JERNINDUSTRIENS FORLAG

Interfaceteknik

1985

Øvelser

Jern- og Metalindustrien

Best. nr. 4060

Forord

Lærebogen anvendes i undervisningen på Metalindustriens Efteruddannelseskursus nr. 5821 Interfaceteknik og EFG-uddannelsen, Datamekaniker, 2. del, trin 2 og 3.

Lærebogen er udarbejdet på foranledning af Metalindustriens Efteruddannelsesudvalg.

Faglærere fra Sønderborg tekniske skole har udarbejdet og tilrettelagt lærebogen i samarbejde med Jernindustriens Forlag.

Lærebogen er delt op i følgende afsnit:

Øvelser
Dokumentation

Øvelserne omfatter:

Interrupt
Direct memory Access
Programstyret Access
Tastatur interface
Display interface
Printer interface
Disk interface
Digital proces interface
Analog proces interface

Dokumentationen omfatter:

Interrupt
Direct memory Access
Programstyret Access
Tastatur interface
Display interface
Printer interface
Disk interface
Digital proces interface
Analog proces interface

Bladene er forsynet med huller og kan sættes ind i ringbind, efterhånden som de tages i brug.

Til brug ved undervisningen har lærebogen fortløbende sidenumre nederst på siderne.

I forbindelse med brug af lærebogen skal der også anvendes en diskette med "Kildetekst - programmel". Den kan bestilles på Jernindustriens Forlag.

Ud over programmel, der er på "Kildetekstdisketten", skal der anvendes færdigkøbt programmel, som fremgår af udstyrslisten til kurset ME 5821 eller Datamekaniker 2.del, trin 2.

København, september 1985

JERNINDUSTRIENS FORLAG

Indholdsfortegnelse

Interfaceteknik

MÅLEØVELSE DIGITAL PROCESSINTERFACE:

Formål: At undersøge signalforløbet på hhv. input og output-modul i et større processanlæg for herved at underbygge teorien omkring digital proces-interface.

Udstyr:
Transportanlæg (BÅND,ROBOT,BORD)
Logikanalysator (HP eller Philips)
Storage oscilloskop m/prober (Philips)
Målepind til skop-stel
2 stk. 20 pin IC-clips
Forlængerkort EUROPA-modul (special)
Fordeler stik med jordterminal

Hjælpemidler: Diagram til NESELCO F2002 in og out-putkort:
601 F 2601 / 601 F 2501
Manual til NESELCO F2002 PC

Gennemførelse:

Opstart af anlæg:

1) Sluk PC`erne (bag i tavle) og monter EEPROM´ER med programmer i de respektive anlæg. Tænd igen for PC´erne.

2) Overfør programmet til RAM ved at aktivere programnøgle og taste <PROM> L104 <INDLÆS>. (PC´en skal kvittere med ´INDLÆST´-lampen).

3) Drej AUT./MAN. omskifteren i stilling MAN. ved hhv. BORD og ROBOT

4) Tænd for hovedstrøm og hydraulik til anlæggene

5) Sæt PC´erne i RUN-mode ved at aktivere ´AUTO´tasten på betjeningspanelet 2 gange. (AUT. lampen skal lyse).

6) Set først BORD dernæst ROBOT i stilling AUT.

7) Kør med anlæget indtil der indikeres en målelig stationær værdi på ind- hhv. udgangs kortene på alle anlæg , stil evt. simulering i stilling MAN. (et eller flere bit skal være aktive på hhv. ind- og udgangs-kortet)

Måling på input kort:

1) Mål på tilslutningsklemmerne (tavle bag midte th.) på hhv. et signal der er "on" og et der er "off"

Uon = ___ V Uoff = ___ V

2) Mål signalerne efter brokoblingen:

Uon = ___ V Uoff = ___ V

Hvad skyldes den lavere spænding efter brokoblingen ?

_____

_____

Hvilket formål har PTC-modstanden ?

_____

3) Mål på signalerne efter optokoblerne

Uon = ___ V Uoff = ___ V

Sammenhold pkt.2 og 3, kommenter målingen :

_____

_____

4) Mål de logiske niveauer efter AND-gatene

NIV on = ____ NIV off = ____

5) Mål med LSA på udgangen af latch'en for hhv.
8-LSB = IC 15 og 8-MSB = IC 16
Clock på positiv flanke af "OE"-signalet (ben 1) ved den latch som måles på.

Stemmer værdierne overens med indgangssignalerne ?

_____

6) Mål med LSA på databussen IC 1 ben 2-9
(clock på positiv flanke af "EN"signalet ben 19)

Sammenhold målingen med indgangssignalerne og kommenter:

_____

_____

_____

Måling på outputkort:

1) Mål på databussen de data der sendes til outputkortet (IC 1 ben 2-9)

   Kommenter målingen:

   _____

   _____

2) Mål med LSA på latchen for hhv. 8-LSB = IC 13 og 8-MSB = IC 14

   Stemmer værdierne overens med de målte på databussen ?

   _____

3) Mål signalerne på udgangsklemmerne (tavle bag midte th.)

   Uon = ___ V Uoff = ___ V

4) Hvilken logikform anvendes på udgangene ?

   _____

5) Hvordan skal et relæ forbindes til udgangen ?

   _____

6) Mål med storageoscilloskop hvor lang tid der forløber imellem hver scanning af et bestemt udgangssignal

   Scanningtid:_____mSek.

MÅLEØVELSE ANALOG PROCESSINTERFACE:

Formål: At undersøge signalforløbet på A/D og D/A konvertering i et processanlæg for herved at underbygge teorien omkring analog processinterface.

Udstyr:
Køletårn m/NAF-ECA 30
Niveauregulering m/NAF-ECA 30 eller 35
Storageoscilloskop m/prober (Philips)
X/Y skriver m/BNC kabler og omsætter
2 stk. IC-clip 16-ben

Hjælpemidler:
Diagram af NAF ECA 30 og 35 regulator
Manual til NAF regulator

Gennemførelse:

Opstart af anlæg:

1) I-time indstilles til max
2) D-time indstilles til 30 sek. (30 x 1)
3) Gain indstilles til 30 gg. (3 x 10)
4) Aut/Man. knappen (M) stilles i Aut.
5) Setpunkt (SP) indstilles til 50%
6) Anlæget startes op
7) Afgangshanen (belastningen) åbnes til 50 %

Ben forbindelser MUX: (V 204)

ben 13 - In 0 = MV
ben 14 - In 1 = LSP/RSP (PD1 = 1 => RSP)
ben 15 - In 2 = stel
ben 12 - In 3 = stel
ben 1 - In 4 = stel
ben 5 - In 5 = GAIN
ben 2 - In 6 = I-time
ben 4 - In 7 = D-time

Måling på A/D konverteringen: (ECA 30/35)

1) Mål med et storage oscilloskop på regulatoren

2) Trig oscilloskopet på neg.flanke af mest betydende adr.indgang på MUX (V 204 ben 9) EXTERN TRIGNING

3) Mål udgangssignal fra MUX (V 204 ben 3) på kanal A og mindst betydende adr.bit på MUX (V 204 ben 11) på kanal B, gem målingen i "STO 1"

4) Mål på udgangen af integratoren (V206 ben 5) på kanal A og udgangen af comparatoren (V206 ben 7) på kanal B, gem målingen i "STO 2"
NB: SLUK REGULATOREN INDEN TILSLUTNINGEN

5) Plot kurverne ud på en X-Y skriver (på samme ark)

6) Påfør kurverne benævnelse hhv.: MUX OUT, MUX AO, INTEGRATOR, COMPARATOR

7) Sammenhold kurverne og noter på kurve 3 (INTEGRATOR) hvilke dele af kurven der er hhv.: MV, LSP, GAIN, I-time, D-time

8) Der kan evt. ændres på indgangsværdier og virkningen kan iagtages.

9) Hvilken sammenhæng er der imellem indgangsværdien og spændingens størrelse på udgangen af integratoren ?

_____

_____

10) Hvilken sammenhæng er der imellem indgangsværdien og pulstiden på udgangen af integratoren ?

_____

_____

11) Hvornår bliver integratoren resat ?

_____

_____

12) Hvor mange pulser modtager uP på RST 5.5 benet for hvert gennemløb ?

_____

KØLETÅRN (KUN ECA 30):

Måling på D/A konverteringen:

1) Mål med et oscilloskop på udgangsporten (V 202 ben 6)

2) Varier setpunktet til hhv.: 20% og 80% og iagttag samtidig signalet på porten samt udgangssignalet fra regulatoren (aflæses på regulatoren).

3) Hvad sker der med signalet fra porten ved hhv. 20% og 80% ?

   20 % : _____

   _____

   80 % : _____

   _____

4) Hvad sker der med udg.signalet fra regulatoren ?

   20 % : _____

   80 % : _____

5) Hvilken funktion har MMV'en (V 203) i forbindelse med udgangssignalet ?

   _____

   _____

6) Hvilken funktion har det analoge udgangsmodul ?

   _____

   _____

   _____

NIVEAU REGULERING (KUN ECA 35):

Måling på D/A konverteringen:

1) Mål på udgangen af ROM/PORT-kredsen 8355 (V201 ben 28 og 29) med oscilloskopet

2) Varier setpunktet til hhv. 20% og 80% og iagtag samtidig hvad der sker med udgangene fra porten.

   Hvad sker der ved 20% hhv. 80% ?

   20 % : _____

   _____

   80 % : _____

   _____

3) Hvilken funktion har AMV`en (V 209) i forbindelse med udgangssignalet ?

   _____

   _____

   _____

   _____

4) Må de to NAND-gates (V 213) som styrer SCR´ene være aktive samtidig ?

   _____

Måleøvelse: Scanning keyboard.

Formål: Formålet med øvelsen er at undersøge sammenhængen mellem row, column og tæller i et scanning keyboard.

Udstyr:

Comet 3400 / keyboard.

Philips logic state analyzer, eller
HP logic state analyzer.

IC clips: 3 stk. 16 pins og 1 stk. 40 pins.

Hjælpemidler:

Diagrammer til Comet 3400.

Gennemførelse:

Sæt LSA'en op til følgende:

Funktion: state analyzer.

Clock: IC 1, pin 3, bagkant.

Opsæt en menu så følgende format fremkommer:

	Row - dekoder	Row - dekoder	Column MUX	Counter	Encoded output
Philips:	IC 5	IC 5	IC 4	IC 3	
	xxxxxxxx	xxxxxxxx	xxxxxx	xxxxxxx	
	d15---d8	d7----d0	i5--i0	o6---o0	
HP:	IC 5	IC 5	IC 4	IC 3	IC 9,10
	xxxxxxxx	xxxxxxxx	xxxxxx	xxxxxxx	xxxxxxxx
	d15---d8	d7----d0	i5--i0	o6---o0	d7----d0

1 0 4 3 2

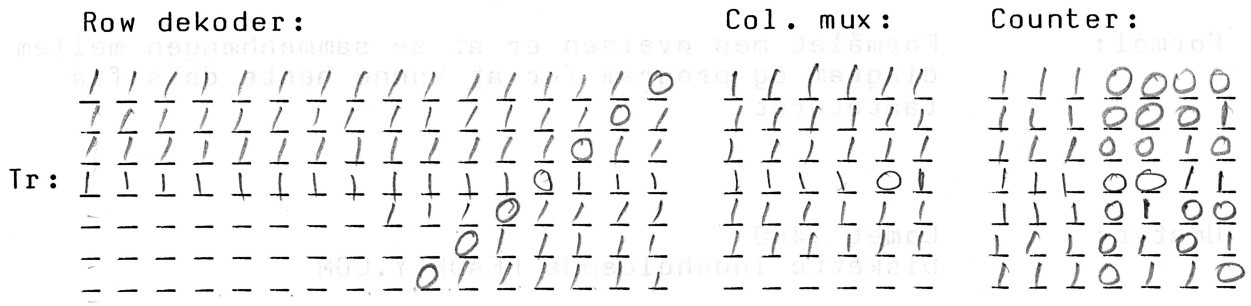
Trig på indholdet af IC 4 = 111101 B, (center).

Start opsamling af data.

Aktiver tast 'S' eller 's', hvorefter LSA'en fuldfører dataopsamlingen.

Beskriv forløbet på næste side:

1) Indtegn måleresultaterne fra LSA'en i skemaet, når tilstanden er stabil:



2) Beskriv output fra row dekoderen før og efter trig:

0 skiftes hele tiden i plads til venstre gennem løber det 2gange efter trig

uden prel vil tæller stoppe første gang

3) Beskriv input til column multiplexeren før og efter trig:

giver 3F ud både før og efter
4 3D ud ved trig

4) Beskriv output fra counteren før og efter trig :

counter tælles op med 1 indtil trig, derefter stoppes

5) HP: beskriv det encodede output før og efter trig:

før output 000
efter 073

6) Kunne et program have udført den samme funktion som kredsene, der netop er blevet analyseret ?

ja med passende hardware

Øvelse: Comet 3400 tastatur, keyboard buffer.

Formål: Formålet med øvelsen er at se sammenhængen mellem diagram og program for at kunne hente data fra tastaturet.

Udstyr: Comet 3400
Diskette indeholdende READKEY.COM

Hjælpemidler: Diagram over Comet 3400.
Dokumentation over READKEY.COM.
Datablade over anvendte TTL kredse.
Datablade over anvendte MOS kredse.

Gennemførelse:

Ved hjælp af diagrammet findes nummeret på den port hvorfra CPU'en aflæser keyboardet.

Dette portnummer skal erstatte værdien 0F4H, som i forvejen fejlagtigt er sat som portnummer i nedenstående program.

Udfør rettelsen i DDT'en og afprøv programmet.

Nr blues CB Hex Keyentry
Blues lagret på adr 1000H
Hvis vi ønsker tekst på skærm
må adr 1000H ændres til
EFFF

Aflæsning af keyværdi, SDK-85 evaluation kit.

Følgende programudsnit af SDK-85 monitor viser hvordan data når frem til accumulator A:
Ved aktivering af en tast genererer keyboard-controller 8279 en strobe, som aktiverer CPU'ens RST 5.5 indgang. Denne interrupt bliver accepteret når CPU'en har fuldført den instruktion den var i færd med, hvorefter adressen på den næste opcode gemmes ude på stack'en. Modulet RDKBD på adr 02E7H eksekveres og der returneres til det sted i programmet hvor CPU'en blev afbrudt, idet programcounterens oprindelige indhold hentes tilbage fra stacken.

```
0000H   START:  MVI   A,00H     ;Sæt keyboard til encoded scan og
                STA   1900H     ;2-key lockout.

002CH   RST5.5: JMP   ININT     ;Rst 5.5 adresse.

028EH   ININT:  PUSH  H         ;Gem de i programmet anvendte reg.
                PUSH  PSW       ;på stack.
                LXI   H,1900H   ;Sæt 8279 til: læs FIFO indhold.
                MVI   M,40H     ;
                DCR   H         ;Peg nu på adresse 1800H og
                MOV   A,M       ;læs keyværdi.
                ANI   3FH       ;0-stil de to mest betydende bit.
                STA   20FEH     ;Gem den læste værdi i RAM-lager
                POP   PSW       ;Bring de oprindelige register-
                POP   H         ;indhold tilbage og
                RET             ;returner.
```

Et andet sted i monitoren findes et modul, der henter indholdet af RAM-kopiet af seneste indtastning på adresse 20FEH. Denne værdi returneres i accumulator A.

```
02E7H   RDKBD:  LXI   H,20FEH   ;Hent RAM-kopi af indtastning.
                MOV   A,M       ;
                ORA   A         ;test MSB,
                JP    RDK10     ;hvis det er 1, er 8279 FIFO coun-
                EI              ;ter tom og check er slut....
                JMP   RDKBD     ;ellers: 'poll' keyværdi.
02F3H   RDK10:  MVI   M,80H     ;RAM kopi af keyværdi = 1 i MSB.
                DI              ;Udeluk mulighed for ny interrupt.
                RET
```

I RAM-lageret kan man nu afprøve de ovenfor omtalte moduler ved at indtaste og eksekvere følgende program:

```
2000H   TESTKB: LXI   SP,20C2H  ;Initialiser stackpointer.
                MVI   A,08H     ;Mask set enable + enable RST 5.5,
                SIM             ;RST 6.5 og RST 7.5.
                CALL  RDKBD     ;Læs keyværdi.
                RST   01H       ;Tilbage til monitor.
```

Herefter kan key-værdien aflæses i accumulator A.

Øvelse: Anvendelse af 8279 keyboard controller.

Hjælpemidler:

Diagram over SDK85 kit.

Datablad over INTEL 8279 keyboard controller.

Dokumentation over de aktuelle afsnit af monitor-programmet til SDK85 kit. (se forrige side).

Gennemførelse:

Hvilken værdi må ligge i accumulator A efter at TESTKB-rutinen er eksekveret, når tasten 'NEXT' har været aktiveret ?

Svar / begrundelse : _____

_____

_____

_____

MÅLEØVELSE INTERRUPT Z80-PIO:

Formål: At sætte en interrupt-port/controller og en microprocessor op til en bestemt interruptmode og måle på interrupt forløbet for herved at underbygge teorien omkring interrupt.

Udstyr:
Comet 3400 incl. I/O-port (MPS 10)
Diskette med INT.COM program
Printer Microline 84
Logik / Timing-analysator (Philips incl. Z 80 disassembler)
1 Stk. 40 pin IC clips
1 Stk. omsætterkabel MOLEX / CANON

Hjælpemidler:
Brugervejledning til MPS 10 I/O-kort
Microprocessor Interfacing Techniques (SYBEX)
Zilog databog (Z80 CPU / Z 80 PIO)
Dokumentation til INT.COM programmet

Gennemførelse:

1) Tilslut printeren til I/O-porten på Comet 3400

2) Tilslut Philips LTA til Comet`ens CPU og Acknowledge-signalet fra printeren:

   CPU`en tilsluttes efter standard til Z80 disassembleren (se Philips manual LTA)

   Acknowledge signalet tilsluttes: POD-T bit 0

PORT FORBINDELSER BLOK A Z80-PIO:

Port A: bit 0 = ACKNOWLEDGE
bit 1 = BUSY
bit 7 = STROBE

Port B: bit 0 - 7 = DATA

3) Find vha. program-udskrift og databog de parametre som skal indføjes i programmet for at få CPU og printer til at arbejde sammen i den ønskede interrupt-mode (anført med "*" i program) og udfyld:

a) INTDISW = 03 ;Interrupt disable word

b) VECTORI = 01 ;Vektor til I-reg.

c) VECTORP = 60 ;Vector til port

d) MOCTRWO = 0F ;Mode control word 0

e) MOCTRW3 = CF ;Mode control word 3

f) IOCTRW = 0F ;I/O reg. control word

g) INTCTRW = D7 ;Interrupt control word

h) MSKCTRW = FE ;Mask control word

4) Kald programmet: INT.COM sammen med ZSID fra diskette og indfør de fundne parametre i programmet på de rigtige adresser.

5) Afprøv programmet (G100), printeren skal nu skrive: "INITIALISERINGEN ER KORREKT !" og programmet skal herefter vende tilbage til ZSID`en.

