F-67 (166)

ARITHMETIC PROCESSOR 166
INSTRUCTION MANUAL

VOLUME 2

PDP-6

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

F-67(166)

PDP-6 ARITHMETIC PROCESSOR 166
INSTRUCTION MANUAL

VOLUME 2

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

COPY NO.

This manual contains proprietary information. It is provided to the customers of Digital Equipment Corporation to help them properly use and maintain DEC equipment. Revealing the contents to any person or organization for any other purpose is prohibited.

ENGINEERING DRAWINGS

ENGINEERING DRAWINGS (continued)

ENGINEERING DRAWINGS (continued)

System Block Diagram
(SD-D-166-0-BD)

System Block Diagram
(SD-D-166-0-BD)

LOGIC PREFIXES

A	5-3
ACBM	5-9
ACCP	5-9
AR	6-4,5,6,7,8,9,10
AS	5-10
BLT	6-18
BOOLE	5-10
CFAC	6-17
CH	6-19,20
CPA	8-5
CR	8-13,14,15
DC	6-20
DS	6-25,26
E	5-5
EX	5-13
F	5-4
FA	6-22
FD	6-22
FE	6-14,15
FM	6-22
FP	6-23
FS	6-19
FWT	5-9
HWT	5-9
I	5-3
IOB	8-1,2
IOT	8-1
IR	5-7,8
JP	5-10
K	5-2
KEY	5-1,2
LC	6-20
MA	7-1,2,3
MAI	7-2
MB	6-1,2,3,4;7-6
MC	7-8,9
MEMAC	5-10
MI	7-7
MP	6-21
MQ	6-7,11,12,13
MR	5-1,2
MS	6-24
NR	6-27
PC	5-11,12
PI	8-3,4
PR	7-4
PTP	8-9,10
PTR	8-6,7,8
RLA	7-5
RLR	7-5
RUN	5-1
S	5-6
SA	6-16
SC	6-14,15,16
SH	6-20
SHC	6-7
SW	5-1
TTI	8-11,12
TTO	8-11,12
TTY	8-11,12
UUO	5-10
XCT	5-10

IR0-2
000 IR UUO A ——— ∧EX UUO SYNC(1)
IR JRST A ∧ IR9∨10(1) ∧ EX USER B(1)
CPA IOT USER(0) ∧ IR IOT A ∧ EX PI SYNC(0) ∧ EX USER B(1)
} EX IR UUO

001 IR FP/CH
IR3-5
000
001
010
011 IR FP/CH3
IR6-8
000
001
010 IR FSC
011 IR CAO (IBP)
100 IR LDCI (ILDB)
101 IR LDC (LDB)
110 IR DPCI (IDPB)
111 IR DPC (DPB)

∧IR3(1) = IR FP
NR ROUND = IR6(1) ∧ MQ1(1) ∧ NRF3(0)
IR7,8
00 IR FP DIR
01 IR FP REM (L)
10 IR FP MEM (M)
11 IR FP BOTH (B)

IR4,5
00 IR FAD
01 IR FSB
10 IR FMP
11 IR FDV

∧IR5,6(01) = FWT SWAP
IR7,8
00 FWT 00
01 FWT 01 (I)
10 FWT 10 (M)
11 FWT 11 (S)

∧IR6(0) ∨ AR0(1) = FWT NEGATE
∧IR6(1) } ∧~IR MD SAC INH = IR MD SAC2
∧IR6(1) = IR MD FAC2

IR7,8
00 } IR MD FC(E)
01 (I)
10 (M) } IR MD SAC INH
11 (B) } IR MD SC(E)

IR3-5
000 IR FWT MOV, S
001 IR FWT MOVN, M
} IR FWT
010 IR MUL
011 IR DIV
01- IR MD
100 IR SH
101 IR 25X
110 IR JP
111 IR AS

010 IR 2XX

IR6
0 DS DIVI
1 DS DIV

(IR6 MD
0 *Integer*
1 *Fraction*)

IR6-8
000 IR ASH
001 IR ROT
010 IR LSH
011
100 IR ASHC
101 IR ROTC
110 IR LSHC
111
} SHIFT OP

IR6-8
000 IR EXCH
001 IR BLT
010 IR AOBJP
011 IR AOBJN
100 IR JRST A ——— ∧~EX IR UUO = IR JRST
101 IR JFCL
110 IR XCT
111
∧IR9∨10(1) ∧ EX USER B(1) = EX IR UUO (*See* IR0-2 = 000)

IR6-8
000 JP PUSHJ
001 JP PUSH
010 JP POP
011 JP POPJ
100 JP JSR
101 JP JSP
110 JP JSA
111 JP JRA
} JP JMP
} JP FLAG STOR

IR6
0 AS + (ADD)
1 AS − (SUB)

IR3-6
0000 BOOLE 0 (SETZ)
0001 BOOLE 1 (AND)
0010 BOOLE 2 (ANDCA)
0011 BOOLE 3 (SETM)
0100 BOOLE 4 (ANDCM)
0101 BOOLE 5 (SETA)
0110 BOOLE 6 (XOR)
0111 BOOLE 7 (IOR)
1000 BOOLE 10 (ANDCB)
1001 BOOLE 11 (EQV)
1010 BOOLE 12 (SETCA)
1011 BOOLE 13 (ORCA)
1100 BOOLE 14 (SETCM)
1101 BOOLE 15 (ORCM)
1110 BOOLE 16 (ORCB)
1111 BOOLE 17 (SETO)

100 IR BOOLE
IR7,8
00 BOOLE ∨ AS 00
01 BOOLE ∨ AS 01 (I)
10 BOOLE ∨ AS 10 (M)
11 BOOLE ∨ AS 11 (B)

101 IR HWT (H)
00 HWT 00
01 HWT 01 (I)
10 HWT 10 (M)
11 HWT 11 (S)

∧IR6(1) = HWT SWAP
(Source)

IR3
0 HWT LT
1 HWT RT
(Destination)

(IR3,6
0..0 LL
0..1 RL
1..0 RR
1..1 LR)

IR4,5
00
01 (Z)
10 (O) } HWT AR ← (0)
11 (E)

∧IR4(1) ∧ [IR5(0) ∨ MB0(1)] = HWT RT SET
∧IR4(1) ∧ [IR5(0) ∨ MB18(1)] = HWT LT SET

011 IR ACCP ∨ MEMAC

IR3,4
00 IR ACCP (CA)
01 IR MEMAC TST
10 IR MEMAC + 1 (AO)
11 IR MEMAC − 1 (SO)
} MEMAC

IR5
0 (1)
1 ACCP DIR (M)
0 MEMAC AC (J)
1 MEMAC MEM (S)

Skip and Jump Conditions
∧IR8(1) ∧ (AR0 ∨ AROV
IR7(1) ∧ AR = 0
} ACCP ETC COND
ACCP ET AL TEST = ACCP ETC COND ∧ IR6(0) ∧ ~ACCP ETC COND ∧ IR6(1)

(IR6-8
000
001 L
010 E
011 LE
100 A
101 GE
110 N
111 G)

IR3,4
00 ACBM DN (N)
01 ACBM CL (Z)
10 ACBM COM (C)
11 ACBM SET (O)

110 IR ACBM (T)
∧IR5(1) = ACBM DIR
(IR5,8
0..0 R
0..1 L
1 0 D
1..1 S)
(IR6,7
00
01 E
10 A
11 N)
∧IR8(1) = ACBM SWAP

IR10-12
000 IOT BLKI
001 IOT DATAI
010 IOT BLKO
011 IOT DATAO
100 IOT CONO
101 IOT CONI
110 IOT CONSZ
111 IOT CONSO
IOT BLK
IOT DATAI ∨ O
IOT OUTGOING
IOT STATUS

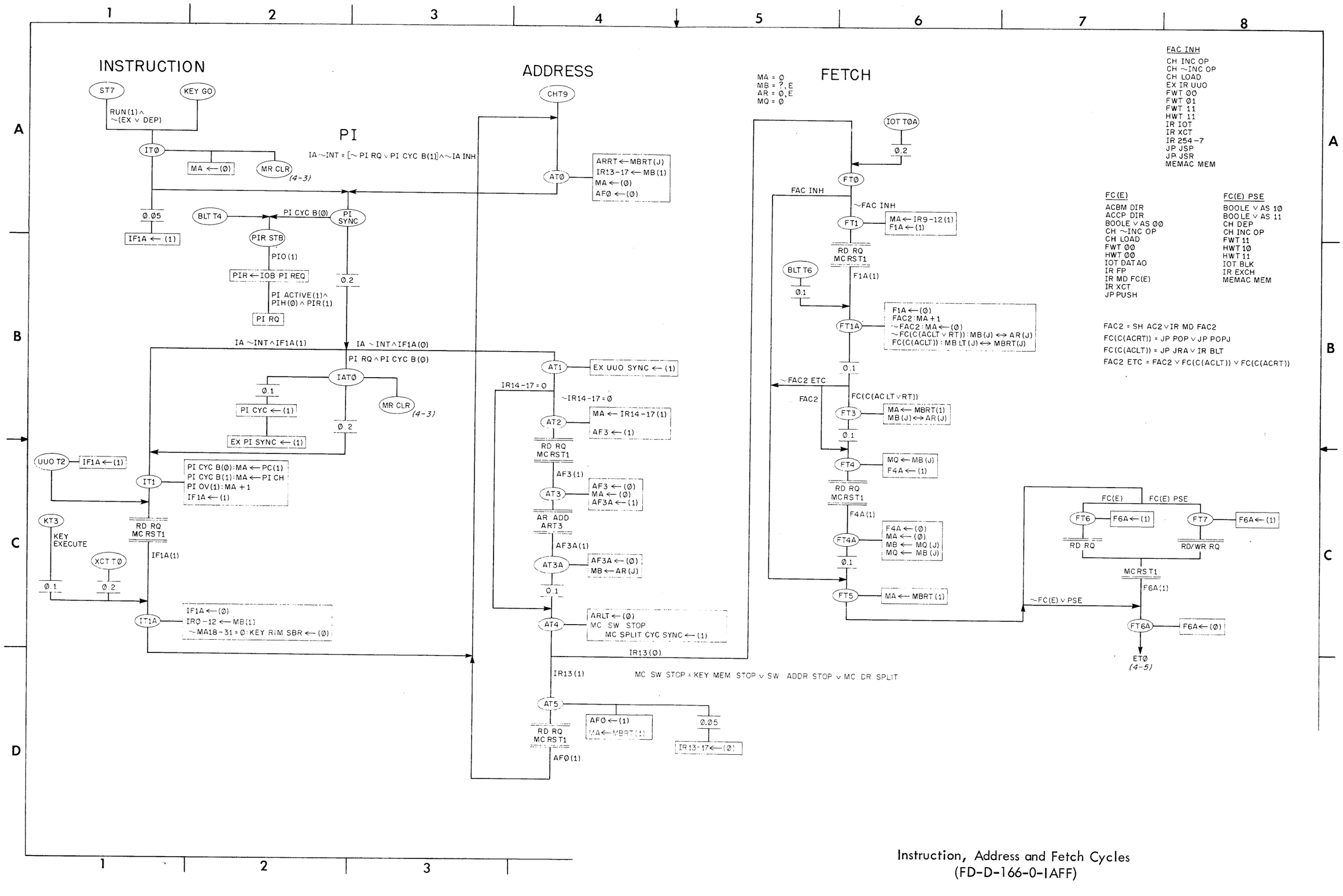
111 IR IOT A ——— ∧~EX IR UUO = IR IOT
∧EX PI SYNC(0) ∧ EX USER B(1) = EX IR UUO (*See* IR0-2 = 000)

Instruction Decoding
(FD-D-166-0-IRD)

AP FIGURE 4-3

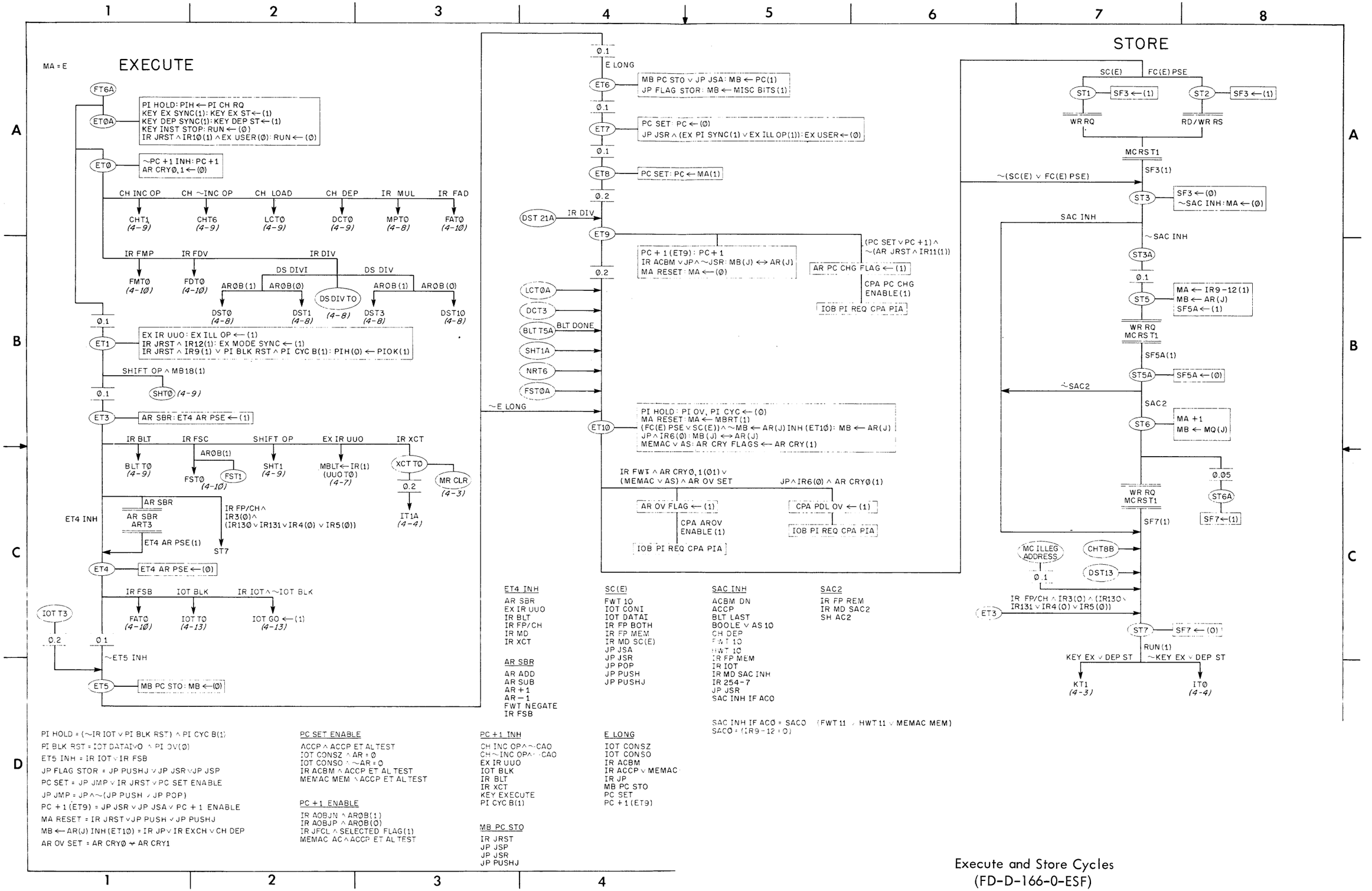
Key Cycle; Memory Subroutine
(FD-D-166-0-KMF)

Instruction, Address and Fetch Cycles
(FD-D-166-0-1AFF)



Instruction, Address and Fetch Cycles
(FD-D-166-0-1AFF)

Execute and Store Cycles
(FD-D-166-0-ESF)



Execute and Store Cycles
(FD-D-166-0-ESF)

HALF WORD TRANSFER
500 - 577

IR = 101 WXX YZZ
IR HWT = IR0-2(101)
W *specifies destination half*
HWT LT = IR HWT ∧ IR3(0)
HWT RT = IR HWT ∧ IR3(1)
XX *specifies action on other half - do nothing, zero, one, extend*
HWT AR ← (0) = IR HWT ∧ [IR4(1) ∨ IR5(1)]
HWT RT SET = HWT LT ∧ IR4(1) ∧ [IR5(0) ∨ MB0(1)]
HWT LT SET = HWT RT ∧ IR4(1) ∧ [IR5(0) ∨ MB18(1)]
Y *specifies source half*
HWT SWAP = IR HWT ∧ IR6(1)
ZZ *specifies mode*

IR HWT ∧ IR7,8 {
00 = HWT 00 — Action: ↓[C(E), AC] → AC
01 = HWT 01 — Action: ↓[(0,E), AC] → AC
10 = HWT 10 — Action: ↓[AC, C(E)] → E
11 = HWT 11 — Action: ↓[C(E)] → E
}

Initial Registers

HWT 00: AR = AC MB = C(E)
HWT 01: AR = AC MB = 0,E
HWT 10: AR = AC MB = C(E)
HWT 11: AR = 0,E MB = C(E)
MQ = 0

Initial Gates

HWT 00: FC(E)
HWT 10: FC(E) PSE
HWT 11: FAC INH, FC(E) PSE

ET0A
HWT 10: MB(J) ⇔ AR(J)
HWT 11: AR ← MB(J)

ET1
HWT SWAP: MB LT(J) ⇔ MB RT(J)
HWT AR ← (0): AR ← (0)

ET3

ET4
HWT LT SET: AR LT COM
HWT RT SET: AR RT COM
HWT LT: AR ← MB LT(J)
HWT RT: AR ← MB RT(J)

ET5

ET10
HWT 10 ∨ 11: MB ← AR(J)

Final Gates
HWT 10: FC(E) PSE, SAC INH
HWT 11 FC(E) PSE
HWT 11 SACØ: SAC INH
SACØ = (IR9-12 = Ø)

FULL WORD TRANSFER
200 - 217

IR = 010 00Y YZZ
IR 2XX = IR0-2(010)
YY *specifies instruction* — *MOVE, MOVS, MOVN, MOVM*
IR MOV, S = IR 2XX ∧ IR3-5(000)
IR MOVN, M = IR 2XX ∧ IR3-5(001)
IR FWT = IR MOV, S ∨ IR MOVN, M
FWT SWAP = IR FWT ∧ IR5,6 (01)
FWT NEGATE = IR MOVN, M ∧ [IR6(0) ∨ AR0(1)]
ZZ *specifies mode*

IR FWT ∧ IR7,8 {
00 = FWT 00 — Action: ↓[C(E)] → AC
01 = FWT 01 — Action: ↓[0,E] → AC
10 = FWT 10 — Action: ↓[AC] → E
11 = FWT 11 — Action: ↓[C(E)] → E
}

Initial Registers

FWT 00: AR = 0,E MB = C(E)
FWT 01: AR = 0,E MB = ?,E
FWT 10: AR = AC MB = 0,E
FWT 11: AR = 0,E MB = C(E)
MQ = 0

Initial Gates

FWT 00: FAC INH, FC(E)
FWT 01: FAC INH
FWT 11: FAC INH, FC(E) PSE
FWT NEGATE: ET4 INH, AR SBR

ET0A
FWT 00 ∨ 11: AR ← MB(J)
FWT 01 ∨ 10: MB ← AR(J)
MB(J) ⇔ AR(J)

ET1
FWT SWAP: MB LT(J) ⇔ MB RT(J)

ET3
FWT NEGATE: ET4 AR PSE ← (1) ‖ AR NEGATE ART3‖

ET4
ET4 AR PSE ← (0)
FWT SWAP: MB(J) ⇔ AR(J)

ET5

ET10
AR CRY0,1 (01): AR OV FLAG ← (1)
FWT 10 ∨ 11: MB ← AR(J)

Final Gates
FWT 10: SC(E), SAC INH
FWT 11: FC(E) PSE
FWT 11 ∧ SACØ: SAC INH
SACØ = (IR9-12 = Ø)

EXCH
250

IR = 010 101 000
IR 2XX = IR0-2(010)
IR 25X = IR 2XX ∧ IR3-5(101)
IR EXCH = IR 25X ∧ IR6-8(000)

AC ⇔ C(E)

AC = AC
MB = C(E)
MQ = 0

FC(E) PSE
MB ← AR(J) INH(ET10)

MB(J) ⇔ AR(J)

ARITHMETIC COMPARE
300 - 377

IR = 011 VVW XYZ
IR ACCP ∨ MEMAC = IR0-2(011)
VV *specifies instruction type*

(IR ACCP ∨ MEMAC) ∧ IR3,4 {
00 = ACCP
01 = MEMAC TST
10 = MEMAC +1
11 = MEMAC -1
}

MEMAC = MEMAC TST ∨ MEMAC +1 ∨ MEMAC -1
X *specifies whether or not action is on condition YZ*
Y *specifies condition 'equals'*
Z *specifies condition 'less than'*
XYZ *together define relation R*

XYZ	R
000	*Never*
001	<
010	=
011	≤
100	*Always*
101	≥
110	≠
111	>

Determination of Skip or Jump Condition
ACCP ETC COND = [IR7(1) ∧ (AR = 0)] ∨ [(IR ACCP ∨ MEMAC) ∧ IR8(1) ∧ (AR0 ≠ AR OV)]
ACCP ET AL TEST = ACCP ETC COND ≠ IR6
AR0 ≠ AR OV = AR0 ≠ (AR OV SET ∧ ~MEMAC)
AR OV SET = AR CRY0 ≠ AR CRY1

Initial Registers

AR = AC
MB = 0,E
MQ = 0

Initial Gates

MEMAC ±1: ET4 INH
AR SBR
E LONG

ET0A

ET1

ET3
MEMAC +1: ET4 AR PSE ← (1) ‖ AR +1 ‖
MEMAC -1: ET4 AR PSE ← (1) ‖ AR -1 ART3‖

ET4
ET4 AR PSE ← (0)

ET5

ET6

ET7
PC SET: PC ← (0)

ET8
PC SET: PC ← MA(1)

ET9
PC SET:
AR PC CHG FLAG ← (1)

ET10
AR OV SET:
AR OV FLAG ← (1)
AR CRY FLAGS ← AR CRY(1)

Final Gates

MEMAC
320 - 377

W *specifies whether* AC *or* C(E) *compared with zero, i.e whether action is jump or skip*
MEMAC AC = MEMAC ∧ IR5(0)
MEMAC MEM = MEMAC ∧ IR5(1)

MEMAC AC	MEMAC MEM
AC R 0: E → PC	C(E) R 0: PC +1
MEMAC ±1: AC ±1 → AC	MEMAC ±1: C(E) ±1 → E
ACCP ET AL TEST: PC SET	~AC0 ∧ MEMAC ±1: C(E) ±1 → AC
	SAC0 = (IR9-12 = 0)

AR = 0,E
MB = C(E)
MQ = 0

FAC INH
FC(E) PSE
MEMAC ±1: ET4 INH
AR SBR
E LONG

AR ← MB(J)

MEMAC +1: ET4 AR PSE ← (1) ‖ AR +1 ‖
MEMAC -1: ET4 AR PSE ← (1) ‖ AR -1 ART3‖

ET4 AR PSE ← (0)

PC SET: PC ← (0)

PC SET: PC ← MA(1)

ACCP ET AL TEST:
PC + 1
AR PC CHG FLAG ← (1)

AR OV SET:
AR OV FLAG ← (1)
AR CRY FLAGS ← AR CRY(1)
MB ← AR(J)

FC(E) PSE
SAC0: SAC INH

ACCP
300 - 317

W *specifies whether* AC *compared with* E *or* C(E)
ACCP DIR = ACCP ∧ IR5(1)
IR5(0) ∧ [AC R (0,E)]: PC + 1
IR5(1) ∧ [AC R C(E)]: PC + 1

AR = AC
~ACCP DIR: MB = 0,E
ACCP DIR: MB = C(E)
MQ = 0

ACCP DIR: FC(E)
ET4 INH
AR SBR
E LONG

ET0A

ET1

ET3
ET4 AR PSE ← (1) ‖ AR SUB ART3‖

ET4
ET4 AR PSE ← (0)

ET5

ET6

ET7

ET8

ET9
ACCP ET AL TEST:
PC + 1
AR PC CHG FLAG ← (1)

ET10

Final Gates
SAC INH

ACBM

R [*Mask* ∧ AC]: PC + 1
↓[*Mask*, AC] → AC

AR = AC
~ACBM DIR: MB = 0,E
ACBM DIR: MB = C(E)
MQ = 0

ACBM DIR: FC(E)
E LONG

ACBM SWAP: MB LT(J) ⇔ MB RT(J)

MB ← AR(0)
ACBM COM: AR ← MB(≠)
ACBM SET: AR ← MB(1)

AR COM

ACBM CL: MB ← AR(0)

AR COM

MB(J) ⇔ AR(J)
ACCP ET AL TEST: PC + 1
AR PC CHG FLAG ← (1)

ACBM DN: SAC INH

LOGICAL COMPARE
600 - 677

IR = 110 VVW XYZ
IR ACBM = IR0-2(110)
VV *specifies action on masked bits*

IR ACBM ∧ IR3,4 {
00 = ACBM DN
01 = ACBM CL
10 = ACBM COM
11 = ACBM SET
}

W *specifies source of mask - immediate or memory*
ACBM DIR = IR ACBM ∧ IR5(1)
X *specifies whether or not action is on condition Y, i.e 'equals zero'*
XY *together define relation* R

XY	R
00	*Never*
01	=0
10	*Always*
11	≠0

Z *specifies swapping mask selected by* W
ACBM SWAP = IR ACBM ∧ IR8(1)
WZ *together specify mask as* (0,E), (E,0), C(E), C(E) RT LT

AR = AC
~ACBM DIR: MB = 0,E
ACBM DIR: MB = C(E)
MQ = 0

ACBM DIR: FC(E)
E LONG

ET4 INH
AR SBR
E LONG

AOBJP
252

IR = 010 101 01X
IR 2XX = IR0-2(010)
IR 25X = IR 2XX ∧ IR3-5(101)
X *specifies instruction*
IR AOBJP = IR 25X ∧ IR6-8(010)
IR AOBJN = IR 25X ∧ IR6-8(011)

AC + 1000001 → AC
AOBJP ∧ (AC ≥ 0): E → PC
AOBJN ∧ (AC < 0): E → PC

IR AOBJP ∧ AR0B(0): PC SET
IR AOBJN ∧ AR0B(1): PC SET

AR = AC
MB = 0,E
MQ = 0

ET4 INH
AR SBR
E LONG

AOBJN
253

JUMP AND PUSHDOWN

260–267

IR = 010 110 XXX
IR 2XX = IR0-2(010)
IR JP = IR 2XX ∧ IR3-5(110)

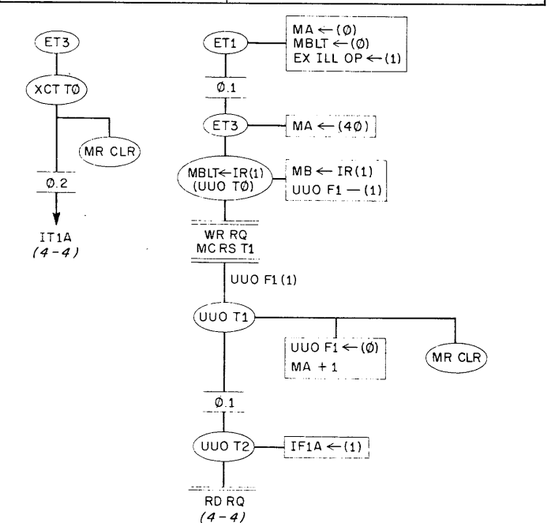
XXX *specifies instruction*

IR JP ∧ IR6-8 {
000 = JP PUSHJ
001 = JP PUSH
010 = JP POP
011 = JP POPJ
100 = JP JSR
101 = JP JSP
110 = JP JSA
111 = JP JRA

JP JMP = IR JP ∧ ~(JP PUSH ∨ JP POP)
JP FLAG STOR = JP PUSHJ ∨ JP JSR ∨ JP JSP
~IR IOT ∧ PI CYC B(1): PI HOLD

	PUSHJ 260	PUSH 261	POP 262	POPJ 263	JSR 264	JSP 265	JSA 266	JRA 267
	AC+1000001 → AC MISC BITS, PC → C(C(ACRT)) E → PC PI ON OVERFLOW	AC+1000001 → AC C(E) → C(C(ACRT)) PI ON OVERFLOW	C(C(ACRT)) → E AC −1000001 → AC PI ON UNDERFLOW	C(C(ACRT)) → PC AC−1000001 → AC PI ON UNDERFLOW	MISC BITS, PC → E E + 1 → PC	MISC BITS, PC → AC E → PC	AC → E E, PC → AC E + 1 → PC	C(C(ACLT)) → AC E → PC
Initial Registers	AR = AC MB = 0, E MQ = 0	AR = AC MB = C(E) MQ = 0	AR = AC MB = 0, E MQ = C(C(ACRT))	AR = AC MB = 0, E MQ = C(C(ACRT))	AR = 0, E MB = ?, E MQ = 0	AR = 0, E MB = ?, E MQ = 0	AR = AC MB = 0, E MQ = 0	AR = AC RTLT MB = 0, E MQ = C(C(ACLT))
Initial Gates	ET4 INH AR SBR E LONG MB ← AR(J) INH (ET10)	FC(E) ET4 INH AR SBR E LONG MB ← AR(J) INH (ET10)	FC(C(ACRT)) ET4 INH AR SBR E LONG MB ← AR(J) INH (ET10)	FC(C(ACRT)) ET4 INH AR SBR E LONG MB ← AR(J) INH (ET10)	FAC INH E LONG MB ← AR(J) INH (ET10)	FAC INH E LONG MB ← AR(J) INH (ET10)	E LONG MB ← AR(J) INH (ET10)	FC(C(ACLT)) E LONG MB ← AR(J) INH (ET10)
ET0A			MB ← MQ(J)	MB ← MQ(J)	PI HOLD: PIH ← PI CH RQ	MB(J) ↔ AR(J)	MBLT(J) ↔ MBRT(J)	MB ← MQ(J)
ET1				MA ← (0)				
ET3	ET4 AR PSE ← (1) ‖AR + 1 LTRT ART3‖	ET4 AR PSE ← (1) ‖AR + 1 LTRT ART3‖	ET4 AR PSE ← (1) ‖AR − 1 LTRT ART3‖	ET4 AR PSE ← (1) ‖AR − 1 LTRT ART3‖				
ET4	ET4 AR PSE ← (0)	ET4 AR PSE ← (0)	ET4 AR PSE ← (0)	ET4 AR PSE ← (0)				
ET5	MB ← (0)				MB ← (0)	MB ← (0)		
ET6	MB ← MISC BITS, PC(1)				MB ← MISC BITS, PC(1)	MB ← MISC BITS, PC(1)	MB ← PC(1)	
ET7	PC ← (0)			PC ← (0)	PC ← (0) EX PI SYNC(1) ∨ EX ILL OP(1): EX USER ← (0)	PC ← (0)	PC ← (0)	PC ← (0)
ET8	PC ← MA(1)			PC ← MA(1)	PC ← MA(1) EX ILL OP ← (0)	PC ← MA(1)	PC ← MA(1)	PC ← MA(1)
ET9	MB(J) ↔ AR(J) MA ← (0)	MB(J) ↔ AR(J) MA ← (0)	MB(J) ↔ AR(J)	MB(J) ↔ AR(J)	PC + 1 CHF7 ← (0)	MB(J) ← AR(J)	MB(J) ↔ AR(J) PC + 1	MB(J) ↔ AR(J)
ET10	MA ← MBRT(1) MB(J) ↔ AR(J) AR CRY0(1): CPA PDL OV ← (1) IOB PI REQ CPA PIA		MB(J) ↔ AR(J) AR CRY0(1): CPA PDL OV ← (1) IOB PI REQ CPA PIA		PI HOLD: PI OV, PI CYC ← (0)			
Final Gates	SC(E)	SC(E)	SC(E)		SC(E) SAC INH		SC(E)	

	JRST 254	JFCL 255	XCT 256	UUO 000–077
	IR = 010 101 100 IR 2XX = IR0-2(010) IR 25X = IR2XX ∧ IR3-5(101) IR JRST A = IR 25X ∧ IR 6-8(100) IR JRST = IR JRST A ∧ ~EX IR UUO	IR = 010 101 101 IR 2XX = IR0-2(010) IR 25X = IR 2XX ∧ IR3-5(101) IR JFCL = IR 25X ∧ IR 6-8(010)	IR = 010 101 110 IR 2XX = IR0-2(010) IR 25X = IR 2XX ∧ IR3-5(101) IR XCT = IR 25X ∧ IR6-8(110) EXECUTE C(E)	IR = 000 XXX XXX IR UUO A = IR0-2(000) EX IR UUO = IR UUO A ∧ EX UUO SYNC(1) ∨ IR JRST A ∧ EX USER B(1) ∧ IR9 ∨ 10(1) ∨ IR IOT A ∧ EX USER B(1) ∧ EX PI SYNC(0) ∧ CPA IOT USER(0) IR, E → 40 EXECUTE 41
	E → PC IR9(1): DISMISS PI IR10(1): HALT IR11(1): RESTORE MISC BITS IR12(1): ENTER USER MODE	SELECTED FLAG(1): E → PC CLEAR SELECTED FLAGS		
Initial Registers	AR = 0, E MB = ?, E MQ = 0	AR = 0, E MB = ?, E MQ = 0	AR = 0, E MB = C(E) MQ = 0	AR = 0, E MB = ?, E MQ = 0
Initial Gates	FAC INH E LONG	FAC INH E LONG	FAC INH FC(E) PC + 1 INH ET4 INH	FAC INH PC + 1 INH ET4 INH
ET0A	IR10(1): RUN ← (0) IR11(1): AR FLAG CLEAR (*Clears* MISC BITS *at* ET0)			
ET1	IR9(1): PIH(0) ← PIOK(1) IR11(1): AR FLAG SET (*Transfers* MB(1) *into* MISC BITS) IR12(1): EX MODE SYNC ← (1)			
ET3				
ET4				
ET5	MB ← (0)			
ET6	MB ← PC(1)			
ET7	PC ← (0)	SELECTED FLAG(1): PC ← (0)		
ET8	PC ← MA(1)	SELECTED FLAG(1): PC ← MA(1)		
ET9	MA ← (0)			
ET10	MA ← MBRT(1)	SELECTED FLAGS ← (0)		
Final Gates	SAC INH	SAC INH		



MISC BITS

0 AR OV FLAG
1 AR CRY0 FLAG
2 AR CRY1 FLAG
3 AR PC CHG FLAG
4 CHF7
5 { EX USER – STORE ONLY
  { EX MODE SYNC – SET ONLY

SELECTED FLAGS

IR9(1) AR OV FLAG
IR10(1) AR CRY0 FLAG
IR11(1) AR CRY1 FLAG
IR12(1) AR PC CHG FLAG

Pushdown and Jump Instructions
(FD-D-166-0-JPF)

BOOLEAN

400-477

IR = 100 XXX XYY
IR BOOLE = IR0-2(100)
XXXX *specifies Boolean function, decoded into* BOOLE 0-17 *as*
BOOLE N = IR BOOLE ∧ IR3-6(N)

ADD-SUBTRACT

270-277

IR = 010 111 XYY
IR 2XX = IR0-2(010)
IR AS = IR 2XX ∧ IR3-5 (111)
X *specifies add or subtract*

IR AS ∧ IR6 { 0 = AS +
1 = AS −

In both instruction groups YY *specifies common mode*

(IR BOOLE ∨ IR AS) ∧ IR7,8 { 00 = BOOLE ∨ AS 00
01 = BOOLE ∨ AS 01
10 = BOOLE ∨ AS 10
11 = BOOLE ∨ AS 11

Action

	BOOLE	AS
BOOLE ∨ AS 00:	f[AC, C(E)] ⟶ AC	AC ± C(E) ⟶ AC
BOOLE ∨ AS 01:	f[AC, (0, E)] ⟶ AC	AC ± (0, E) ⟶ AC
BOOLE ∨ AS 10:	f[AC, C(E)] ⟶ E	AC ± C(E) ⟶ E
BOOLE ∨ AS 11:	f[AC, C(E)] ⟶ AC, E	AC ± C(E) ⟶ AC, E

Initial Registers

AR = AC
BOOLE ∨ AS 00 ∨ 10 ∨ 11: MB = C(E)
BOOLE ∨ AS 01: MB = (0, E)
MQ = 0

Common Initial Gates

BOOLE ∨ AS 00: FC(E)
BOOLE ∨ AS 10 11: FC(E) PSE

Initial Gates AS *Only*

ET4 INH
AR SBR

ET0A	BOOLE 0, 3, 14, 17: AR ⟵ (0) BOOLE 2, 4, 12, 13, 15: AR COM	
ET1	BOOLE 6, 11, 14: AR ⟵ MB (∨) BOOLE 1, 2, 15, 16: AR ⟵ MB (0) BOOLE 3, 4, 7, 10, 13: AR ⟵ MB (1)	
ET3		ET4 AR PSE ⟵ (1) AS +: AR ADD AS −: AR SUB ART3
ET4	BOOLE 4, 10, 11, 14, 15, 16, 17: AR COM	ET4 AR PSE ⟵ (0)
ET5		
ET10	FC(E) PSE: MB ⟵ AR(J)	FC(E) PSE: MB ⟵ AR(J) AR OV SET: AR OV FLAG ⟵ (1) AR CRY FLAGS ⟵ AR CRY(1) AR OV SET = AR CRY0 ∨ AR CRY1

Final Gates

BOOLE ∨ AS 10: FC(E) PSE, SAC INH
BOOLE ∨ AS 11: FC(E) PSE

BOOLEAN FUNCTIONS

	0 SETZ	1 AND	2 ANDCA	3 SETM
ET0A	AR ⟵ (0)		AR COM	AR ⟵ (0)
ET1		AR ⟵ MB(0)	AR ⟵ MB(0)	AR ⟵ MB(1)
ET4				

	4 ANDCM	5 SETA	6 XOR	7 IOR
ET0A	AR COM			
ET1	AR ⟵ MB(1)		AR ⟵ MB(∨)	AR ⟵ MB(1)
ET4	AR COM			

	10 ANDCB	11 EQV	12 SETCA	13 ORCA
ET0A			AR COM	AR COM
ET1	AR ⟵ MB(1)	AR ⟵ MB(∨)		AR ⟵ MB(1)
ET4	AR COM	AR COM		

	14 SETCM	15 ORCM	16 ORCB	17 SETO
ET0A	AR ⟵ (0)	AR COM		AR ⟵ (0)
ET1	AR ⟵ MB(∨)	AR ⟵ MB(0)	AR ⟵ MB(0)	
ET4	AR COM	AR COM	AR COM	AR COM

MULTIPLY-DIVIDE

220-237

IR = 010 01X YZZ
IR 2XX = IR0-2(010)
X *specifies operation*

IR 2XX ∧ IR3-5 { 010 = IR MUL
011 = IR DIV

Y *specifies operand type:* IR6 *is gate in multiply flow; for divide*
DS DIVI = IR DIV ∧ IR6(0)
DS DIV = IR DIV ∧ IR6(1)

ZZ *specifies mode*

Levels for fetch and store according to operand type and mode

IR MD = IR 2XX ∧ IR3(0) ∧ IR4(1)
IR MD FC(E) = IR MD ∧ ~(IR7(0) ∧ IR8(1))
IR MD FAC2 = IR DIV ∧ IR6(1)
IR MD SC(E) = IR MD ∧ IR7(1)
IR MD SAC INH = IR MD ∧ IR7(1) ∧ IR8(0)
IR MD SAC2 = ~IR MD SAC INH ∧ (IR DIV ∨ IR MUL ∧ IR6(1))

IR MUL

Action

For IMUL (XY = 00) *store only low order half of product*
ZZ = 00: AC × C(E) ⟶ AC
ZZ = 01: AC × (0, E) ⟶ AC
ZZ = 10: AC × C(E) ⟶ E
ZZ = 11: AC × C(E) ⟶ AC; E

For MUL (XY = 01) *store entire product or high order half*
ZZ = 00: AC × C(E) ⟶ AC, AC2
ZZ = 01: AC × (0, E) ⟶ AC, AC2
ZZ = 10: AC × C(E) ⟶ E
ZZ = 11: AC × C(E) ⟶ AC, AC2; E

Initial Registers

AR = AC
ZZ = 00 ∨ 10 ∨ 11: MB = C(E)
ZZ = 01: MB = 0, E
MQ = 0

Initial Gates

ZZ = 00 ∨ 10 ∨ 11: FC(E)
ET4 INH

IDIV (XY = 10)

Action

ZZ = 00: AC ÷ C(E) ⟶ AC, AC2
ZZ = 01: AC ÷ (0, E) ⟶ AC, AC2
ZZ = 10: AC ÷ C(E) ⟶ E
ZZ = 11: AC ÷ C(E) ⟶ AC, AC2; E

Initial Registers

AR = AC
ZZ = 00 ∨ 10 ∨ 11: MB = C(E)
ZZ = 10: MB = 0, E
MQ = 0

Initial Gates

ZZ = 00 ∨ 10 ∨ 11: FC(E)
ET4 INH

DIV (XY = 11)

