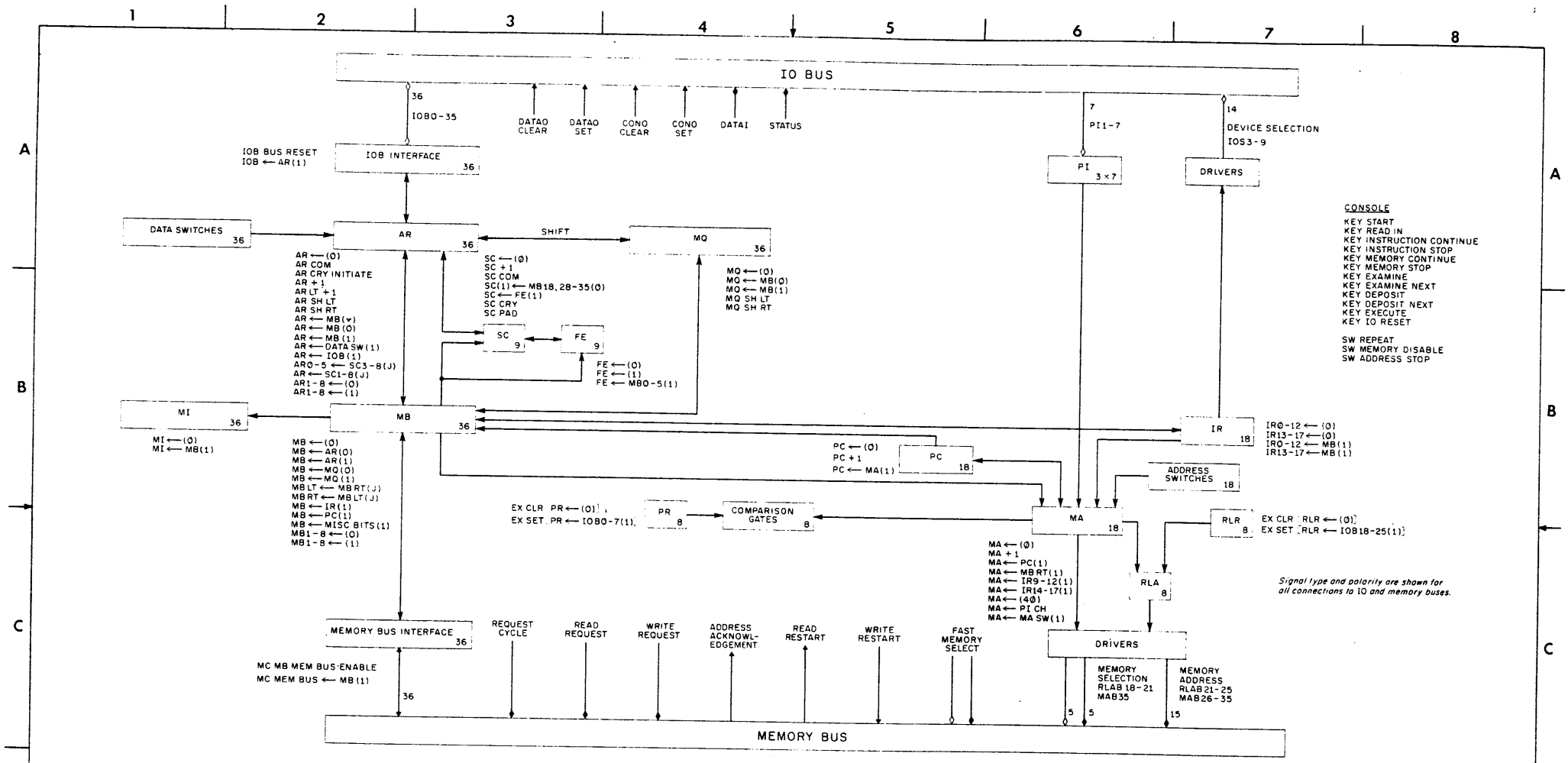
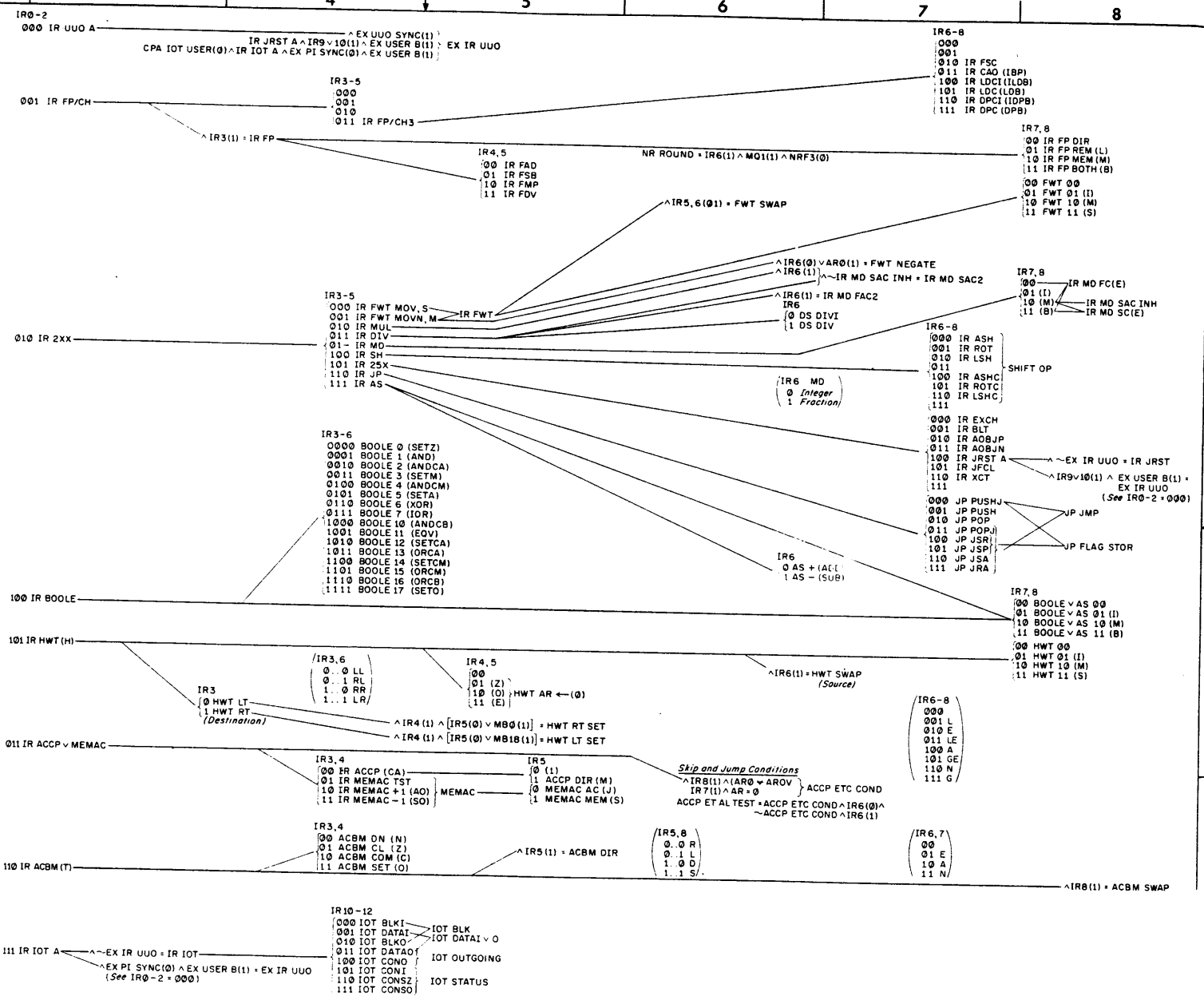


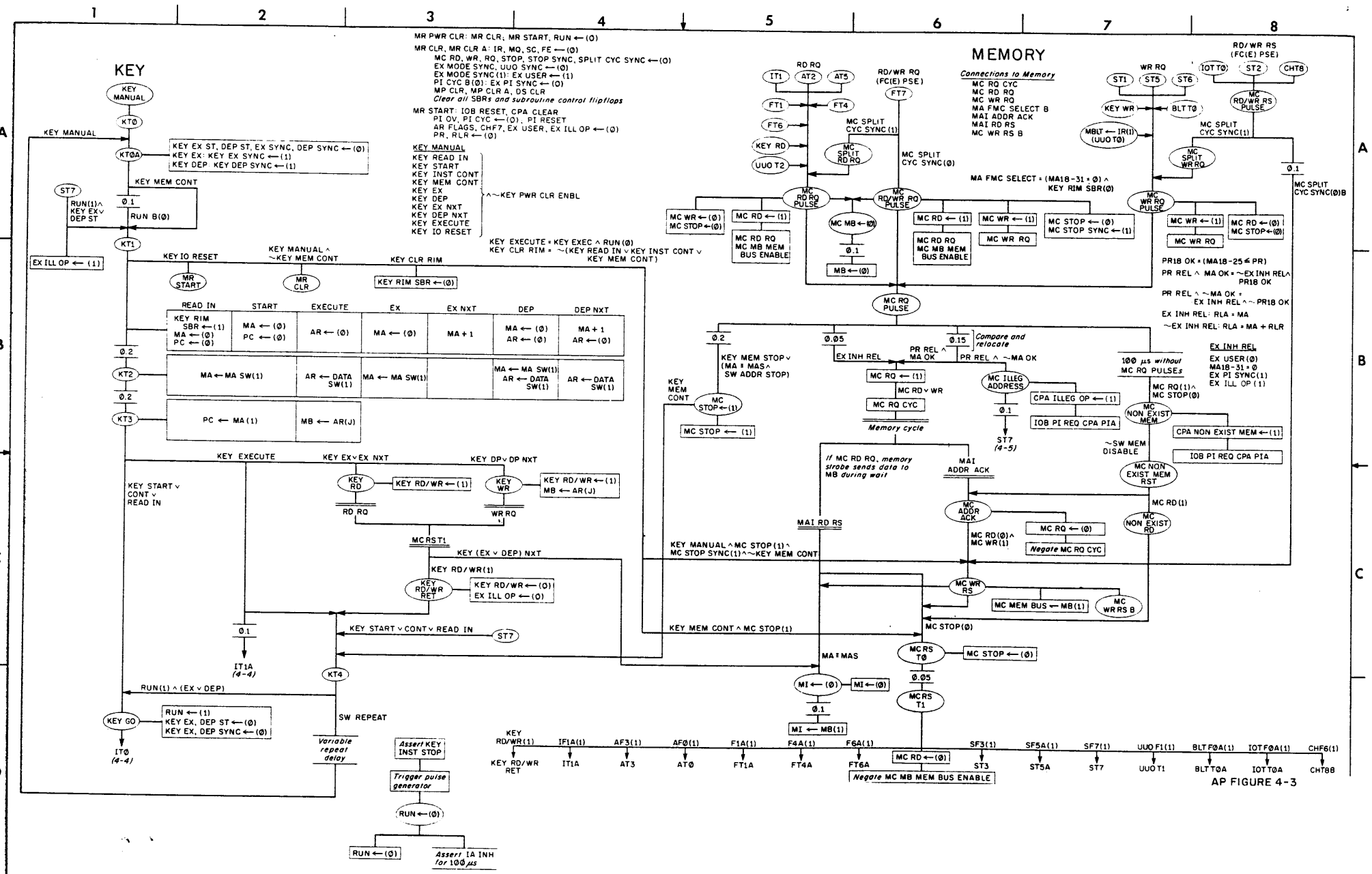
System Block Diagram
(SD-D-166-0-BD)

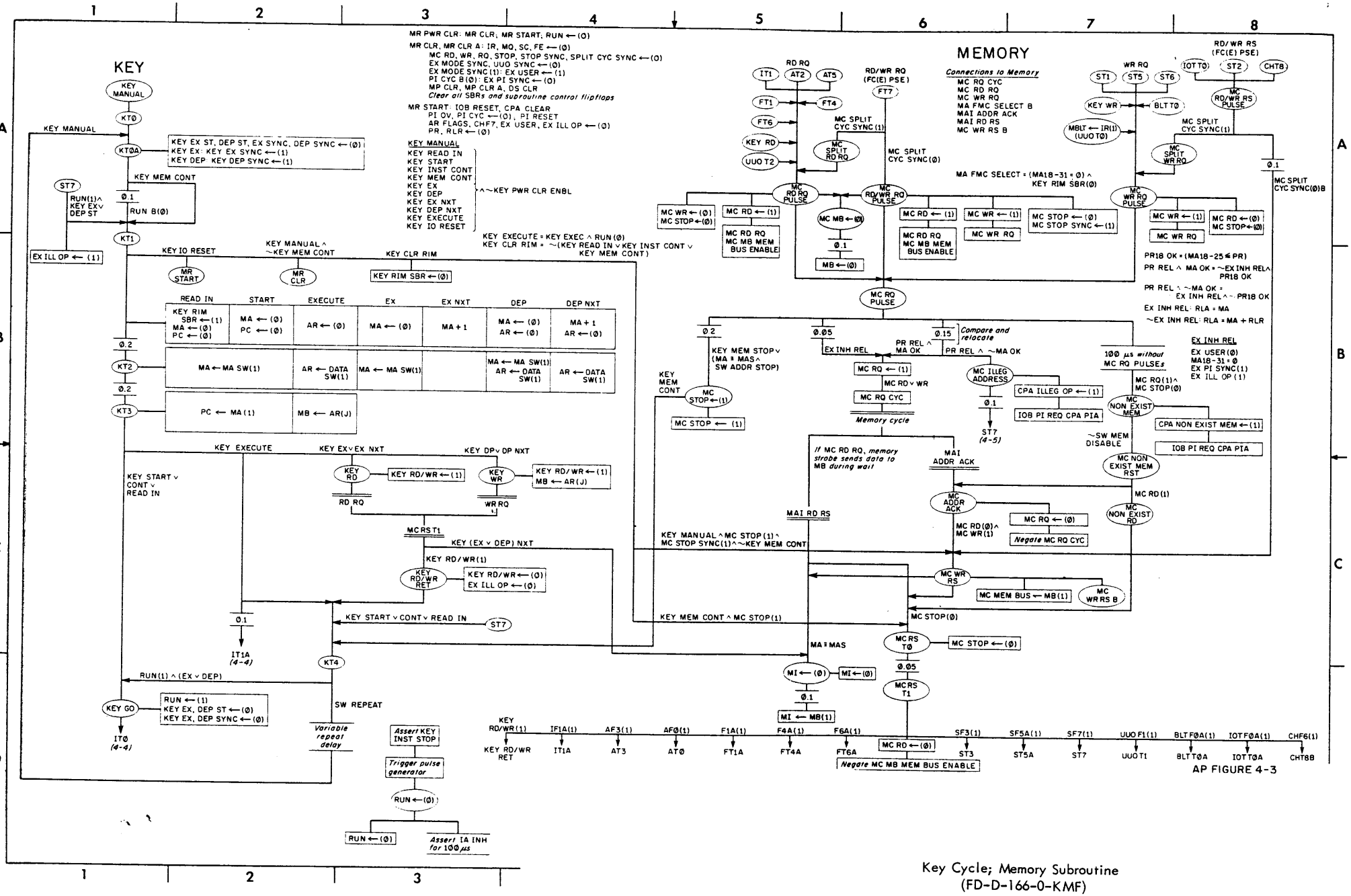
LOGIC PREFIXES

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



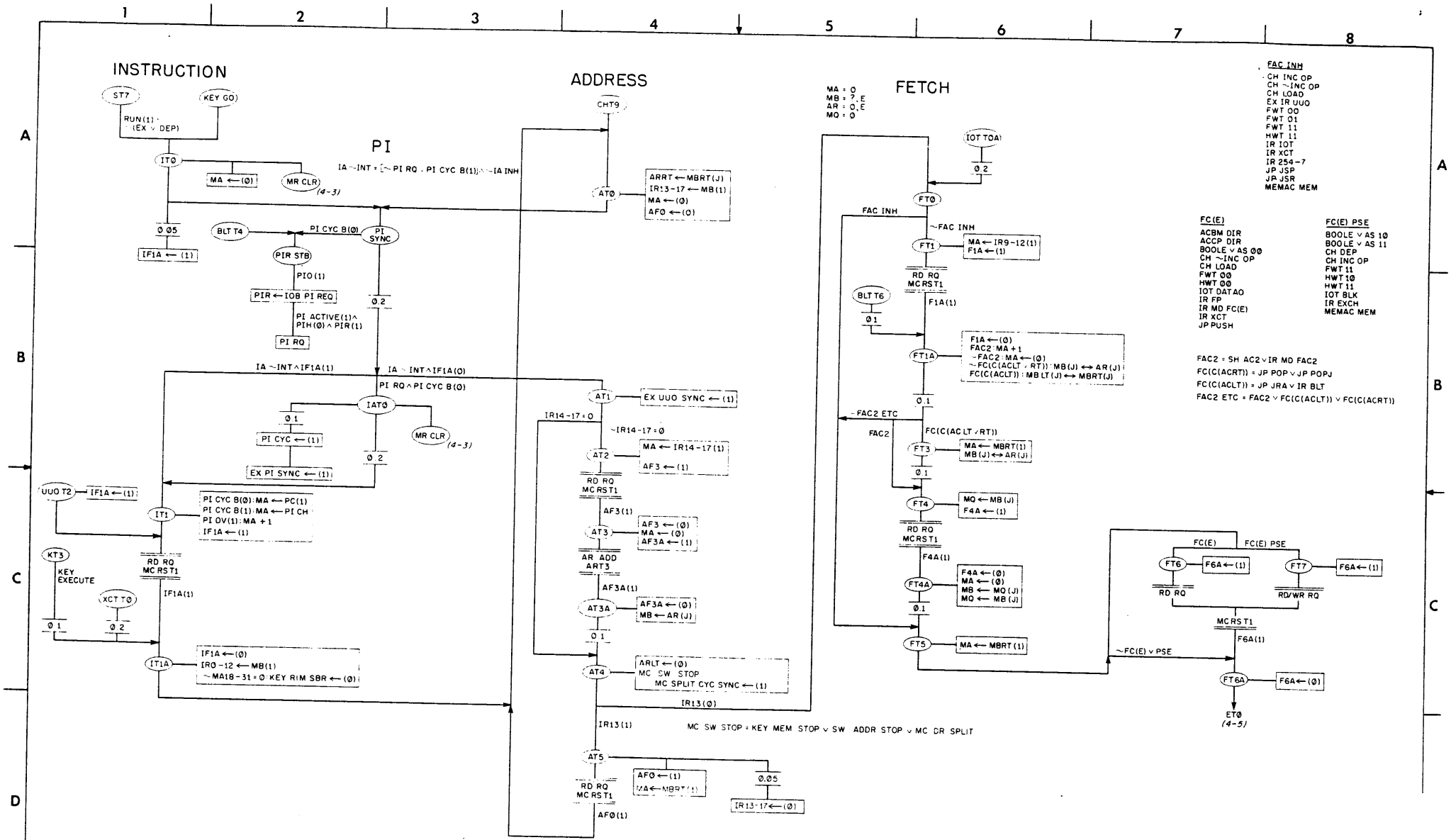
Instruction Decoding
(FD-D-166-0-IRD)


KEY
MEMORY
MR PWR CLR: MR CLR; MR START, RUN ← (0)
MR CLR, MR CLR A: IR, MQ, SC, FE ← (0)
MC RD, WR, RQ, STOP, STOP SYNC, SPLIT CYC SYNC ← (0)
EX MODE SYNC, UUO SYNC ← (0)
EX MODE SYNC (1): EX USER ← (1)
PI CYC B (0): EX PI SYNC ← (0)
MP CLR, MP CLR A, DS CLR
Clear all SBRs and subroutine control flipflops
MR START: IOB RESET, CPA CLEAR
PI OV, PI CYC ← (0), PI RESET
AR FLAGS, CHF7, EX USER, EX ILL OP ← (0)
PR, RLR ← (0)
KEY MANUAL
KEY READ IN
KEY START
KEY INST CONT
KEY MEM CONT
KEY EX
KEY DEP
KEY EX NXT
KEY DEP NXT
KEY EXECUTE
KEY IO RESET
∧~KEY PWR CLR ENBL
KEY EXECUTE = KEY EXEC ∧ RUN (0)
KEY CLR RIM = ~(KEY READ IN ∨ KEY INST CONT ∨ KEY MEM CONT)
Connections to Memory
MC RQ CYC
MC RD RQ
MC WR RQ
MA FMC SELECT B
MAI ADDR ACK
MAI RD RS
MC WR RS B
MA FMC SELECT = (MA18-31 = Ø) ∧ KEY RIM SBR (Ø)
PR18 OK = (MA18-25 ≤ PR)
PR REL ∧ MA OK = ~EX INH REL ∧ PR18 OK
PR REL ∧ ~MA OK = EX INH REL ∧ ~PR18 OK
EX INH REL: RLA = MA
~EX INH REL: RLA = MA + RLR
EX INH REL
EX USER (Ø)
MA18-31 = Ø
EX PI SYNC (1)
EX ILL OP (1)
Compare and relocate
100 μs without MC RQ PULSEs
MC RQ (1) ∧ MC STOP (Ø)
~SW MEM DISABLE
If MC RD RQ, memory strobe sends data to MB during wait
Negate MC RQ CYC
Negate MC MB MEM BUS ENABLE
Memory cycle
KEY MANUAL ∧ MC STOP (1) ∧ MC STOP SYNC (1) ∧ ~KEY MEM CONT
KEY MEM CONT ∧ MC STOP (1)
MC STOP (Ø)
MA = MAS
KEY RD/WR (1)
KEY RD/WR RET
KEY RD/WR ← (Ø)
EX ILL OP ← (Ø)
KEY START ∨ CONT ∨ READ IN ST7
KEY GO
RUN ← (1)
KEY EX, DEP ST ← (Ø)
KEY EX, DEP SYNC ← (Ø)
IT Ø
(4-4)
IT1A
(4-4)
RUN (1) ∧ (EX ∨ DEP)
SW REPEAT
Variable repeat delay
Assert KEY INST STOP
Trigger pulse generator
RUN ← (Ø)
Assert IA INH for 100 μs
KEY MANUAL
KEY MEM CONT
KEY CLR RIM
KEY KEY RD/WR (1) IF1A (1) AF3 (1) AFØ (1) F1A (1) F4A (1) F6A (1) SF3 (1) SF5A (1) SF7 (1) UUO F1 (1) BLT FØA (1) IOT FØA (1) CHF6 (1)
KEY RD/WR RET
IT1A
AT3
ATØ
FT1A
FT4A
FT6A
ST3
ST5A
ST7
UUO T1
BLT TØA
IOT TØA
CHT88
MC RD ← (Ø)
Negate MC MB MEM BUS ENABLE
MC RS TØ
MC RS T1
MC STOP ← (Ø)
MI ← (Ø)
MI ← MB (1)
MC MEM BUS ← MB (1)
MC WR RS
MC WR RS B
MC ADDR ACK
MC NON EXIST MEM
MC NON EXIST MEM RST
MC NON EXIST RD
CPA ILLEG OP ← (1)
IOB PI REQ CPA PIA
CPA NON EXIST MEM ← (1)
IOB PI REQ CPA PIA
MC ILLEG ADDRESS
MC RD ∨ WR
MC RQ ← (1)
MC RQ ← (Ø)
MC STOP ← (1)
MC RD (1)
MC RD (Ø) ∧ MC WR (1)
KEY MEM STOP ∨ (MA = MAS ∧ SW ADDR STOP)
EX INH REL
PR REL ∧ MA OK
PR REL ∧ ~MA OK
MC RD RQ MC MB MEM BUS ENABLE
MC WR RQ
MC STOP ← (Ø) MC STOP SYNC ← (1)
MC WR ← (Ø) MC STOP ← (Ø)
MC RD ← (1)
MC WR ← (1)
MC RD ← (Ø) MC STOP ← (Ø)
MC MB ← (Ø)
MB ← (Ø)
MC RQ PULSE
MC RD/WR RQ PULSE
MC RD RQ PULSE
MC WR RQ PULSE
MC SPLIT RD RQ
MC SPLIT WR RQ
MC SPLIT CYC SYNC (1)
MC SPLIT CYC SYNC (Ø)
MC SPLIT CYC SYNC (Ø) B
MC RD/WR RS PULSE
MAI RD RS
MAI ADDR ACK
RD RQ
WR RQ
RD/WR RQ (FC(E) PSE)
RD/WR RS (FC(E) PSE)
IOT TØ
BLT TØ
MBLT ← IR (1) (UUO TØ)
KEY RD
KEY WR
UUO T2
KEY MANUAL
KEY IO RESET
KEY MEM CONT ∧ ~KEY MEM CONT
KEY CLR RIM
MR START
MR CLR
KEY RIM SBR ← (Ø)
KEY RD/WR ← (1) MB ← AR (J)
KEY RD/WR ← (1)
RD RQ
WR RQ
MC RST1
KEY (EX ∨ DEP) NXT
KEY RD/WR (1)
KEY RD/WR RET
KEY RD/WR ← (Ø) EX ILL OP ← (Ø)
KEY START ∨ CONT ∨ READ IN ST7
KEY MEM CONT ∧ MC STOP (1)
MA = MAS
MC STOP ← (1)
MC RQ CYC
MC RD ∨ WR
MC RQ ← (1)
MC RQ ← (Ø)
MC STOP ← (1)
MC RD (1)
MC RD (Ø) ∧ MC WR (1)
MC WR RS
MC MEM BUS ← MB (1)

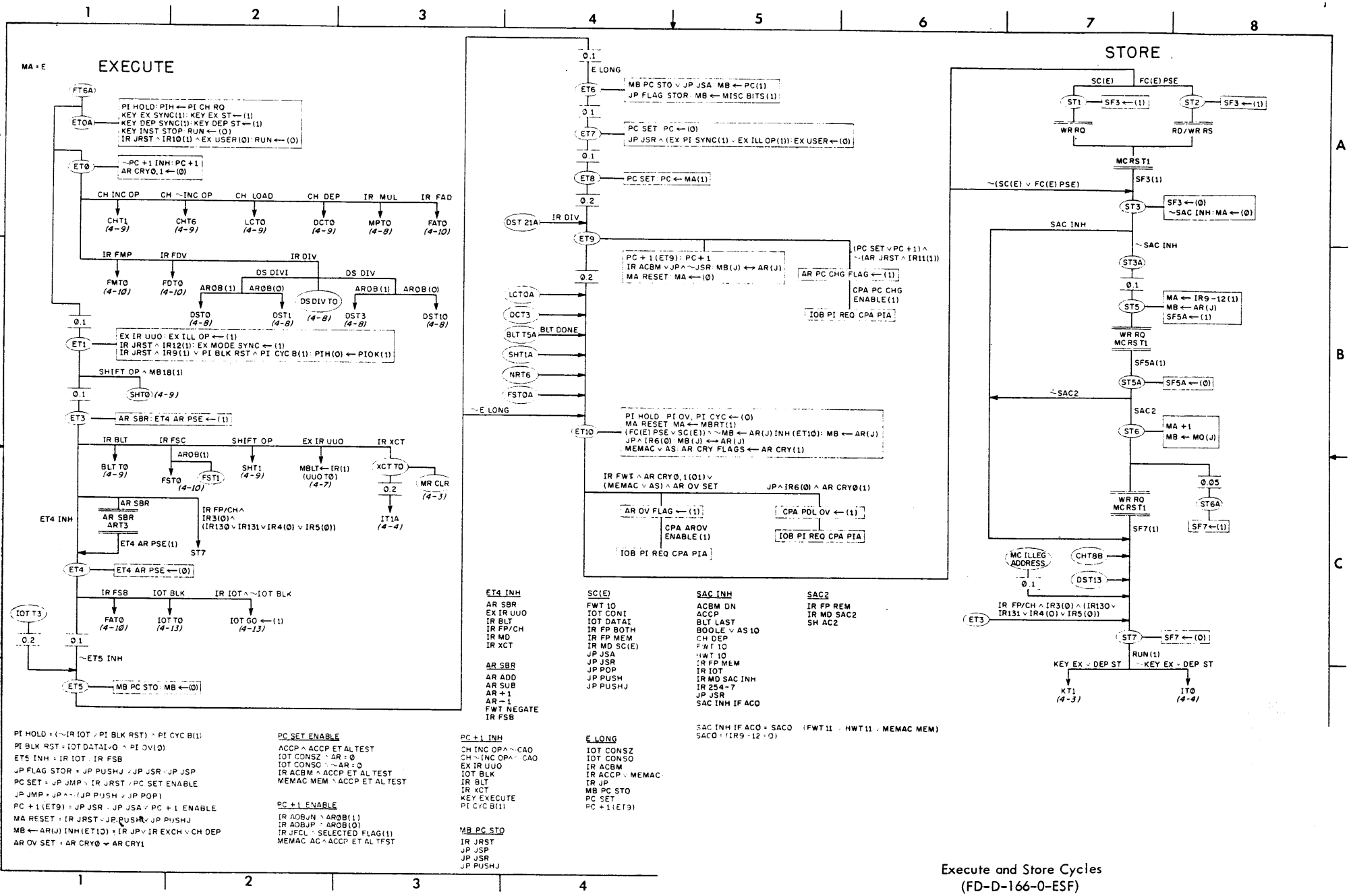


AP FIGURE 4-3

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



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



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		Action
00	= HWT 00	∕[C(E), AC] → AC
01	= HWT 01	∕[(0,E), AC] → AC
10	= HWT 10	∕[AC, C(E)] → E
11	= HWT 11	∕C(E) → E

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(001)
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) ∨ ARO(1)]
ZZ specifies mode

IR FWT ∧ IR7,8		Action
00	= FWT 00	∕C(E) → AC
01	= FWT 01	∕0,E → AC
10	= FWT 10	∕AC → E
11	= FWT 11	∕C(E) → E

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)

	HWT	FWT	EXCH
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	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	AR = AC MB = C(E) MQ = 0
Initial Gates	HWT 00: FC(E) HWT 10: FC(E) PSE HWT 11: FAC INH, FC(E) PSE	FWT 00: FAC INH, FC(E) HWT 01: FAC INH HWT 11: FAC INH, FC(E) PSE FWT NEGATE: ET4 INH, AR SBR	FC(E) PSE MB ← AR(J) INH(ET10)
ET0A	HWT 10: MB(J) ⟷ AR(J) HWT 11: AR ← MB(J)	FWT 00, 11: AR ⟷ MB(J) FWT 01, 10: MB ← AR(J)	MB(J) ⟷ AR(J)
ET1	HWT SWAP: MB LT(J) ⟷ MB RT(J) HWT AR ← (0): AR ← (0)	FWT SWAP: MBLT(J) ⟷ MBRT(J)	
ET3		FWT NEGATE: ET4 AR PSE ← (1) AR NEGATE ART3	
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)	ET4 AR PSE ← (0) FWT SWAP: MB(J) ⟶ AR(J)	
ET5			
ET10	HWT 10, 11: MB ⟶ AR(J)	AR CRY0,1(01) AR OV FLAG ← (1) FWT 10 ∨ 11: MB ⟶ AR(J)	
Final Gates	HWT 10: FC(E) PSE, SAC INH HWT 11: FC,E) PSE HWT 11: SACO: SAC INH SACO = (IR9-12 = 0)	FWT 10: SC(E), SAC INH FWT 11: FC(E) PSE FWT 11 ∧ SACO: SAC INH SACO = (IR9-12 = 0)	FC(E) PSE

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) ∧ (ARØ ⟷ AR OV)]
ACCP ET AL TEST = ACCP ETC COND ⟷ IR6
ARO ⟷ AR OV = ARO ⟷ (AR OV SET ∧ ~MEMAC)
AR OV SET = AR CRY0 ⟷ AR CRY1

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	= Ø
10	*Always*
11	≠ Ø

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

AOBJP · AOBJN
252 · 253

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 ∧ ARØ8(0): PC SET
IR AOBJN ∧ ARØ8 (1): PC SET

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 Ø: E ⟶ PC	C(E) R Ø: PC+1
MEMAC ±1: AC ±1 → AC	MEMAC ±1: C(E) ±1 → E
ACCP ET AL TEST: PC SET	~ACØ ∧ MEMAC ±1: C(E) ±1 → AC
	SACØ = (IR9-12 = Ø)

ACCP
300-317

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

ACBM

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

	MEMAC	ACCP	ACBM
Initial Registers	AR = AC MB = Ø,E MQ = 0	AR = Ø,E MB = C(E) MQ = 0	AR = AC ~ACCP DIR: MB = Ø,E ACCP DIR: MB = C(E) MQ = 0	AR = AC ~ACBM DIR: MB = Ø,E ACBM DIR: MB = C(E) MQ = 0	AR = AC MB = Ø,E MQ = Ø
Initial Gates	MEMAC ±1: ET4 INH AR SBR E LONG	FAC INH FC(E) PSE MEMAC ±1: ET4 INH AR SBR E LONG	ACCP DIR: FC(E) ET4 INH AR SBR E LONG	ACBM DIR: FC(E) E LONG	ET4 INH AR SBR E LONG
ET0A		AR ⟶ MB(J)		ACBM SWAP: MB LT(J) ⟷ MB RT(J)	
ET1				MB ⟶ AR(Ø) ACBM COM: AR ⟷ MB(⟷) ACBM SET: AR ⟷ MB(1)	
ET3	MEMAC +1: ET4 AR PSE ⟵ (1) ∴AR +1 MEMAC −1: ET4 AR PSE ⟵ (1) ∴AR −1 ART3		ET4 AR PSE ⟵ (1) ∴AR SUB ART3		ET4 AR PSE ⟵ (1) ∴AR +1 LT RT ART3
ET4	ET4 AR PSE ⟵ (0)	ET4 AR PSE ⟵ (0)	ET4 AR PSE ⟵ (0)	MB(J) ⟷ AR(J)	ET4 AR PSE ⟵ (Ø)
ET5				AR COM	
ET6				ACBM CL: MB ⟵ AR(0)	
ET7	PC SET: PC ⟵ (0)			AR COM	PC SET: PC ⟵ (Ø)
ET8	PC SET: PC ⟵ MA(1)				PC SET: PC ⟵ MA(1)
ET9	PC SET: AR PC CHG FLAG ⟵ (1)	ACCP ET AL TEST: PC + 1 AR PC CHG FLAG ⟵ (1)	ACCP ET AL TEST: PC + 1 AR PC CHG FLAG ⟵ (1)	MB(J) ⟵ AR(J) ACCP ET AL TEST: PC + 1 AR PC CHG FLAG ⟵ (1)	PC SET: AR PC CHG FLAG ⟵ (1)
ET10	AR OV SET: AR OV FLAG ⟵ (1) AR CRY FLAGS ⟵ AR CRY(1)	AR OV SET: AR OV FLAG ⟵ (1) AR CRY FLAGS ⟵ AR CRY(1) MB ⟵ AR(J)			
Final Gates		FC(E) PSE SACØ: SAC INH	SAC INH	ACBM DN: SAC INH	

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

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

E→PC
IR9(1): DISMISS PI
IR10(1): HALT
IR11(1): RESTORE MISC BITS
IR12(1): ENTER USER MODE

AR = 0,E MB = ?,E MQ = 0	
FAC INH E LONG	
IR10(1): RUN←(0) IR11(1): AR FLAG CLEAR (*Clears* MISC BITS *at* ET0)	
IR9(1): PIH(0)←PIOK(1) IR11(1): AR FLAG SET (*Transfers* MB(1) *into* MISC BITS) IR12(1): EX MODE SYNC←(1)	
MB←(0)	
MB←PC(1)	
PC←(0)	SELECTED FLAG(1): PC←(0)
PC←MA(1)	SELECTED FLAG(1): PC←MA(1)
MA←(0)	
MA←MBRT(1)	SELECTED FLAGS←(0)
SAC INH	SAC INH

JFCL 255

IR = 010 101 101
IR 2xx = IR0-2(010)
IR 25x = IR 2xx ∧ IR3-5(101)
IR JFCL = IR 25x ∧ IR 6-8(010)

SELECTED FLAG(1): E→PC
CLEAR SELECTED FLAGS

AR = 0,E MB = ?,E MQ = 0
FAC INH E LONG

XCT 256

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)

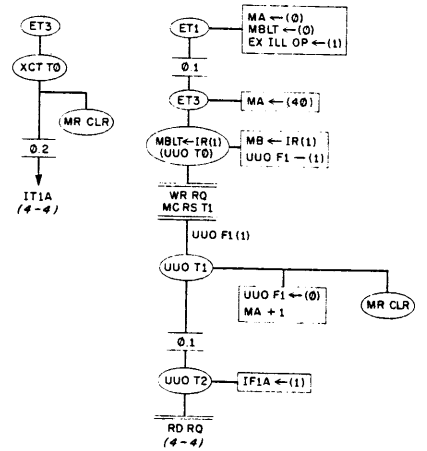
AR = 0,E MB = C(E) MQ = 0
FAC INH FC(E) PC + 1 INH ET4 INH

UUO 000-077

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

AR = 0,E MB = ?,E MQ = 0
FAC INH PC + 1 INH ET4 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)

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: ƒ[AC, C(E)] → AC	AC ±C(E) → AC
BOOLE ∨ AS 01: ƒ[AC, (0,E)] → AC	AC ±(0,E) → AC
BOOLE ∧ AS 10: ƒ[AC, C(E)] → E	AC ±C(E) → E
BOOLE ∧ AS 11: ƒ[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
ETOA	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
ETOA	AR COM			
ET1	AR ← MB(1)		AR ← MB(∨)	AR ← MB(1)
ET4	AR COM			

	10 ANDCB	11 EQV	12 SETCA	13 ORCA
ETOA			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
ETOA	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 ∨ 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

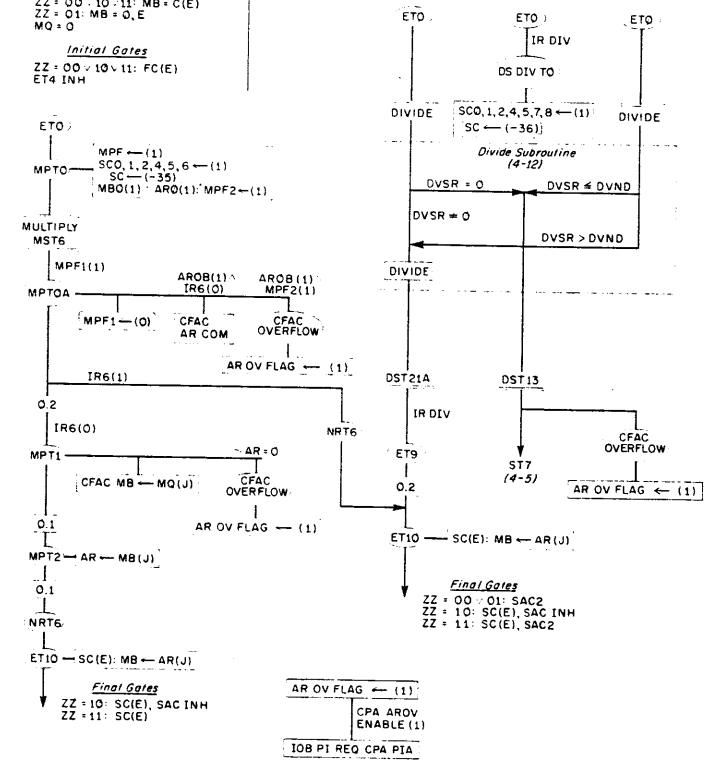
ZZ = 00: AC, AC2 ← C(E) → AC, AC2
ZZ = 01: AC, AC2 ← (0,E) → AC, AC2
ZZ = 10: AC, AC2 ← C(E) → E
ZZ = 11: AC, AC2 ← C(E) → AC, AC2; E

Initial Registers

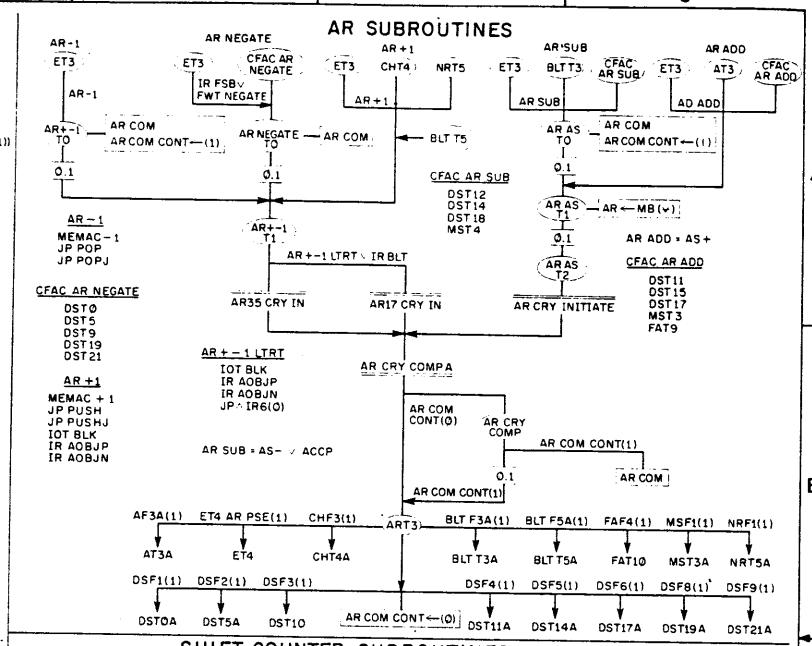
AR = AC
ZZ = 00 ∨ 10 ∨ 11: MB = C(E)
ZZ = 10: MB = 0, E
MQ = AC2