VIRKER PROGRAMMET ? JA-GÅ TIL PKT. 6 NEJ-GÅ TIL PKT. 3

6) Set LTA op til:

OPTION: Z80 disassembler

CONFIG: COMBIMODE

Slet de timing-bit som ikke benyttes

TRIG:

FIND WORD 1
FIND WORD 2 ENABLE LTA,DELAY 1000 STATES OF CLOCK 0
END

TRIG.WORD 1 = "BEGYNDELSES ADR. PÅ HOVEDRUTINE"
TRIG.WORD 2 = "BEGYNDELSES ADR. PÅ INT.RUTINE"

DATA:

COUNTER ON
COUNT TIME
START AFTER WORD 1
FINISH AT WORD 2

7) Start LTA og kør programmet.

8) Mål hvor lang tid der forløber fra printeren anmoder om service til CPU`en starter eksekvering at interruptrutinen ?

Måling 1 = 6,8µ Sek. 6 µ

Måling 2 = 94µ Sek. 674µ

Måling 3 = 6,8µ Sek. 5,1µ

Middelværdi = 7,66µ Sek. 5,95µ

9) Hvilke faktorer i programmet indvirker på responsetiden ?

hvor lang tid i programmet I interupt er Enablet

10) Hvor lang tid ville der være forløbet hvis CPU`en istedet havde POLLET indgangene (en gang for hvert gennemløb) ?

Ca. antal instruktioner = 20 stk.
Ca. antal clockpulser pr.instruktion = 10 stk.
Clockpulsens periodetid = 250 nSek.

Ca. responsetid = 50µ Sek.

11) Hvad sker der med responsetiden hvis hovedprogrammet gøres længere (med POLLING) ?

Responce tiden bliver længere

12) Hvordan finder CPU`en jumpadressen til interruptrutinen ?

man loader I registeret med 01 i initialisering som high byte interruptvektor som low byte til den adresse i jump tabellen hvor jump adressen findes.

interrupt vektor skal ende på 0 det vil sige 0 i bit 0 da interrupt henter 2 adresse byte i vektor tabel

13) Hvilke andre muligheder er der for portkredsen at overføre en interrupt til CPU`en ?

i de forskellige modes kan strobe generere interrupts

14) Hvordan prioriteres interrupt`ene inden for en enkelt Z80 PIO ?

Port A højest prioritet

15) Hvordan prioriteres der når CPU`en tilkobles flere Z80 PIO`er ?

Den der ligger nærmest CPU har første prioritet

ØVELSE DMA CONTROLLER:

Nævn tre principper til data overførsel vha DMA:

Hvilket princip er det hurtigste til dataoverførsel ?

Hvilke ulemper har dette princip ?

Hvilket princip optager mindst af CPU`ens tid ?

Hvilke ulemper har dette princip ?

Øvelse 1: Anvendelse af 6845 CRT controller.

Formål:

Formålet med øvelsen er at ændre initialiseringen af en CRT controller, så en hensigtsmæssig funktion opnås.

Udstyr:

Comet 3400.
Diskette indeholdende EDITOR.COM.

Hjælpemidler:

Diagram over Comet 3400.
Datablad over Motorola 6845 CRT controller.
Dokumentation over EDITOR.COM.

Gennemførelse:

EDITOR skal opfattes som et lille program, der først og fremmest illustrerer hvordan 6845 CRT-controller initialiseres, og dernæst hvordan en tekst kan placeres på skærmen. Programmet eksekveres i DDT'en fra adr. 0100H.

Visse registre er i initialiseringen sat forkert op, så EDITOR programmet ikke virker hensigtsmæssigt.

Foretag de nødvendige ændringer i kontrolregistrene i 6845, så følgende egenskaber fremkommer:

80 karakterer pr. linje.
Bredde på liniesync-puls: 10 kar. (5 uSek).
24 linjer pr billede.
Cursor-blink: 1/16 field rate.
Cursor-højde: 9 linjer.
Cursor-position: 1. skrivefelt i 3. linje.

Programmet gennemfører initialiseringen via modulet INIT samt tabellen TABEL1.

Når rettelserne er udført, skal det nu være muligt at skrive på ethvert sted på skærmen, men cursoren bliver stadig stående i sin udgangsposition.
Denne fejl rettes i næste øvelse.

Øvelse 2: Anvendelse af 6845 CRT controller.

Formål: Formålet med øvelsen er at opnå kontrol over en cursor via kontrolregisterene i en CRT controller.

Udstyr:

Comet 3400.
Diskette indeholdende EDITOR.COM.

Hjælpemidler:

Datablad over Motorola 6845 CRT controller.
Dokumentation over EDITOR.COM.
Diagrammer over Comet 3400.

Gennemførelse:

Cursorens position i EDITOR.COM kontrolleres af modulet CURSOR.

Modulet skal overføre indholdet af HL registeret til kontrolregistrene i 6845, som herefter placerer cursoren på sin rette position.

I dette modul optræder fejl, så cursoren ikke kan bevæges.

Ret fejlen(e), så cursoren kan styres.

Måleøvelse: CRT kredsløb.

Formål: Formålet med øvelsen er at undersøge virkemåden af en computers CRT-kredsløb, ved at undersøge sammenhængen mellem:

* Refresh-adresse til video-RAM fra CRTC.
* Det til refresh-adressen hørende data (ASCII).
* Row select til karaktergenerator.
* Output fra karaktergenerator (Dots).

Udstyr:
Comet 3400.
Diskette indeholdende:
TEKST.COM
LOLIGHT.COM
HILIGHT.COM
Logic state analyzer (HP eller Philips).
Storage scope (Philips).
2 stk 40 pin IC - clips.

Hjælpemidler:
Diagrammer til Comet 3400.
Dokumentation over:
TEKST.COM
LOLIGHT.COM
HILIGHT.COM
Datablad over Motorola 6845.
Datablad over RAM 4016.
Datablad over EPROM 2716.

Gennemførelse: Fjern dækslet til Comet 3400 og afmonter modulet MPS-26B.

Placer istedet dette modul på 'expansion'-stikket på bagpladen af computeren.

Herefter føres forbindelsen til VIDEO 1 gennem et af hullerne i computerens bagplade.

Sæt LSA'en op som state analyzer.

Forbind LSA'en til MPS-26, og sæt menuen på LSA'en så følgende format fremkommer:

Video-RAM-adresse:	Row karakter-generator:	Video-RAM data:	DOT - output:
IC 10	IC16	IC10	IC16
XXXXXXXXXXX	XXXX	XXXXXXX	XXXXXXXX
A10------A0	R3-0	D6---D0	D7----D0
BIN	BIN	ASCII	BIN

NB: Diagrammets nummerering af IC 16 er soklens numre, og ikke EPROM'ens !!!!!!

Clock = "Latch Kar. Gen" bagkant (IC 15, pin 11).

Trace: Single trace.

Eksekver TEKST.COM i DDT'en fra adresse 0100H.

Trig på video-RAM-adr = 00000000000 (BIN),
og ROW (Kar. gen) = 0000 (BIN)
(Billed start).

Opsæt LSA'en, så alle states udlæses.

Foretag en opsamling af data.

Bemærk, at output fra EPROM (DOT's) først udlæses een clockpuls senere end udlæsningen af den tilhørende adresse !

1) Forklar årsagen til dette udfra diagrammet:

forskellige clock pulser
LSA samler op på negativ flanke
hvad der sker der imellem ved vi
ikke

2) Afmærk de aktive niveauer fra output / EPROM i nedenstående tabel:

Video-RAM adresse:	ROW kar.-gen.:	Video-RAM-Data:	Dot - output fra kar. gen.: (1 cp senere)
A10-A0 BIN	R3-0 BIN	D6-0 ASCII	D7-D0 BIN
0000000000	0000	(0)	0 0 0 0 0 0 0 0
0000000000	0001	(0)	0 0 1 1 0 0 0 0
0000000000	0010	(0)	0 1 0 0 1 0 0 0
0000000000	0011	(0)	0 1 0 0 1 0 0 0
0000000000	0100	(0)	0 1 0 0 1 0 0 0
0000000000	0101	(0)	0 1 0 0 1 0 0 0
0000000000	0110	(0)	0 1 0 0 1 0 0 0
0000000000	0111	(0)	0 0 1 1 0 0 0 0

3) Beskriv sammenhængen mellem adresse, row, karakterdata fra RAM og DOT data fra EPROM:

ascii karakter fra videoram er adresse
ført til karakter rom hvor de laveste
adresse bit tælles op af row decoder
row teller giver hvilken linie af karak
teren der udskrives

4) Lad LSA'en trigge på:

Video-RAM adr. = 00000000101 (BIN) 5

og foretag nu en dataopsamling af de hændelser, der finder sted på trigger-tidspunktet.

Hvilken karakterlinje på CRT'en måles der på ?

første linie på 5tal

Beskriv DOT output fra EPROM: intet output

bit mønster for omvendt 5tal

5) Gentag forløbet fra før, men denne gang trigges på

Video-RAM adr. = 00010100101 (BIN) 165

Hvilken karakterlinje på CRT'en måles der på ?

3 kavalkte linie på 5talet

Dannes linjens karakteristiske egenskab i karak-generatoren ?

nej underline

Beskriv DOT output fra EPROM: intet output

bit mønster for omvendt 5tal

6) Gentag forløbet fra før, men denne gang trigges på

Video-RAM adr. = 00101000101 (BIN) 256 64 4 1 325

Hvilken karakterlinje på CRT'en måles der på ?

5 kavalkte linie på 5tallet

Dannes linjens karakteristiske egenskab i karak-

tergeneratoren ? nej Reverse

Beskriv DOT output fra EPROM:

bit mønster for omvendt 5tal

7) Eksekver programmet LOLIGHT.COM i DDT'en fra adresse 0100H.

Mål output J1 med storage scope.

Angiv spændingsniveaurne for:

Hvidt: 1,6 V

Sort: 0,7 0,4 V

Sync.: 0,3 0 V

8) Eksekver programmet HILIGHT.COM i DDT'en fra adresse 0100H.

Mål output J1 med storage scope.

Angiv spændingsniveauerne for:

Hvidt: 2,4 V

Sort: 0,7 V

Sync. 0,3 V

Øvelse: Højopløsningsgrafik.

Formål: Formålet med øvelsen er at foretage ændringer i et program, der styrer en grafik-processor, så en ønsket funktion opnås.

Udstyr: Comet 3400.
Diskette indeholdende FUNKTION.COM

Hjælpemidler: Brugervejledning, MPS-24 grafikmodul.
Dokumentation over FUNKTION.COM

Gennemførelse:

Monter grafikmodulet i Comet 3400. Vær opmærksom på krystallet

Forbindelsen fra video-modulet til CRT'en fjernes.

Istedet føres denne forbindelse fra nederste stik på grafik-modulet til CRT'en (Stel nederst).

Opret en forbindelse mellem udgangen fra video-modulet til det næst-nederste stik på grafik-modulet (Stel nederst).

Eksekver FUNKTION.COM via DDT`en fra adr. 0100H.

Skift tilbage til video foregår ved at aktivere en vilkårlig tast.

Udfør følgende ændringer ved hjælp af DDT'ens faciliteter:

1) Ret 'f(Y)' ved y-aksen til 'f(X)'.

2) Placer pilen korrekt på y-aksen.

3) Den ustiplede funktion skærer y-aksen i '4' og x-aksen i '2'.
Placer den ustiplede funktion, så skæringspunktet med y-aksen forbliver '4', men skæringspunktet med x-aksen bliver '5'.

4) Ret teksten 'Koordinatsystem' til 'Årlig omsætning'.

Øvelse: Floppy disk controller.

Formålet med øvelsen er at opsætte en multi-byte instruktion til en floppy disk controller, så en ønsket funktion opnås.

Udstyr: Comet 3400.

Diskette indeholdende DISKREAD.EKS

Hjælpemidler: Datablad over uPD 765 A.

Brugervejledning til Comet 3400.

Dokumentation over DISKREAD.EKS

Udskrift af i forvejen ilagte instruktioner til programmet DISKREAD.EKS.

Gennemførelse:

For at kunne læse et spor fra disketten er det nødvendigt at eksekvere instruktionen SEEK for at placere læse/skrive hovedet på det ønskede spor. Denne instruktion er i forvejen sat op til at læse spor 1.
Instruktionen eksekveres fra adr. 0450H.

Det er yderligere hensigtsmæssigt at kontrollere at læse/skrive hovedet er placeret på det ønskede spor.
Instruktionen, der giver tilbagemelding om bl. a. det aktuelle spor hedder READ ID.
Denne instruktion er også sat op på forhånd, og kan eksekveres fra adr. 0440H.

Med denne mulighed for at placere hovedet og identificere dets position kan følgende opgave nu løses:

Indsæt de nødvendige parametre fra adresse 03D0H og fremefter, så de tilsammen danner instruktionen READ DATA. Instruktionen skal medføre, at sektor 1 på spor 1 bliver aflæst. Der skal læses fra side 0.
Øvrige specifikationer fremgår af databladet for FDC'en og af brugervejledningen til Comet 3400.

Instruktionen READ DATA kan nu eksekveres fra adr. 0470H. Læse/skrive hovedet må dog forinden være placeret på spor 1 med instruktionen SEEK, og positionen kan kontrolleres med READ ID.

Besvar spørgsmål til forløbet på de to næste sider !

1) Eksekver SEEK instruktionen fra adr. 0450.

Gives der tilbagemelding (results) efter at instruktionen er udført ?

Ved 00

Hvor må læse/skrive hovedet befinde sig nu?

drive 0 side 0 spor 1

2) Eksekver instruktionen READ ID fra adr. 0440H.

Angiv de værdier, som READ ID afleverer i resultat-fasen:

Cylinder number : 01

Head address: 00

Sector number: 0A

Bytes / sector: 02

D200

10 han varierer fo 01 – 0A tel

3) Hvis læse/skrive hovedet er placeret korrekt, skal READ DATA nu eksekveres fra adresse 0470H. De læste data bliver placeret fra adr. 0800H og fremefter.

Beskriv den læste sektor fra disketten:

Directory fra disk

Angiv de værdier, som READ DATA afleverer i resultat-fasen:

Cylinder number: 01

Head address: 00

Sector number: 02

Bytes / sector: 02

```
dir

A: LOLIGHT  COM : HILIGHT  COM : EDITOR   COM : DDT      COM
A: PRTEKS   COM : READKEY  COM : TTY      COM : TRANS    COM
A: FUNKTION COM : TEKST    COM : IEC      COM : INT      COM
A: PRT      COM : ZSID     COM : DISKREAD EKS : READ     COM
A: IHP      COM : IHP          : D
A>era lolight.com
```

Lolight slettes

```
A>dir

A: HILIGHT  COM : EDITOR   COM : DDT      COM : PRTEKS   COM
A: READKEY  COM : TTY      COM : TRANS    COM : FUNKTION COM
A: TEKST    COM : IEC      COM : INT      COM : PRT      COM
A: ZSID     COM : DISKREAD EKS : READ     COM : IHP      COM
A: IHP          : D
A>lolight er væk
```

Kontrol af at Lolight er stellet

```
LOLIGHT?
```

med hoved på detektor

```
A>ddt d
```

loadprogram

```
DDT VERS 2.2
NEXT  PC
0E00 0100
-g450
```

find track 1

```
*0139
-d22g440
```

kontroler at hoved er på track 1

```
*0139
-d200

0200 00 01 02 03 04 05 06 07 08 09 52 45 41 44 49 44 ..........READID
0210 02 4A 00 FF FF FF FF FF FF FF FF 43 4F 4D 44 53 .J.........COMDS
0220 FF FF FF FF FF FF FF FF FF FF FF 45 58 43 55 54 ...........EXCUT
0230 07 00 00 00 01 00 01 02 FF FF FF 52 53 55 4C 54 ...........RSULT
0240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

```
0260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
02A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
02B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-g470
```

læsdata

E5 = "initidata

```
*0139
```

Der er lagt E5 i første position på

```
-d800,9ff
```

dump data Lolight.com

```
0800 E5 4C 4F 4C 49 47 48 54 20 43 4F 4D 00 00 00 01 .LOLIGHT COM....
0810 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0820 00 48 49 4C 49 47 48 54 20 43 4F 4D 00 00 00 01 .HILIGHT COM....
0830 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0840 00 45 44 49 54 4F 52 20 20 43 4F 4D 00 00 00 06 .EDITOR  COM....
0850 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0860 00 44 44 54 20 20 20 20 20 43 4F 4D 00 00 00 26 .DDT     COM...&
0870 04 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0880 00 50 52 54 45 4B 53 20 20 43 4F 4D 00 00 00 20 .PRTEKS  COM...
0890 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
08A0 00 52 45 41 44 4B 45 59 20 43 4F 4D 00 00 00 01 .READKEY COM....
08B0 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
08C0 00 54 54 59 20 20 20 20 20 43 4F 4D 00 00 00 20 .TTY     COM...
08D0 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
08E0 00 54 52 41 4E 53 20 20 20 43 4F 4D 00 00 00 06 .TRANS   COM....
08F0 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0900 00 46 55 4E 4B 54 49 4F 4E 43 4F 4D 00 00 00 11 .FUNKTIONCOM....
0910 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0920 00 54 45 4B 53 54 20 20 20 43 4F 4D 00 00 00 01 .TEKST   COM....
0930 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0940 00 49 45 43 20 20 20 20 20 43 4F 4D 00 00 00 0A .IEC     COM....
0950 0E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0960 00 49 4E 54 20 20 20 20 20 43 4F 4D 00 00 00 03 .INT     COM....
0970 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0980 00 50 52 54 20 20 20 20 20 43 4F 4D 00 00 00 03 .PRT     COM....
0990 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
```

```
09A0 00 5A 53 49 44 20 20 20 20 43 4F 4D 00 00 00 50 .ZSID    COM...P
09B0 11 12 13 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09C0 00 44 49 53 4B 52 45 41 44 45 4B 53 00 00 00 10 .DISKREADEKS....
09D0 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
09E0 00 52 45 41 44 20 20 20 20 43 4F 4D 00 00 00 02 .READ    COM....
09F0 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-s800
```

Erstat første byte i LoLight.com dir med 00

