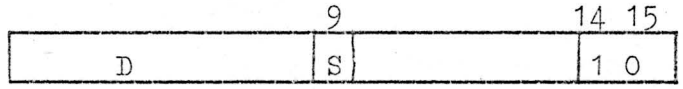
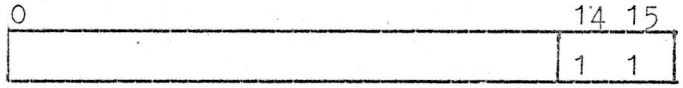
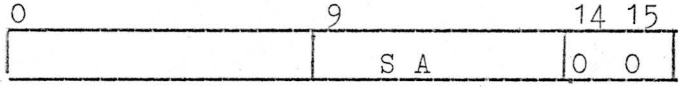
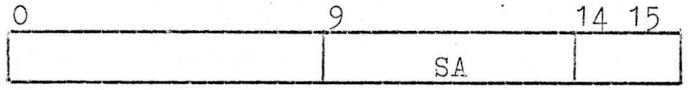
Bit	Description	CU								
		ASR	CR	DU	LP	TP	PTR	CASS Tape	PLOT	
0	—									
1	has been unready			x				x		
2	—									
3	tape mark has been read							x		
4	—									
5	on cylinder			x						
6	seek error			x						
	write unable							x		
7	A or B side							x		
8	Dev. Addr.							x		
9	"			x				x		
10	EOT						x	x		
	tape low					x				
11	Progr. error			x				x	x	
12	incorr. length		x	x				x		
	Y limit overpass								x	
13	Parity error							x		
	data fault		x	x						
14	throughput error	x	x	x			x	x		
15	not operable (only significant bit for TST)	x	x	x	x	x	x	x	x	

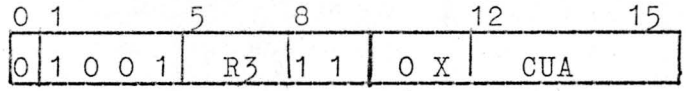
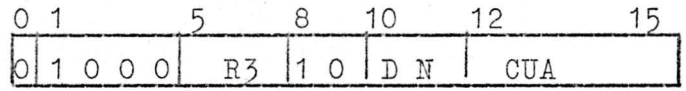
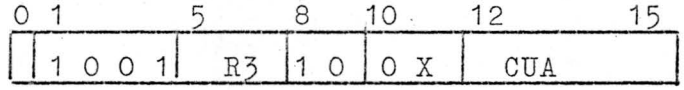
ASR I/O typewriter
CR card reader
DU disc unit
LP line printer
TP tape punch
PTR punched tape reader
CASS cassette tape
PLOT plotter

Device					recognised command	Format	Meaning
PTR	CR	PLOT PTP	ASR	LP		0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	
x			x	x	TST	0 | 1 0 0 1 | R3 | 1 0 | D A	Test status DA = Device Address R3 = reg into which status is loaded see status CU for contents R3 reg.
x	x	x	x	x	CIO start (read a card for CR)	0 | 1 0 0 0 | R3 | 1 1 | D A 15 Reg. indicated in R3	R3 only significant for ASR Start input bit 15 = 1 start input bit 15 = 0 start output
x	x	x	x	x	CIO stop	0 | 1 0 0 0 | R3 | 1 0 | D A	Stop input R3 not significant
x	x		x		INR	0 | 1 0 0 1 | R3 | 0 X | D A	Input to register indicated in R3 field. X: no significance
x	x	x	x	x	SST	0 | 1 0 0 1 | R3 | 1 1 | D A	Send status The status word of the controlunit i sent to the register indicated in R3 field
		X	X	X	OTR	0 | 1 0 0 0 | R3 | 0 X | D A	Output from the register indicated in R3 field to CU
				x		format control character 8 9 10 11 12 13 14 15 1 | 1 | X | 0 | Chan.Nr	Advance till given channel
				x		8 9 10 11 12 13 14 15 1 | 1 | X | 1 | Nr.of lines	Skip the given number of lines

Recognised Command	Format	Meaning
OTR		Output data word R3 indicator register from data exchange CUA = CU address WA = Word address 00 = word A 01 = " B 10 = " C 11 = " D
INR		Input data word R3 indicates register into which data is exchanged. CUA = CU address WA = Word address (see above)
INR	 Format of R3 	Input word address R3 indicates the register ... into which the address is loaded CUA = CU address WA = not significant A one-bit in field 12-15 indicates that the corresponding word is active.