Action

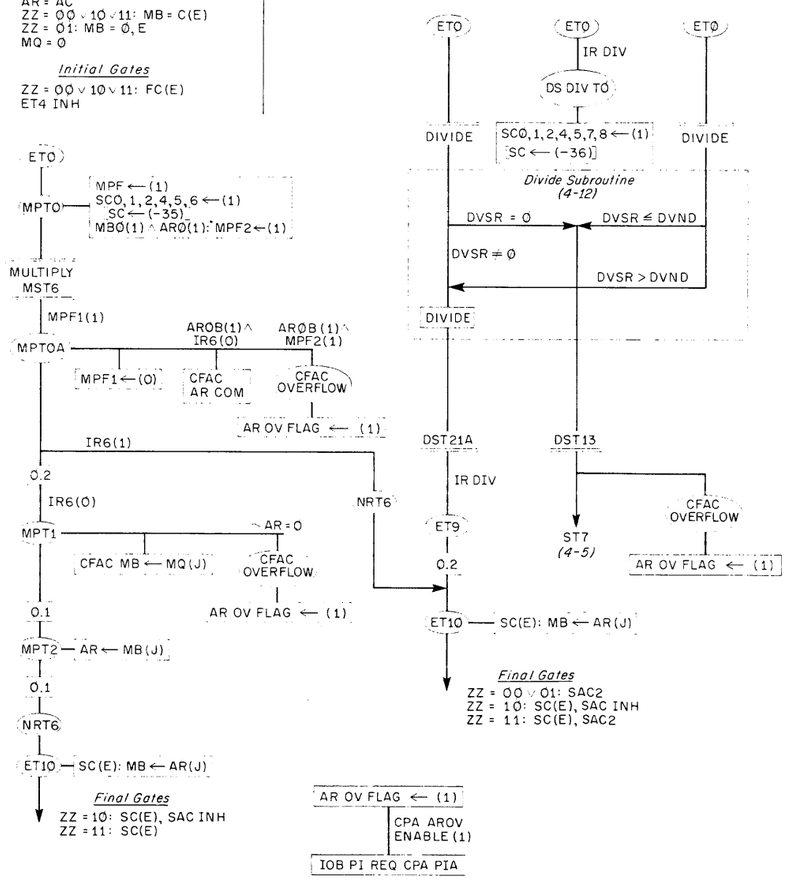
ZZ = 00: AC, AC2 ÷ C(E) ⟶ AC, AC2
ZZ = 01: AC, AC2 ÷ (0, E) ⟶ AC, AC2
ZZ = 10: AC, AC2 ÷ C(E) ⟶ E
ZZ = 11: AC, AC2 ÷ C(E) ⟶ AC, AC2; E

Initial Registers

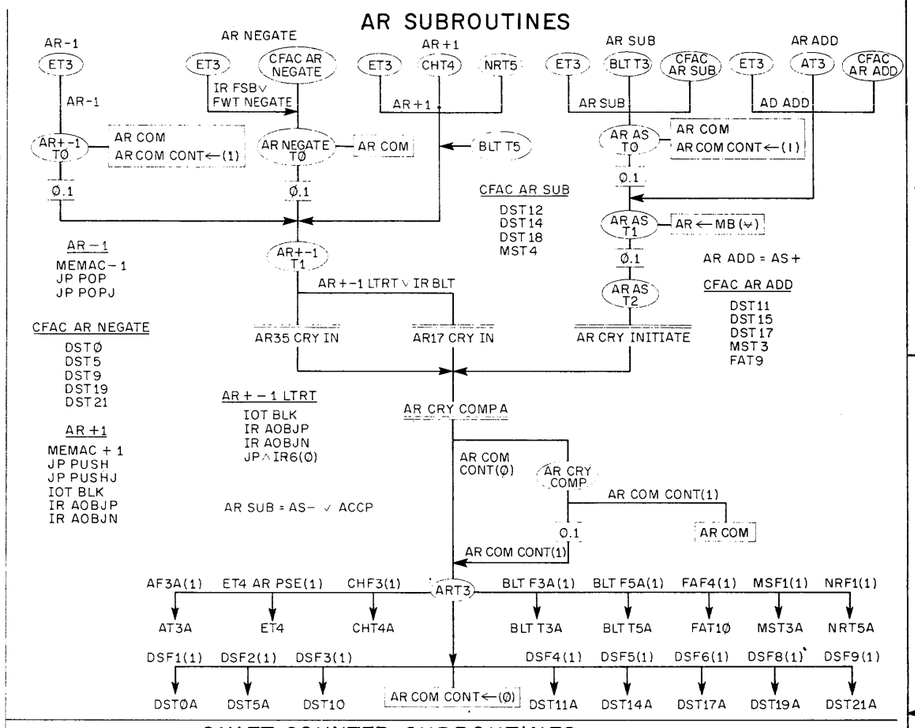
AR = AC
ZZ = 00 ∨ 10 ∨ 11: MB = C(E)
ZZ = 10: MB = 0, E
MQ = AC2

Initial Gates

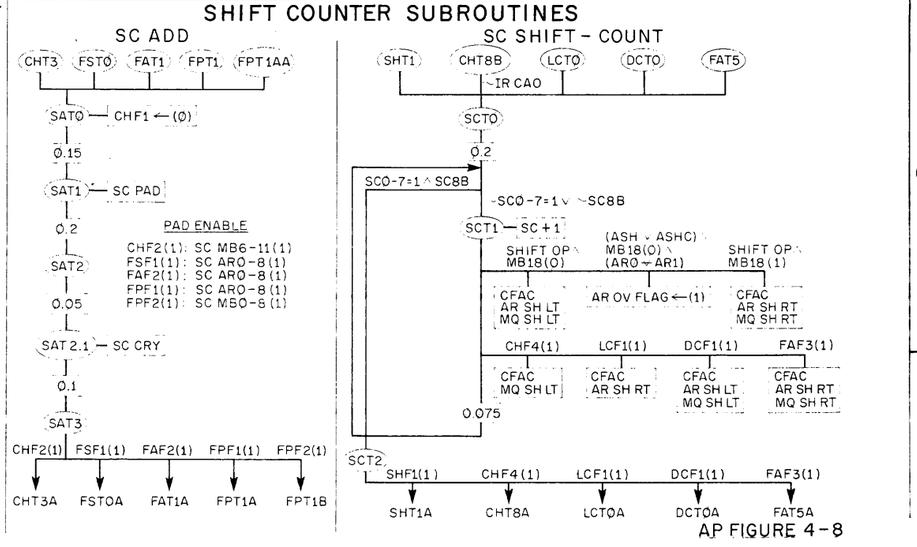
ZZ = 00 ∨ 10 ∨ 11: FC(E)
FAC2
ET4 INH



AR SUBROUTINES



SHIFT COUNTER SUBROUTINES



AP FIGURE 4-8

Fixed Point and Logical Instructions; AR and SC Subroutines
(FD-D-166-0-ALF)

CHARACTER OPERATIONS
133-137

IR = 001 011 XXX
IR FP/CH = IR0-2(001)
IR FP/CH3 = IR FP/CH ∧ IR3-5(011)
XXX specifies instruction

IR FP/CH3 ∧ IR6-8 {
011 = IR CAO
100 = IR LDC1
101 = IR LDC
110 = IR DPC1
111 = IR DPC }

FIRST PART-CHF5(0)

CH INC = (IR LDCI ∨ IR DPCI ∨ IR CAO) ∧ CHF5(0)
CH INC OP = CH INC ∧ CHF7(0)
CH ~INC OP = [(IR LDC ∨ IR DPC) ∧ CHF5(0)] ∨ [CH INC ∧ CHF7(1)]

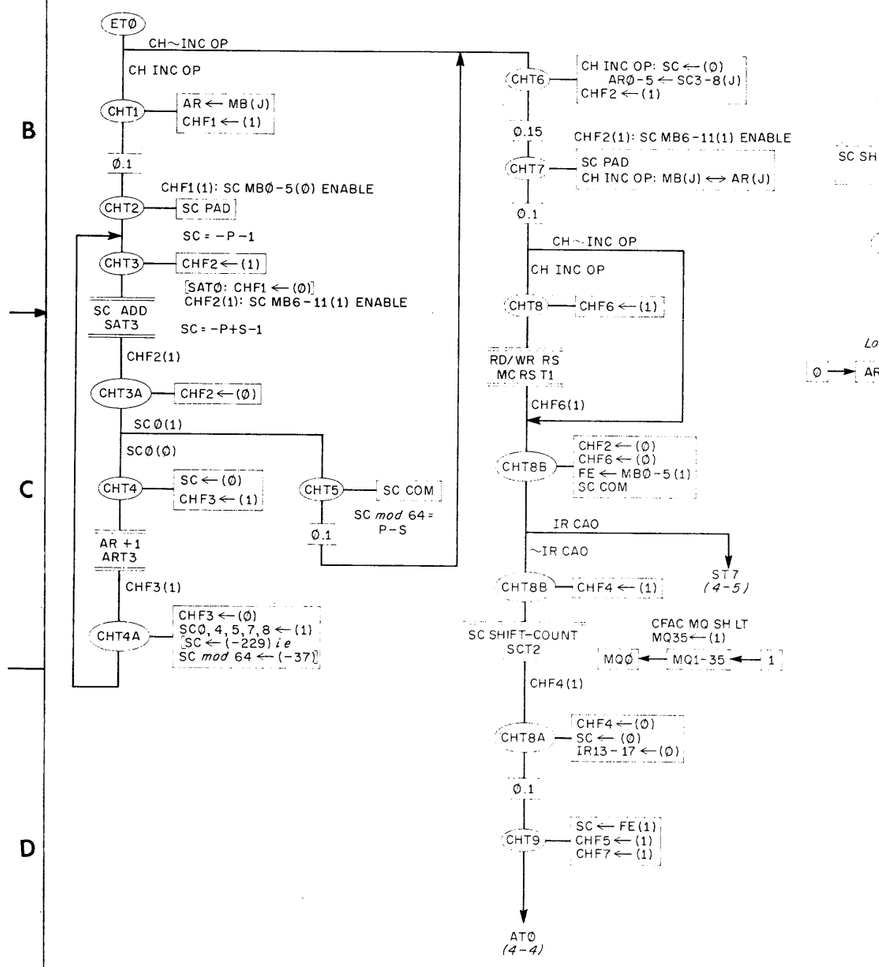
Pointer Format

P	S		I	X	Y
0-5	6-11	12	13	14-17	18-35

Fetch and increment pointer

Initial Registers
AR = 0,E
MB = C(E)
MQ = 0

Initial Gates
CH INC OP: FAC INH, FC(E) PSE
CH ~INC OP: FAC INH, FC(E)
~IR CAO: PC+1 INH
ET4 INH



SECOND PART-CHF5(1)

CH LOAD = (IR LDC ∨ IR LDCI) ∧ CHF5(1)
CH DEP = (IR DPC ∨ IR DPCI) ∧ CHF5(1)

CH LOAD
Fetch and load character

CH DEP
Fetch and deposit character

Initial Registers

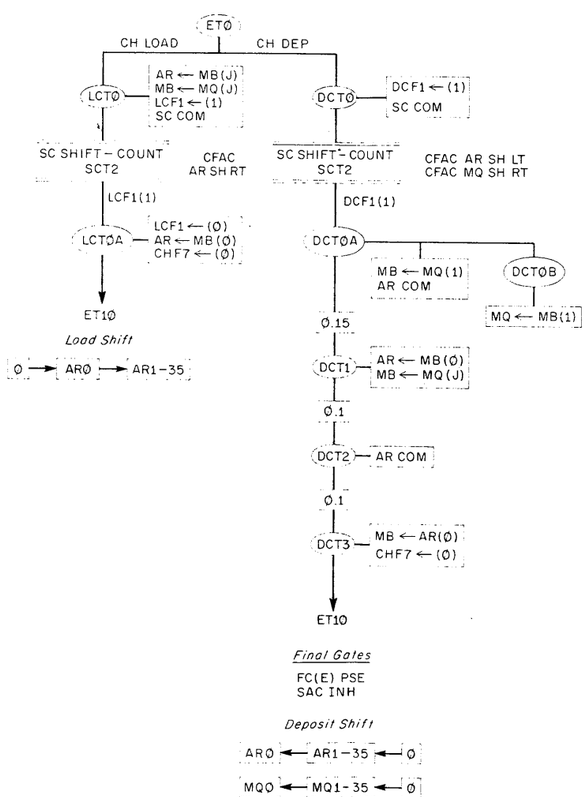
AR = 0,E
MB = C(E)
MQ = 0...0 1...1 (S)

AR = 0,E
MB = C(E)
MQ = 0...0 1...1 (S)

Initial Gates

FAC INH
FC(E)
ET4 INH

FC(E) PSE
ET4 INH
MB ← AR(J) INH(ET0)



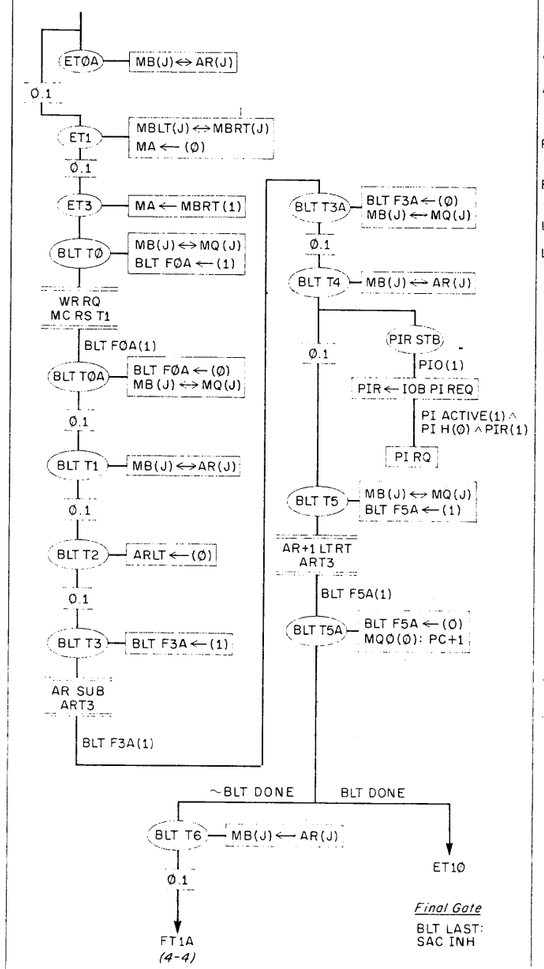
BLT
251

IR = 010 101 001
IR 2XX = IR0-2(010)
IR 25X = IR 2XX ∧ IR3-5(101)
IR BLT = IR 25X ∧ IR6-8(001)
BLT DONE = MQ0(0) ∨ PI RQ
BLT LAST = IR BLT ∧ MQ0(0)

S = *Source address*
D = *Destination address*
C(S) → D
(S,D) + 1000001 → (S,D)
Repeat until D = E

Initial Registers
AR = D,S *initially* AC RT LT
MB = 0,E
MQ = C(S) *initially* C(C(ACLT))

Initial Gates
FC(C(ACLT))
PC+1 INH
ET4 INH



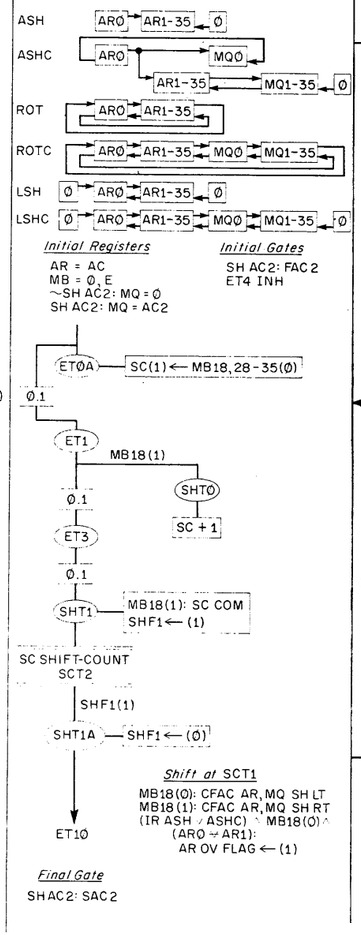
SHIFT OPERATIONS
240-242, 244-246

IR = 010 100 XXX
IR 2XX = IR0-2(010)
IR SH = IR 2XX ∧ IR3-5(100)

IR SH ∧ IR6-8 {
000 = IR ASH
001 = IR ROT
010 = IR LSH
100 = IR ASHC
101 = IR ROTC
110 = IR LSHC }

SHIFT OP = IR ASH ∨ IR ROT ∨ IR LSH ∨ IR ASHC ∨ IR ROTC ∨ IR LSHC
SH AC2 = IR ASHC ∨ IR ROTC ∨ IR LSHC
Shift E *mod* 256 *places*
~SH AC2: *f* [AC] → AC
SH AC2: *f* [AC, AC2] → AC

SHIFT OP AR, MQ CONNECTIONS
IR ASHC: SHC ASHC



FLOATING POINT

140-177

XX specifies instruction

IR = ØØ1 1XX YZZ
IR FP/CH = IRØ-2(ØØ1)
IR FP = IR FP/CH ∧ IR3(1)
IR FP ∧ IR4,5 { ØØ = IR FAD; Ø1 = IR FSB; 1Ø = IR FMP; 11 = IR FDV }
Y specifies rounding, see Figure 4-11

ZZ specifies mode

IR FP ∧ IR7,8	Action
ØØ = IR FP DIR	AC R C(E) → AC
Ø1 = IR FP REM	AC R C(E) → AC, AC2
1Ø = IR FP MEM	AC R C(E) → E
11 = IR FP BOTH	AC R C(E) → E, AC

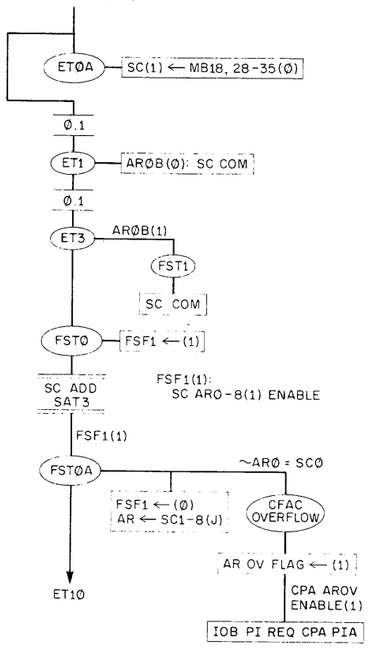
Operator R = +, −, ×, ÷

FSC

132

IR = ØØ1 Ø11 Ø1Ø
IR FP/CH = IRØ-2(ØØ1)
IR FP/CH3 = IR FP/CH ∧ IR3-5(Ø11)
IR FSC = IR FP/CH3 ∧ IR6-8(Ø1Ø)
AC × 2^E mod 256 → AC

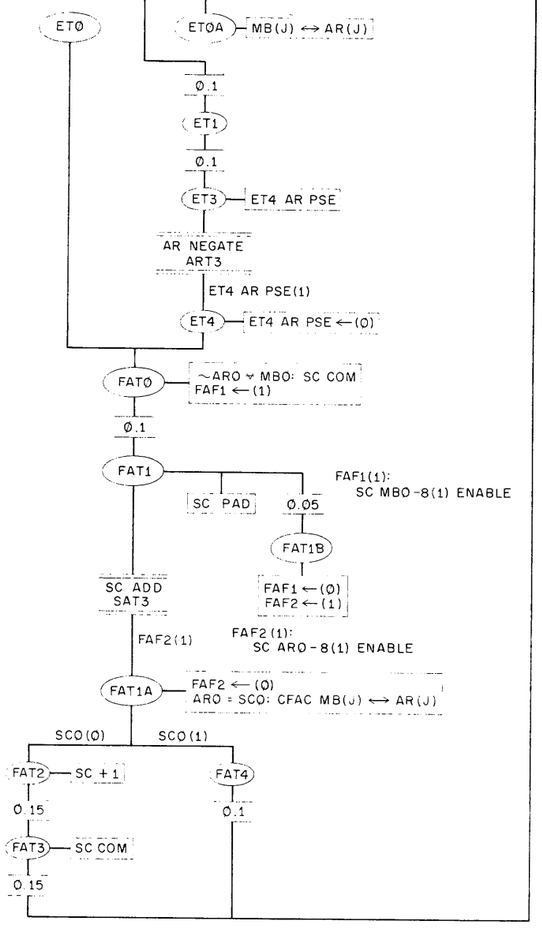
Initial Registers: AR = AC; MB = C(E); MQ = Ø
Initial Gate: ET4 INH



FAD

140-147

Initial Gates: FC(E); ET4 INH

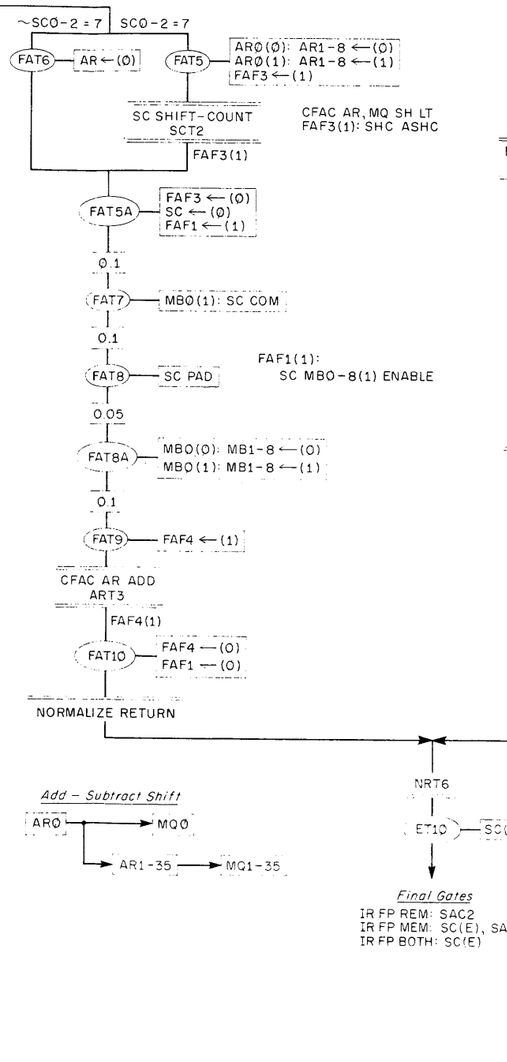


FSB

150-157

Initial Gates: FC(E); ET4 INH; AR SBR; ET5 INH

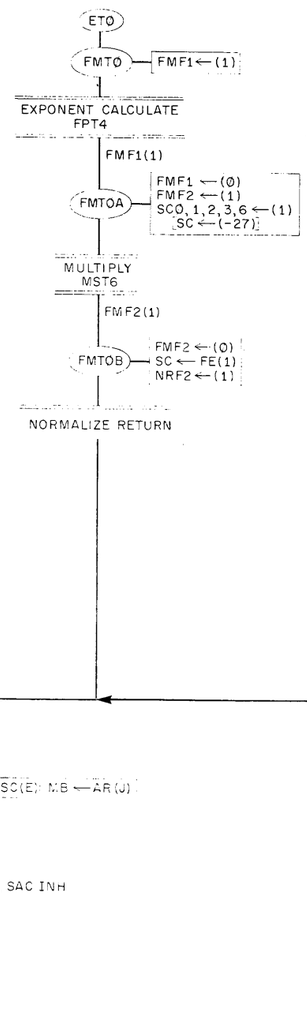
FAD / FSB (continued)



FMP

160-167

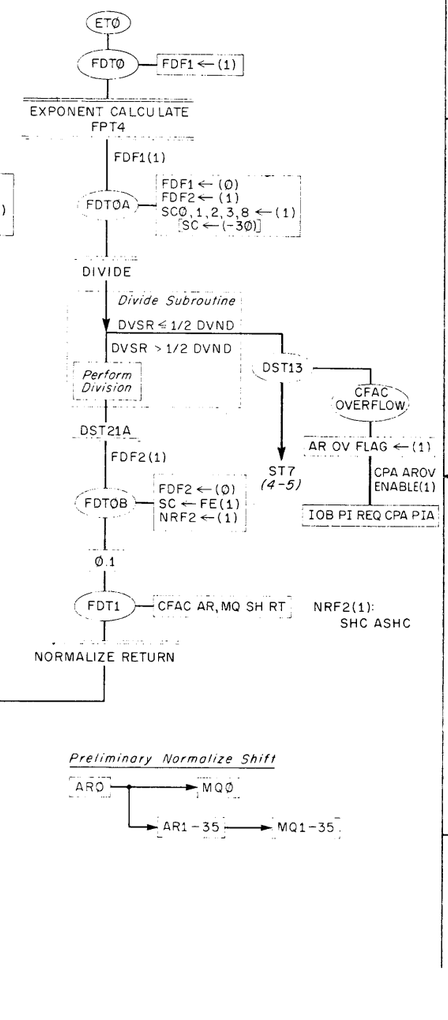
Initial Registers: AR = AC; MB = C(E); MQ = Ø
Initial Gates: FC(E); ET4 INH



FDV

170-177

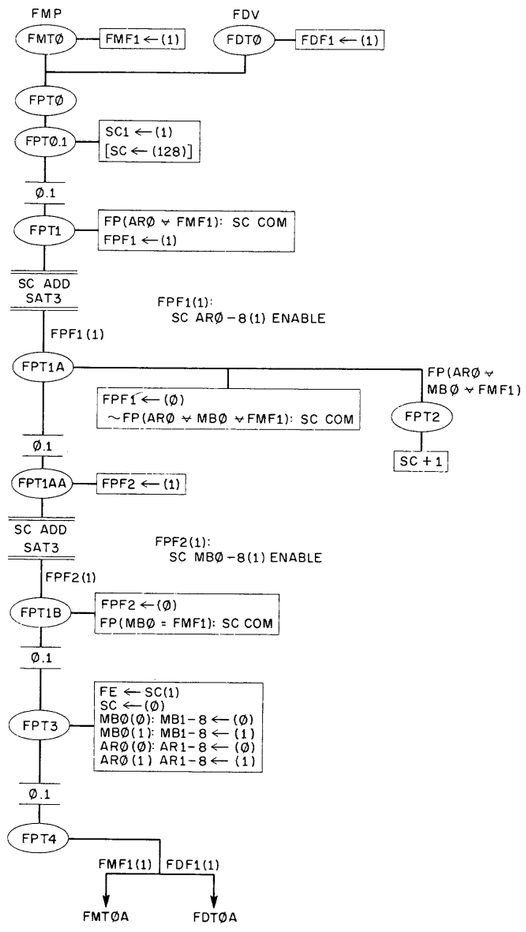
Initial Gates: FC(E); ET4 INH



FMP / FDV (continued)

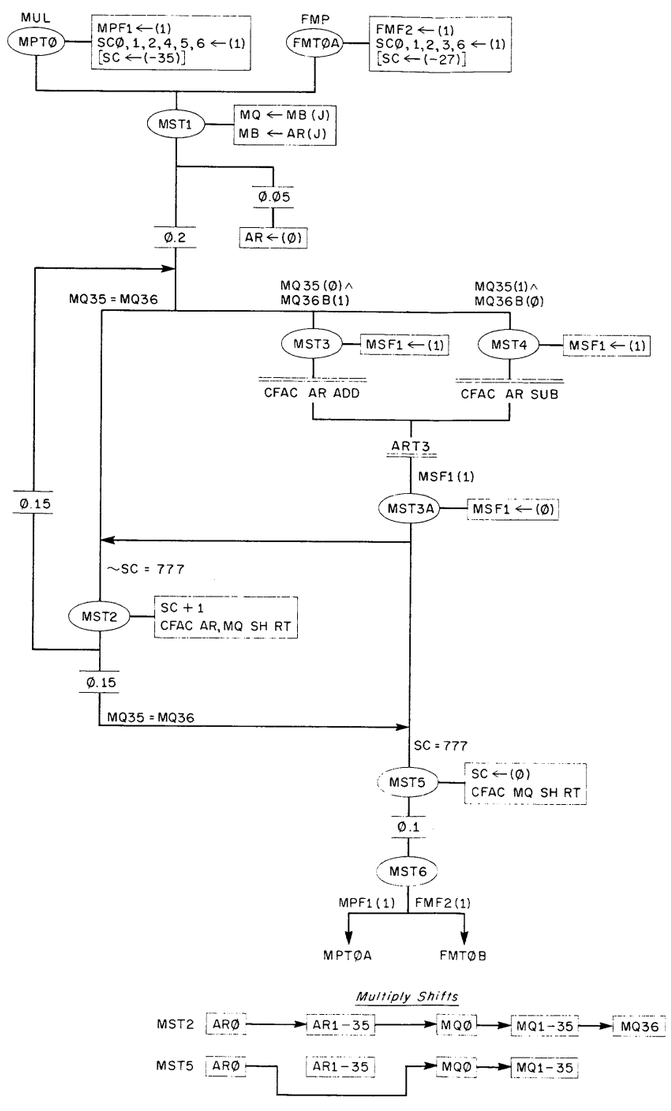
CODE FD | DRWG. NO D-166-0-EMNF | REV LTR

EXPONENT CALCULATE



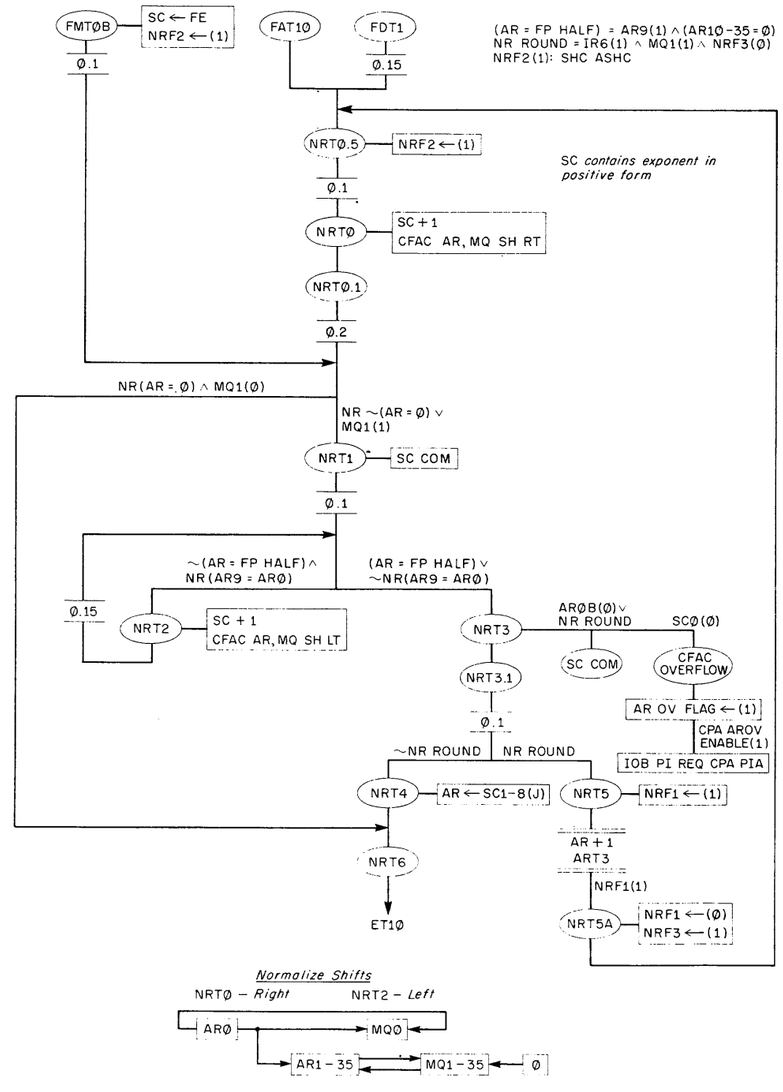
MULTIPLY

MS MUL = MPF1(1) ∨ FMF2(1)



Multiply Shifts

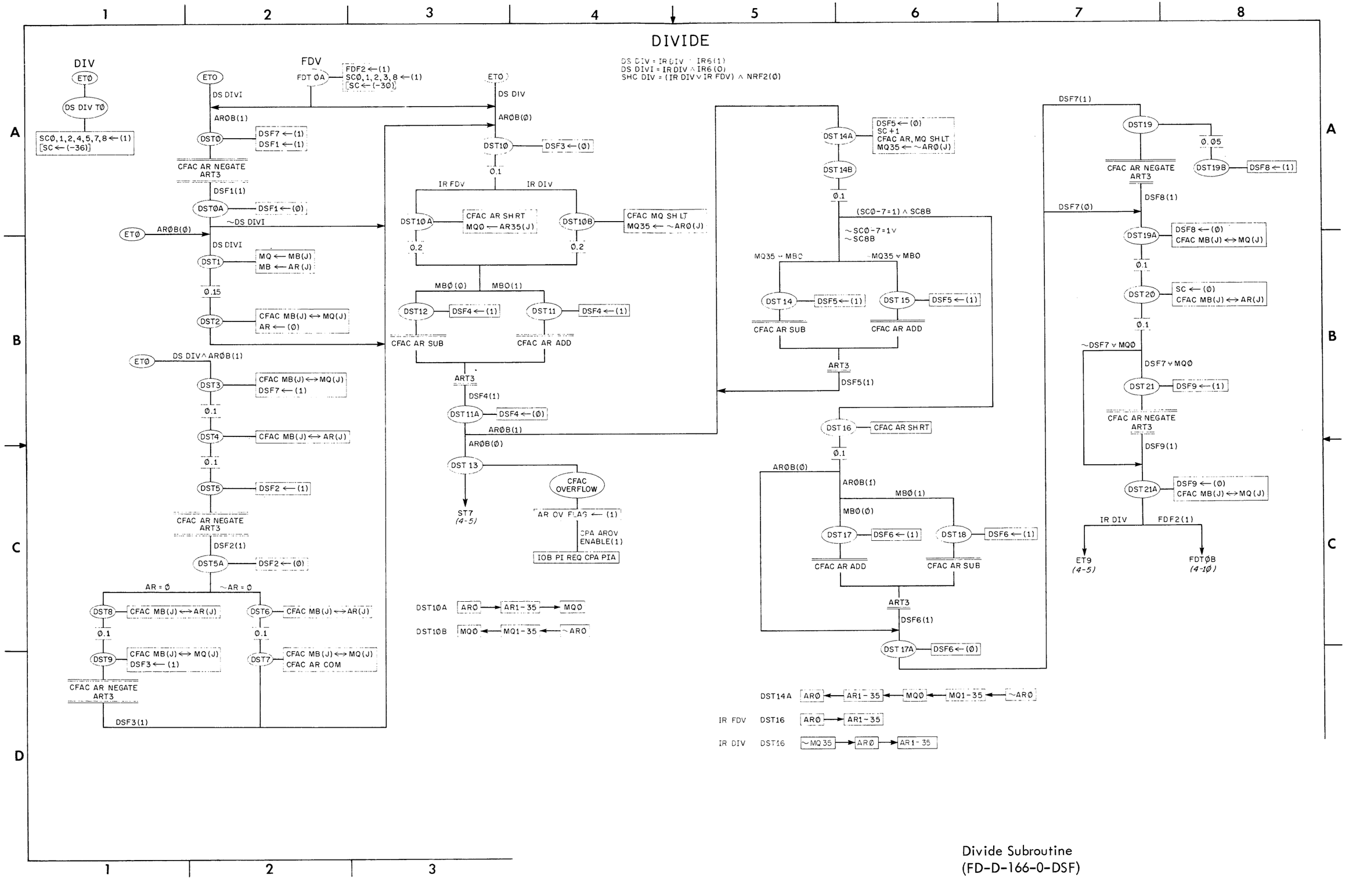
NORMALIZE RETURN



Normalize Shifts

Exponent Calculate, Multiply and Normalize Return Subroutines
(FD-D-166-0-EMNF)

DIVIDE



Divide Subroutine
(FD-D-166-0-DSF)

IN-OUT TRANSFER

IR = 111 XXX XXX XYY Y

IR IOT A = IR Ø-2 (111)

IR IOT = IR IOT A ∧ ~EX IR UUO

YYY *specifies instruction*

IR IOT ∧ IR1Ø-12 {
- ØØØ = IOT BLKI
- ØØ1 = IOT DATAI
- Ø1Ø = IOT BLKO
- Ø11 = IOT DATAO
- 1ØØ = IOT CONO
- 1Ø1 = IOT CONI
- 11Ø = IOT CONSZ
- 111 = IOT CONSO

XXX XXX X *specifies device.* IR3-9 *outputs thru bus drivers to* IOS3-9 *lines on* IO *bus. Each* IO *control unit contains a decoding net that allows proper code to select device by gating in command signals from bus*

Instruction Action

BLKI	C(E) + 1ØØØØØ1 → E, DEVICE BUFFER → C(C(ERT))
DATAI	DEVICE BUFFER → E
BLKO	C(E) + 1ØØØØØ1 → E, C(C(ERT)) → DEVICE BUFFER
DATAO	C(E) → DEVICE BUFFER
CONO	E → DEVICE CONTROL
CONI	STATUS → E
CONSZ	(STATUS ∧ E) = Ø: PC + 1
CONSO	(STATUS ∧ E) ≠ Ø: PC + 1

IOT BLK = IOT BLKI ∨ IOT BLKO
IOT OUTGOING = IOT DATAO ∨ IOT CONO
IOT STATUS = IOT CONI ∨ IOT CONSZ ∨ IOT CONSO
IOT DATAI ∨ O = IOT DATAI ∨ IOT DATAO
PI BLK RST = IOT DATAI ∨ O ∧ PI OV(Ø)
PI BLK RST ∧ PI CYC B(1): PI HOLD, PI RST
IOT CONSZ ∨ IOT CONSO: E LONG

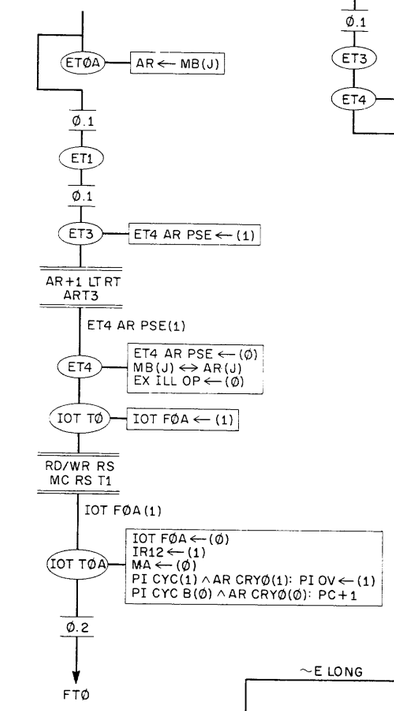
BLK

Initial Registers

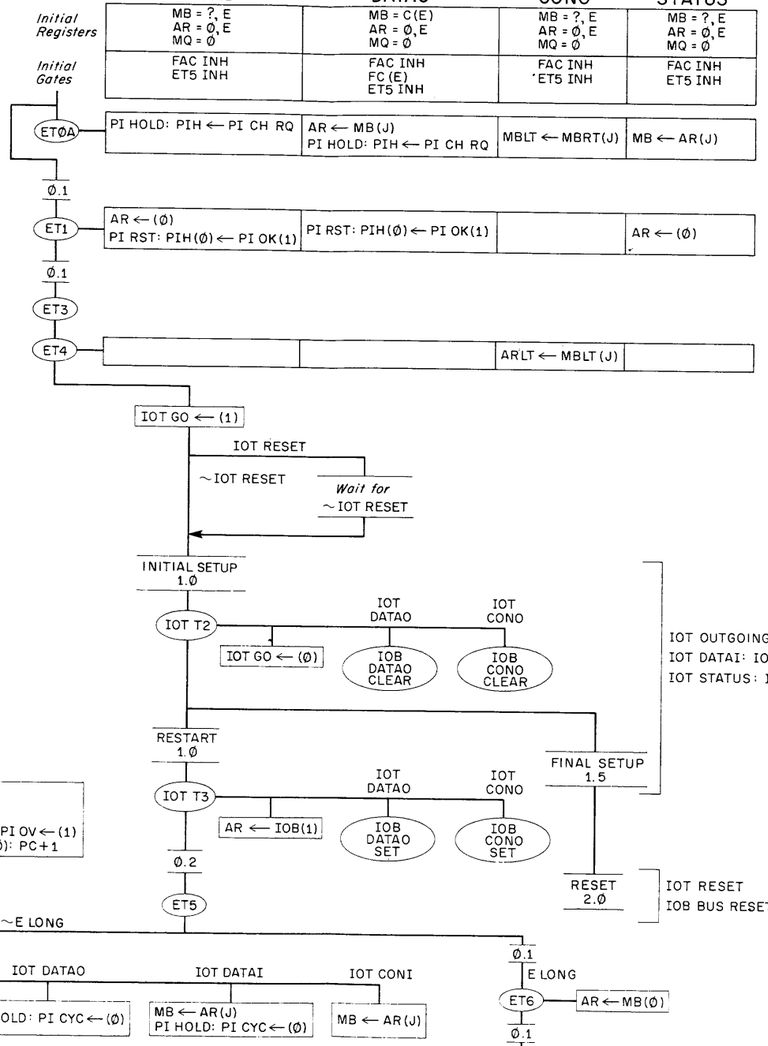
MB = C(E)
AR = Ø, E
MQ = Ø

Initial Gates

FAC INH
FC(E) PSE
PC + 1 INH
ET4 INH
AR SBR
ET5 INH



Return to fetch and perform data instruction with C(MBRT) *as* E



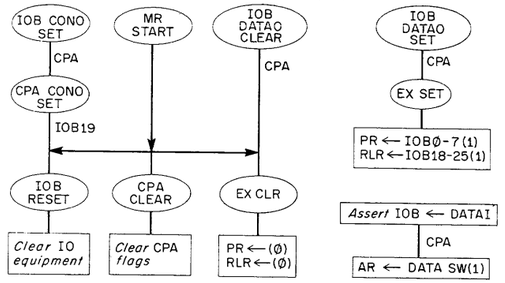
Final Gates
IOT DATAI ∨ IOT CONI: SC(E)
SAC INH

IOT OUTGOING: IOB ← AR(1)
IOT DATAI: IOB ← DATAI
IOT STATUS: IOB ← STATUS

CPA

ØØØ

CPA = IOS3-9 (ØØØ ØØØ Ø)