```
0800 E5 00
0801 4C .
```

kontrollen skrive registre

```
-d1000,1ff1ff
1000 78 C3 F3 0D E1 F1 CA 28 0E 23 22 B9 0F EB 21 A8 x......(.#"...!.
1010 06 4E 23 46 CD 57 08 DA 28 0E CD C8 0D 2A 4A 0F .N#F.W..(....*J.
1020 EB 3E 82 B7 37 C3 85 08 FB 2A 4D 0F 7C B5 CA 4E .>..7....*M.|..N
1030 0E 2B 22 4D 0F CD 1F 0C C2 4E 0E 3A 4C 0F B7 C2 .+"M.....N.:L...
1040 48 0E CD 85 0E C3 85 08 CD 44 0D C3 85 08 CD C8 H........D......
1050 0D 3E 2A CD C7 0B 2A B9 0F CD 93 09 D2 62 0E 22 .>*...*......b."
1060 0C 00 CD 2E 0C 2A B7 0F 22 5D 0F C3 FE 06 11 0D .....*.."]......
1070 00 21 2F 0F 7E A0 23 BE 23 CA 81 0E 14 1D C2 74 .!/.~.#.#......t
1080 0E 5A 16 00 C9 2A B9 0F 46 23 E5 CD 6E 0E 21 49 .Z...*..F#..n.!I
1090 0F 73 21 9C 0E 19 19 5E 23 56 EB E9 B8 0E E0 0E .s!....^#V......
10A0 B8 0E E0 0E BE 0E F2 0E 04 0F 26 0F 26 0F 23 0F ..........&.&.#.
10B0 23 0F 19 0F 26 0F 14 0F CD CE 0E C2 29 0F CD D9 #...&.......)...
10C0 0E C3 29 0F 3A A8 06 BB C0 3A A9 06 BA C9 C1 E1 ..).:....:......
10D0 5E 23 56 23 E5 C5 C3 C4 0E 2A B5 0F 5E 23 56 C9 ^#V#.....*..^#V.
10E0 CD CE 0E CA ED 0E C1 C5 3E 02 C3 2B 0F D1 D5 C3 ........>..+....
10F0 29 0F 78 FE FF C2 FC 0E AF C3 2D 0F E6 38 5F 16 ).x.......-..8_.
1100 00 C3 29 0F 2A B7 0F EB CD C4 0E C2 29 0F C3 BE ..).*.......)...
1110 0E C3 29 0F D1 D5 C3 29 0F CD D9 0E C1 C5 3E 02 ..)....).....>.
1120 C3 2B 0F D1 13 D5 D1 13 D5 3E 01 3C 37 E1 C9 FF .+.......>.<7...
1130 C3 C7 C2 FF CD C7 C4 FF C9 C7 C7 FF E9 C7 06 C7 ................
1140 C6 CF 01 E7 22 C7 C0 F7 D3 03 CD 39 08 CD 2E 08 ....".......9....
1150 CD 13 08 FE 1A C2 59 18 C9 C3 DB 17 C9 01 8F 03 ......Y.........
1160 CD AF 09 C9 2A 20 1D 4D CD 5E 08 11 9E 03 01 E1 ....* .M.^......
1170 1D CD FD 15 32 B6 1D 01 E1 1D C5 1E 03 01 55 1F ....2.........U.
```

```
1180 CD 18 0A 3A E1 1D E6 7F 32 E1 1D 3A E2 1D E6 7F ...:....2..:....
1190 32 E2 1D 01 A2 03 CD EA 15 01 D8 1D CD B3 08 01 2...............
11A0 D8 1D CD E3 08 3A 5F 1E FE FF C2 B3 18 01 A5 03 .....:_.........
11B0 CD AF 09 21 F8 1D 36 00 21 00 00 00 20 90 00 40 ...!..6.!... ..@
11C0 00 08 21 10 92 10 21 12 42 48 00 09 10 02 40 00 ..!...!.BH....@.
11D0 10 40 08 08 41 02 00 82 42 48 09 09 20 42 21 01 .@..A...BH.. B!.
11E0 20 08 22 12 11 10 10 88 42 48 49 24 24 92 42 49  ."......BHI$$.BI
11F0 24 42 49 20 84 24 84 10 92 09 10 92 48 49 09 20 $BI .$......HI.
```

-m800,9ff,1000, Flytt de første data fra directory til adresse 1000H fra 800 H.I 1Blokk

-d1000,11ff

```
1000 00 4C 4F 4C 49 47 48 54 20 43 4F 4D 00 00 00 01 .LOLIGHT COM....
1010 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1020 00 48 49 4C 49 47 48 54 20 43 4F 4D 00 00 00 01 .HILIGHT COM....
1030 02 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1040 00 45 44 49 54 4F 52 20 20 43 4F 4D 00 00 00 06 .EDITOR  COM....
1050 03 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1060 00 44 44 54 20 20 20 20 20 43 4F 4D 00 00 00 26 .DDT     COM...&
1070 04 05 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1080 00 50 52 54 45 4B 53 20 20 43 4F 4D 00 00 00 20 .PRTEKS  COM...
1090 06 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
10A0 00 52 45 41 44 4B 45 59 20 43 4F 4D 00 00 00 01 .READKEY COM....
10B0 08 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
10C0 00 54 54 59 20 20 20 20 20 43 4F 4D 00 00 00 20 .TTY     COM...
10D0 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
10E0 00 54 52 41 4E 53 20 20 20 43 4F 4D 00 00 00 06 .TRANS   COM....
10F0 07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1100 00 46 55 4E 4B 54 49 4F 4E 43 4F 4D 00 00 00 11 .FUNKTIONCOM....
1110 0C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1120 00 54 45 4B 53 54 20 20 20 43 4F 4D 00 00 00 01 .TEKST   COM....
1130 0D 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1140 00 49 45 43 20 20 20 20 20 43 4F 4D 00 00 00 0A .IEC     COM....
1150 0E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1160 00 49 4E 54 20 20 20 20 20 43 4F 4D 00 00 00 03 .INT     COM....
1170 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
1180 00 50 52 54 20 20 20 20 20 43 4F 4D 00 00 00 03 .PRT     COM....
```

```
1190 10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
11A0 00 5A 53 49 44 20 20 20 20 43 4F 4D 00 00 00 50 .ZSID    COM...P
11B0 11 12 13 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
11C0 00 44 49 53 4B 52 45 41 44 45 4B 53 00 00 00 10 .DISKREADEKS....
11D0 14 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
11E0 00 52 45 41 44 20 20 20 20 43 4F 4D 00 00 00 02 .READ    COM....
11F0 09 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-df450

?
-g450
```

søek

find spor 1

```
*0139
-g440
```

kontroler at spor 1 er fundet

```
*0139
-d200

0200 00 01 02 03 04 05 06 07 08 09 52 45 41 44 49 44 ..........READID
0210 02 4A 00 FF FF FF FF FF FF FF FF 43 4F 4D 44 53 .J.........COMDS
0220 FF FF FF FF FF FF FF FF FF FF FF 45 58 43 55 54 ...........EXCUT
0230 07 00 00 00 01 00 09 02 FF FF FF 52 53 55 4C 54 ...........RSULT
0240 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0250 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0260 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0270 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0280 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0290 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
02A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
02B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
-vi er på spor 1
```

ok

```
?
-g480
```

skriv data i directory på spor 1

```
*0139
-go
```

vend tilbage til operativsystem

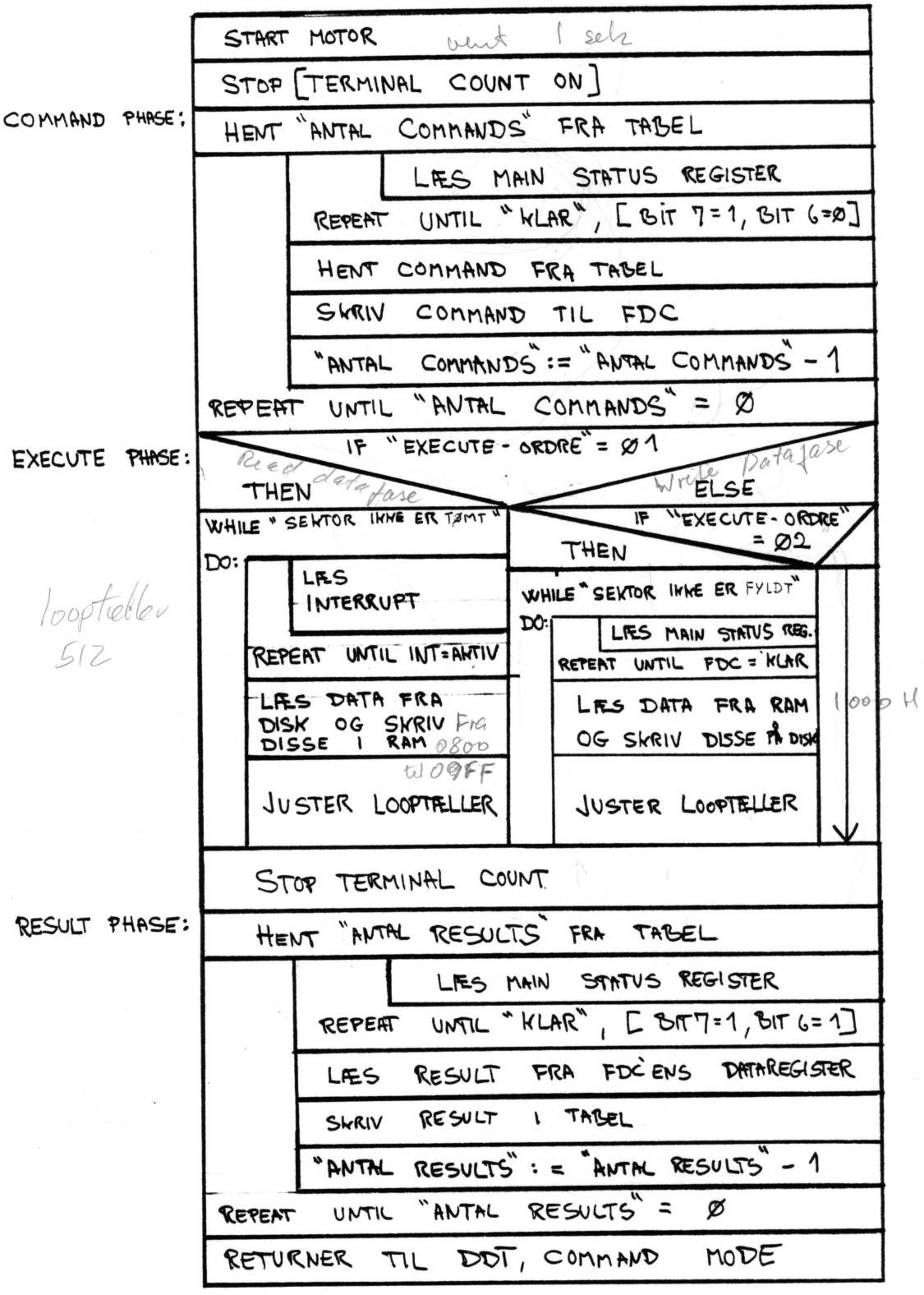
```
kontrol
```

KONTROL?

A>dir

```
A: LOLIGHT  COM : HILIGHT  COM : EDITOR   COM : DDT      COM
A: PRTEKS   COM : READKEY  COM : TTY      COM : TRANS    COM
A: FUNKTION COM : TEKST    COM : IEC      COM : INT      COM
A: PRT      COM : ZSID     COM : DISKREAD EKS : READ     COM
A: IHP      COM : IHP          : D
A>
```

lolight en tillbaga i dirchtorp


START MOTOR vent 1 sek
STOP [TERMINAL COUNT ON]
COMMAND PHASE:
HENT "ANTAL COMMANDS" FRA TABEL
LÆS MAIN STATUS REGISTER
REPEAT UNTIL "KLAR", [BIT 7=1, BIT 6=Ø]
HENT COMMAND FRA TABEL
SKRIV COMMAND TIL FDC
"ANTAL COMMANDS" := "ANTAL COMMANDS" - 1
REPEAT UNTIL "ANTAL COMMANDS" = Ø
EXECUTE PHASE:
IF "EXECUTE-ORDRE" = Ø1
Read datafase
THEN
Write Datafase
ELSE
WHILE "SEKTOR IKKE ER TØMT"
DO:
LÆS INTERRUPT
REPEAT UNTIL INT=AKTIV
LÆS DATA FRA DISK OG SKRIV DISSE I RAM
Fra 0800
til 09FF
JUSTER LOOPTELLER
IF "EXECUTE-ORDRE" = Ø2
THEN
WHILE "SEKTOR IKKE ER FYLDT"
DO:
LÆS MAIN STATUS REG.
REPEAT UNTIL FDC = KLAR
LÆS DATA FRA RAM OG SKRIV DISSE PÅ DISK
1000 H
JUSTER LOOPTELLER
looptæller 512
STOP TERMINAL COUNT
RESULT PHASE:
HENT "ANTAL RESULTS" FRA TABEL
LÆS MAIN STATUS REGISTER
REPEAT UNTIL "KLAR", [BIT 7=1, BIT 6=1]
LÆS RESULT FRA FDC'ENS DATAREGISTER
SKRIV RESULT I TABEL
"ANTAL RESULTS" := "ANTAL RESULTS" - 1
REPEAT UNTIL "ANTAL RESULTS" = Ø
RETURNER TIL DDT, COMMAND MODE

4) Foretag følgende ændring i DISKREAD.EKS, så et spor læses:

Instruktionen ADI 02H på adr.0141H ændres til ADI 14H.

Eksekver instruktionen READ DATA fra adr. 0470H

Beskriv de data, der nu er placeret fra adr. 0800H og fremefter (Der skal ledes efter en forskel!).

02 læses en sektor

14 læses alle 10 sektorer

Angiv de værdier, som READ DATA nu afleverer i resultat-fasen:

Cylinder number: 02

Head adress: 00

Sector number: 01

Bytes / sector: 02

4FF

5) Sæt SEEK op, så side 0, spor 79 (dec) læses og eksekver denne instruktion.

Eksekver READ ID, og beskriv indholdet af status-register 0 - 2 (result phase):

0 1 2

00 00 00 | 4F | 00 06 02

Sæt nu SEEK op, så side 0, spor 80 (dec) læses og eksekver denne instruktion.

50

Eksekver READ ID, og beskriv indholdet af status-register 0 - 2 (result phase):

00 01 00 50 00 01 02

6) Skriv dit navn på disketten´s side 0, spor 40H, sektor 3. Kontroller v.h.a. READ DATA at det er skrevet korrekt. (Se modulet WRDATA i dokumentationen for DISKREAD.EKS).

```
D0200,023F
0200 00 01 02 03 04 05 06 07 08 09 52 45 41 44 49 44 ..........READID
0210 FF FF FF FF FF FF FF FF FF FF FF 43 4F 4D 44 53 ...........COMDS
0220 FF FF FF FF FF FF FF FF FF FF FF 45 58 43 55 54 ...........EXCUT
0230 FF FF FF FF FF FF FF FF FF FF FF 52 53 55 4C 54 ...........RSULT
D0300,048F
0300 00 01 02 03 04 05 06 07 08 09 52 45 41 44 49 44 ..........READID
0310 02 4A 00 FF FF FF FF FF FF FF FF 43 4F 4D 44 53 .J..........COMDS
0320 FF FF FF FF FF FF FF FF FF FF FF 45 58 43 55 54 ...........EXCUT
0330 07 FF FF FF FF FF FF FF FF FF FF 52 53 55 4C 54 ...........RSULT
0340 00 01 02 03 04 05 06 07 08 09 53 45 45 4B 2E 2E ...........SEEK..
0350 03 0F 00 01 FF FF FF FF FF FF FF 43 4F 4D 44 53 ...........COMDS
0360 FF FF FF FF FF FF FF FF FF FF FF 45 58 43 55 54 ...........EXCUT
0370 00 FF FF FF FF FF FF FF FF FF FF 52 53 55 4C 54 ...........RSULT
0380 00 01 02 03 04 05 06 07 08 09 52 44 54 52 43 4B ..........RDTRCK
0390 09 42 00 01 00 01 02 0A 07 FF FF 43 4F 4D 44 53 .B.........COMDS
03A0 01 FF FF FF FF FF FF FF FF FF FF 45 58 43 55 54 ...........EXCUT
03B0 07 FF FF FF FF FF FF FF FF FF FF 52 53 55 4C 54 ...........RSULT
03C0 00 01 02 03 04 05 06 07 08 09 52 44 44 41 54 41 ..........RDDATA
03D0 FF FF FF FF FF FF FF FF FF FF FF 43 4F 4D 44 53 ...........COMDS
03E0 FF FF FF FF FF FF FF FF FF FF FF 45 58 43 55 54 ...........EXCUT
03F0 FF FF FF FF FF FF FF FF FF FF FF 52 53 55 4C 54 ...........RSULT
0400 00 01 02 03 04 05 06 07 08 09 57 52 44 41 54 41 ..........WRDATA
0410 FF FF FF FF FF FF FF FF FF FF FF 43 4F 4D 44 53 ...........COMDS
0420 FF FF FF FF FF FF FF FF FF FF FF 45 58 43 55 54 ...........EXCUT
0430 FF FF FF FF FF FF FF FF FF FF FF 52 53 55 4C 54 ...........RSULT
0440 21 00 03 C3 C0 02 4A 4D 50 00 52 45 41 44 49 44 !.....JMP.READID
0450 21 40 03 C3 C0 02 4A 4D 50 00 53 45 45 4B 2E 2E !§....JMP.SEEK..
0460 21 80 03 C3 C0 02 4A 4D 50 00 52 44 54 52 43 4B !.....JMP.RDTRCK
0470 21 C0 03 C3 C0 02 4A 4D 50 00 52 44 44 41 54 41 !.....JMP.RDDATA
0480 21 00 04 C3 C0 02 4A 4D 50 00 57 52 44 41 54 41 !.....JMP.WRDATA
```