Initial Gates

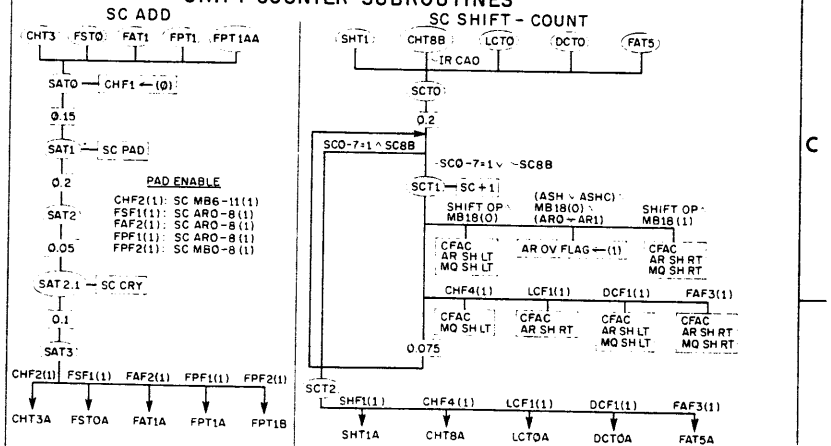
ZZ = 00 ∨ 10 ∨ 11: FC(E)
FAC2
ET4 INH



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(Ø)

CH INC = (IR LDCI ∨ IR DPCI ∨ IR CAO) ∧ CHF5(Ø)
CH INC OP = CH INC ∧ CHF7(Ø)
CH ~INC OP = ~[(IR LDC ∨ IR DPC) ∧ CHF5(Ø)] ∨ [CH INC ∧ CHF7(11)]

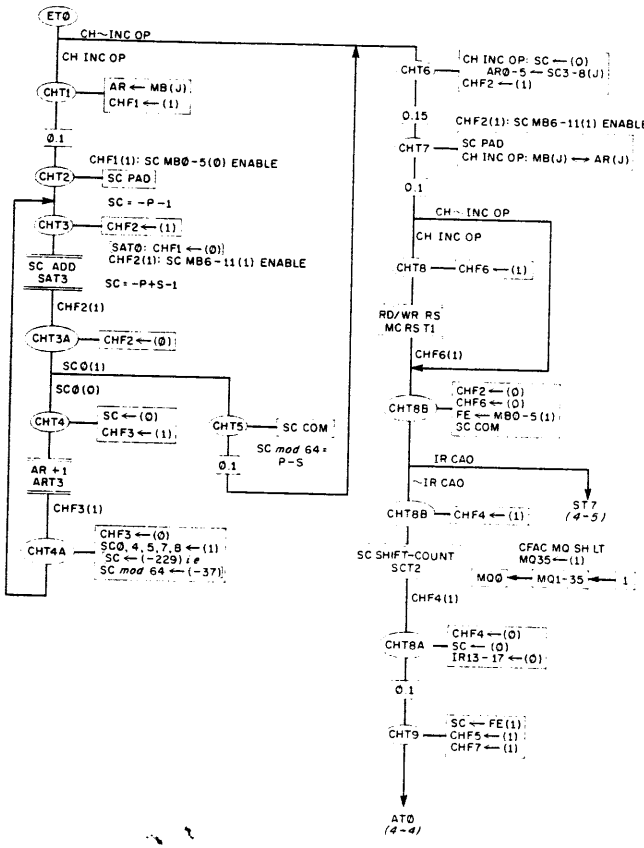
Pointer Format

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

Fetch and increment pointer

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

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)

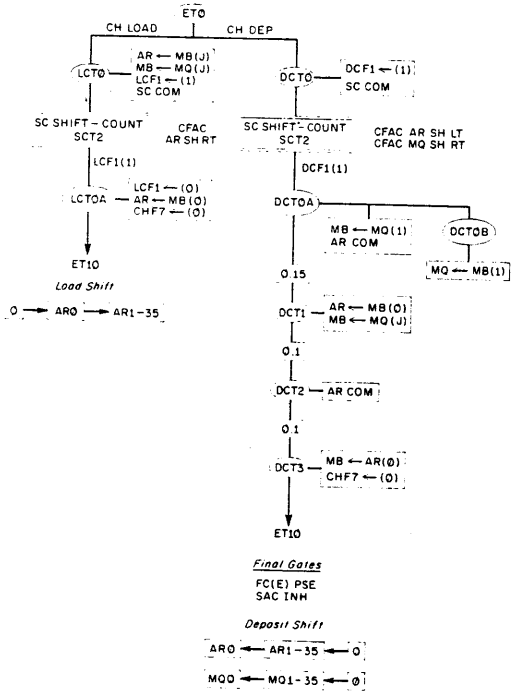
CH LOAD
Fetch and load character

CH DEP
Fetch and deposit character

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

Initial Gates
FAC INH
FC(E)
ET4 INH

FC(E) PSE
ET4 INH
MB ← AR(J) INH(ETØ)



BLT

251

IR = Ø10 101 ØØ1
IR 2XX = IR0-2(Ø10)
IR 25X = IR 2XX ∧ IR3-5(101)
IR BLT = IR 25X ∧ IR6-8(ØØ1)
BLT DONE = MQØ(Ø) ∨ PI RQ
BLT LAST = IR BLT ∧ MQØ(Ø)

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 = Ø,E
MQ = C(S) *initially* C(C(ACLT))

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

SHIFT OPERATIONS

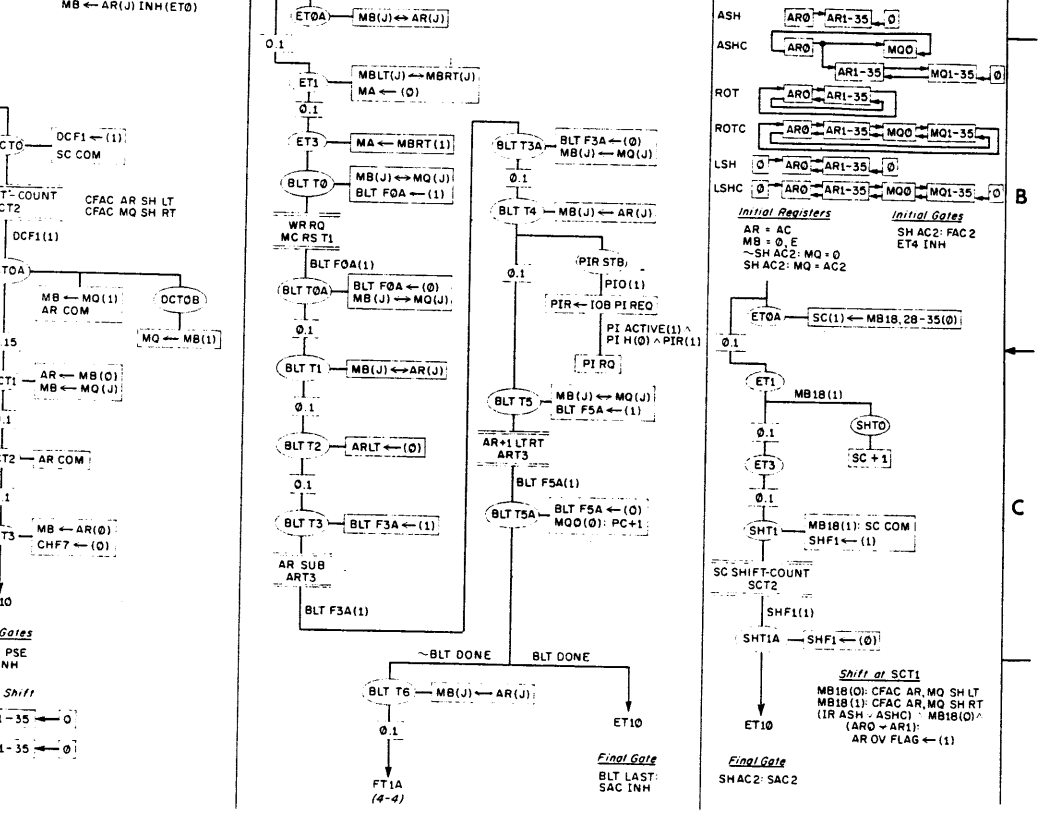
240-242, 244-246

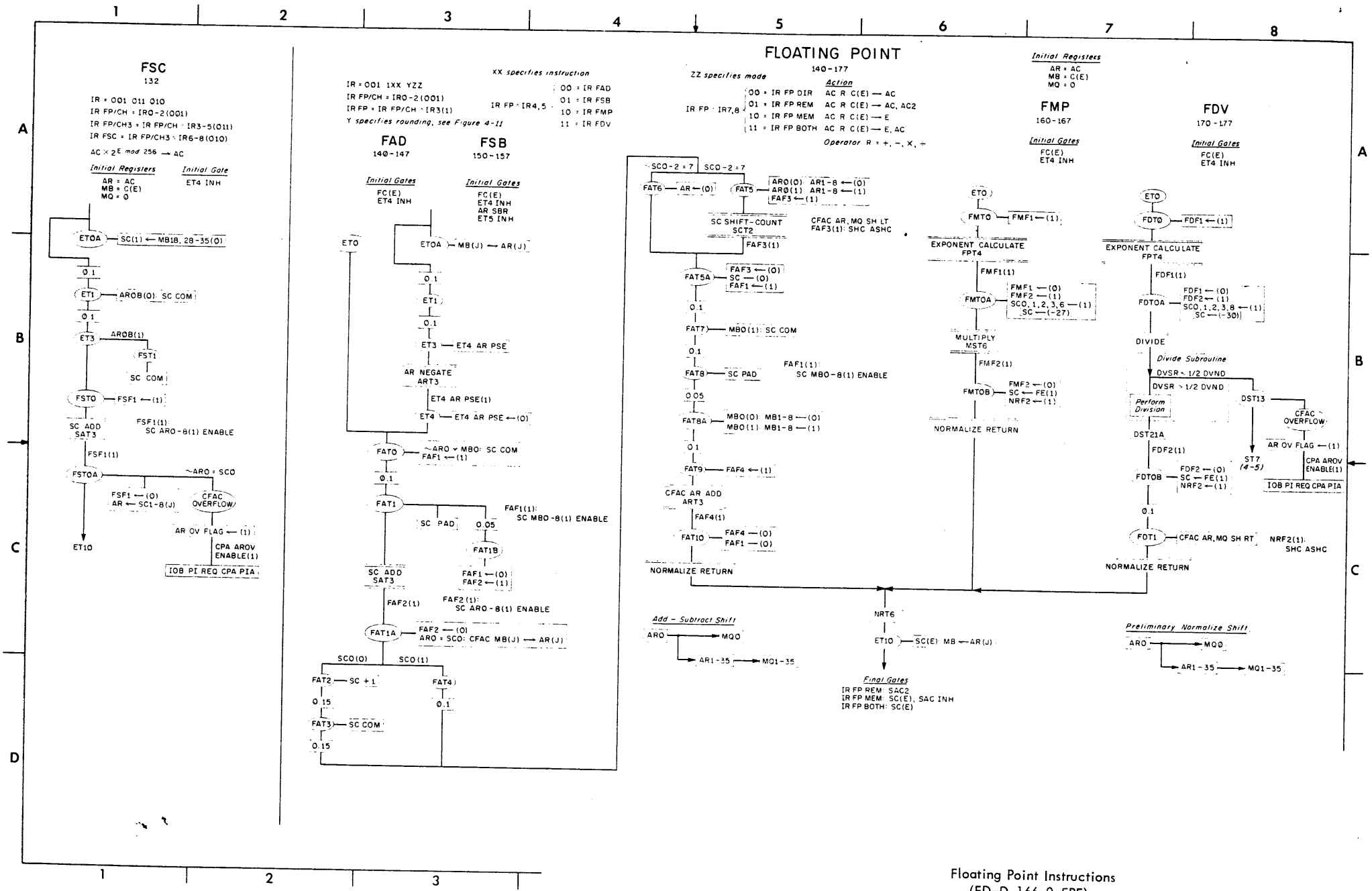
IR = Ø10 100 xxx
IR 2XX = IR0-2(Ø10)
IR SH = IR 2XX ∧ IR3-5(100)

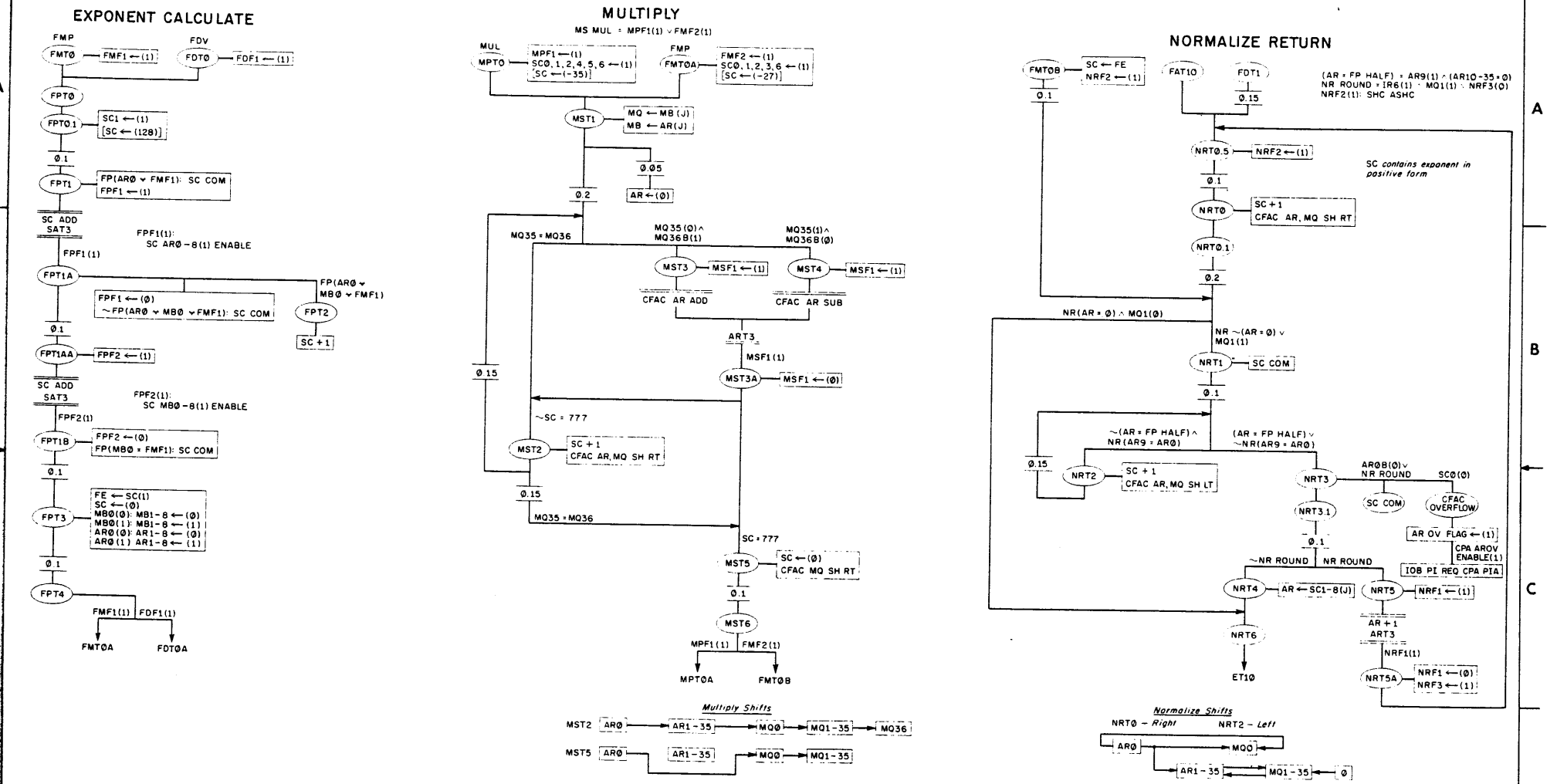
IR SH ∧ IR6-8 →
- ØØØ = IR ASH
- ØØ1 = IR ROT
- Ø1Ø = IR LSH
- 1ØØ = IR ASHC
- 101 = IR ROTC
- 11Ø = IR LSHC

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: / [AC] → AC
SH AC2: / [AC, AC2] → AC

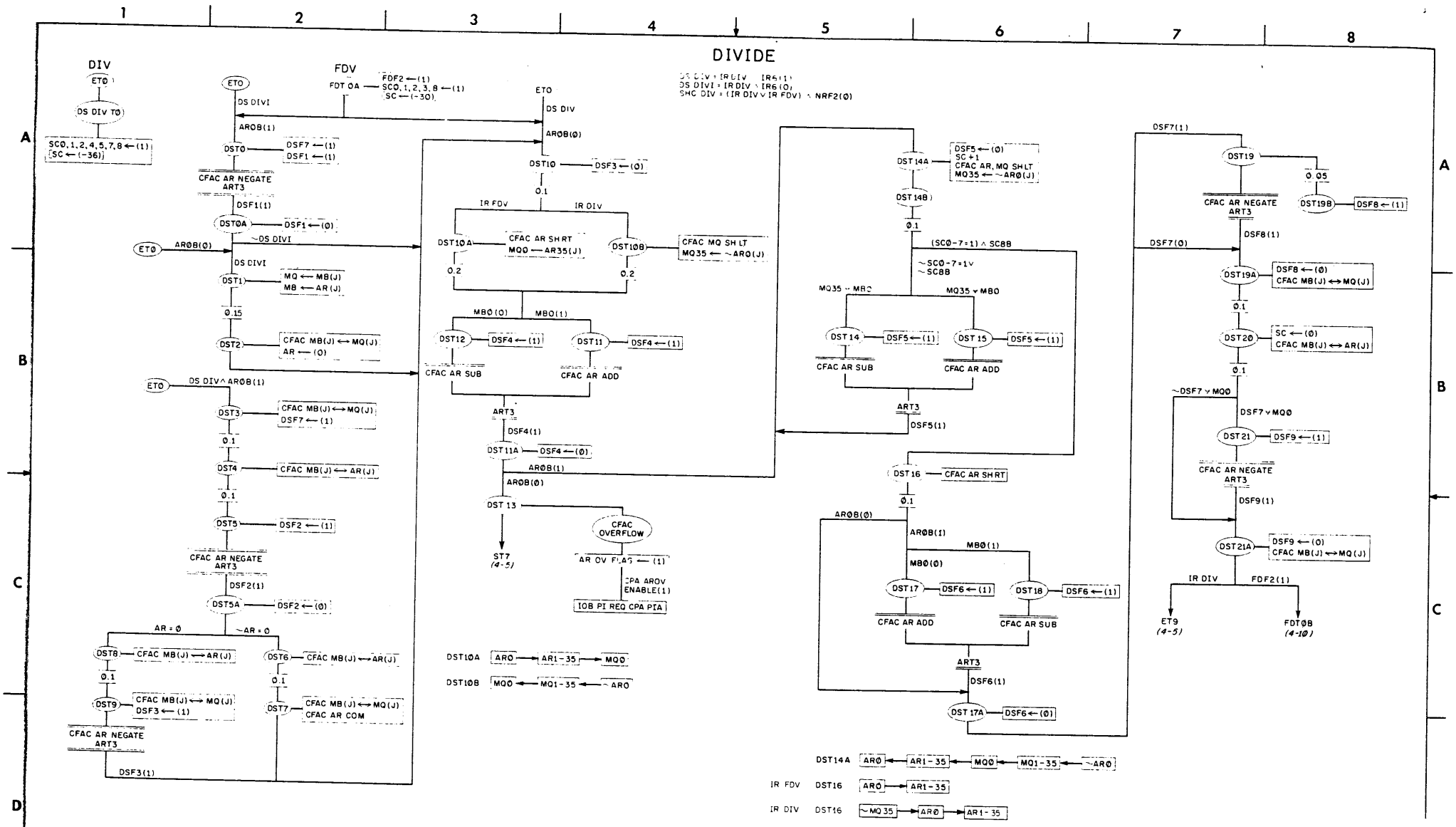
SHIFT OP AR, MQ CONNECTIONS
IR ASHC: SHC ASHC




FSC
132
IR = 001 011 010
IR FP/CH = IR0-2(001)
IR FP/CH3 = IR FP/CH ∧ IR3-5(011)
IR FSC = IR FP/CH3 ∧ IR6-8(010)
AC × 2^E mod 256 → AC
Initial Registers
AR = AC
MB = C(E)
MQ = 0
Initial Gate
ET4 INH
ETOA
SC(1) ← MB18, 28-35(0)
0.1
ET1
AROB(0): SC COM
0.1
ET3
AROB(1)
FST1
SC COM
FSTO
FSF1 ← (1)
SC ADD SAT3
FSF1(1): SC ARO-8(1) ENABLE
FSF1(1)
FSTOA
~ARO = SCO
FSF1 ← (0)
AR ← SC1-8(J)
CFAC OVERFLOW
AR OV FLAG ← (1)
CPA AROV ENABLE(1)
IOB PI REQ CPA PIA
ET10
IR = 001 1XX YZZ
IR FP/CH = IR0-2(001)
IR FP = IR FP/CH ∧ IR3(1)
Y specifies rounding, see Figure 4-11
XX specifies instruction
IR FP = IR4,5
00 = IR FAD
01 = IR FSB
10 = IR FMP
11 = IR FDV
FAD
140-147
FSB
150-157
Initial Gates
FC(E)
ET4 INH
Initial Gates
FC(E)
ET4 INH
AR SBR
ET5 INH
ETO
ETOA
MB(J) → AR(J)
0.1
ET1
0.1
ET3 → ET4 AR PSE
AR NEGATE ART3
ET4 AR PSE(1)
ET4 → ET4 AR PSE ← (0)
FAT0
~ARO ∨ MBO: SC COM
FAF1 ← (1)
0.1
FAT1
FAF1(1): SC MBO-8(1) ENABLE
SC PAD
0.05
FAT1B
FAF1 ← (0)
FAF2 ← (1)
SC ADD SAT3
FAF2(1)
FAF2(1): SC ARO-8(1) ENABLE
FAT1A
FAF2 ← (0)
ARO = SCO: CFAC MB(J) → AR(J)
SCO(0)
SCO(1)
FAT2
SC + 1
0.15
FAT3
SC COM
0.15
FAT4
0.1
FLOATING POINT
140-177
ZZ specifies mode
IR FP = IR7,8
00 = IR FP DIR
01 = IR FP REM
10 = IR FP MEM
11 = IR FP BOTH
Action
AC R C(E) → AC
AC R C(E) → AC, AC2
AC R C(E) → E
AC R C(E) → E, AC
Operator R = +, −, ×, ÷
Initial Registers
AR = AC
MB = C(E)
MQ = 0
FMP
160-167
Initial Gates
FC(E)
ET4 INH
FDV
170-177
Initial Gates
FC(E)
ET4 INH
SCO-2 = 7
SCO-2 = 7
FAT6
AR ← (0)
FAT5
ARO(0) AR1-8 ← (0)
ARO(1) AR1-8 ← (1)
FAF3 ← (1)
SC SHIFT-COUNT SCT2
CFAC AR, MQ SH LT
FAF3(1): SHC ASHC
FAF3(1)
FAT5A
FAF3 ← (0)
SC ← (0)
FAF1 ← (1)
0.1
FAT7
MBO(1): SC COM
0.1
FAT8
SC PAD
FAF1(1): SC MBO-8(1) ENABLE
0.05
FAT8A
MBO(0): MB1-8 ← (0)
MBO(1): MB1-8 ← (1)
0.1
FAT9
FAF4 ← (1)
CFAC AR ADD ART3
FAF4(1)
FAT10
FAF4 ← (0)
FAF1 ← (0)
NORMALIZE RETURN
ETO
FMTO
FMF1 ← (1)
EXPONENT CALCULATE FPT4
FMF1(1)
FMTOA
FMF1 ← (0)
FMF2 ← (1)
SCO.1,2,3,6 ← (1)
SC ← (-27)
MULTIPLY MST6
FMF2(1)
FMTOB
FMF2 ← (0)
SC ← FE(1)
NRF2 ← (1)
NORMALIZE RETURN
ETO
FDTO
FDF1 ← (1)
EXPONENT CALCULATE FPT4
FDF1(1)
FDTOA
FDF1 ← (0)
FDF2 ← (1)
SCO.1,2,3,8 ← (1)
SC ← (-30)
DIVIDE
Divide Subroutine
DVSR ∧ 1/2 DVND
DVSR ∧ 1/2 DVND
Perform Division
DST13
CFAC OVERFLOW
AR OV FLAG ← (1)
ST7 (4-5)
CPA AROV ENABLE(1)
IOB PI REQ CPA PIA
DST21A
FDF2(1)
FDTOB
FDF2 ← (0)
SC ← FE(1)
NRF2 ← (1)
0.1
FDT1
CFAC AR, MQ SH RT
NRF2(1): SHC ASHC
NORMALIZE RETURN
NRT6
ET10
SC(E) MB ← AR(J)
Add - Subtract Shift
ARO
MQ0
AR1-35
MQ1-35
Preliminary Normalize Shift
ARO
MQ0
AR1-35
MQ1-35
Final Gates
IR FP REM: SAC2
IR FP MEM: SC(E), SAC INH
IR FP BOTH: SC(E)



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



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

IN-OUT TRANSFER

IR = 111 XXX XXX XYY Y

IR IOT A = IR0-2 (111)

IR IOT = IR IOT A ∧ ~EX IR UUO

YYY specifies instruction

IR IOT ∧ 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

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

Instruction Action

BLKI	C(E) + 1000001 → E, DEVICE BUFFER → C(C(ERT))
DATAI	DEVICE BUFFER → E
BLKO	C(E) + 1000001 → E, C(C(ERT)) → DEVICE BUFFER
DATAO	C(E) → DEVICE BUFFER
CONO	E → DEVICE CONTROL
CONI	STATUS → E
CONSZ	(STATUS ∧ E) = 0: PC + 1
CONSO	(STATUS ∧ E) ≠ 0: 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(0)
PI BLK RST ∧ PI CYC B(1): PI HOLD, PI RST
IOT CONSZ ∨ IOT CONSO: E LONG

BLK

Initial Registers

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

Initial Gates

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

ET0A → AR ← MB(J)
→ 0.1
ET1
→ 0.1
ET3 → ET4 AR PSE ← (1)
AR + 1 LT RT ART3
ET4 AR PSE(1)
ET4 → ET4 AR PSE ← (0); MB(J) ↔ AR(J); EX ILL OP ← (0)
IOT T0 → IOT F0A ← (1)
RD/WR RS MC RS T1
IOT F0A(1)
IOT T0A → IOT F0A ← (0); IR12 ← (1); MA ← (0); PI CYC(1) ∧ AR CRY0(1): PI OV ← (1); PI CYC B(0) ∧ AR CRY0(0): PC + 1
→ 0.2
FT0

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

	DATAI	DATAO	CONO	STATUS
Initial Registers	MB = ?, E AR = 0, E MQ = 0	MB = C(E) AR = 0, E MQ = 0	MB = ?, E AR = 0, E MQ = 0	MB = ?, E AR = 0, E MQ = 0
Initial Gates	FAC INH ET5 INH	FAC INH FC(E) ET5 INH	FAC INH ET5 INH	FAC INH ET5 INH

ET0A: PI HOLD: PIH ← PI CH RQ | AR ← MB(J); PI HOLD: PIH ← PI CH RQ | MBLT ← MBRT(J) | MB ← AR(J)
→ 0.1
ET1: AR ← (0); PI RST: PIH(0) ← PI OK(1) | PI RST: PIH(0) ← PI OK(1) | | AR ← (0)
→ 0.1
ET3
ET4: | | ARLT ← MBLT(J) |

IOT GO ← (1)
IOT RESET
~IOT RESET — *Wait for* ~IOT RESET

INITIAL SETUP 1.0
IOT T2 → IOT GO ← (0); IOT DATAO → IOB DATAO CLEAR; IOT CONO → IOB CONO CLEAR

RESTART 1.0
IOT T3 → AR ← IOB(1); IOT DATAO → IOB DATAO SET; IOT CONO → IOB CONO SET
→ 0.2
ET5

FINAL SETUP 1.5
IOT OUTGOING: IOB ← AR(1)
IOT DATAI: IOB ← DATAI
IOT STATUS: IOB ← STATUS

RESET 2.0
IOT RESET
IOB BUS RESET

~E LONG
ET10 → IOT DATAO: PI HOLD: PI CYC ← (0); IOT DATAI: MB ← AR(J); PI HOLD: PI CYC ← (0); IOT CONI: MB ← AR(J)

E LONG
→ 0.1
ET6 → AR ← MB(0)
→ 0.1
ET7
→ 0.1
ET8
→ 0.2
ET9 → IOT CONSZ: AR = 0: PC + 1; IOT CONSO: ~AR = 0: PC + 1
ET10

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

CPA

000

CPA = IOS3-9 (000 000 0)

IOB CONO SET → CPA → CPA CONO SET → IOB19 → IOB RESET → *Clear IO equipment*
MR START → CPA CONO SET
IOB DATAO CLEAR → CPA → CPA CLEAR → *Clear CPA flags*
IOB19 → EX CLR → PR ← (0); RLR ← (0)
IOB DATAI SET → CPA → EX SET → PR ← IOB0-7(1); RLR ← IOB18-25(1)
Assert IOB ← DATAI → CPA → AR ← DATA SW(1)

IOB	CPA CONO SET	CPA STATUS
18	CPA PDL OV ← (0)	
19	IOB RESET	
20	CPA IOT USER ← (1)	CPA PDL OV(1)
21	CPA IOT USER ← (0)	CPA IOT USER(1)
22	CPA ILLEG OP ← (0)	EX USER B(1)
23	CPA NON EXIST MEM ← (0)	CPA ILLEG OP(1)
24	CPA CLOCK ENABLE ← (0)	CPA NON EXIST MEM(1)
25	CPA CLOCK ENABLE ← (1)	
26	CPA CLOCK FLAG ← (0)	CPA CLOCK ENABLE(1)
27	CPA PC CHG ENABLE ← (0)	CPA CLOCK FLAG(1)
28	CPA PC CHG ENABLE ← (1)	
29	AR PC CHG FLAG ← (0)	CPA PC CHG ENABLE(1)
30	CPA AROV ENABLE ← (0)	AR PC CHG FLAG B(1)
31	CPA AROV ENABLE ← (1)	
32	AR OV FLAG ← (0)	CPA AROV ENABLE(1)
33-35	CPA PIA ← IOB33-35(1)	AR OV FLAG B(1)
		CPA PIA B(1)

60 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)
ET10 ∧ [MB(J) ↔ AR(J) (ET10)] ∧ AR CRY0(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)

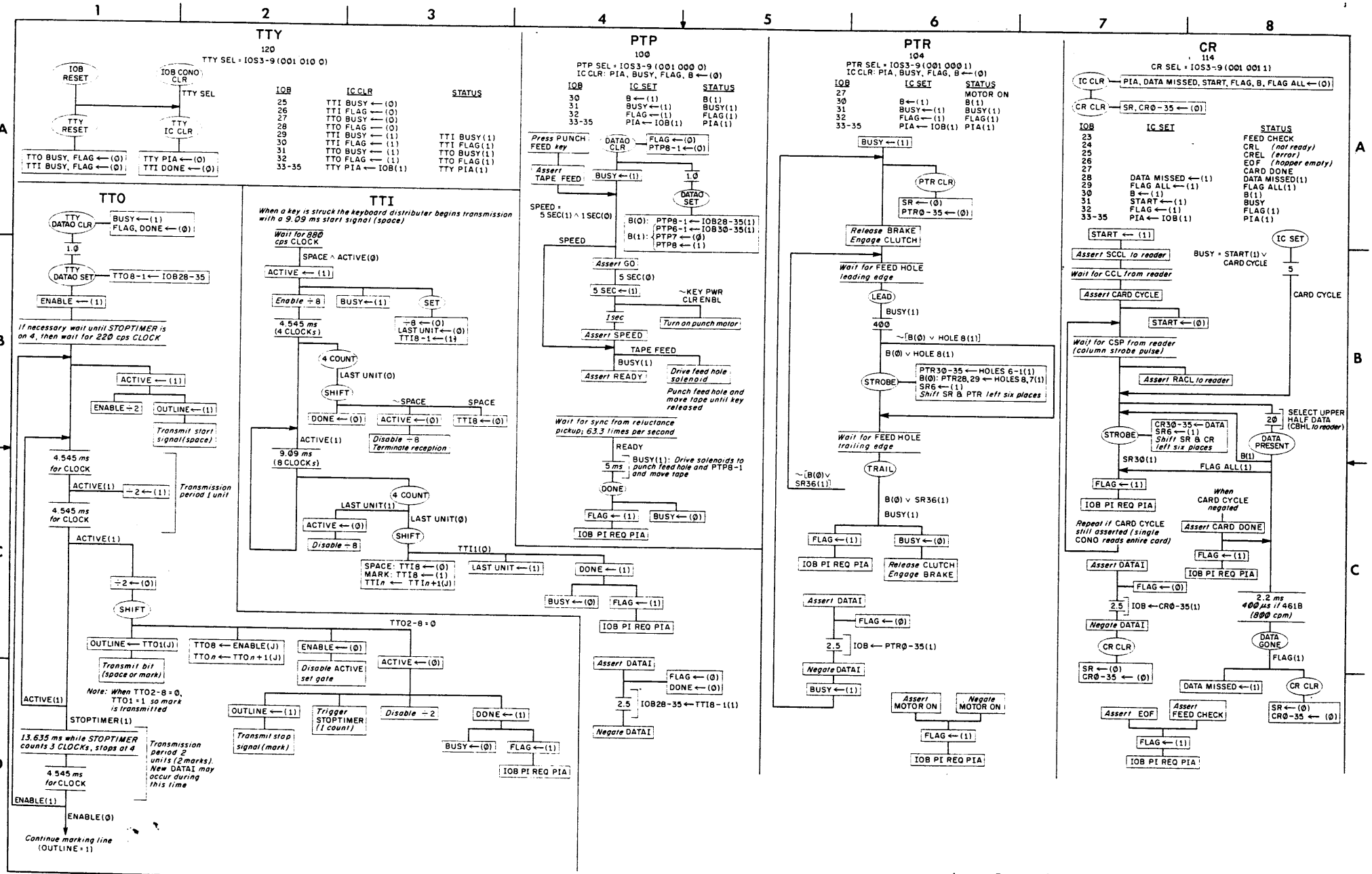
PI

004

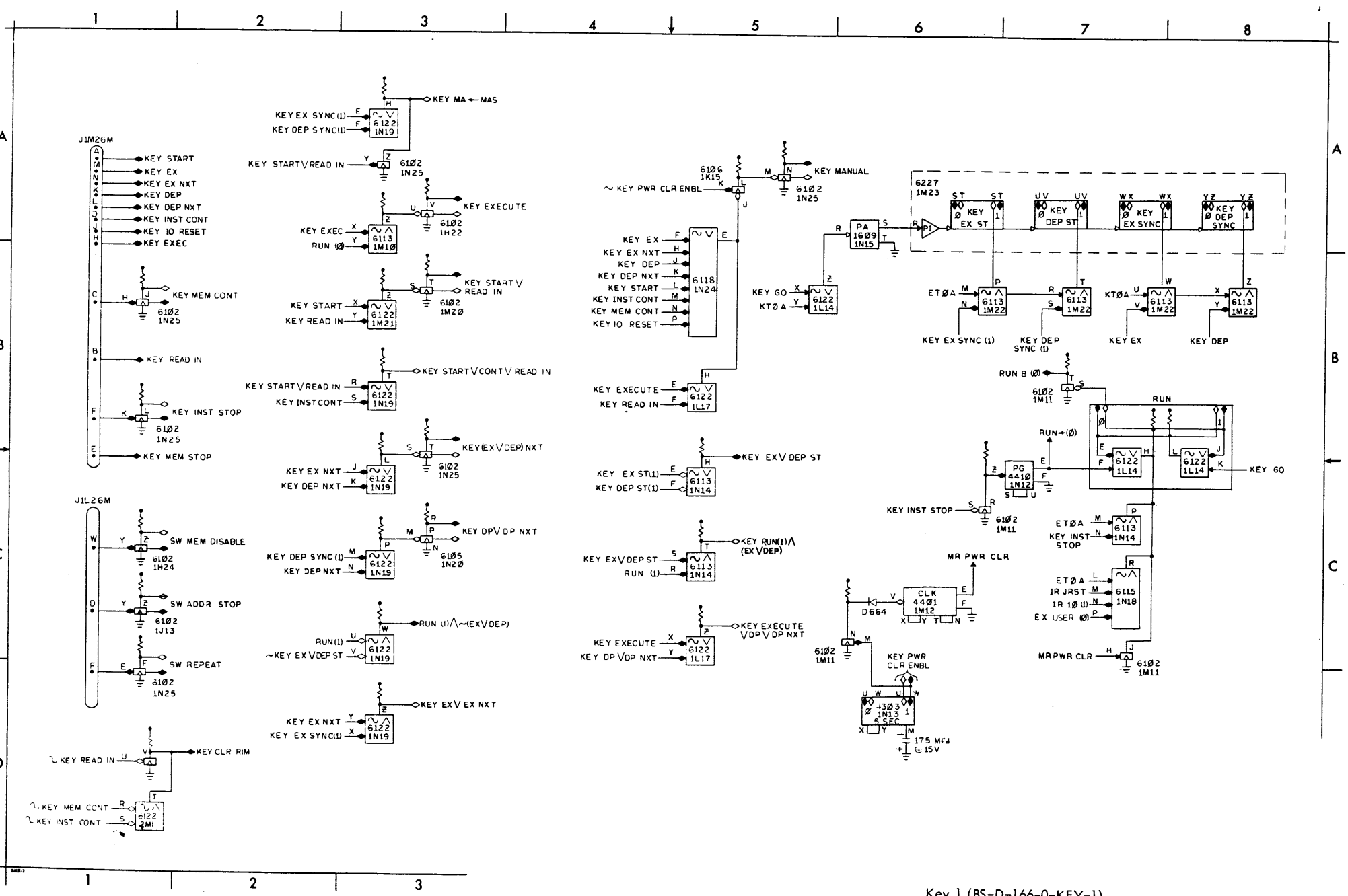
PI SELECT = IOS3-9 (000 000 1)

MR START → PI OV ← (0); PI CYC ← (0)
IOB CONO CLEAR → PI SELECT ∧ IOB23 → PI RESET → PI ACTIVE ← (0); PIO ← (0); PIR ← (0); PIH ← (0)