Moving head disc CU

recognised commands	Format 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	Meaning
CIO start general		Start an Operation DN = Device number 00 or 01 CUA = Control Unit address R3 = specifies the command
	R3 Formats	
seek command		move the head to another cylinder D = difference between cylinder numbe S = direction of motion 1 = forward 0 = reverse
Seek to zero command		move heads to cylinder 00
Write a sector		Start a write command on the sector address on the BOU lines on the cylinder of the moment. SA = Sector Address 0-31
Read a sector		Start a read operation on the sector which is addressed on the BOU lines on the cylinder of the moment SA = Sector address 0-31

Recognised commands	Format 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	Meaning
SST		Exchange Status between CPU and CU CUA = CU address X = not significant R3 = specifies register into which status is sent (see status word)
Stop command		Stop data transfer R3 = not significant DN = Device number CUA = CU address
TST		Test status of CU CUA = CU Address R3 indicates register into which status is exchanged.

gh

TST

Format

0	1			5		8		10		12			15
				R3		1	0	D	N	CUA			

Meaning

Test status
CUA = CU address
DN = Device number (00-11)
R3 = indicating register into which status is exchanged
see status

CIO Start

Format

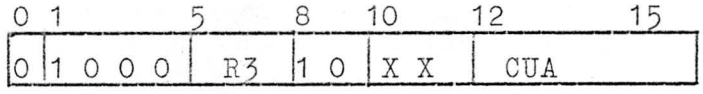
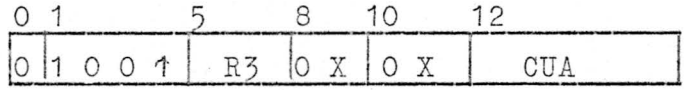
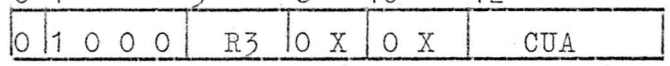
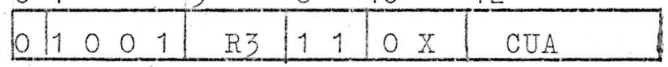
0	1				5		8		10		12			15
0	1	0	0	0	R3		1	1	D	N	CUA			