```
L02C3,02D8
 02C3  LXI  D,0200
 02C6  MVI  B,40
 02C8  MOV  A,M
 02C9  XCHG
 02CA  MOV  M,A
 02CB  XCHG
 02CC  DCR  B
 02CD  MOV  A,B
 02CE  CPI  00
 02D0  JZ   0100
 02D3  INX  H
 02D4  INX  D
 02D5  JMP  02C8
 02D8  RST  07
 02D9
```

Måleøvelse: Floppy disk controller.

Formål: Formålet med øvelsen er at måle på det tidsmæssige forløb når en floppy disk controller henter data fra et floppy disk drive.

Udstyr:
Comet 3400.
Diskette indeholdende DISKREAD.EKS.
Philips LSA.
1 stk. 40 pin IC - clip.

Hjælpemidler:
Brugervejledning til Philips LSA.
Dokumentation over DISKREAD.EKS.

Gennemførelse: Forbind LSA'ens pod's efter brugervejledningen til et Z80 set up.

Aktiver Z80 disassembler i LSA'en.

Sæt CONFIG til: 35 channel states.

Sæt TRIG til:
Find WORD 1
Find WORD 2
END.

Sæt DATA til:
Data sel. off.
All states captured.
Counter on.
Start after WORD 1.
Finish at WORD 2.

1) Sæt Word 1 til 0148H (det tidspunkt, hvor 1. data ønskes læst) og Word 2 til 0800H (det tidspunkt, hvor 1. data fra floppy-disk'en skrives i RAM'en).

Foretag en tidsmåling med LSA'en.

Hvor lang tid måles ? 216,5m sek

Beskriv hvad der sker i dette tidsrum: _____

2) Sæt WORD 1 til 0800H og WORD 2 til 0801H.
Foretag en tidsmåling med LSA'en.

Hvor lang tid måles ? 29 µ sek

Beskriv hvad der sker i dette tidsrum:

controler afleveres en enkelt byte
tel CPU
data fra desk tel CPU

3) Sæt WORD 1 til 0800H og WORD 2 til 09FFH.
Foretag en tidsmåling med LSA'en.

Hvor lang tid måles ? 16,4 m sek

Beskriv hvad der sker i dette tidsrum:

data for et helt spor

4) Mål hvor lang tid tager det at læse et spor:

2,2 sek

01FA

5 The fourth bit of the PPE command is the sense bit for the parallel poll response.
The first three bits specify the parallel poll response (PPR) message to be sent when a parallel poll is executed.

6 Bits 1 to 4 of the PPD command carry no particular message; these bits should all be sent as zeroes, but need not be decoded by the receiving device.

' CAL MESSAGES

• Received by interface functions

end	end (of byte string)	(via T, TE)
gts	go to standby	(C)
ist	individual status	(PP)
lon	listen only	(L, LE)
lpe	local poll enable	(PP)
ltn	listen	(L, LE)
lun	local unlisten	(L, LE)
nba	new byte available	(SH)
pon	power on	
rdy	ready for next message	(AH)
rpp	request parallel poll	(C)
rsc	request system control	(SC)
rsv	request service	(SR)
rtl	return to local	(RL)
sic	send interface clear	(SC)
sre	send remote enable	(SC)
tca	take control asynchronously	(C)
tcs	take control synchronously	(AH, C)
ton	talk only	(T, TE)

• Sent by interface functions (not defined by IEC)

clr	clear device	(SH)
dcd	don't change data	(AH)
dvd	data valid device	(AH)
loc	local	(RL)
lsb	last byte	(via L, LE)
srq	service requested	(C)
trg	trigger device	(DT)

Appendix B

ISO 7-bit Code representation of multiline interface messages

	b_7	0	PCG		0	PCG		0	PCG		0	PCG		1	PCG		1	PCG		1	SCG		1	SCG	
	b_6	0	ACG		0	UCG		1	LAG		1	LAG		0	TAG		0	TAG		1			1		
	b_5	0	MSG	dec	1	MSG	dec	0	MSG	dec	1	MSE	dec	0	MSG	dec	1	MSG	dec	0	MSG	dec	1	MSG	dec
b_4 b_3 b_2 b_1	C/R	0			1			2			3			4			5			6			7		
0 0 0 0	0	NUL		0	DLE		16	SP	LAD	32	0	LAD	48	@	TAD	64	P	TAD	80	'		96	p		112
1 0 0 0	1	SOH	GTL	1	DC1	LLO	17	!	LAD	33	1	LAD	49	A	TAD	65	Q	TAD	81	a		97	q		113
0 0 1 0	2	STX		2	DC2		18	"	LAD	34	2	LAD	50	B	TAD	66	R	TAD	82	b		98	r		114
0 0 1 1	3	ETX		3	DC3		19	#	LAD	35	3	LAD	51	C	TAD	67	S	TAD	83	c		99	s		115
0 1 0 0	4	EOT	SDC	4	DC4	DCL	20	$	LAD	36	4	LAD	52	D	TAD	68	T	TAD	84	d		100	t		116
0 1 0 1	5	ENQ	PPC	5	NAK	PPU	21	%	LAD	37	5	LAD	53	E	TAD	69	U	TAD	85	e		101	u		117
0 1 1 0	6	ACK		6	SYN		22	&	LAD	38	6	LAD	54	F	TAD	70	V	TAD	86	f		102	v		118
0 1 1 1	7	BEL		7	ETB		23	'	LAD	39	7	LAD	55	G	TAD	71	W	TAD	87	g		103	w		119
1 0 0 0	8	BS	GET	8	CAN	SPE	24	(	LAD	40	8	LAD	56	H	TAD	72	X	TAD	88	h		104	x		120
1 0 0 1	9	HT	TCT	9	EM	SPD	25	)	LAD	41	9	LAD	57	I	TAD	73	Y	TAD	89	i		105	y		121
1 0 1 0	10	LF		10	SUB		26	*	LAD	42	:	LAD	58	J	TAD	74	Z	TAD	90	j		106	z		122
1 0 1 1	11	VT		11	ESC		27	+	LAD	43	;	LAD	59	K	TAD	75	[	TAD	91	k		107	{		123
1 1 0 0	12	FF		12	FS		28	,	LAD	44	<	LAD	60	L	TAD	76	\	TAD	92	l		108	|		124
1 1 0 1	13	OR		13	GS		29	-	LAD	45	=	LAD	61	M	TAD	77	]	TAD	93	m		109	}		125
1 1 1 0	14	SO		14	RS		30	.	LAD	46	>	LAD	62	N	TAD	78	^	TAD	94	n		110	~		126
1 1 1 1	15	SI		15	US		31	/	LAD	47	?	UNL	63	O	TAD	79	_	UNT	95	o		111	DEL		127

Measuring defined by PCG code

Notes:

1 Requires secondary command

2 DIO1 = b_1... DIO7 = b_7.

PCG = Primary Command Group

SCG = Secondary Command Group

MSG = Interface message

ACG = Addressed Command Group

UCG = Universal Command Group

LAG = Listen Address Group

TAG = Talk Address Group

LAD = Listen Address

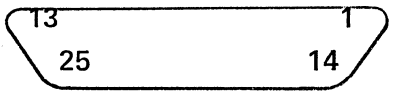
TAD = Talk Address

Appendix E Bus Connector Contact Assignments

a) MIL-C-24308 Connector; IEC-Bus

Contact	Signal line	Contact	Signal Line
1	DIO 1	14	DIO 5
2	DIO 2	15	DIO 6
3	DIO 3	16	DIO 7
4	DIO 4	17	DIO 8
5	REN	18	Gnd, (5)
6	EOI	19	Gnd, (6)
7	DAV	20	Gnd, (7)
8	NRFD	21	Gnd, (8)
9	NDAC	22	Gnd, (9)
10	IFC	23	Gnd, (10)
11	SRQ	24	Gnd, (11)
12	ATN	25	Gnd, (12)
13	Shield		

Mounting:



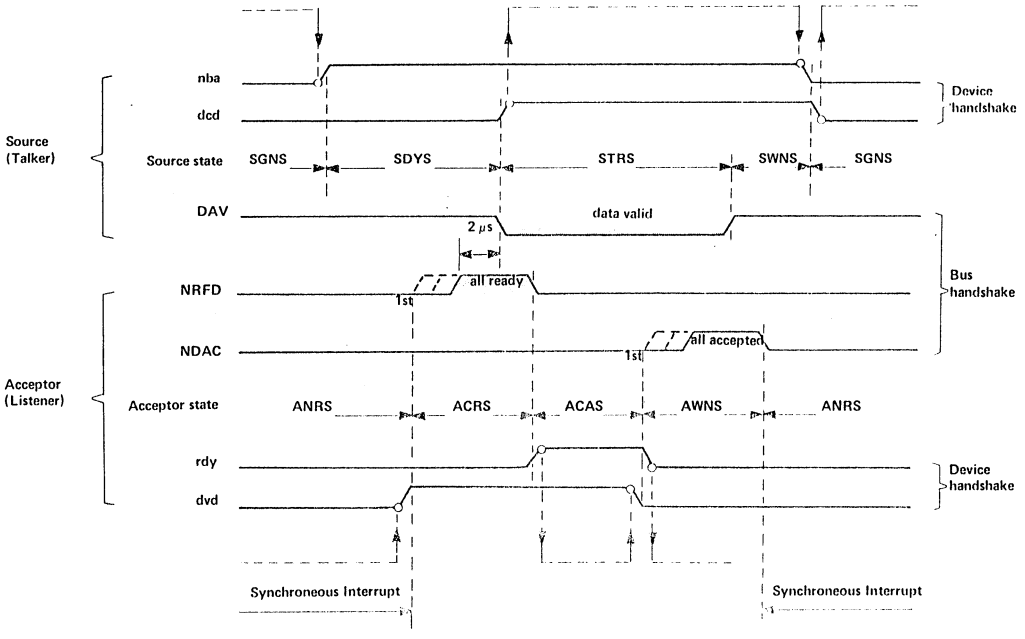
b) Micro-Ribbon Connector; IEEE Bus

Contact	Signal Line	Contact	Signal Line
1	DIO 1	13	DIO 5
2	DIO 2	14	DIO 6
3	DIO 3	15	DIO 7
4	DIO 4	16	DIO 8
5	EOI	17	REN
6	DAV	18	Gnd, (6)
7	NRFD	19	Gnd, (7)
8	NDAC	20	Gnd, (8)
9	IFC	21	Gnd, (9)
10	SRQ	22	Gnd, (10)
11	ATN	23	Gnd, (11)
12	Shield	24	Gnd, Logic

Mounting:

Note: Gnd (n) refers to the signal ground return of the referenced contact.

Appendix F Handshake between source and acceptor(s)



General Interface management				Data Bus	Message	Comments
REN	EOI	IFC	ATN	DIO 1-8		
•					REN	Enables all instruments for remote control
•		•			IFC	Clears all interfaces
•			•	"Q"	TAD	Talk address Computer
•			•	"6"	LAD	Listen address DVM
•				"D"	DAB	Programming instructions
				"R"	DAB	for DVM (D = DC voltage)
				"2"	DAB	(R2 = mV Range)
				"ETX"	EOS	End code (Record delimiter)
•			•	"?"	UNL	Unlisten bus command
•			•	"5"	LAD	Listen address DC Source
•				"A"	DAB	
•				"M"	DAB	
•				"V"	DAB	Data bytes for DC Source
•				"+"	DAB	with coded information
•				"0"	DAB	on voltage to be delivered
•				"1"	DAB	(DC voltage of 180 mV)
•				"8"	DAB	
•				"0"	DAB	
•				"ETX"	EOS	End code (Record delimiter)
•			•	"?"	UNL	Unlisten bus command
•			•	"6"	LAD	Listen address DVM
•				"E"	DAB	Start command for DVM
•			•	"?"	UNL	Unlisten bus command
•			•	"V"	TAD	Talk address DVM
•			•	"1"	LAD	Listen address Computer
•			•	"4"	LAD	Listen address Printer
•				"D"	DAB	
•				"C"	DAB	
•				"M"	DAB	
•				"V"	DAB	Data bytes (string)
•				" ¡ "	DAB	from DVM to computer
•				"0"	DAB	and printer with
•				"2"	DAB	information
•				"•"	DAB	about measured voltage
•				"3"	DAB	(DC voltage of +230 mV)
•				"0"	DAB	
•				"E"	DAB	
•				"+"	DAB	
•				"0"	DAB	
•	•			"2"	END^DAB	END message as Record delimiter
•			•	"?"	UNL	Unlisten bus command
•			•	"Q"	TAD	Talk address Computer
•			•	"4"	LAD	Listen address Printer
•				"CR"	DAB	Carriage Return
•				"LF"	DAB	Line Feed

Table 3.2. Sequence of events on the IEC bus during linearity check on DC amplifier (see fig. 2.2).

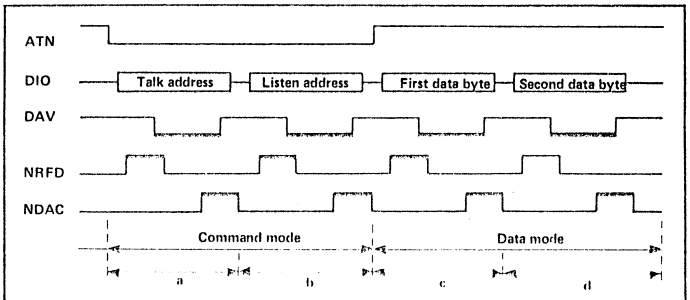


Fig. 3.1. Timing of a handshake procedure

Figure 3.1. shows a time diagram for the handshake lines, ATN and the data bus.
In this example a talker and a listener are addressed and two data bytes are sent from the talker to the listener.

a. ATN = "low" Talk address code on data bus
b. ATN = "low" Listen address code on data bus
c. ATN = "high" First data byte on data bus
d. ATN = "high" Second data byte on data bus

When designing an IEC bus system, one should make sure that no two instruments have the same address – although two listeners may be given the same address if they always may receive the same data. An instrument with both a talk and a listen function will have two addresses, one for each function. In this case, the 5 bits $b_5b_4b_3b_2b_1$ are often given the same values in the talk and the listen address, so that one only has to set the 5 address switches once.
Example:
Digital Voltmeter. Talk address "V" × 10 10110
Listen address "6" × 01 10110

Only one of the devices on the IEC bus can be addressed as talker at the time. Several devices can be addressed simultaneously as listener, since listeners can operate in parallel.
To avoid the possibility of two talkers in parallel, each time a talker is addressed all other talkers are automatically set to "untalk".

Talkers and listeners can be reset to the inactive (unaddressed) state together by the general interface management message IFC. This interface clear message is sent by the system controller.
To unaddress listeners only, we can use the bus command UNL (Unlisten). The code for this is placed on the data bus by the controller in the command mode.

Talk and Listen Addresses

Listen Addresses									Talk Addresses								
Bits								ISO	Bits								ISO
b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	Character	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1	Character
×	0	1	0	0	0	0	0	SP	×	1	0	0	0	0	0	0	@
×	0	1	0	0	0	0	1	!	×	1	0	0	0	0	0	1	A
×	0	1	0	0	0	1	0	"	×	1	0	0	0	0	1	0	B
×	0	1	0	0	0	1	1	#	×	1	0	0	0	0	1	1	C
×	0	1	0	0	1	0	0	$	×	1	0	0	0	1	0	0	D
×	0	1	0	0	1	0	1	%	×	1	0	0	0	1	0	1	E
×	0	1	0	0	1	1	0	&	×	1	0	0	0	1	1	0	F
×	0	1	0	0	1	1	1	'	×	1	0	0	0	1	1	1	G
×	0	1	0	1	0	0	0	(	×	1	0	0	1	0	0	0	H
×	0	1	0	1	0	0	1	)	×	1	0	0	1	0	0	1	I
×	0	1	0	1	0	1	0	*	×	1	0	0	1	0	1	0	J
×	0	1	0	1	0	1	1	+	×	1	0	0	1	0	1	1	K
×	0	1	0	1	1	0	0	,	×	1	0	0	1	1	0	0	L
×	0	1	0	1	1	0	1	–	×	1	0	0	1	1	0	1	M
×	0	1	0	1	1	1	0	.	×	1	0	0	1	1	1	0	N
×	0	1	0	1	1	1	1	/	×	1	0	0	1	1	1	1	O
×	0	1	1	0	0	0	0	0	×	1	0	1	0	0	0	0	P
×	0	1	1	0	0	0	1	1	×	1	0	1	0	0	0	1	Q
×	0	1	1	0	0	1	0	2	×	1	0	1	0	0	1	0	R
×	0	1	1	0	0	1	1	3	×	1	0	1	0	0	1	1	S
×	0	1	1	0	1	0	0	4	×	1	0	1	0	1	0	0	T
×	0	1	1	0	1	0	1	5	×	1	0	1	0	1	0	1	U
×	0	1	1	0	1	1	0	6	×	1	0	1	0	1	1	0	V
×	0	1	1	0	1	1	1	7	×	1	0	1	0	1	1	1	W
×	0	1	1	1	0	0	0	8	×	1	0	1	1	0	0	0	X
×	0	1	1	1	0	0	1	9	×	1	0	1	1	0	0	1	Y
×	0	1	1	1	0	1	0	:	×	1	0	1	1	0	1	0	Z
×	0	1	1	1	0	1	1	;	×	1	0	1	1	0	1	1	[
×	0	1	1	1	1	0	0	<	×	1	0	1	1	1	0	0	\
×	0	1	1	1	1	0	1	=	×	1	0	1	1	1	0	1	]
×	0	1	1	1	1	1	0	>	×	1	0	1	1	1	1	0	∩

Table 3.1

The sixth and seventh bits determine whether the address in question is a talk or a listen address:

b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
x	1	0	•	•	•	•	•
x	0	1	•	•	•	•	•

- Talk address
- Listen address

The five remaining bits of the address determine the device involved, e.g.:

b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
x	0	1	1	0	1	0	0
x	1	0	0	0	1	0	1

- Listen address "4"
- Talk address "E"

The address of a device can generally be changed with the aid of five address switches which select the appropriate five bits of the address. If a controller wants to address a particular instrument, it places the address code for that device on the 8 data lines, and at the same time makes ATN = "true".
All devices on the IEC bus compare the address code presented with their own address. For this purpose, each device must have an acceptor handshake function which is active all the time during the command mode.
The device address and the address on the bus are compared by the "message decoding" function – part of the IEC interface of the device.
Only the device that finds its address equal to that on the bus assumes the function of talker or listener.

MÅLEØVELSE INSTRUMENT INTERFACE:

Formål: At undersøge signalforløbet på IEC-bussen når der sendes hhv. modtages data fra en controller til et instrument og herved underbygge teorien omkring IEC-interface. IEEE 488 HP-IB

Udstyr:
Comet 3400 incl. IEC-instrument interface (MPS 29)
Diskette med IEC.COM program
Logikanalysator (HP eller Philips)
Storage oscilloskop (Philips PM 3310)
2 Stk. 20 pin IC-clips

Hjælpemidler:
Diagram til MPS 29 IEC-interfacekort
Microprocessor Interfacing Techniques (SYBEX)
Philips IEC-625 interface bog
NEC uPD 7210 product description

Gennemførelse:

Sendemode: Tilslut et storageoscilloskop Philips PM 3310 til IEC-BUS`en på Comet 3400

2) Tilslut en LSA til udgangsbufferne på IEC-kortet:

IC 2 ben 2-9 = Pod 0 (neg.) "DATA"
IC 3 ben 2-9 = Pod 1 (pos.) "CONTR"

Clock på: DAV (neg. flanke)

Trig på: DAV = 0
NRFD = 1
NDAC = 0

Indspil TRIGGER ORD

3) Kald IEC-programmet fra disketten

4) Indtast oscilloskopets bus-adr. (08)

5) Noter hvilke data der skal overføres på IEC-bussen for at indstille oscilloskopet til at måle cal.-signalet (3Vpp / 2,5kHz) med hhv.:

Probe 1:10, Y x 5, A = ON, B = OFF,
Display ACCU, NO Delay.

1	2	3	4	5	6	7	8
EX	0	M	J	C	4	Σ	B	C	4
9	E	3	8	13	14	15	E	17	18
19	2E	21	22	23	24	25	62	^C	ret

Sendemode fortsat:

6) Sæt LSA til at optage og send "forpladen" til oscilloskopet, start LSA'en inden der trykkes 'S' for Send.

-MÅLING OK ? JA -GÅ TIL PKT.7, NEJ -GENTAG FRA PKT.5

7) Noter hvilke karakterer der sendes på IEC-BUS`en:

? ! (! @ ! ESC ! O ! M ! T ! C ! 4 ! 2 !
B ! C ! 4 ! 2 ! F ! 3 ! 8 ! 4 ! 0 ! E !
E ! 3 ! 0 ! 1 ! 0 ! 0 ! 0 ! 0 ! 0 ! 0 ! ETX

8) Stemmer data overens med det afsendte ? ja

9) Hvad er det der overføres før og efter "forplade-data" ?

? (@ før til 7 0 ATN command mode
ETX efter

10) Beskriv betydningen af disse:

? unlisten
(listen adres
@ talk address
ETX eoi limiter

11) Hvornår er systemet i command-mode hhv. datamode ?

Receivemode: Modtag en "forplade" fra oscilloskopet med flg. indstilling (indstilles manuelt på forplade):

KANAL A:
1V/div.
probe 1:1
kalibreret
kanal ON (normal)
AC koblet

KANAL B:
10V/div.
probe 1:1
kalibreret
kanal ON (normal)
DC koblet

TIME BASE:
1mSek/div.
RECURRENT

DISPLAY:
ACCU ONLY
Y x 1
X = t
NO DOTS
NO PLOT

TRIG.DELAY:
+15 div.

2) Kald IEC-programmet og mål med LSA hvad der overføres på IEC-BUS`en. Start LSA før tryk på 'R' for Receive.

3) Indtast startordre til oscilloskop:
(Delimiter = <US> = <^ INS>).

4) Noter hvad der overføres:

____!____!____!____!____!____!____!____!____!____!

____!____!____!____!____!____!____!____!____!____!

____!____!____!____!____!____!____!____!____!____!

____!____!____!____!____!____!____!____!____!____!

____!____!____!____!____!____!____!____!____!____!

____!____!____!____!____!____!____!____!____!____!

5) Stemmer det målte overens med indstillede ?_____

6) Hvad er det der overføres før "forpladedata" ?

_____

_____

_____

_____

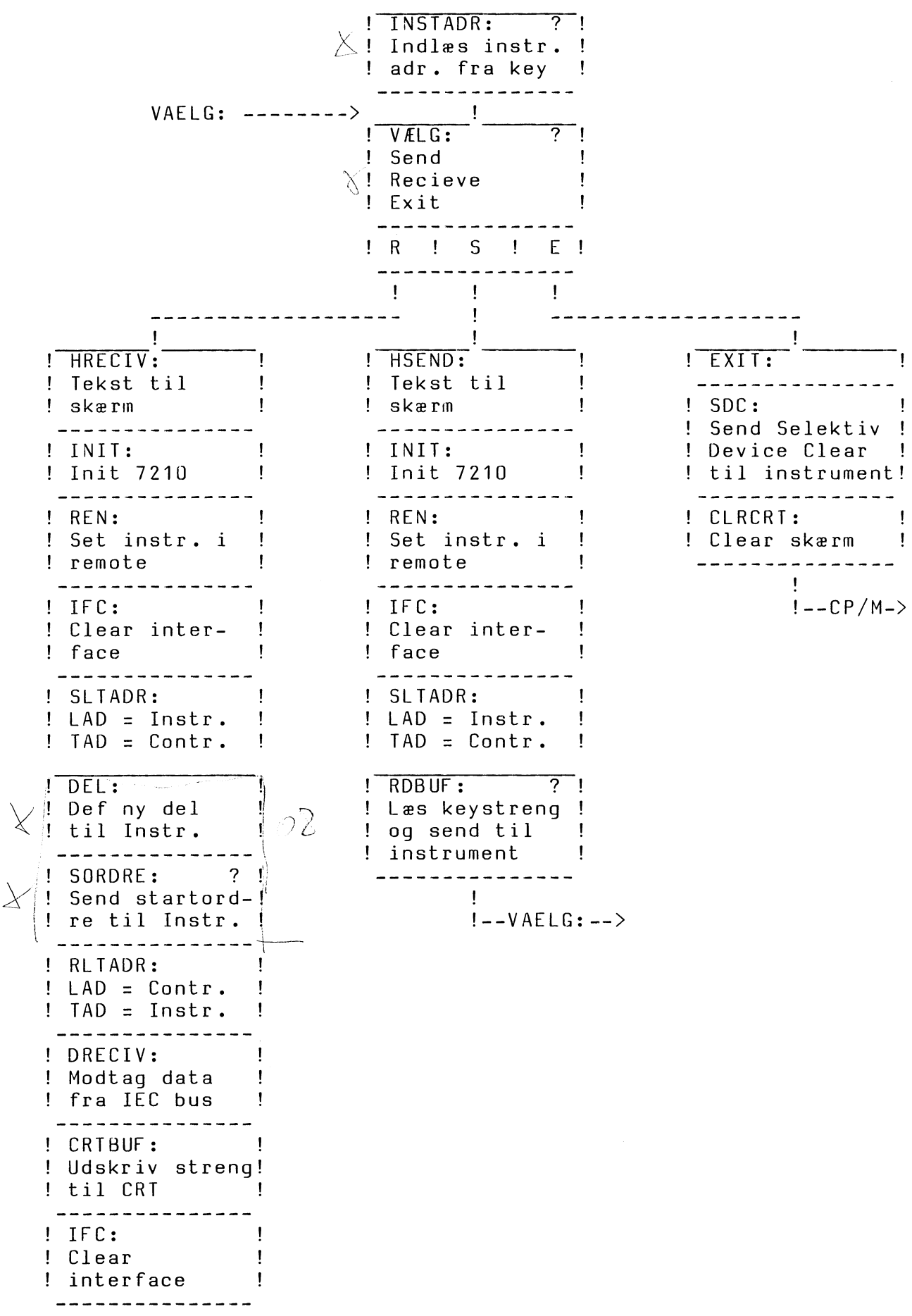
_____

_____

7) Hvilke karakterer udsendes i command-mode ?

_____

STRUKTUR AF IEC-PROGRAM:



!--VAELG:-->

ØVELSE INSTRUMENT INTERFACE:

1) En fordel ved IEC-bus systemet i sammenligning med andre systemer er:

A) Instrumenterne kan gøres billigere A __ ?

B) Relativ få buslinier B __ ?

C) Alle instrumenter med en IEC-interface kan tilsluttes systemet C __ ?

2) Transmissionen via IEC-bussen er:

A) Byte-seriel bit-parallel A __ ?

B) Byte-seriel bit-seriel B __ ?

C) Byte-parallel bit-parallel C __ ?

3) Hvad bliver transmitteret via databussen i DATA-mode ?

A) Listen adr. A __ ?

B) Talk adr. B __ ?

C) Måle-data C __ ?

4) Hvor mange apparater kan være aktive som "TALKER" på bussen ad gangen ?

A) 24 Stk. A __ ?

B) 1 Stk. B __ ?

C) 31 Stk. C __ ?

5) IFC (Interface Clear) og REN (Remote Enable) kan udsendes fra:

A) Alle apparater A __ ?

B) System controlleren B __ ?

C) Alle apparater med TALKER funktion C __ ?

INSTRUMENT INTERFACE fortsat:

6) Hvor bliver disse tre handshake signaler aktiveret fra: "DAV","NRFD","NDAC" ?

A) "DAV" af TALKER, "NRFD" og "NDAC" af LISTENER A __ ?

B) "DAV" og "NRFD" af TALKER, "NDAC" af LISTENER B __ ?

C) "DAV" og "NDAC" af LISTENER, "NRFD" af TALKER C __ ?

7) Hvilken logikform kører IEC-bussen med ?

A) Positiv logik A __ ?

B) Negativ logik B __ ?

C) Dataledningerne = Pos. logik og kontrolledningerne = Neg. logik C __ ?

8) Hvilke af følgende signaler kan aktiveres af et apparat som ikke er controller ?

A) ATN,SRQ,REN A __ ?

B) EOI,SRQ B __ ?

C) Intet af ovenstående C __ ?

9) Hvor mange bit anvendes til at adressere apparaterne med ?

A) 8 bit A __ ?

B) 7 bit B __ ?

C) 5 bit C __ ?

10) Hvilke af følgende karakterer kan anvendes som TALK ADDRESS ?

A) + A __ ?

B) = B __ ?

C) § C __ ?

MÅLEØVELSE PRINTER:

Formål: At undersøge nogle af de styresignaler som forekommer i en printer og styre printeren med forskellige kontrolkoder for herved at underbygge teorien omkring printer interface.

Udstyr:
Comet 3400 incl. I/O-kort (MPS 10)
Diskette med TTY.COM og PRT.COM program
Printer Microline u84
Logik/Timing-analysator (Philips / HP)
1 Stk.40 pin IC clips
2 Stk.14 pin IC clips
1 Stk. omsætterkabel MOLEX / CANON

Hjælpemidler:
Manual til Microline 84 printer
Diagram til Microline 84 printer
Brugervejledning til I/O-kort (MPS 10)

Gennemførelse:

Måling på printer:

1) Demonter låget på printeren

2) Tilslut printeren til paralleludgangen på Comet`en

3) Tilslut en LTA (Philips/HP) til printeren:

PHILIPS SETUP:

POD T:

bit 0 =	Q11 ben 30	Stepmotor Strobe (SP)
bit 1 =	Q11 ben 31	Stepmotor vikling 4
bit 2 =	Q11 ben 32	Stepmotor vikling 3
bit 3 =	Q11 ben 33	Stepmotor vikling 2
bit 4 =	Q11 ben 34	Stepmotor vikling 1
bit 5 =	Q06 ben 09	Printnål nr. 3
bit 6 =	Q08 ben 04	Printnål Strobe
GND. =	Q11 ben 20	Stel (T)

CONFIG: 8 Channel timing

Nulstil bit 7

TRIG:

TRIG MODE = WORD
TRIG AT OCCUR AT WORD X X10XXXXX > OR = 100 nSek.
(første gang nålen slår ind)
TIMEDELAY 0 nSek
FINAL DELAY 200 TRANSITIONS

HP SETUP:

POD 0:

bit 0 = Q11 ben 30	Stepmotor Strobe (SP)	
bit 1 = Q11 ben 31	Stepmotor vikling 4	1
bit 2 = Q11 ben 32	Stepmotor vikling 3	2
bit 3 = Q11 ben 33	Stepmotor vikling 2	3
bit 4 = Q11 ben 34	Stepmotor vikling 1	4
bit 5 = Q06 ben 09	Printnål nr. 3	
bit 6 = Q08 ben 04	Printnål Strobe	
GND. = Q11 ben 20	Stel	

SYSTEM: 16 CHANAL TIMING

FORMAT: Nulstil de bit som ikke benyttes (15-7)

TRACE: SAMPLE PERIOD 10 uSek

PATTERN 1ØXXXXX

4) Kald programmet TTY fra disketten og indtast en række I`er (skærm bredde)

5) Optag en måling ved at starte LTA`en og taste <CR>

6) Hvordan ligger strobe-pulserne til stepmotor og nål i forhold til hinanden ?

forskudt stepmotor ca 200 µsek
for strobe til print a31

7) Hvor mange gange slår nålen ind for hvert tegn ?

1 gang vi er på lodret streg i I

8) Hvor mange dot`s (vandret) vil et tegn max. kunne bestå af ? 11

8) Efter hvilken sekvens bliver de 4 viklinger i stepmotoren aktiveret ?

Bit 4321, 4321 Vikling 1234

9) Foretag en ny indspilning, men med tegn i dobbelt bredde: ^<INS>

10) Hvor mange gange slår nålen ind for hvert tegn ?

2 gange 12 studser

Styring af printer:

1) Tilslut printeren til I/O kortet på Comet en vha adapter kablet.

2) Kald PRT.COM programmet sammen med ZSID.

Tekst opg:

1) Skriv ASCII koderne til f.eks dit fornavn i programmet fra adr. 200 med hhv.: 17 CPI, 12 CPI, 10 CPI og dobbelt brede karakterer

Afslut hver "mode" med: <CR> <LF>

Afslut sidste "mode" med <CR> <LF> <LF> <LF> <ETX>

(ASCII-koder kan indtastes i ZSID vha: "xxxx")

Noter koderne:

17 CPI:	1C									
12 CPI:	1D									
10 CPI:	1E									
DUB.WD:	1F									

Afprøv programmet

2) Skriv ASCII koderne til en række tegn (16 stk.), hvor bit 8 er high (alternativ karakterset) og afslut strengen med <CR> <LF> <LF> <LF> <ETX>

OD OA OA OA (03)

Noter koderne:

A0	A1	A2	A3	A4	A5	A6	A7	A8		
A9	AA	AB	AC	AD	AE	AF				

Afprøv programmet

Format opg: 1) Sæt printeren op til følgende, ved at kode ASCII-karaktererne ind fra adr. 200:

Printer til A4 format = 12" 1B 47 (ESC G) 32 34

Set TOP OF FORM 1B 35 (ESC 5)

8 LINES/inch. 1B 38 (ESC 8)

10 CHARAKTERS/inch. 1E (RS)

HORIZONTAL TAB. til 036 tegn 1B 09 24 (ESC HT)
(HUSK CR) 30 33 36 0D

2) Efterfølg set-up`et af denne tekst:

<CR> <HT> SIDE <SP> <SP> 1 <CR>

SKIP 46 linier 1B 0B 34 36 (ESC VT)

0 A <HT> H E J <SP> ! <LF>
1 B <LF>
2 C <LF>
3 D <LF>
4 E <LF>
5 F <LF>
6 G <LF>
7 H <LF> <FF> <ETX>

Noter koderne:

1B	47	18 (32 34)	1B	35	1B	38	1E	1B	09
24	0D	0D	09	53	49	44	45	20	20
31	0D	1B	0B	34	36	41	09	48	45
4A	20	21	0A	42	0A	43	0A	44	0A
45	0A	46	0A	47	0A	4A	0A	0C	03
					48	0A			

(30 33 36)

3) Start programmet og iagttag printhoveds vandring over papiret

4) I hvilken retning printes: "SIDE 1" ?

fremad

5) I hvilken retning printes: "A HEJ !" ?

tilbage

Grafik opg: 1) Set printeren op til følgende, ved at kode ASCII-karaktererne ind fra adr. 200:

Line space = 15/144 inch. (ESC % 9)

Reserver 16 pladser til grafik dot (ESC % 2)

Efterfølg med koder for en firkant med
16 vandrette punkter og 3 tegns højde

NB: efter hver linie skal der reserveres
16 nye pladser (ESC % 2)

Afslut med <ETX>

Noter koderne:

2) Afprøv programmet