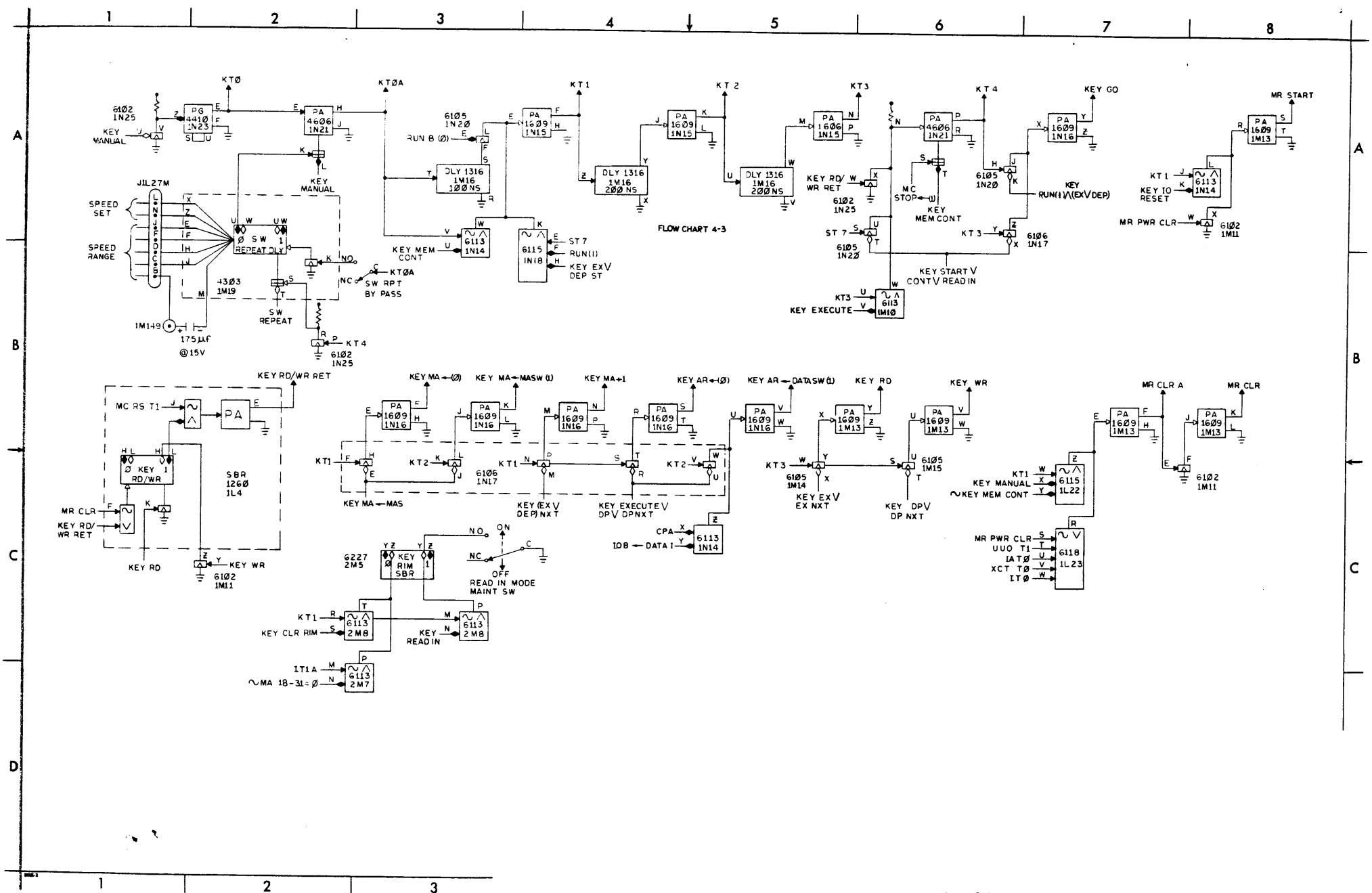
IOB	PI CONO SET	PI STATUS
24	PIR ← IOB29-35(1)	
25	PIO ← IOB29-35(1)	
26	PIO(0) ← IOB29-35(1)	
27	PI ACTIVE ← (0)	
28	PI ACTIVE ← (1)	PI ACTIVE(1)
29-35		PIO1-7(1)

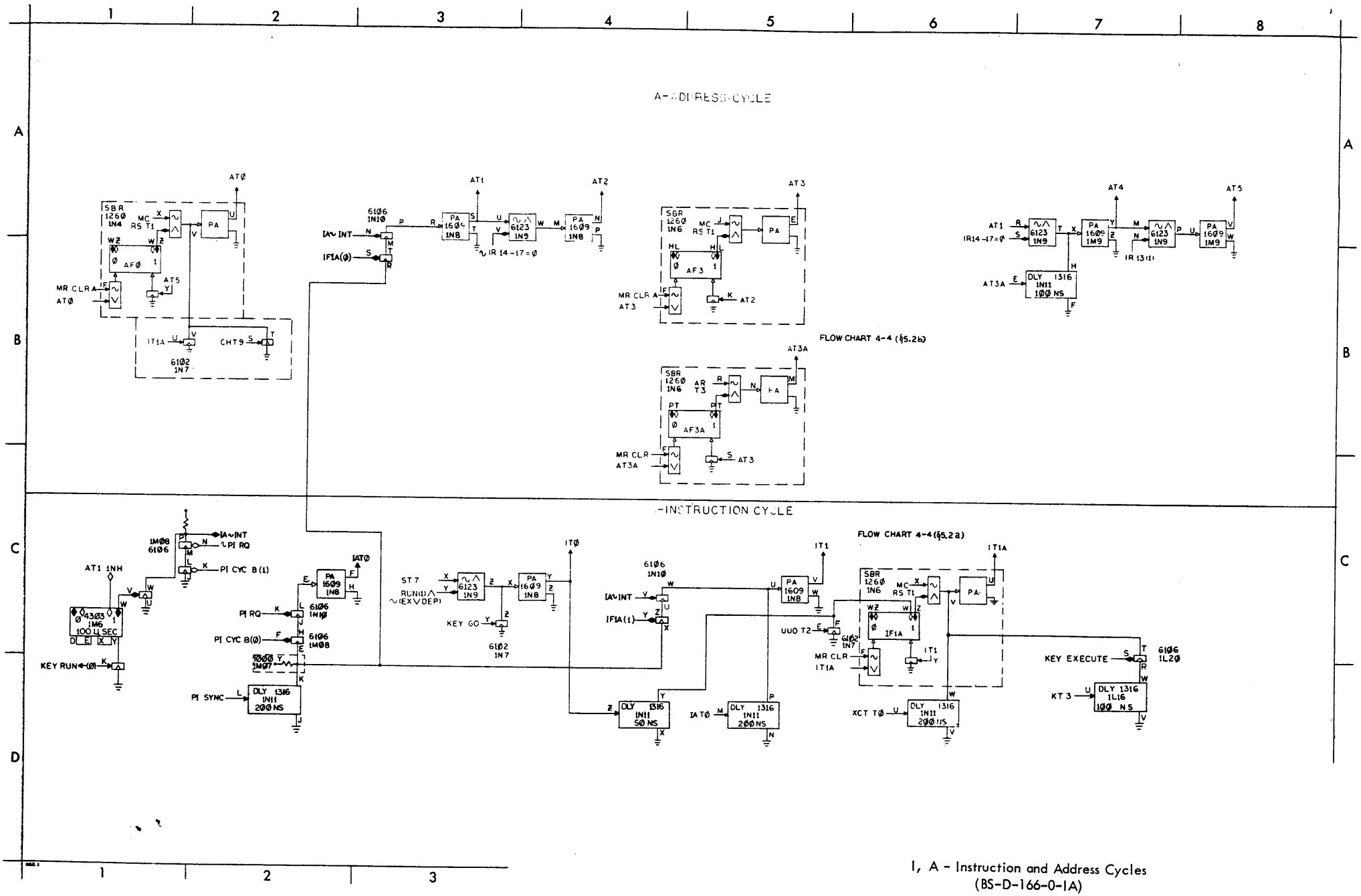


Input-Output Operations
(FD-D-166-0-1OF)

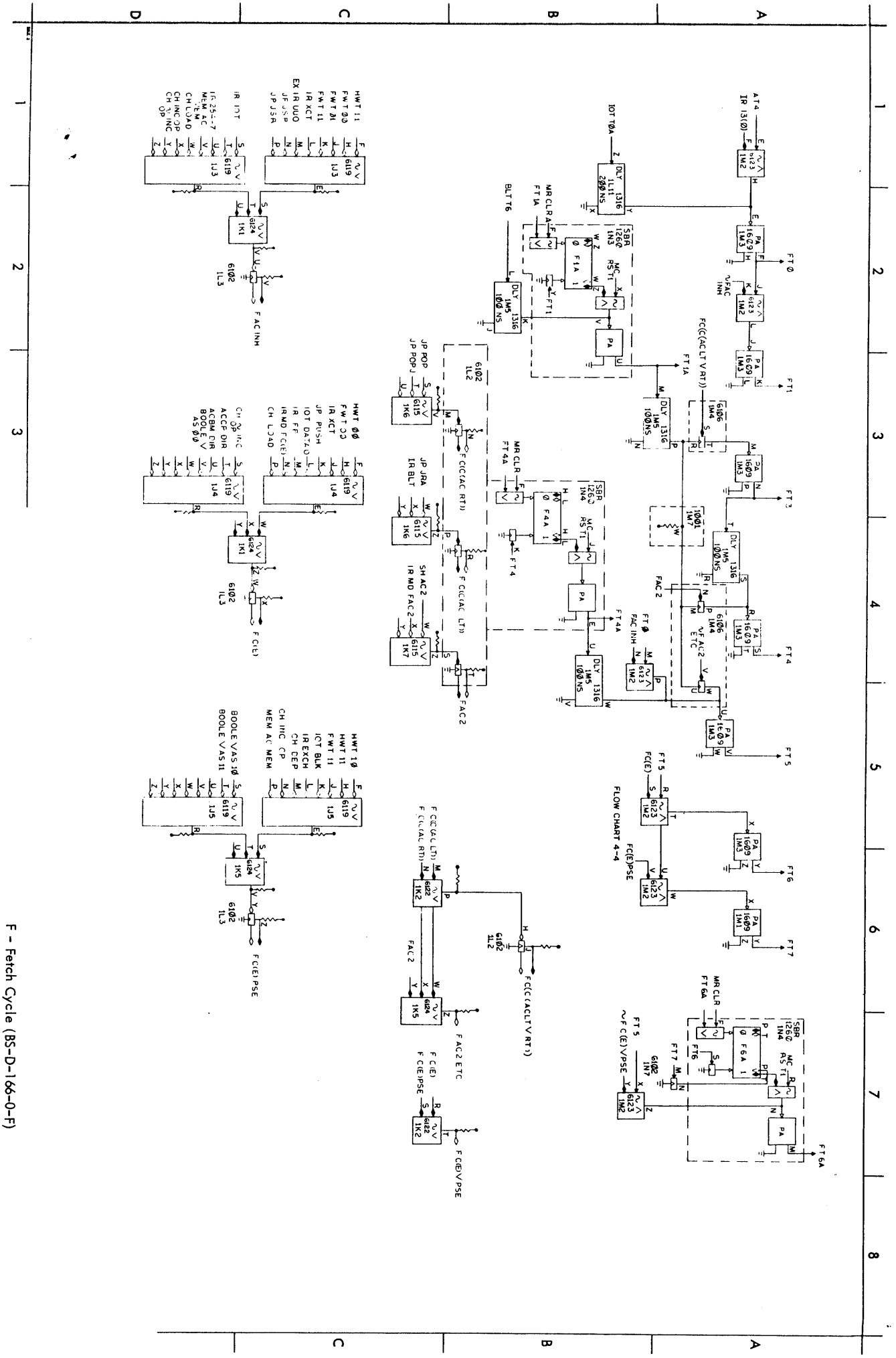


Key 1 (BS-D-166-0-KEY-1)

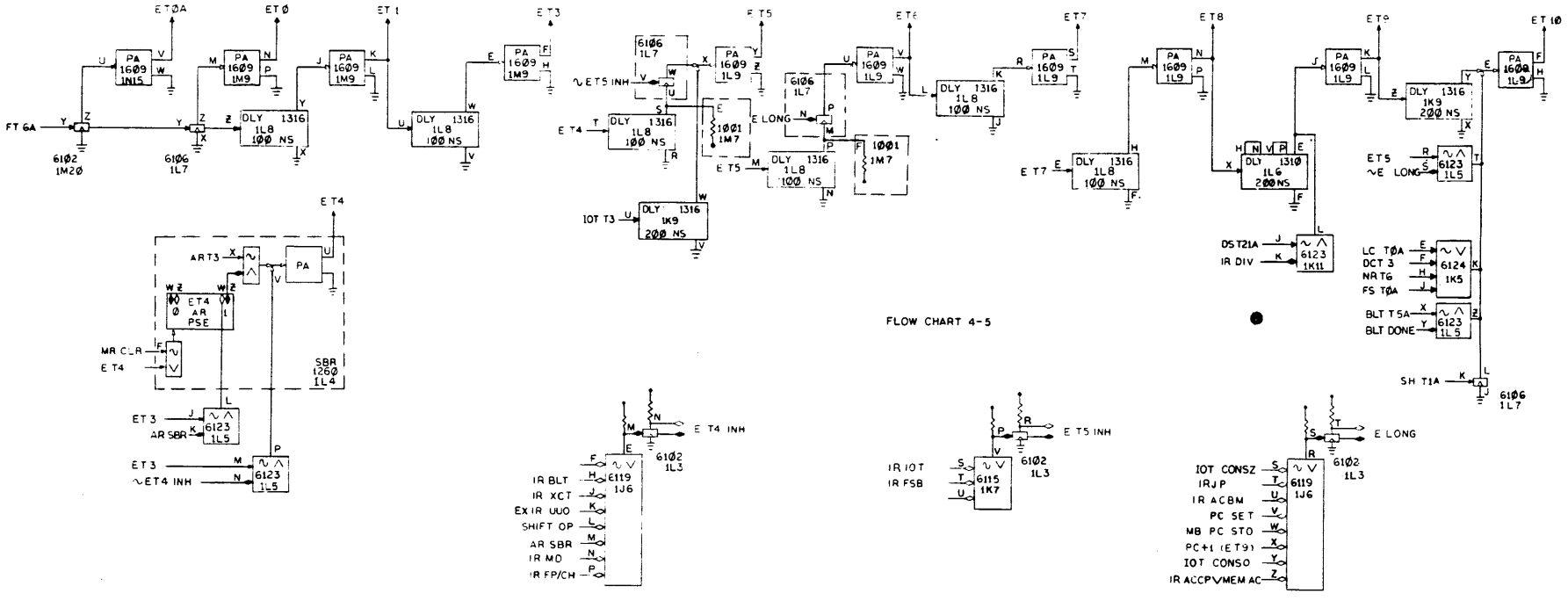

KTØ
KTØA
KT1
KT 2
KT3
KT 4
KEY GO
MR START
KEY MANUAL
SPEED SET
SPEED RANGE
REPEAT DLY
SW REPEAT
SW RPT BY PASS
KEY MEM CONT
RUN B (Ø)
FLOW CHART 4-3
KEY RD/ WR RET
MC STOP
KEY MEM CONT
KEY RUN(I)\(EX\DEP)
KEY IO RESET
MR PWR CLR
KEY START\ CONT\ READ IN
KEY EXECUTE
MC RS T1
SBR 1260 1L4
KEY RD/WR
KEY RD
KEY WR
KEY MA←(Ø)
KEY MA←MASW (1)
KEY MA+1
KEY AR←(Ø)
KEY AR←DATASW (1)
KEY RD
KEY WR
KEY MA←MAS
KEY (EX\ DEP) NXT
KEY EXECUTE\ DP\ DPNXT
KEY EX\ EX NXT
KEY DP\ DP NXT
MR CLR A
MR CLR
KEY MANUAL
KEY MEM CONT
MR PWR CLR
UUO T1
[AT Ø
XCT T Ø
IT Ø
CPA
IDB — DATA 1
KEY RIM SBR
READ IN MODE MAINT SW
KEY CLR RIM
KEY READ IN
IT1A
∿MA 18-31=Ø



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

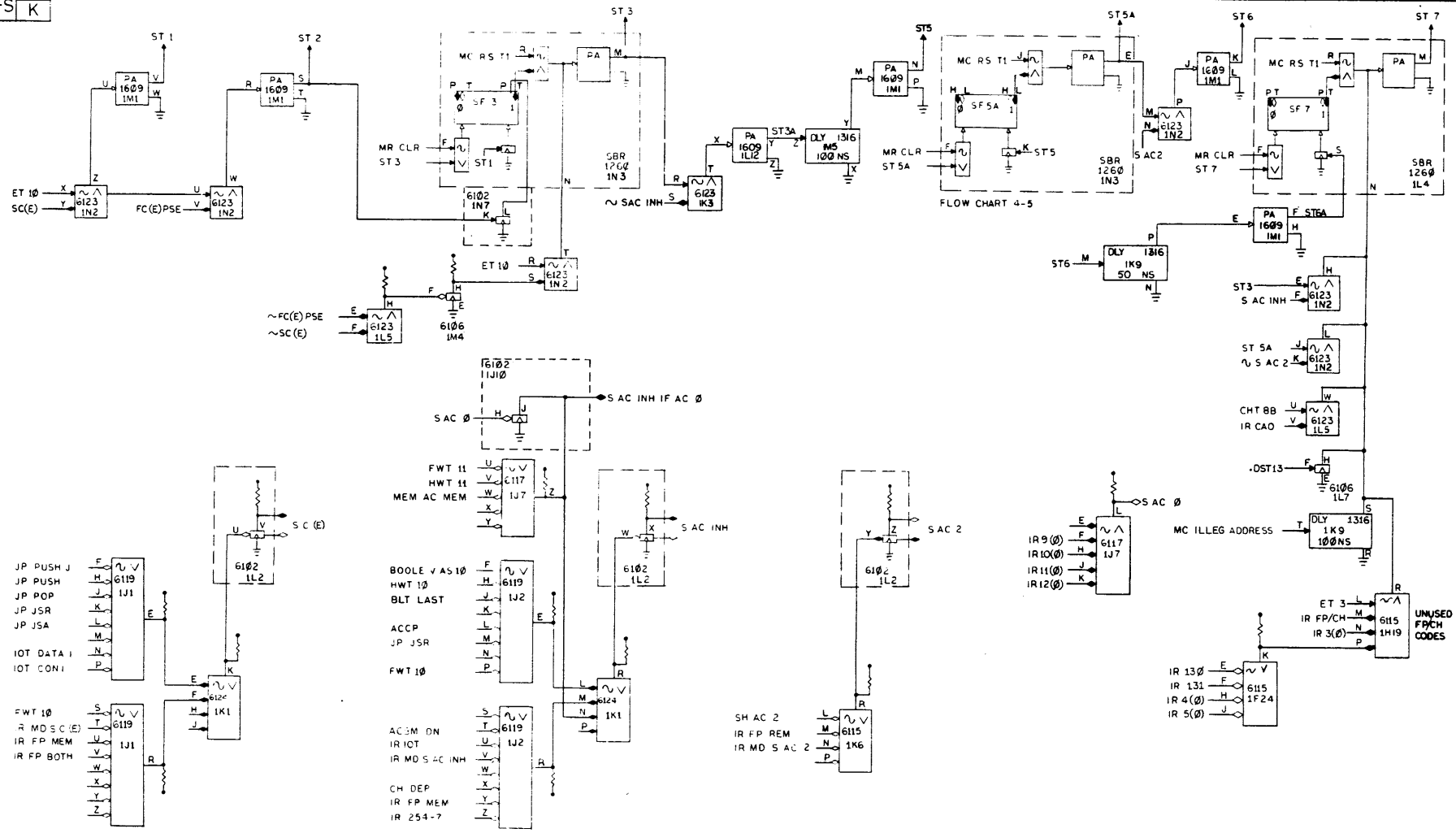


F - Fetch Cycle (BS-D-166-0-F)

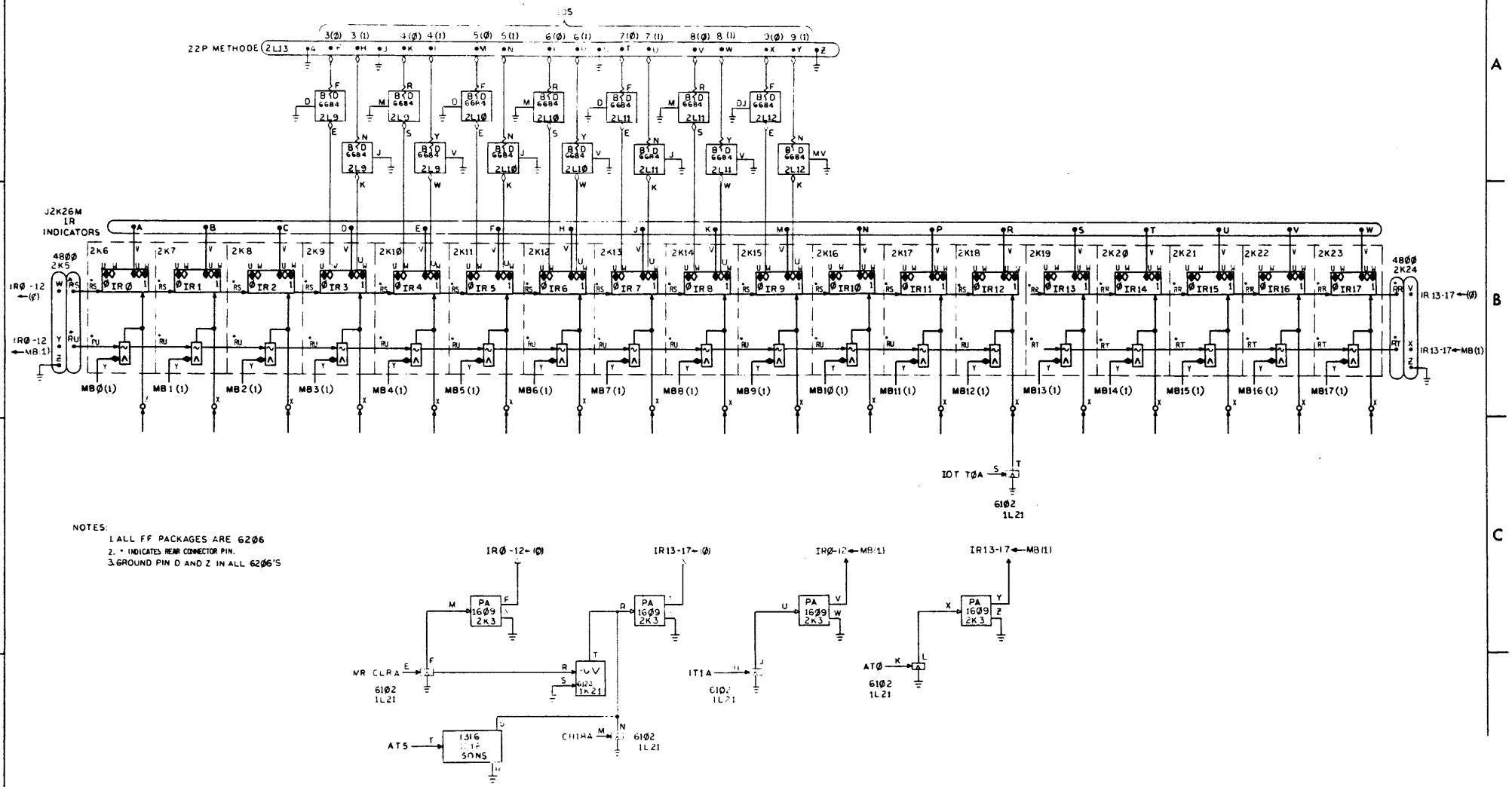


FLOW CHART 4-5

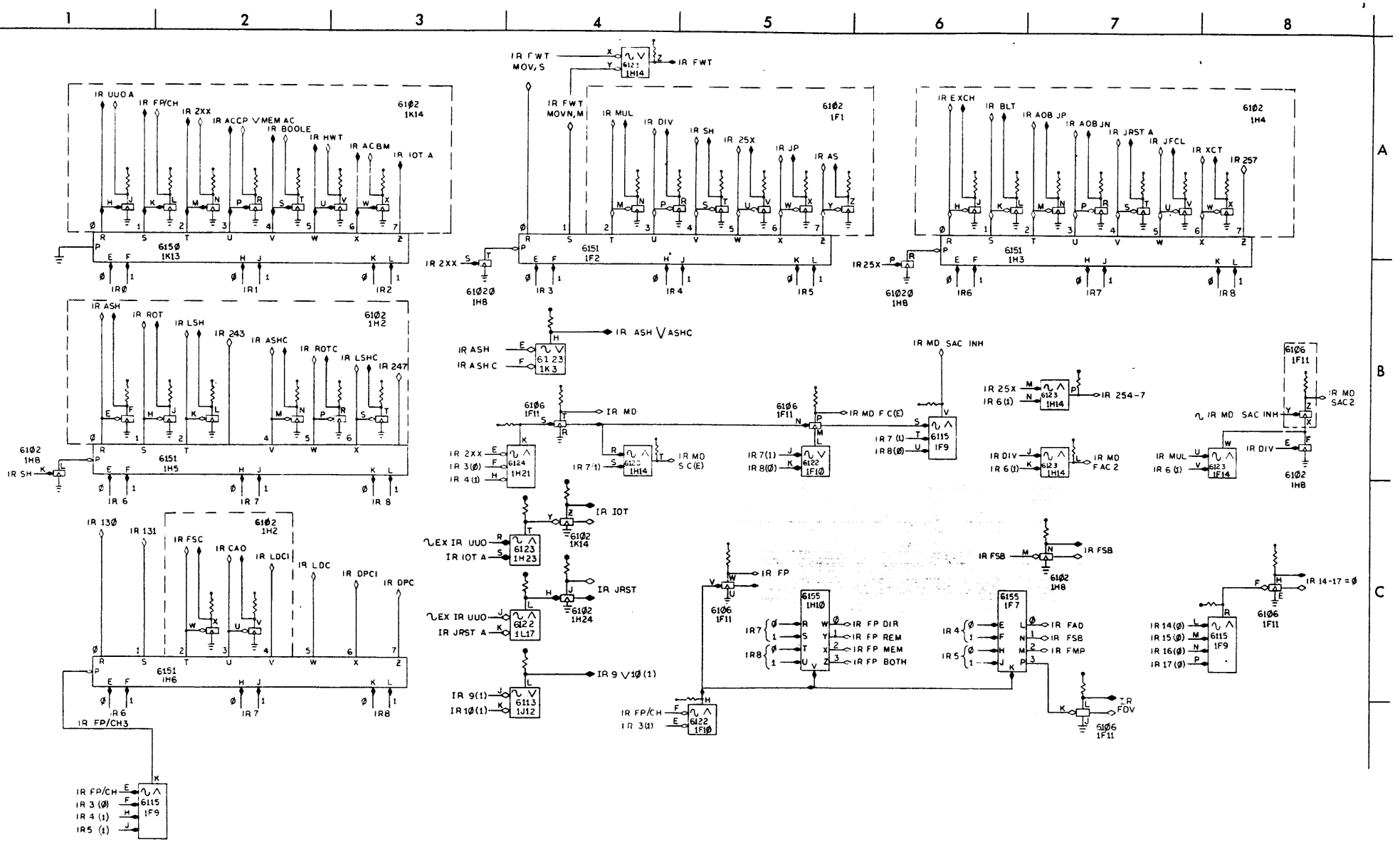
D-166-Ø-S REV LTR K



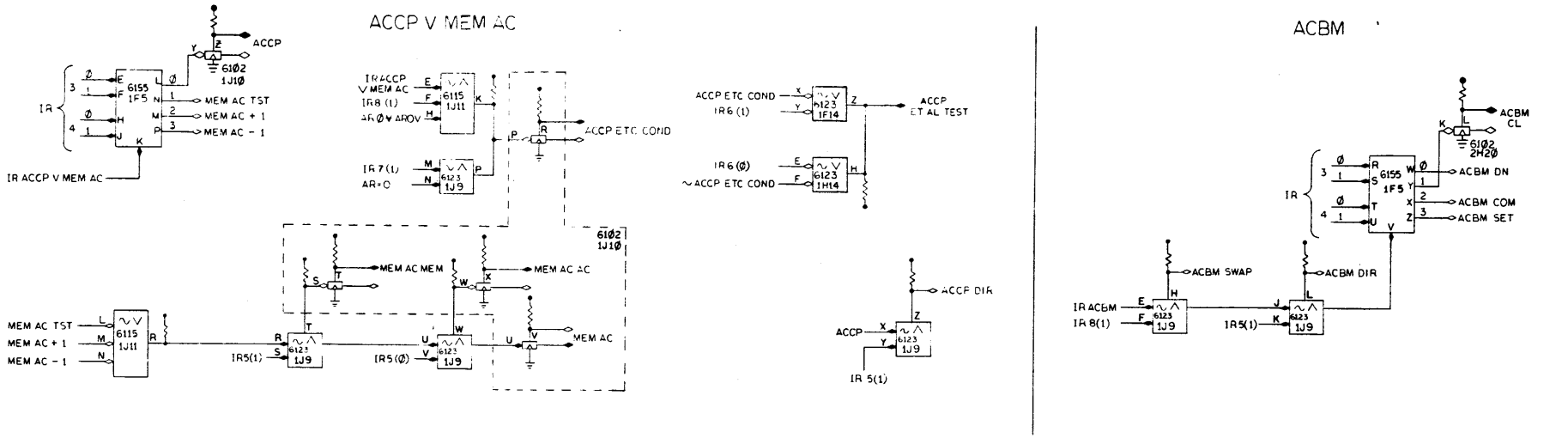
S - Store Cycle (BS-D-166-0-S)



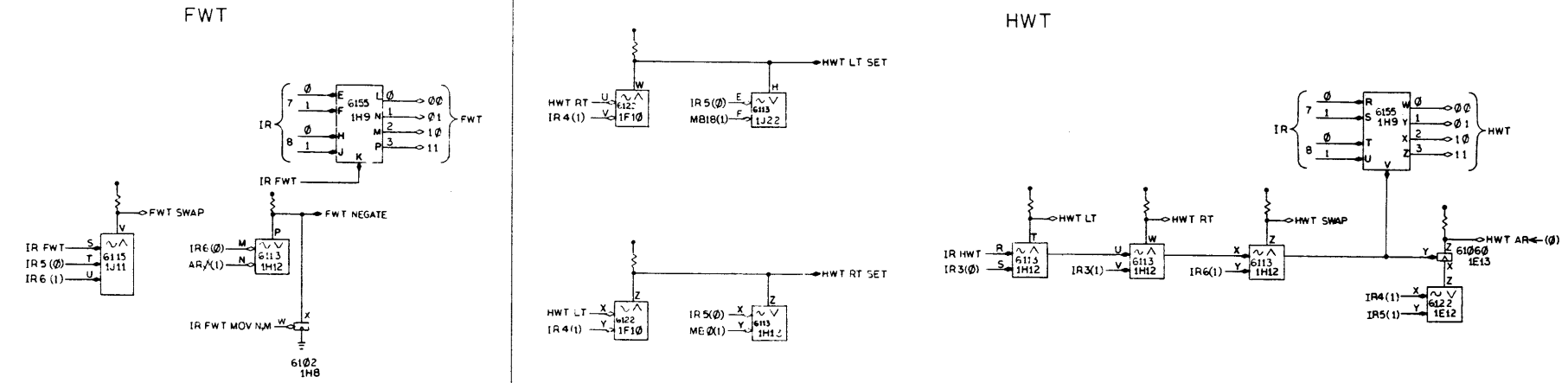
IR - Instruction Register
(BS-D-166-0-IR-1)



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



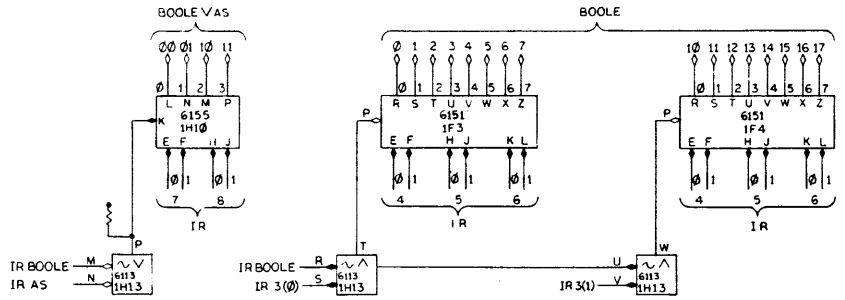
FLOW CHART 4-6



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

BOOLE

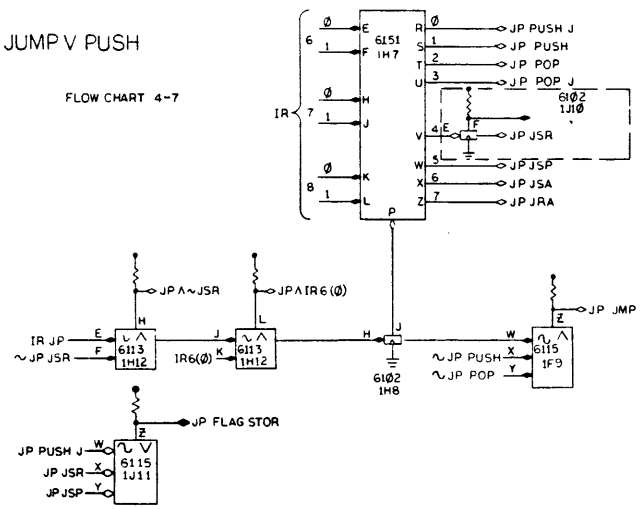
		RESULT BIT FOR OPERAND BIT CONFIGURATION				ET0 T1	ET1 T2		ET4 T3
		MEM AC 0 0	MEM AC 0 1	MEM AC 1 0	MEM AC 1 1	AR CLR	AR COM	AR←MB(0)	AR←MB(1)	AR←MB ▼	AR COM
SETZ	0	00	0	0	0	0	✓				
AND	MEM ∧ AC	01	0	0	0	1		✓		✓	
ANDCA	MEM ∧ ∼AC	02	0	0	1	0		✓		✓	
SETM	MEM	03	0	0	1	1	✓			✓	
ANDCM	∼MEM ∧ AC	04	0	1	0	0		✓		✓	✓
SETA	AC	05	0	1	0	1					
XOR	MEM ∨ AC	06	0	1	1	0				✓	
IOR	MEM ∨ AC	07	0	1	1	1				✓	
ANDCB	∼MEM ∧ ∼AC	10	1	0	0	0				✓	✓
EQV	MEM ≡ AC	11	1	0	0	1		✓			✓
SETCA	∼AC	12	1	0	1	0		✓			
ORCA	MEM ∨ ∼AC	13	1	0	1	1		✓		✓	
SETCM	∼MEM	14	1	1	0	0	✓				✓
ORCM	∼MEM ∨ AC	15	1	1	0	1		✓	✓		✓
ORCB	∼MEM ∨ ∼AC	16	1	1	1	0		✓	✓		✓
SETO	1	17	1	1	1	1	✓				✓



FLOW CHART 4-8

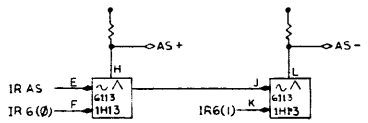
JUMP V PUSH

FLOW CHART 4-7



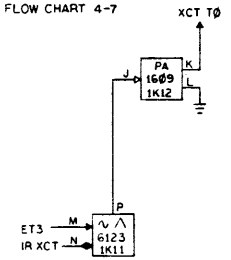
AS

FLOW CHART 4-8

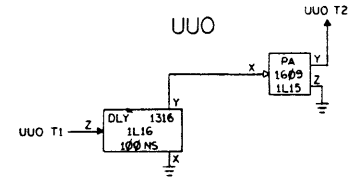


XCT

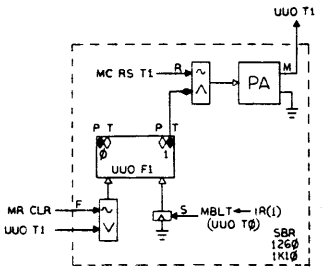
FLOW CHART 4-7



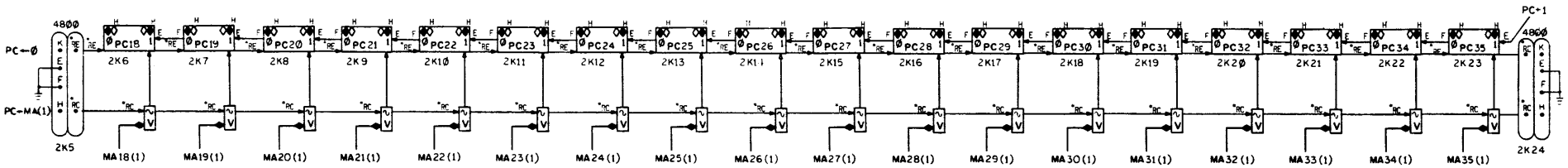
UUO



FLOW CHART 4-7



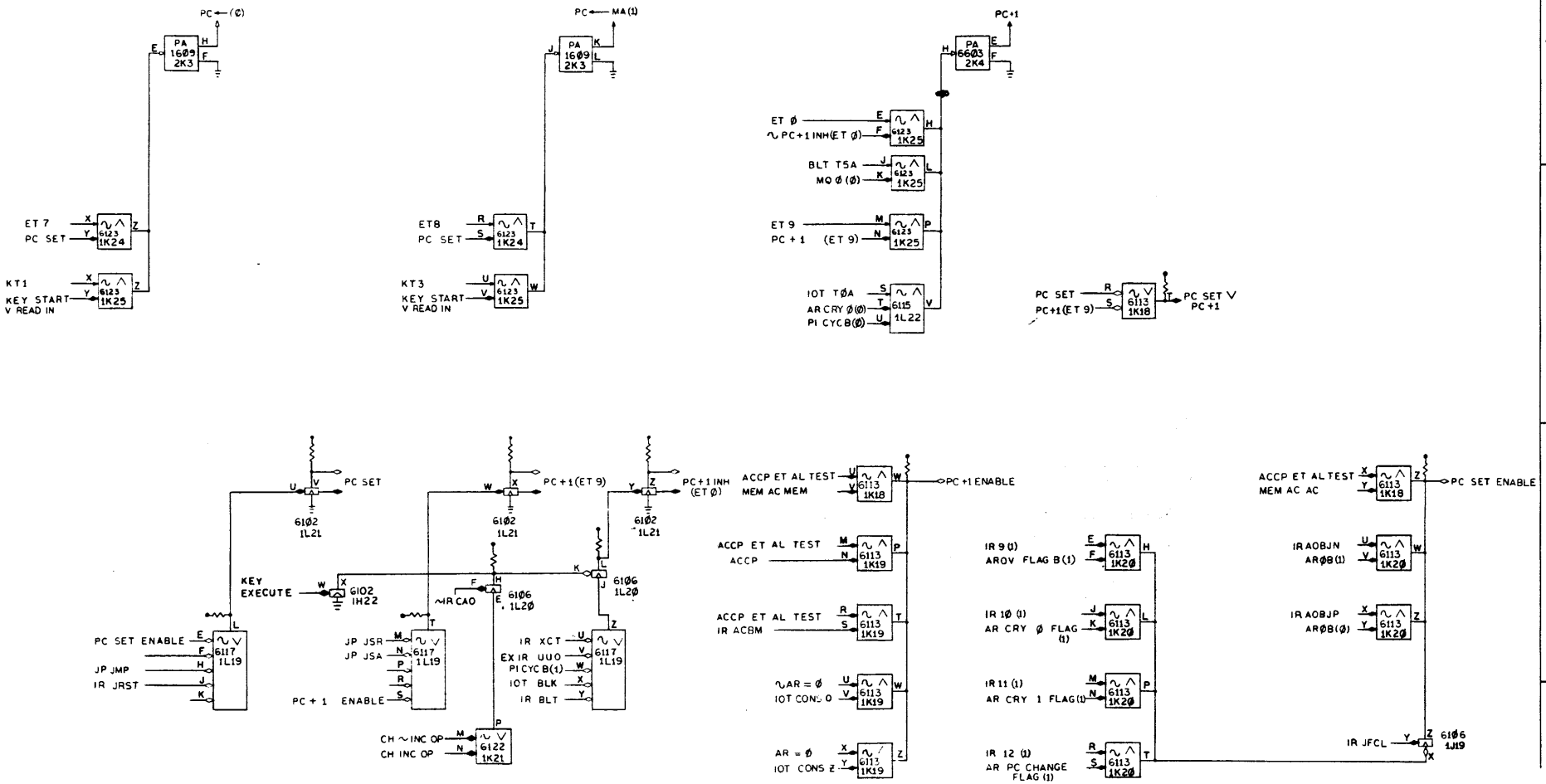
BOOLE, JP, AS, XCT, UUO
(BS-D-166-0-INS-2)