IOB	CPA CONO SET	CPA STATUS
18	CPA PDL OV ← (Ø)	
19	IOB RESET	CPA PDL OV(1)
2Ø	CPA IOT USER ← (1)	CPA IOT USER(1)
21	CPA IOT USER ← (Ø)	EX USER B(1)
22	CPA ILLEG OP ← (Ø)	CPA ILLEG OP(1)
23	CPA NON EXIST MEM ← (Ø)	CPA NON EXIST MEM(1)
24	CPA CLOCK ENABLE ← (Ø)	
25	CPA CLOCK ENABLE ← (1)	CPA CLOCK ENABLE(1)
26	CPA CLOCK FLAG ← (Ø)	CPA CLOCK FLAG(1)
27	CPA PC CHG ENABLE ← (Ø)	
28	CPA PC CHG ENABLE ← (1)	CPA PC CHG ENABLE(1)
29	AR PC CHG FLAG ← (Ø)	AR PC CHG FLAG B(1)
3Ø	CPA AROV ENABLE ← (Ø)	
31	CPA AROV ENABLE ← (1)	CPA AROV ENABLE(1)
32	AR OV FLAG ← (Ø)	AR OV FLAG B(1)
33-35	CPA PIA ← IOB33-35(1)	CPA PIA B(1)

6Ø times per second from power control: CPA CLOCK FLAG ← (1)
MC ILLEG ADDRESS: CPA ILLEG OP ← (1)
MC NON EXIST MEM: CPA NON EXIST MEM ← (1)
ET1Ø ∧ [MB(J) ↔ AR(J) (ET1Ø)] ∧ AR CRYØ(1): CPA PDL OV ← (1)

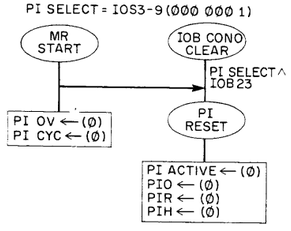
IOB PI REQ CPA PIA

CPA ILLEG OP(1)
CPA NON EXIST MEM(1)
CPA PDL OV(1)
CPA CLOCK ENABLE(1) ∧ CPA CLOCK FLAG(1)
CPA PC CHG ENABLE(1) ∧ AR PC CHG FLAG B(1)
CPA AROV ENABLE(1) ∧ AR OV FLAG(1)

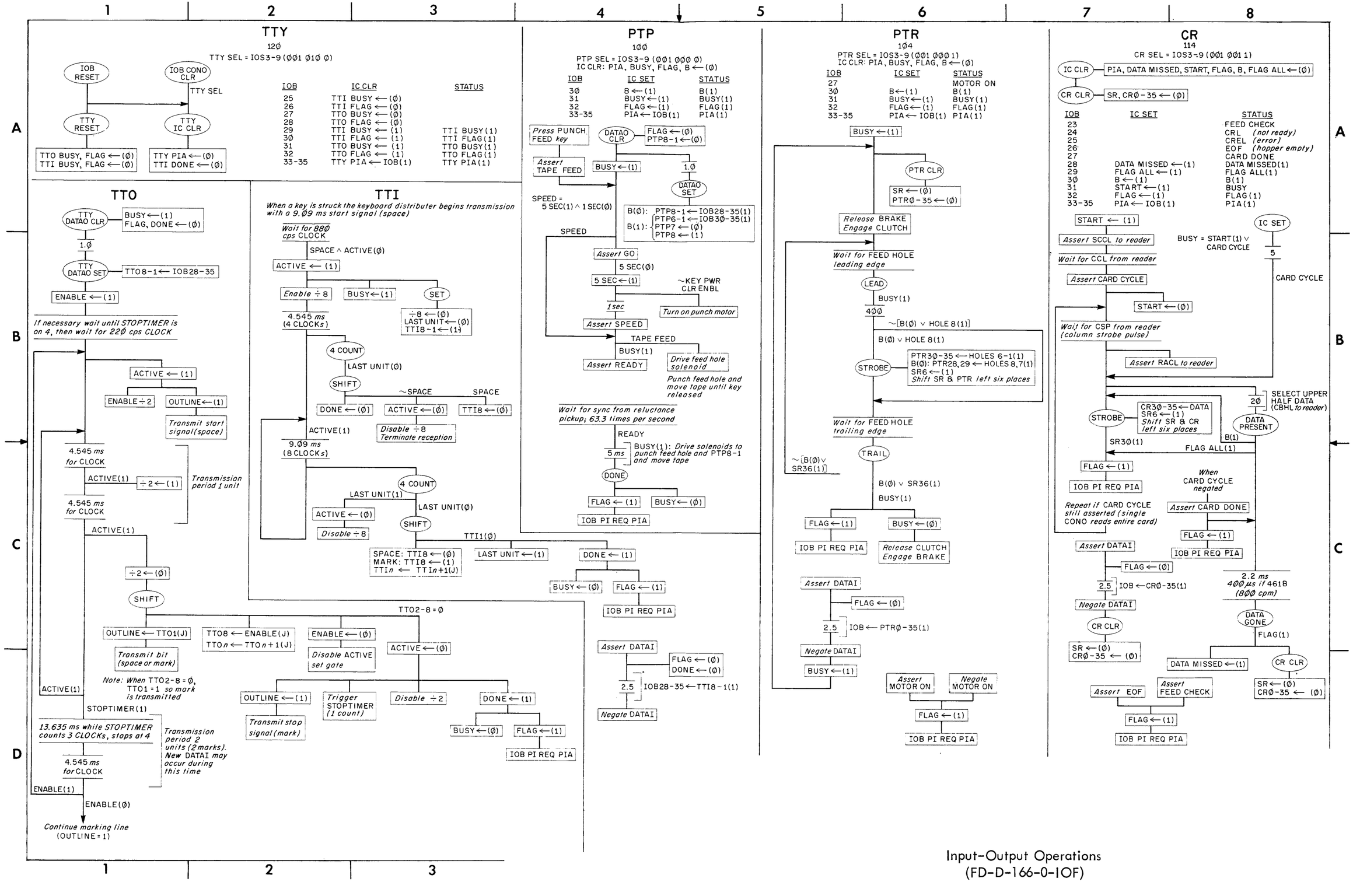
PI

ØØ4

PI SELECT = IOS3-9 (ØØØ ØØØ 1)

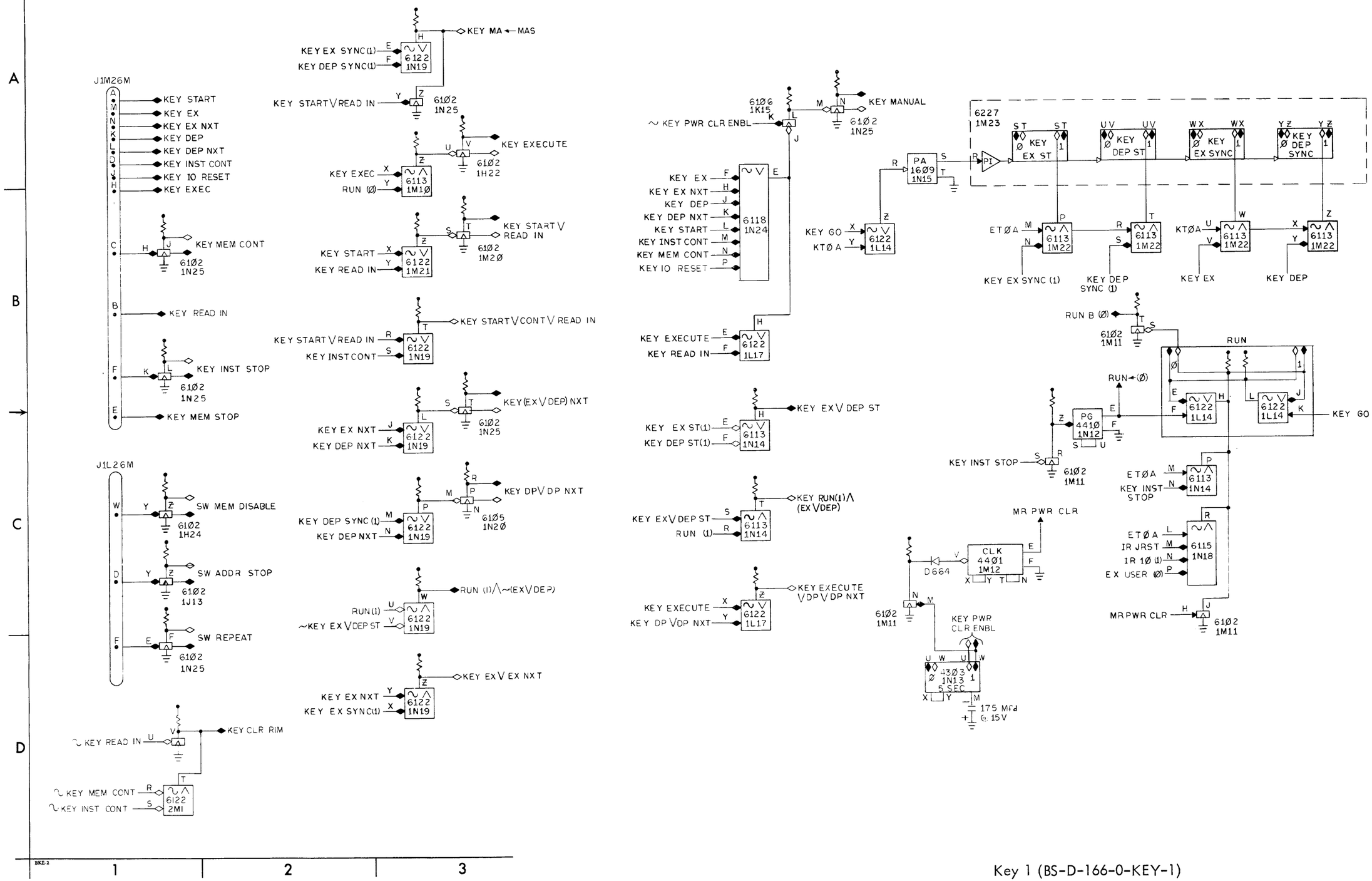


IOB	PI CONO SET	PI STATUS
24	PIR ← IOB29-35(1)	
25	PIO ← IOB29-35(1)	
26	PIO(Ø) ← IOB29-35(1)	
27	PI ACTIVE ← (Ø)	
28	PI ACTIVE ← (1)	
29-35		PI ACTIVE(1)
		PIO1-7(1)



Input-Output Operations
(FD-D-166-0-1OF)

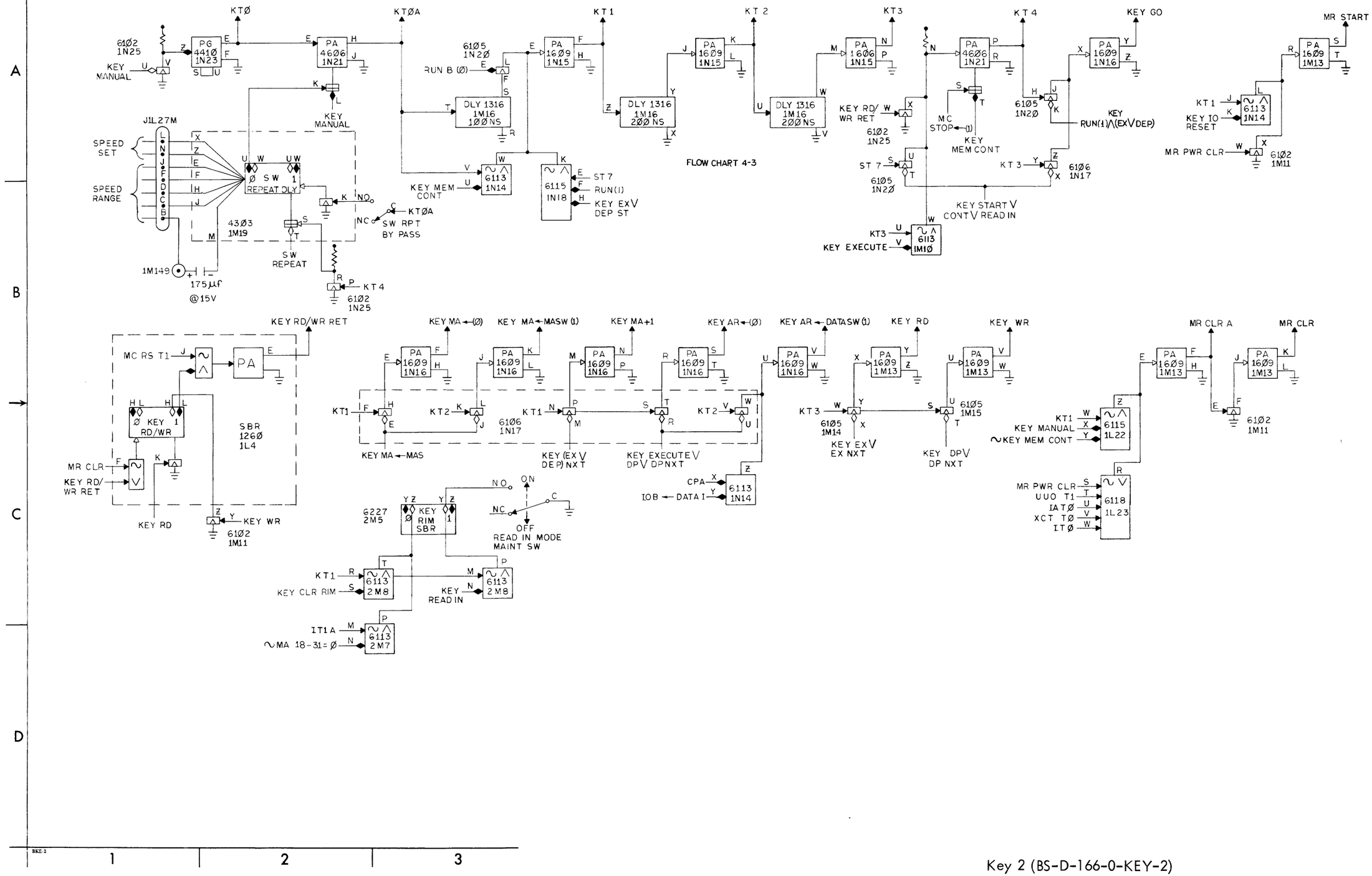
Key 1 (BS-D-166-0-KEY-1)



BKE-2

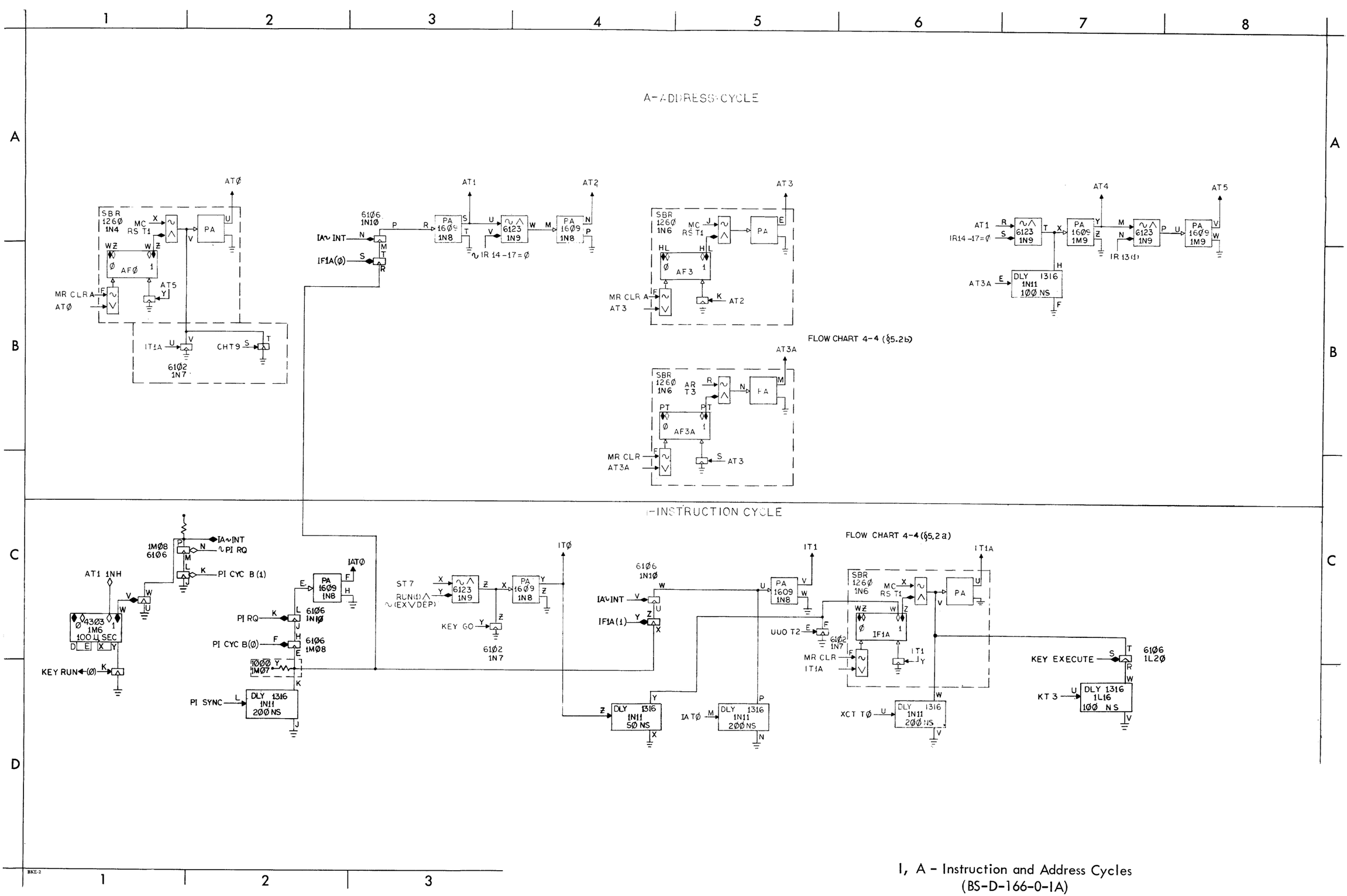
Key 1 (BS-D-166-0-KEY-1)

Key 2 (BS-D-166-0-KEY-2)



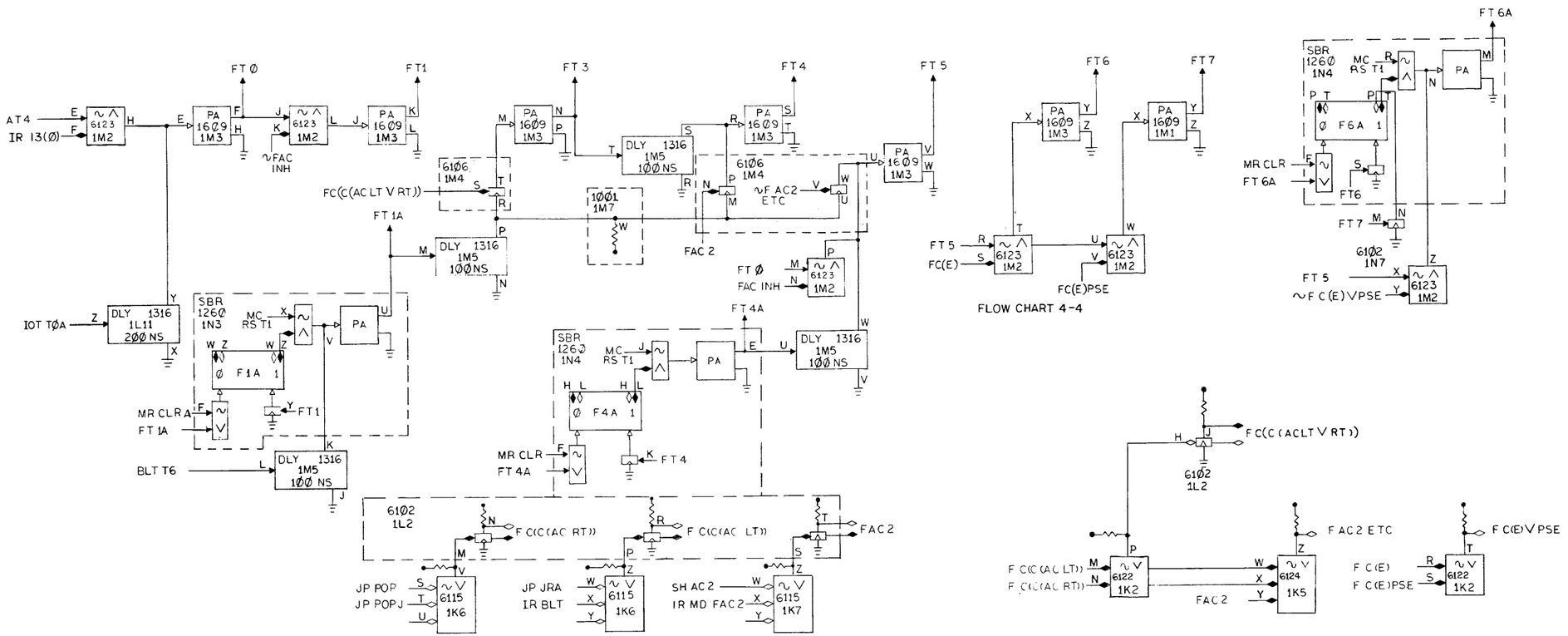
Key 2 (BS-D-166-0-KEY-2)

I, A - Instruction and Address Cycles
(BS-D-166-0-1A)



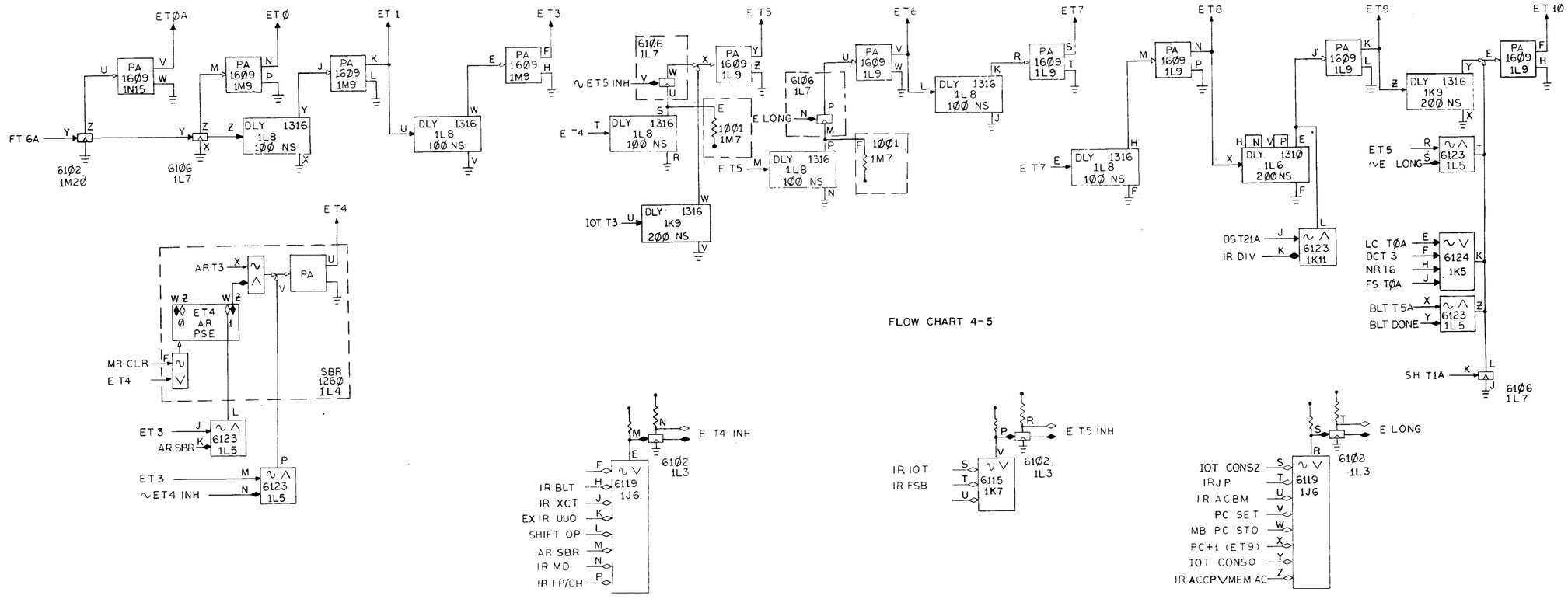
I, A - Instruction and Address Cycles
(BS-D-166-0-1A)

F - Fetch Cycle (BS-D-166-0-F)



F - Fetch Cycle (BS-D-166-0-F)

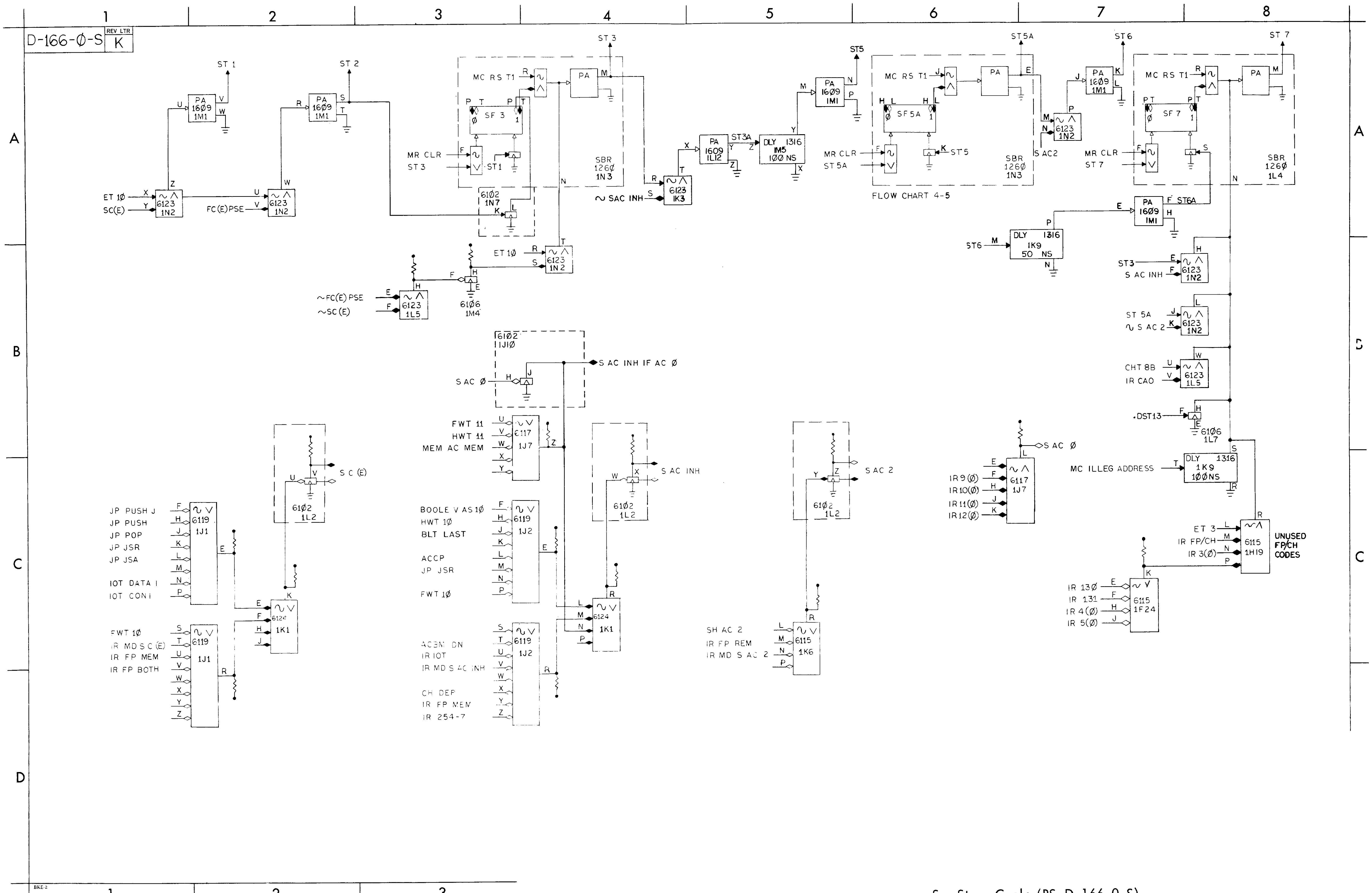
E - Execute Cycle (BS-D-166-0-E)



FLOW CHART 4-5

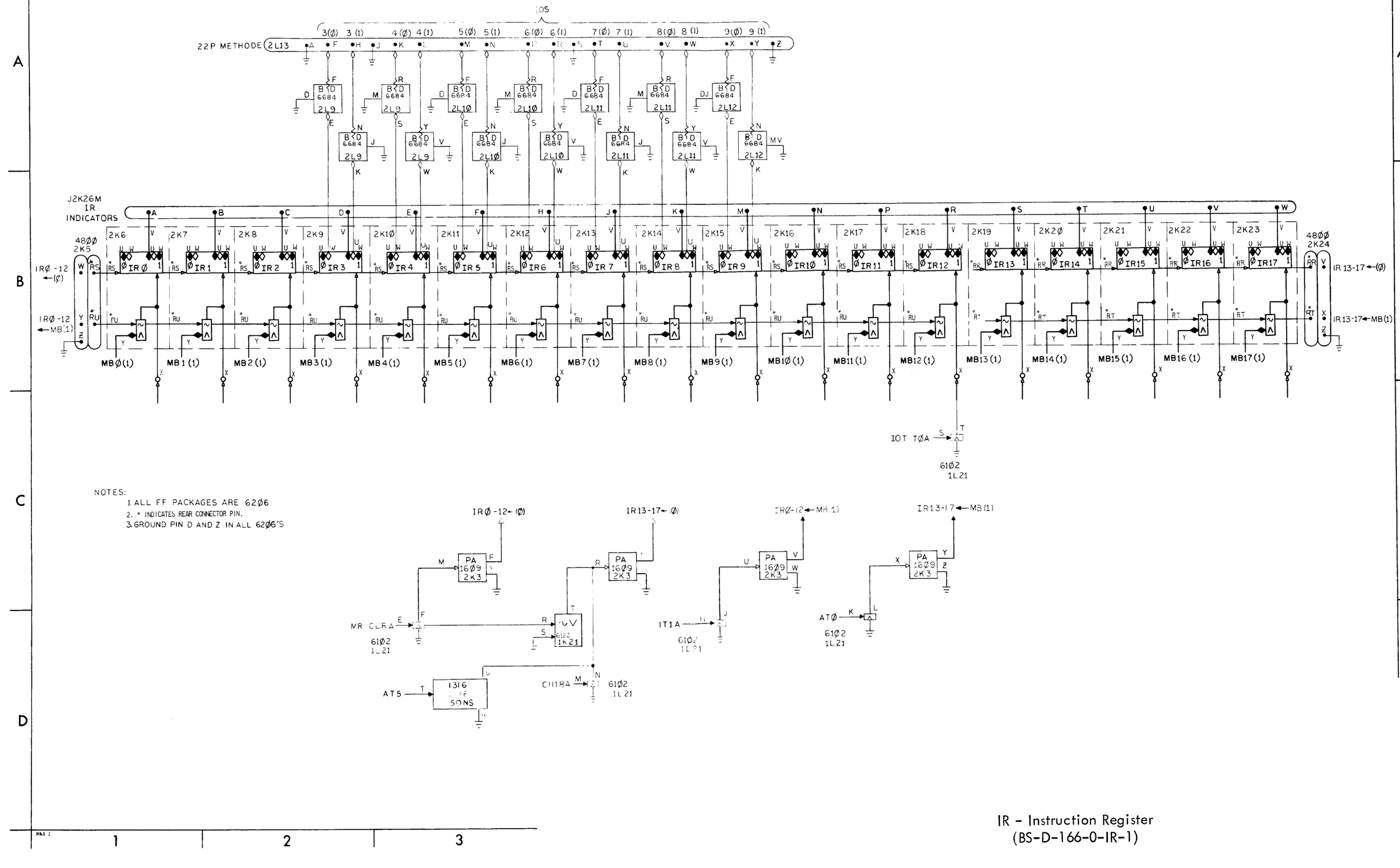
E - Execute Cycle (BS-D-166-0-E)

S - Store Cycle (BS-D-166-0-S)



S - Store Cycle (BS-D-166-0-S)

IR - Instruction Register
(BS-D-166-0-IR-1)



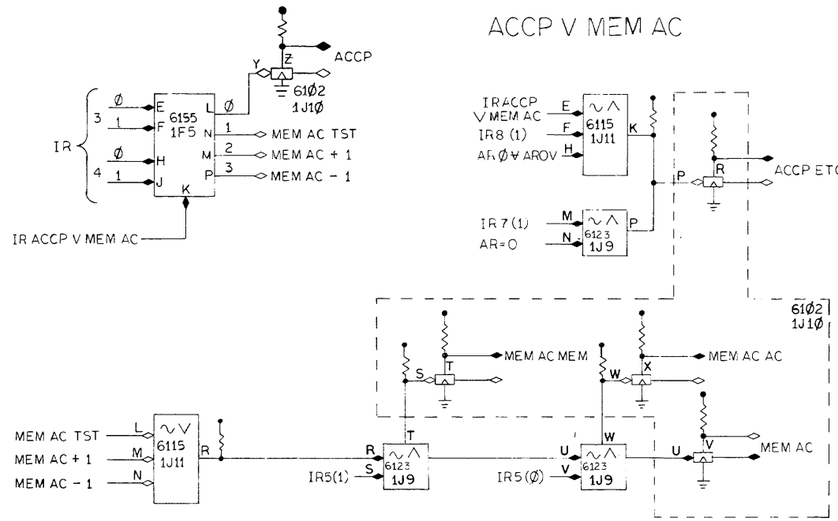
NOTES:

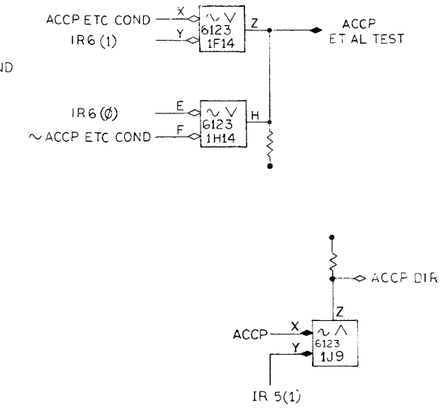
1. ALL FF PACKAGES ARE 62Ø6
2. * INDICATES REAR CONNECTOR PIN.
3. GROUND PIN D AND Z IN ALL 62Ø6'S

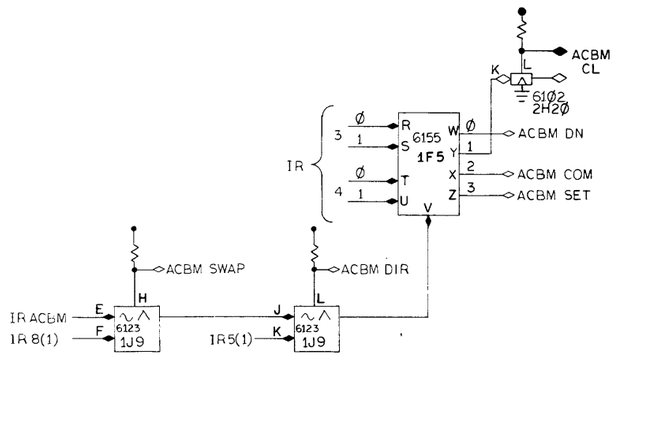
IR - Instruction Register
(BS-D-166-0-IR-1)

IR Decoding (BS-D-166-0-IR-2)

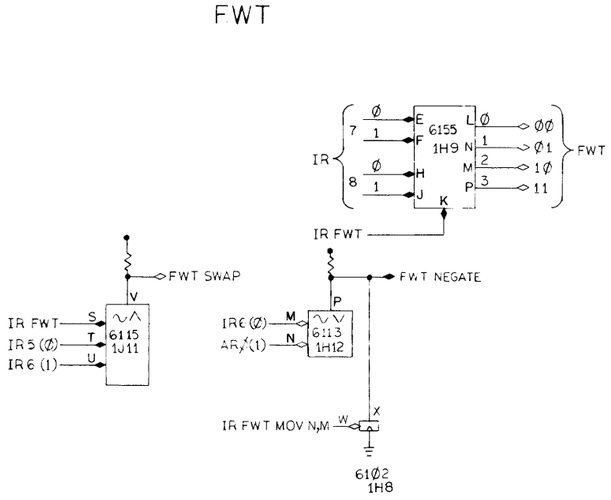
IR Decoding (BS-D-166-0-IR-2)

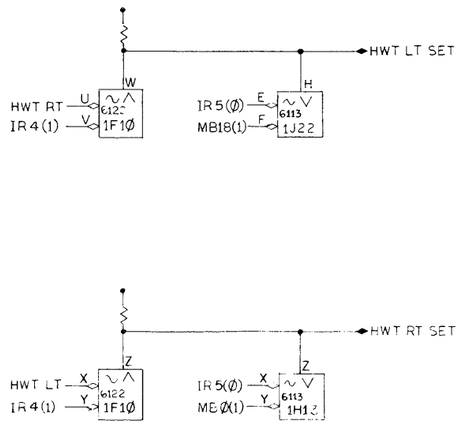


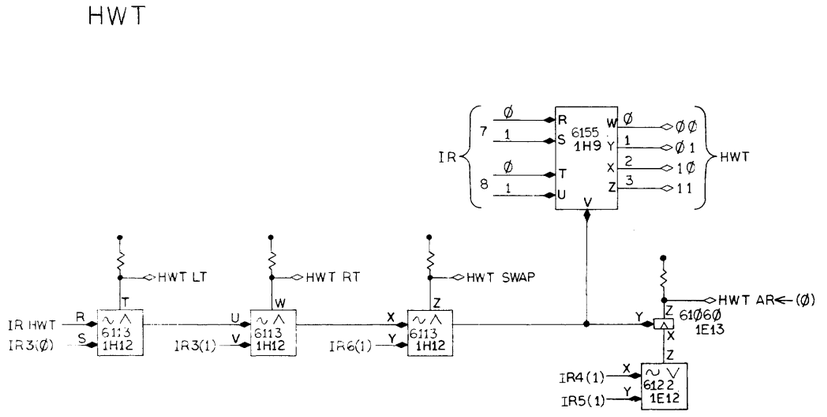




FLOW CHART 4-6



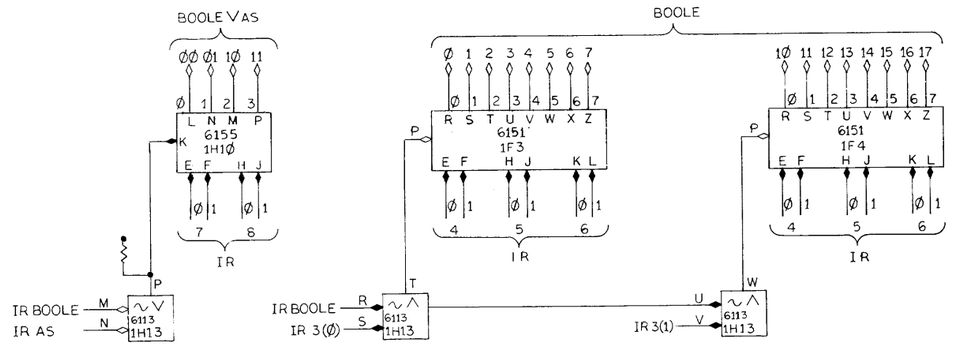




ACCP, MEMAC, ACBM, FWT, HWT
(BS-D-166-0-INS-1)

BOOLE

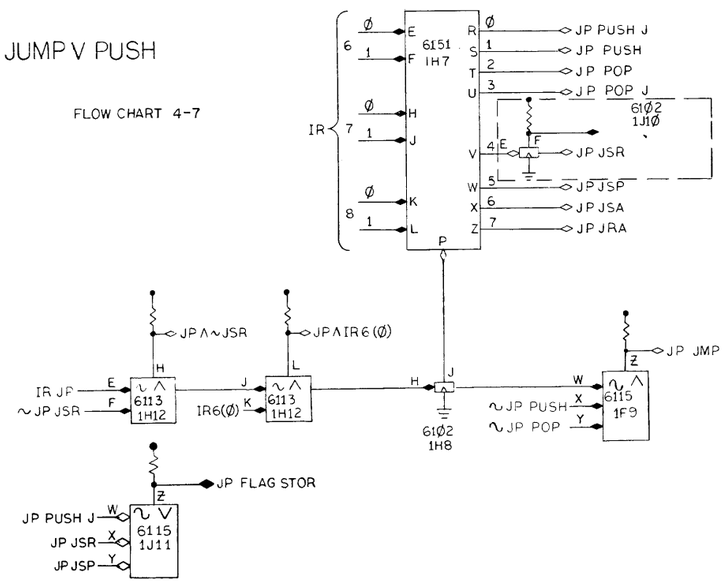
			RESULT BIT FOR OPERAND BIT CONFIGURATION				ET∅ T1		ET1 T2			ET4 T3
			MEM AC	MEM AC	MEM AC	MEM AC	AR CLR	AR COM	AR←MB(∅)	AR←MB(1)	AR←MB ⊽	AR COM
			∅ ∅	∅ 1	1 ∅	1 1						
SETZ	∅	∅∅	∅	∅	∅	∅	✓					
AND	MEM ∧ AC	∅1	∅	∅	∅	1			✓			
ANDCA	MEM ∧ ∼ AC	∅2	∅	∅	1	∅		✓	✓			
SETM	MEM	∅3	∅	∅	1	1	✓			✓		
ANDCM	∼ MEM ∧ AC	∅4	∅	1	∅	∅		✓		✓		✓
SETA	AC	∅5	∅	1	∅	1						
XOR	MEM ⊽ AC	∅6	∅	1	1	∅				✓	✓	
IOR	MEM ∨ AC	∅7	∅	1	1	1				✓		
ANDCB	∼ MEM ∧ ∼ AC	1∅	1	∅	∅	∅				✓	✓	✓
EQV	MEM ≡ AC	11	1	∅	∅	1						✓
SETCA	∼ AC	12	1	∅	1	∅		✓				
ORCA	MEM ∨ ∼ AC	13	1	∅	1	1		✓		✓		✓
SETCM	∼ MEM	14	1	1	∅	∅	✓				✓	✓
ORCM	∼ MEM ∨ AC	15	1	1	∅	1		✓	✓			✓
ORCB	∼ MEM ∨ ∼ AC	16	1	1	1	∅			✓			✓
SETO	1	17	1	1	1	1	✓					✓