Meaning

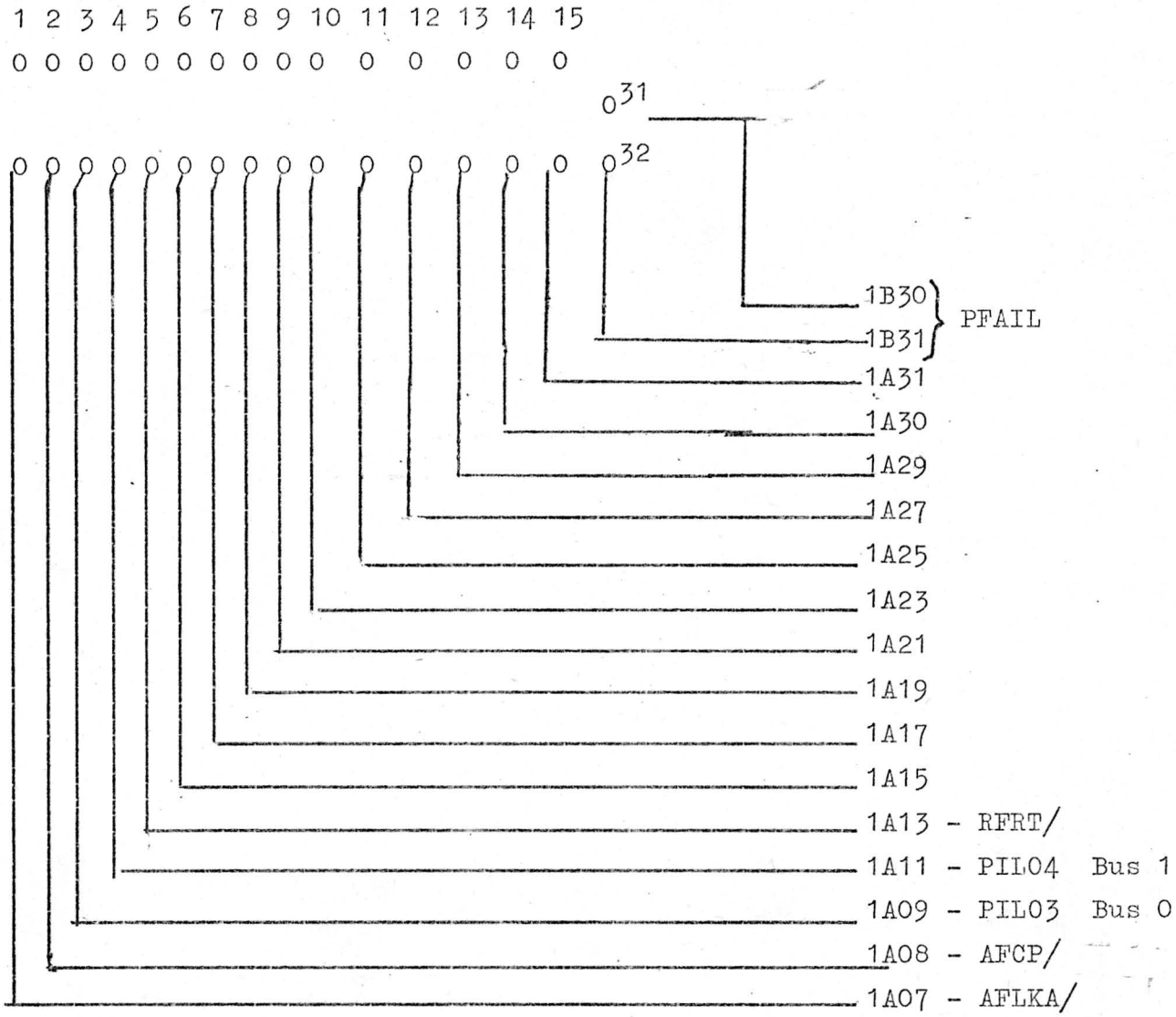
Start command
CUA = CU address
DN = Device number
R3 = indicates register specifying the command.

R3 contents

bit number 12	13	14	15	Command
0	0	0	0	Lock/unlock
0	0	0	1	Erase forward
0	0	1	0	Backwards space block
0	0	1	1	Forward " "
0	1	0	1	Write a block forward
0	1	1	1	Read " " "
1	0	0	0	Rewind at fast speed
1	0	1	0	Search tape mark backwards
1	0	1	1	" " " forward

Recognised commands	Format 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	Meaning
CIO Stop		Stop transfer CUA = CU address DN = not significant Contents of register indicated in R3 not significant. X =
INR		Input to register DN = not significant R3 indicates register into which data is exchanged. X not significant
OTR		Output from register DN = not significant X = not significant R3 indicates register from which the data is exchanged.
SST		Send Status see status meaning for explanation of field see above

Option Card (OPTI)



INTerrupt Mask wiring upto P850-26

Standard sequence interrupt mask bits

Bit

0 PFAURE

1 LKM

2 RTC

4 CP

5 Cass. Tape

6 PTR

7 PTP

8 ASR

9 DISC

10 LP

11 CR

12 CP

Interrupt mask bits	Pin location	Connected to
BMS KIL 00/	F 1A11	
BMS KIL 01/	F 1A14	
BMS KIL 02/	F 1A16	
BMS KIL 03/	F 1B16	
BMS KIL 04/	F 1B25	
BMS KIL 05/	F 1B17	
BMS KIL 06/	F 1B18	
BMS KIL 07/	F 1B22	
BMS KIL 08/	F 1A23	
BMS KIL 09/	F 1B28	
BMS KIL 10/	F 1A19	
BMS KIL 11/	F 1A20	
BMS KIL 12/	F 2B10	
BMS KIL 13/	F 1A28	
BMS KIL 14/	F 1A31	
BMS KIL 15/	F 2B09	