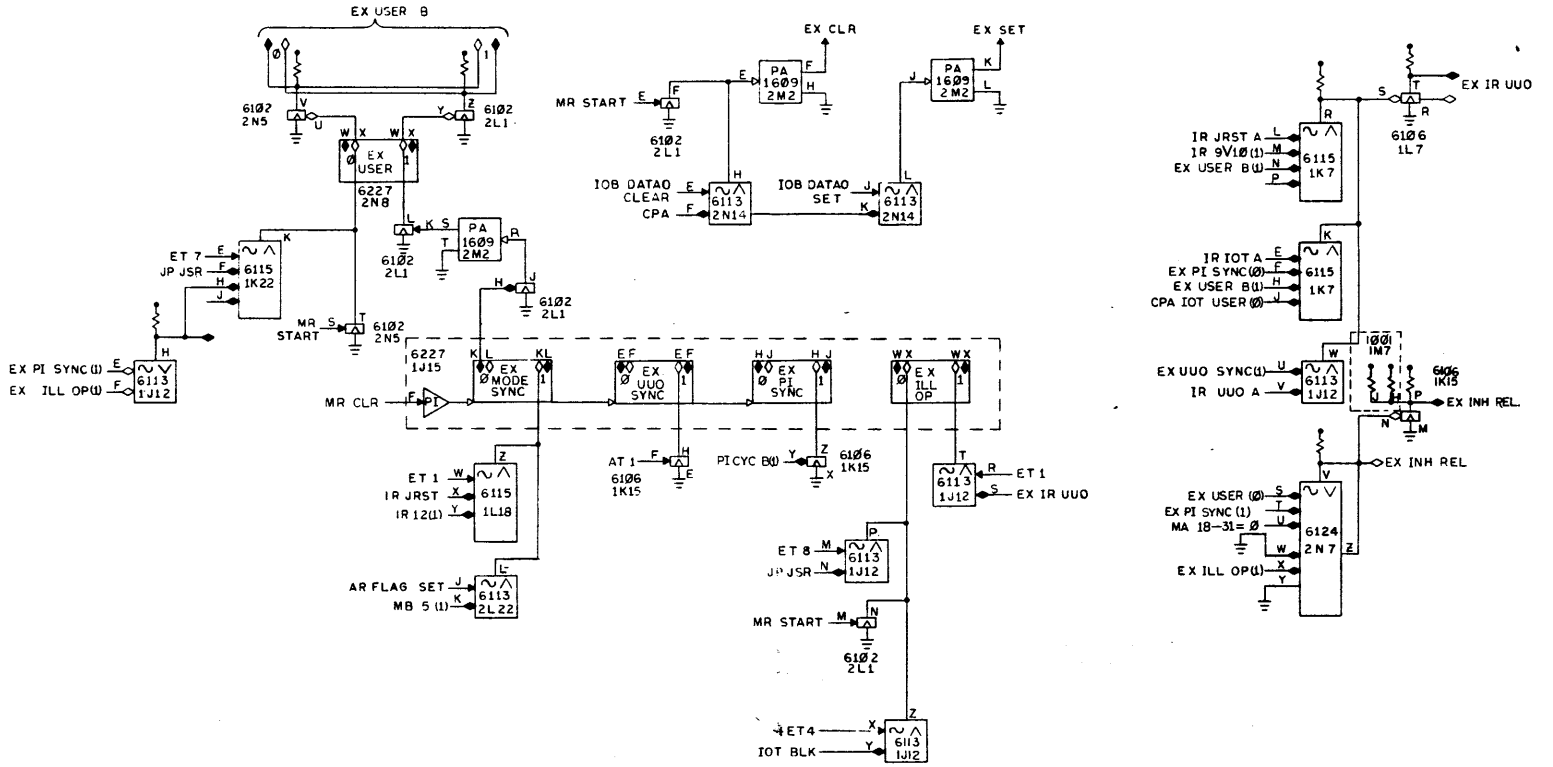
NOTES:

1. ALL FF PACKAGES ARE 6284
2. # INDICATES REAR CONNECTOR PIN
3. GROUND PIN D AND Z IN ALL 6206'S

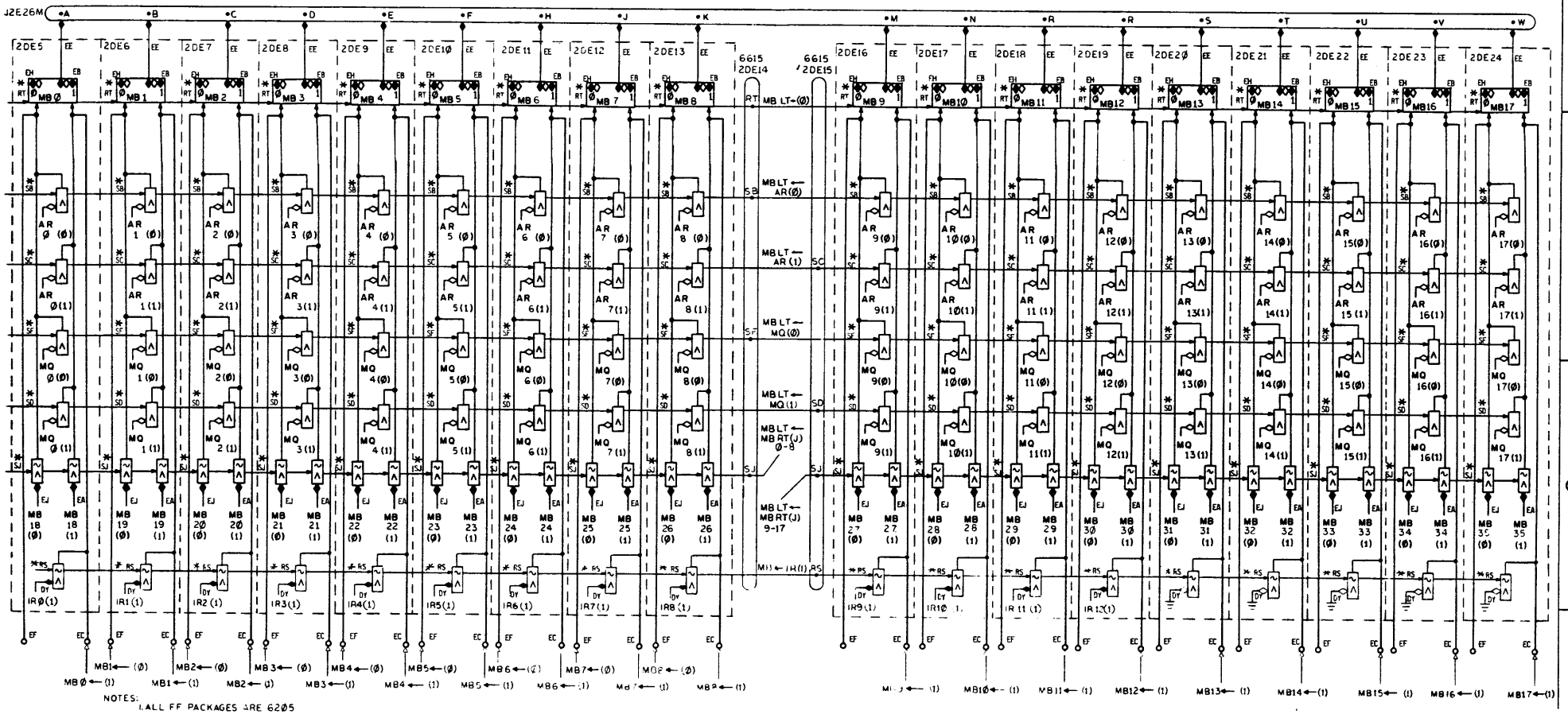
PC - Program Counter (BS-D-166-0-PC-2)



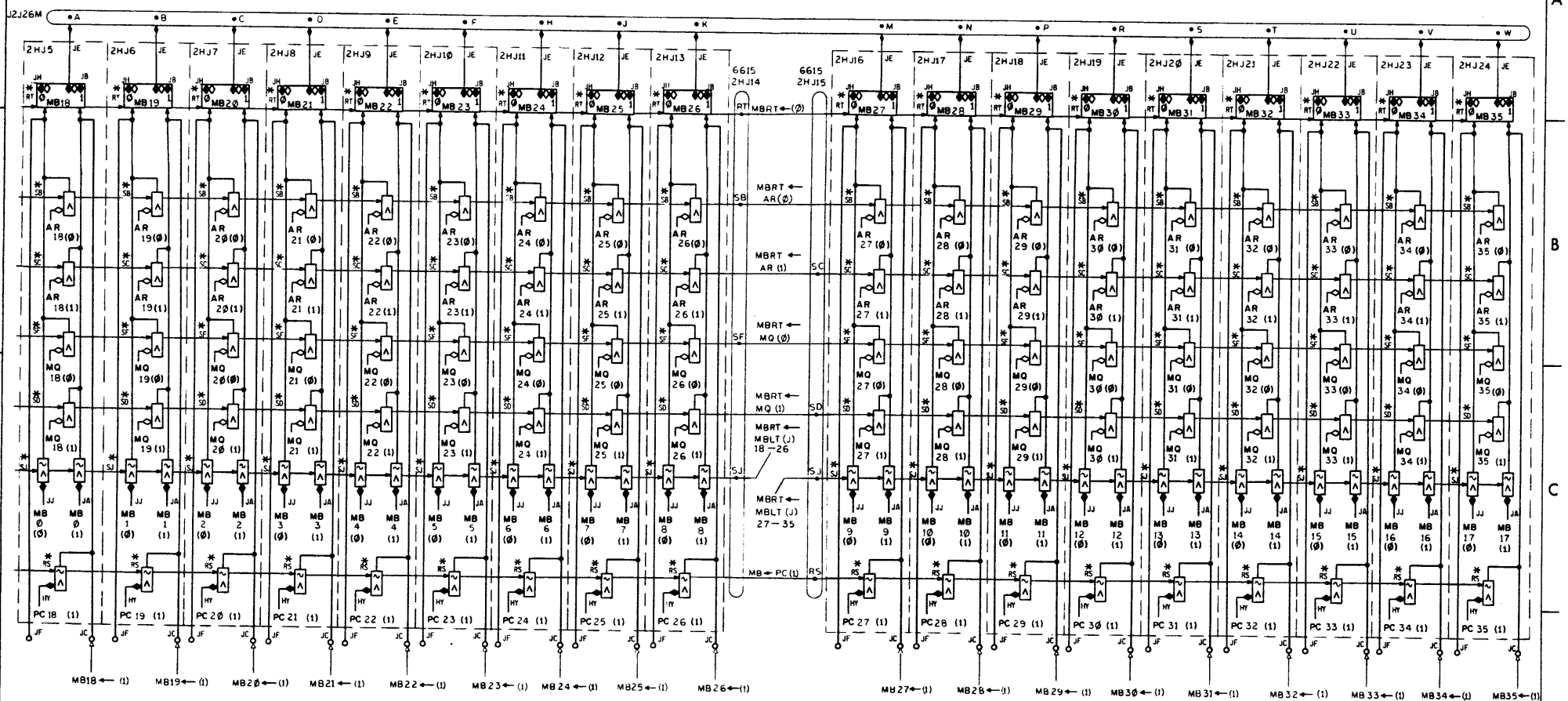
PC Control (BS-D-166-0-PC-1)



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



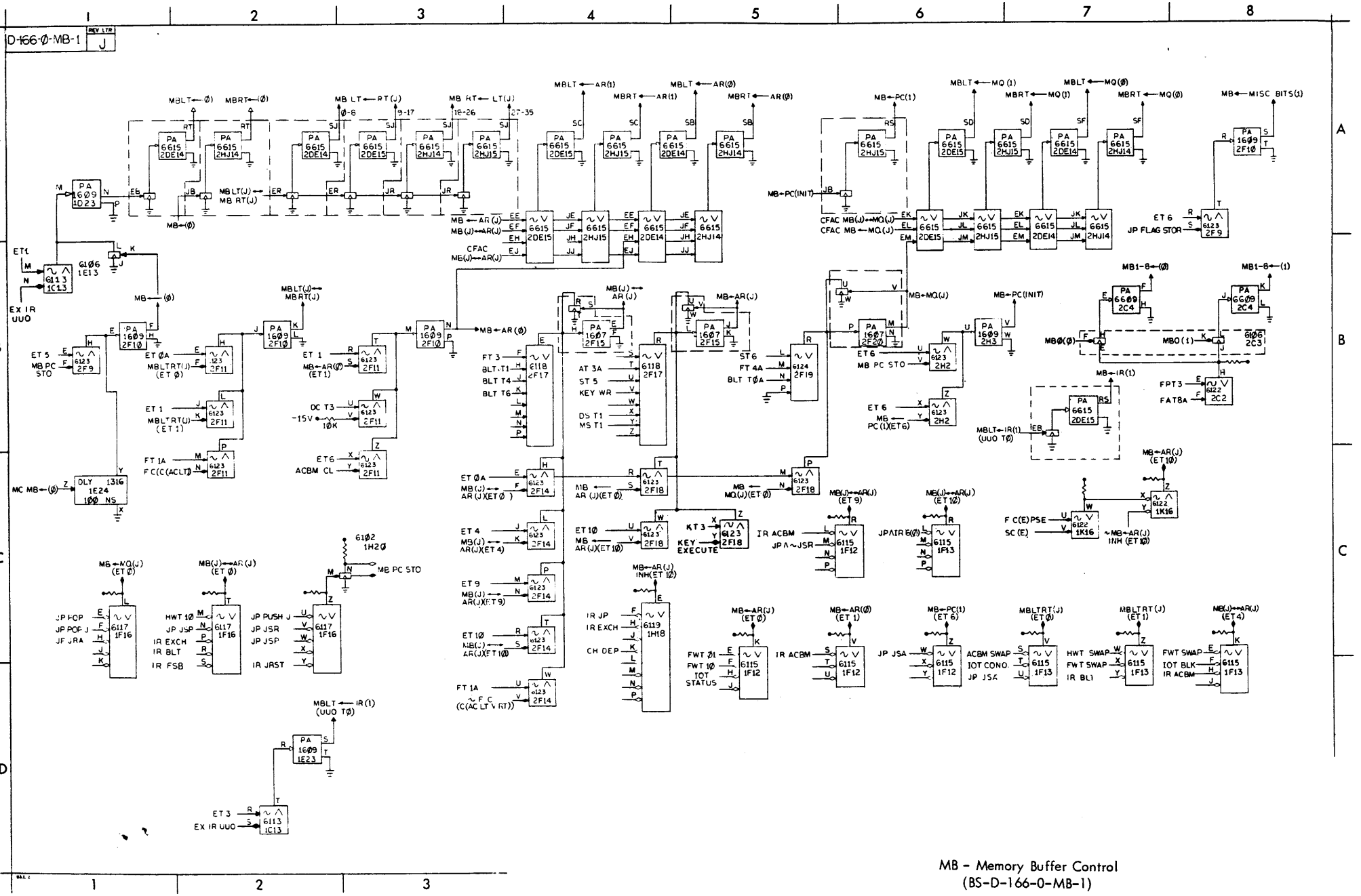
MBLT Register 0-17 (BS-D-166-0-MB-2)



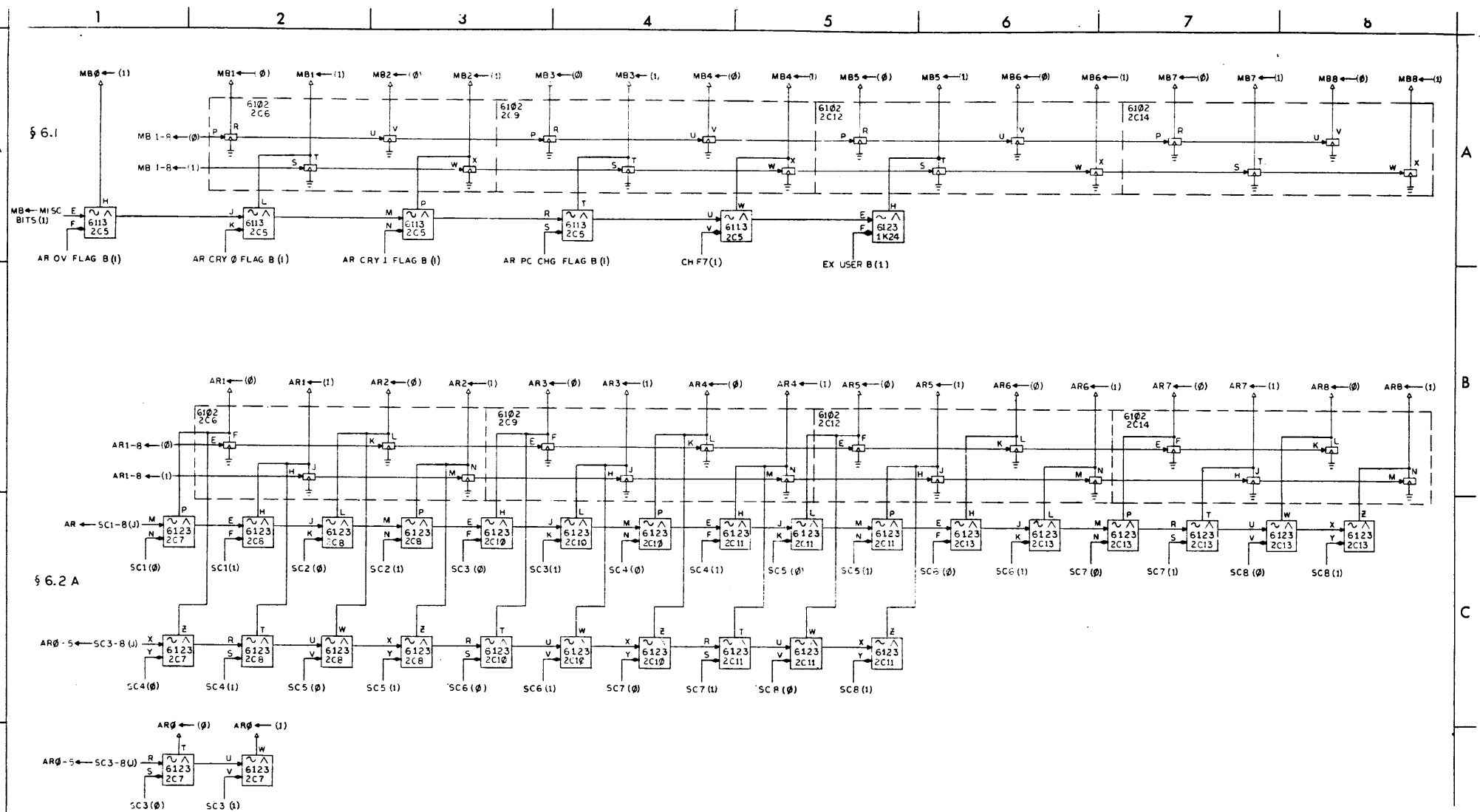
NOTES

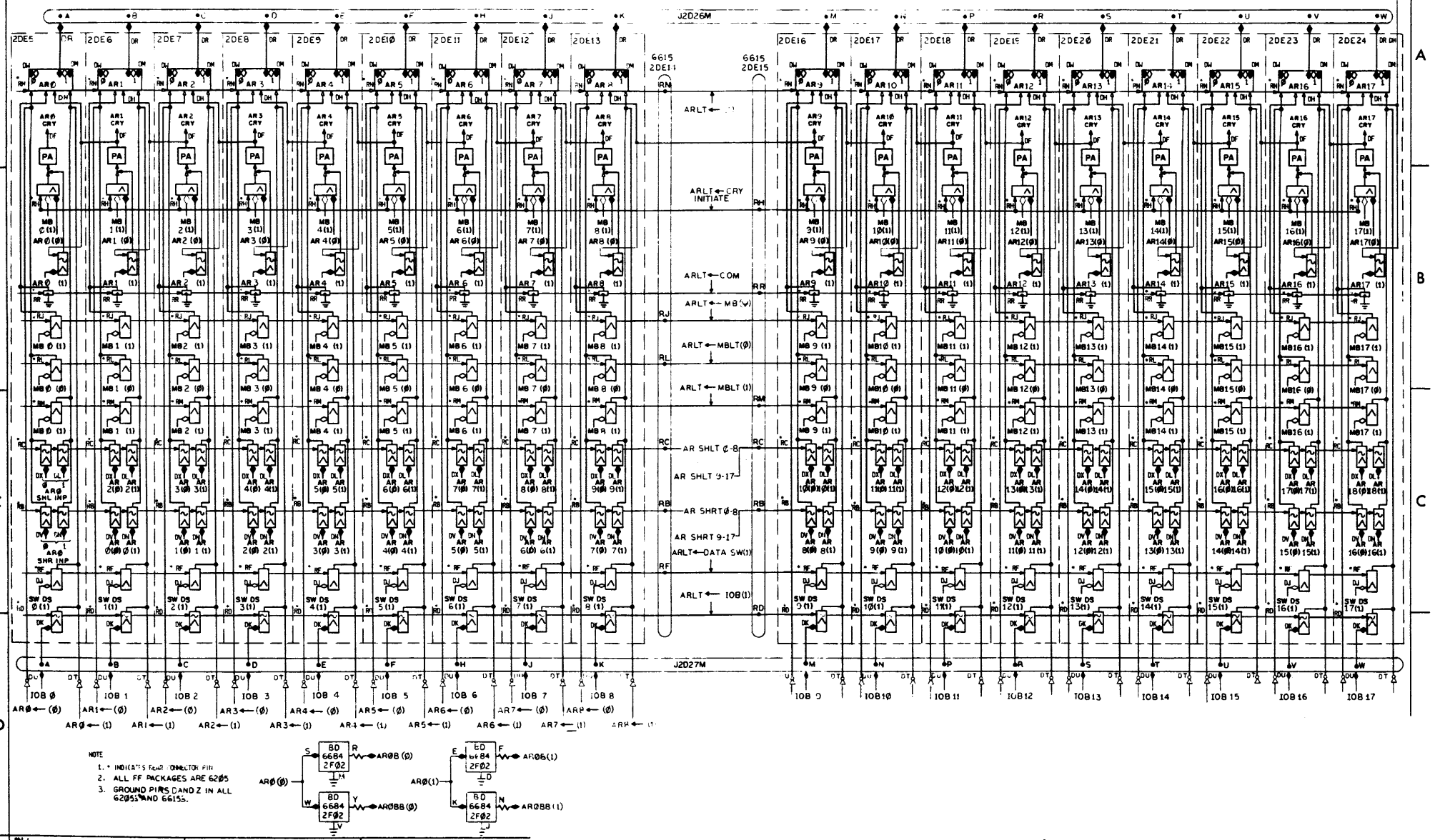
1. ALL FF PACKAGES ARE 6205
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D, P AND Z IN ALL 6205's AND 6615's.

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



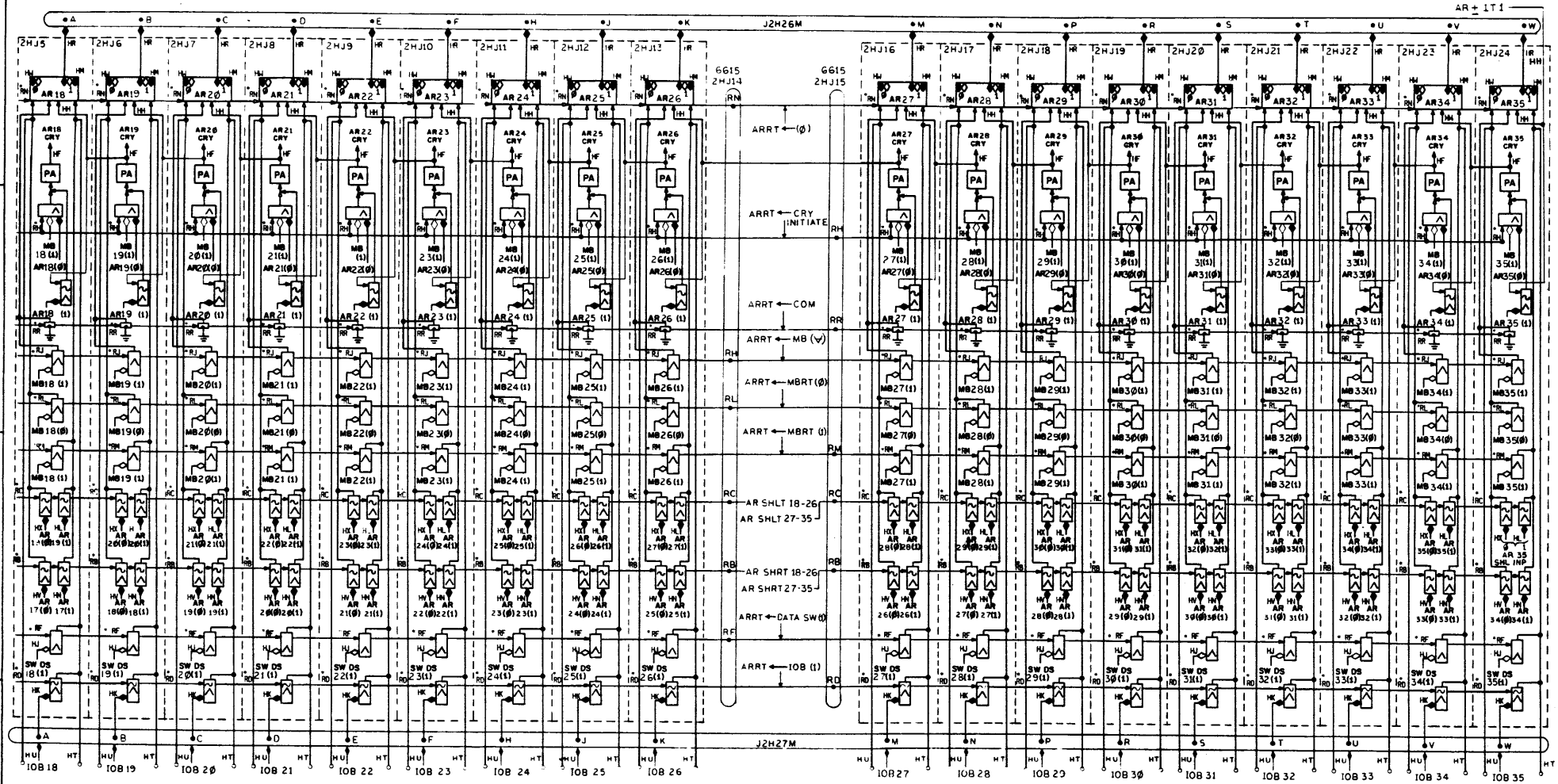
MB - Memory Buffer Control
(BS-D-166-0-MB-1)


§ 6.1
§ 6.2 A
MB1-8 ← (Ø)
MB1-8 ← (1)
AR1-8 ← (Ø)
AR1-8 ← (1)
AR ← SC1-8(J)
ARØ-5 ← SC3-8(J)
MB ← MISC BITS (1)
AR OV FLAG B (1)
AR CRY Ø FLAG B (1)
AR CRY 1 FLAG B (1)
AR PC CHG FLAG B (1)
CH F7(1)
EX USER B(1)
6102 2C6
6102 2C9
6102 2C12
6102 2C14
6113 2C5
6123 1K24
6123 2C7
6123 2C8
6123 2C10
6123 2C11
6123 2C13



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

D-166-Ø-AR-5 REV LTR B

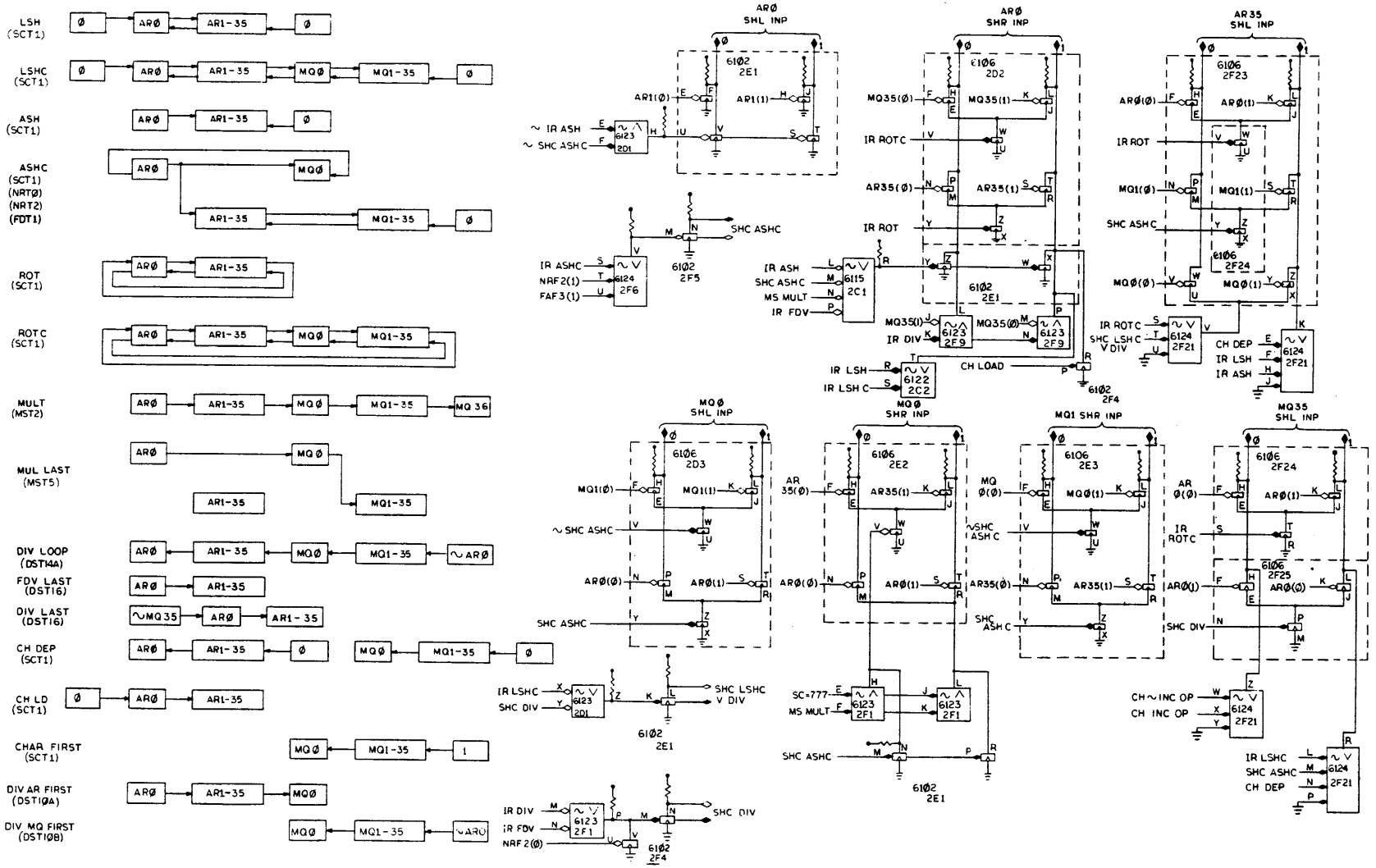


NOTES:

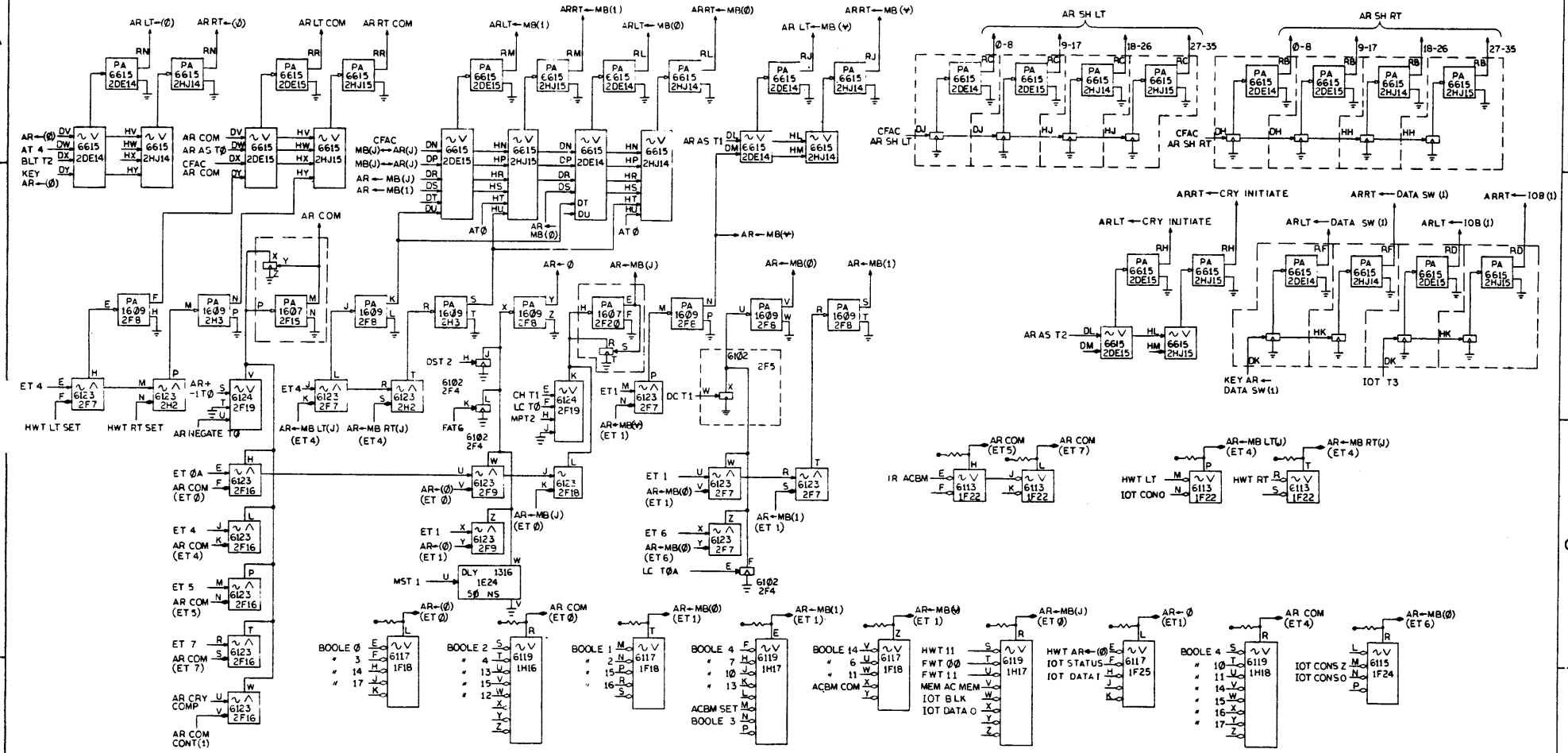
1. ALL FF PACKAGES ARE 62Ø5
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D AND Z IN ALL 62Ø5's AND 6615's.

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

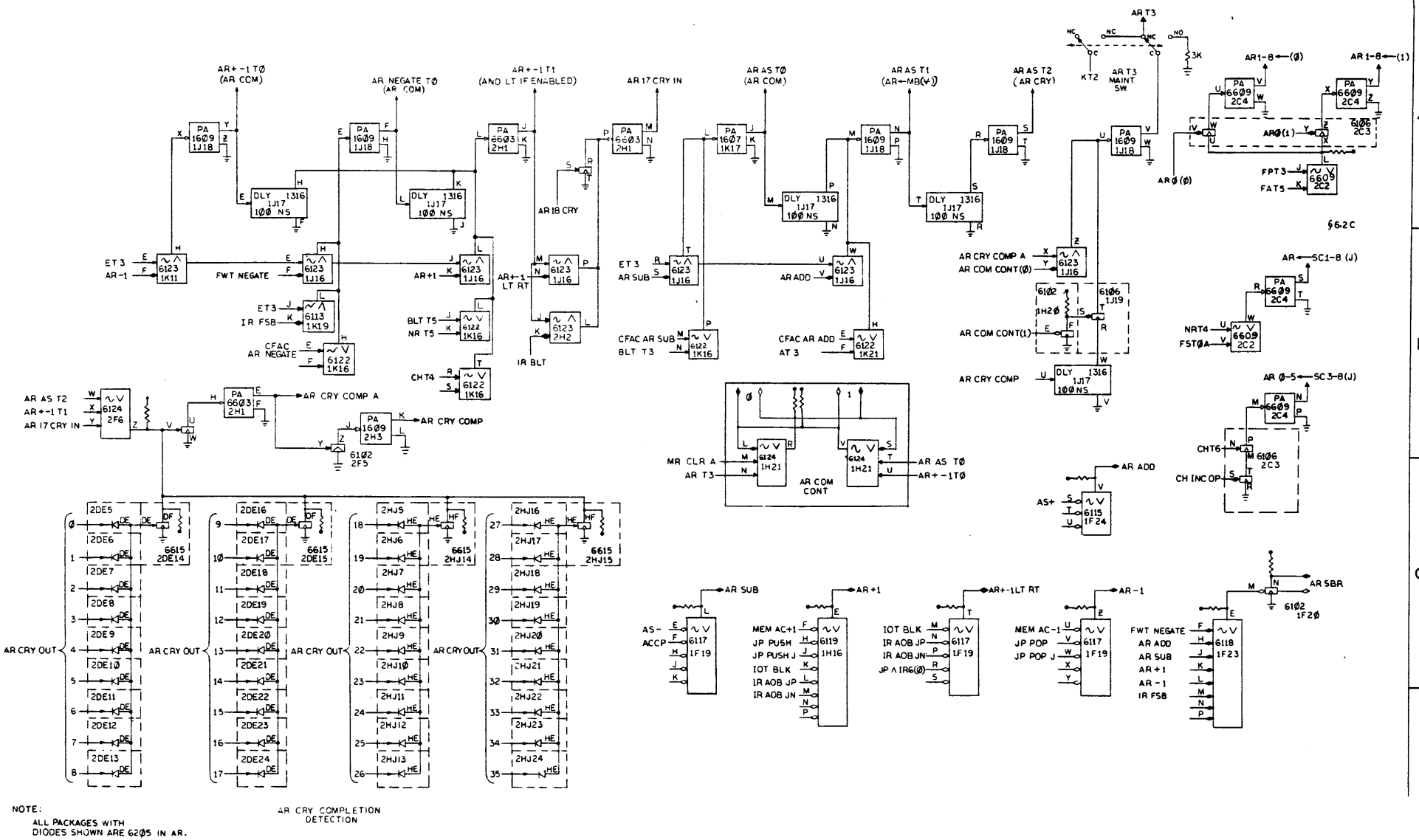
D-166-Ø-ARMQ REV LTR D



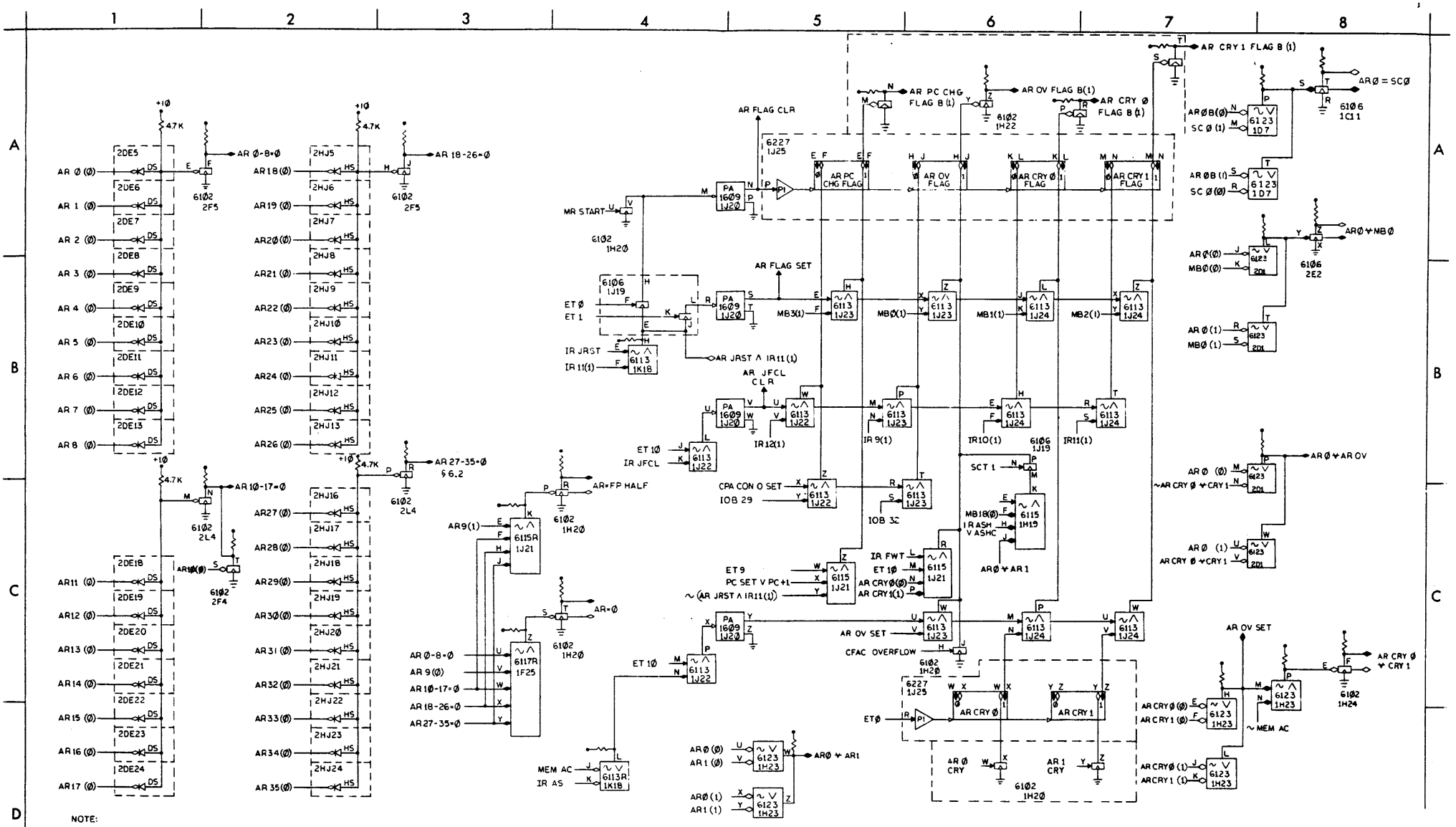
AR, MQ Shift Connections (BS-D-166-0-ARMQ)



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

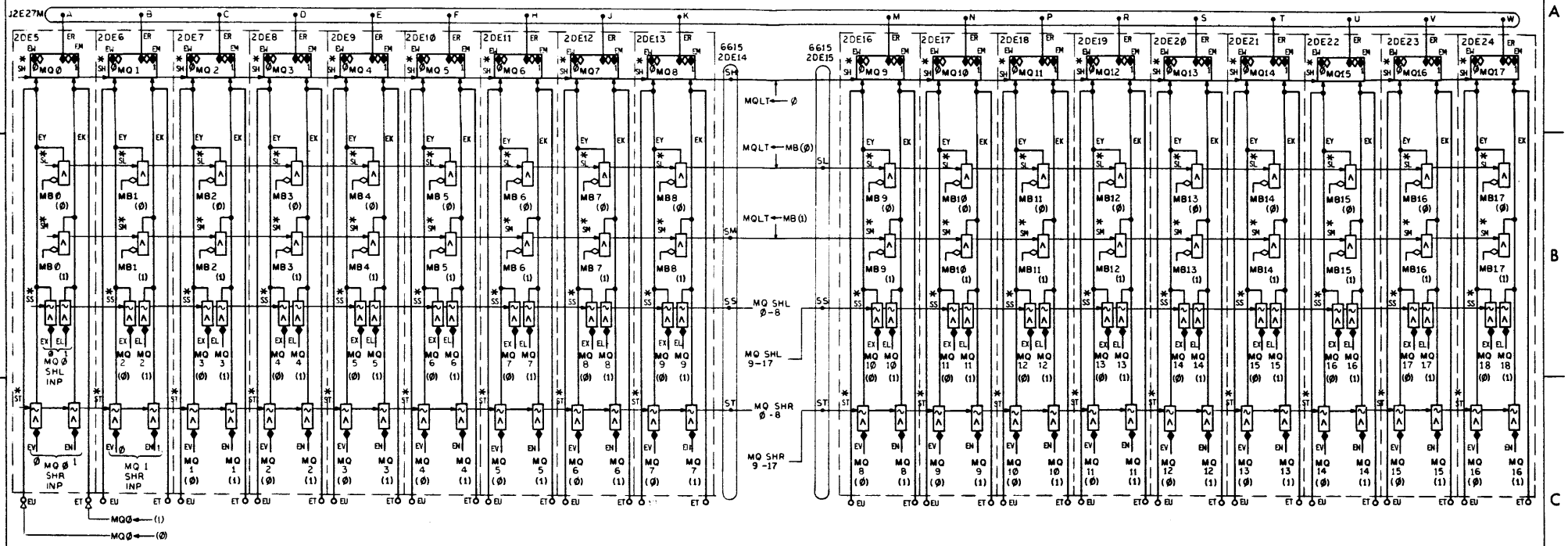


AR Control 2 (BS-D-166-0-AR-2)



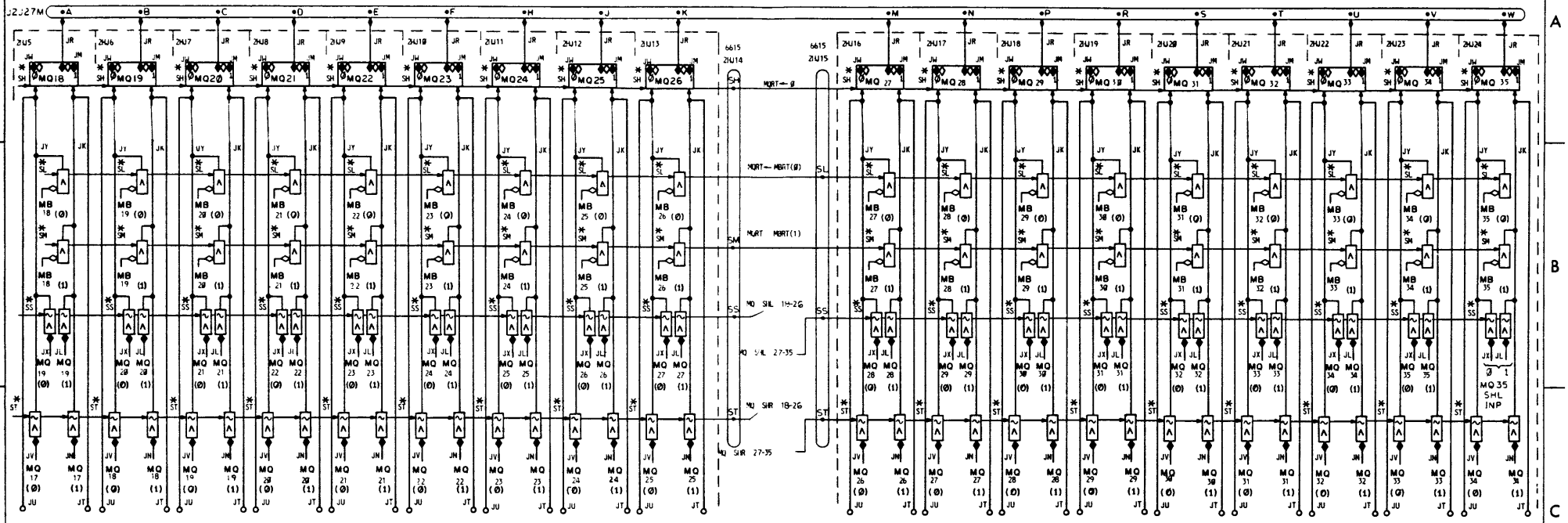
NOTE:
ALL PACKAGES WITH DIODES SHOWN ARE 6205 IN AR.