FLOW CHART 4-8

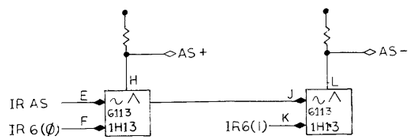
JUMP V PUSH

FLOW CHART 4-7



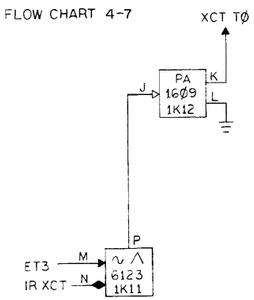
AS

FLOW CHART 4-8

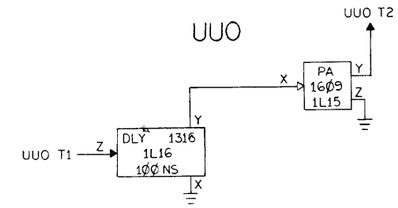


XCT

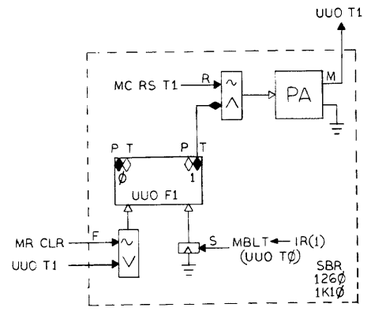
FLOW CHART 4-7



UUO

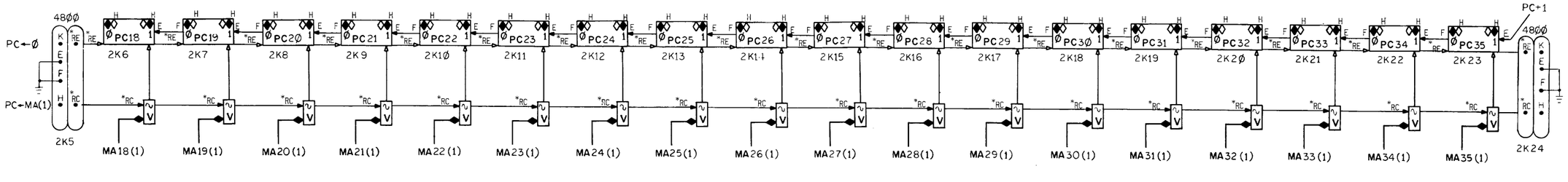


FLOW CHART 4-7



BOOLE, JP, AS, XCT, UUO
(BS-D-166-0-INS-2)

PC - Program Counter (BS-D-166-0-PC-2)

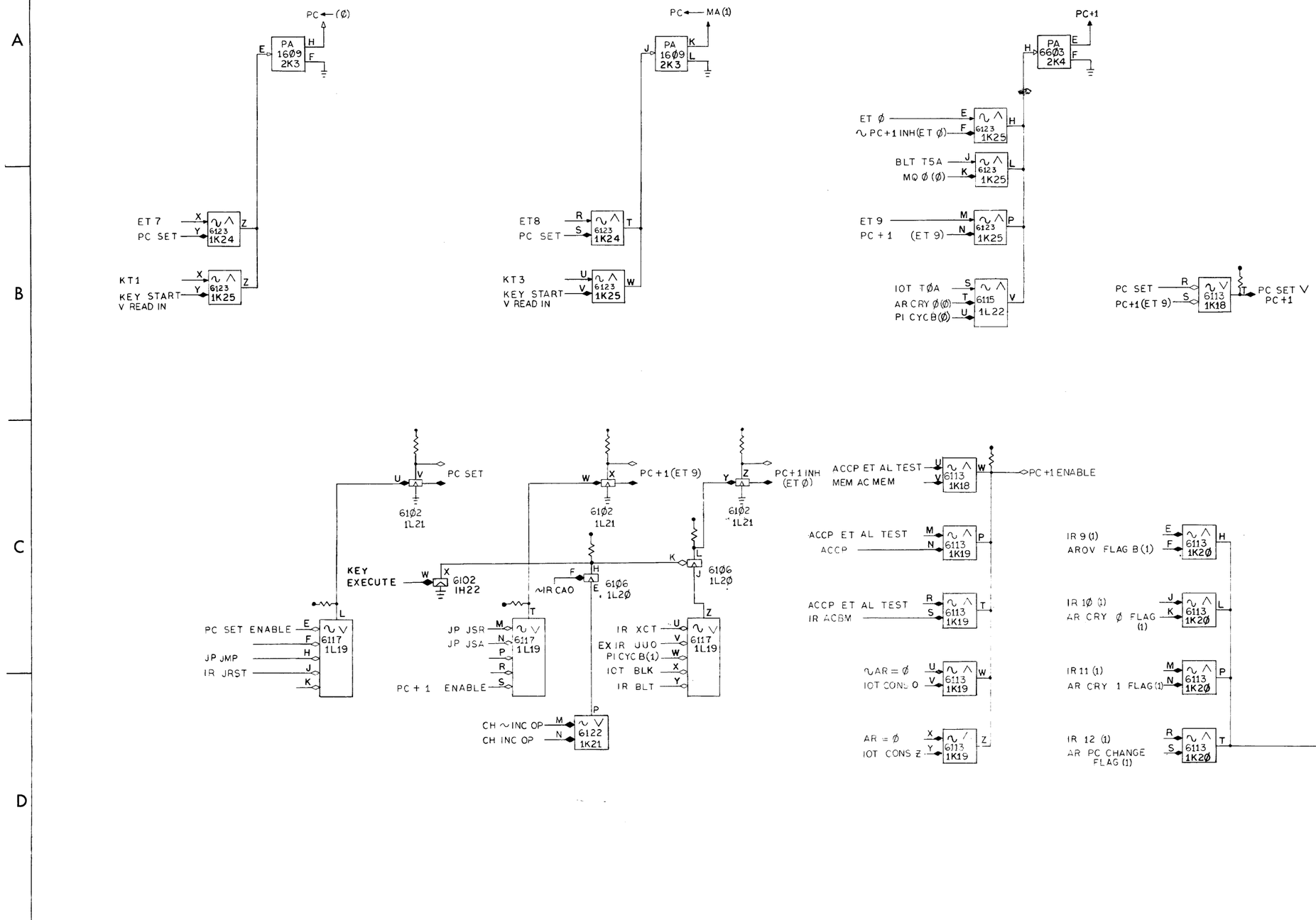


NOTES.

1. ALL FF PACKAGES ARE 62Ø6
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D AND Z IN ALL 62Ø6'S

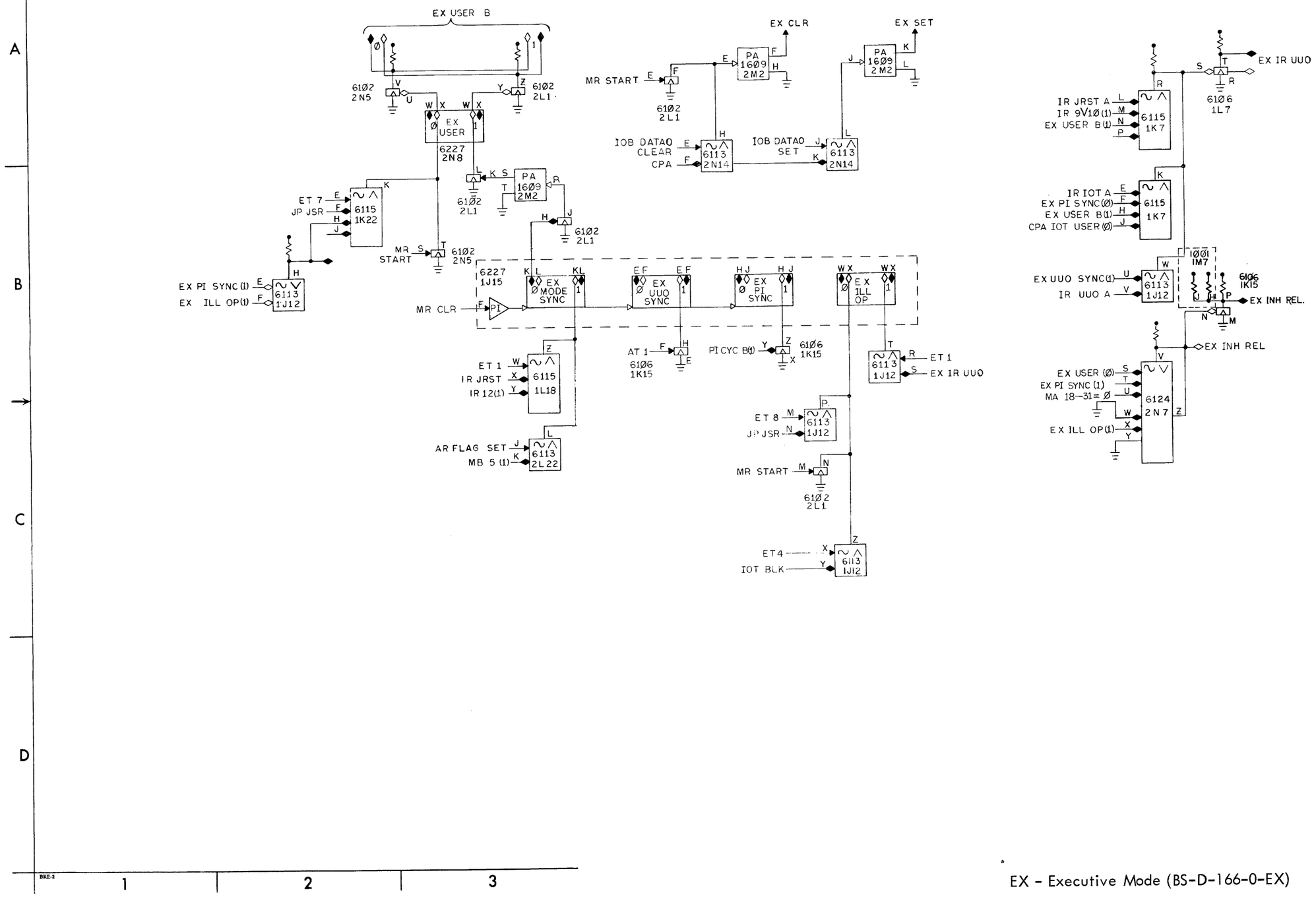
PC - Program Counter (BS-D-166-0-PC-2)

PC Control (BS-D-166-0-PC-1)



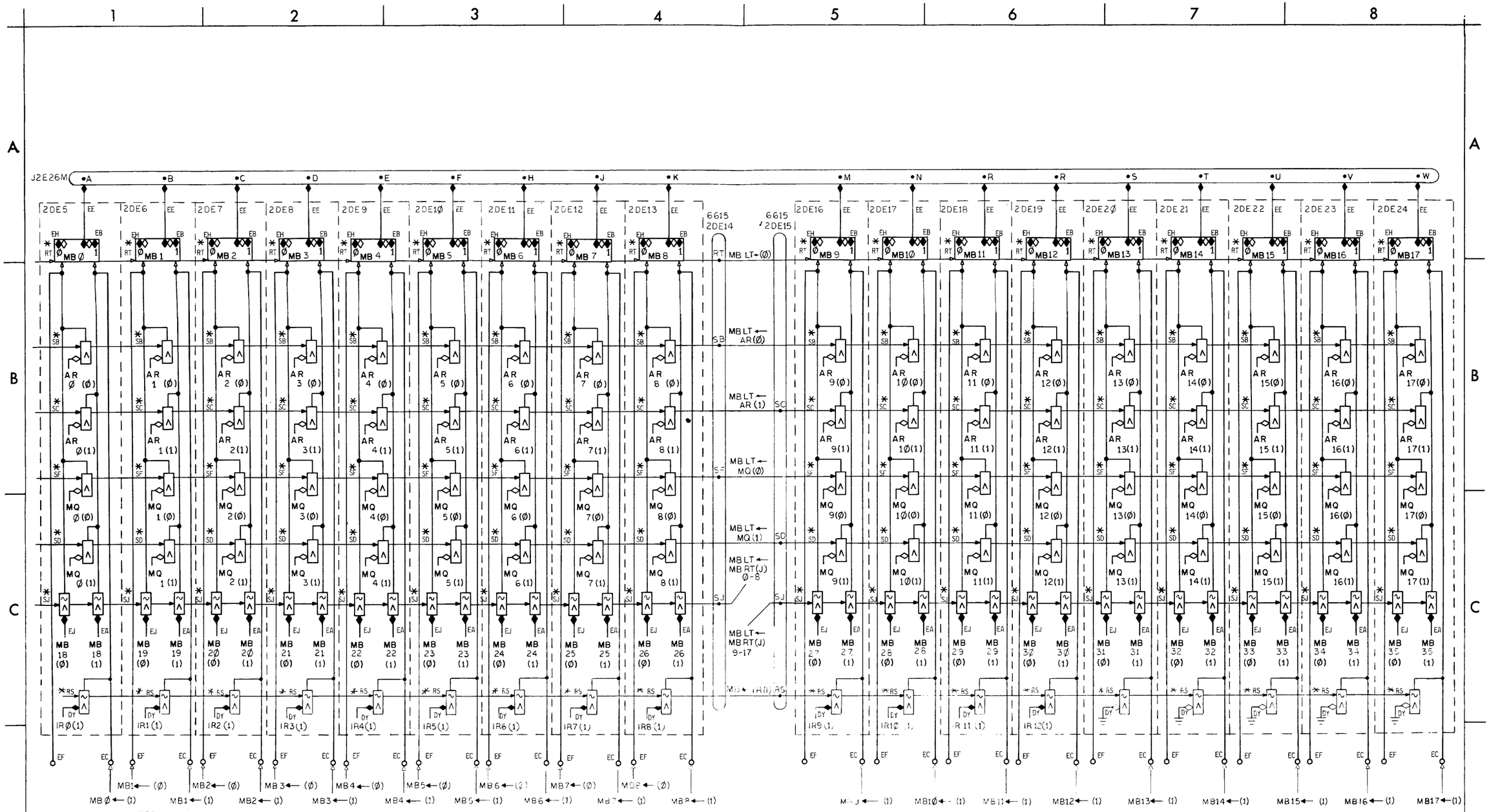
PC Control (BS-D-166-0-PC-1)

EX - Executive Mode (BS-D-166-0-EX)

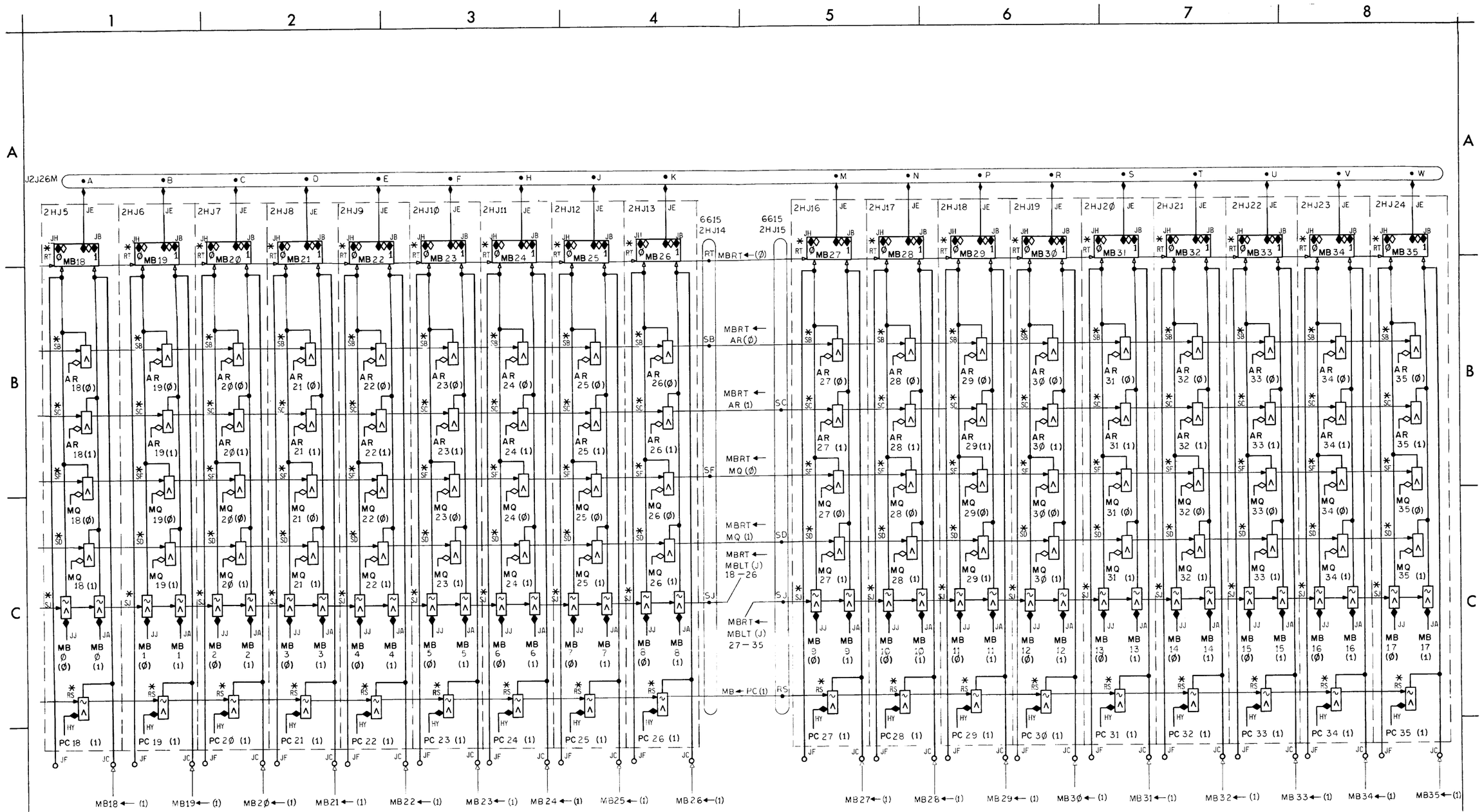


EX - Executive Mode (BS-D-166-0-EX)

MBLT Register 0-17 (BS-D-166-0-MB-2)



MBLT Register 0-17 (BS-D-166-0-MB-2)

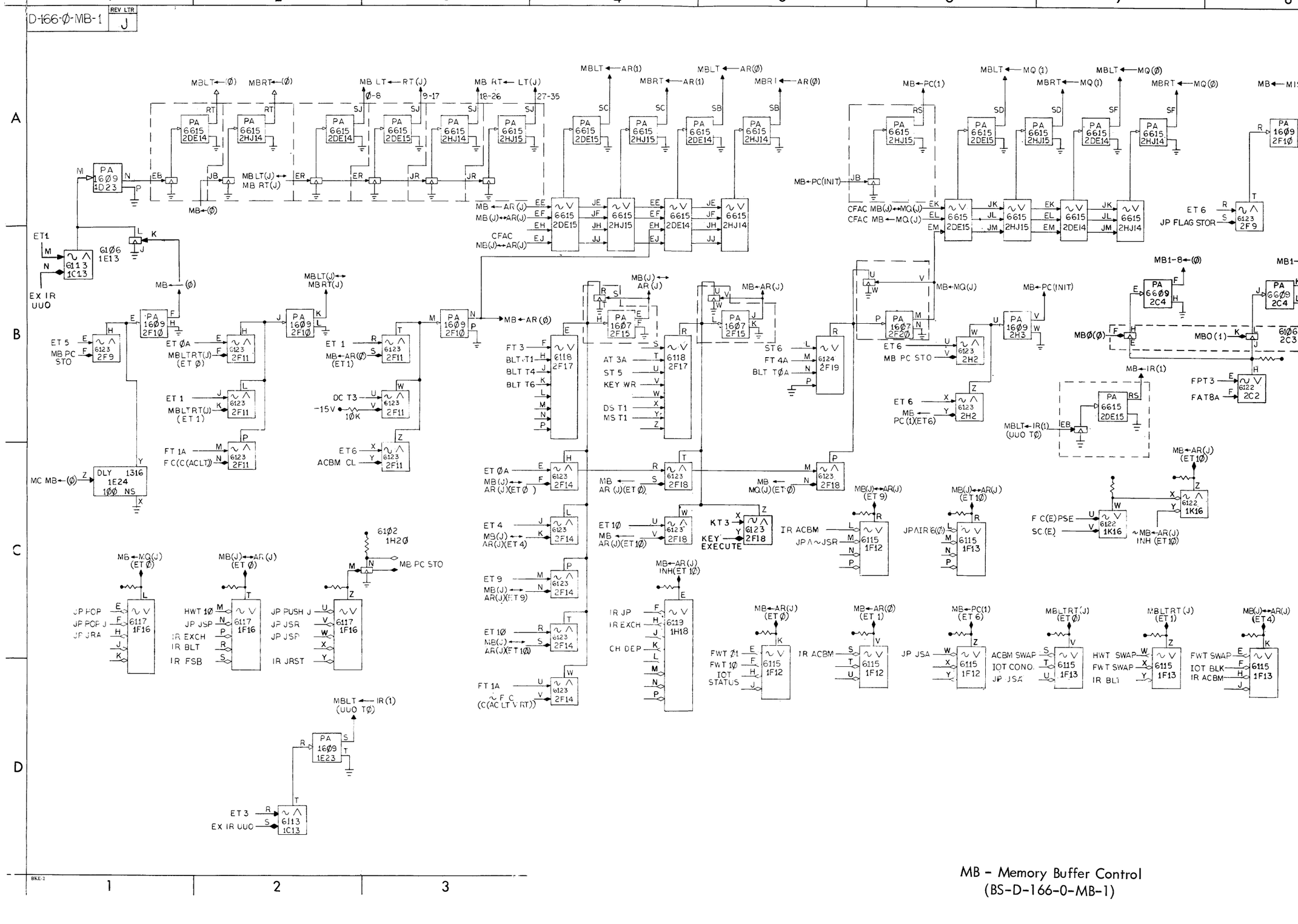


NOTES
1. ALL FF PACKAGES ARE 62Ø5
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D, P AND Z IN ALL 62Ø5's AND 6615's.

MBRT Register 18-35 (BS-D-166-0-MB-3)

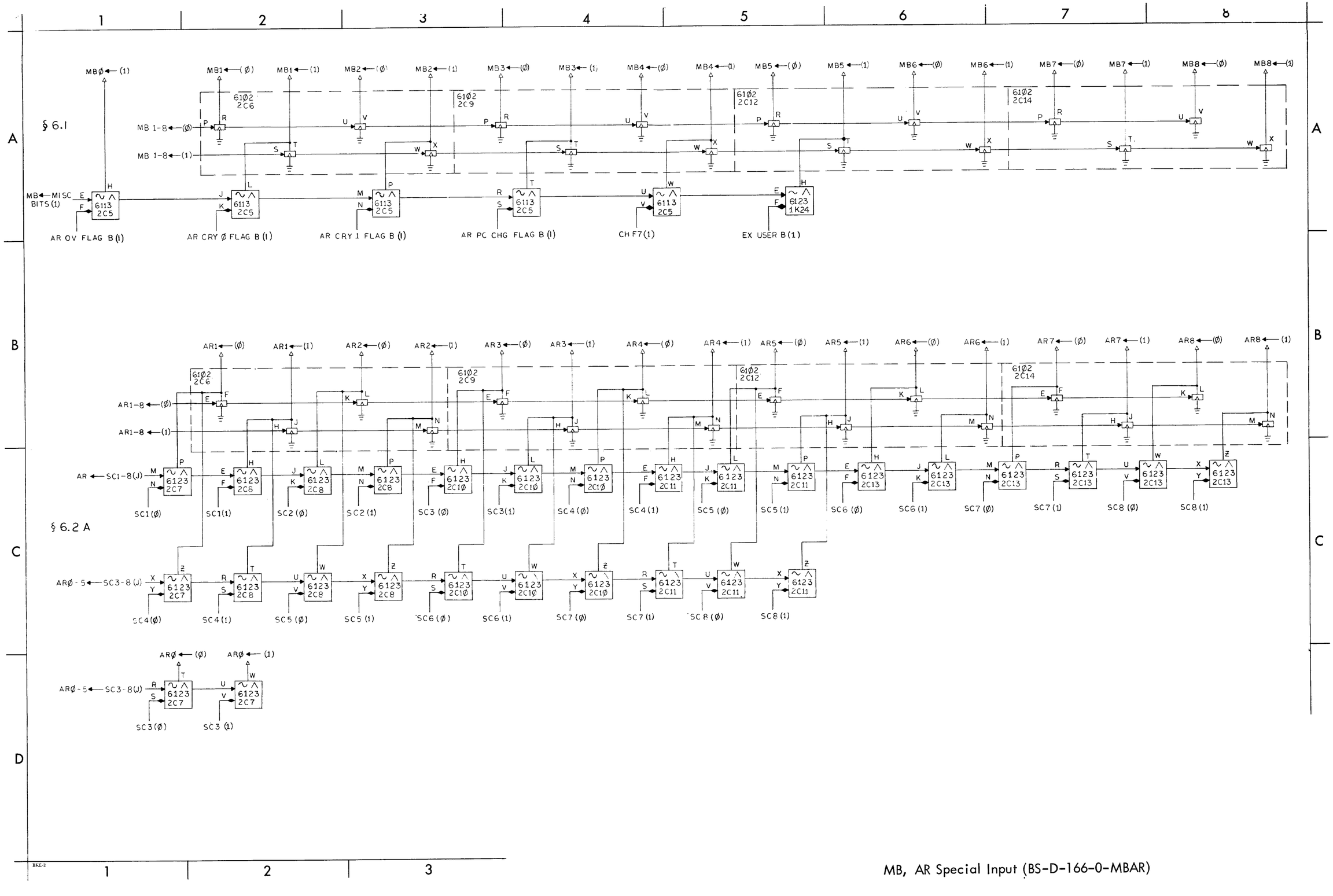
MB - Memory Buffer Control
(BS-D-166-0-MB-1)

D-166-Ø-MB-1 | REV LTR J

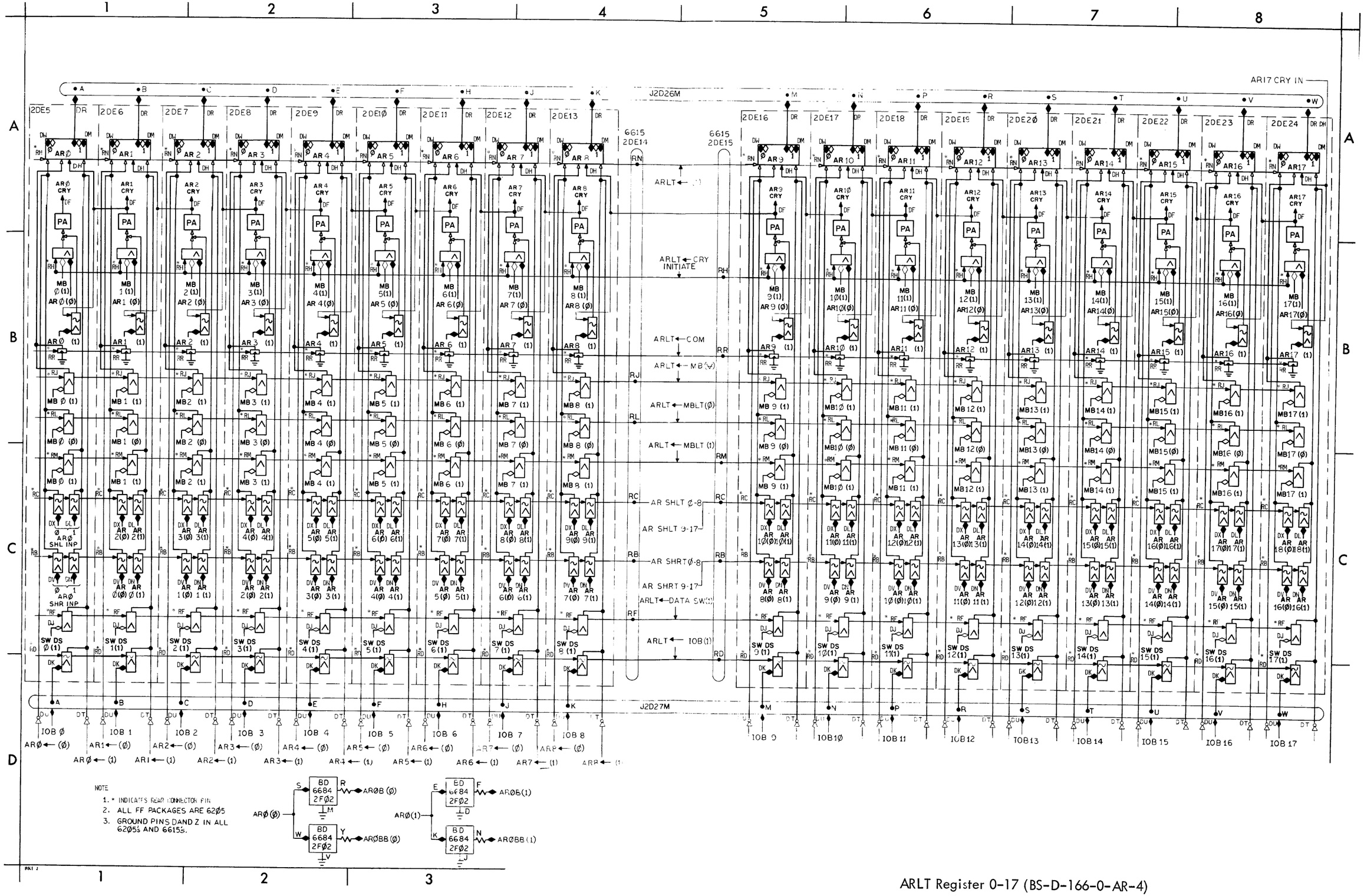


MB - Memory Buffer Control
(BS-D-166-0-MB-1)

MB, AR Special Input (BS-D-166-0-MBAR)

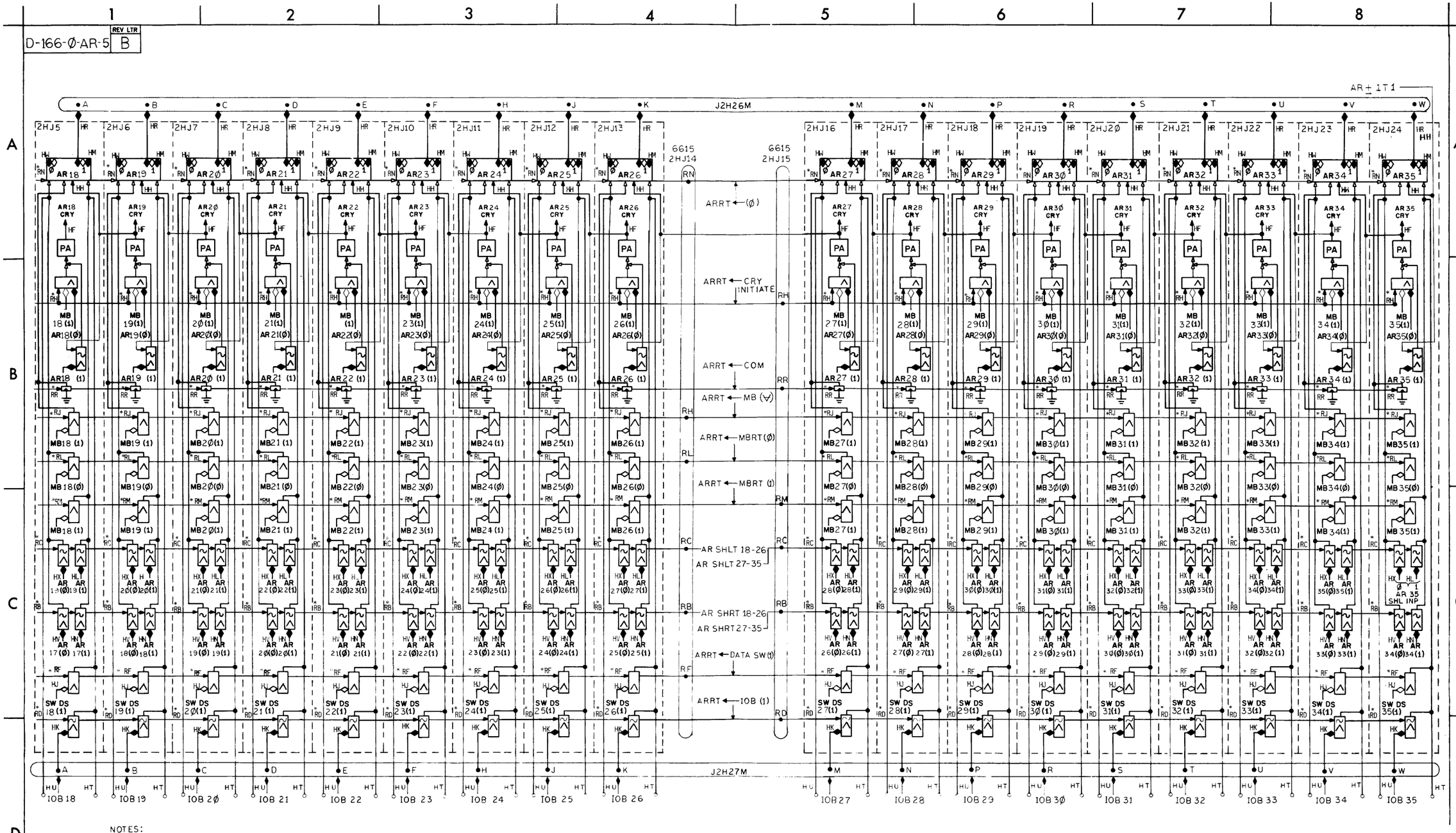


MB, AR Special Input (BS-D-166-0-MBAR)



ARLT Register 0-17 (BS-D-166-0-AR-4)

D-166-Ø-AR-5 | REV LTR B



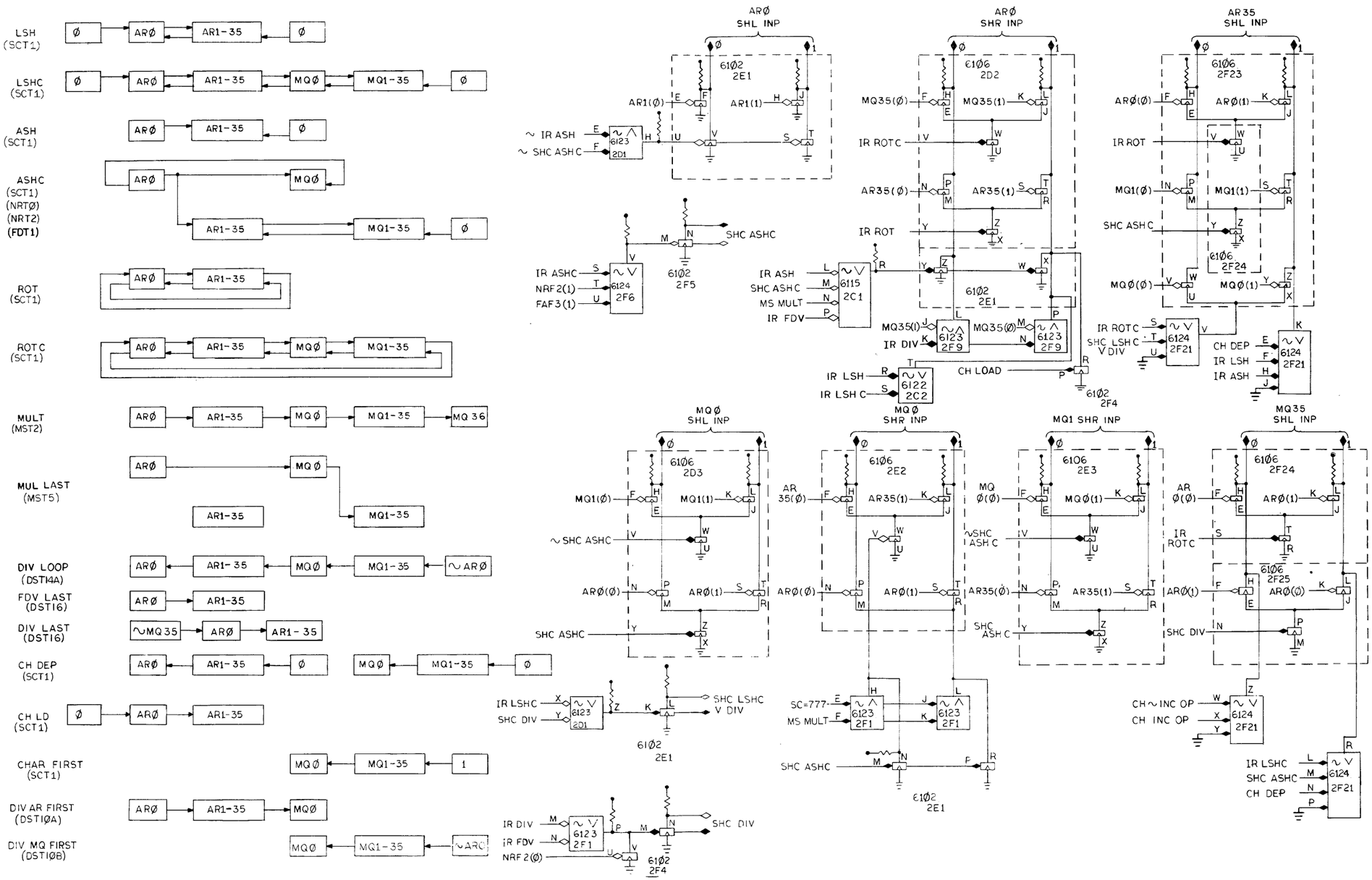
NOTES:

1. ALL FF PACKAGES ARE 62Ø5
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D AND Z IN ALL 62Ø5's AND 6615's.

ARRT Register 18-35 (BS-D-166-0-AR-5)

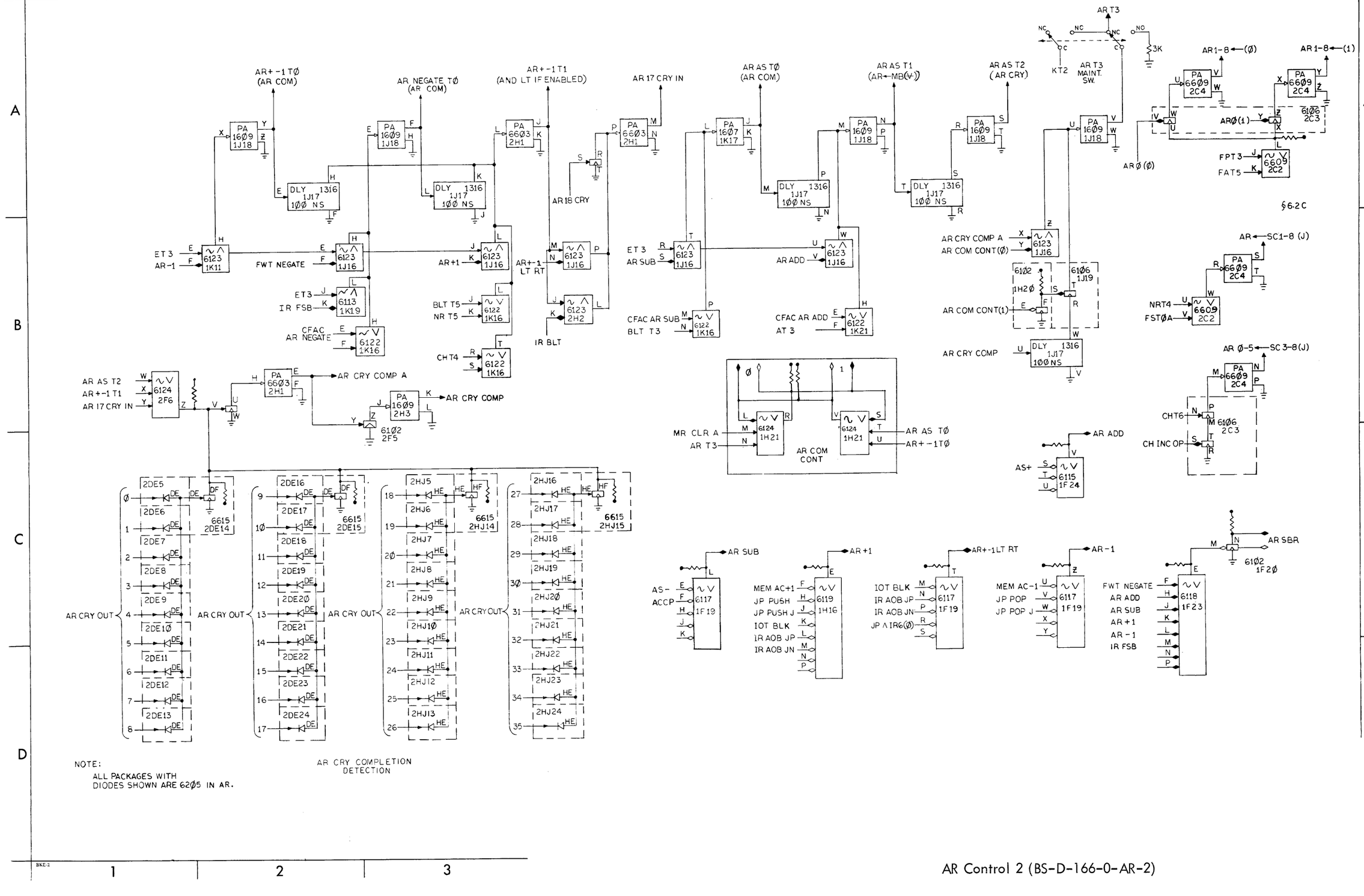
AR, MQ Shift Connections (BS-D-166-0-ARMQ)