```
1
2
3
4
5                                   .COMMENT^
6
7                                                   READKEY
8                                   ------------------------------------------------
9                                   !                                              !
10                                  ! Programmet henter ASCII karakterer fra keyboard, og !
11                                  ! placerer disse fra adresse 1000H og fremefter.  !
12                                  !                                              !
13                                  ! Indtastningen standses ved (ESC) eller (CNTRL C)  !
14                                  !                                              !
15                                  ------------------------------------------------
16
17                                  ^
18
19                                          .PHASE  100H
20
21  1000                                    TEKST   EQU     1000H   For skærmbrug video ram
22  00F4                                    PORT    EQU     0F4H        adresse EFFF
23  0003                                    CNTRLC  EQU     03H
24  001B                                    ESC     EQU     1BH
25  00FF                                    NONASC  EQU     0FFH
26
27  0100    21 1000             RDKEY:  LXI     H,TEKST ;Peg på startadresse for ASCII - kar.
28  0103    DB F4               LOOP:   IN      PORT    ;Hent data fra keyboard.
29  0105    FE FF                       CPI     NONASC  ;Hvis data () ASCII, så hent næste data.
30  0107    CA 0116                     JZ      IGEN
31  010A    FE 03                       CPI     CNTRLC  ;Hvis ^C er tastet, så hop ud af program.
32  010C    CA 0119                     JZ      ENDRD
33  010F    FE 1B                       CPI     ESC     ;Hvis ESC er tastet, så hop ud af program.
34  0111    CA 0119                     JZ      ENDRD
35  0114    77                          MOV     M,A     ;Skriv ASCII på den tildelte adresse.
36  0115    23                          INX     H       ;klargør næste adresse, og
37  0116    C3 0103             IGEN:   JMP     LOOP    ;check for ny indtastning.
38  0119    FF                  ENDRD:  RST     07H     ;Tilbage i DDT, command mode.
39
40                                      .DEPHASE
41
42                                      END
```

```
                                .Z80
                                .COMMENT^
                                                INT.COM
                                 ----------------------------------------
                                !                                        !
                                ! Programmet sender en tekst-streng til printer.!
                                ! Tekst-strengen skal ligge fra adr. 200 --)    !
                                ! og afsluttes med (ETX).                       !
                                !                                        !
                                 ----------------------------------------
                                ^

00FF                            INTDISW EQU  0FFH ;Interrupt Disable Word *:
00FF                            VECTORI EQU  0FFH ;Vektor til I-reg       *:
00FF                            VECTORP EQU  0FFH ;Vektor til port        *:
00FF                            MOCTRW0 EQU  0FFH ;Mode Control Word 0    *:PORT B
00FF                            MOCTRW3 EQU  0FFH ;Mode Control Word 3    *:PORT A
00FF                            IOCTRW  EQU  0FFH ;I/O Reg. Contr. Word   *:PORT A
00FF                            INTCTRW EQU  0FFH ;Interrupt Contr.Word   *:Activ low
00FF                            MSKCTRW EQU  0FFH ;Mask Control Word      *:Aktiv for BUSY/ACKNOWL.
0160                            VECTOR  EQU  160H ;Interruptvectoradresse
0000                            PORTA   EQU   00H ;portadresse A
0001                            PORTB   EQU   01H ;portadresse B
0002                            CRA     EQU   02H ;Controlreg. A
0003                            CRB     EQU   03H ;Controlreg. B
0170                            EOT     EQU  170H ;End of text marker
0180                            COUNT   EQU  180H ;Karakteradressetæller
0200                            TABEL   EQU  200H ;Startadresse for printerkoder
01A0                            INT     EQU  1A0H ;Startadresse for Int.rutine

0000'                                   ASEG
                                        ORG     100H
0100    F3                              DI
0101    3E FF                           LD A,INTDISW    ;*****                  *****
0103    D3 02                           OUT (CRA),A     ;Int. fra port A disablet

                                ;;;;;;;;;;;;;;; Her begynder initialisering af CPU ;;;;;;;;;;;;;;;

0105    ED 5E                           IM 2            ;CPU int.mode 2
0107    3E FF                           LD A,VECTORI    ;*****                  *****
0109    ED 47                           LD I,A
010B    21 01A0                         LD HL,INT
010E    22 0160                         LD (VECTOR),HL  ;Startadr. til vector
0111    3E 00                           LD A,00H
0113    32 0170                         LD (EOT),A      ;EOT marker = 0
0116    21 0200                         LD HL,TABEL
0119    22 0180                         LD (COUNT),HL   ;Adr. tæller = TABEL

                                ;;;;;;;;;;;;;;; Her begynder initialisering af porte ;;;;;;;;;;;;;;;

011C    3E FF                           LD A,VECTORP    ;*****                  *****
011E    D3 02                           OUT (CRA),A
0120    3E FF                           LD A,MOCTRW0    ;*****                  *****
```