Standard interrupt signals	Pin location
CFPF/	IM 1A16
CFEM/	IM 1A17
CFRT/	IM 2A27
CFPI/	F 3B31
CFCP/	F 4B04
CFMIDB/	X 1A06
PIL	CU 1A15
CISMSK/	F 2A02

Standard interrupt levels	Pin location	Dedicated memory address	Connected to
BIS 00/	F 2B12	/0040	PFAURE
BIS 01/	F 2B11	/0042	LKM
BIS 02/	F 2A12	/0044	RTC
BIS 03/	F 2A11	/0046	memory protection
BIS 04/	F 2A10	/0048	control panel
BIS 05/	F 2A09	/004A	cassette tape
BIS 06/	F 2A08	/004C	PTR
BIS 07/	F 2A07	/004E	PTP
Optional interrupt levels			
JIS 08/	IM 1A02	/0050	operator's typewriter
JIS 09/	IM 1A03	/0052	disc
JIS 10/	IM 1A04	/0054	line printer
JIS 11/	IM 1A05	/0056	card reader
JIS 12/	IM 1A06	/0058	magtape
JIS 13/	IM 1A07	/005A	
JIS 14/	IM 1A08	/005C	
JIS 15/	IM 1A09	/005E	
JIS 16/	IM 1A10	/0060	
JIS 17/	IM 1A11	/0062	

Optional interrupt levels	Pin location	Dedicated memory address	Connected to
KIS 18/	IM 2A05	/0064	
KIS 19/	IM 2A04	/0066	
KIS 20/	IM 2A03	/0068	
KIS 21/	IM 2A02	/006A	
KIS 22/	IM 2A01	/006C	
KIS 23/	IM 2B01	/006E	
KIS 24/	IM 2B02	/0070	
KIS 25/	IM 2B03	/0072	
KIS 26/	IM 2B04	/0074	
KIS 27/	IM 2B05	/0076	
KIS 28C/	IM 5A23	/0078	
KIS 29C/	IM 5A22	/007A	
KIS 30C/	IM 5A21	/007C	
KIS 31C/	IM 5A20	/007E	
LIS 32C/	IM 5A19	/00C0	
LIS 33C/	IM 5A18	/00C2	
LIS 34C/	IM 5A17	/00C4	
LIS 35C/	IM 5A16	/00C6	
LIS 36C/	IM 5A15	/00C8	
LIS 37C/	IM 5A14	/00CA	
LIS 38C/	IM 5A13	/00CC	
LIS 39C/	IM 5A12	/00CE	
LIS 40C/	IM 5A11	/00D0	
LIS 41C/	IM 5A10	/00D2	
LIS 42C/	IM 5A09	/00D4	
LIS 43C/	IM 5A08	/00D6	
LIS 44C/	IM 5A07	/00D8	
LIS 45C/	IM 5A06	/00DA	
LIS 46C/	IM 5Ao5	/00DC	
LIS 47C/	IM 5A04	/00DE	

P855/P860 Break levels

Level number	Pin location	Address on header connector E	Dedicated Multiplex address	Connected to
XBS 01	X 1A02	X 5A11	/0084 - /0086	
XBS 02	X 1B02	X 5A12	/0088 - /008A	
XBS 03	X 1B03	X 5A13	/008C - /008E	
XBS 04	X 1B04	X 5A15	/0090 - /0092	
WBS 05	X 1B05	X 5A17	/0094 - /0096	
WBS 06	X 1B06	X 5A14	/0098 - /009A	
WBS 07	X 1B07	X 5A16	/009C - /009E	
WBS 08	X 2B26	X 5A03	/00A0 - /00A2	
WBS 09	X 2B27	X 5A04	/00A4 - /00A6	
WBS 10	X 2B28	X 5A05	/00A8 - /00AA	
WBS 11	X 2B29	X 5A06	/00AC - /00AE	
WBS 12	X 2B30	X 5A07	/00B0 - /00B2	
WBS 13	X 1A05	X 5A08	/00B4 - /00B6	
WBS 14	X 1A04	X 5A09	/00B8 - /00BA	
WBS 15	X 1A03	X 5A10	/00BC - /00BE	