D-166-Ø-ARMQ REV LTR D

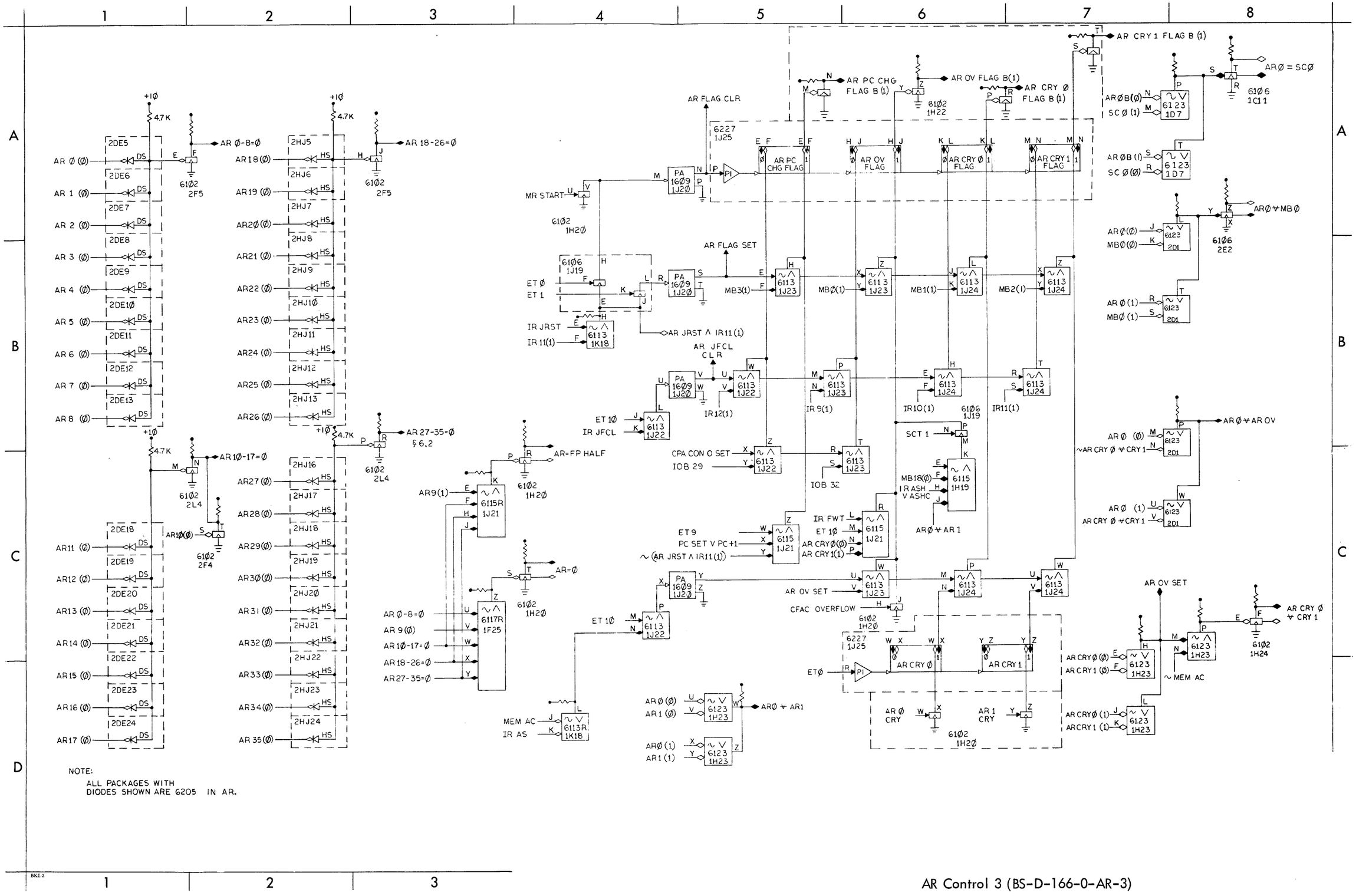


AR, MQ Shift Connections (BS-D-166-0-ARMQ)

AR Control 1 (BS-D-166-0-AR-1)



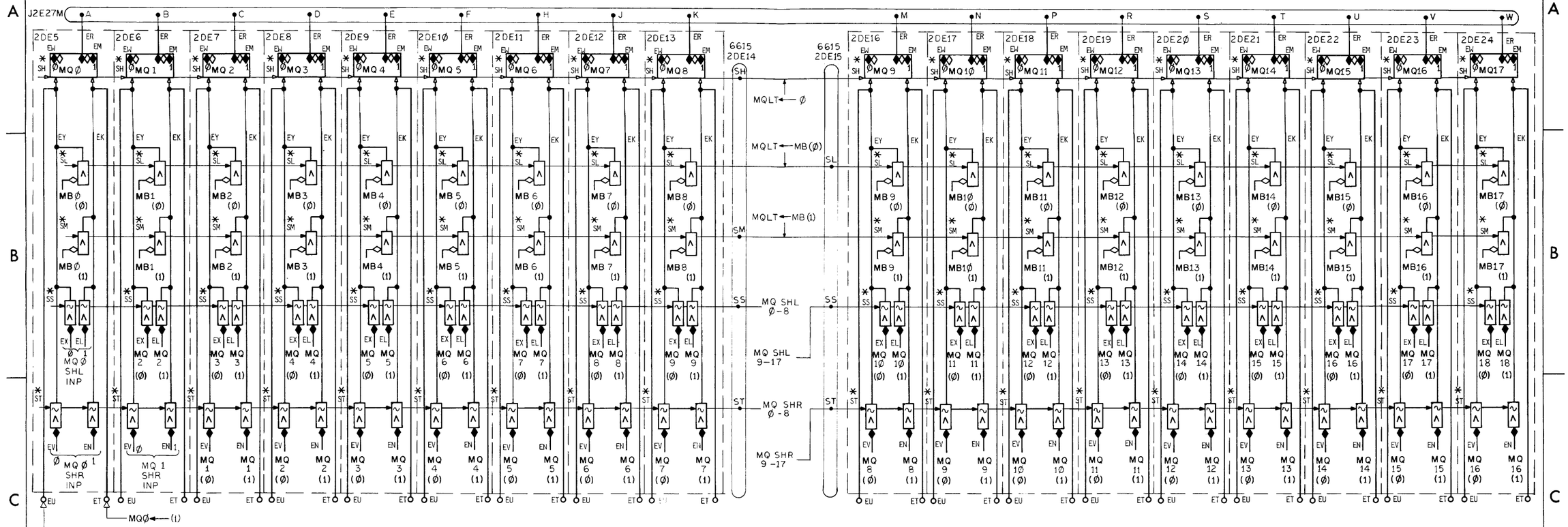
AR Control 2 (BS-D-166-0-AR-2)



BKE-2

AR Control 3 (BS-D-166-0-AR-3)

MQLT Register 0-17 (BS-D-166-0-MQ-2)

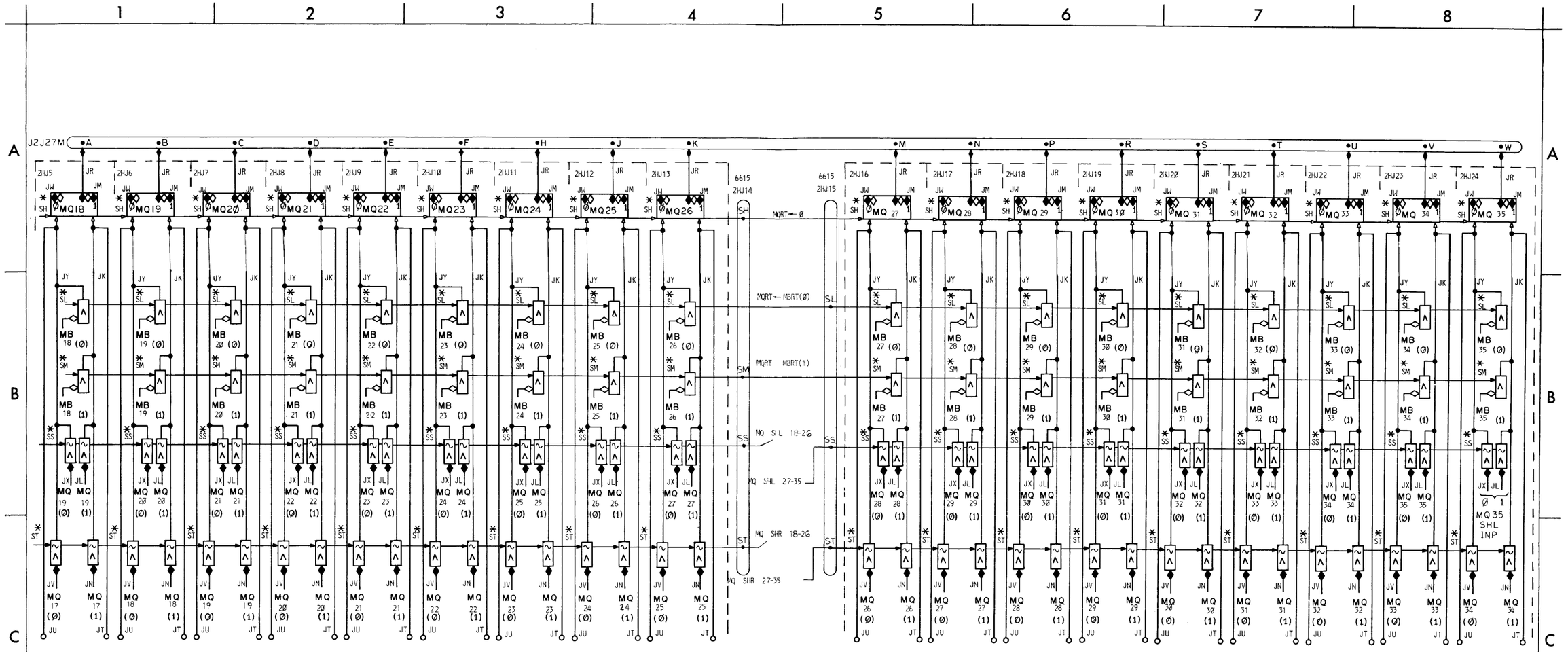


NOTES:

1. ALL FF PACKAGES ARE 62Ø5
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D,P AND Z IN ALL 62Ø5'S AND 6615'S.

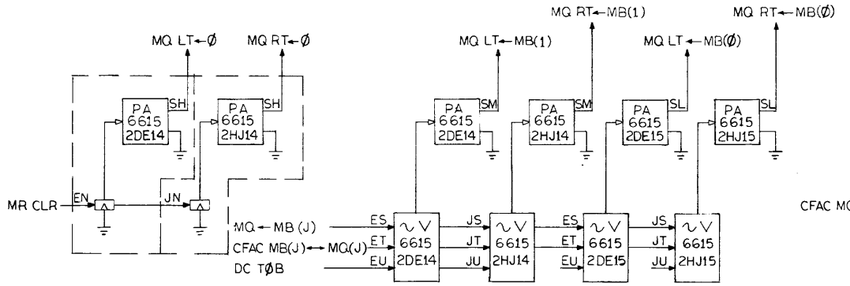
MQLT Register 0-17 (BS-D-166-0-MQ-2)

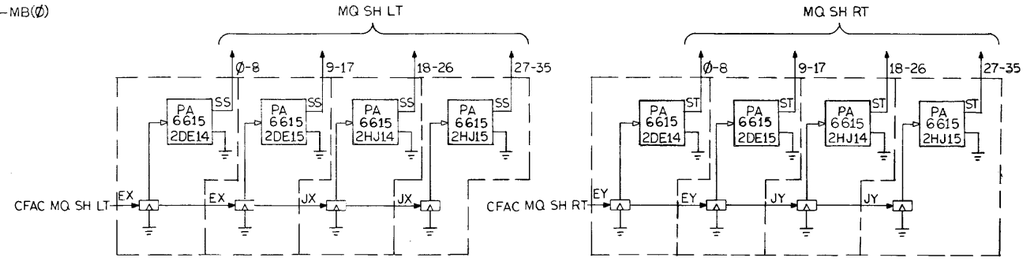
MQRT Register 18-35 (BS-D-166-0-MQ-3)

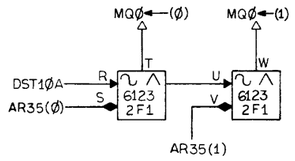


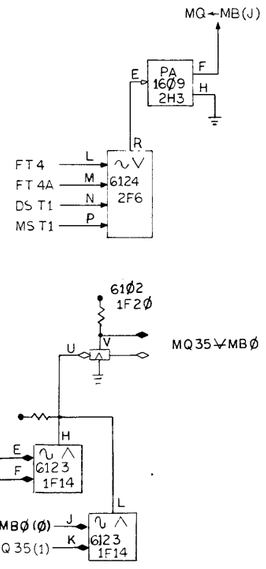
MQRT Register 18-35 (BS-D-166-0-MQ-3)

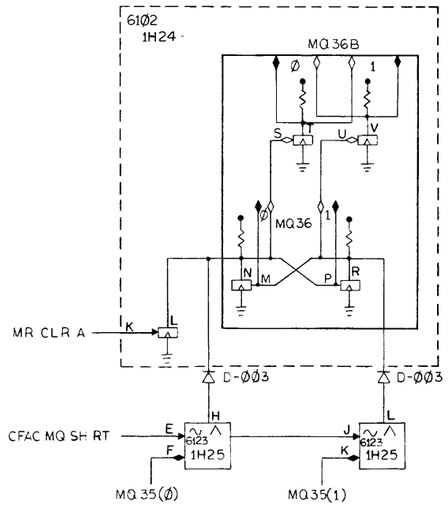
MQ - Multiplier Quotient Control
(BS-D-166-0-MQ-1)

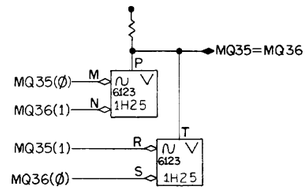






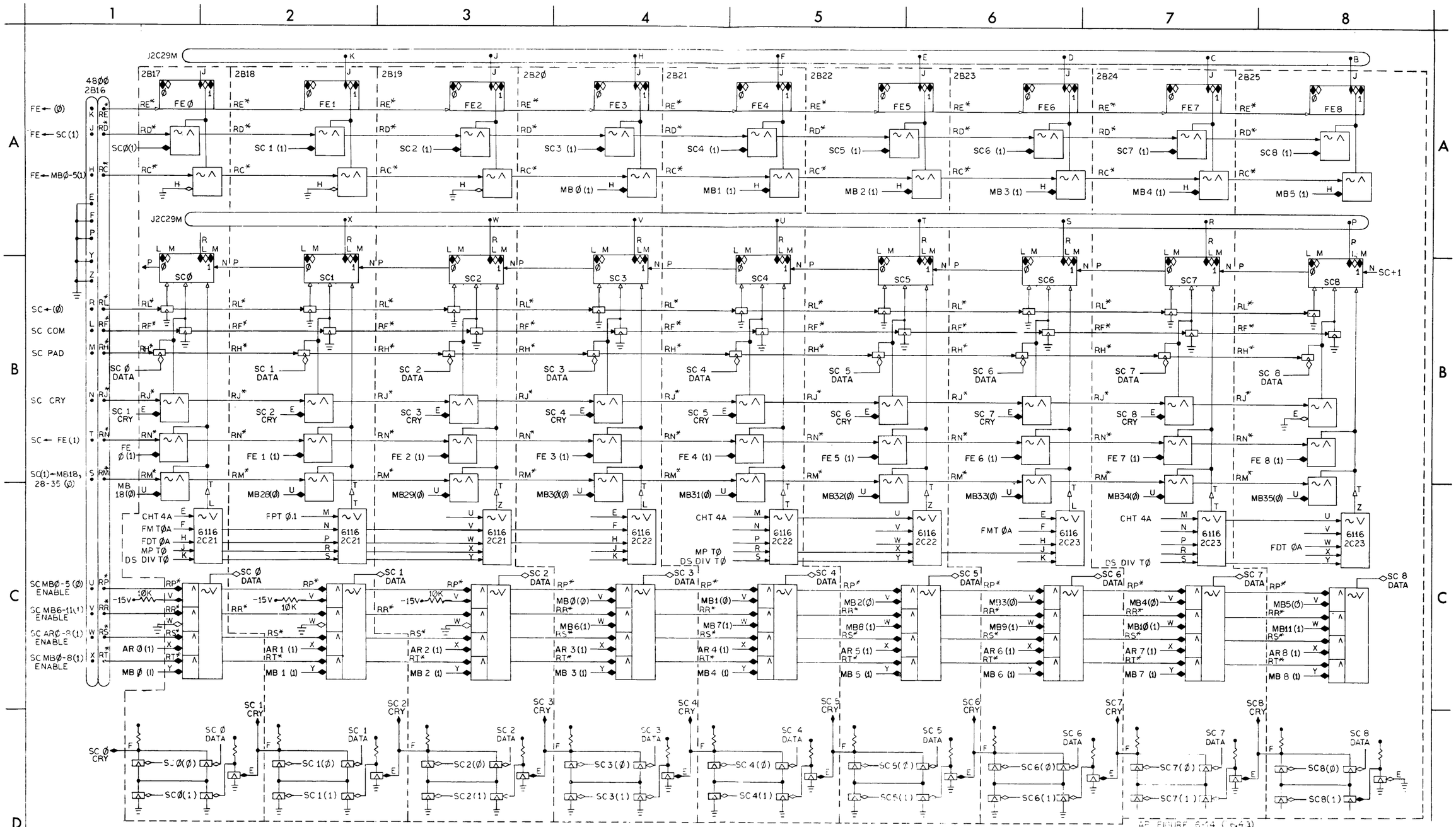






MQ - Multiplier Quotient Control
(BS-D-166-0-MQ-1)

SC, FE - Shift Counter, Floating Exponent
(BS-D-166-0-SCFE)

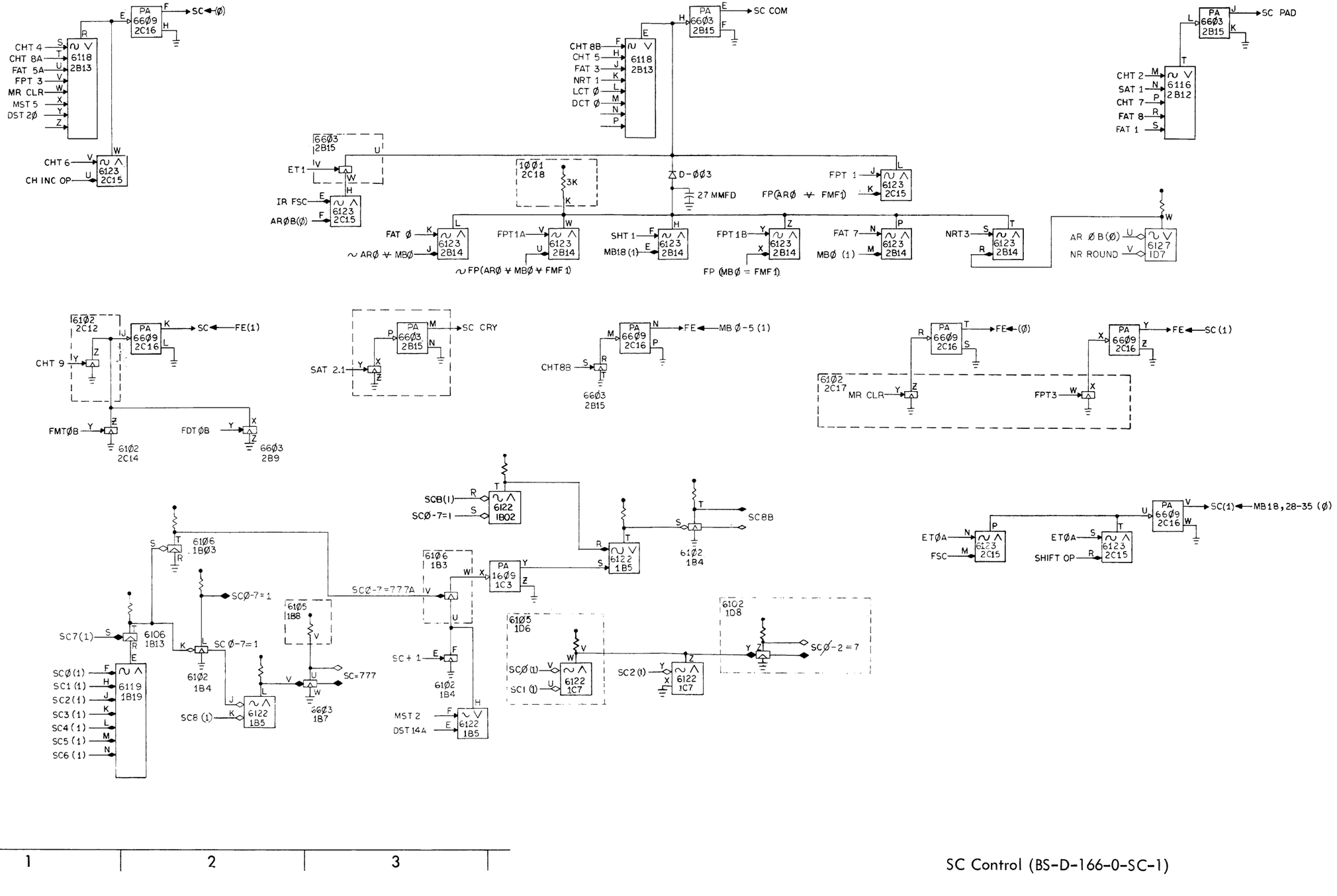


NOTE:

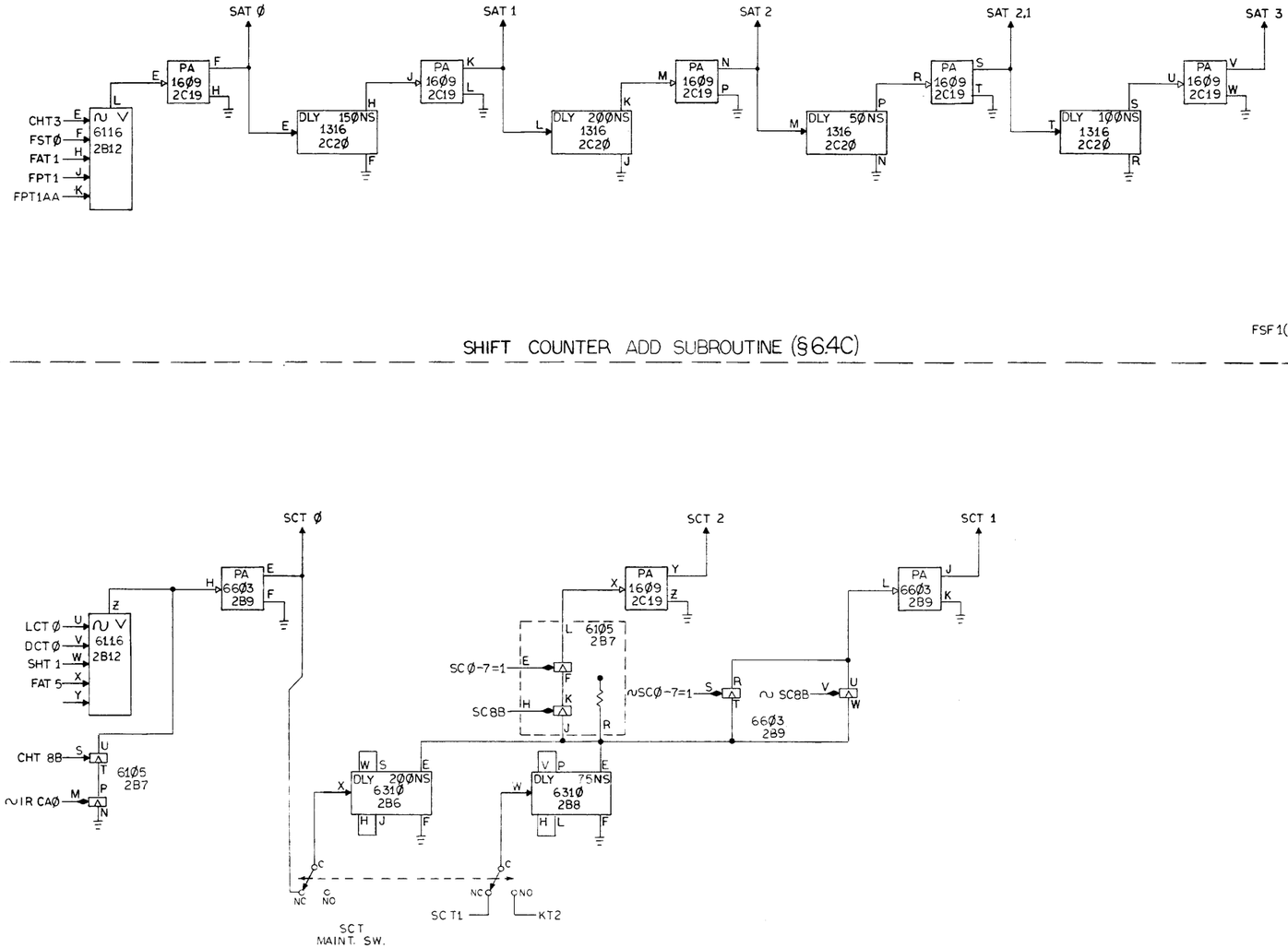
1. UNLESS OTHERWISE INDICATED ALL PACKAGES ARE 6203s.
2. * INDICATES REAR CONNECTOR PIN.

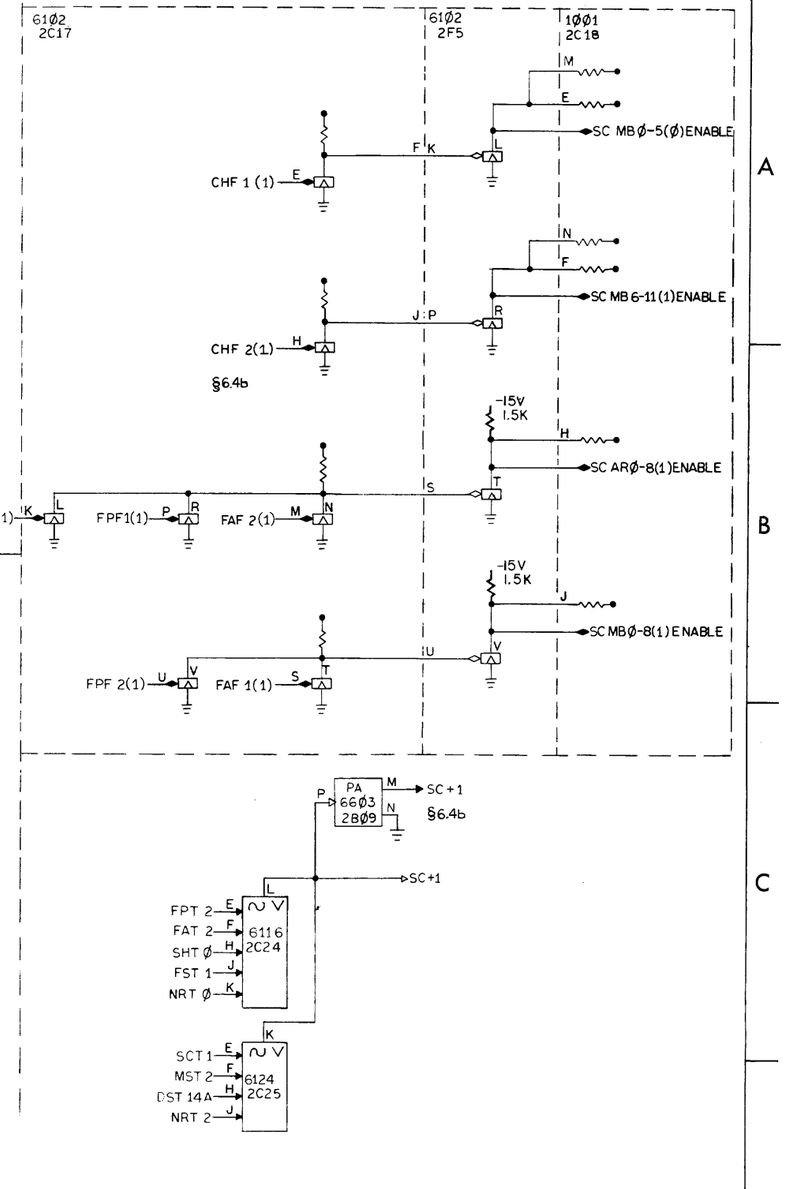
AP FIGURE 5-14 (5.4.1)

SC, FE - Shift Counter, Floating Exponent
(BS-D-166-0-SCFE)



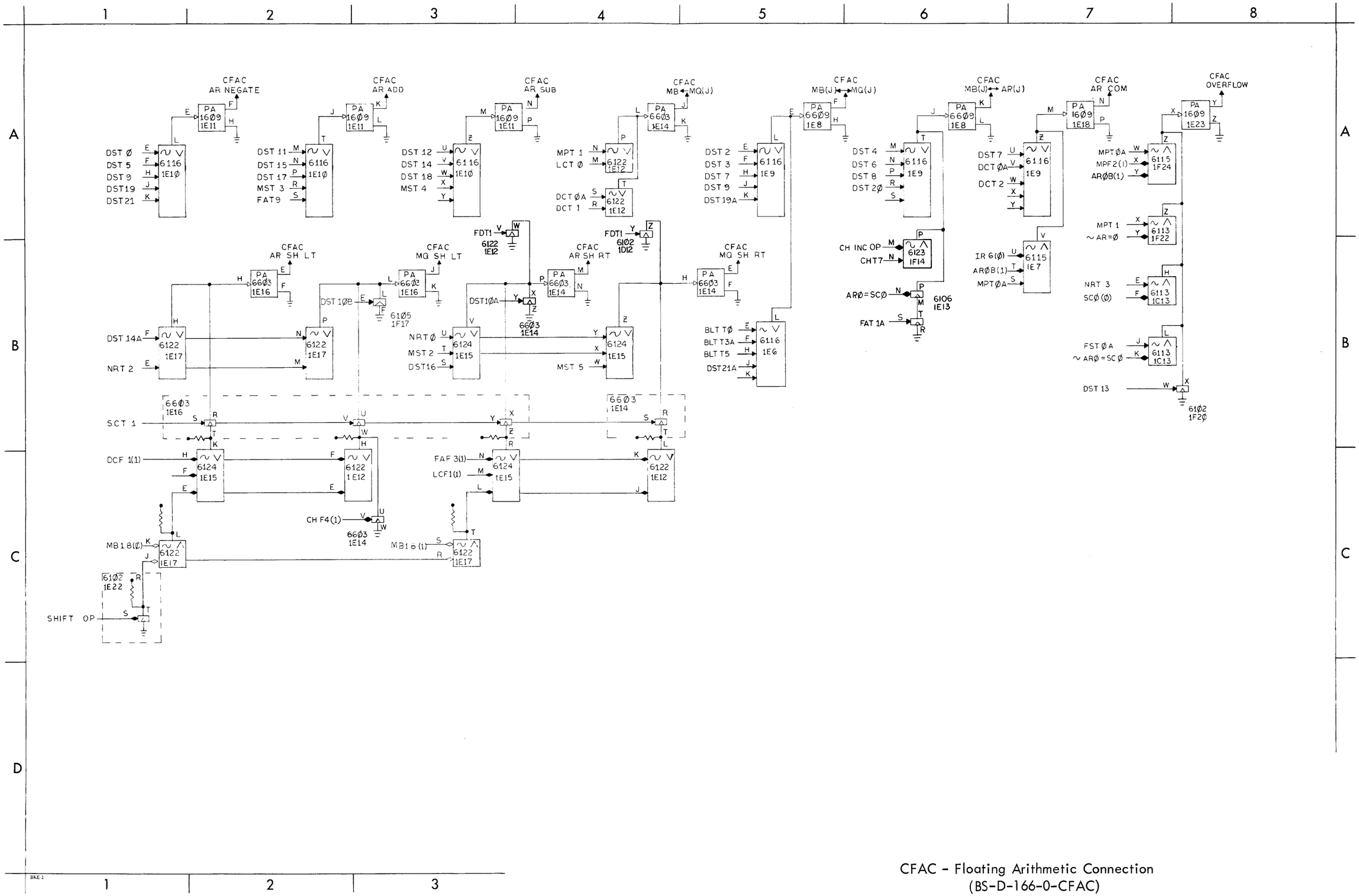
SC Control (BS-D-166-0-SC-1)





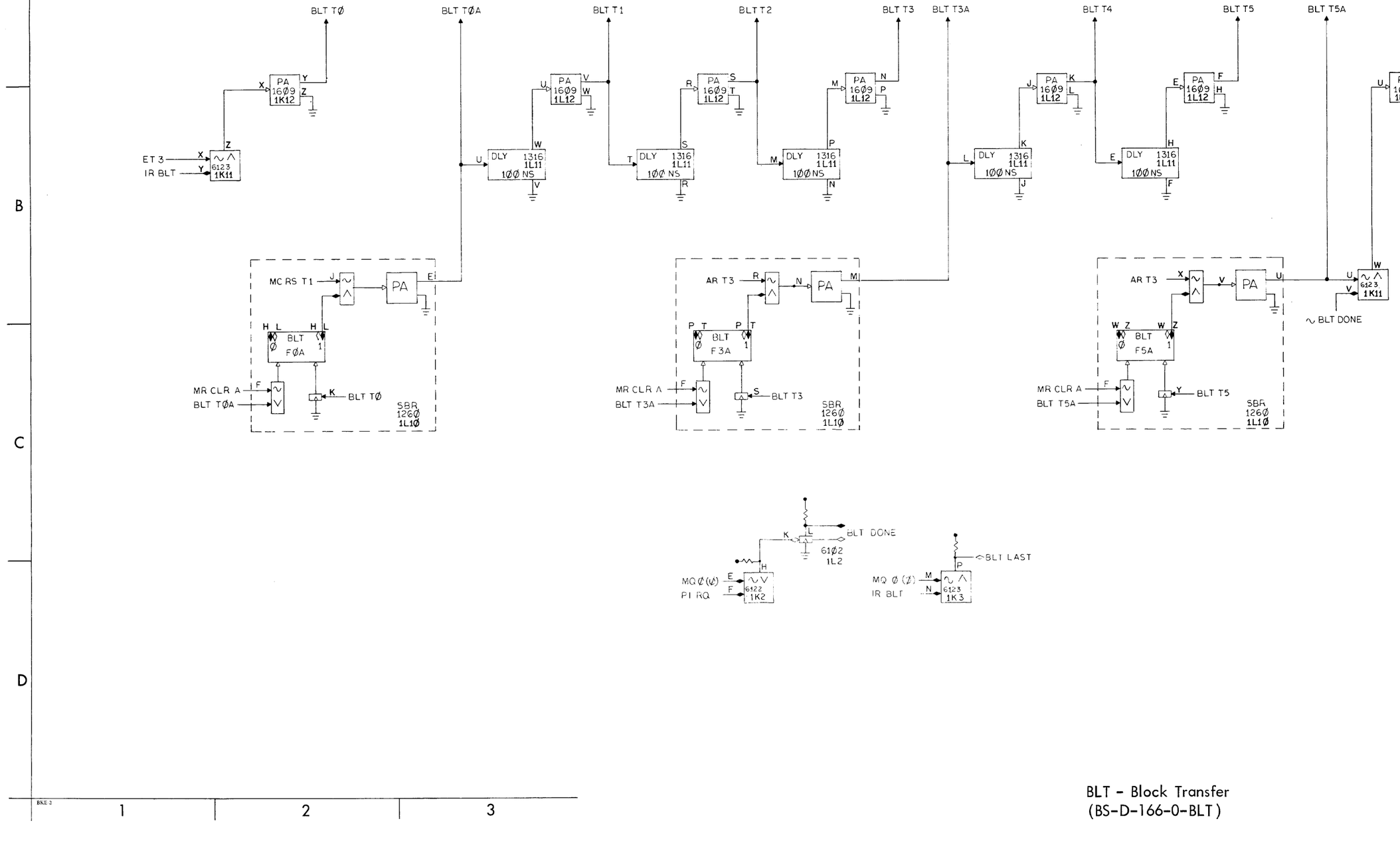
SC Control (BS-D-166-0-SC-2)

CFAC - Floating Arithmetic Connection
(BS-D-166-0-CFAC)



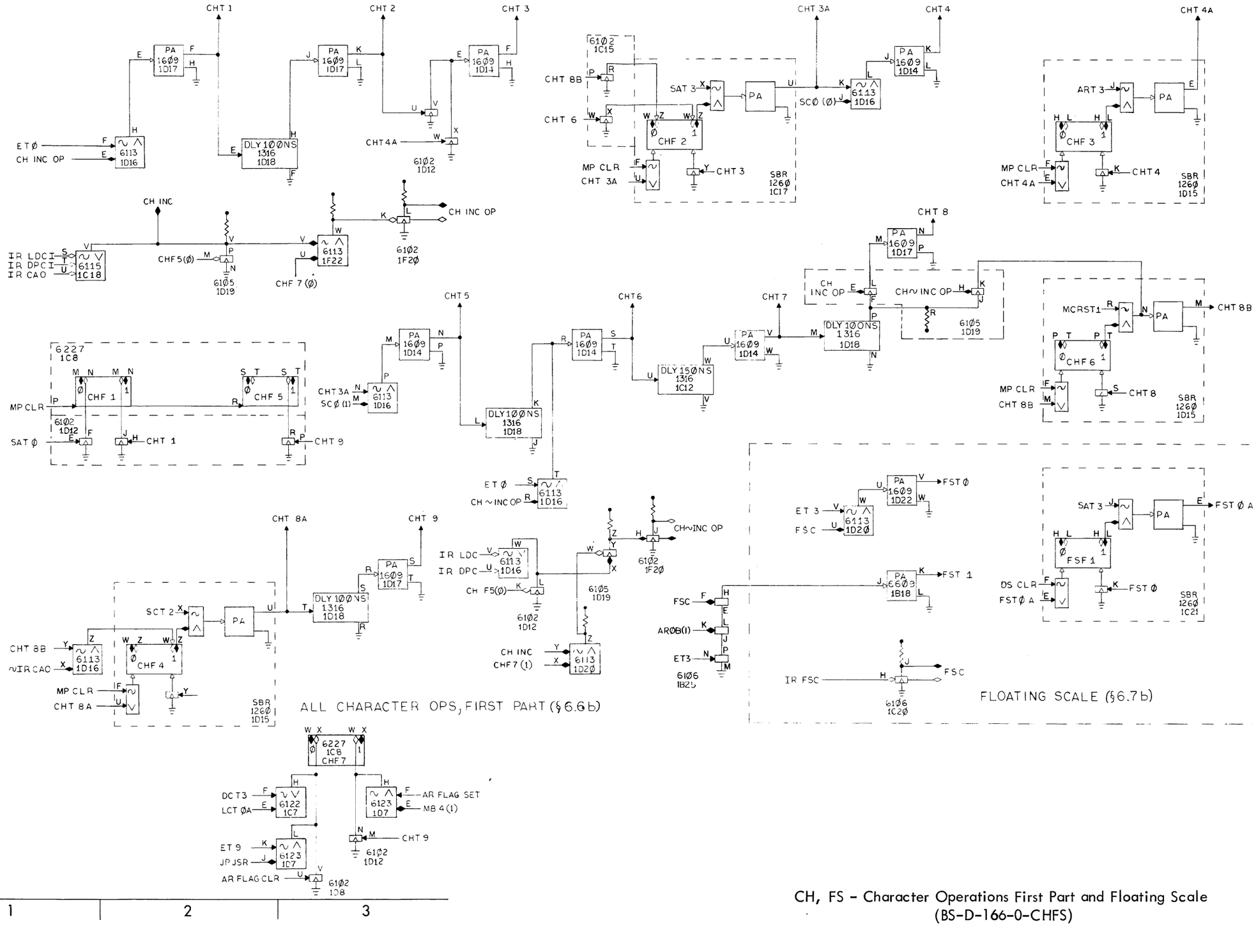
CFAC - Floating Arithmetic Connection
(BS-D-166-0-CFAC)

BLT - Block Transfer
(BS-D-166-0-BLT)



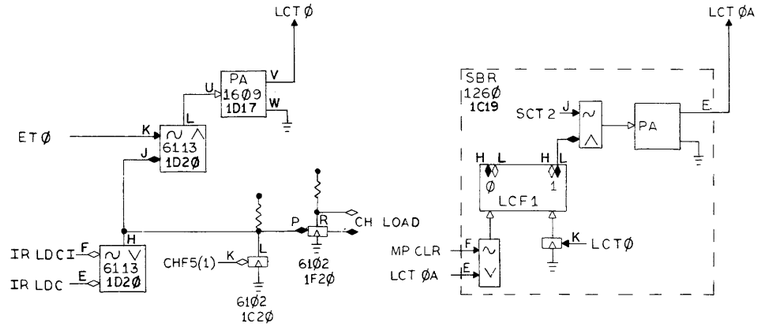
BLT - Block Transfer
(BS-D-166-0-BLT)