```
 57   0122   D3 03                OUT (CRB),A     ;port B = Mode 0
 58   0124   3E FF                LD A,MODCTRW3   ;*****                    *****
 59   0126   D3 02                OUT (CRA),A     ;port A = Mode 3
 60   0128   3E FF                LD A,IOCTRW     ;*****                    *****
 61   012A   D3 02                OUT (CRA),A     ;port A  = in + out
 62   012C   3E 80                LD A,80H        ;Strobe = high
 63   012E   D3 00                OUT (PORTA),A
 64   0130   3E FF                LD A,INTCTRW    ;*****                    *****
 65   0132   D3 02                OUT (CRA),A     ;Int. initialisering + Int.enable
 66   0134   3E FF                LD A,MSKCTRW    ;*****                    *****
 67   0136   D3 02                OUT (CRA),A     ;Interrupt maske
 68   0138   FB                   EI
 69   0139   3E 0D                LD A,0DH        ;Carriage return til printer og vent
 70   013B   CD 01B6              CALL PRINT      ;på interrupt
 71
 72
 73                        ;;;;;;;;;;;;;;;;;;;;;;;;; Hovedprogram ;;;;;;;;;;;;;;;;;;;;;;;;;
 74
 75   013E   FB          LOOP: EI              ;gøringentingogventpåinterrupt
 76   013F   00                NOP
 77   0140   00                NOP
 78   0141   00                NOP
 79   0142   00                NOP
 80   0143   00                NOP
 81   0144   00                NOP
 82   0145   00                NOP
 83   0146   00                NOP
 84   0147   00                NOP
 85   0148   00                NOP
 86   0149   F3                DI
 87
 88   014A   3A 0170            LD A,(EOT)      ;Undersøg om det var sidste karakter
 89   014D   FE 03              CP 03H
 90   014F   20 ED              JR NZ,LOOP      ;Nej gå til LOOP
 91   0151   3E FF              LD A,INTDISW    ;*****                    *****
 92   0153   D3 02              OUT (CRA),A     ;disable interrupt fra port
 93   0155   ED 46              IM 0            ;CPU til mode 0
 94   0157   FF                 RST 38H         ;Vend tilbage til ZSID'en
 95
 96
 97                        ;;;;;;;;;;;;;;;;;;;;;;;;; Interruptrutine ;;;;;;;;;;;;;;;;;;;;;;;;;
 98
 99                              ORG     INT
100   01A0   2A 0180              LD HL,(COUNT)
101   01A3   01 0170              LD BC,EOT
102   01A6   7E                   LD A,(HL)       ;hent kode
103   01A7   FE 03                CP 03H          ;er det den sidste?
104   01A9   28 07                JR Z,SLUT
105   01AB   CD 01B6              CALL PRINT      ;fyr koden af på printeren!!
106   01AE   23                   INC HL
107   01AF   22 0180              LD (COUNT),HL   ;sæt adr. tæller til næste værdi
108   01B2   02          SLUT:    LD (BC),A       ;læg EOT ud på adr.EOT
109   01B3   FB                   EI
110   01B4   ED 4D                RETI
111
112
```

```
113                             ;;;;;;;;;;;;;;;;;;;;;;;;; Print rutine ;;;;;;;;;;;;;;;;;;;;;;;;;
114
115     01B6    D3 01           PRINT:  OUT (01),A      ;koden til port B
116
117                             ;strobesløjfe:
118
119     01B8    3E 70                   LD A,70H
120     01BA    D3 00                   OUT (00),A      ;strobe = low
121     01BC    3C              OMIGEN: INC A
122     01BD    FE 80                   CP 80H
123     01BF    20 FB                   JR NZ,OMIGEN
124     01C1    D3 00                   OUT (00),A      ;strobe = high
125     01C3    C9                      RET
126
127
128                             ;;;;;;;;;;;;;;;;;;;;;;;;;; Print tabel ;;;;;;;;;;;;;;;;;;;;;;;;;;
129
130                             ORG TABEL
131
132     0200    0D 0A                   DEFB 0DH,0AH
133     0202    49 4E 49 54             DEFM "INITIALISERINGEN ER KORREKT !"
134     0206    49 41 4C 49
135     020A    53 45 52 49
136     020E    4E 47 45 4E
137     0212    20 45 52 20
138     0216    4B 4F 52 52
139     021A    45 4B 54 20
140     021E    21
141     021F    0D 0A 03                DEFB 0DH,0AH,03H
142
143                             END
```

```
Macros:

Symbols:
0180    COUNT           0002    CRA             0003    CRB
0170    EOT             01A0    INT             00FF    INTCTRW
00FF    INTDISW         00FF    IOCTRW          013E    LOOP
00FF    MOCTRW0         00FF    MOCTRW3         00FF    MSKCTRW
01BC    OMIGEN          0000    PORTA           0001    PORTB
01B6    PRINT           01B2    SLUT            0200    TABEL
0160    VECTOR          00FF    VECTORI         00FF    VECTORP


No Fatal error(s)
```

```
COUNT     27#   49    100   107
CRA       24#   36    55    59    61    65    67    92
CRB       25#   57
EOT       26#   47    88    101
INT       29#   44    99
INTCTRW   19#   64
INTDISW   13#   35    91
IOCTRW    18#   60
LOOP      75#   90
MOCTRW0   16#   56
MOCTRW3   17#   58
MSKCTRW   20#   66
OMIGEN    121#  123
PORTA     22#   63
PORTB     23#
PRINT     70    105   115#
SLUT      104   108#
TABEL     28#   48    130
VECTOR    21#   45
VECTORI   14#   42
VECTORP   15#   54
```

```
                           .COMMENT^

                                            EDITOR
                           -----------------------------------------------------------
                           !                                                         !
                           ! Programmet henter ASCII karakterer fra keyboard, og skriver !
                           ! disse på skærmen samt i adresse 0400H og fremefter.     !
                           !                                                         !
                           ! Cursoren kontrolleres med de fire pile samt BS.         !
                           !                                                         !
                           ! DEL sletter den sidst skrevne karakter                  !
                           !                                                         !
                           ! Return skifter til næste linje.                         !
                           !                                                         !
                           ! Indtastningen standses ved (ESC) eller (CNTRL C)        !
                           !                                                         !
                           -----------------------------------------------------------

                           ^

0000'                               ASEG
                                    ORG     100H

0400                                STACKT EQU  0400H
F0A0                                START  EQU  0F0A0H
00CB                                KEY    EQU  0CBH
0003                                CNTRLC EQU  03H
001B                                ESC    EQU  1BH
00FF                                NONASC EQU  0FFH
0300                                TABEL1 EQU  0300H
0310                                TABEL2 EQU  0310H
00FC                                ADREG  EQU  0FCH
00FD                                CNTREG EQU  0FDH
F000                                VIDRAM EQU  0F000H
0020                                SPACE  EQU  20H

                           ;;;;;;; Hovedprogram: ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

0100    31 0400            EDITOR: LXI   SP,STACKT ;Initialiser stackpointer.
0103    CD 015C                    CALL  INIT      ;Initialiser CRT controller.
0106    CD 0170                    CALL  CLEAR     ;Slet skærm.
0109    CD 0183                    CALL  ATTR      ;Attribut RAM: øverste linje = reverse.
010C    CD 0192                    CALL  TEKST     ;Skriv tekst på øverste linje.
010F    21 F0A0                    LXI   H,START   ;Peg på 1. skrivefelt.
0112    11 F0A0                    LXI   D,START   ;Peg på 1.linje i skrivefelt.
0115    DB CB              LOOP:   IN    KEY       ;Hent karakter fra keyboard.
0117    FE FF                      CPI   NONASC    ;Udfør, hvis karakter er ASCII-kode:
0119    CA 0155                    JZ    IGEN
011C    FE 03                      CPI   CNTRLC    ;Udfør, hvis karakter () ^C:
011E    CA 0158                    JZ    ENDED
0121    FE 1B                      CPI   ESC       ;Udfør, hvis karakter () ESC:
0123    CA 0158                    JZ    ENDED
0126    FE 20              IFO:    CPI   20H       ;Skriv ASCII karakter, hvis den
0128    DA 012E                    JC    ENDIFO    ;er 'synlig'.
```

```
 57  012B  CD 01F2          THENO:  CALL   WRITE
 58  012E  FE 7F            ENDIFO: CPI    7FH       ;Hvis kar. = DEL, så slet.
 59  0130  CC 01B8                  CZ     DELETE
 60  0133  FE 08                    CPI    08H       ;Cursor til venstre.
 61  0135  CC 01BF                  CZ     LEFT
 62  0138  FE 0C                    CPI    0CH       ;Cursor til højre.
 63  013A  CC 01C1                  CZ     RIGHT
 64  013D  FE 0B                    CPI    0BH       ;Cursor op.
 65  013F  CC 01C3                  CZ     UP
 66  0142  FE 0A                    CPI    0AH       ;Cursor ned.
 67  0144  CC 01CA                  CZ     DOWN
 68  0147  CD 01D1                  CALL   LPLAC     ;Juster (DE) hvis cursor er flyttet.
 69  014A  FE 0D                    CPI    0DH       ;Ny linje hvis RETURN er aktiv.
 70  014C  CC 01E8                  CZ     NYLIN
 71  014F  CD 01F5                  CALL   OUT       ;Test om cursor er udenfor skrivefelt.
 72  0152  CD 01A5                  CALL   CURSOR    ;Opdater cursor'ens position.
 73  0155  C3 0115          IGEN:   JMP    LOOP      ;Gentag polling.
 74  0158  CD 022E          ENDED:  CALL   KOPI      ;Flyt skærmens indhold til 0400H -
 75  015B  FF                       RST    07H       ;Tilbage til DDT command mode.
 76
 77                         ;;;;;;; Modulet initialiserer 6845 CRT controller: ;;;;;;;;;;;;;;;
 78
 79  015C  21 0300          INIT:   LXI    H,TABEL1  ;Læs tabel og placer tabellens
 80  015F  06 10                    MVI    B,10H     ;indhold i CRT controllerens
 81  0161  0E 00                    MVI    C,00H     ;første 16 registre:
 82  0163  79               AGAIN:  MOV    A,C
 83  0164  D3 FC                    OUT    0FCH
 84  0166  0C                       INR    C
 85  0167  7E                       MOV    A,M
 86  0168  D3 FD                    OUT    0FDH
 87  016A  23                       INX    H
 88  016B  05                       DCR    B
 89  016C  C2 0163                  JNZ    AGAIN
 90  016F  C9                       RET
 91
 92                         ;;;;;;; Modulet sletter skærmens hidtidige indhold: ;;;;;;;;;;;;;;
 93
 94  0170  21 F000          CLEAR:  LXI    H,VIDRAM  ;Peg på laveste adresse i video-RAM.
 95  0173  36 00            LOOPCL: MVI    M,00H     ;Nulstil indholdet af video -RAM
 96  0175  23                       INX    H         ;til og med slutadresse:
 97  0176  7C                       MOV    A,H
 98  0177  FE F7                    CPI    0F7H
 99  0179  C2 0173                  JNZ    LOOPCL
100  017C  7D                       MOV    A,L
101  017D  FE FF                    CPI    0FFH
102  017F  C2 0173                  JNZ    LOOPCL
103  0182  C9                       RET
104
105                         ;;;;;;; Modulet danner reverse skrift på øverste linje: ;;;;;;;;;
106
107  0183  21 F000          ATTR:   LXI    H,VIDRAM  ;Peg på laveste adresse i attribut-RAM.
108  0186  06 50                    MVI    B,50H     ;Initialiser øverste linje til
109  0188  D3 FE                    OUT    0FEH      ;reverse-skrift.
110  018A  36 40            LOOPAT: MVI    M,40H
111  018C  23                       INX    H
112  018D  05                       DCR    B
```

```
113   018E   C2 018A              JNZ   LOOPAT
114   0191   C9                  RET
115
116                      ;;;;;;; Modulet udskriver en ASCII-streng på øverste linje: ;;;;;
117
118   0192   21 F003    TEKST:   LXI   H,0F003H  ;Peg på 3. felt/øverste linje.
119   0195   11 0310              LXI   D,TABEL2  ;Peg på tabel med ASCII-karakterer.
120   0198   06 44                MVI   B,44H     ;Bestem tekststrengens længde.
121   019A   D3 FF                OUT   0FFH      ;Valg af video-RAM.
122   019C   1A        LOOPTK:  LDAX  D         ;Skriv tekst på øverste linje:
123   019D   77                  MOV   M,A
124   019E   23                  INX   H
125   019F   13                  INX   D
126   01A0   05                  DCR   B
127   01A1   C2 019C             JNZ   LOOPTK
128   01A4   C9                  RET
129
130                      ;;;;;;; Modulet placerer cursor'en på næste skrivefelt: ;;;;;;;;
131
132   01A5   F5        CURSOR:  PUSH  PSW
133   01A6   3E 10               MVI   A,10H     ;Lad 6845-adresse register pege på
134   01A8   D3 FC               OUT   ADREG     ;register XX.
135   01AA   7C                  MOV   A,H       ;Bring (H) ud til register XX.
136   01AB   E6 0F               ANI   0FH
137   01AD   D3 FD               OUT   CNTREG
138   01AF   3E 11               MVI   A,11H     ;Lad 6845 adresse-register pege på
139   01B1   D3 FC               OUT   ADREG     ;register XX.
140   01B3   7D                  MOV   A,L       ;Bring (L) ud til register XX.
141   01B4   D3 FD               OUT   CNTREG
142   01B6   F1                  POP   PSW
143   01B7   C9                  RET
144
145                      ;;;;;;; Modulet sletter den sidst skrevne karakter: ;;;;;;;;;;;;
146
147   01B8   36 20     DELETE:  MVI   M,SPACE   ;Skriv ASCII (SP) på cursorens plads.
148   01BA   2B                  DCX   H
149   01BB   36 20               MVI   M,SPACE   ;Skriv ASCII (SP) på pladsen før
150   01BD   2B                  DCX   H         ;cursoren.
151   01BE   C9                  RET
152
153                      ;;;;;;; Modulet flytter cursoren's position een til venstre: ;;;
154
155   01BF   2B        LEFT:    DCX   H         ;Flyt cursor et felt tilbage.
156   01C0   C9                  RET
157
158                      ;;;;;;; Modulet flytter cursoren's position een til højre: ;;;;;
159
160   01C1   23        RIGHT:   INX   H         ;Flyt cursor et felt frem.
161   01C2   C9                  RET
162
163                      ;;;;;;; Modulet flytter cursoren's position een op: ;;;;;;;;;;;;
164
165   01C3   C5        UP:      PUSH  B
166   01C4   01 FFB0             LXI   B,0FFB0H  ;Subtraher 80 fra (HL), hvilket
167   01C7   09                  DAD   B         ;flytter cursor et felt op.
168   01C8   C1                  POP   B
```

```
169   01C9   C9                    RET
170
171                         ;;;;;;; Modulet flytter cursoren's position een ned: ;;;;;;;;;;;
172
173   01CA   C5             DOWN:   PUSH   B
174   01CB   01 0050                LXI    B,0050H   ;Adder 80 til (HL), hvilket flytter
175   01CE   09                     DAD    B         ;cursoren en ned.
176   01CF   C1                     POP    B
177   01D0   C9                     RET
178
179                         ;;;;;;; Modulet opdaterer linjeposition efter brug af cursor: ;;;
180
181   01D1   F5             LPLAC:  PUSH   PSW
182   01D2   C5                     PUSH   B
183   01D3   E5                     PUSH   H
184   01D4   7C             IF1:    MOV    A,H       ;Hvis (HL), som peger på skrivefeltet
185   01D5   BA                     CMP    D         ;er mindre end linipointeren (DE):
186   01D6   DA 01DE                JC     THEN1
187   01D9   7D                     MOV    A,L
188   01DA   BB                     CMP    E
189   01DB   D2 01E4                JNC    ENDIF1
190   01DE   01 FFB0         THEN1:  LXI    B,0FFB0H  ;så subtraher (DE) med 80, så (DE)
191   01E1   EB                     XCHG             ;peger på linjen ovenfor.
192   01E2   09                     DAD    B
193   01E3   EB                     XCHG
194   01E4   E1              ENDIF1: POP    H
195   01E5   C1                     POP    B
196   01E6   F1                     POP    PSW
197   01E7   C9                     RET
198
199                         ;;;;;;; Modulet peger på 1. plads / næste linje efter CR: ;;;;;;
200
201   01E8   C5             NYLIN:  PUSH   B
202   01E9   01 0050                LXI    B,0050H   ;Adder (HL) med 80, og kopier
203   01EC   EB                     XCHG             ;(HL) over i (DE).
204   01ED   09                     DAD    B
205   01EE   5D                     MOV    E,L
206   01EF   54                     MOV    D,H
207   01F0   C1                     POP    B
208   01F1   C9                     RET
209
210                         ;;;;;;; Modulet skriver ASCII kar. i RAM ;;;;;;;;;;;;;;;;;;;;;;;;
211
212   01F2   77             WRITE:  MOV    M,A       ;Skriv karakter i video-RAM.
213   01F3   23                     INX    H
214   01F4   C9                     RET
215
216                         ;;;;;;; Modulet undersøger om cursor er uden for skrivefelt: ;;;;
217
218   01F5   F5             OUT:    PUSH   PSW
219   01F6   7C             CHECK1: MOV    A,H       ;Hvis (HL) ( 0F0A0H, som er laveste
220   01F7   FE F0                  CPI    0F0H      ;adresse på tekstfeltet:
221   01F9   DA 020B                JC     DEC1
222   01FC   CA 0202                JZ     CHL01
223   01FF   C3 0211                JMP    CHECK2
224   0202   7D             CHL01:  MOV    A,L
```

```
225   0203   FE A0              CPI   0A0H
226   0205   DA 020B            JC    DEC1
227   0208   C3 0211            JMP   CHECK2
228   020B   21 F0A0   DEC1:    LXI   H,START   ;så sæt cursoren til udgangsposition.
229   020E   11 F0A0            LXI   D,START
230   0211   3E F7     CHECK2:  MVI   A,0F7H    ;Hvis (HL) ) 0F780H, som er højeste
231   0213   BC                 CMP   H         ;adresse i skrivefeltet:
232   0214   DA 0226            JC    DEC2
233   0217   CA 021D            JZ    CHL02
234   021A   C3 022C            JMP   ENDOUT
235   021D   3E 80     CHL02:   MVI   A,80H
236   021F   BD                 CMP   L
237   0220   DA 0226            JC    DEC2
238   0223   C3 022C            JMP   ENDOUT
239   0226   21 F0A0   DEC2:    LXI   H,START   ;så sæt cursoren til udgangsposition.
240   0229   11 F0A0            LXI   D,START
241   022C   F1       ENDOUT:  POP   PSW
242   022D   C9                 RET
243
244                    ;;;;;;; Modulet flytter skærmens indhold til adr. 0400 og frem: ;
245
246   022E   11 0400   KOPI:    LXI   D,0400H   ;Peg på destination-adresse.
247   0231   21 F0A0            LXI   H,START   ;Peg på source-adresse.
248   0234   7C       WHILE:   MOV   A,H       ;Udfør sålænge source-adresse (= øverste
249   0235   FE F7               CPI   0F7H      ;adresse i video-RAM:
250   0237   C2 0240            JNZ   CONT
251   023A   7D                 MOV   A,L
252   023B   FE FF               CPI   0FFH
253   023D   CA 0251            JZ    ENDWHL
254   0240   7E       CONT:    MOV   A,M
255   0241   FE 20    IF:      CPI   SPACE     ;Hvis indholdet af video-RAM er tomt,
256   0243   D2 024B            JNC   ENDIF
257   0246   CA 024B            JZ    ENDIF
258   0249   3E 20    THEN:    MVI   A,20H     ;så erstat dette indhold med ASCII (SP)
259   024B   12       ENDIF:   STAX  D         ;skriv data i kopi-felt.
260   024C   23                 INX   H         ;Juster source-pointer.
261   024D   13                 INX   D         ;Juster destination pointer.
262   024E   C3 0234            JMP   WHILE
263   0251   C9       ENDWHL:  RET
264
265                    ;;;;;;; Tabel over initialisering af 6845 CRTC: ;;;;;;;;;;;;;;;;
266
267                            ORG 0300H
268
269   0300   7F 28 61 00 TABEL1:  DB 07FH,28H,61H,00H,1EH
270   0304   1E
271   0305   02 07 1B 00         02H,07H,1BH,00H,09H,44H,05H,00H,00H,0EH,0A0H
272   0309   09 44 05 00
273   030D   00 0E A0
274
275                    ;;;;;;; Tabel over tekst på øverste linie: ;;;;;;;;;;;;;;;;;;;;;
276
277                            ORG 0310H
278
279   0310   2A 20 49 6E TABEL2:  DB 2AH,20H,49H,6EH,74H,65H,72H,66H,61H,63H,65H,20H
280   0314   74 65 72 66
```