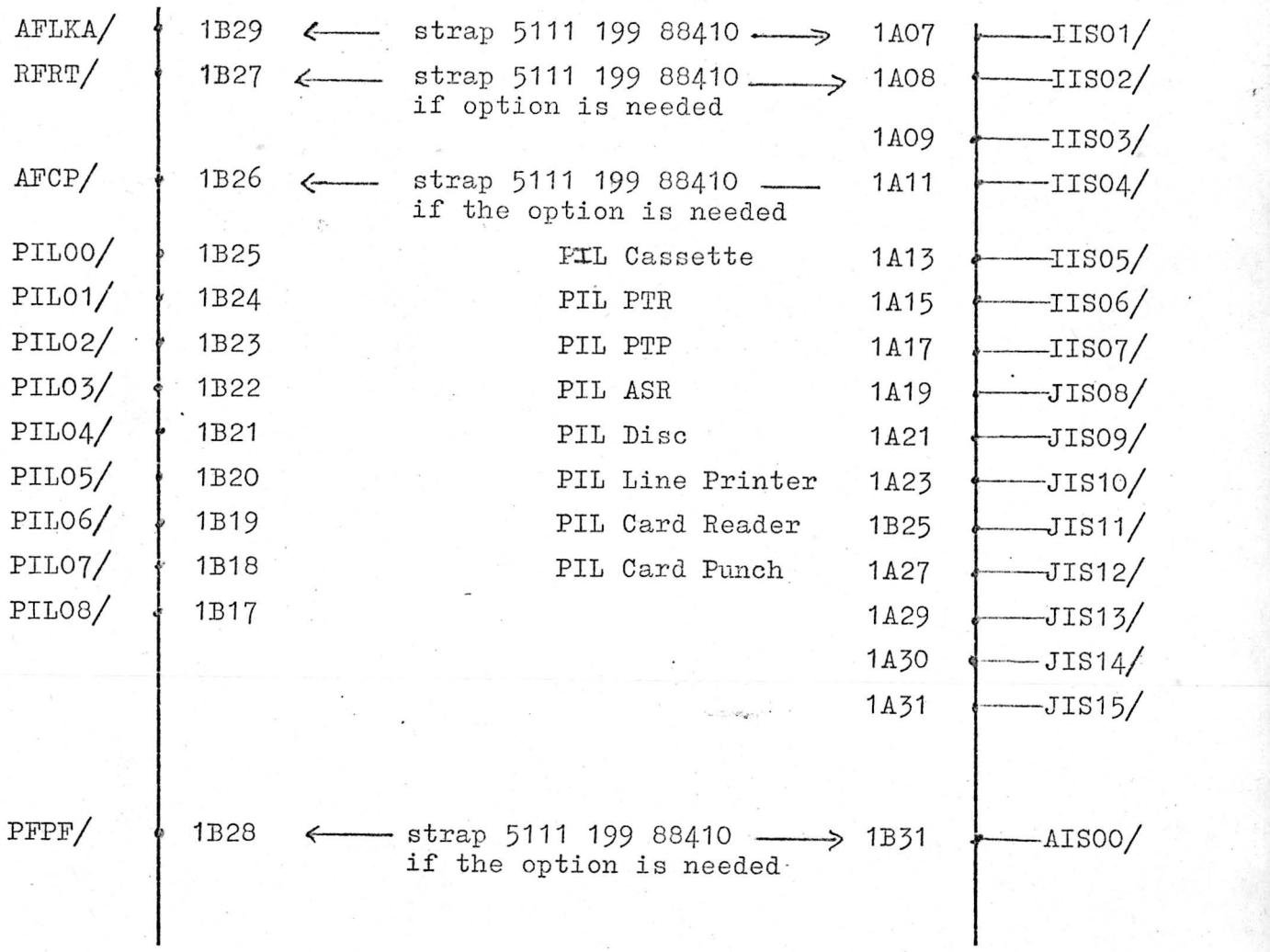
Note

Break level = device address.

Break is generated on c.u. 1A16.

Pin Board Cabling

Option Card S 307A (5111 199 93920)



data systems

September 1972

A publication of

NV Philips-Electrologica

P.O. Box 245, Apeldoorn, Netherlands

5122 991 16901