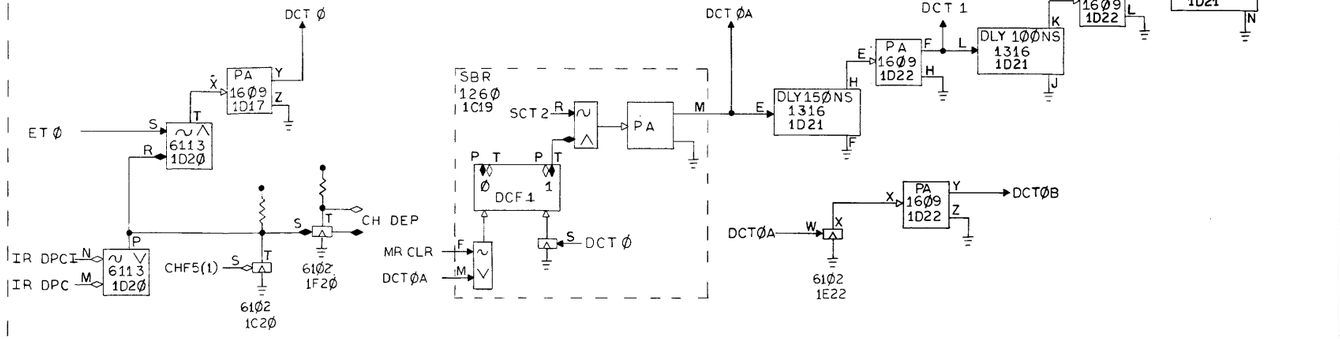
CH, FS - Character Operations First Part and Floating Scale
(BS-D-166-0-CHFS)



CH, FS - Character Operations First Part and Floating Scale
(BS-D-166-0-CHFS)

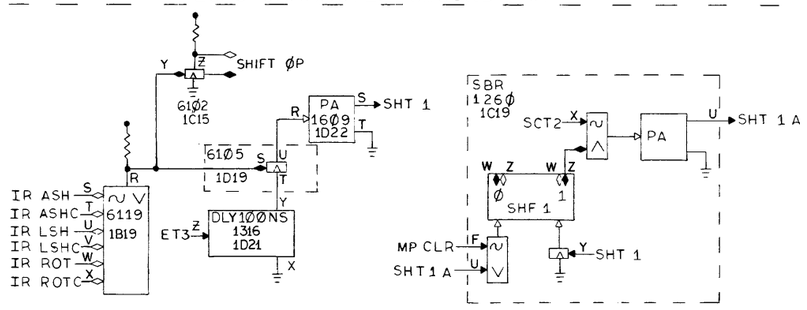
LC, DC, SH - Character Operations Second Part
and Shift Operations
(BS-D-166-0-LDCS)

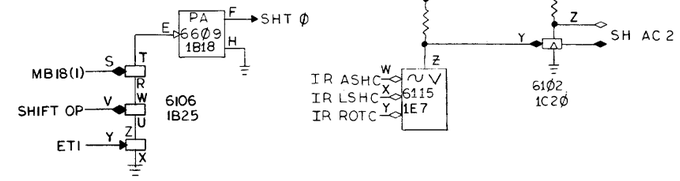




LOAD CHARACTER SECOND PART

DEPOSIT CHARACTER SECOND PART

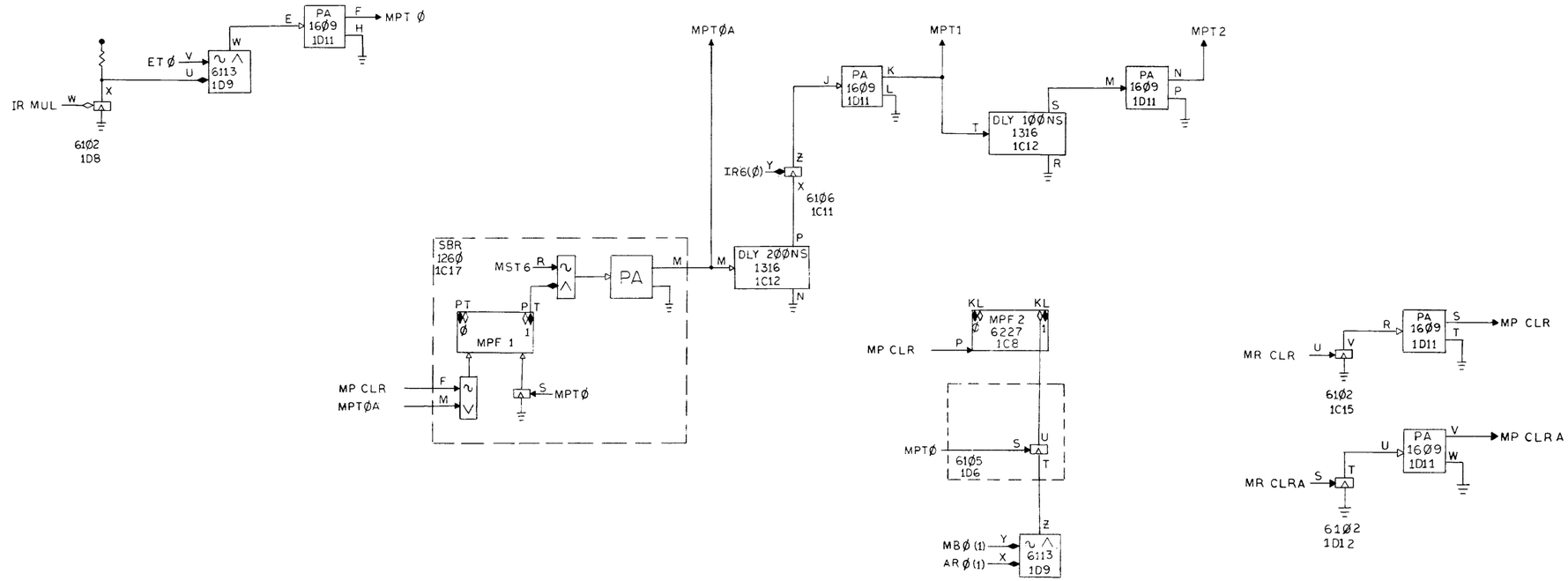




SHIFT OPERATIONS

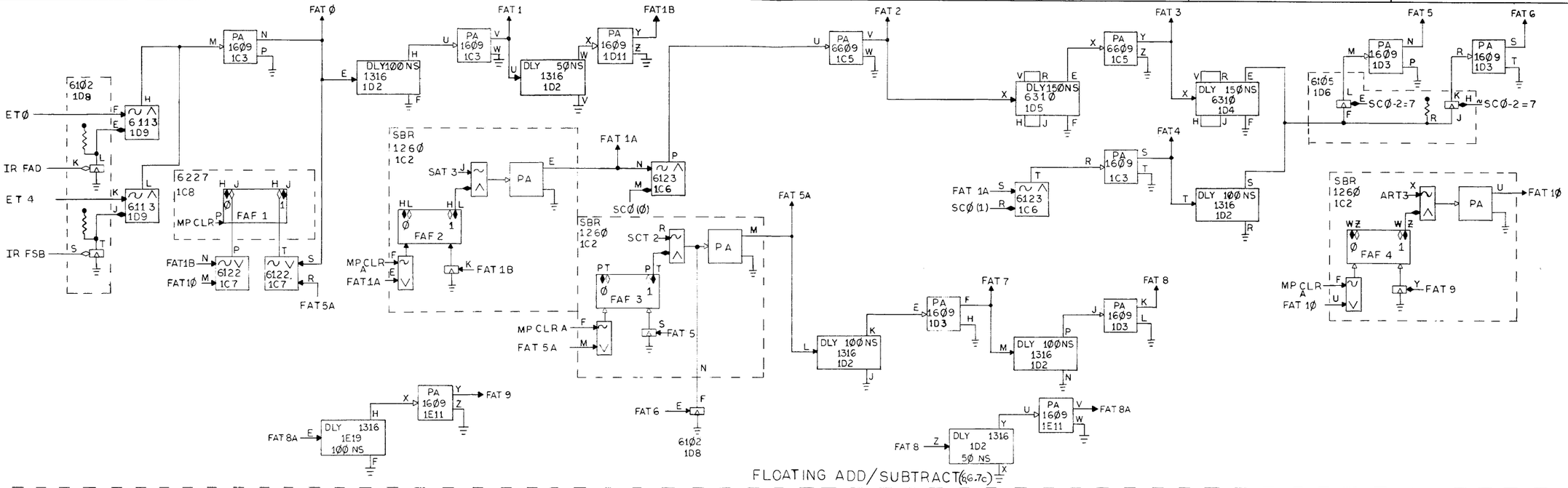
LC, DC, SH - Character Operations Second Part
and Shift Operations
(BS-D-166-0-LDCS)

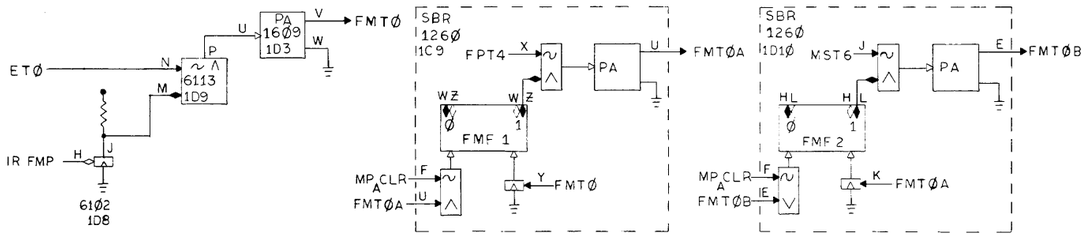
MP - Fixed Point Multiply
(BS-D-166-0-MP)



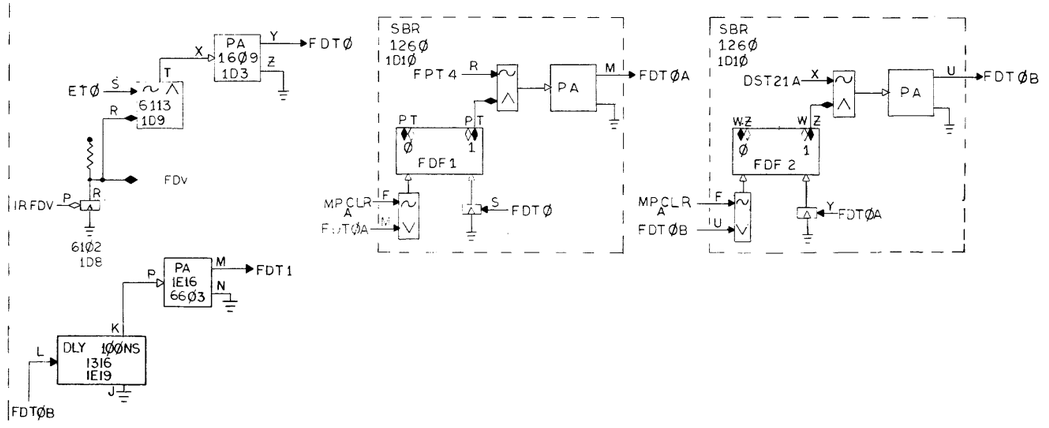
FIXED POINT MULTIPLY

MP - Fixed Point Multiply
(BS-D-166-0-MP)





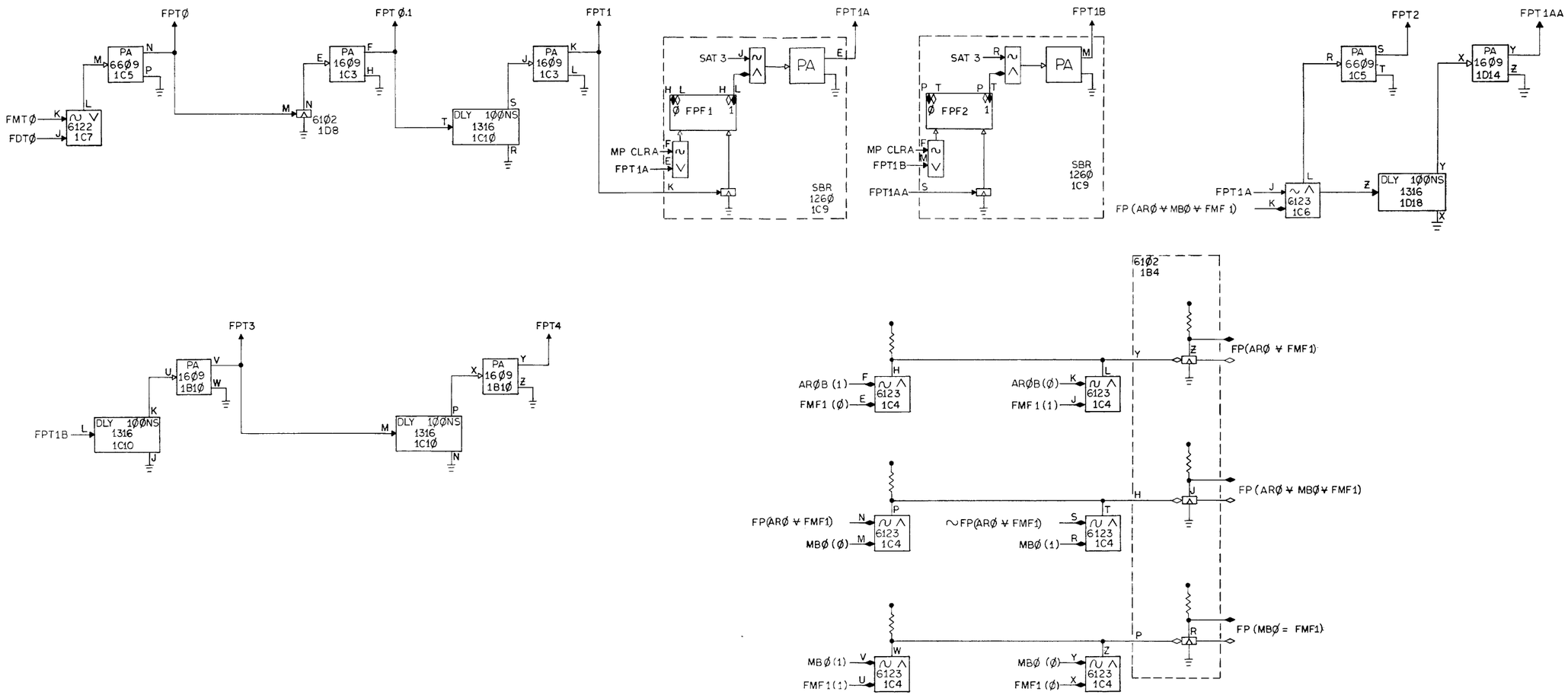
FLOATING MULTIPLY (§6.7d)



FLOATING DIVIDE (§6.7d)

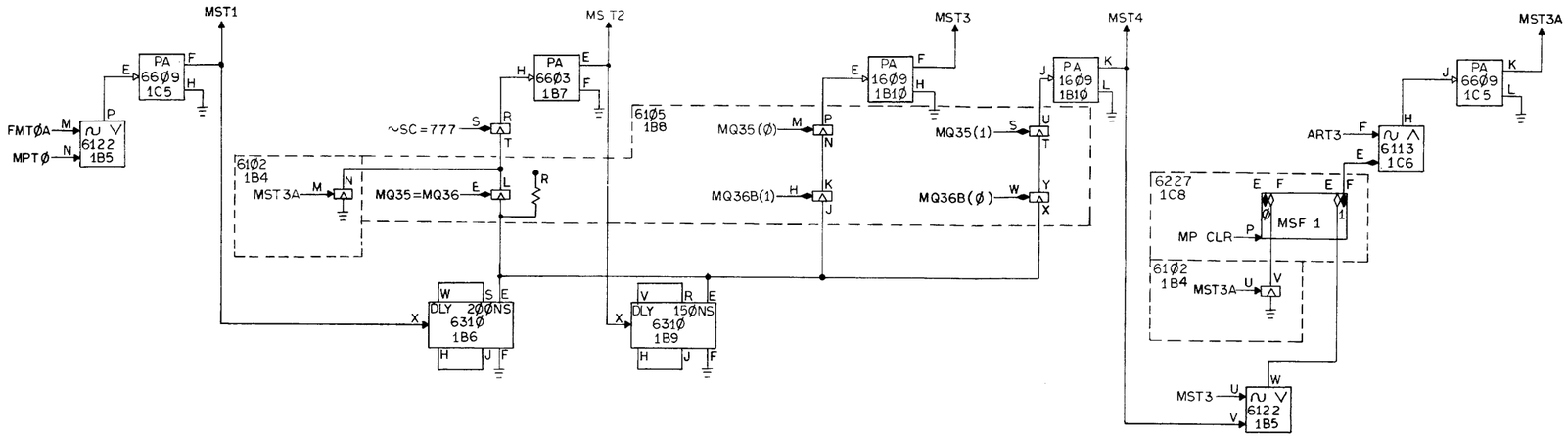
FA, FD, FM - Floating Point Instructions
(BS-D-166-0-FADM)

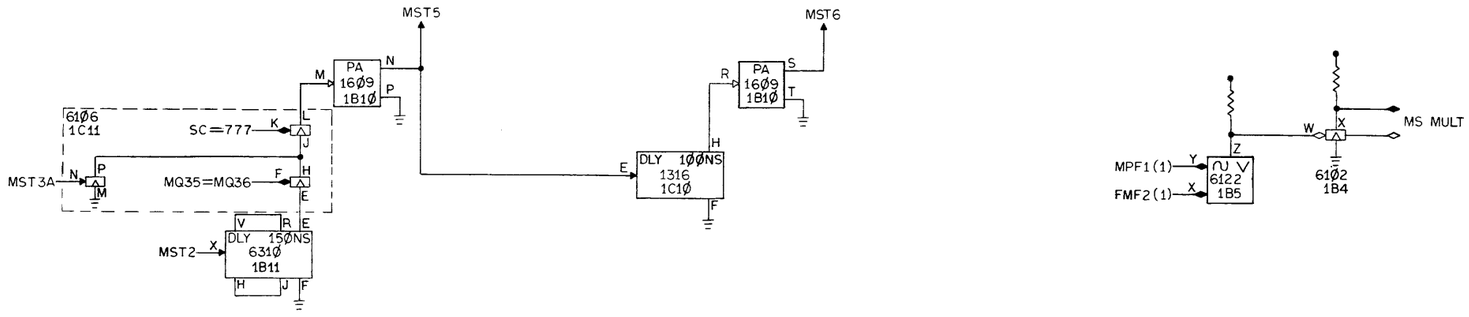
FP - Exponent Calculate Subroutine
(BS-D-166-0-FP)



FP - Exponent Calculate Subroutine
(BS-D-166-0-FP)

MS - Multiply Subroutine (BS-D-166-0-MS)

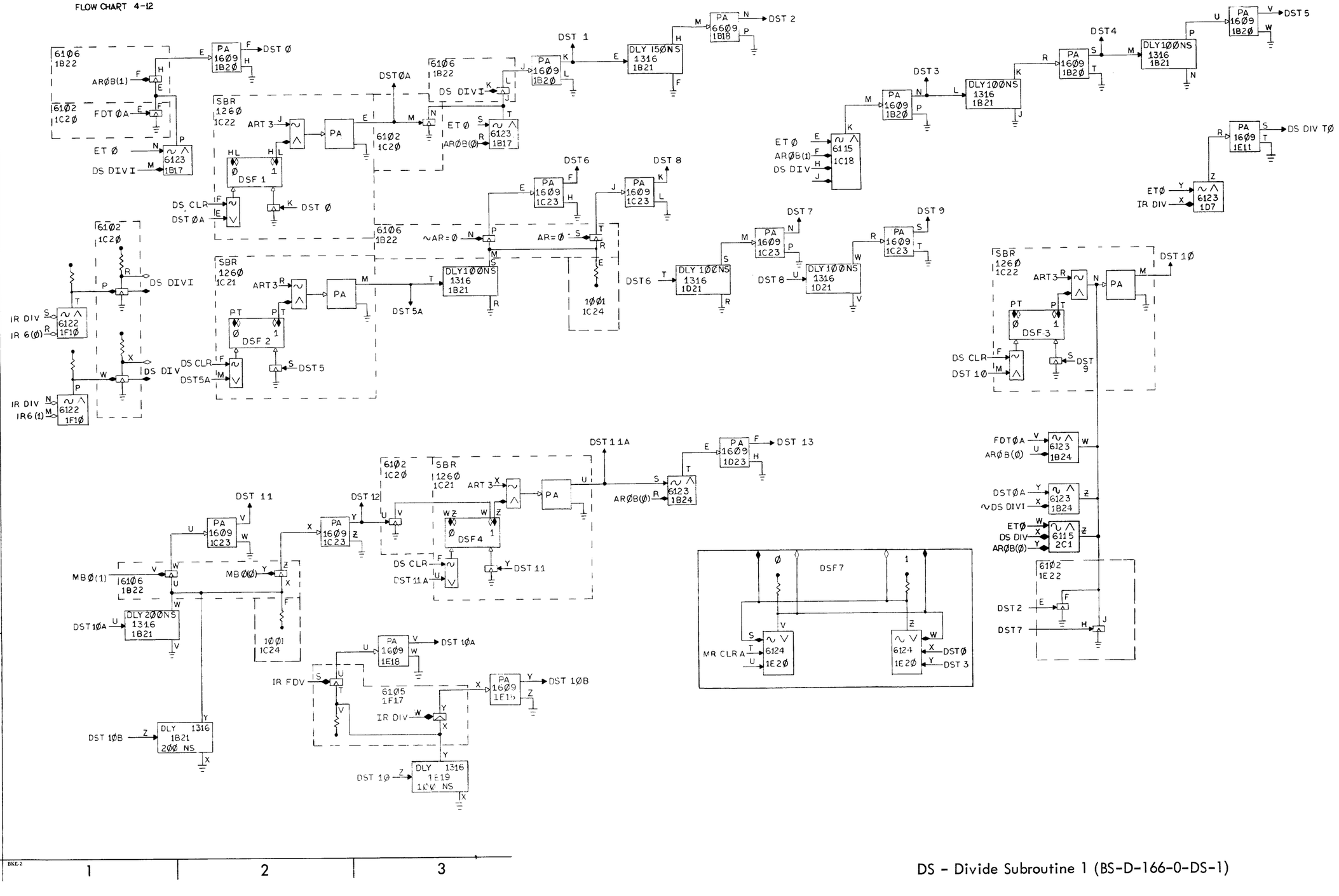




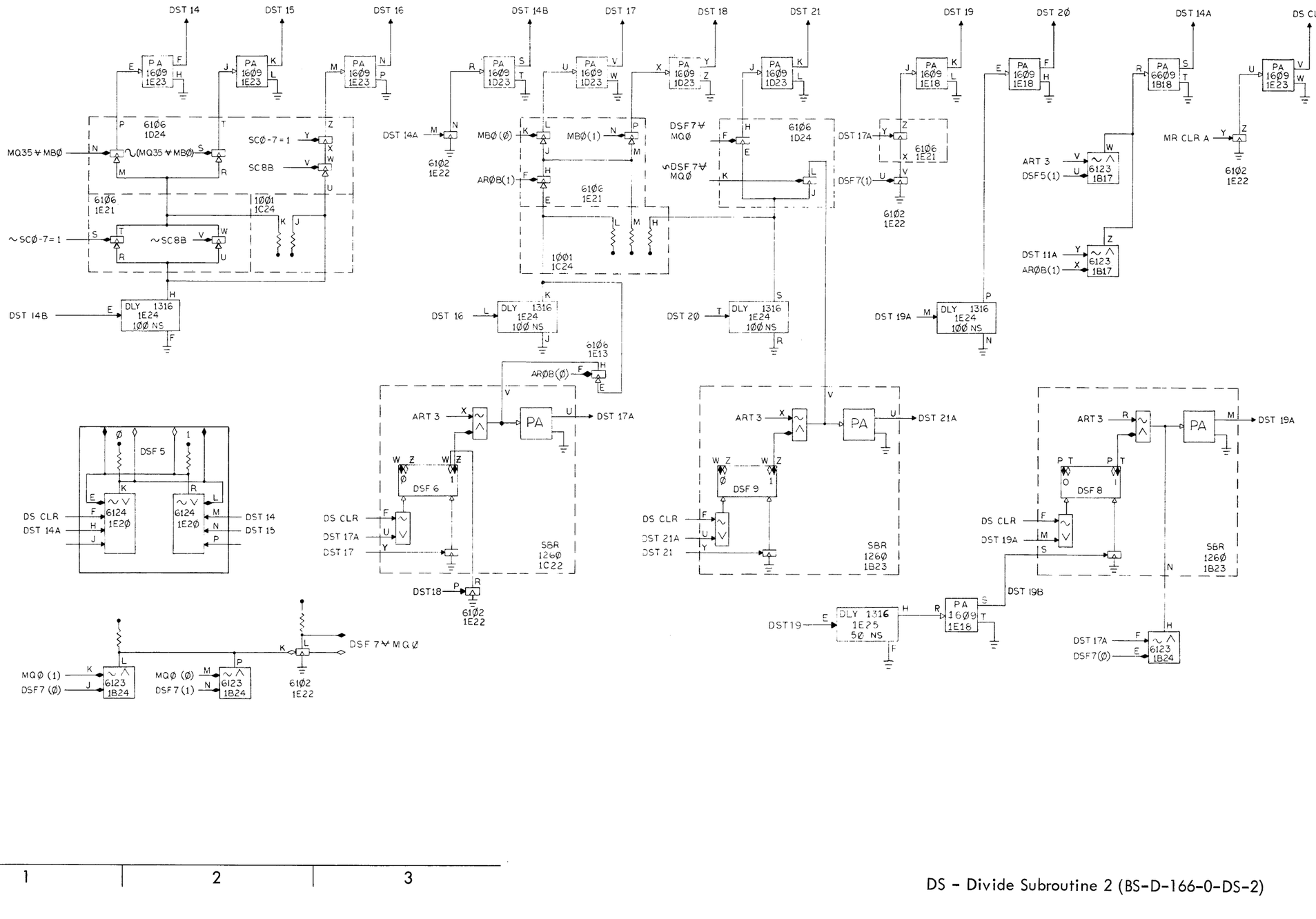
MS - Multiply Subroutine (BS-D-166-0-MS)

DS - Divide Subroutine 1 (BS-D-166-0-DS-1)

FLOW CHART 4-12

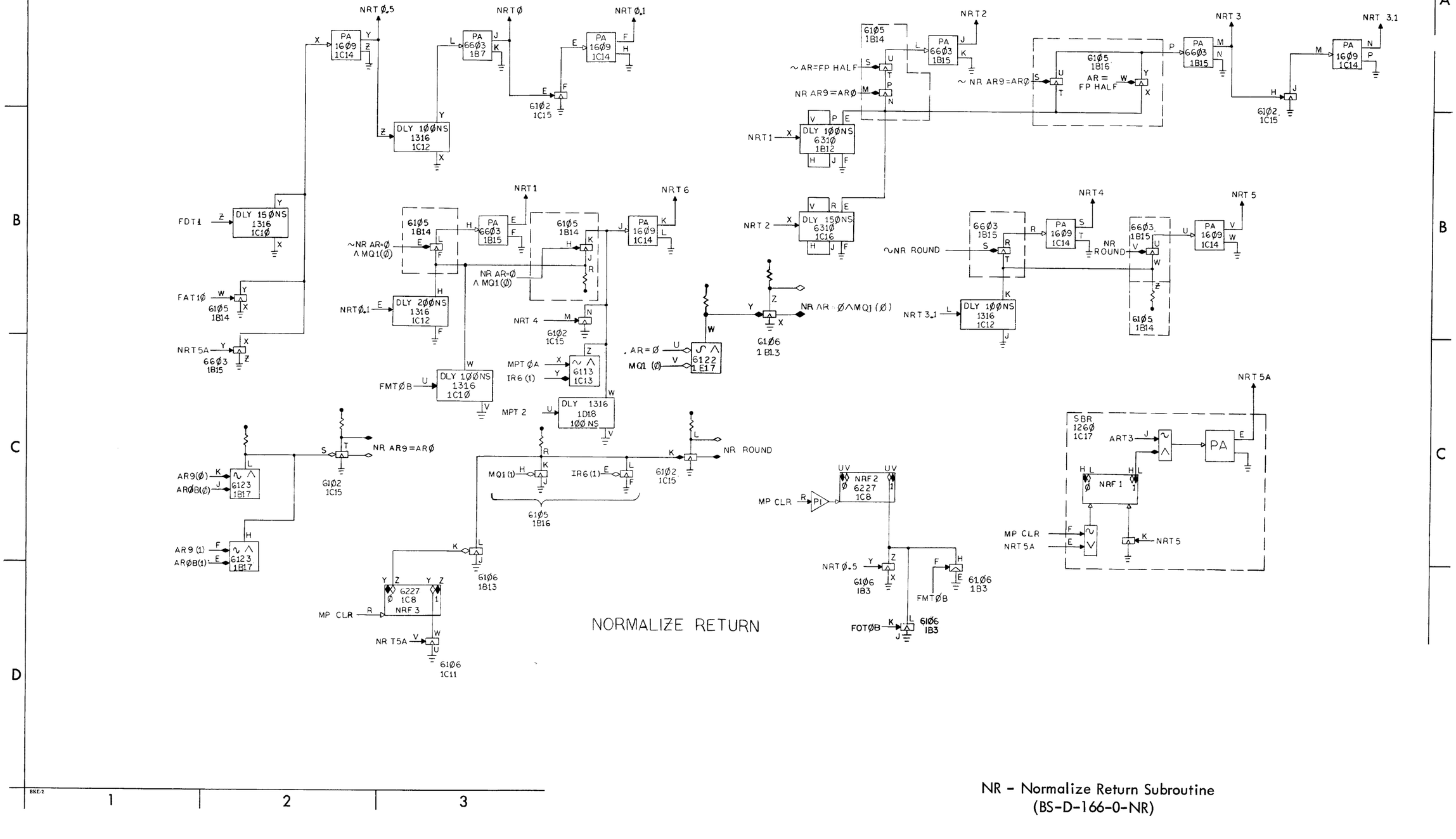


DS - Divide Subroutine 1 (BS-D-166-0-DS-1)



DS - Divide Subroutine 2 (BS-D-166-0-DS-2)

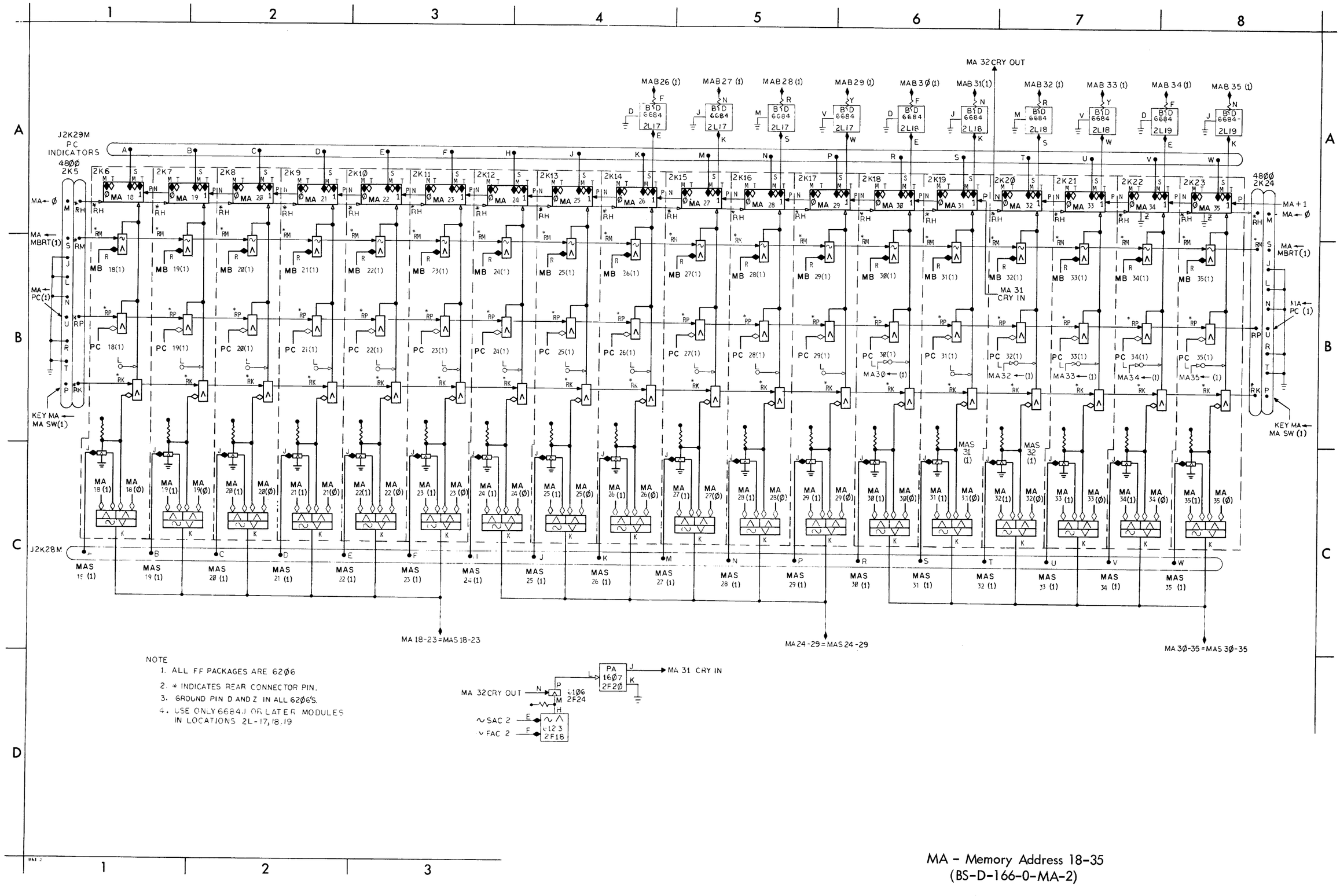
NR - Normalize Return Subroutine
(BS-D-166-0-NR)



NORMALIZE RETURN

NR - Normalize Return Subroutine
(BS-D-166-0-NR)

MA - Memory Address 18-35
(BS-D-166-0-MA-2)

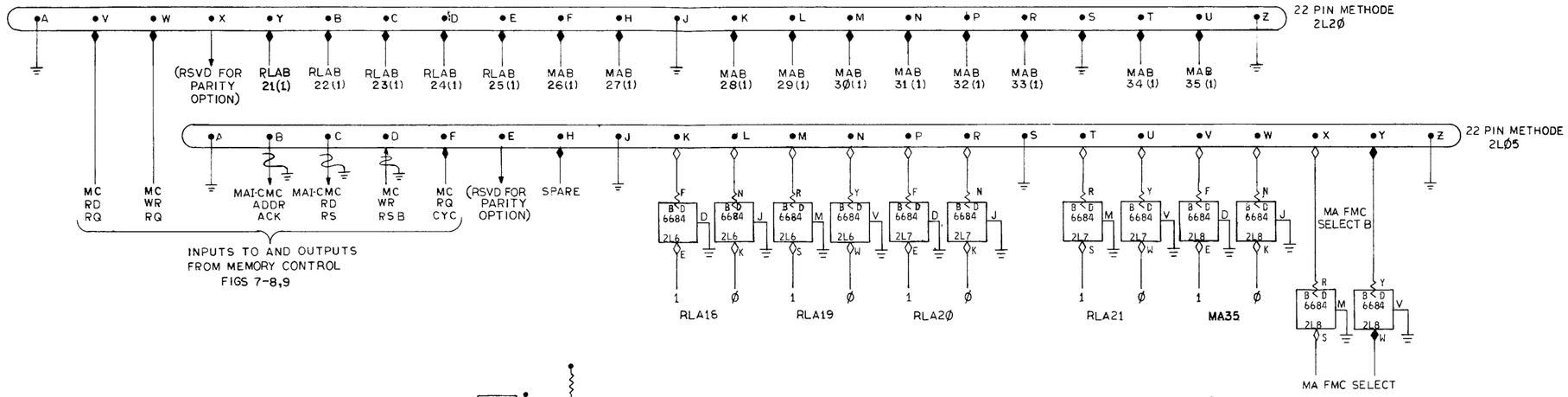


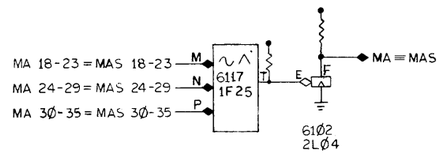
NOTE

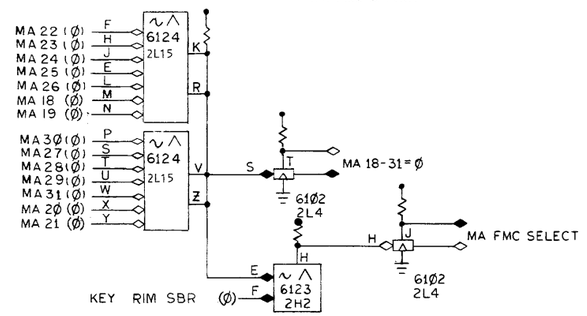
1. ALL FF PACKAGES ARE 62Ø6
2. * INDICATES REAR CONNECTOR PIN.
3. GROUND PIN D AND Z IN ALL 62Ø6'S.
4. USE ONLY 6684J OR LATER MODULES IN LOCATIONS 2L-17, 18, 19

MA - Memory Address 18-35
(BS-D-166-0-MA-2)

MA, MAI - Memory Address Interface
(BS-D-166-0-MA-3)



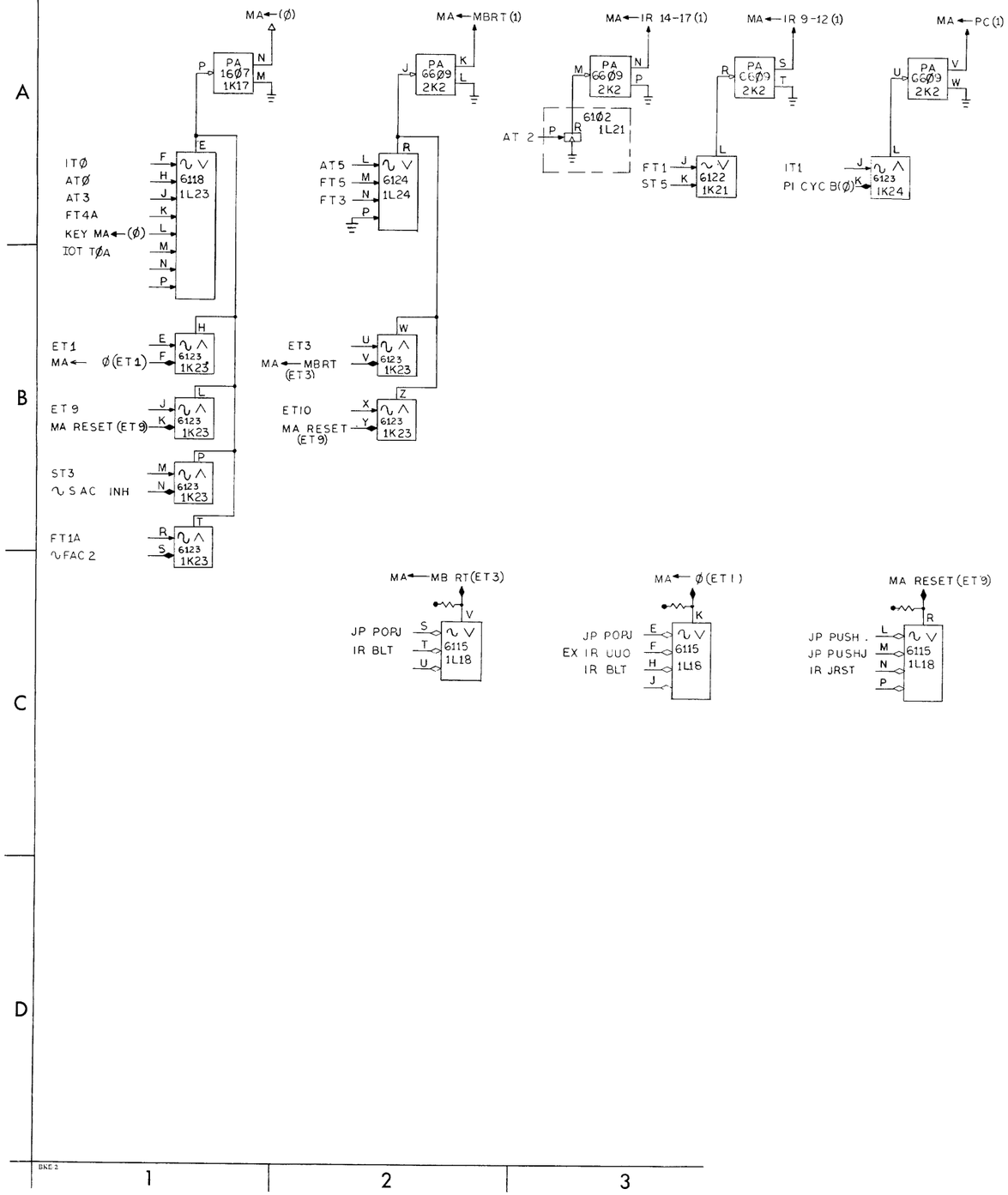




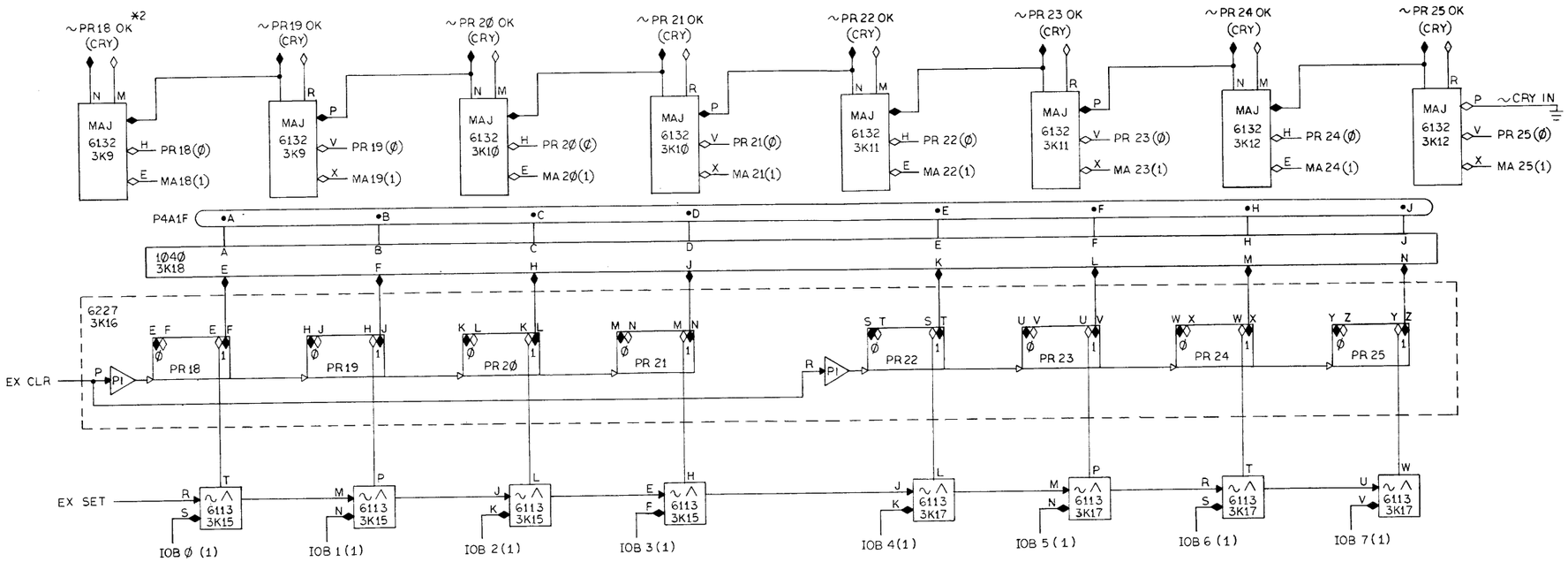
NOTE: 1 USE ONLY 6684J OR LATER MODULES IN LOCATIONS 2L-6,7,8