```
0318    61 63 65 20
031C    6B 75 72 73             DB 6BH,75H,72H,73H,75H,73H,20H,2AH,20H,20H,20H
0320    75 73 20 2A
0324    20 20 20
0327    20 20 20 45             DB 20H,20H,20H,45H,44H,49H,54H,3AH,20H,20H
032B    44 49 54 3A
032F    20 20
0331    44 45 4C 2C             DB 44H,45H,4CH,2CH,20H,42H,53H,2CH,20H,63H,75H
0335    20 42 53 2C
0339    20 63 75
033C    72 73 6F 72             DB 72H,73H,6FH,72H,20H,20H,20H,20H
0340    20 20 20 20
0344    20 20 45 58             DB 20H,20H,45H,58H,49H,54H,3AH,20H,20H,5EH,43H
0348    49 54 3A 20
034C    20 5E 43
034F    2C 20 45 53             DB 2CH,20H,45H,53H,43H
0353    43

                        ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

                                END


Macros:

Symbols:
00FC  ADREG      0163  AGAIN      0183  ATTR
01F6  CHECK1     0211  CHECK2     0202  CHLO1
021D  CHLO2      0170  CLEAR      00FD  CNTREG
0003  CNTRLC     0240  CONT       01A5  CURSOR
020B  DEC1       0226  DEC2       01B8  DELETE
01CA  DOWN       0100  EDITOR     0158  ENDED
024B  ENDIF      012E  ENDIF0     01E4  ENDIF1
022C  ENDOUT     0251  ENDWHL     001B  ESC
0241  IF         0126  IF0        01D4  IF1
0155  IGEN       015C  INIT       00CB  KEY
022E  KOPI       01BF  LEFT       0115  LOOP
018A  LOOPAT     0173  LOOPCL     019C  LOOPTK
01D1  LPLAC      00FF  NONASC     01E8  NYLIN
01F5  OUT        01C1  RIGHT      0020  SPACE
0400  STACKT     F0A0  START      0300  TABEL1
0310  TABEL2     0192  TEKST      0249  THEN
012B  THEN0      01DE  THEN1      01C3  UP
F000  VIDRAM     0234  WHILE      01F2  WRITE


No Fatal error(s)

C
```

```
ADREG     34#   134    139
AGAIN     82#    89
ATTR      44    107#
CHECK1   219#
CHECK2   223    227    230#
CHLO1    222    224#
CHLO2    233    235#
CLEAR     43     94#
CNTREG    35#   137    141
CNTRLC    29#    51
CONT     250    254#
CURSOR    72    132#
DEC1     221    226    228#
DEC2     232    237    239#
DELETE    59    147#
DOWN      67    173#
EDITOR    41#
ENDED     52     54     74#
ENDIF    256    257    259#
ENDIF0    56     58#
ENDIF1   189    194#
ENDOUT   234    238    241#
ENDWHL   253    263#
ESC       30#    53
IF       255#
IF0       55#
IF1      184#
IGEN      50     73#
INIT      42     79#
KEY       28#    48
KOPI      74    246#
LEFT      61    155#
LOOP      48#    73
LOOPAT   110#   113
LOOPCL    95#    99    102
LOOPTK   122#   127
LPLAC     68    181#
NONASC    31#    49
NYLIN     70    201#
OUT       71    218#
RIGHT     63    160#
SPACE     37#   147    149    255
STACKT    26#    41
START     27#    46     47    228    229    239    240    247
TABEL1    32#    79    269#
TABEL2    33#   119    279#
TEKST     45    118#
THEN     258#
THEN0     57#
THEN1    186    190#
UP        65    165#
VIDRAM    36#    94    107
WHILE    248#   262
WRITE     57    212#
```

```
                                .COMMENT^

                                                    TEKST
                                ------------------------------------------------------
                                !                                                     !
                                !     TEKST, skriver ASCII karakterer på 3 linjer, hvor:
                                !                                                     !
                                !           1. linje skrives normalt.                 !
                                !           2. linje skrives med underline skrift.    !
                                !           3. linje skrives med reverse skrift.      !
                                !                                                     !
                                !     Programmet anvendes til måling på MPS 26 A.     !
                                !     Programmet standses ved (ESC).                  !
                                !                                                     !
                                ------------------------------------------------------
                                ^
0000'                                   ASEG
                                        ORG     0100H

0100    31 0200         BEGIN:  LXI     SP,0200H ; Sæt stackpointer.
0103    CD 0126                 CALL    CLEAR    ; Ryd skærm.
0106    21 F000                 LXI     H,0F000H ; Peg på 1. linje.
0109    0E 00                   MVI     C,00H    ; Vælg attribut = normal skrift.
010B    CD 013F                 CALL    WRITE    ; Skriv 1. linje.
010E    21 F0A0                 LXI     H,0F0A0H ; Peg på 3. linje.
0111    0E 20                   MVI     C,20H    ; Vælg attribut = underline.
0113    CD 013F                 CALL    WRITE    ; Skriv 3. linje.
0116    21 F140                 LXI     H,0F140H ; Peg på 5. linje.
0119    0E 40                   MVI     C,40H    ; Vælg attribut = reverse.
011B    CD 013F                 CALL    WRITE    ; Skriv 5. linje.
011E    DB CB           END:    IN      0CBH     ; Hop ud af program,
0120    FE 1B                   CPI     1BH      ; når ESC aktiveres.
0122    C2 011E                 JNZ     END
0125    FF                      RST     07H

0126    21 F000         CLEAR:  LXI     H,0F000H ; Peg på 1. skrivefelt.
0129    36 00           LOOP1:  MVI     M,00     ; Slet skrivefelt.
012B    23                      INX     H        ; Peg på næste skrivefelt.
012C    7C                      MOV     A,H      ; Gentag processen indtil
012D    FE F7                   CPI     0F7H     ; sidste skrivefelt er nået:
012F    C2 013B                 JNZ     AGAIN
0132    7D                      MOV     A,L
0133    FE 80                   CPI     80H
0135    C2 013B                 JNZ     AGAIN
0138    C3 013E                 JMP     ENDCL
013B    C3 0129         AGAIN:  JMP     LOOP1
013E    C9              ENDCL:  RET             ; Tilbage til hovedprogram.

013F    06 50           WRITE:  MVI     B,50H    ; Antal karakterer / linje = 80 D.
0141    1E 30                   MVI     E,30H    ; 1. karakter = ASCII 0.
0143    D3 FE           LOOP2:  OUT     0FEH     ; Vælg attribut RAM.
0145    71                      MOV     M,C      ; Udlæs attribut.
0146    D3 FF                   OUT     0FFH     ; Vælg video RAM.
0148    73                      MOV     M,E      ; Udlæs ASCII karakter.
0149    1C                      INR     E        ; Næste karakter.
014A    23                      INX     H        ; Næste skrivefelt.
```

```
 57    014B   05                        DCR     B       ; Gentag indtil linjen er udfyldt.
 58    014C   CA 0152                   JZ      ENDWR
 59    014F   C3 0143                   JMP     LOOP2
 60    0152   C9              ENDWR:    RET             ; Tilbage til hovedprogram.
 61
 62                                     END


Macros:

Symbols:
013B    AGAIN           0100    BEGIN           0126    CLEAR
011E    END             013E    ENDCL           0152    ENDWR
0129    LOOP1           0143    LOOP2           013F    WRITE


No Fatal error(s)

C


AGAIN       41      44      46#
BEGIN       20#
CLEAR       21      36#
END         31#     33
ENDCL       45      47#
ENDWR       58      60#
LOOP1       37#     46
LOOP2       51#     59
WRITE       24      27      30      49#
```

```
                                .COMMENT^

                                                    LOLIGHT
                                ------------------------------------------------
                                !                                              !
                                !   Funktion:                                  !
                                !                                              !
                                !        1.   Fjerner cursor.                  !
                                !        2.   Skriver 24 * 80 'firkanter'.     !
                                !        3.   Attribut RAM = normal lysstyrke. !
                                !                                              !
                                !   Programmet anvendes i forbindelse med måle-!
                                !   øvelse på 6845 CRT controller.             !
                                !   Programmet afbrydes ved (ESC).             !
                                !                                              !
                                ------------------------------------------------
                                ^
0000'                                   ASEG
                                        ORG     0100H

0100    3E 0A           START:  MVI     A,0AH   ;Fjern cursor.
0102    D3 FC                   OUT     0FCH
0104    3E 20                   MVI     A,20H
0106    D3 FD                   OUT     0FDH
0108    21 F000                 LXI     H,0F000H ;Peg på skærmens startadresse.
010B    7C              WHILE:  MOV     A,H     ;Mens startadresse ( = slutadresse:
010C    FE F7                   CPI     0F7H
010E    C2 0114                 JNZ     TEST
0111    7D                      MOV     A,L
0112    FE 80                   CPI     80H
0114    CA 011A         TEST:   JZ      DO
0117    D2 0126                 JNC     ENDWHL
011A    D3 FE           DO:     OUT     0FEH    ;Vælg attribut RAM.
011C    36 00                   MVI     M,00H   ;Sæt skrivefelt til normal lysstyrke.
011E    D3 FF                   OUT     0FFH    ;Vælg video RAM.
0120    36 1C                   MVI     M,1CH   ;Skriv firkant i video RAM.
0122    23                      INX     H       ;Klarger næste adresse.
0123    C3 010B                 JMP     WHILE   ;Hop til check for gentagelse.
0126    DB CB           ENDWHL: IN      0CBH    ;Returner til DDT command mode,
0128    FE 1B                   CPI     1BH     ;når ESC aktiveres.
012A    C2 0126                 JNZ     ENDWHL
012D    FF                      RST     07H

                                END
```

```
1                                .COMMENT^
2
3                                                    HILIGHT
4                                ------------------------------------------------------
5                                !                                                    !
6                                !   Funktion:                                        !
7                                !                                                    !
8                                !        1. Fjerner cursor.                          !
9                                !        2. Skriver 24 * 80 'firkanter'              !
10                               !        3. Attribut RAM sættes til highlight.       !
11                               !                                                    !
12                               !   Programmet anvendes i forbindelse med måle-      !
13                               !   øvelse på 6845 CRT controller.                   !
14                               !   Programmet standses ved (ESC).                   !
15                               !                                                    !
16                               ------------------------------------------------------
17                               ^
18   0000'                               ASEG
19                                       ORG     0100H
20
21   0100   3E 0A          LIGHT:  MVI     A,0AH    ;Fjern cursor.
22   0102   D3 FC                  OUT     0FCH
23   0104   3E 20                  MVI     A,20H
24   0106   D3 FD                  OUT     0FDH
25   0108   21 F000                LXI     H,0F000H ;Peg på skærmens startadresse
26   010B   7C            WHILE:  MOV     A,H      ;Udfør, mens startadresse ( F000H )
27   010C   FE F7                  CPI     0F7H     ;er (= slutadresse ( F780H ):
28   010E   C2 0114                JNZ     TEST
29   0111   7D                    MOV     A,L
30   0112   FE 80                  CPI     80H
31   0114   CA 011A        TEST:   JZ      DO
32   0117   D2 0126                JNC     ENDWHL
33   011A   D3 FE          DO:     OUT     0FEH     ;Vælg attribut-RAM.
34   011C   36 80                  MVI     M,80H    ;Sæt aktuel karakter til highlight.
35   011E   D3 FF                  OUT     0FFH     ;Vælg video-RAM.
36   0120   36 1C                  MVI     M,1CH    ;Skriv 'firkant' i video-RAM.
37   0122   23                    INX     H        ;Klargør næste adresse
38   0123   C3 010B                JMP     WHILE    ;Hop til check for gentagelse.
39   0126   DB CB        ENDWHL: IN      0CBH     ;Vedvarende loop indtil RESET.
40   0128   FE 1B                  CPI     1BH      ;Returner til DDT, command mode,
41   012A   C2 0126                JNZ     ENDWHL   ;når ESC aktiveres.
42   012D   FF                     RST     07H
43
44                                       END
```

```
                                .COMMENT^

                                                FUNKTION
                                ------------------------------------------------------
                                !                                                    !
                                ! Programmet danner følgende grafikbillede:         !
                                !                                                    !
                                ! -x-akse og y-akse.                                 !
                                ! -pile på x-akse og y-akse.                         !
                                ! -9 afmærkninger på x-akse og y-akse.               !
                                ! -tal (1-9) på x-akse og y-akse.                    !
                                ! -teksten: 'X' og 'f(X)'      ved hhv. x-akse og y-akse.  !
                                ! -teksten 'Koordinatsystem' øverst på billedet.     !
                                ! -3 forskellige funktioner med forskellige 'streg-typer'.  !
                                !                                                    !
                                ! Programmet kan eksekveres i en Comet 3400 under forud-  !
                                ! sætning af, at grafikmodul MPS-24 er monteret.     !
                                !                                                    !
                                ! Ved at aktivere en tast vendes tilbage til video.  !
                                !                                                    !
                                ------------------------------------------------------
                                ^

0000'                                   ASEG                    ; Startadresse = 0100H.
                                        ORG     0100H

1000                                    STACK   EQU     1000H ; Stacktop.
0900                                    TABEL   EQU     0900H ; Adr. på ASCII-tabel.
00A0                                    CMD     EQU     160   ; GP's kommando-register.
00A1                                    CTRL1   EQU     161   ; GP's 1. kontrol-register.
00A2                                    CTRL2   EQU     162   ; GP's 2. kontrol-register.
00A3                                    CSIZE   EQU     163   ; Karakter-format.
00A5                                    DELTAX  EQU     165   ; Linjetilvækst i x retning.
00A7                                    DELTAY  EQU     167   ; Linjetilvækst i y-retning.
00A8                                    XHIGH   EQU     168   ; x-position ( high byte ).
00A9                                    XLOW    EQU     169   ; x-position ( low byte ).
00AA                                    YHIGH   EQU     170   ; y-position ( high byte ).
00AB                                    YLOW    EQU     171   ; y-position ( low byte ).

0100    31 1000                 START:  LXI     SP,STACK ; Initialiser stackpointer.

0103    3E FF                           MVI     A,255    ; Vælg grafik.
0105    D3 B0                           OUT     176

0107    3E 07                           MVI     A,7      ; Nulstil x-reg. og y-reg.
0109    D3 A0                           OUT     CMD
010B    CD 03C6                         CALL    STATUS

010E    CD 012D                         CALL    XYAKS    ; Tegn x-akse og y-akse.
0111    CD 01BB                         CALL    XPUNKT   ; Afsæt punkter på x-akse.
0114    CD 01F5                         CALL    YPUNKT   ; Afsæt punkter på y-akse.
0117    CD 02A1                         CALL    X        ; Afsæt 'X' ved x-akse.
011A    CD 02C0                         CALL    FX       ; Afsæt 'f(X)' ved y-akse.
011D    CD 02F4                         CALL    TEKST    ; Skriv tekst.
0120    CD 031C                         CALL    FUNK1    ; Tegn 'ustiplet' funktion.
```

```
 57   0123   CD 033B                  CALL   FUNK2    ; Tegn 'punkteret' funktion.
 58   0126   CD 037C                  CALL   FUNK3    ; Tegn 'stiplet' funktion.
 59   0129   CD 03BB                  CALL   VIDEO    ; Skift til video, når tast aktiveres.
 60   012C   FF                       RST    7        ; DDT, command mode.
 61
 62
 63                          ;;;;;;; XYAKS, modulet tegner x-akse og y-akse ;;;;;;;;;;;;;;;
 64
 65   012D   3E 0B            XYAKS:  MVI    A,11     ; Pen nede, cyklisk skærm.
 66   012F   D3 A1                    OUT    CTRL1
 67
 68   0131   3E 32                    MVI    A,50     ; (x,y) = (50,50)
 69   0133   D3 A9                    OUT    XLOW
 70   0135   D3 AB                    OUT    YLOW
 71
 72   0137   3E CE                    MVI    A,206    ; Delta y = 206.
 73   0139   D3 A7                    OUT    DELTAY
 74
 75   013B   3E 12                    MVI    A,18     ; Tegn 1. del af y-akse.
 76   013D   D3 A0                    OUT    CMD
 77   013F   CD 03C6                  CALL   STATUS
 78
 79   0142   3E 01                    MVI    A,1      ; Tegn 2. del af y-akse.
 80   0144   D3 AA                    OUT    YHIGH
 81   0146   3E 00                    MVI    A,0
 82   0148   D3 AB                    OUT    YLOW
 83   014A   3E 12                    MVI    A,18
 84   014C   D3 A0                    OUT    CMD
 85   014E   CD 03C6                  CALL   STATUS
 86   0151   CD 017F                  CALL   YPIL
 87
 88   0154   3E 05                    MVI    A,5      ; Nulstil x-reg. og y-reg.
 89   0156   D3 A0                    OUT    CMD
 90   0158   CD 03C6                  CALL   STATUS
 91
 92   015B   3E 32                    MVI    A,50     ; (x,y) = (50,50).
 93   015D   D3 A9                    OUT    XLOW
 94   015F   D3 AB                    OUT    YLOW
 95
 96   0161   3E CE                    MVI    A,206    ; Sæt delta x til 206.
 97   0163   D3 A5                    OUT    DELTAX
 98
 99   0165   3E 10                    MVI    A,16     ; Tegn 1. del af x-akse.
100   0167   D3 A0                    OUT    CMD
101   0169   CD 03C6                  CALL   STATUS
102
103   016C   3E 01                    MVI    A,1      ; Tegn 2. del af x-akse.
104   016E   D3 A8                    OUT    XHIGH
105   0170   3E 00                    MVI    A,0
106   0172   D3 A9                    OUT    XLOW
107   0174   3E 10                    MVI    A,16
108   0176   D3 A0                    OUT    CMD
109   0178   CD 03C6                  CALL   STATUS
110   017B   CD 019D                  CALL   XPIL
111   017E   C9                       RET
112
```