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



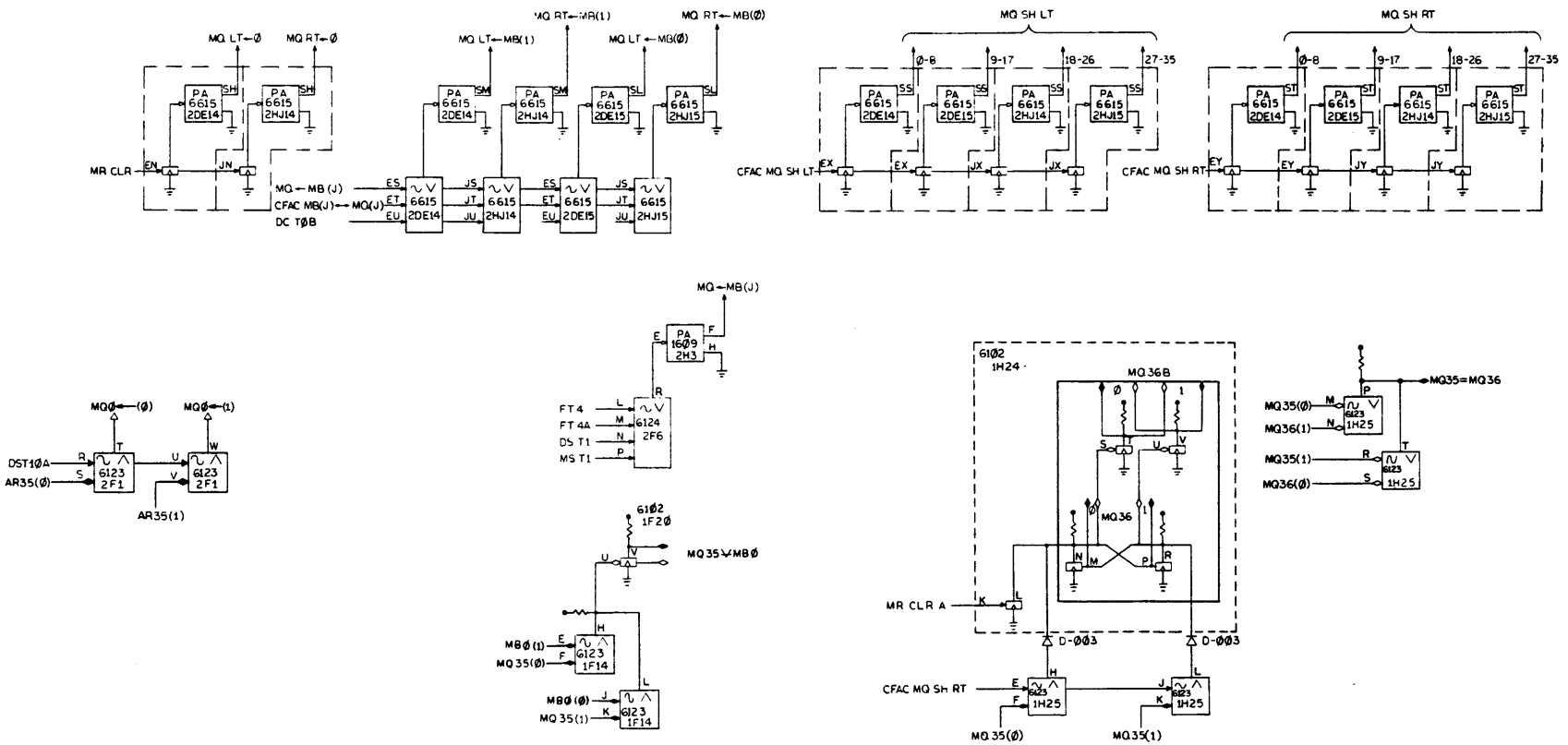
NOTES:
1. ALL FF PACKAGES ARE 6205
2. * INDICATES REAR CONNECTOR PIN
3. GROUND PIN D,P AND Z IN ALL 6205'S AND 6615'S.

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

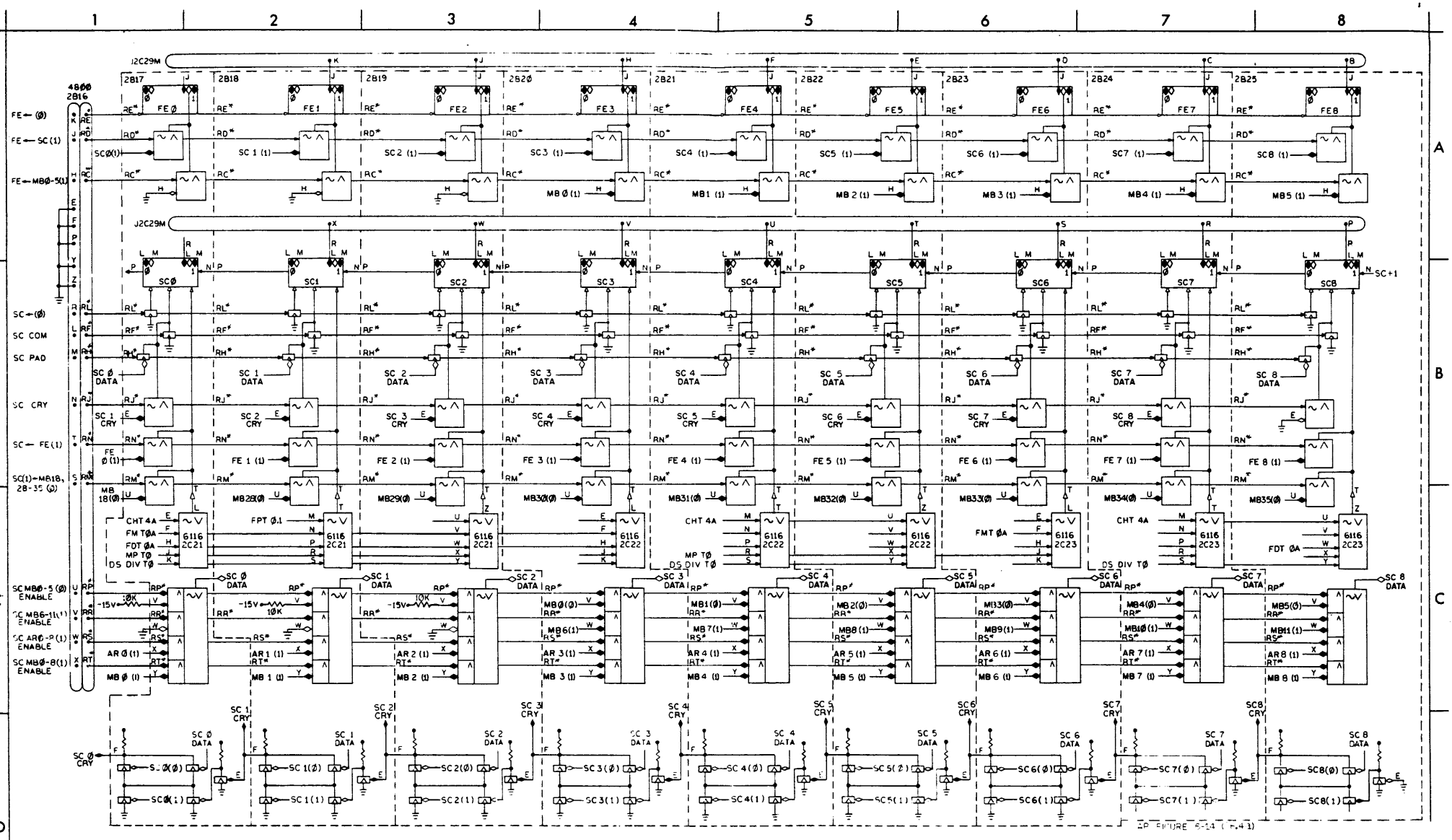


NOTE:

1. ALL FF PACKAGES ARE 6205.
2. * INDICATES REAR CONNECTOR PIN.
3. GROUND PIN D,P AND Z IN ALL 6205s AND 6615s.

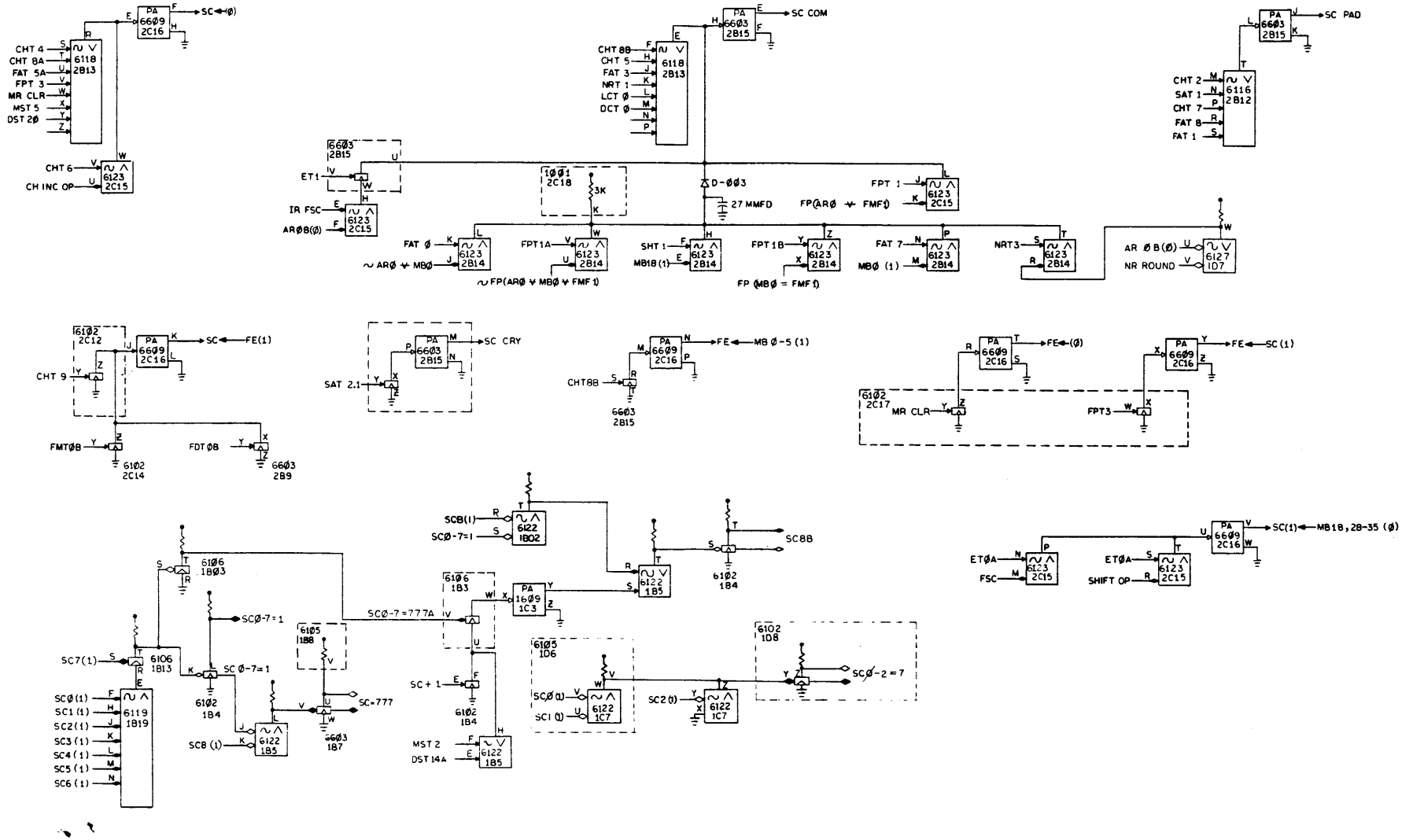


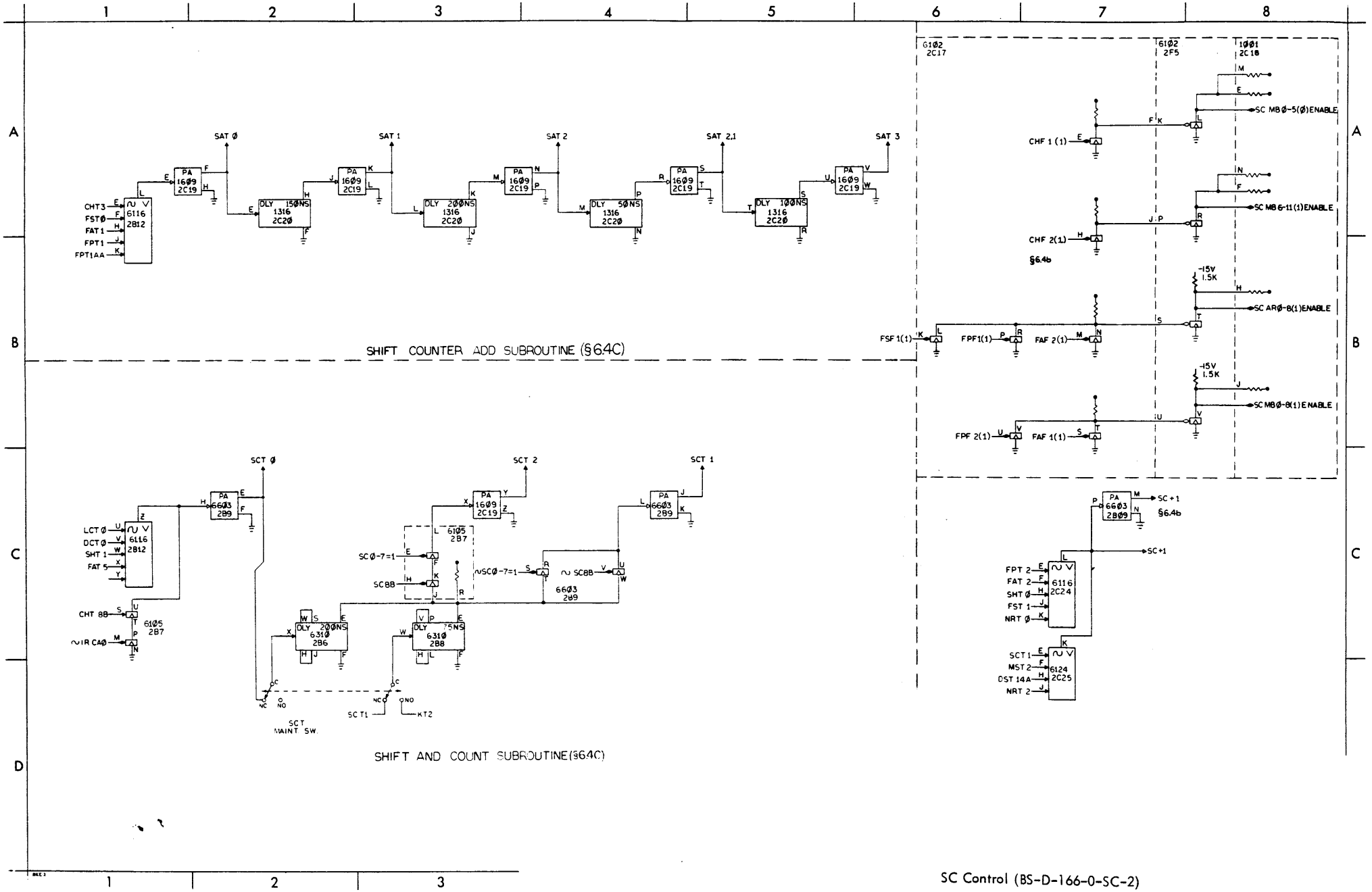
MQ - Multiplier Quotient Control
(BS-D-166-0-MQ-1)

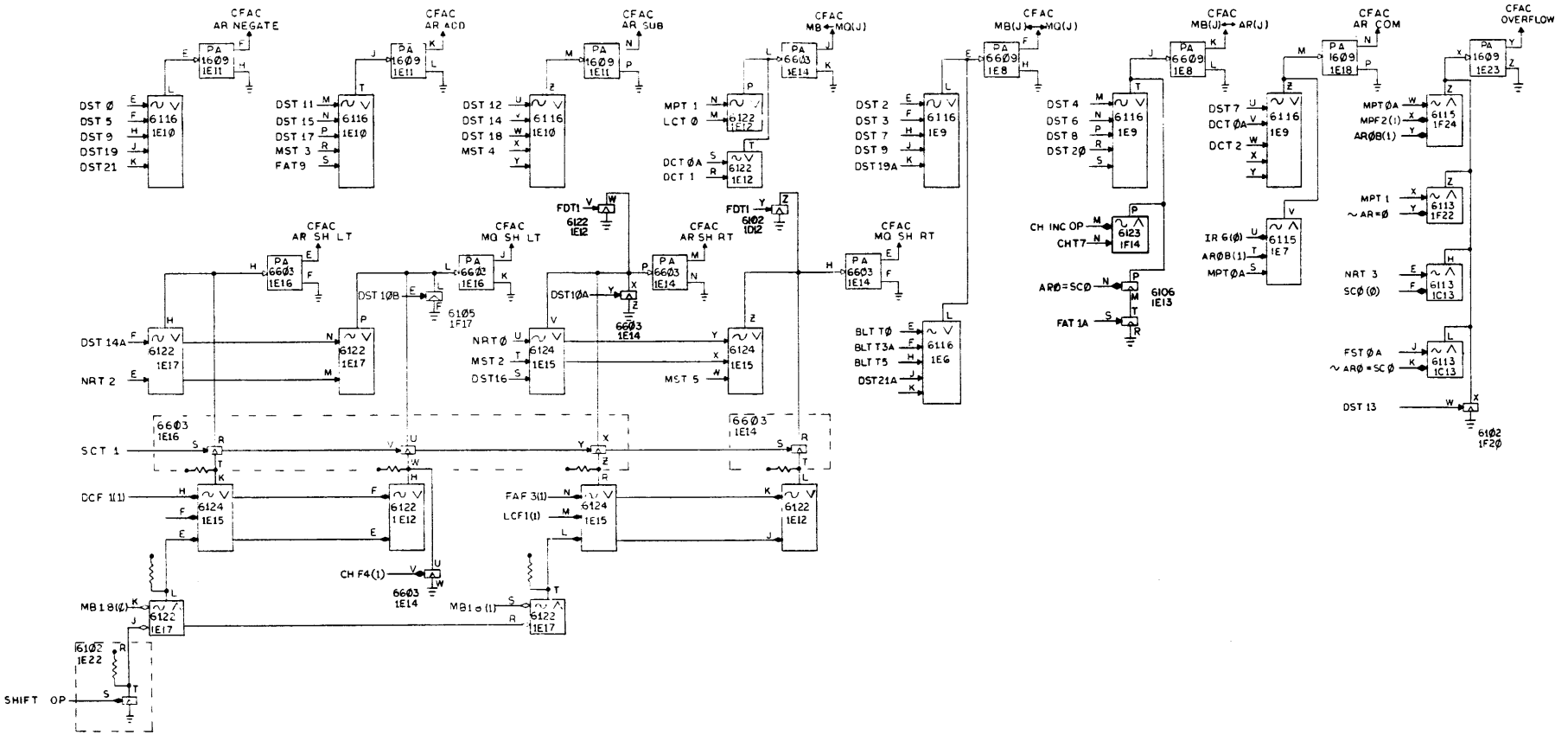


NOTE:
1. UNLESS OTHERWISE INDICATED ALL PACKAGES ARE 6203s.
2. * INDICATES REAR CONNECTOR PIN.

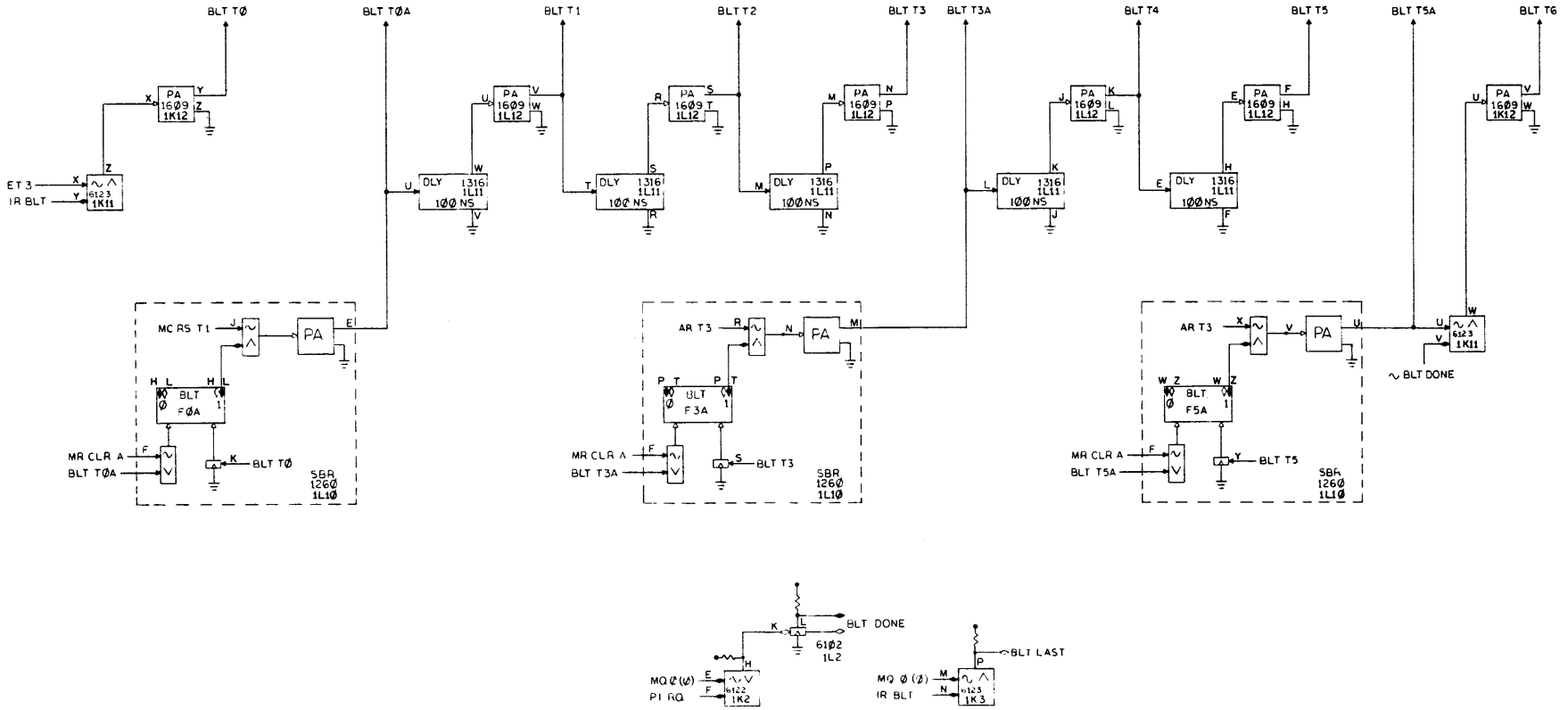
SC, FE - Shift Counter, Floating Exponent
(BS-D-166-0-SCFE)


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

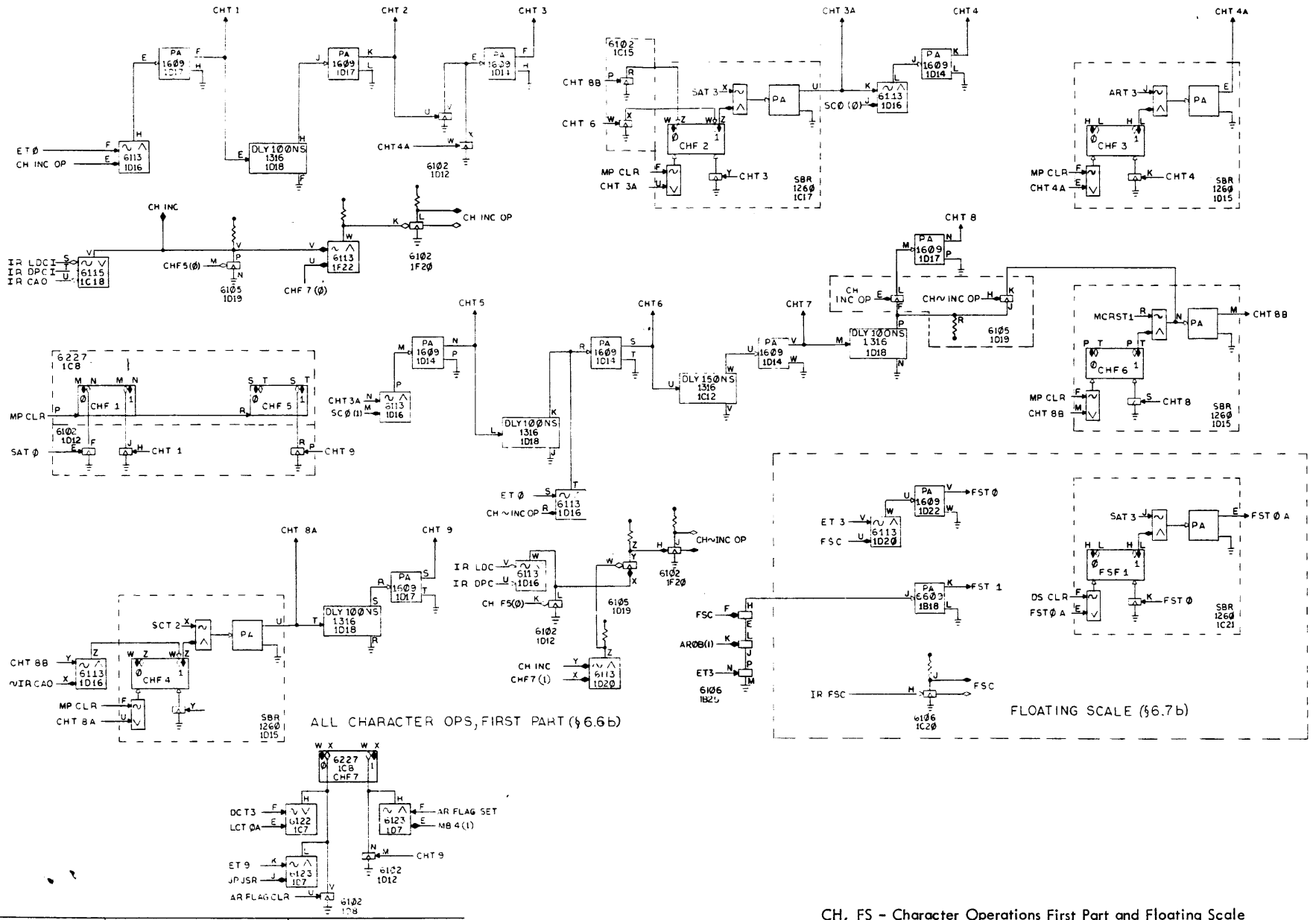

SAT Ø
SAT 1
SAT 2
SAT 2,1
SAT 3
CHT3
FSTØ
FAT 1
FPT 1
FPT1AA
6116 2B12
PA 16Ø9 2C19
DLY 150NS 1316 2C2Ø
DLY 2ØØNS 1316 2C2Ø
DLY 5ØNS 1316 2C2Ø
DLY 1ØØNS 1316 2C2Ø
SHIFT COUNTER ADD SUBROUTINE (§6.4C)
G1Ø2 2C17
6102 2F5
1Ø1 2C18
CHF 1 (1)
CHF 2(1)
§6.4b
SC MBØ-5(Ø)ENABLE
SC MB 6-11(1)ENABLE
SC ARØ-8(1)ENABLE
SC MBØ-8(1)ENABLE
-15V 1.5K
FSF 1(1)
FPF1(1)
FAF 2(1)
FPF 2(1)
FAF 1(1)
SCT Ø
SCT 2
SCT 1
LCT Ø
DCT Ø
SHT 1
FAT 5
CHT 8B
IR CAØ
6105 2B7
6Ø3 2B9
SCØ-7=1
SC8B
DLY 2ØØNS 6310 2B6
DLY 75NS 6310 2B8
SCT MAINT SW.
SC T1
KT2
SC+1
§6.4b
FPT 2
FAT 2
SHT Ø
FST 1
NRT Ø
6116 2C24
SCT 1
MST 2
DST 14A
NRT 2
6124 2C25
SHIFT AND COUNT SUBROUTINE(§6.4C)



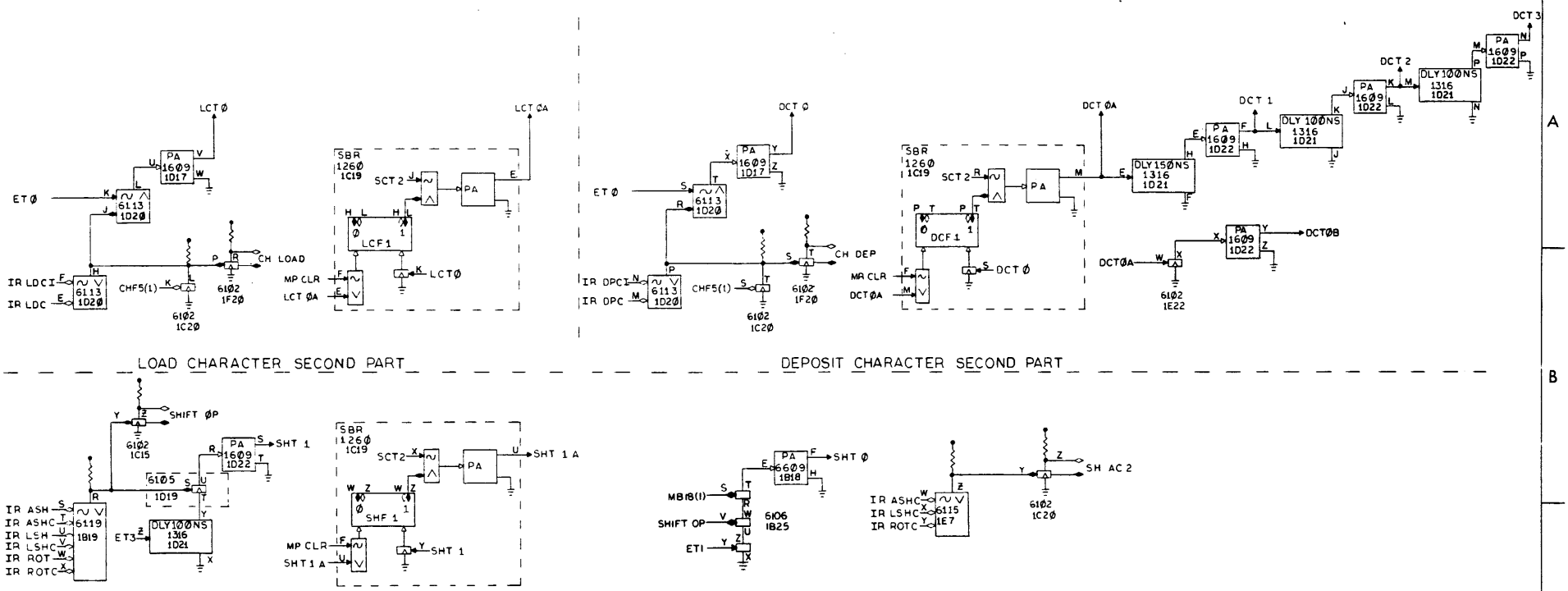
CFAC - Floating Arithmetic Connection
(BS-D-166-0-CFAC)



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

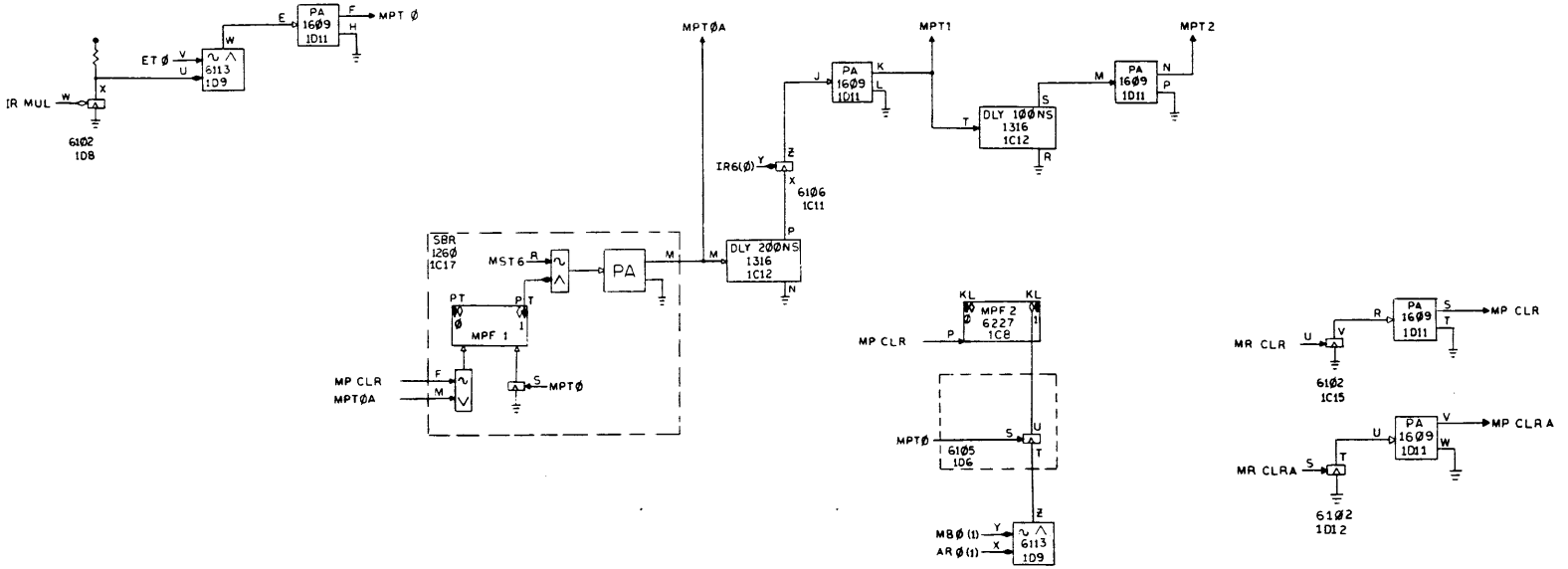


CH, FS - Character Operations First Part and Floating Scale
(BS-D-166-0-CHFS)