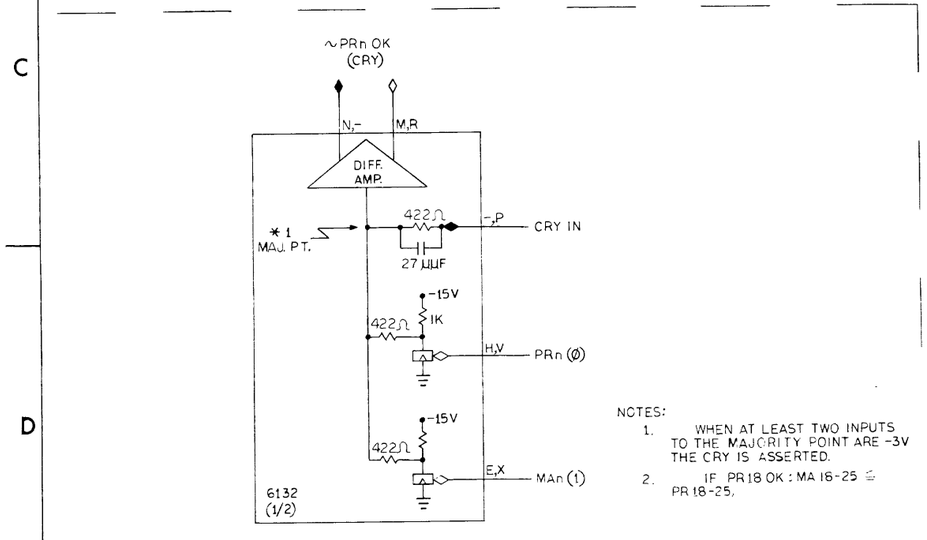
MA, MAI - Memory Address Interface
(BS-D-166-0-MA-3)

MA Control (BS-D-166-0-MA-1)



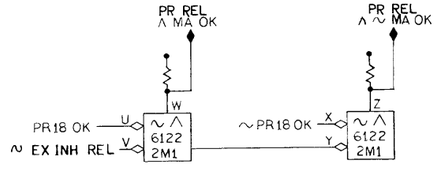
MA Control (BS-D-166-0-MA-1)



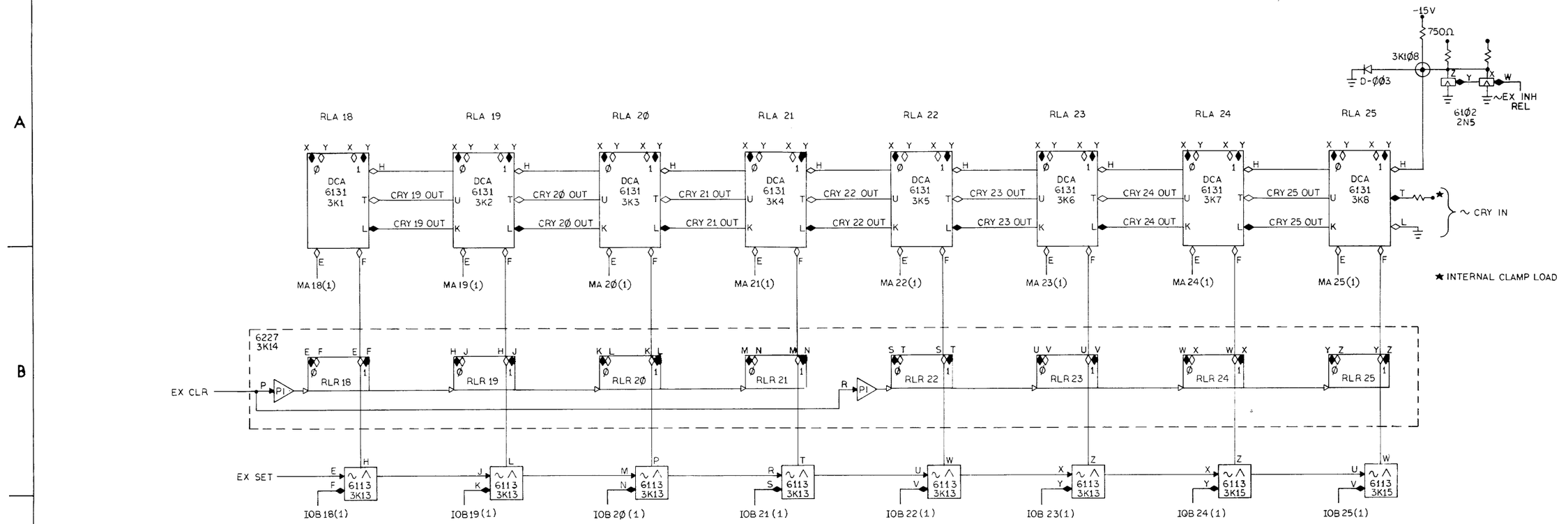


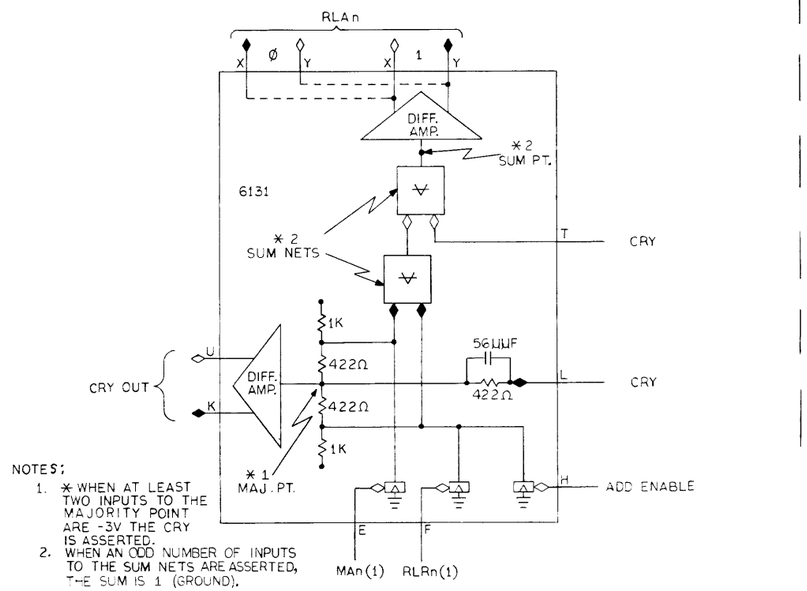
NOTES:

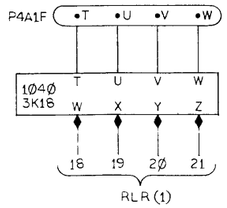
1. WHEN AT LEAST TWO INPUTS TO THE MAJORITY POINT ARE -3V THE CRY IS ASSERTED.
2. IF PR18 OK : MA 18-25 ≙ PR 18-25.

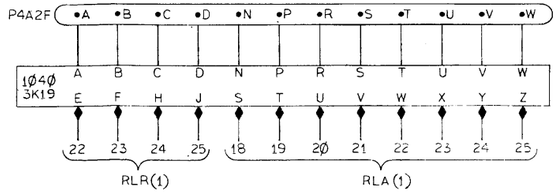


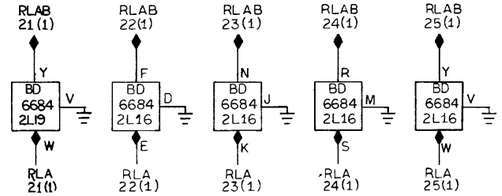
RLA, RLR - Relocation Adder and Register
(BS-D-166-0-RLAR)





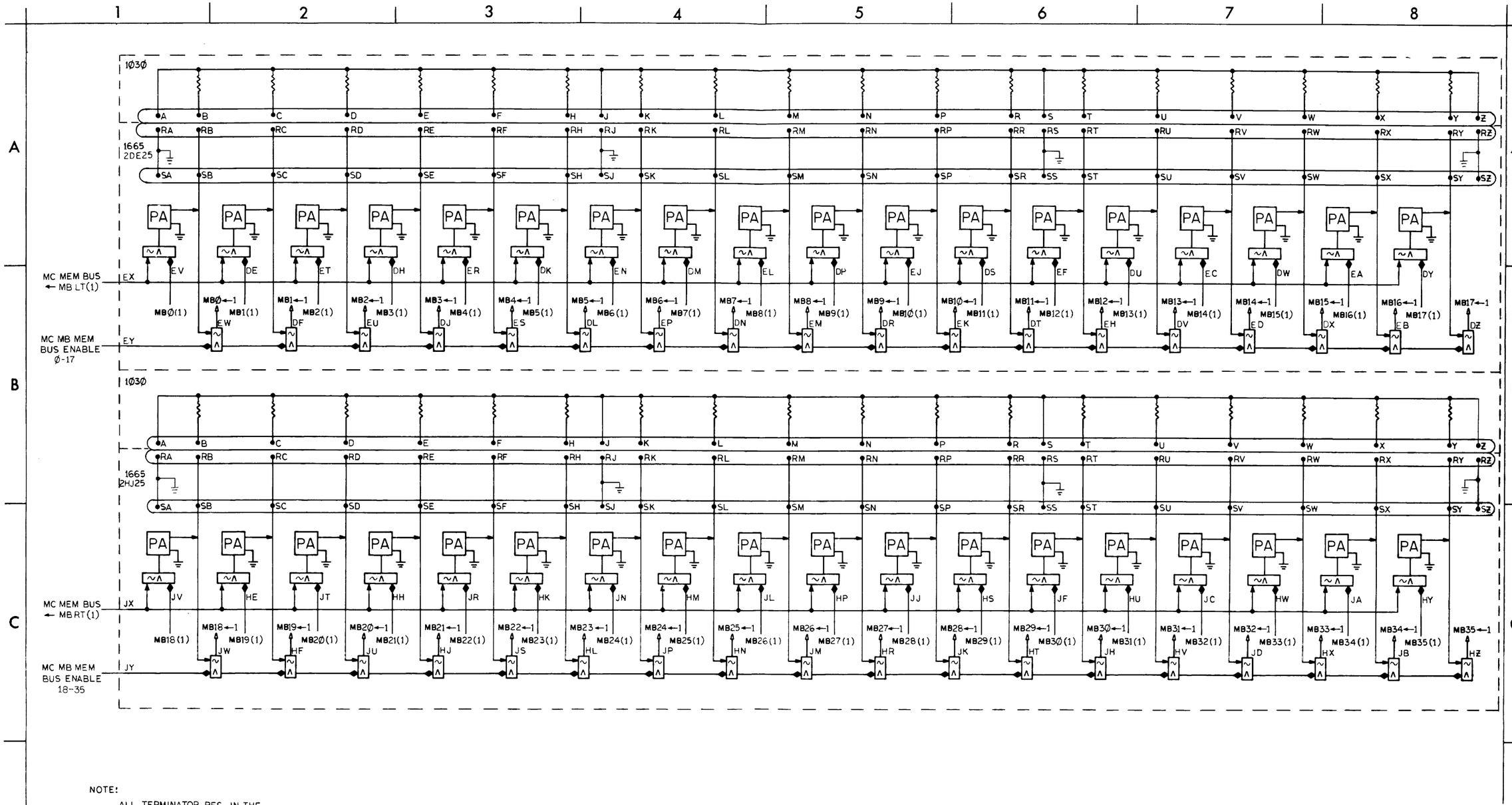






RLA, RLR - Relocation Adder and Register
(BS-D-166-0-RLAR)

MB - Memory Buffer Interface
(BS-D-166-0-MB-4)

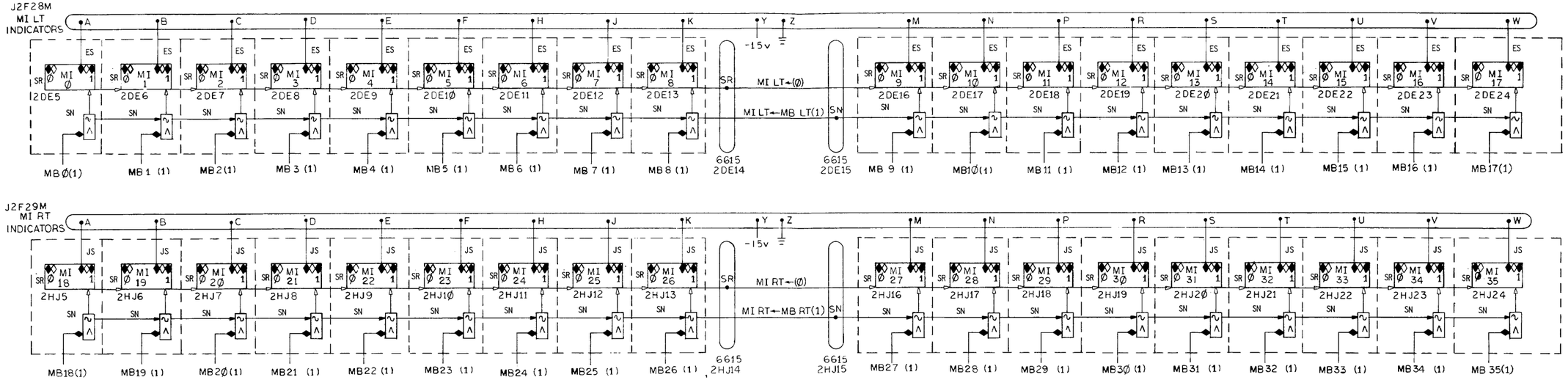


NOTE:

ALL TERMINATOR RES. IN THE
1Ø3Ø MODULES ARE 1ØØ OHMS.

MB - Memory Buffer Interface
(BS-D-166-0-MB-4)

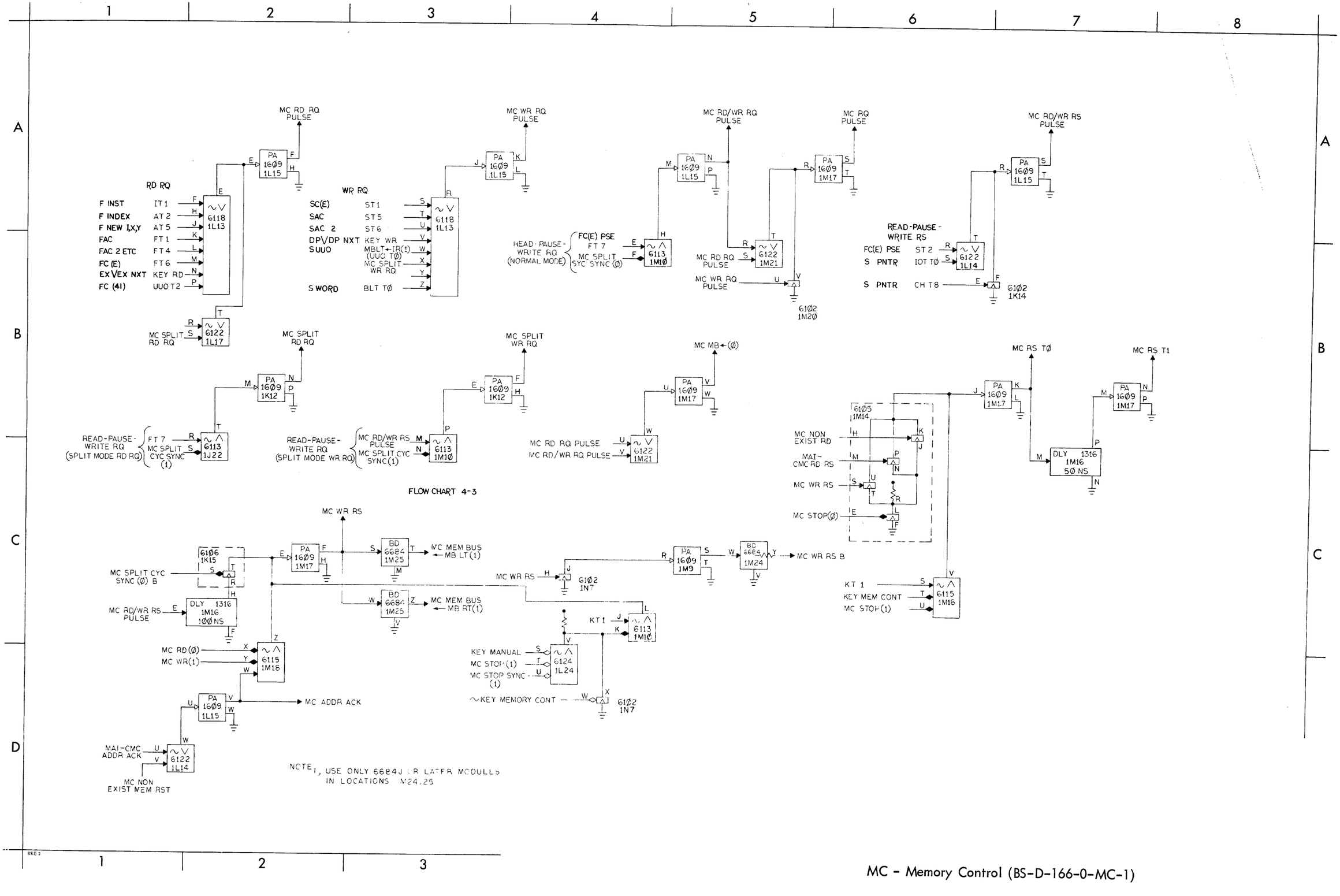
MI - Memory Indicators (BS-D-166-0-MI)



NOTE:

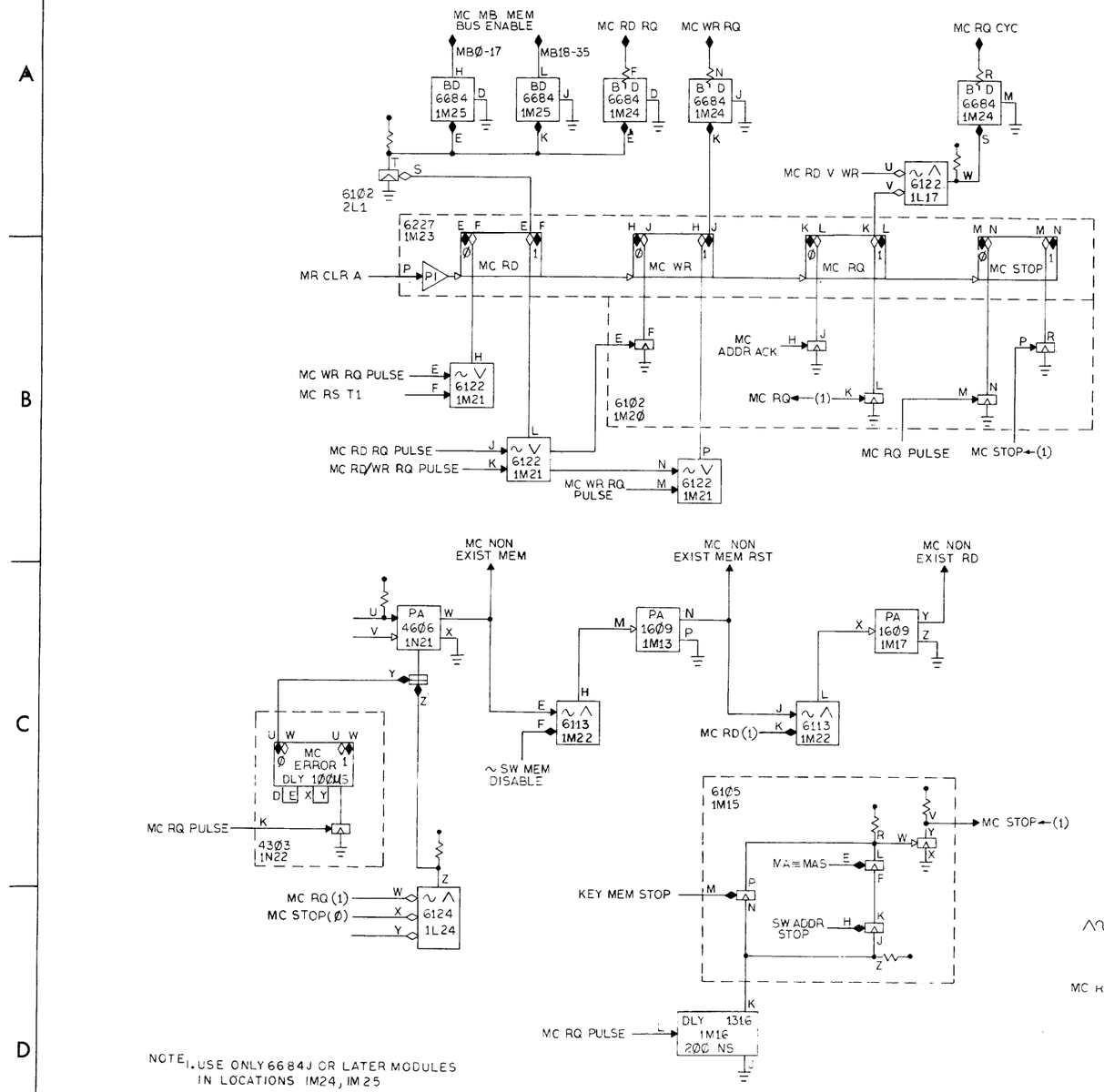
1. ALL FF PACKAGES ARE 62Ø5
2. SN AND SR ARE PINS IN LOWER REAR CONNECTOR

MC - Memory Control (BS-D-166-0-MC-1)



MC - Memory Control (BS-D-166-0-MC-1)

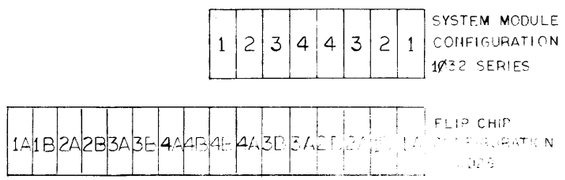
MC - Memory Control (BS-D-166-0-MC-2)


MC MB MEM BUS ENABLE
MB Ø-17
MB 18-35
MC RD RQ
MC WR RQ
MC RQ CYC
BD 6684 1M25
BD 6684 1M24
6101 2L1
MC RD V WR
6122 1L17
6227 1M23
MC RD
MC WR
MC RQ
MC STOP
MR CLR A
PI
MC WR RQ PULSE
MC RS T1
6122 1M21
MC ADDR ACK
MC RQ ← (1)
6102 1M20
MC RD RQ PULSE
MC RD/WR RQ PULSE
MC WR RQ PULSE
MC RQ PULSE
MC STOP ← (1)
MC NON EXIST MEM
MC NON EXIST MEM RST
MC NON EXIST RD
PA 4606 1N21
PA 1609 1M13
PA 1609 1M17
MC ERROR DLY 100MS
6113 1M22
~ SW MEM DISABLE
MC RQ (1)
MC STOP(Ø)
6124 1L24
MC RQ PULSE
4303 1N22
6105 1M15
MA ≠ MAS
KEY MEM STOP
SW ADDR STOP
DLY 1316 1M16 200 NS
MC RQ PULSE
NOTE 1. USE ONLY 6684J OR LATER MODULES IN LOCATIONS 1M24, 1M25
MC SPLIT CYC SYNC (Ø) B
6102 1J13
6227 1J15
MC STOP SYNC
MC SPLIT CYC SYNC
MR CLR A
MC RD/WR RQ PULSE
6102 1J10
MC RD V WR
MC RD(1)
MC WR(1)
6122 1L14
FLOW CHART 4-3
MC SW STOP
AT 4
6115 1L22
KEY MEM STOP
SW ADDR STOP
MC DR SPLIT
6115 1K22
ON
NO
NC
OFF
SPLIT CYCLE OVERRIDE SWITCH
6105 1M14
2L13
MC DR SPLIT
MC ILLEG ADDRESS
MC RQ ← (1)
PA 1609 2M2
PA 1609 1N15
6105 1N20
PR REL MA OK
6106 1L20
EX INH REL
DLY 1316 1L16 150 NS
DLY 1316 1L16 50 NS
MC RQ PULSE

In-out and Memory Busses (CD-D-166-0-10MB)

IO BUS(§8.4)

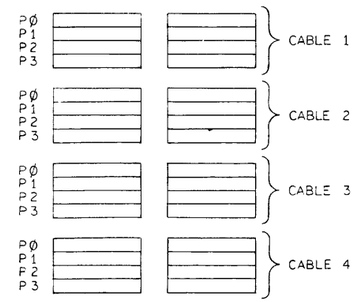
FLIP CHIP PIN	SYSTEM MODULE PIN	IO CABLE 1	IO CABLE 2	IO CABLE 3	IO CABLE 4
C	A	GND	GND	GND	GND
D	B	IOB Ø (1) —◇	IOB 18 (1) —◇	IOB RESET →	DATAO CLEAR →
E	C	IOB 1 (1) —◇	IOB 19 (1) —◇	POWER ON -15V	DATAO SET →
H	D	IOB 2 (1) —◇	IOB 2Ø (1) —◇		CONO CLEAR →
K	E	IOB 3 (1) —◇	IOB 21 (1) —◇	MC DR SPLIT —◇	CONO SET →
M	F	IOB 4 (1) —◇	IOB 22 (1) —◇	IOS 3 (Ø) —◇	IOB ← DATAI —●
P	H	IOB 5 (1) —◇	IOB 23 (1) —◇	IOS 3 (1) —◇	IOB ← STATUS —●
N	J	GND	GND	GND	GND
S	K	IOB 6 (1) —◇	IOB 24 (1) —◇	IOS 4 (Ø) —◇	
T	L	IOB 7 (1) —◇	IOB 25 (1) —◇	IOS 4 (1) —◇	
V	M	IOB 8 (1) —◇	IOB 26 (1) —◇	IOS 5 (Ø) —◇	
D	N	IOB 9 (1) —◇	IOB 27 (1) —◇	IOS 5 (1) —◇	
E	P	IOB 1Ø (1) —◇	IOB 28 (1) —◇	IOS 6 (Ø) —◇	
H	R	IOB 11 (1) —◇	IOB 29 (1) —◇	IOS 6 (1) —◇	PI REQ 1 —◇
J	S	GND	GND	GND	GND
K	T	IOB 12 (1) —◇	IOB 3Ø (1) —◇	IOS 7 (Ø) —◇	PI REQ 2 —◇
M	U	IOB 13 (1) —◇	IOB 31 (1) —◇	IOS 7 (1) —◇	PI REQ 3 —◇
P	V	IOB 14 (1) —◇	IOB 32 (1) —◇	IOS 8 (Ø) —◇	PI REQ 4 —◇
S	W	IOB 15 (1) —◇	IOB 33 (1) —◇	IOS 8 (1) —◇	PI REQ 5 —◇
T	X	IOB 16 (1) —◇	IOB 34 (1) —◇	IOS 9 (Ø) —◇	PI REQ 6 —◇
V	Y	IOB 17 (1) —◇	IOB 35 (1) —◇	IOS 9 (1) —◇	PI REQ 7 —◇
U	Z	GND	GND	GND	GND
		(SOURCE 3L6)	(SOURCE 3L7)	(SOURCE 2L13)	(SOURCE 2N25)



NOTE: 1 SYSTEM MODULE CONNECTOR (1Ø32) = 2 FLIP CHIP CONNECTORS (W028)

MEM BUS(§§7.1a,7.2)

FLIP CHIP PIN	SYSTEM MODULE PIN	MEM CABLE 1	MEM CABLE 2	MEM CABLE 3	MEM CABLE 4
C	A	GND	GND	GND	GND
D	B	ADDR ACK →	MA 22 (1) —●	MB Ø (1) →	MB 18 (1) →
E	C	RD RS →	MA 23 (1) —●	MB 1 (1) →	MB 19 (1) →
H	D	WR RS →	MA 24 (1) —●	MB 2 (1) →	MB 2Ø (1) →
K	E	PAR (1) →	MA 25 (1) —●	MB 3 (1) →	MB 21 (1) →
M	F	RQ CYCLE —●	MA 26 (1) —●	MB 4 (1) →	MB 22 (1) →
P	H	SPARE	MA 27 (1) —●	MB 5 (1) →	MB 23 (1) →
N	J	GND	GND	GND	GND
S	K	MA 18 (1) —◇	MA 28 (1) —●	MB 6 (1) →	MB 24 (1) →
T	L	MA 18 (Ø) —◇	MA 29 (1) —●	MB 7 (1) →	MB 25 (1) →
V	M	MA 19 (1) —◇	MA 3Ø (1) —●	MB 8 (1) →	MB 26 (1) →
D	N	MA 19 (Ø) —◇	MA 31 (1) —●	MB 9 (1) →	MB 27 (1) →
E	P	MA 2Ø (1) —◇	MA 32 (1) —●	MB 1Ø (1) →	MB 28 (1) →
H	R	MA 2Ø (Ø) —◇	MA 33 (1) —●	MB 11 (1) →	MB 29 (1) →
J	S	GND	GND	GND	GND
K	T	MA 21 (1) —◇	MA 34 (1) —●	MB 12 (1) →	MB 30 (1) →
M	U	MA 21 (Ø) —◇	MA 35 (1) —●	MB 13 (1) →	MB 31 (1) →
P	V	MA 35 (1) —◇	MC RD RQ —●	MB 14 (1) →	MB 32 (1) →
S	W	MA 35 (Ø) —◇	MC WR RQ —●	MB 15 (1) →	MB 33 (1) →
T	X	MAFNC } —◇	PAR OPTION →	MB 16 (1) →	MB 34 (1) →
V	Y	SELECT B } —●	MA 21 (1) —●	MB 17 (1) →	MB 35 (1) →
U	Z	GND	GND	GND	GND
		(SOURCE 2L5)	(SOURCE 2L2Ø)	(SOURCE 2E25)	(SOURCE 2J25)

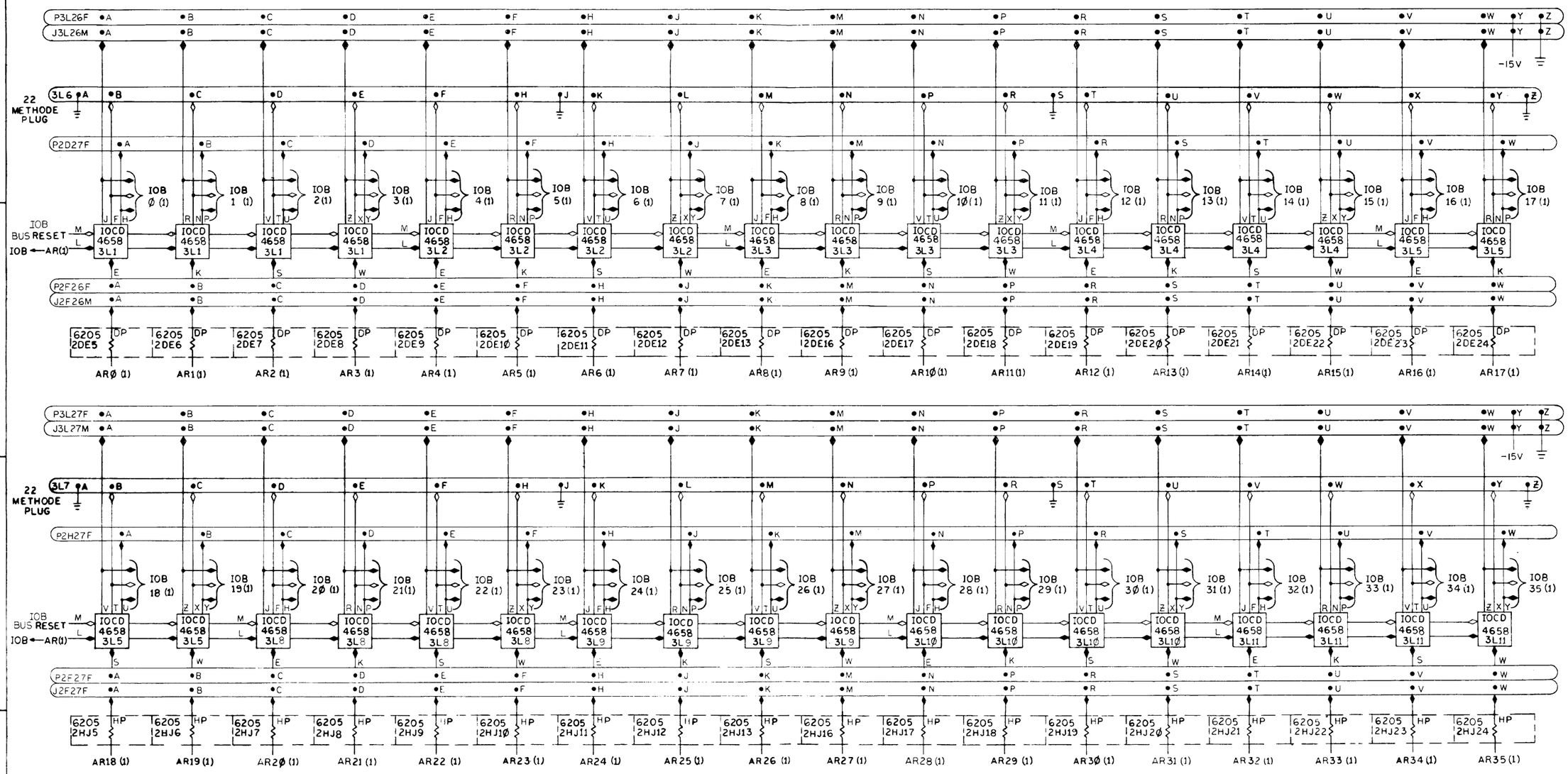


In-out and Memory Busses (CD-D-166-0-10MB)

IOT, IOB Control (BS-D-166-0-IOTB)

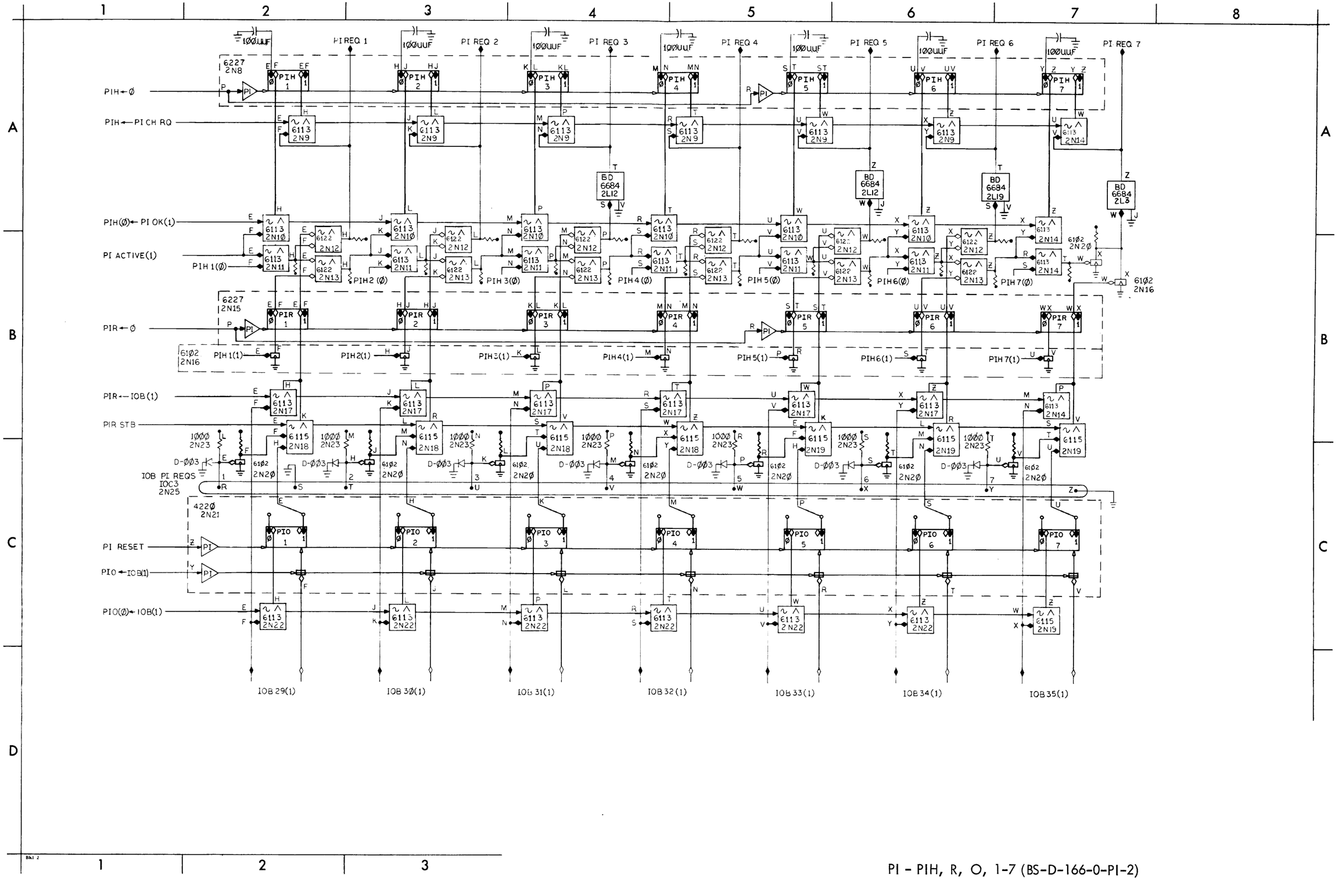
IOT, IOB Control (BS-D-166-0-1OTB)

10B - 10 Bus (BS-D-166-0-10B)



IOB - IO Bus (BS-D-166-0-IOB)

PI - PIH, R, O, 1-7 (BS-D-166-0-PI-2)

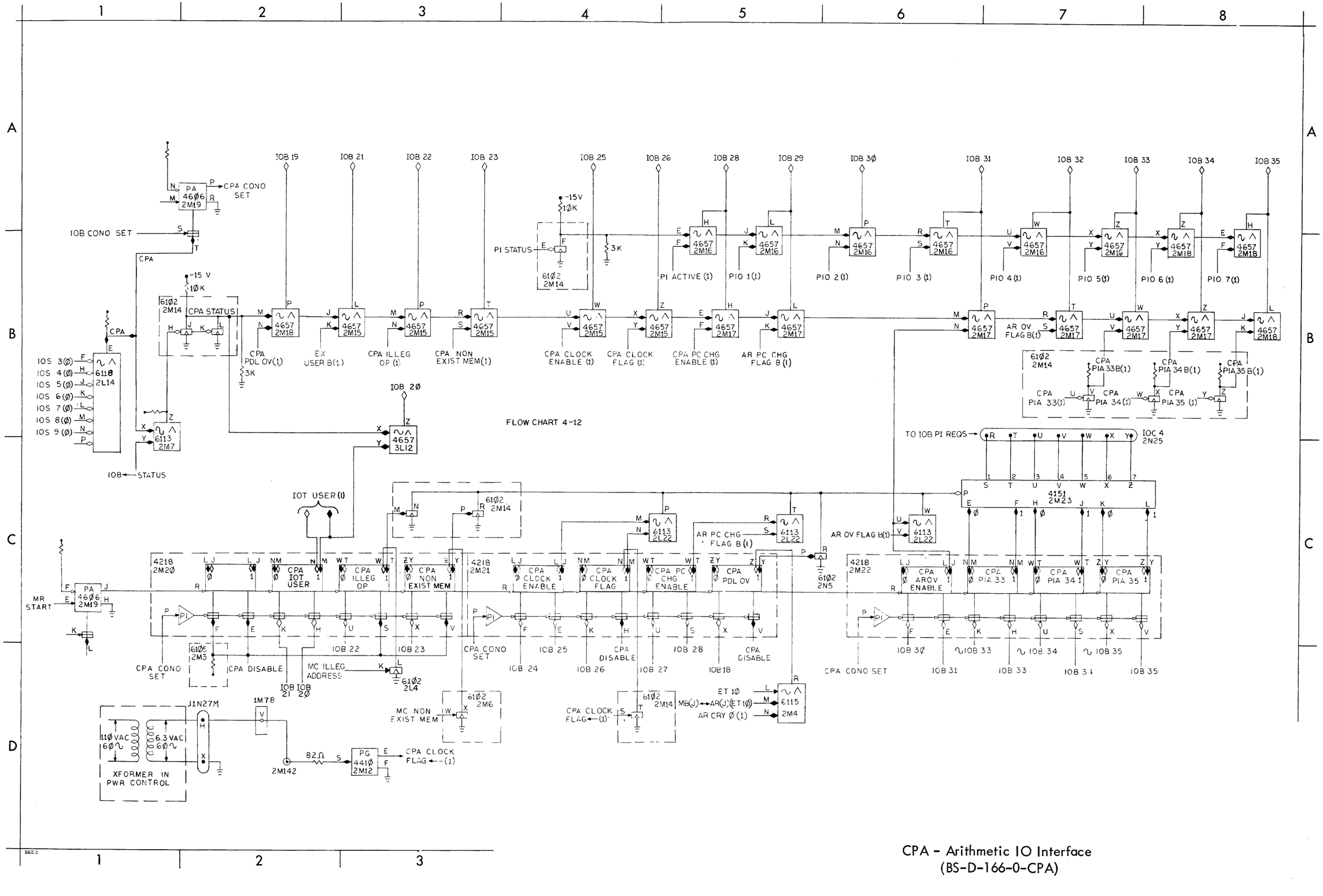


PI - PIH, R, O, 1-7 (BS-D-166-0-PI-2)

PI Control (BS-D-166-0-PI-1)

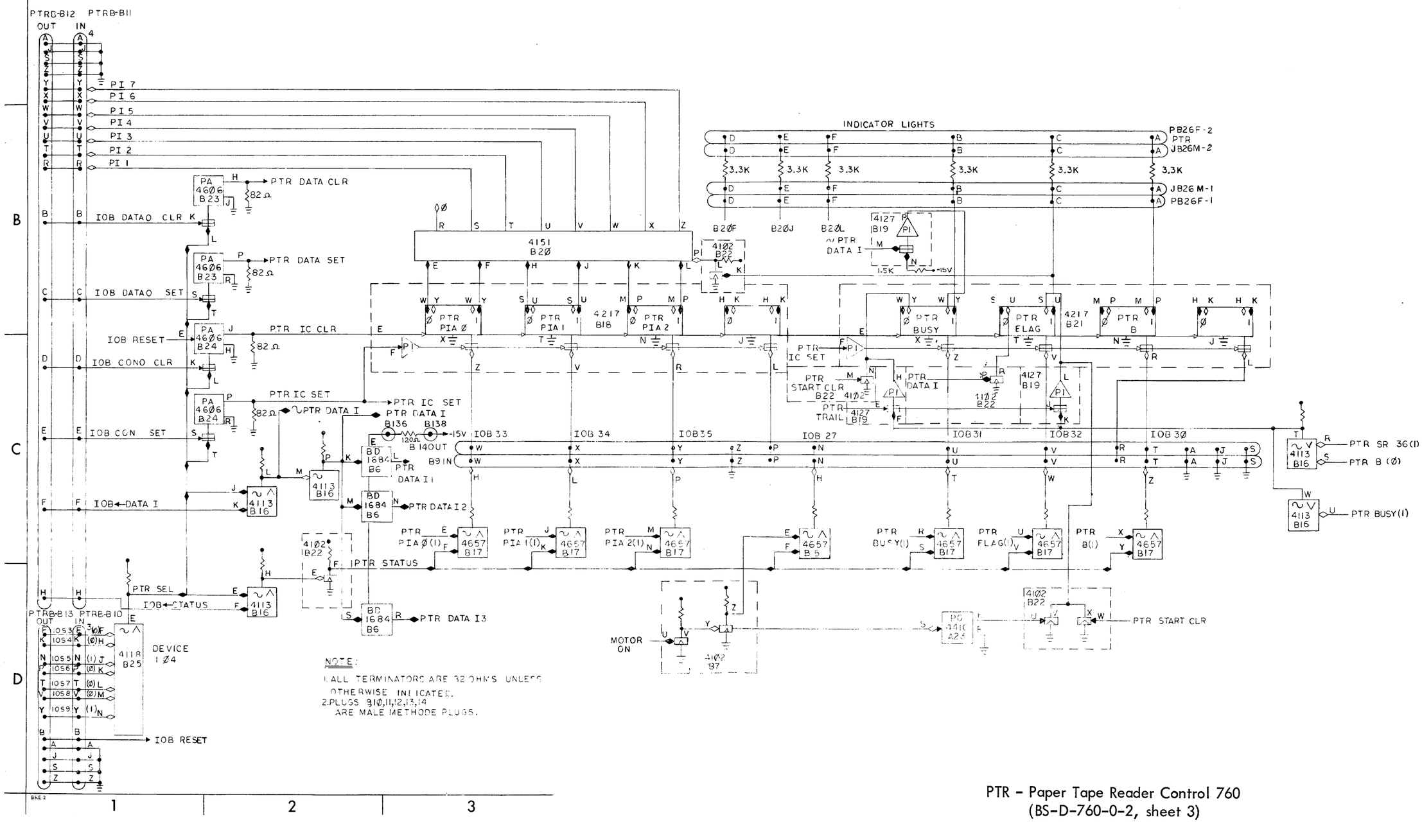
PI Control (BS-D-166-0-PI-1)

CPA - Arithmetic IO Interface
(BS-D-166-0-CPA)



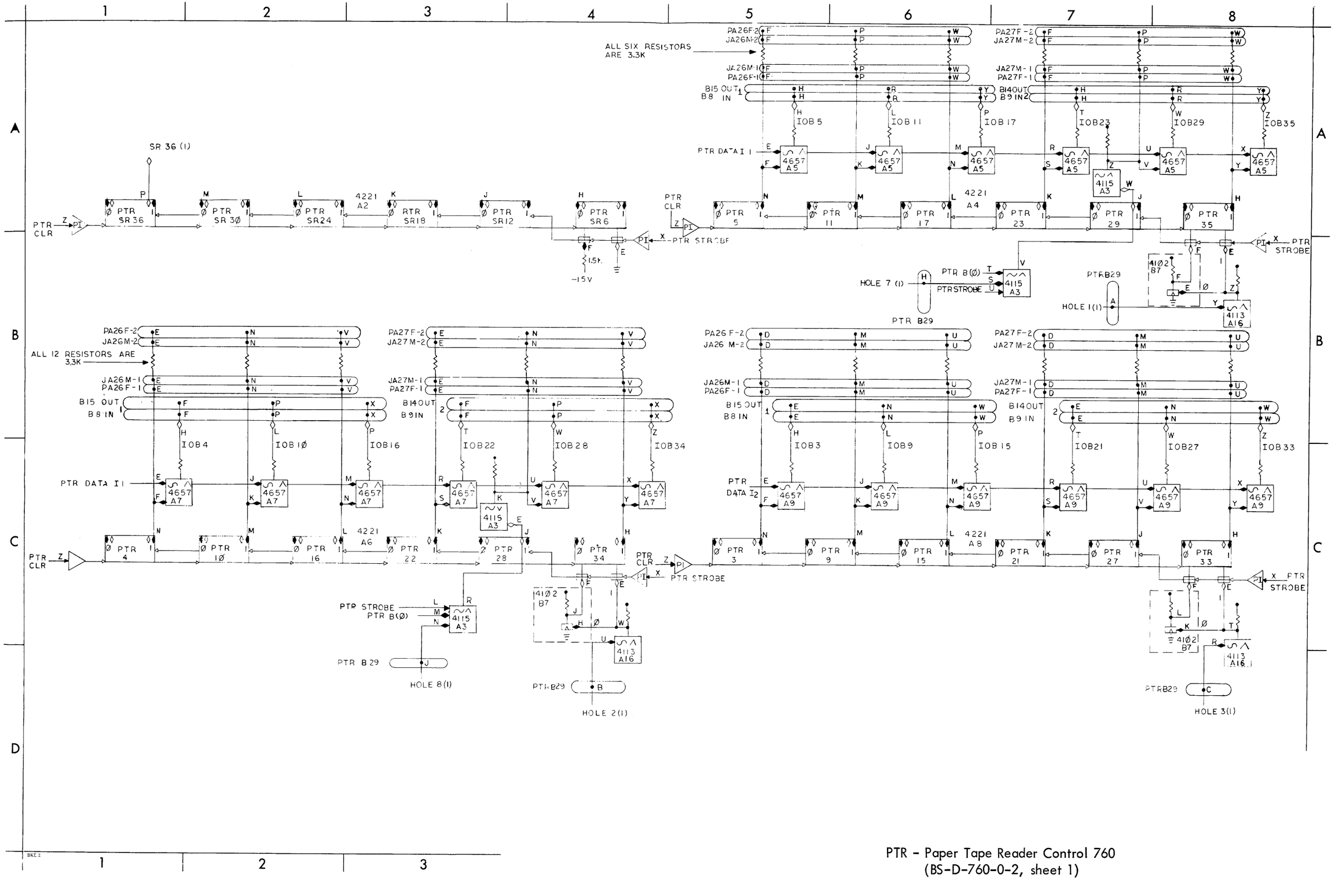
CPA - Arithmetic IO Interface
(BS-D-166-0-CPA)

PTR - Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 3)



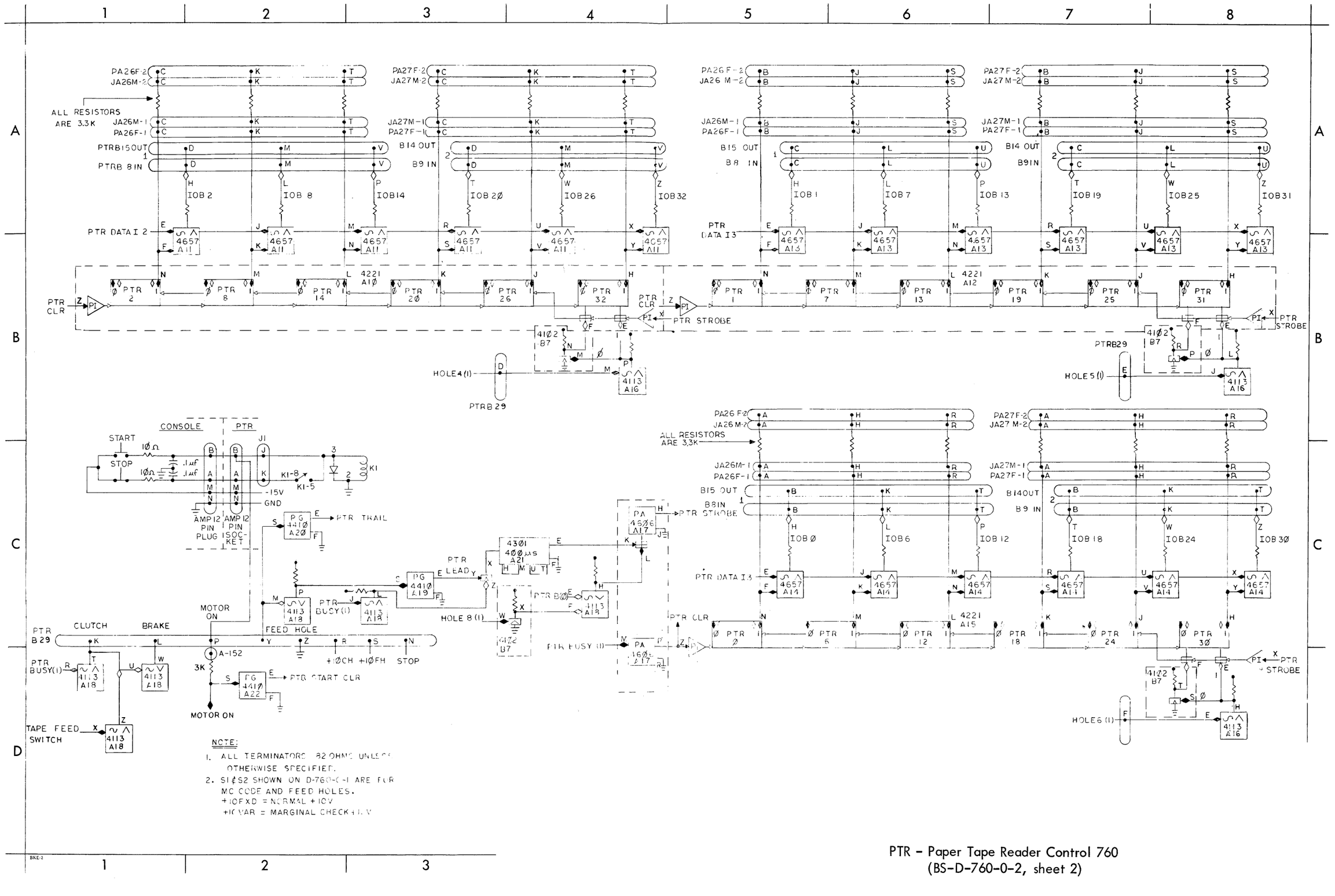
PTR - Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 3)

PTR - Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 1)



PTR - Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 1)

PTR - Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 2)

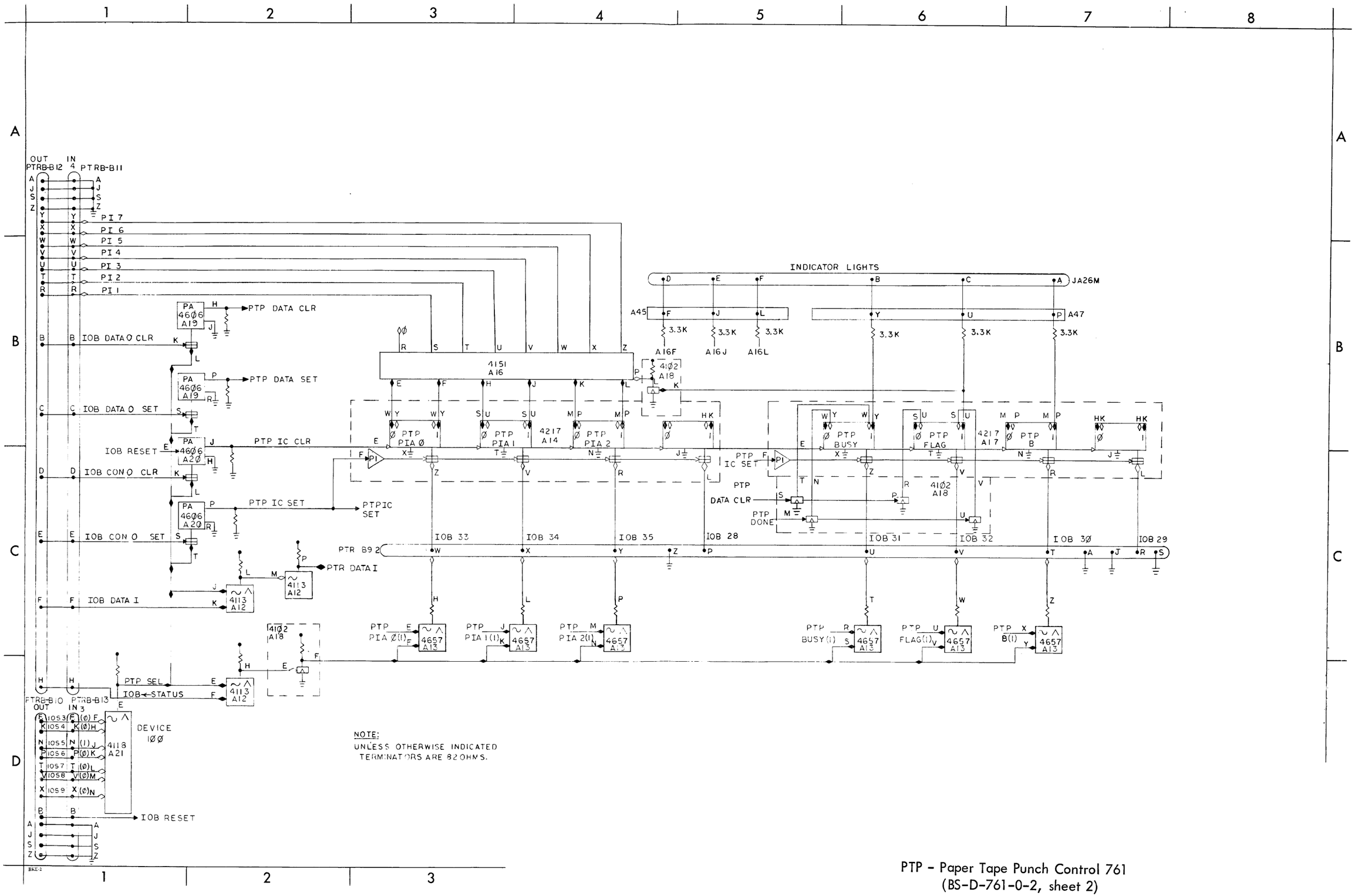


NOTE:

1. ALL TERMINATORS 82 OHMS UNLESS OTHERWISE SPECIFIED.
2. SIGS2 SHOWN ON D-760-0-1 ARE FOR MC CODE AND FEED HOLES.
   +IOFXD = NORMAL +10V
   +IOFVAR = MARGINAL CHECK +1.V

PTR – Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 2)

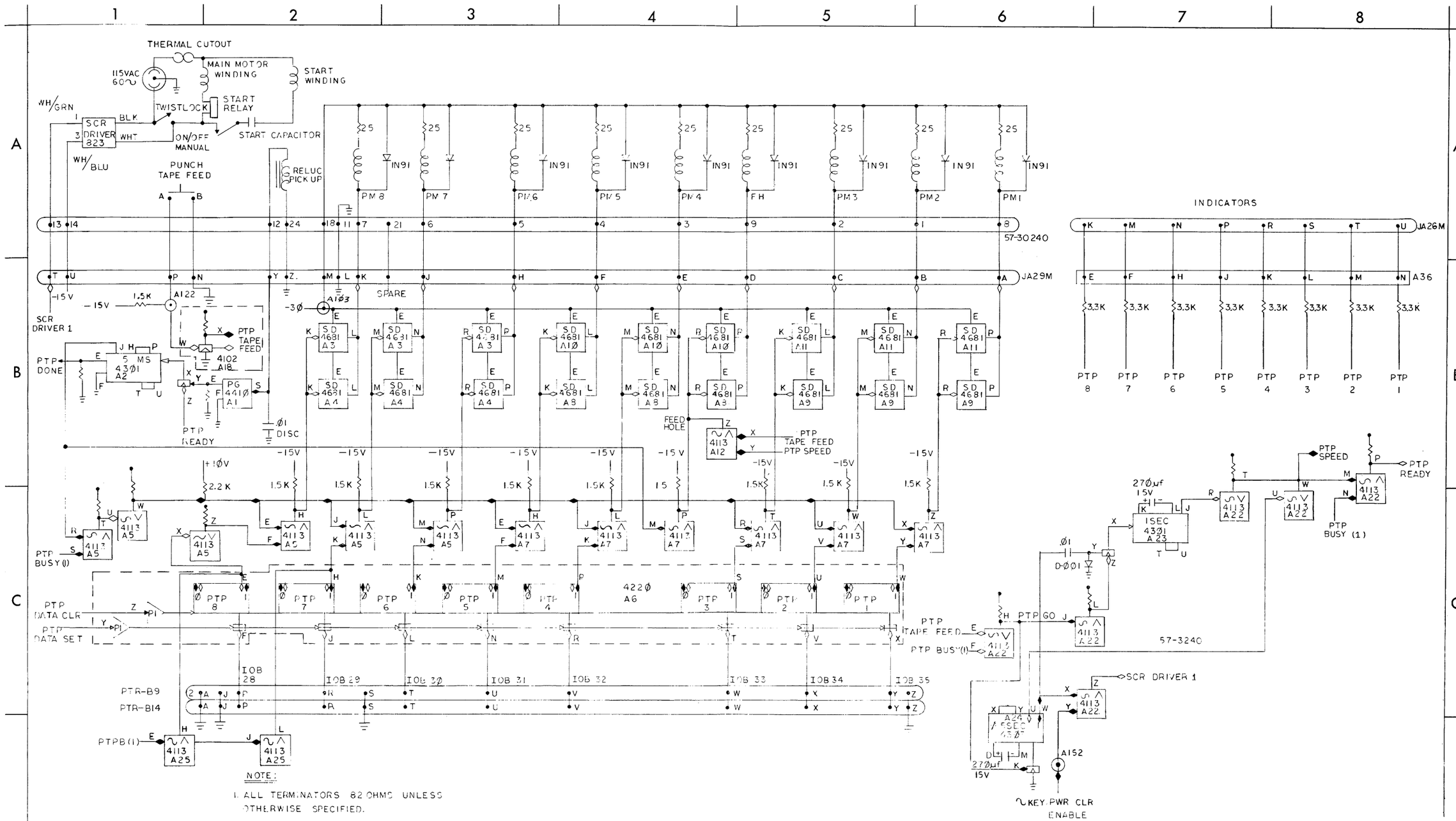
PTP - Paper Tape Punch Control 761
(BS-D-761-0-2, sheet 2)



NOTE:
UNLESS OTHERWISE INDICATED
TERMINATORS ARE 82 OHMS.

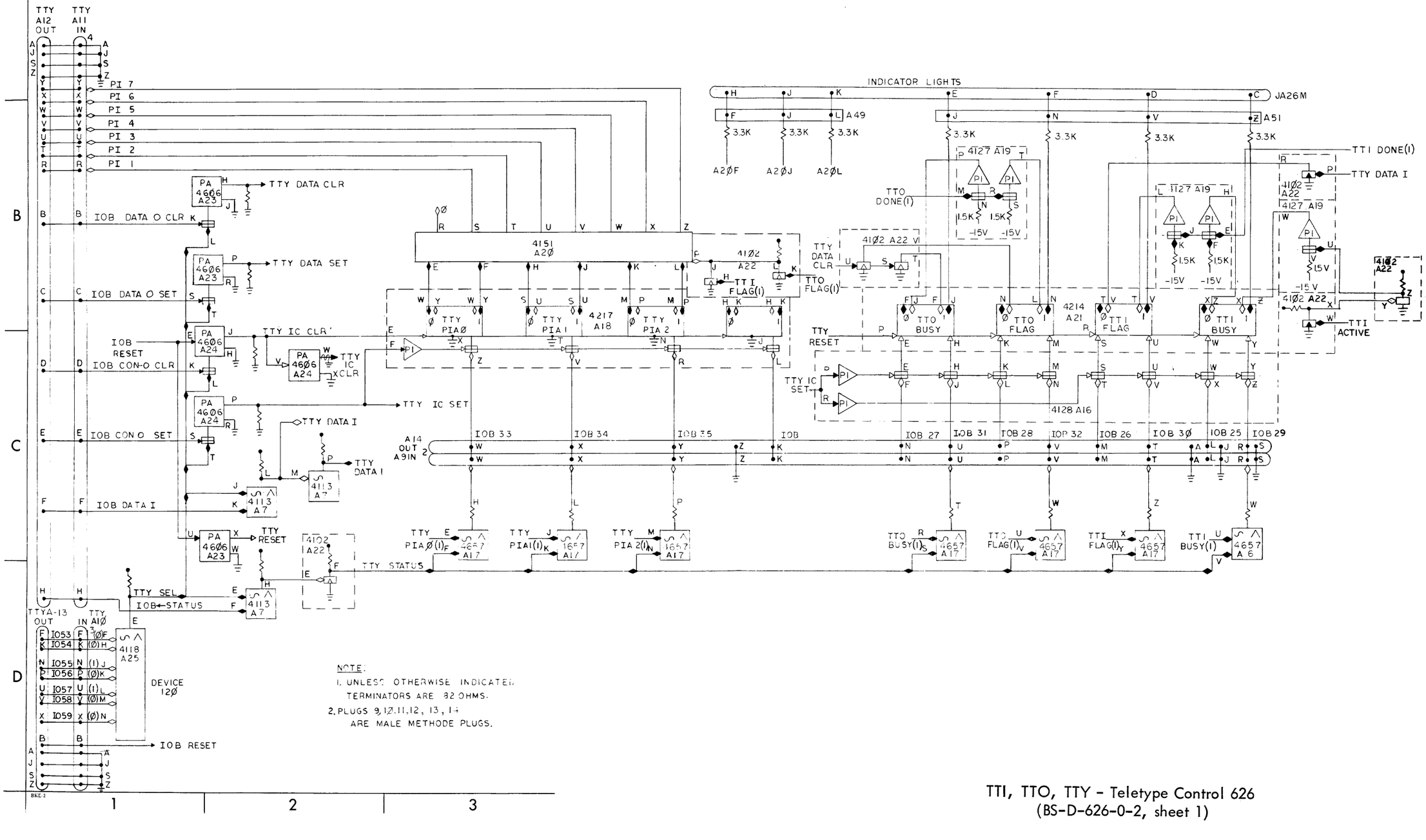
PTP - Paper Tape Punch Control 761
(BS-D-761-0-2, sheet 2)

PTP - Paper Tape Punch Control 761
(BS-D-761-0-2, sheet 1)



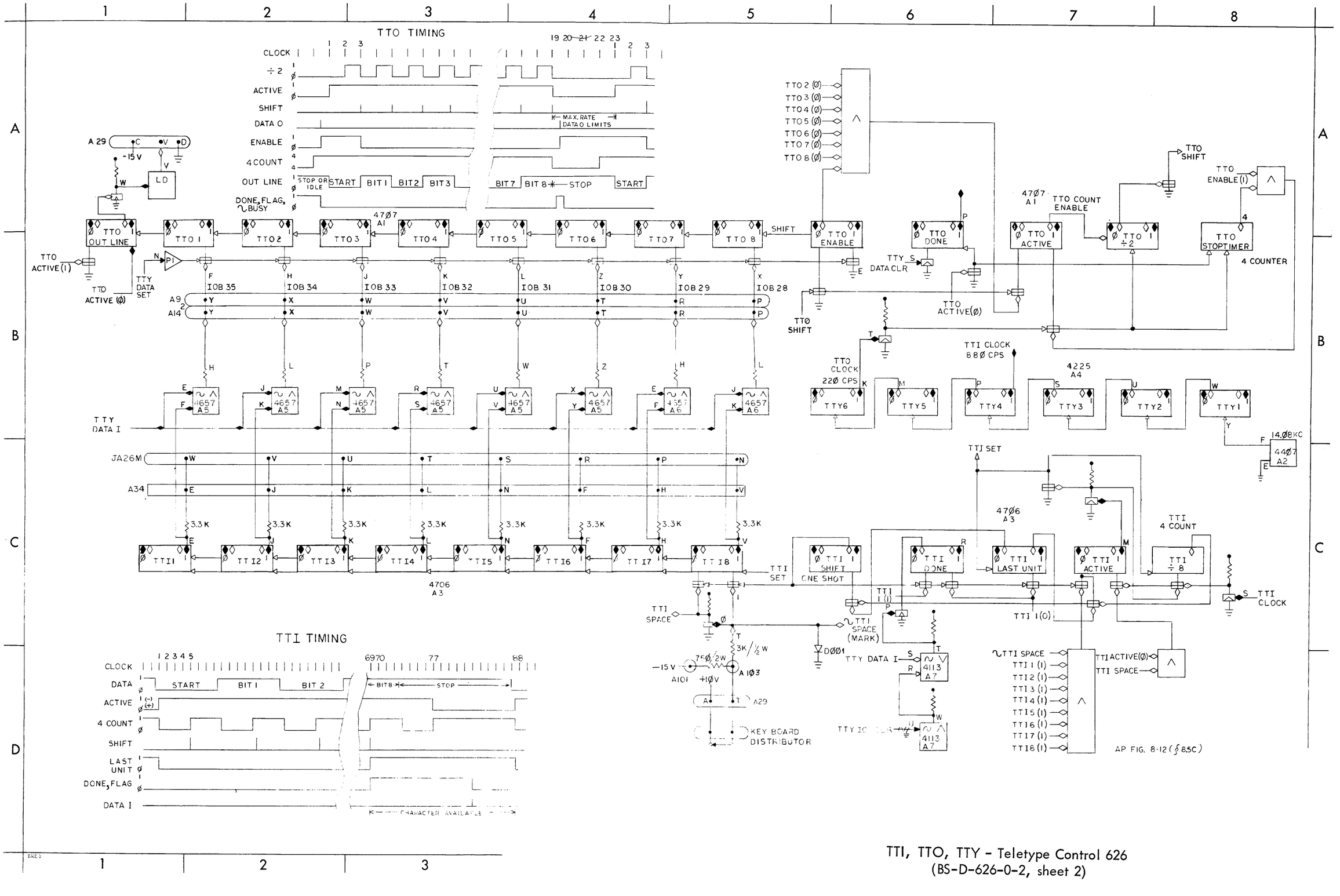
PTP - Paper Tape Punch Control 761
(BS-D-761-0-2, sheet 1)

TTI, TTO, TTY - Teletype Control 626
(BS-D-626-0-2, sheet 1)



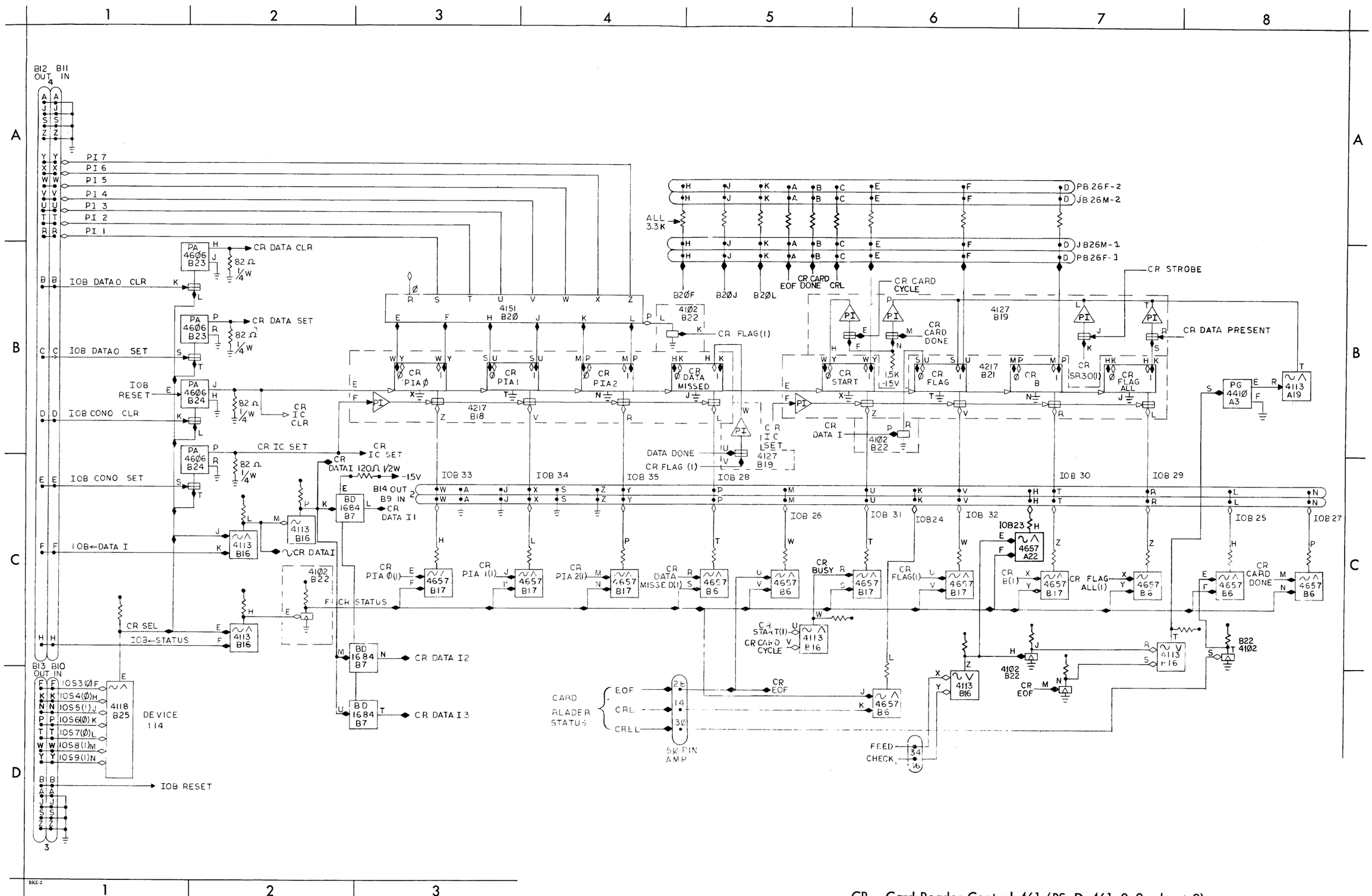
TTI, TTO, TTY - Teletype Control 626
(BS-D-626-0-2, sheet 1)

TTI, TTO, TTY - Teletype Control 626
(BS-D-626-0-2, sheet 2)



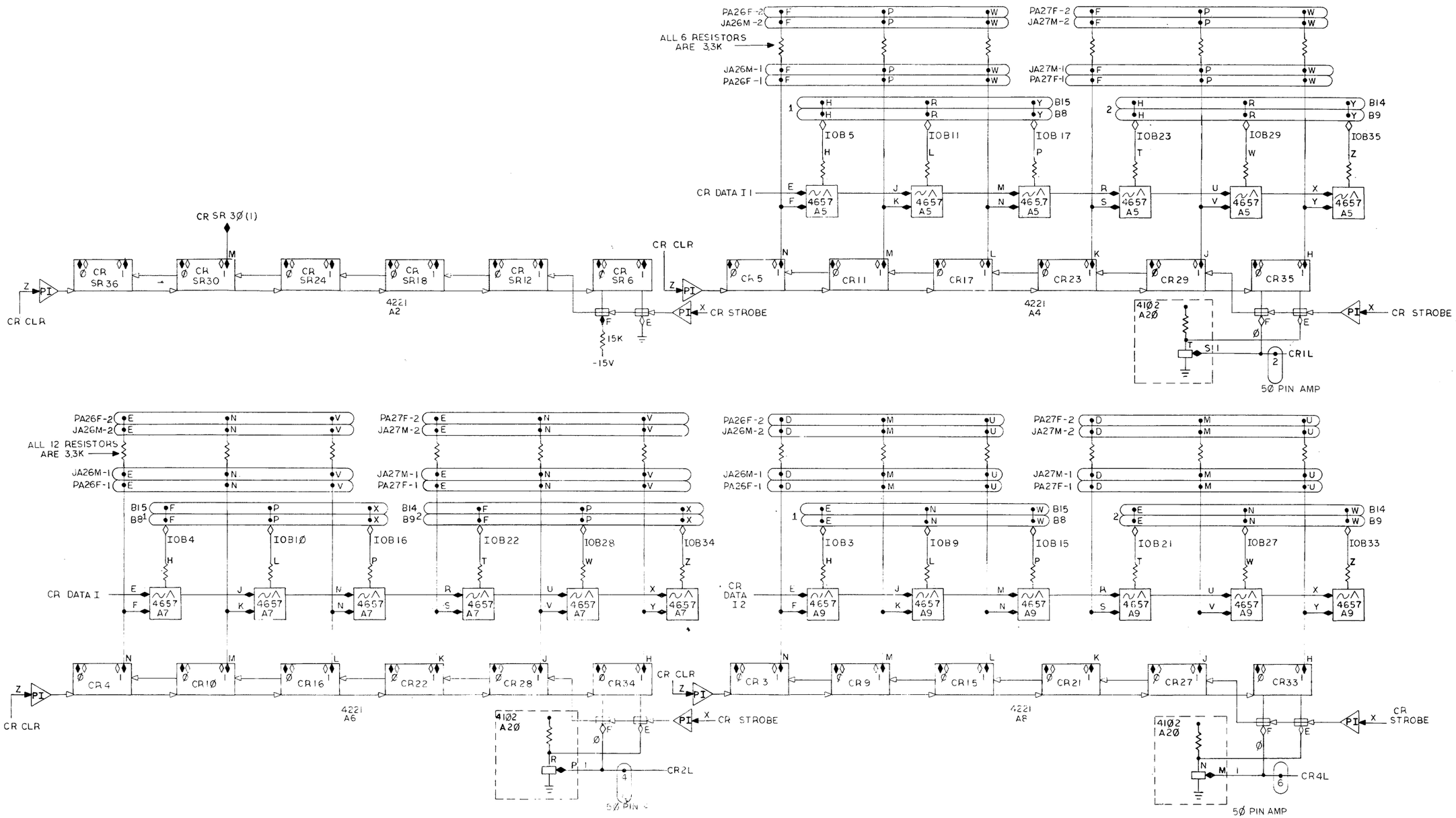
TTI, TTO, TTY - Teletype Control 626
(BS-D-626-0-2, sheet 2)

CR - Card Reader Control 461 (BS-D-461-0-2, sheet 3)



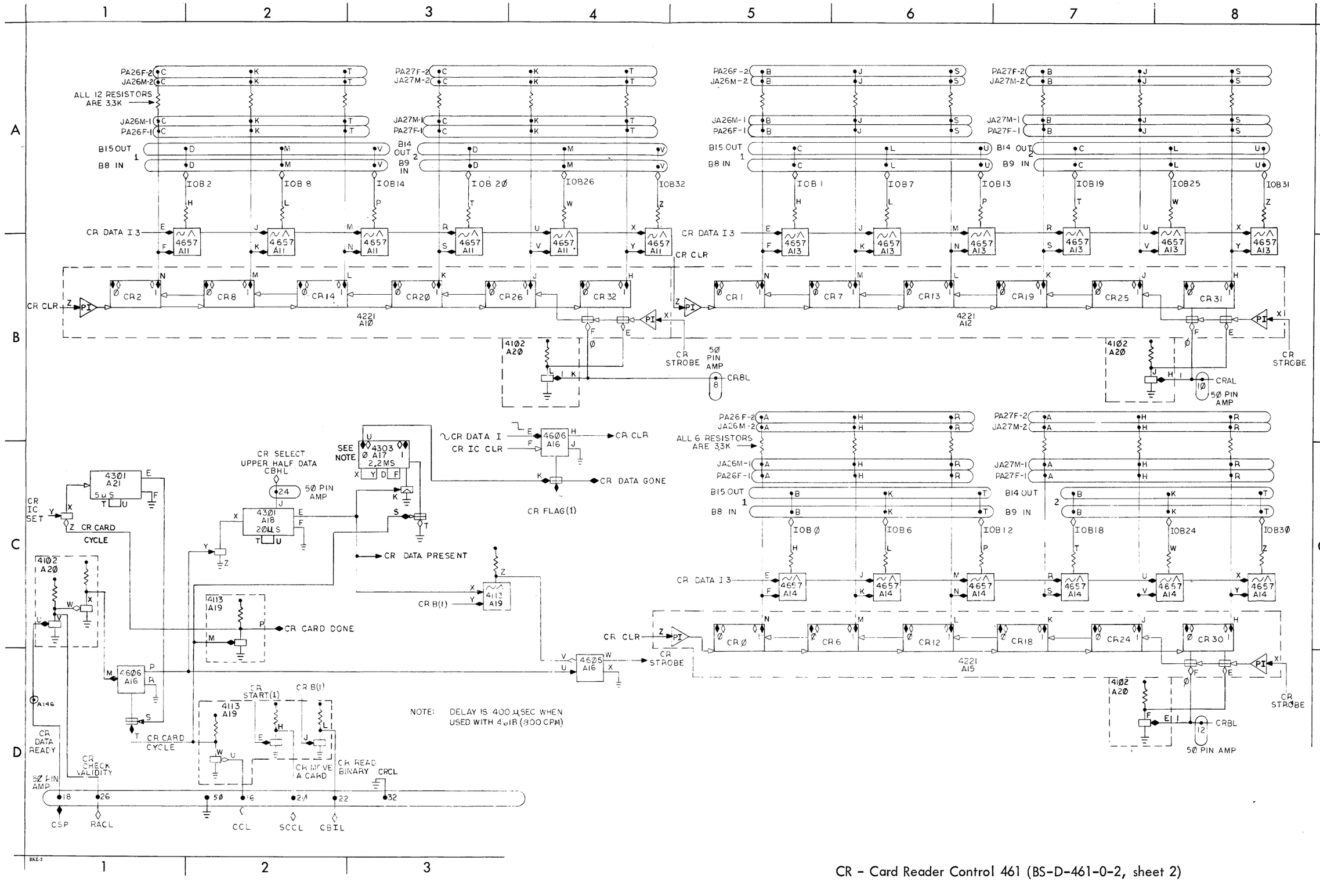
CR - Card Reader Control 461 (BS-D-461-0-2, sheet 3)

CR - Card Reader Control 461 (BS-D-461-0-2, sheet 1)



CR - Card Reader Control 461 (BS-D-461-0-2, sheet 1)

CR - Card Reader Control 461 (BS-D-461-0-2, sheet 2)



CR – Card Reader Control 461 (BS-D-461-0-2, sheet 2)

digital
EQUIPMENT
CORPORATION
MAYNARD, MASSACHUSETTS

Cambridge, Mass. • Washington, D. C. • Parsippany, N.J. • Rochester, N.Y. • Los Angeles
Palo Alto • Chicago • Ann Arbor • Pittsburgh
Denver • Huntsville • Orlando • Carleton Place
and Toronto, Ont. • Reading, England • Paris,
France • Munich, Germany • Sydney, Australia