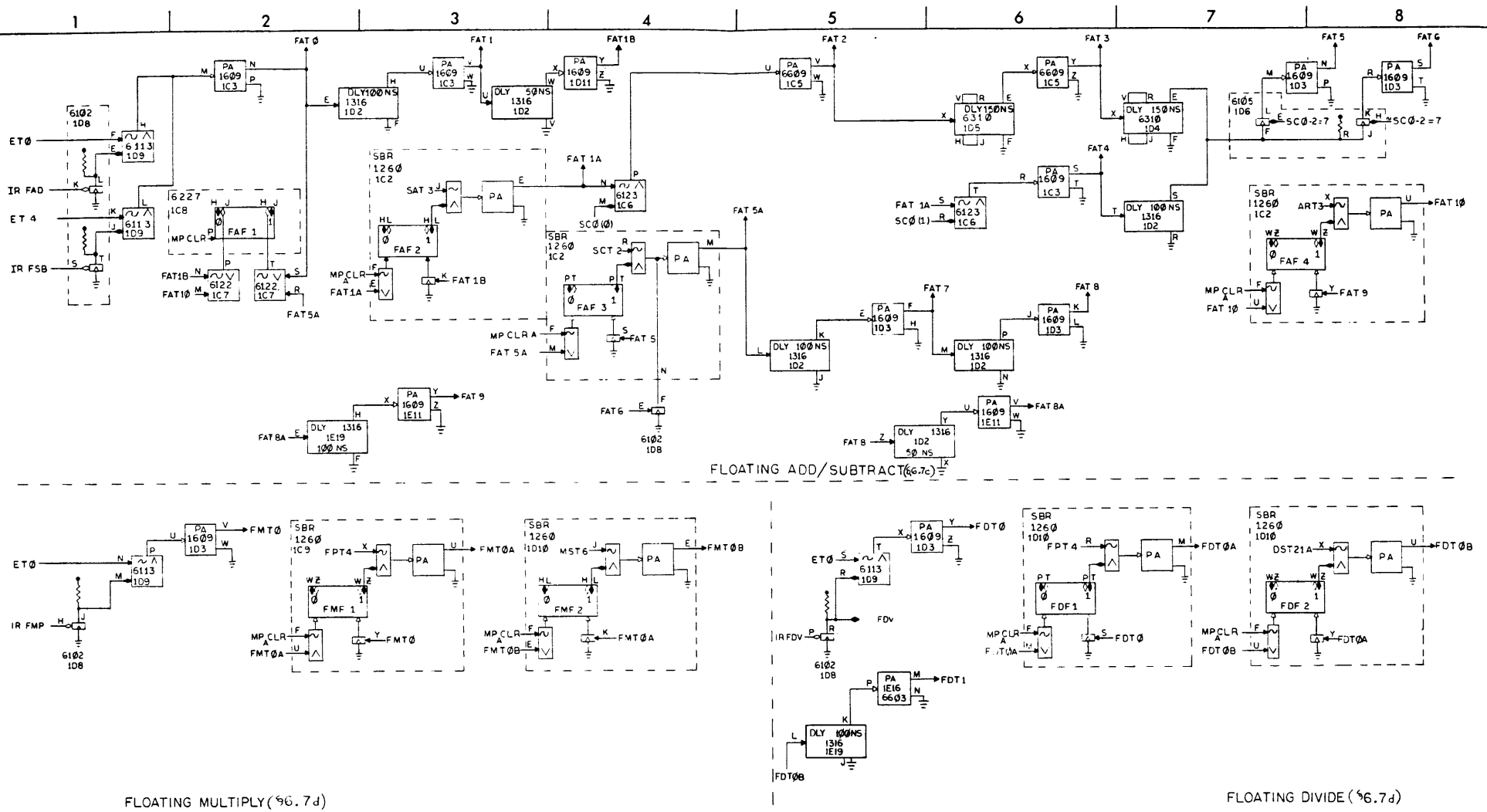
SHIFT OPERATIONS

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

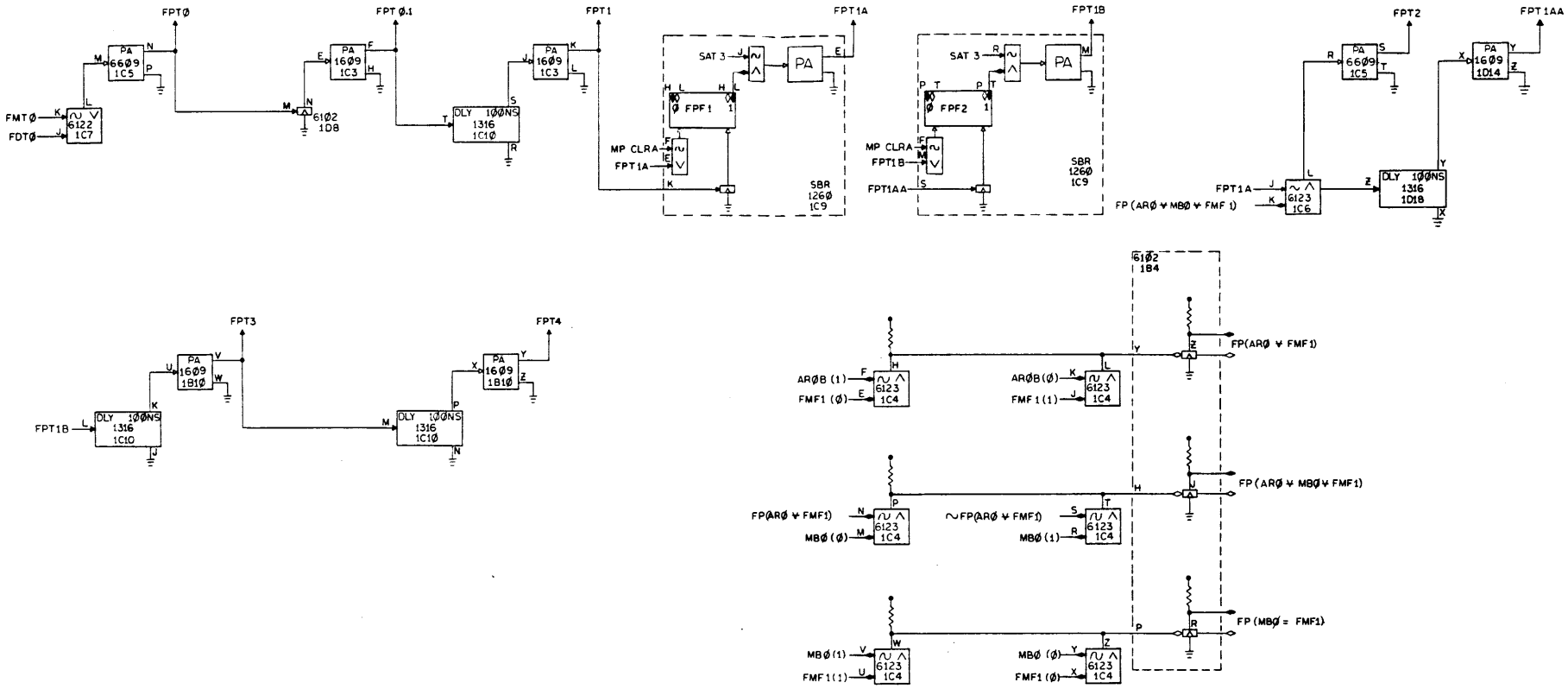


FIXED POINT MULTIPLY

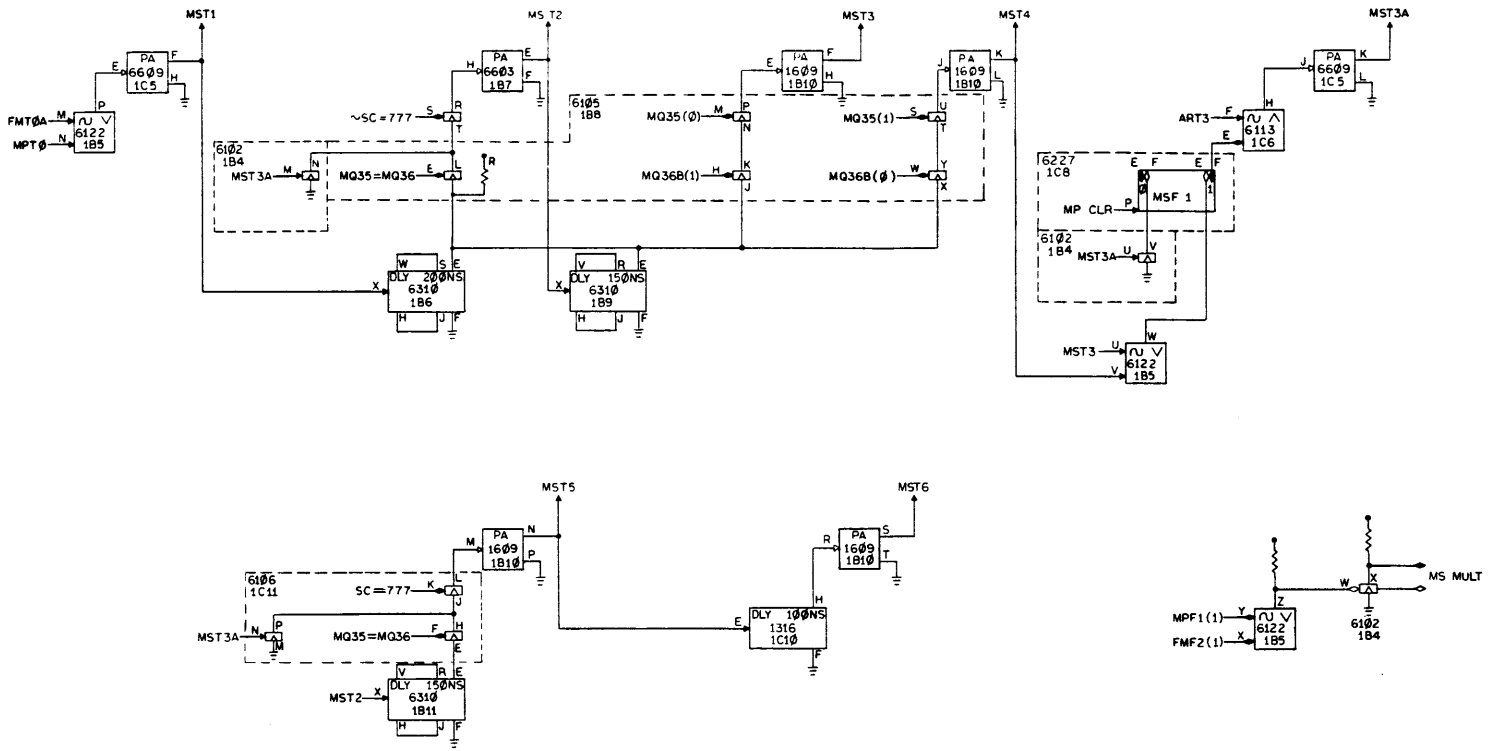
MP - Fixed Point Multiply
(BS-D-166-0-MP)



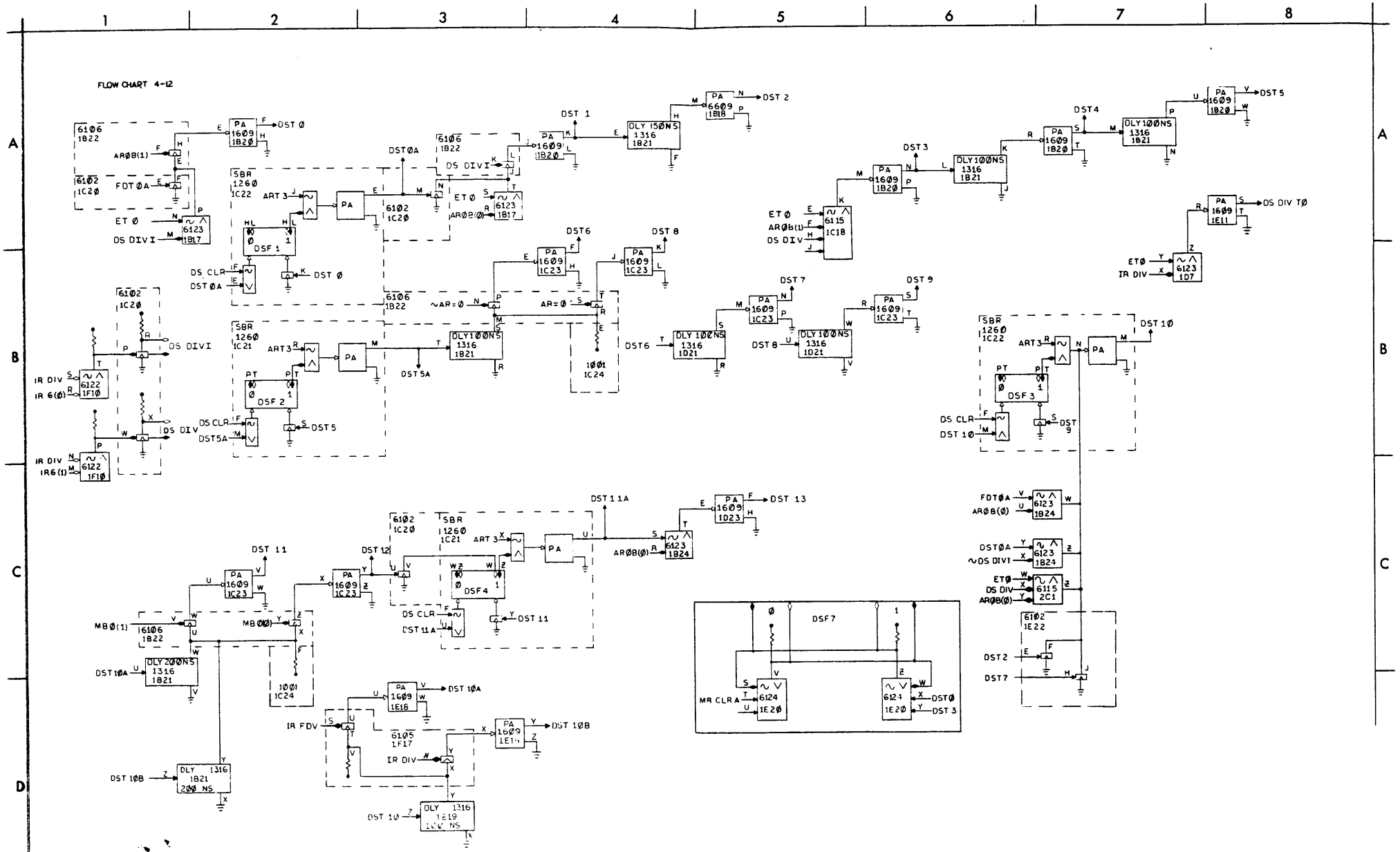
FA, FD, FM - Floating Point Instructions
(BS-D-166-0-FADM)

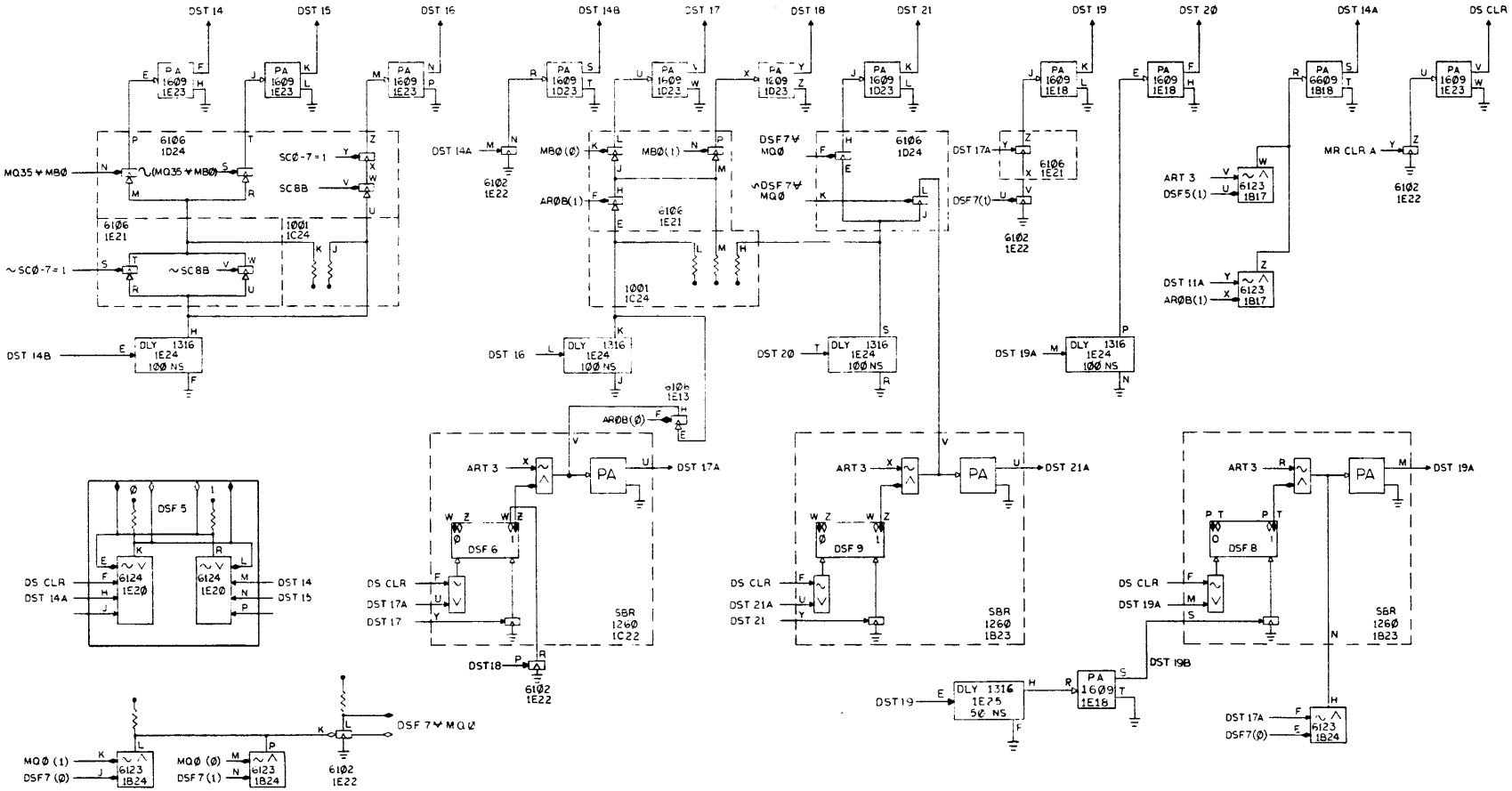


FP - Exponent Calculate Subroutine
(BS-D-166-0-FP)

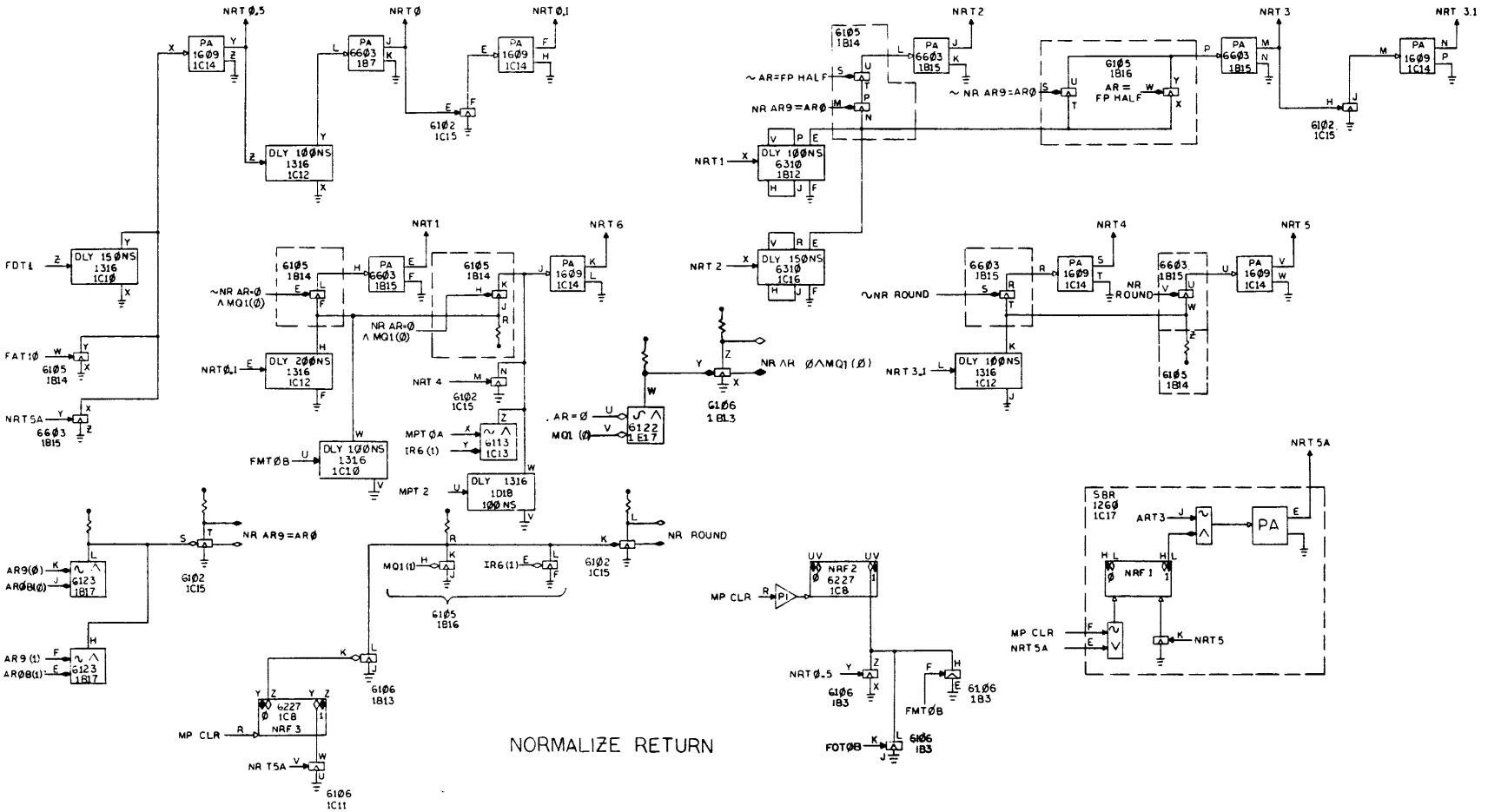


MS - Multiply Subroutine (BS-D-166-0-MS)

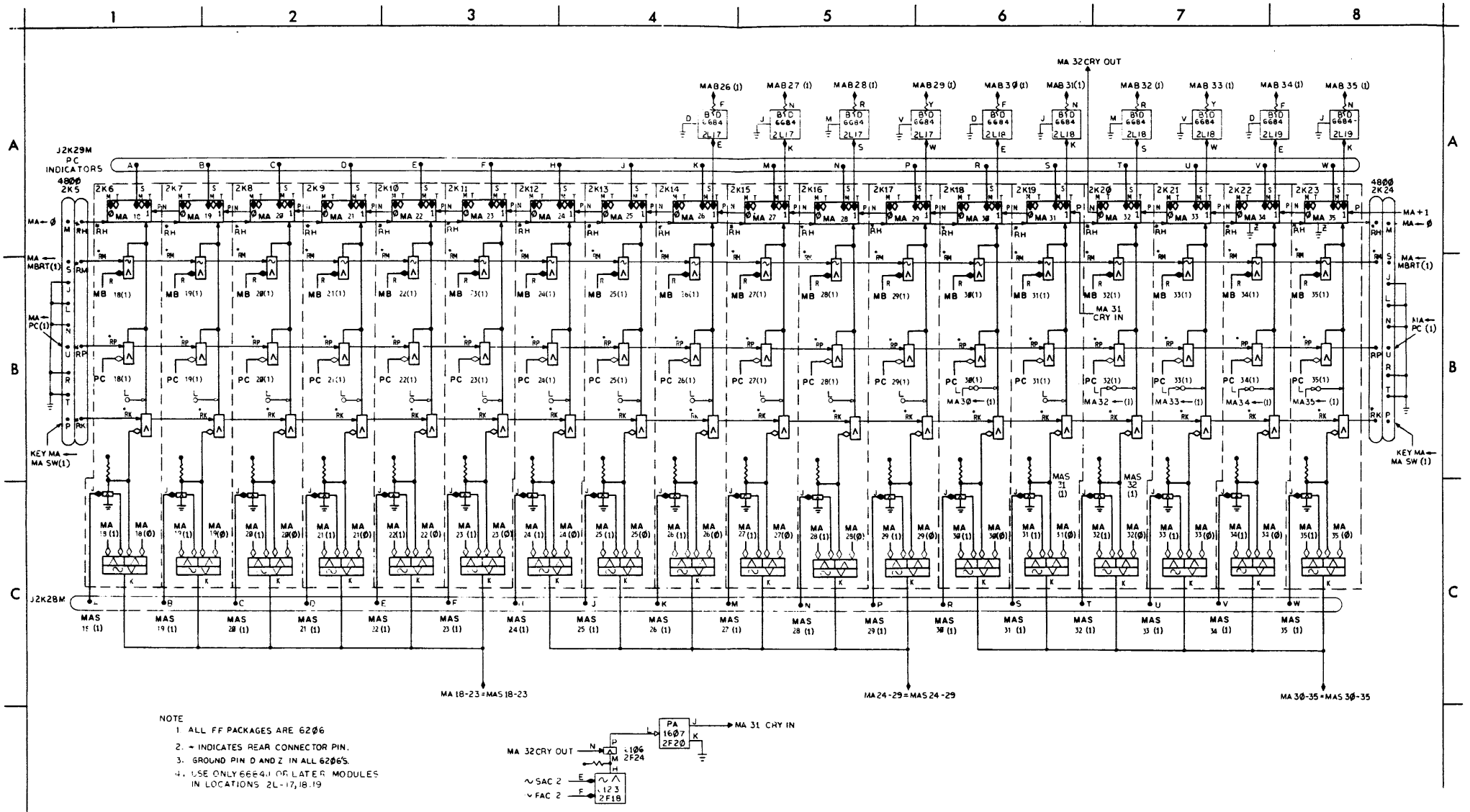

FLOW CHART 4-12
DS - Divide Subroutine 1 (BS-D-166-0-DS-1)



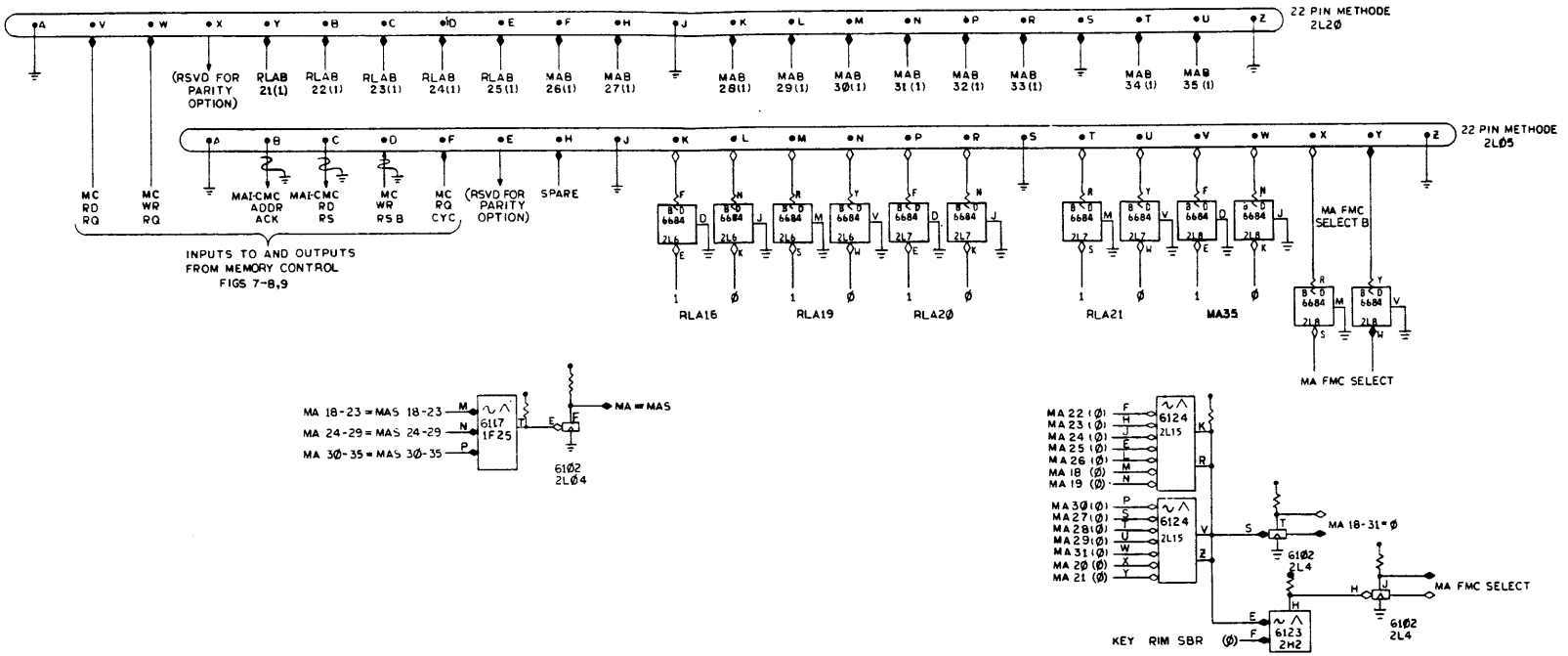
DS - Divide Subroutine 2 (BS-D-166-0-DS-2)



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

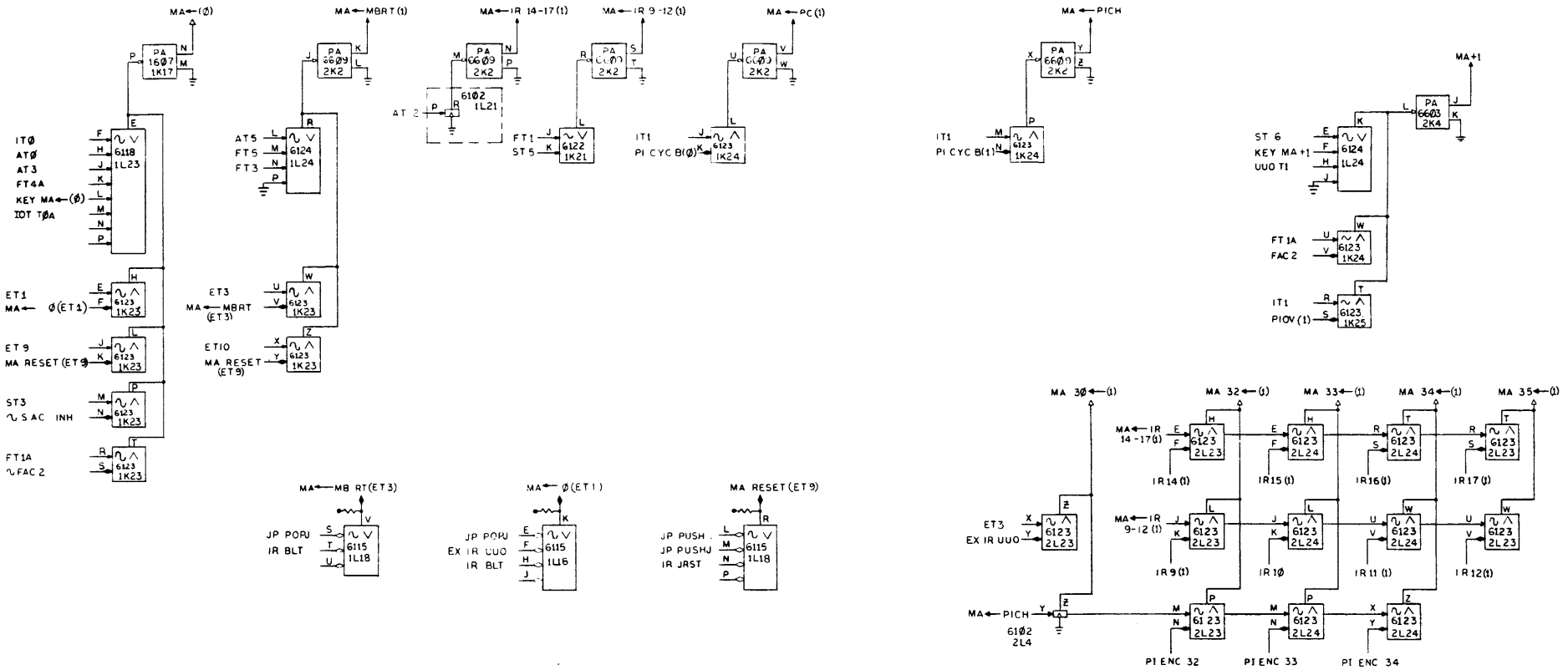


MA - Memory Address 18-35
(BS-D-166-0-MA-2)

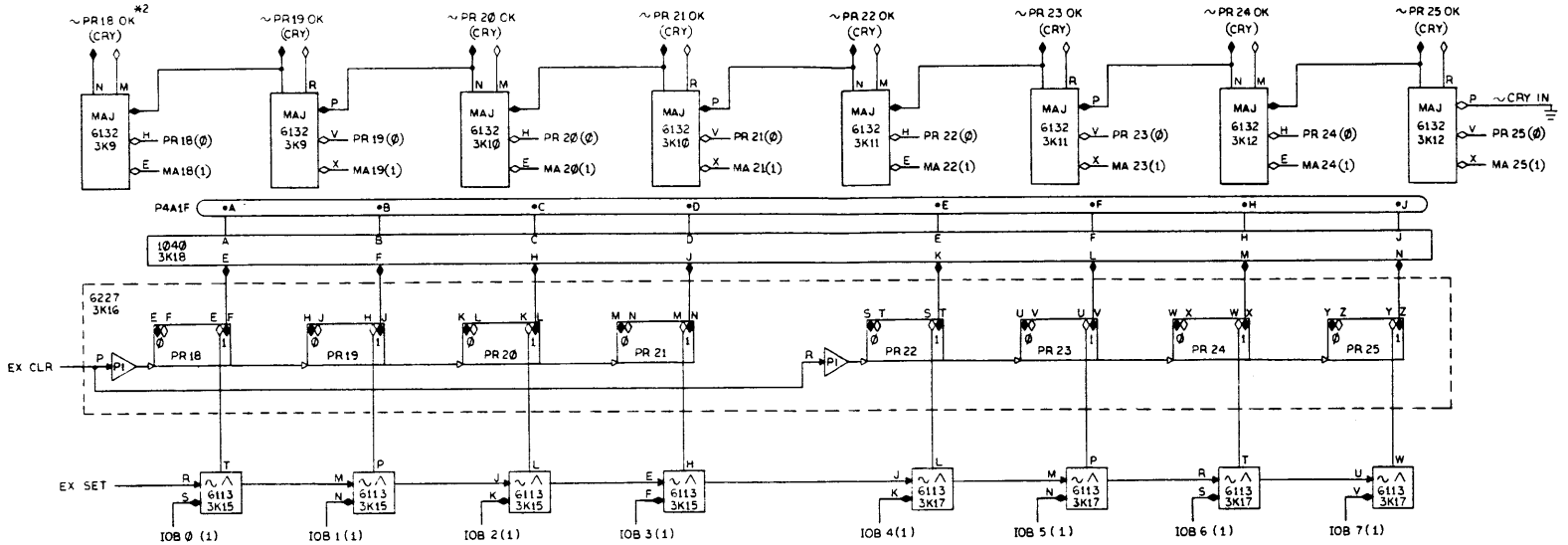


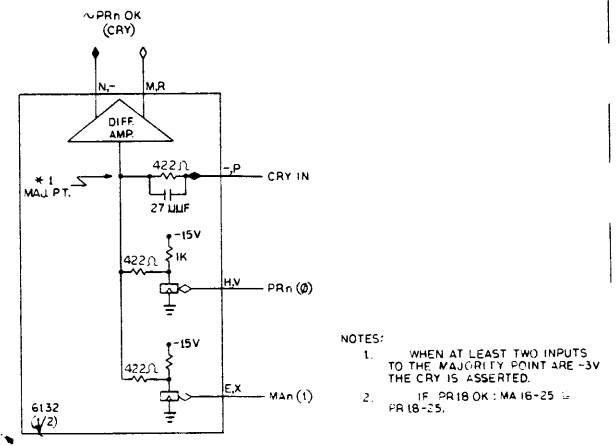
NOTE: 1 USE ONLY 6684J OR LATER MODULES IN LOCATIONS 2L-6,7,8

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

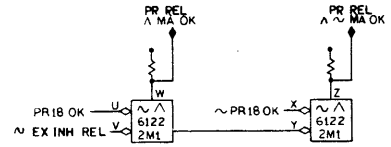


MA Control (BS-D-166-0-MA-1)

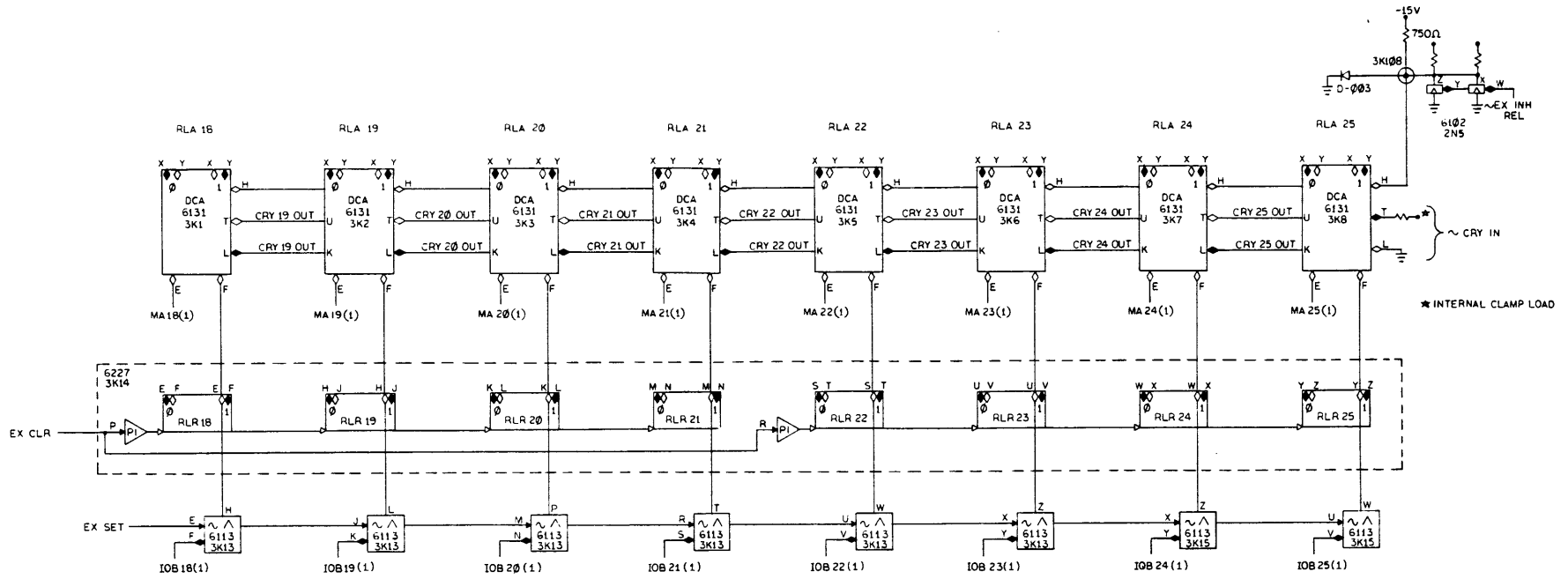


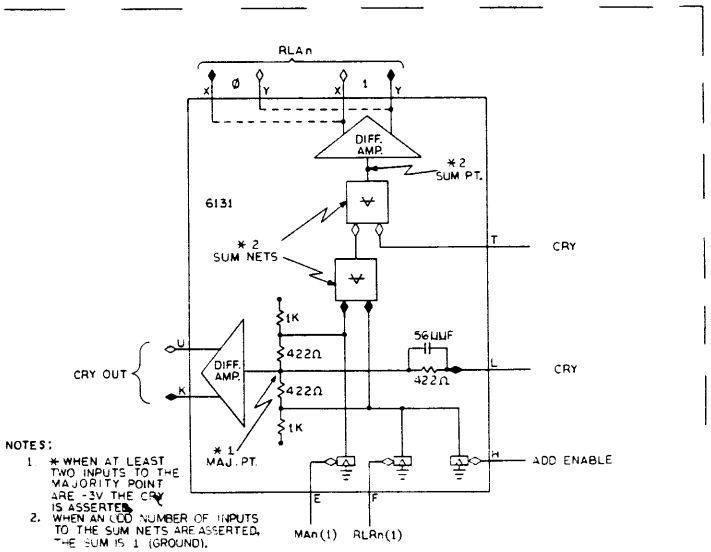


NOTES:
1. WHEN AT LEAST TWO INPUTS TO THE MAJORITY POINT ARE -3V THE CRY IS ASSERTED.
2. IF PR18 OK : MA 16-25 = PR 18-25.

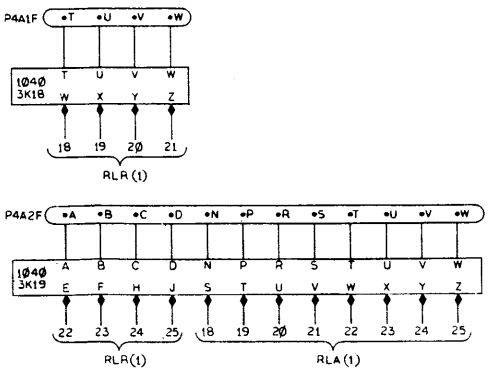


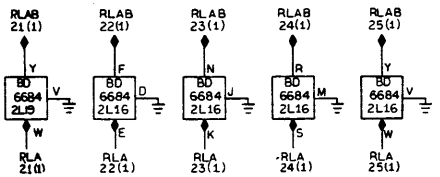
PR - Protection (BS-D-166-0-PR)



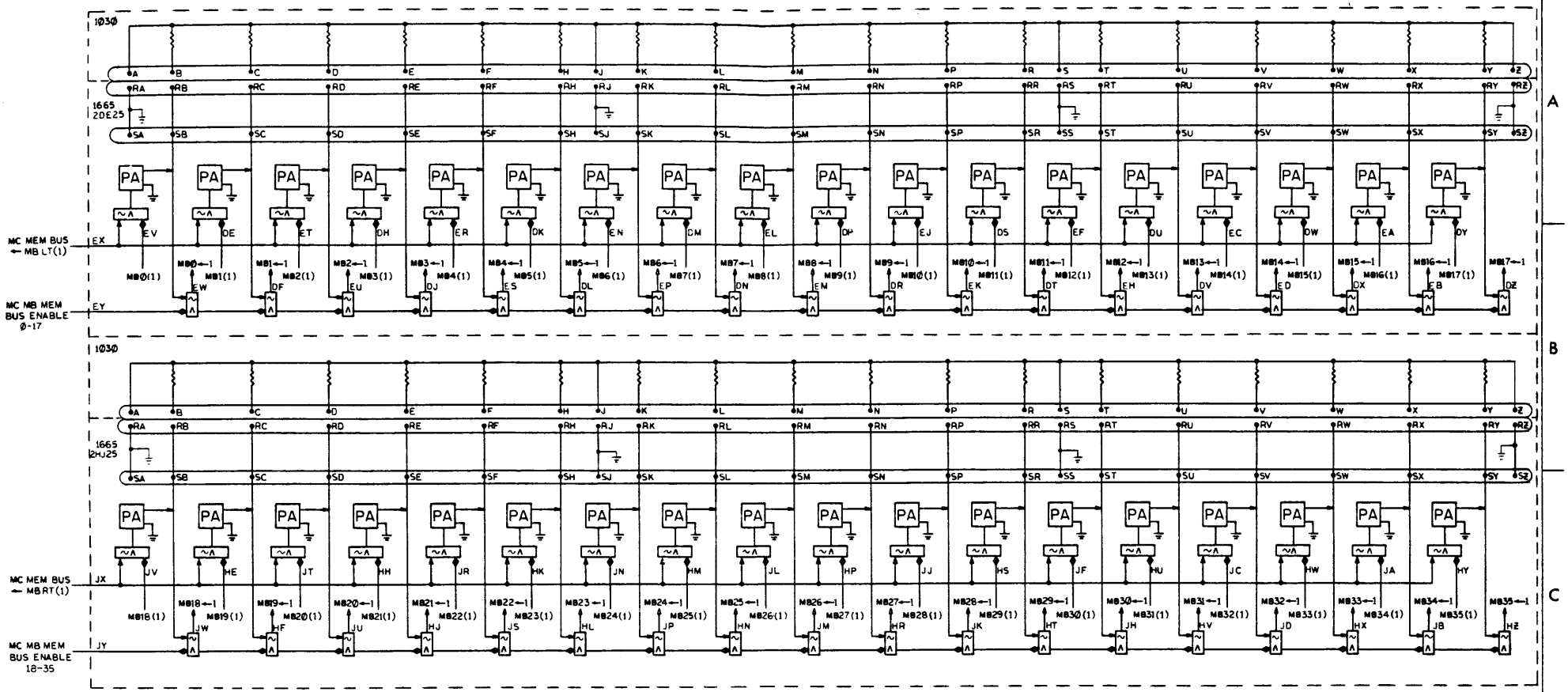


NOTES:
1. * WHEN AT LEAST TWO INPUTS TO THE MAJORITY POINT ARE -3V THE CRY IS ASSERTED.
2. WHEN AN EQD NUMBER OF INPUTS TO THE SUM NETS ARE ASSERTED, THE SUM IS 1 (GROUND).
3. USE ONLY 6684J OR LATER MODULES IN LOCATIONS 2L-16,19.





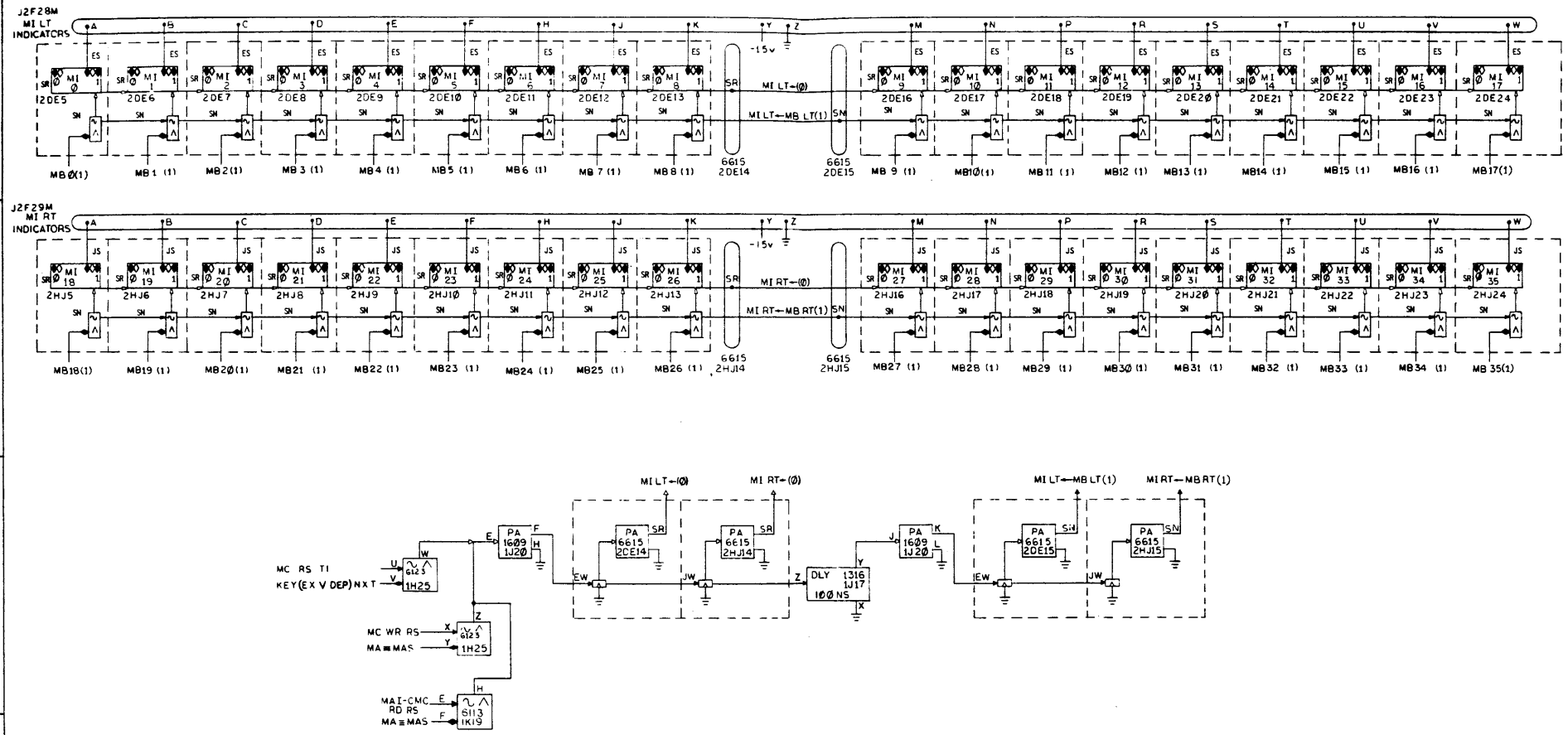
RLA, RLR - Relocation Adder and Register
(BS-D-166-0-RLAR)



NOTE:

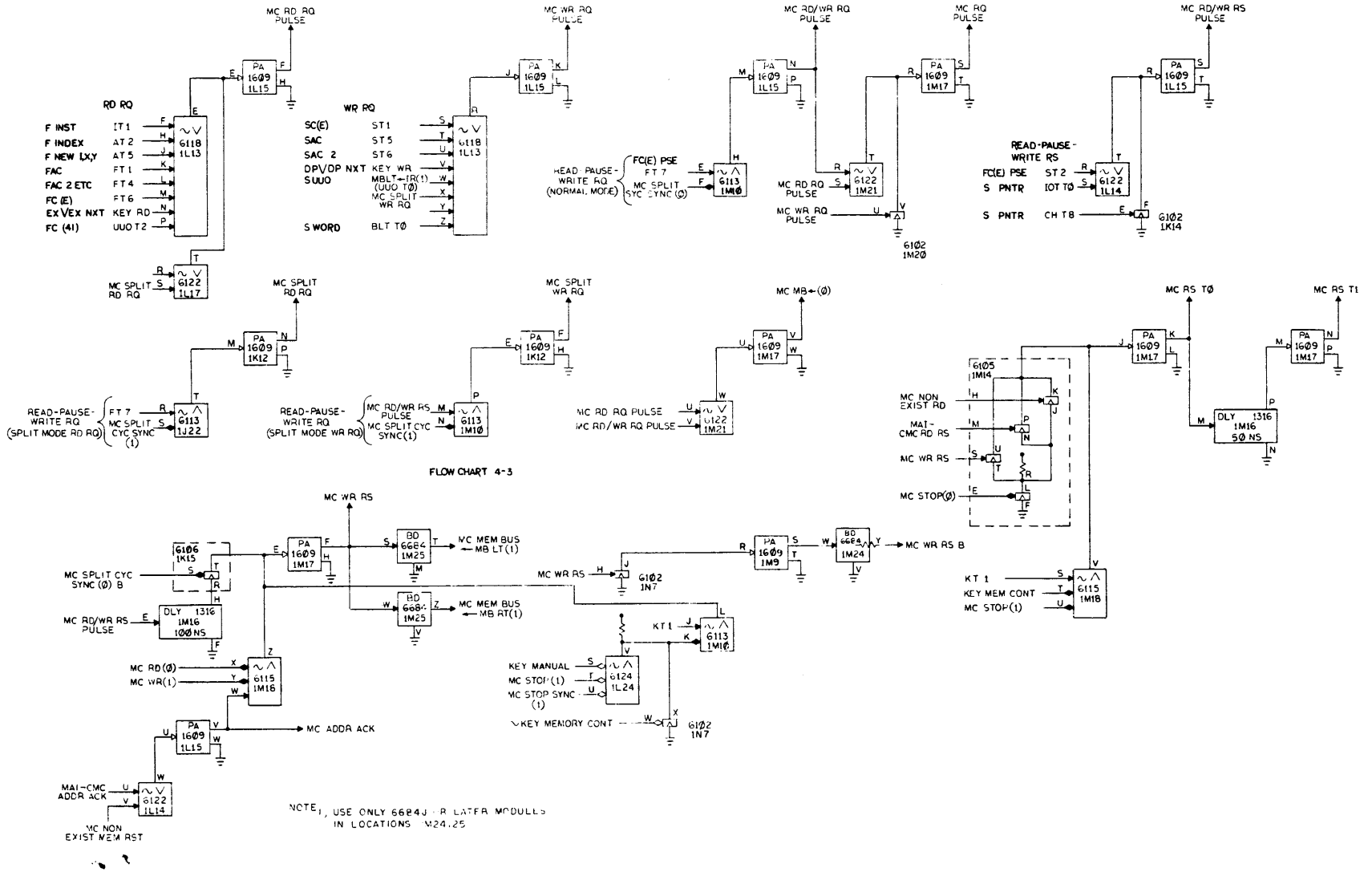
ALL TERMINATOR RES. IN THE 1030 MODULES ARE 100 OHMS.

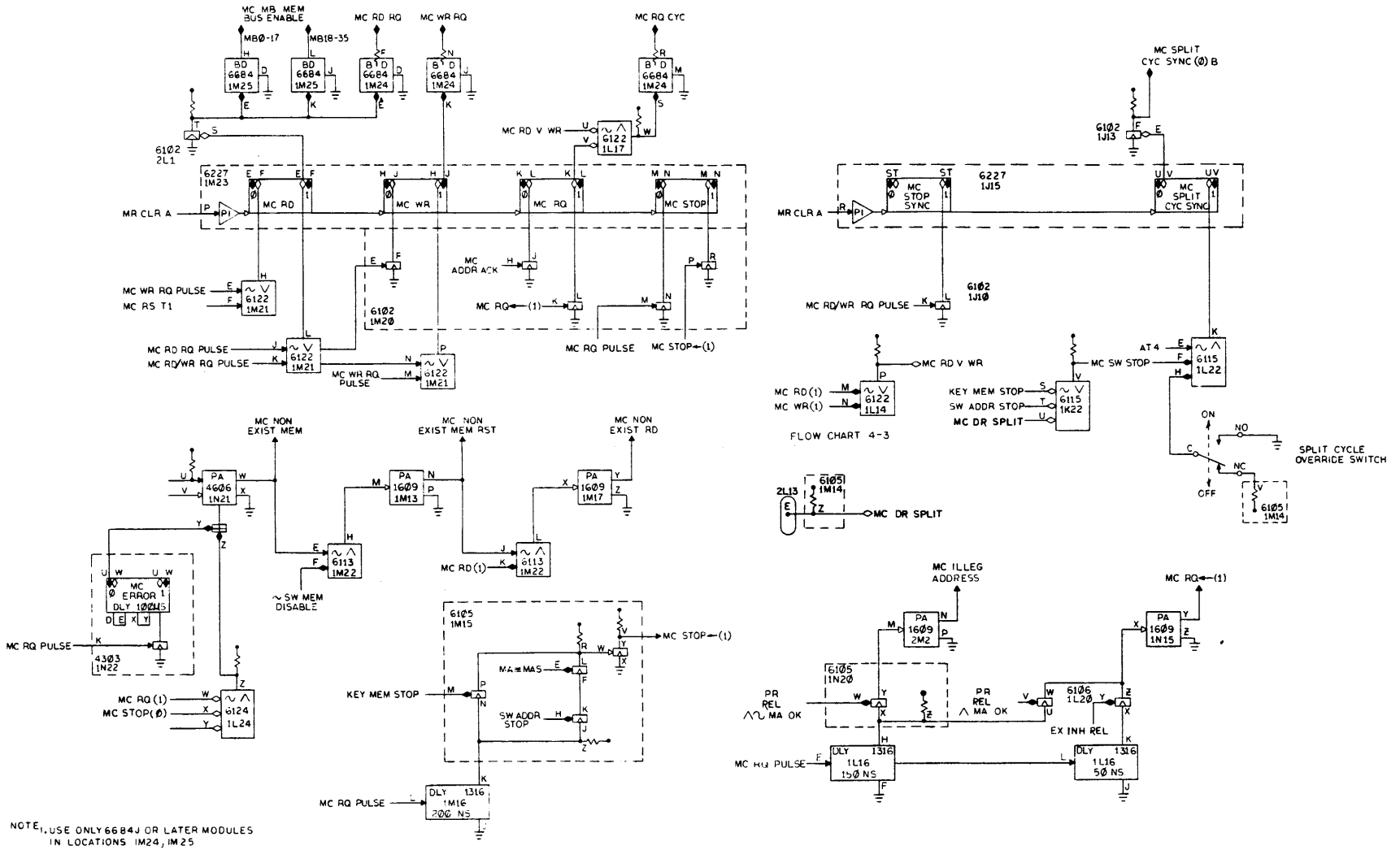
MB - Memory Buffer Interface
(BS-D-166-0-MB-4)



NOTE:

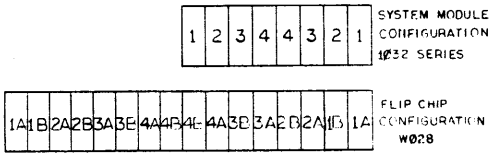
1. ALL FF PACKAGES ARE 62Ø5
2. SN AND SR ARE PINS IN LOWER REAR CONNECTOR


MC RD RQ PULSE
MC WR RQ PULSE
MC RD/WR RQ PULSE
MC RQ PULSE
MC RD/WR RS PULSE
RD RQ
F INST
F INDEX
F NEW LX,Y
FAC
FAC 2 ETC
FC (E)
EXVEX NXT
FC (41)
WR RQ
SC(E)
SAC
SAC 2
DPVDP NXT
SUUO
S WORD
MC SPLIT RD RQ
MC SPLIT WR RQ
MC MB←(Ø)
MC RS TØ
MC RS T1
READ-PAUSE-WRITE RQ (NORMAL MODE)
FC(E) PSE
MC SPLIT SYC SYNC (Ø)
MC RD RQ PULSE
MC WR RQ PULSE
READ-PAUSE-WRITE RS
S PNTR
READ-PAUSE-WRITE RQ (SPLIT MODE RD RQ)
MC SPLIT CYC SYNC (1)
READ-PAUSE-WRITE RQ (SPLIT MODE WR RQ)
MC RD/WR RS PULSE
MC SPLIT CYC SYNC(1)
MC RD/WR RQ PULSE
MC NON EXIST RD
MAI-CMC RD RS
MC WR RS
MC STOP(Ø)
DLY 1316 1M16 50 NS
FLOW CHART 4-3
MC SPLIT CYC SYNC (Ø) B
MC RD/WR RS PULSE
DLY 1316 1M16 100 NS
MC RD(Ø)
MC WR(1)
MC ADDR ACK
MAI-CMC ADDR ACK
MC NON EXIST MEM RST
MC MEM BUS ← MB LT(1)
MC MEM BUS ← MB RT(1)
MC WR RS B
KT 1
KEY MEM CONT
MC STOP(1)
KEY MANUAL
MC STOP SYNC (1)
NOTE1, USE ONLY 6684J-R LATER MODULES IN LOCATIONS 1M24,25


MC MB MEM BUS ENABLE
MB0-17
MB18-35
MC RD RQ
MC WR RQ
MC RQ CYC
MC RD V WR
MR CLR A
MC RD
MC WR
MC RQ
MC STOP
MC WR RQ PULSE
MC RS T1
MC ADDR ACK
MC RD RQ PULSE
MC RD/WR RQ PULSE
MC WR RQ PULSE
MC RQ PULSE
MC STOP←(1)
MC SPLIT CYC SYNC (Ø) B
MC STOP SYNC
MC SPLIT CYC SYNC
MC RD/WR RQ PULSE
MC RD V WR
MC RD(1)
MC WR(1)
MC SW STOP
KEY MEM STOP
SW ADDR STOP
MC DR SPLIT
FLOW CHART 4-3
SPLIT CYCLE OVERRIDE SWITCH
ON
NO
NC
OFF
MC NON EXIST MEM
MC NON EXIST MEM RST
MC NON EXIST RD
SW MEM DISABLE
MC ERROR
MC RQ PULSE
MC RQ (1)
MC STOP(Ø)
MA≠MAS
SW ADDR STOP
MC ILLEG ADDRESS
PR REL MA OK
EX INH REL
MC RQ←(1)
NOTE 1. USE ONLY 6684J OR LATER MODULES IN LOCATIONS 1M24, 1M25

IO BUS(§§8.4)

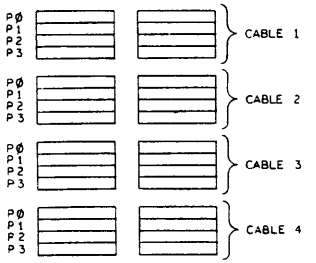
FLIP CHIP PIN	SYSTEM MODULE PIN	IO CABLE 1	IO CABLE 2	IO CABLE 3	IO CABLE 4
C	A	GND	GND	GND	GND
D	B	IOB Ø (1) ⊸	IOB 18 (1) ⊸	IOB RESET ⟶	DATAO CLEAR ⟶
E	C	IOB 1 (1) ⊸	IOB 19 (1) ⊸	POWER ON -15V	DATAO SET ⟶
H	D	IOB 2 (1) ⊸	IOB 20 (1) ⊸		CONO CLEAR ⟶
K	E	IOB 3 (1) ⊸	IOB 21 (1) ⊸	MC DR SPLIT ⊸	CONO SET ⟶
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 10 (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 30 (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)

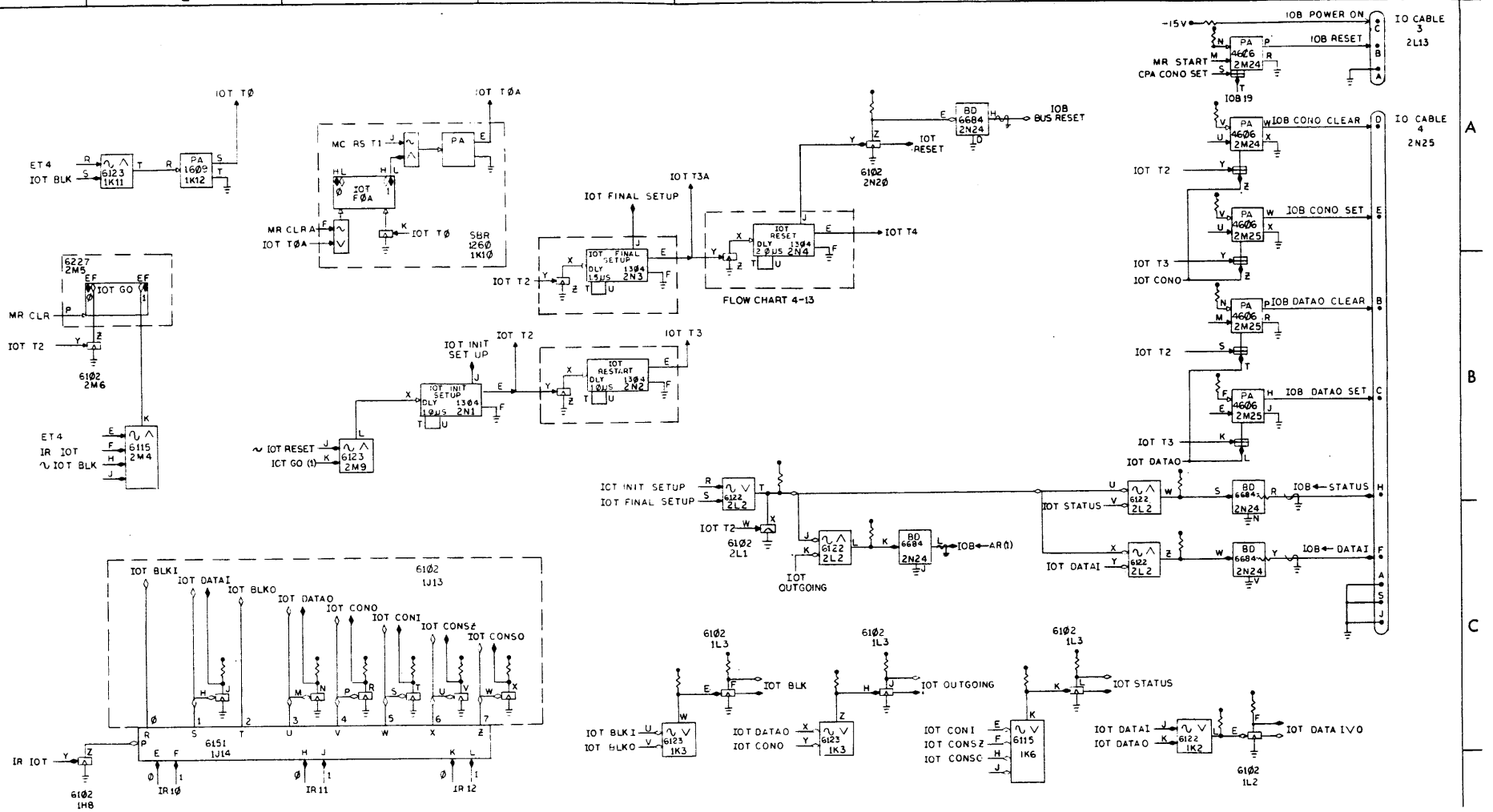


NOTE: 1 SYSTEM MODULE CONNECTOR (1Ø32) = 2 FLIP CHIP CONNECTORS (WØ28)

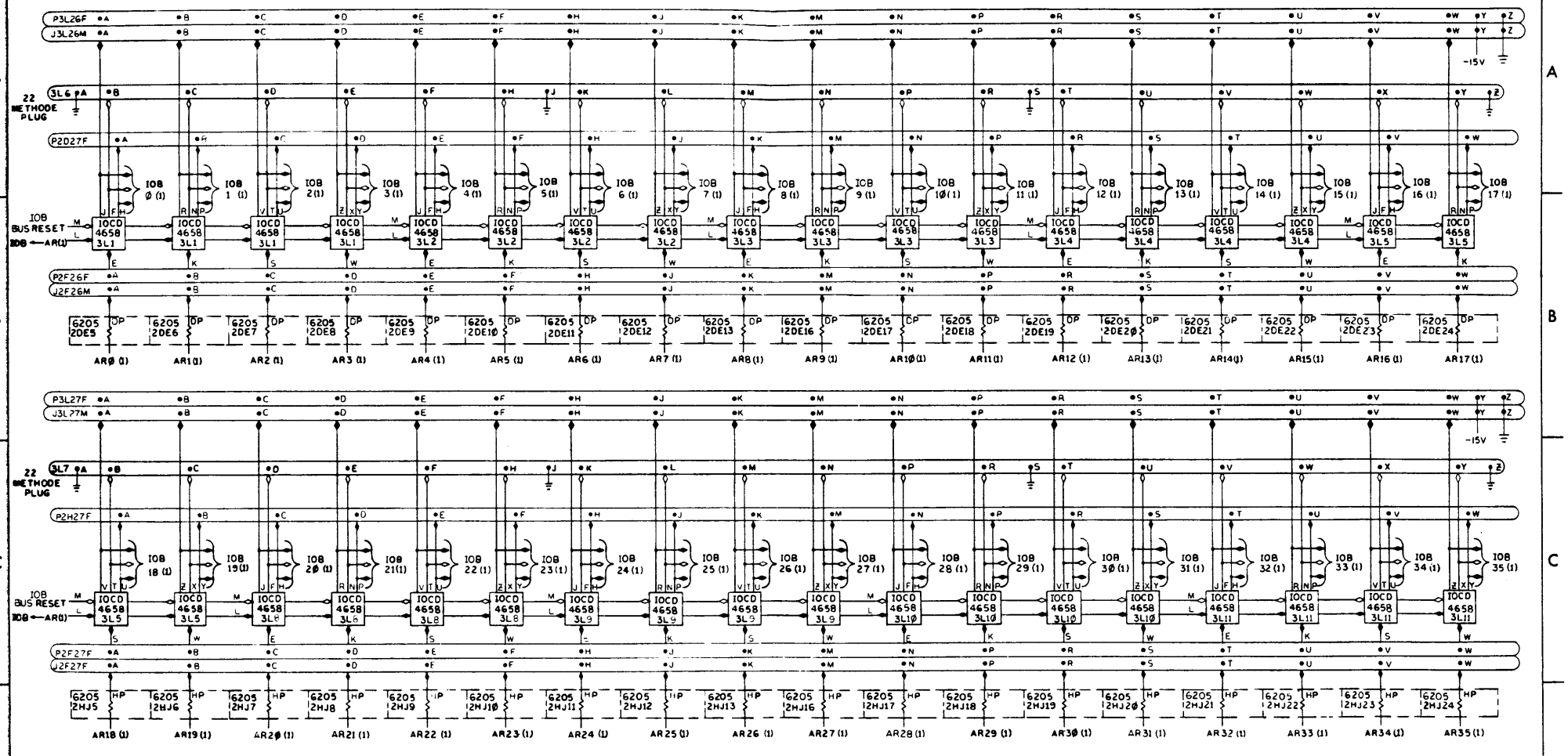
MEM BUS(§§7.1a,7.2)

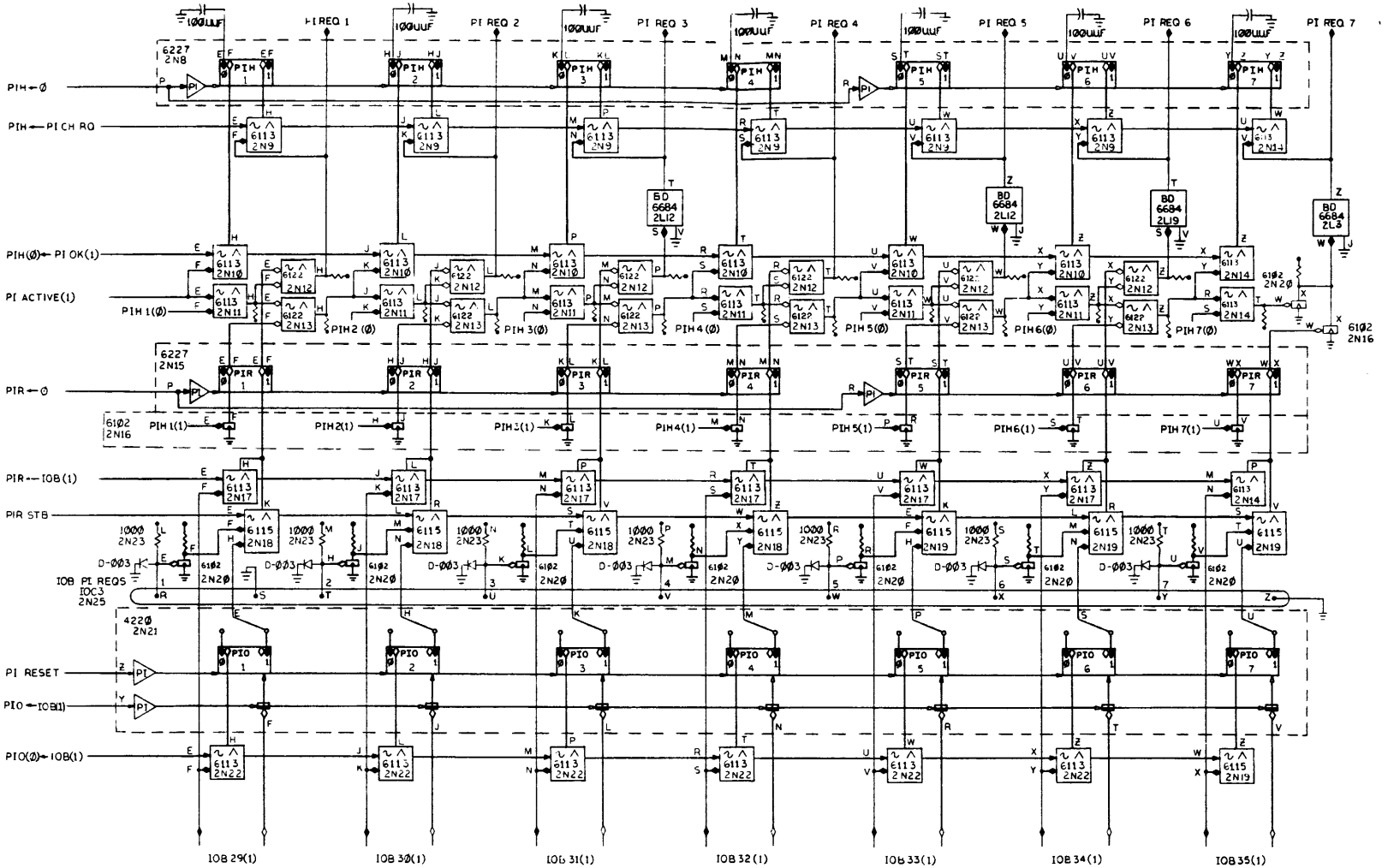
FLIP CHIP PIN	SYSTEM MODULE PIN	MEM CABLE 1	MEM CABLE 2	MEM CABLE 3	MEM CABLE 4
C	A	GND	GND	GND	GND
D	B	ADDR ACK ⟶	MA 22 (1) ⟶	MB Ø (1) ⟶	MB 18 (1) ⟶
E	C	RD RS ⟶	MA 23 (1) ⟶	MB 1 (1) ⟶	MB 19 (1) ⟶
H	D	WR RS ⟶	MA 24 (1) ⟶	MB 2 (1) ⟶	MB 20 (1) ⟶
K	E	PAR (1) ⟶	MA 25 (1) ⟶	MB 3 (1) ⟶	MB 21 (1) ⟶
M	F	RQ CYCLE ⟶	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 30 (1) ⟶	MB 8 (1) ⟶	MB 26 (1) ⟶
D	N	MA 19 (Ø) ⊸	MA 31 (1) ⟶	MB 9 (1) ⟶	MB 27 (1) ⟶
E	P	MA 20 (1) ⊸	MA 32 (1) ⟶	MB 10 (1) ⟶	MB 28 (1) ⟶
H	R	MA 20 (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	MAFMC ⊸	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)



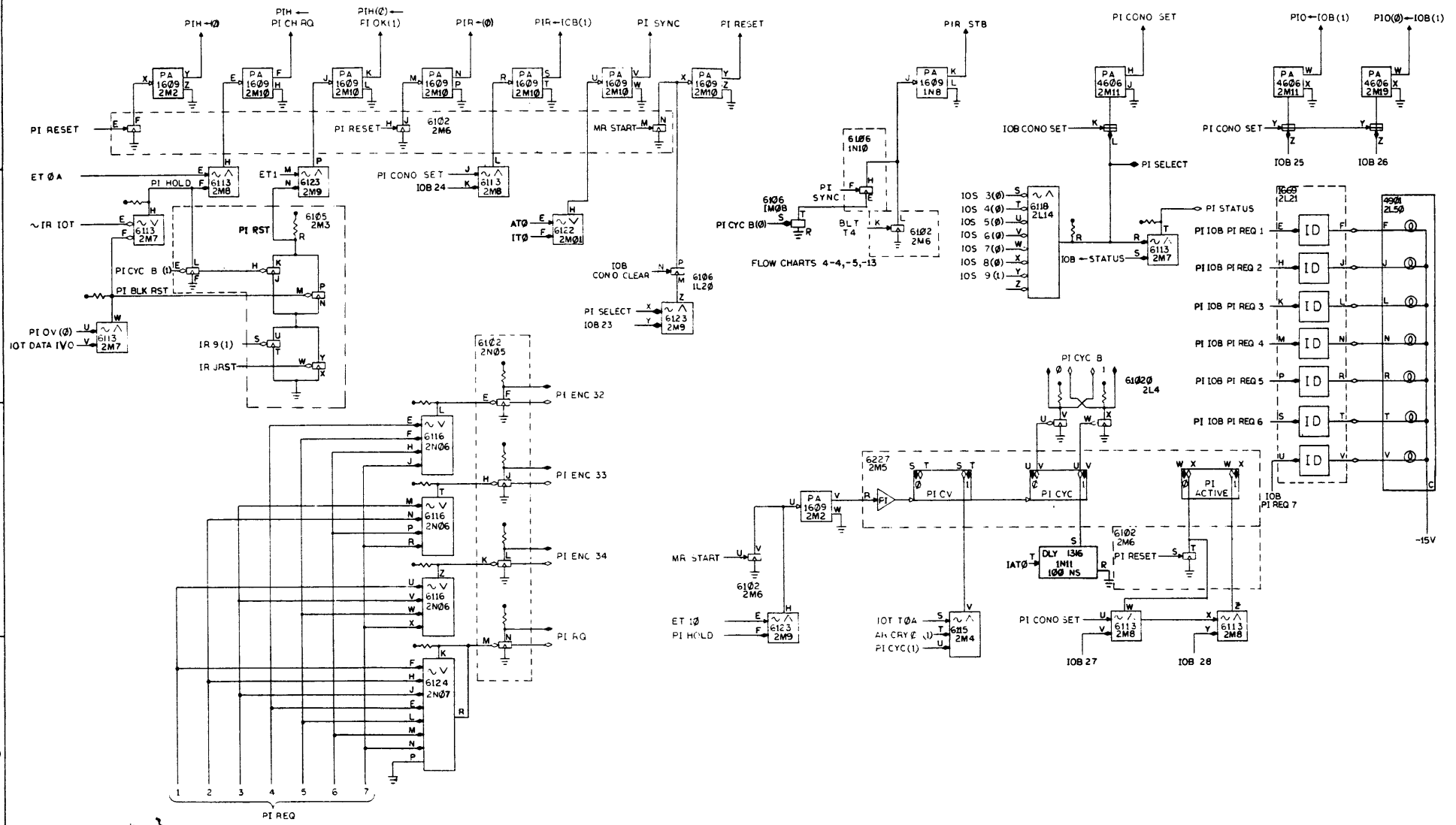


IOT, IOB Control (BS-D-166-0-IOTB)

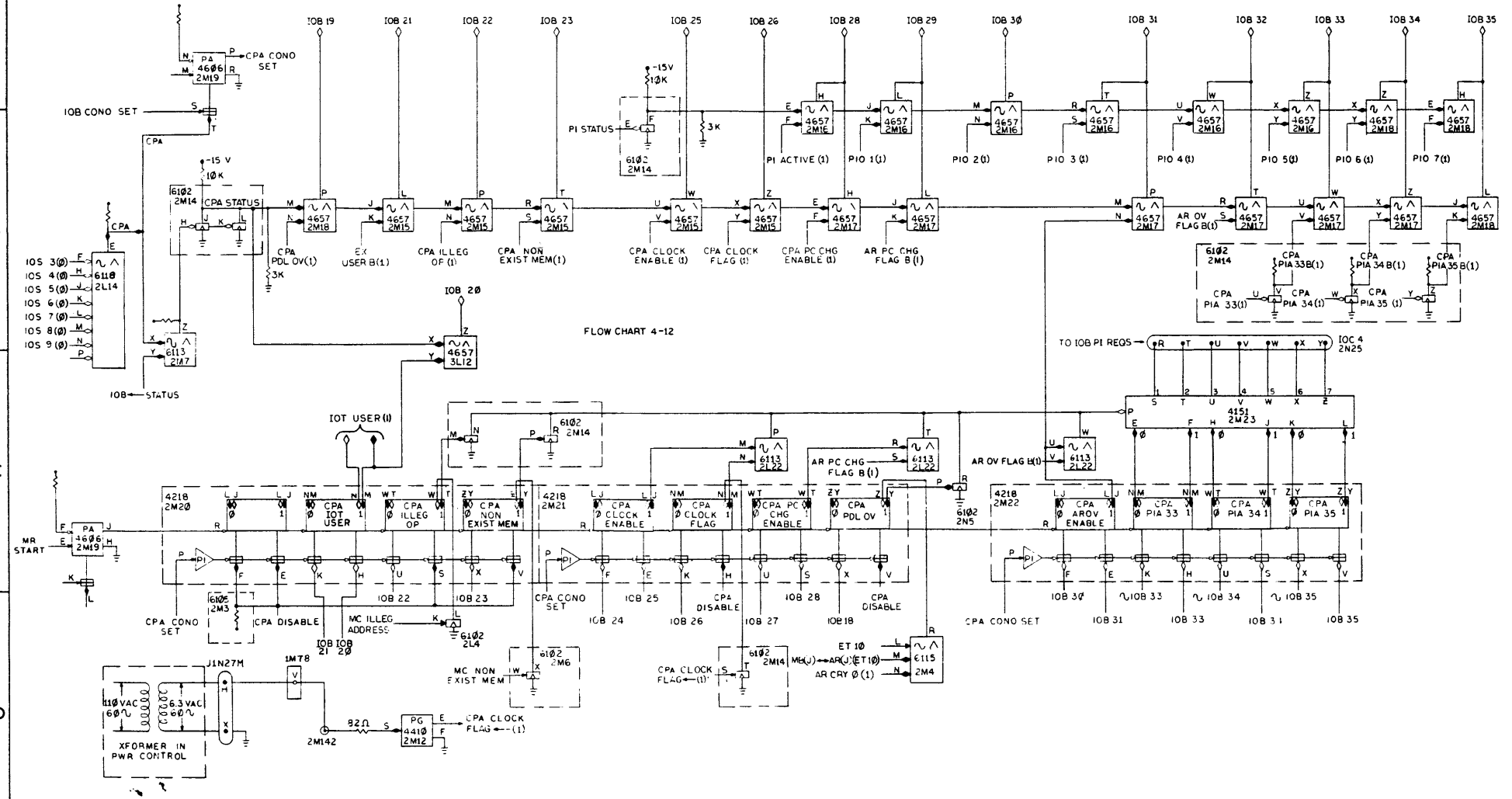

P3L26F
J3L26M
-15V
22 METHODE PLUG
3L6
P2D27F
IOB BUS RESET
IOB ← AR(1)
IOCD 4658 3L1
IOCD 4658 3L2
IOCD 4658 3L3
IOCD 4658 3L4
IOCD 4658 3L5
P2F26F
J2F26M
6205 2DE5
6205 2DE6
6205 2DE7
6205 2DE8
6205 2DE9
6205 2DE10
6205 2DE11
6205 2DE12
6205 2DE13
6205 2DE16
6205 2DE17
6205 2DE18
6205 2DE19
6205 2DE20
6205 2DE21
6205 2DE22
6205 2DE23
6205 2DE24
AR0 (1)
AR1 (1)
AR2 (1)
AR3 (1)
AR4 (1)
AR5 (1)
AR6 (1)
AR7 (1)
AR8 (1)
AR9 (1)
AR10 (1)
AR11 (1)
AR12 (1)
AR13 (1)
AR14 (1)
AR15 (1)
AR16 (1)
AR17 (1)
P3L27F
J3L27M
-15V
22 METHODE PLUG
3L7
P2H27F
IOB BUS RESET
IOB ← AR(1)
IOCD 4658 3L5
IOCD 4658 3L6
IOCD 4658 3L8
IOCD 4658 3L9
IOCD 4658 3L10
IOCD 4658 3L11
P2F27F
J2F27F
6205 2HJ5
6205 2HJ6
6205 2HJ7
6205 2HJ8
6205 2HJ9
6205 2HJ10
6205 2HJ11
6205 2HJ12
6205 2HJ13
6205 2HJ16
6205 2HJ17
6205 2HJ18
6205 2HJ19
6205 2HJ20
6205 2HJ21
6205 2HJ22
6205 2HJ23
6205 2HJ24
AR18 (1)
AR19 (1)
AR20 (1)
AR21 (1)
AR22 (1)
AR23 (1)
AR24 (1)
AR25 (1)
AR26 (1)
AR27 (1)
AR28 (1)
AR29 (1)
AR30 (1)
AR31 (1)
AR32 (1)
AR33 (1)
AR34 (1)
AR35 (1)



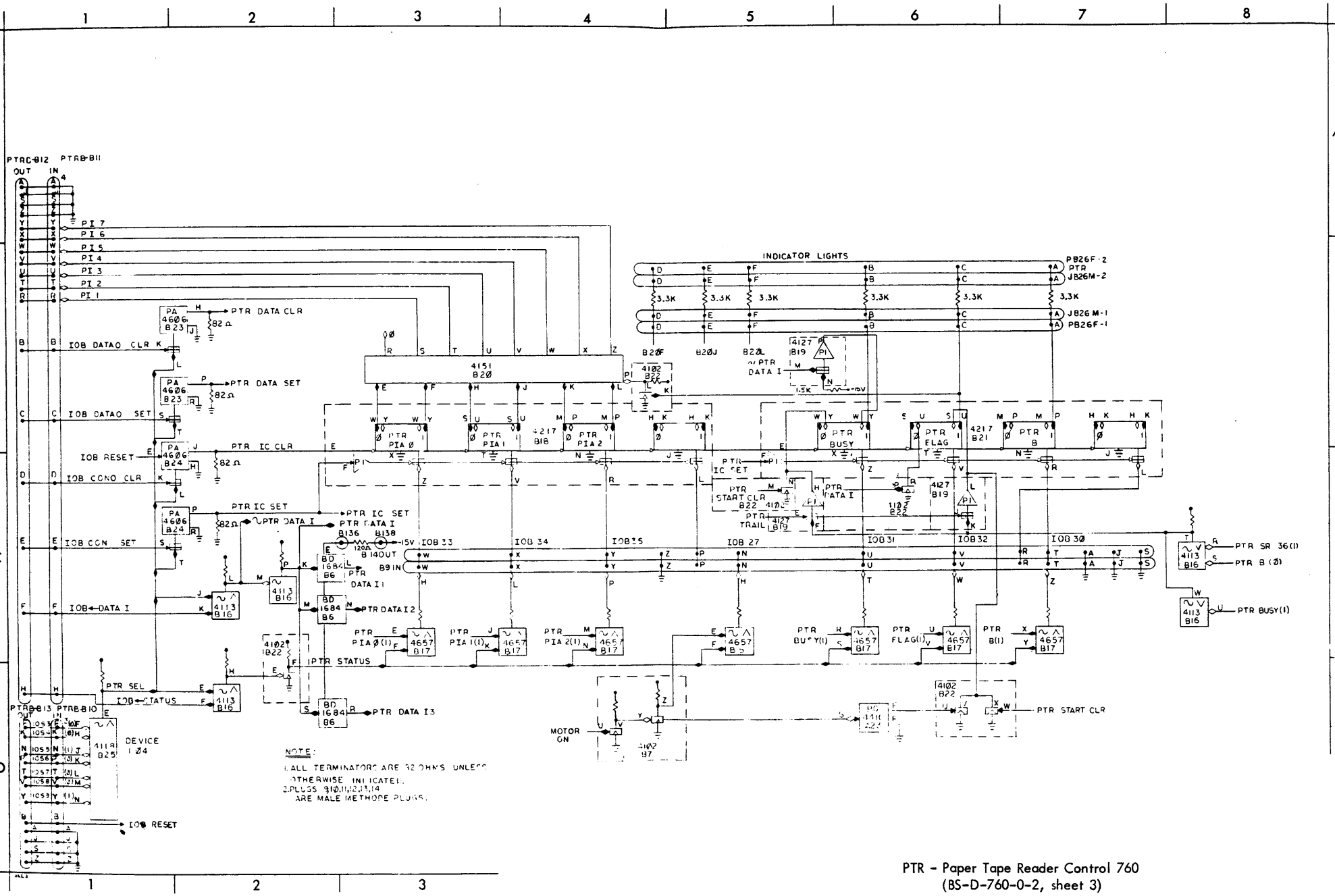
PI - PIH, R, O, 1-7 (BS-D-166-0-PI-2)



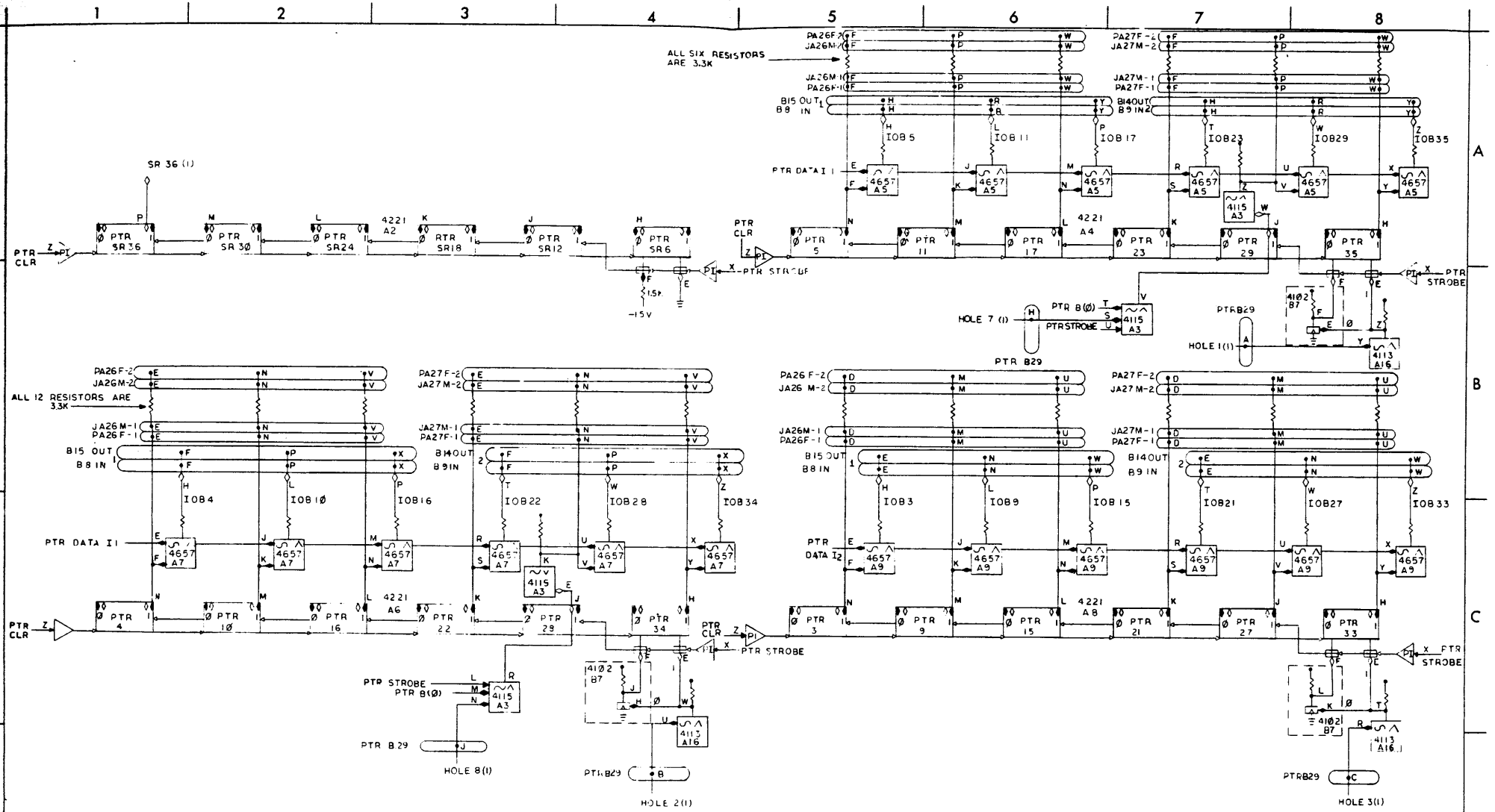
PI Control (BS-D-166-0-PI-1)



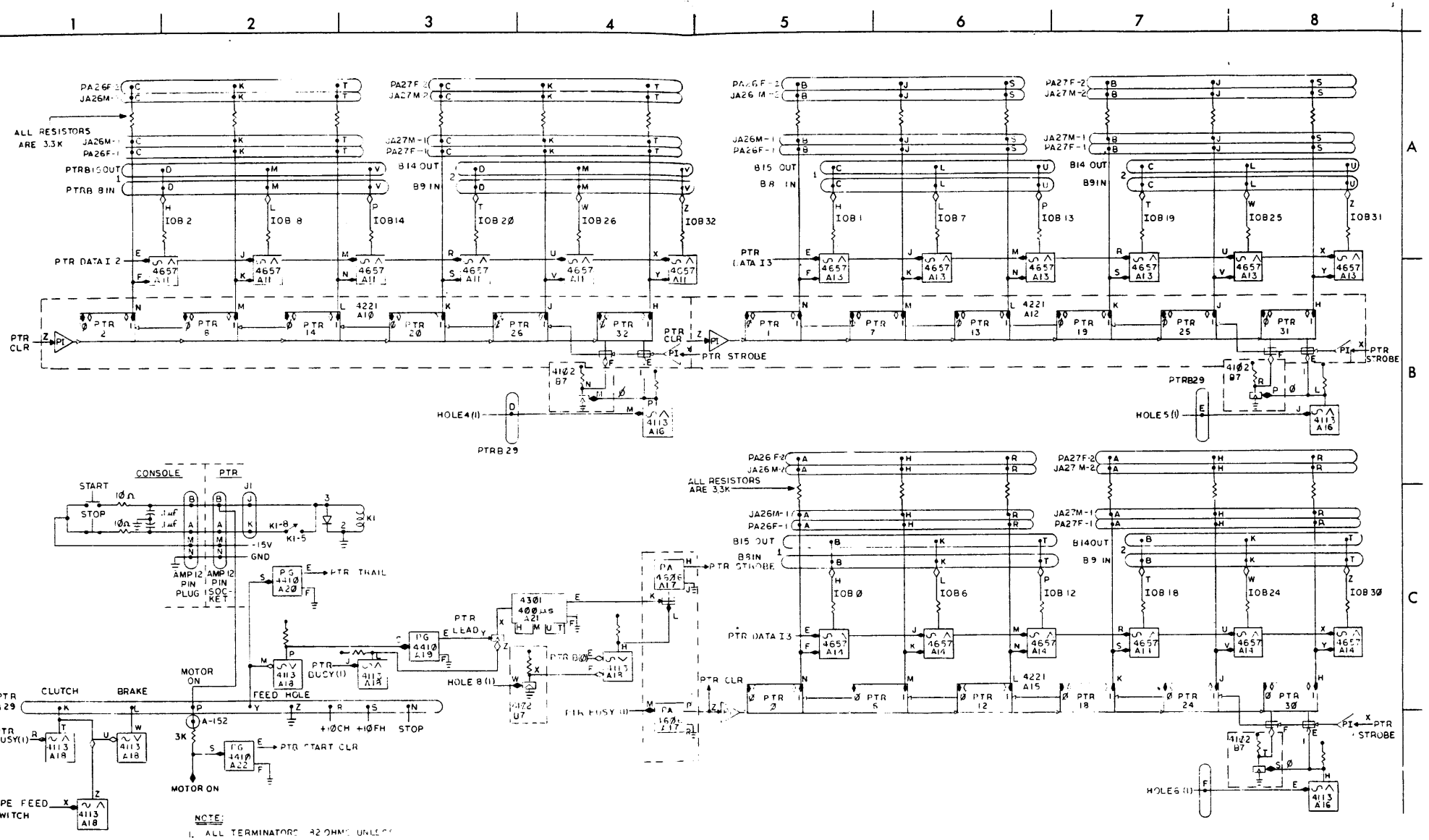
CPA - Arithmetic IO Interface
(BS-D-166-0-CPA)



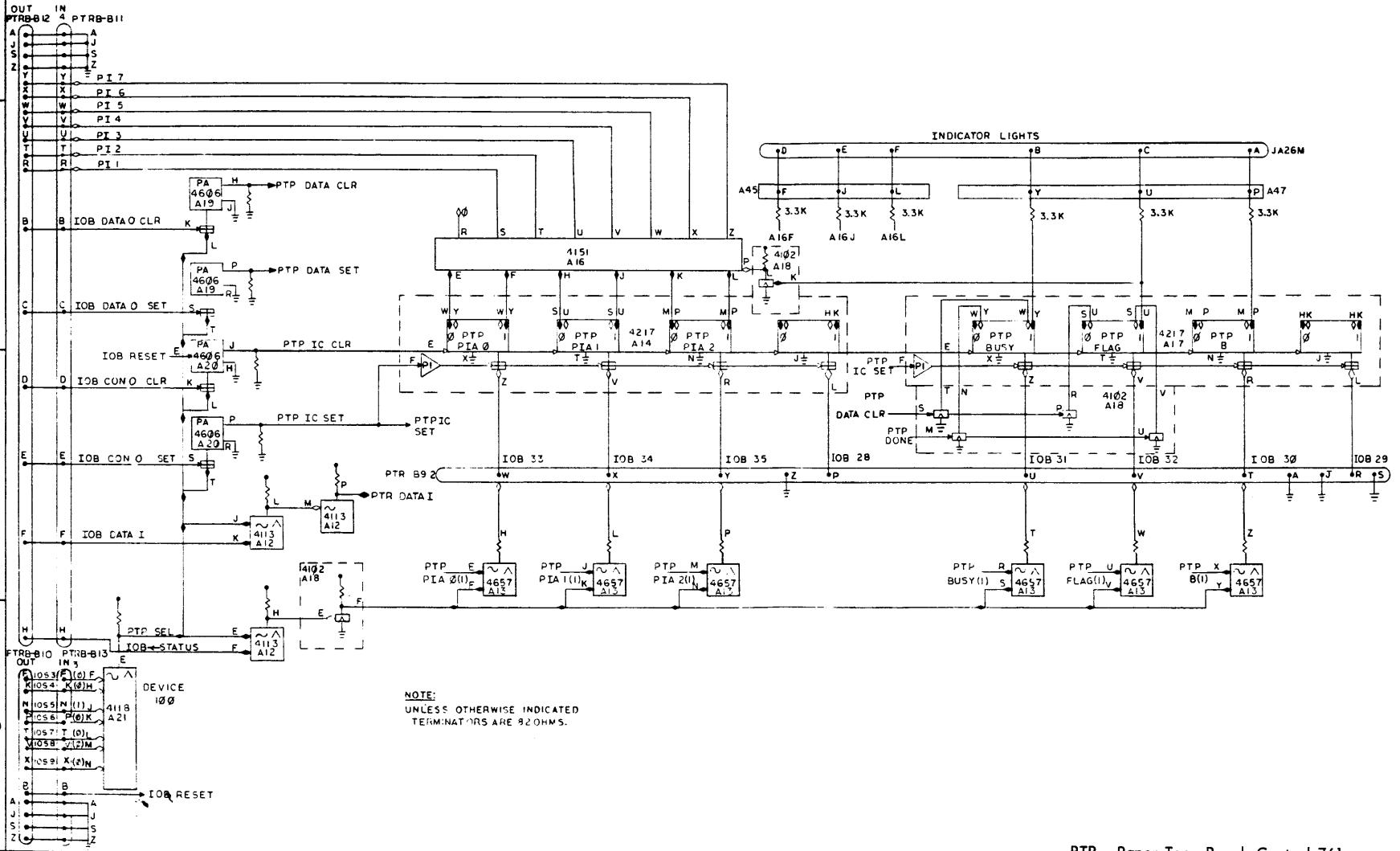
PTR – Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 3)



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

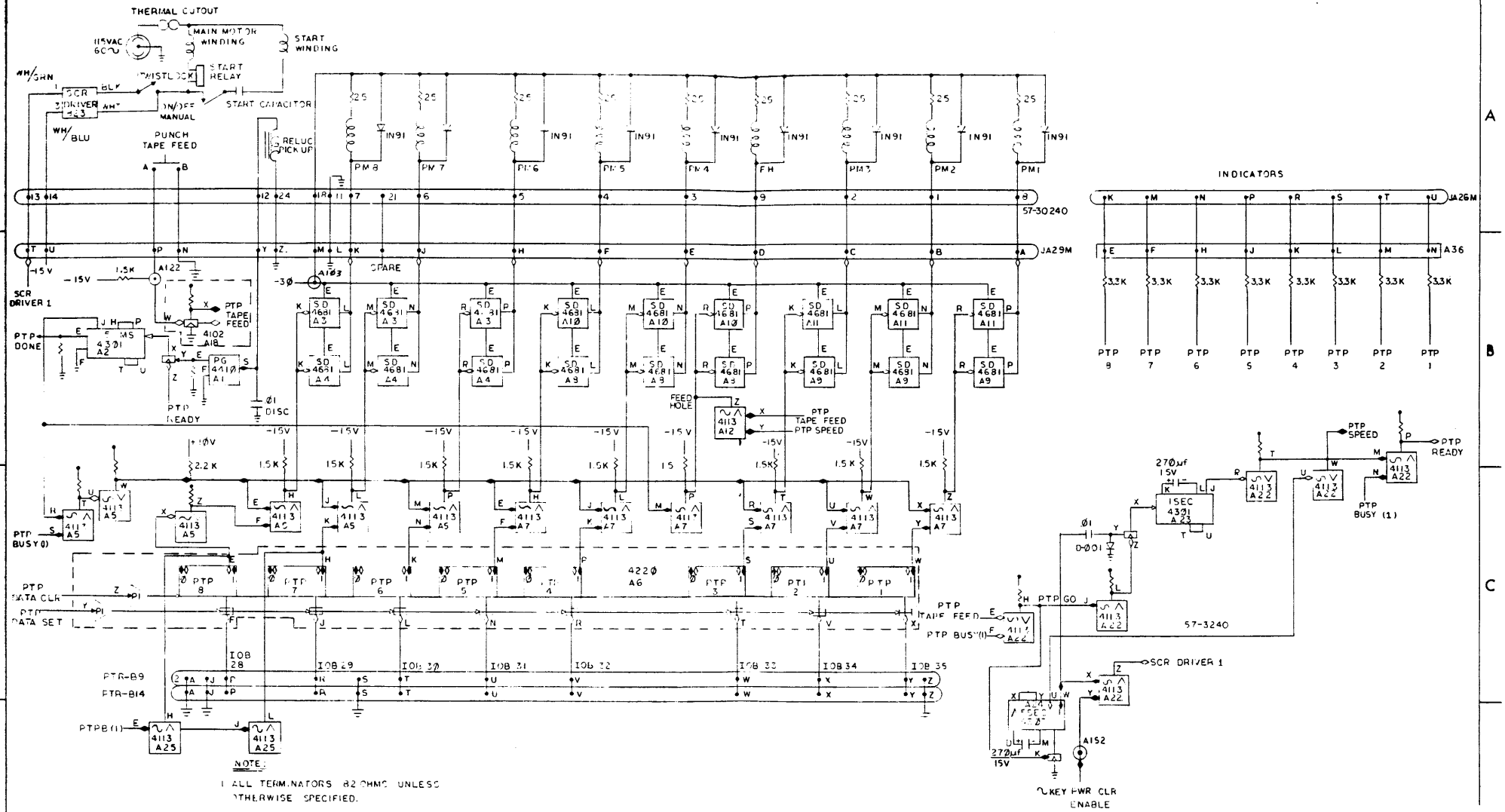


PTR – Paper Tape Reader Control 760
(BS-D-760-0-2, sheet 2)

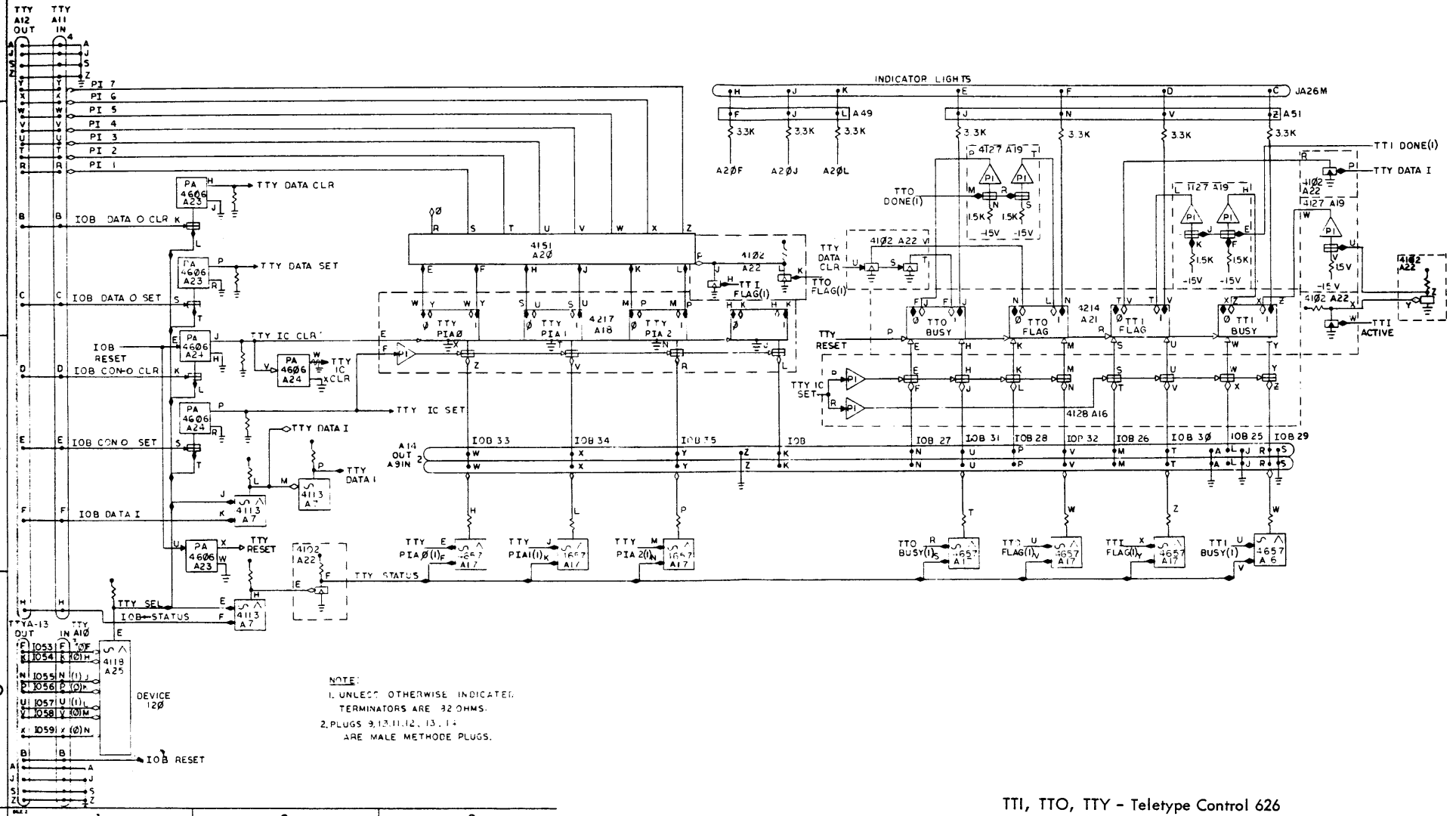


NOTE:
UNLESS OTHERWISE INDICATED
TERMINATORS ARE 82 OHMS.

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

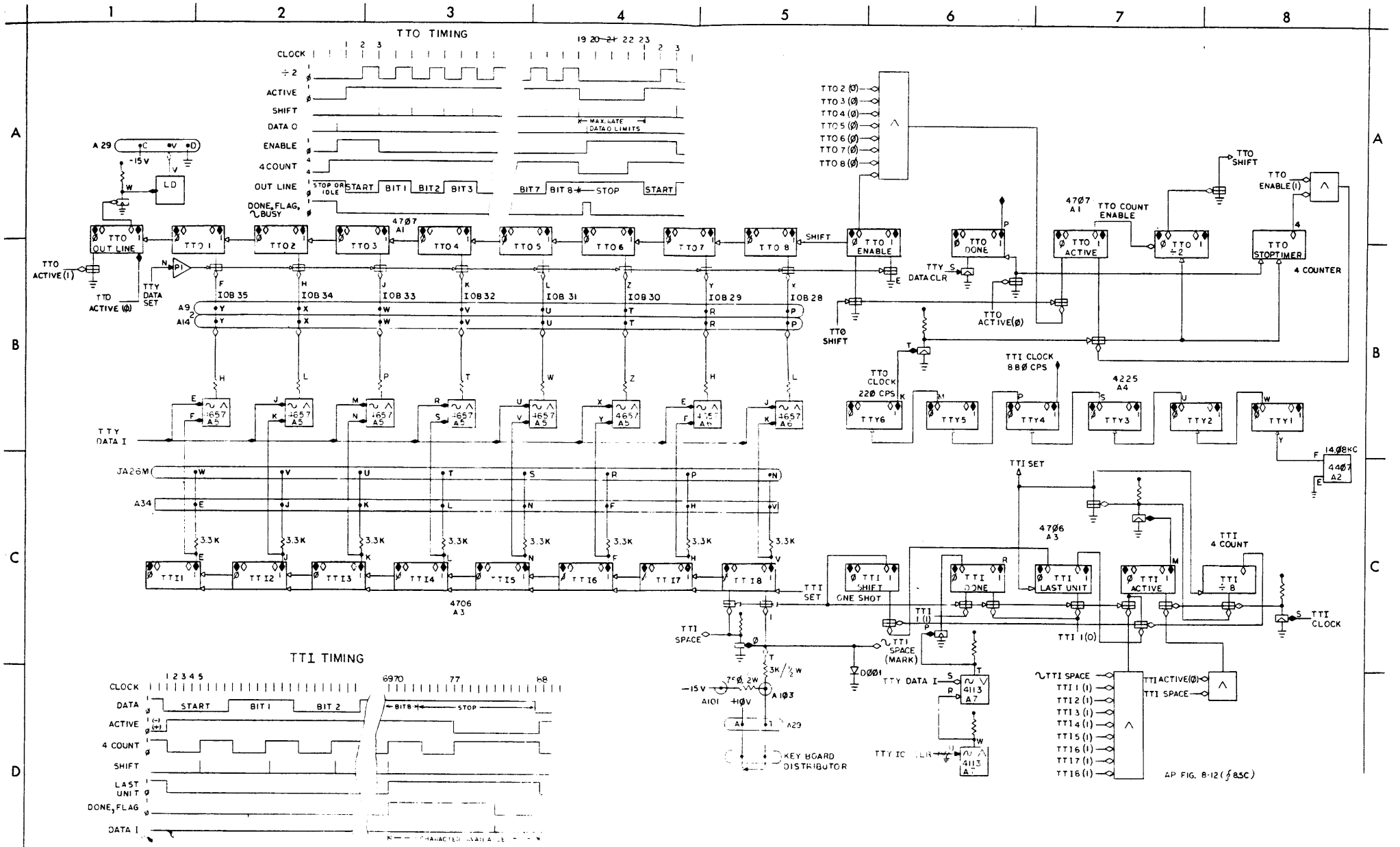


PTP - Paper Tape Punch Control 761
(BS-D-761-0-2, sheet 1)

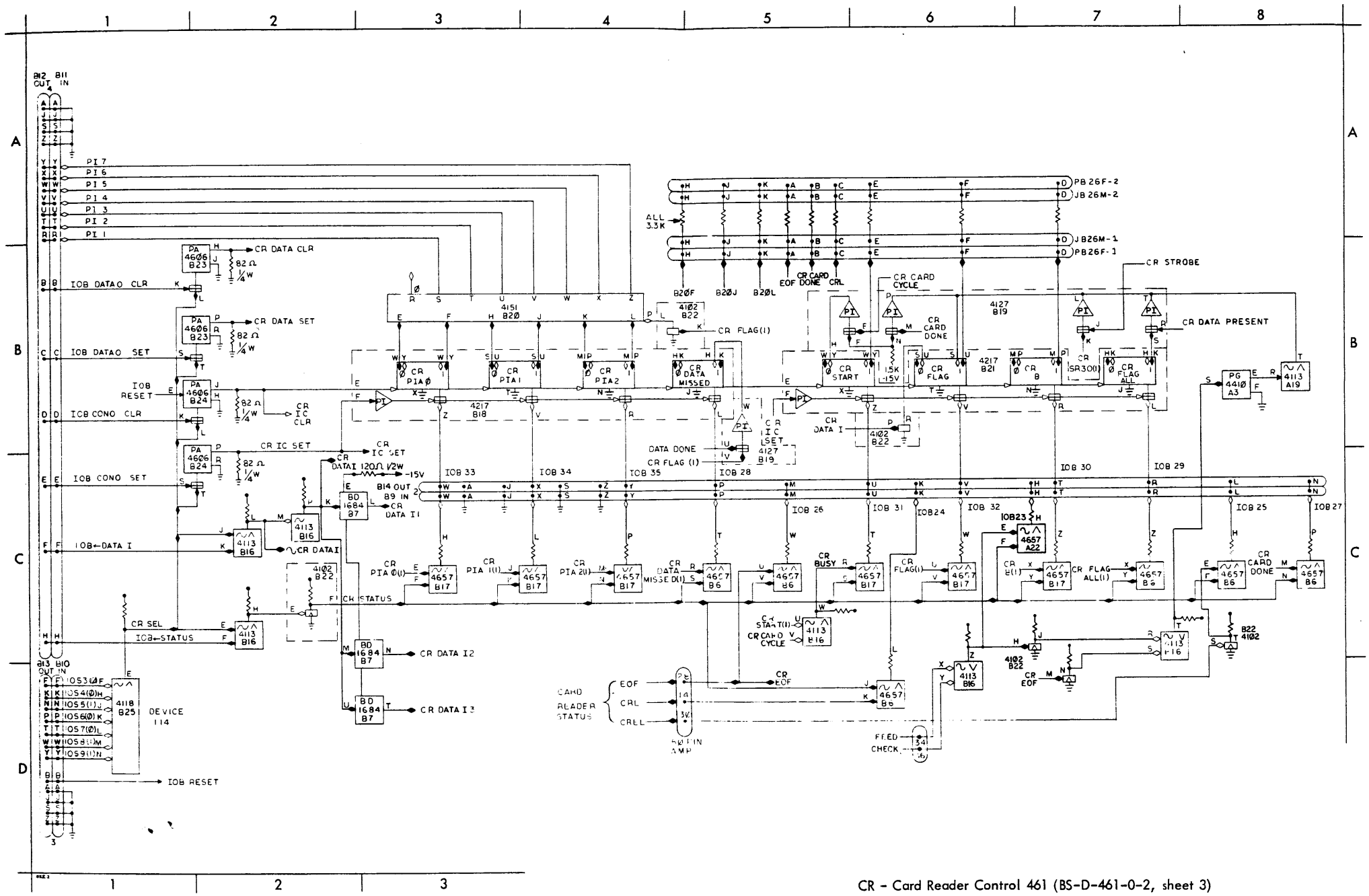


NOTE:
1. UNLESS OTHERWISE INDICATED, TERMINATORS ARE 82 OHMS.
2. PLUGS 9, 13, 11, 12, 13, 14 ARE MALE METHODE PLUGS.

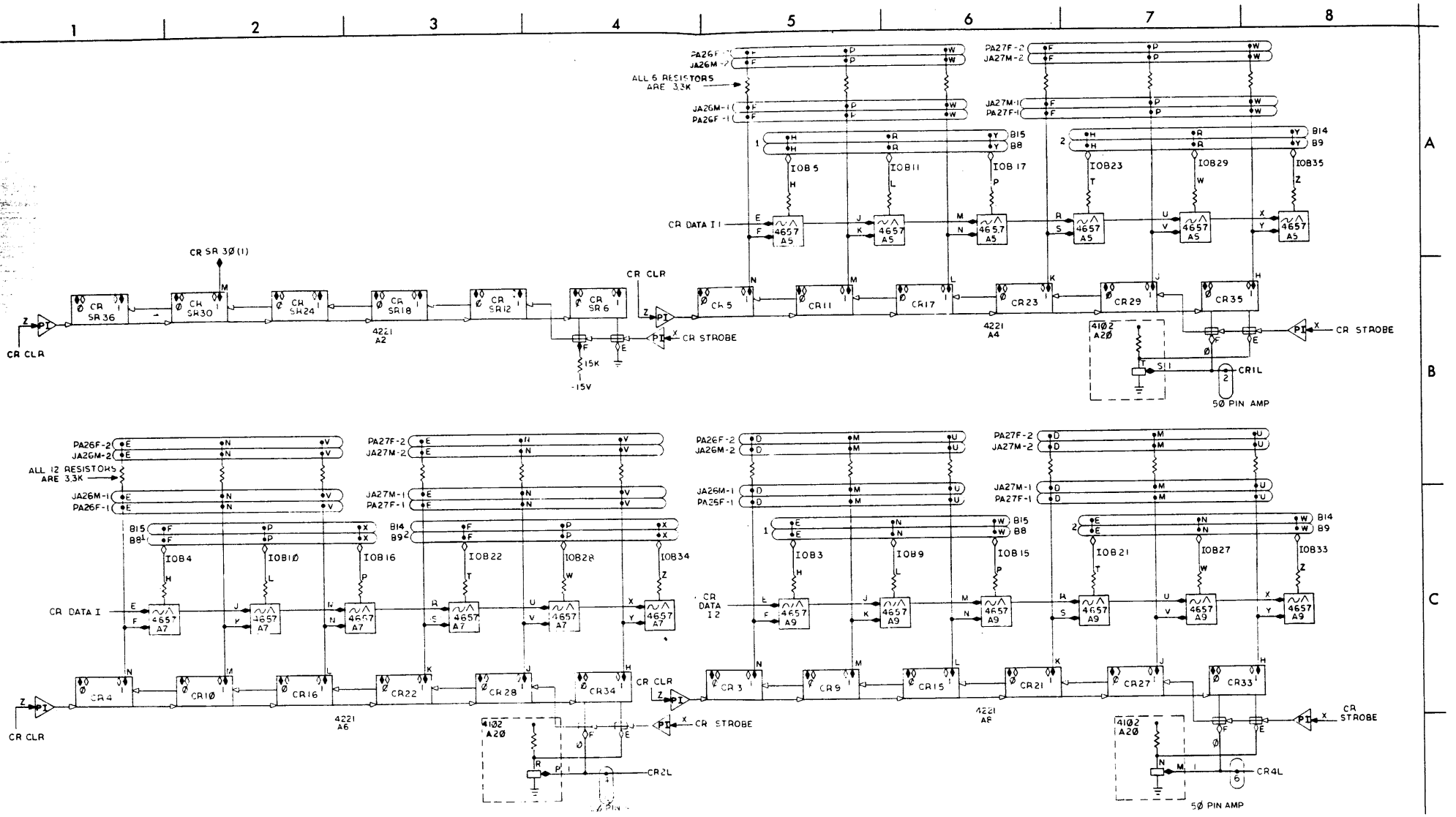
TTI, TTO, TTY - Teletype Control 626
(BS-D-626-0-2, sheet 1)



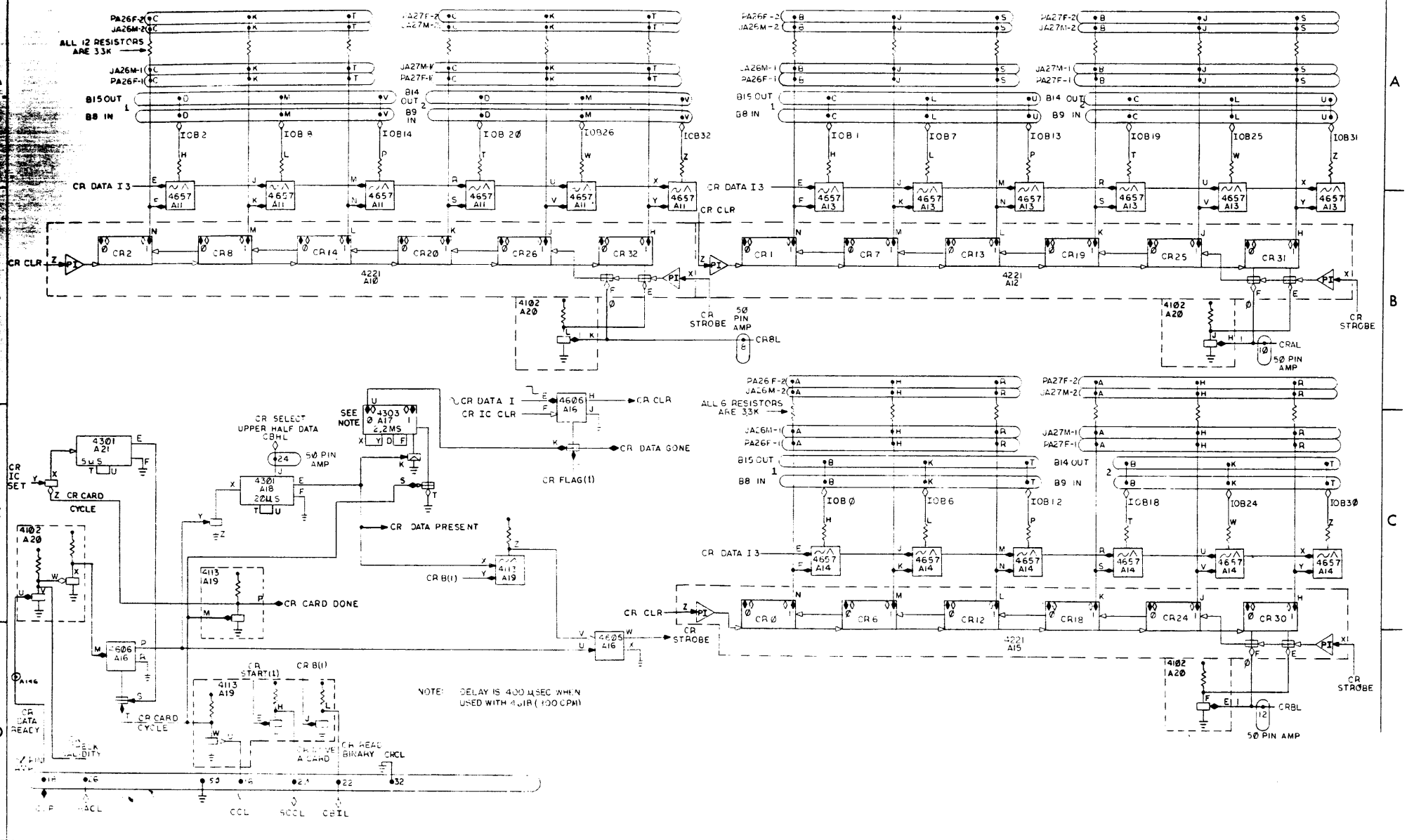
TTI, TTO, TTY – Teletype Control 626
(BS-D-626-0-2, sheet 2)



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 2)