```
113
114                              ;;;;;;; YPIL, tegner pil på y-akse ;;;;;;;;;;;;;;;;;;;;;
115
116   017F   3E 08              YPIL:  MVI    A,8      ; Tegn højre halvdel af pil.
117   0181   D3 A5                     OUT    DELTAX
118   0183   3E 0F                     MVI    A,15
119   0185   D3 A7                     OUT    DELTAY
120   0187   3E 11                     MVI    A,17
121   0189   D3 A0                     OUT    CMD
122   018B   CD 03C6                   CALL   STATUS
123
124   018E   3E 17                     MVI    A,23     ; Tilbage til udgangspunkt.
125   0190   D3 A0                     OUT    CMD
126   0192   CD 03C6                   CALL   STATUS
127
128   0195   3E 13                     MVI    A,19     ; Tegn venstre halvdel af pil.
129   0197   D3 A0                     OUT    CMD
130   0199   CD 03C6                   CALL   STATUS
131   019C   C9                        RET
132
133
134                              ;;;;;;; XPIL, tegner pil på x-aksen ;;;;;;;;;;;;;;;;;;;;;;;;;;;
135
136   019D   3E 0A              XPIL:  MVI    A,10     ; Tegn nederste halvdel af pil.
137   019F   D3 A7                     OUT    DELTAY
138   01A1   3E 0F                     MVI    A,15
139   01A3   D3 A5                     OUT    DELTAX
140   01A5   3E 17                     MVI    A,23
141   01A7   D3 A0                     OUT    CMD
142   01A9   CD 03C6                   CALL   STATUS
143
144   01AC   3E 11                     MVI    A,17     ; Tilbage til udgangspunkt.
145   01AE   D3 A0                     OUT    CMD
146   01B0   CD 03C6                   CALL   STATUS
147
148   01B3   3E 13                     MVI    A,19     ; Tegn øverste halvdel af pil.
149   01B5   D3 A0                     OUT    CMD
150   01B7   CD 03C6                   CALL   STATUS
151   01BA   C9                        RET
152
153
154                              ;;;;;;; XPUNKT, afsætter 9 punkter hen ad x-aksen ;;;;;;;;;;;
155
156   01BB   3E 05              XPUNKT: MVI    A,5      ; Slet x-reg. og y-reg.
157   01BD   D3 A0                     OUT    CMD
158   01BF   CD 03C6                   CALL   STATUS
159
160   01C2   06 01                     MVI    B,1      ; Initialiser looptæller.
161
162   01C4   3E 2D                     MVI    A,45     ; Sæt y fem punkter under x-akse.
163   01C6   D3 AB                     OUT    YLOW
164
165   01C8   3E 32                     MVI    A,50     ; Lad x starte fra y-aksen.
166   01CA   D3 A9                     OUT    XLOW
167
168   01CC   3E 23              REP1:  MVI    A,35     ; Afstand mellem punkter = 35.
```

```
169   01CE   D3 A5                    OUT     DELTAX
170   01D0   3E 0A                    MVI     A,10     ; Punkterne 'fylder' 10 dots lodret.
171   01D2   D3 A7                    OUT     DELTAY
172   01D4   3E 0A                    MVI     A,10     ; Løft pen.
173   01D6   D3 A1                    OUT     CTRL1
174   01D8   3E 11                    MVI     A,17     ; Flyt pen ( usynlig linje tegnes ).
175   01DA   D3 A0                    OUT     CMD
176   01DC   CD 03C6                  CALL    STATUS
177   01DF   3E 0B                    MVI     A,11     ; Sænk pen.
178   01E1   D3 A1                    OUT     CTRL1
179   01E3   3E 14                    MVI     A,20     ; Tegn afmærkningen.
180   01E5   D3 A0                    OUT     CMD
181   01E7   CD 03C6                  CALL    STATUS
182   01EA   CD 022F                  CALL    XTAL     ; Skriv tal.
183   01ED   04                       INR     B        ; Juster looptæller.
184   01EE   78                       MOV     A,B
185   01EF   FE 0A          UNTIL1:   CPI     10       ; Gentag til og med '9'.
186   01F1   DA 01CC                  JC      REP1
187   01F4   C9                       RET
188
189
190                         ;;;;;;;  YPUNKT, afsætter 9 punkter op ad y-aksen ;;;;;;;;;;;;;;;
191
192   01F5   3E 05          YPUNKT:   MVI     A,5      ; Slet x-reg. og y-reg.
193   01F7   D3 A0                    OUT     CMD
194   01F9   CD 03C6                  CALL    STATUS
195
196   01FC   06 01                    MVI     B,1      ; Initialiser looptæller.
197
198   01FE   3E 37                    MVI     A,55     ; Sæt x 5 punkter til højre for y-akse.
199   0200   D3 A9                    OUT     XLOW
200
201   0202   3E 32                    MVI     A,50     ; Lad y starte fra x-aksen
202   0204   D3 AB                    OUT     YLOW
203
204   0206   3E 23          REP2:     MVI     A,35     ; Afstand mellem punkter = 35.
205   0208   D3 A7                    OUT     DELTAY
206   020A   3E 0A                    MVI     A,10     ; Punkterne 'fylder' 10 dots henad.
207   020C   D3 A5                    OUT     DELTAX
208   020E   3E 0A                    MVI     A,10     ; Løft pen.
209   0210   D3 A1                    OUT     CTRL1
210   0212   3E 13                    MVI     A,19     ; Flyt pen.
211   0214   D3 A0                    OUT     CMD
212   0216   CD 03C6                  CALL    STATUS
213   0219   3E 0B                    MVI     A,11     ; Sænk pen.
214   021B   D3 A1                    OUT     CTRL1
215   021D   3E 10                    MVI     A,16     ; Tegn afmærkning.
216   021F   D3 A0                    OUT     CMD
217   0221   CD 03C6                  CALL    STATUS
218   0224   CD 0266                  CALL    YTAL     ; Skriv tal.
219   0227   04                       INR     B        ; Juster looptæller.
220   0228   78                       MOV     A,B
221   0229   FE 0A          UNTIL2:   CPI     10       ; Gentag til og med '9'.
222   022B   DA 0206                  JC      REP2
223   022E   C9                       RET
224
```

```
225
226                             ;;;;;;; XTAL, afsætter tal ( 1-9 ) henad x-aksen ;;;;;;;;;;;;;;;
227
228   022F   3E 0A             XTAL:  MVI    A,10      ; Løft pen.
229   0231   D3 A1                    OUT    CTRL1
230
231   0233   3E 05                    MVI    A,5       ; Delta x = 5.
232   0235   D3 A5                    OUT    DELTAX
233
234   0237   3E 16                    MVI    A,22      ; Delta y = 22.
235   0239   D3 A7                    OUT    DELTAY
236
237   023B   3E 17                    MVI    A,23      ; Flyt pen ned og til venstre.
238   023D   D3 A0                    OUT    CMD
239   023F   CD 03C6                  CALL   STATUS
240
241   0242   3E 0B                    MVI    A,11      ; Sænk pen.
242   0244   D3 A1                    OUT    CTRL1
243
244   0246   3E 22                    MVI    A,22H     ; Karakterstørrelse fordobles.
245   0248   D3 A3  regnet-o 3        OUT    CSIZE
246
247   024A   78                      MOV    A,B       ; Register B's indhold ændres til ASCII,
248   024B   F6 30                    ORI    30H
249   024D   D3 A0                    OUT    CMD       ; og skrives som kommando.
250   024F   CD 03C6                  CALL   STATUS
251
252   0252   3E 07                    MVI    A,7       ; x reg og y-reg ændres til
253   0254   D3 A5                    OUT    DELTAX    ; deres oprindelige værdi.
254   0256   3E 16                    MVI    A,22
255   0258   D3 A7                    OUT    DELTAY
256   025A   3E 0A                    MVI    A,10
257   025C   D3 A1                    OUT    CTRL1
258   025E   3E 13                    MVI    A,19
259   0260   D3 A0                    OUT    CMD
260   0262   CD 03C6                  CALL   STATUS
261   0265   C9                       RET
262
263
264                             ;;;;;;; YTAL, afsætter tal ( 1-9 ) op ad y-aksen ;;;;;;;;;;;;;;
265
266   0266   3E 0A             YTAL:  MVI    A,10      ; Løft pen.
267   0268   D3 A1                    OUT    CTRL1
268
269   026A   3E 19                    MVI    A,25      ; Delta x = 25.
270   026C   D3 A5                    OUT    DELTAX
271
272   026E   3E 08                    MVI    A,8       ; Delta y = 8.
273   0270   D3 A7                    OUT    DELTAY
274
275   0272   3E 17                    MVI    A,23      ; Flyt pen ned og til venstre.
276   0274   D3 A0                    OUT    CMD
277   0276   CD 03C6                  CALL   STATUS
278
279   0279   3E 0B                    MVI    A,11      ; Sænk pen.
280   027B   D3 A1                    OUT    CTRL1
```

```
281
282   027D   3E 22              MVI    A,22H    ; Karakterstørrelse fordobles.
283   027F   D3 A3              OUT    CSIZE
284
285   0281   78                 MOV    A,B      ; Register B's indhold ændres til ASCII,
286   0282   F6 30              ORI    30H
287   0284   D3 A0              OUT    CMD      ; og skrives som kommando.
288   0286   CD 03C6            CALL   STATUS
289
290   0289   3E 0D              MVI    A,13     ; x-reg. og y reg. ændres til
291   028B   D3 A5              OUT    DELTAX   ; deres oprindelige værdi.
292   028D   3E 08              MVI    A,8
293   028F   D3 A7              OUT    DELTAY
294   0291   3E 0A              MVI    A,10
295   0293   D3 A1              OUT    CTRL1
296   0295   3E 11              MVI    A,17
297   0297   D3 A0              OUT    CMD
298   0299   CD 03C6            CALL   STATUS
299   029C   3E 0B              MVI    A,11
300   029E   D3 A1              OUT    CTRL1
301   02A0   C9                 RET
302
303
304                     ;;;;;;; X, skriver X ved pilen af x-aksen ;;;;;;;;;;;;;;;;;;;
305
306   02A1   3E 05        X:     MVI    A,5      ; Nulstil x-reg. og y-reg.
307   02A3   D3 A0               OUT    CMD
308   02A5   CD 03C6             CALL   STATUS
309
310   02A8   3E 01               MVI    A,1      ; Sæt startposition.
311   02AA   D3 A8               OUT    XHIGH
312   02AC   3E A0               MVI    A,160
313   02AE   D3 A9               OUT    XLOW
314   02B0   3E 14               MVI    A,20
315   02B2   D3 AB               OUT    YLOW
316
317   02B4   3E 12               MVI    A,12H    ; Sæt bogstav-format.
318   02B6   D3 A3               OUT    CSIZE
319
320   02B8   3E 58               MVI    A,58H    ; Skriv 'X'
321   02BA   D3 A0               OUT    CMD
322   02BC   CD 03C6             CALL   STATUS
323   02BF   C9                  RET
324
325
326                     ;;;;;;; FX, skriver f(X) ved pilen af y-aksen ;;;;;;;;;;;;;;;;
327
328   02C0   3E 05        FX:    MVI    A,5      ; Nulstil x-reg. og y-reg.
329   02C2   D3 A0               OUT    CMD
330   02C4   CD 03C6             CALL   STATUS
331
332   02C7   3E 0F               MVI    A,15     ; Sæt startposition.
333   02C9   D3 A9               OUT    XLOW
334   02CB   3E 01               MVI    A,1
335   02CD   D3 AA               OUT    YHIGH
336   02CF   3E 96               MVI    A,150
```

```
337   02D1   D3 AB              OUT    YLOW
338
339   02D3   3E 12              MVI    A,12H    ; Sæt bogstav-format.
340   02D5   D3 A3              OUT    CSIZE
341
342   02D7   3E 66              MVI    A,66H    ; Skriv 'f'.
343   02D9   D3 A0              OUT    CMD
344   02DB   CD 03C6            CALL   STATUS
345
346   02DE   3E 28              MVI    A,28H    ; Skriv '('.
347   02E0   D3 A0              OUT    CMD
348   02E2   CD 03C6            CALL   STATUS
349
350   02E5   3E 59              MVI    A,59H    ; Skriv 'X'.
351   02E7   D3 A0              OUT    CMD
352   02E9   CD 03C6            CALL   STATUS
353
354   02EC   3E 29              MVI    A,29H    ; Skriv ')'.
355   02EE   D3 A0              OUT    CMD
356   02F0   CD 03C6            CALL   STATUS
357   02F3   C9                 RET
358
359
360                       ;;;;;;; TEKST, skriver 'Koordinatsystem' ;;;;;;;;;;;;;;;;;;;;;;
361
362   02F4   3E 05        TEKST: MVI    A,5      ; Nulstil x-reg. og y-reg.
363   02F6   D3 A0              OUT    CMD
364   02F8   CD 03C6            CALL   STATUS
365
366   02FB   3E 33              MVI    A,33H    ; Sæt bogstav-format.
367   02FD   D3 A3              OUT    CSIZE
368
369   02FF   3E 78              MVI    A,120    ; Sæt tekststrengens startposition.
370   0301   D3 A9              OUT    XLOW
371   0303   3E 01              MVI    A,1
372   0305   D3 AA              OUT    YHIGH
373   0307   3E C8              MVI    A,200
374   0309   D3 AB              OUT    YLOW
375
376   030B   0E 0F              MVI    C,15     ; Sæt tekststrengens længde.
377
378   030D   21 0900            LXI    H,TABEL  ; Peg på tekst-tabel.
379
380   0310   7E          GENTAG: MOV    A,M      ; Udskriv tekststreng.
381   0311   D3 A0              OUT    CMD
382   0313   CD 03C6            CALL   STATUS
383   0316   23                 INX    H
384   0317   0D                 DCR    C
385   0318   C2 0310            JNZ    GENTAG
386   031B   C9                 RET
387
388
389                       ;;;;;;; FUNK1, afbildning af 'ustiplet' funktion ;;;;;;;;;;;;
390
391   031C   3E 05        FUNK1: MVI    A,5      ; Slet indholdet af x-reg. og y-reg.
392   031E   D3 A0              OUT    CMD
```

```
393   0320   CD 03C6                  CALL   STATUS
394
395   0323   3E 32                    MVI    A,50     ; x = 50 ( 0 i koordinatsystemet ).
396   0325   D3 A9                    OUT    XLOW
397
398   0327   3E BE                    MVI    A,190    ; y = 190 ( 4 i koordinatsystemet ).
399   0329   D3 AB                    OUT    YLOW
400
401   032B   3E 46                    MVI    A,70     ; Delta x = 70
402   032D   D3 A5                    OUT    DELTAX
403
404   032F   3E 8C                    MVI    A,140    ; Delta y = 140
405   0331   D3 A7                    OUT    DELTAY
406
407   0333   3E 15                    MVI    A,21     ; Tegn linie 1.
408   0335   D3 A0                    OUT    CMD
409   0337   CD 03C6                  CALL   STATUS
410   033A   C9                       RET
411
412
413                          ;;;;;;; FUNK2, afbildning af 'punkteret' funktion ;;;;;;;;;;;;;
414
415   033B   3E 05           FUNK2:  MVI    A,5      ; Slet indholdet af x-reg. og y-reg.
416   033D   D3 A0                    OUT    CMD
417   033F   CD 03C6                  CALL   STATUS
418
419   0342   3E 01                    MVI    A,1      ; Tegn med punkteret linje.
420   0344   D3 A2                    OUT    CTRL2
421
422   0346   3E 32                    MVI    A,50     ; x = 0 i koordinatsystemet.
423   0348   D3 A9                    OUT    XLOW
424
425   034A   3E 55                    MVI    A,85     ; y = 1 i koordinatsystemet.
426   034C   D3 AB                    OUT    YLOW
427
428   034E   3E 42                    MVI    A,66     ; Tegn 1. afsnit af funktionen.
429   0350   D3 A5                    OUT    DELTAX
430   0352   3E 4B                    MVI    A,75
431   0354   D3 A7                    OUT    DELTAY
432   0356   3E 11                    MVI    A,17
433   0358   D3 A0                    OUT    CMD
434   035A   CD 03C6                  CALL   STATUS
435
436   035D   3E A4                    MVI    A,164    ; Tegn 2. afsnit af funktionen.
437   035F   D3 A5                    OUT    DELTAX
438   0361   3E 2C                    MVI    A,44
439   0363   D3 A7                    OUT    DELTAY
440   0365   3E 15                    MVI    A,21
441   0367   D3 A0                    OUT    CMD
442   0369   CD 03C6                  CALL   STATUS
443
444   036C   3E 54                    MVI    A,84     ; Tegn 3. afsnit af funktionen.
445   036E   D3 A5                    OUT    DELTAX
446   0370   3E 86                    MVI    A,134
447   0372   D3 A7                    OUT    DELTAY
448   0374   3E 11                    MVI    A,17
```

```
449   0376   D3 A0                    OUT    CMD
450   0378   CD 03C6                  CALL   STATUS
451   037B   C9                       RET
452
453
454                          ;;;;;;; FUNK3, afbildnig af 'stiplet' funktion ;;;;;;;;;;;;;;;;
455
456   037C   3E 05             FUNK3:  MVI    A,5      ; Slet indholdet af x-reg. og y-reg.
457   037E   D3 A0                     OUT    CMD
458   0380   CD 03C6                   CALL   STATUS
459
460   0383   3E 02                     MVI    A,2      ; Tegn med stiplet linje.
461   0385   D3 A2                     OUT    CTRL2
462
463   0387   3E 32                     MVI    A,50     ; Sæt startposition for x og y.
464   0389   D3 A9                     OUT    XLOW
465   038B   3E 8F                     MVI    A,143
466   038D   D3 AB                     OUT    YLOW
467
468   038F   3E 56                     MVI    A,86     ; Tegn 1. afsnit af funktionen.
469   0391   D3 A5                     OUT    DELTAX
470   0393   3E 05                     MVI    A,5
471   0395   3E 11                     MVI    A,17
472   0397   D3 A0                     OUT    CMD
473   0399   CD 03C6                   CALL   STATUS
474
475   039C   3E 8E                     MVI    A,142    ; Tegn 2. afsnit af funktionen.
476   039E   D3 A5                     OUT    DELTAX
477   03A0   3E 50                     MVI    A,80
478   03A2   D3 A7                     OUT    DELTAY
479   03A4   3E 11                     MVI    A,17
480   03A6   D3 A0                     OUT    CMD
481   03A8   CD 03C6                   CALL   STATUS
482
483   03AB   3E 50                     MVI    A,80     ; Tegn 3. afsnit af funktionen.
484   03AD   D3 A5                     OUT    DELTAX
485   03AF   3E 0E                     MVI    A,14
486   03B1   D3 A7                     OUT    DELTAY
487   03B3   3E 15                     MVI    A,21
488   03B5   D3 A0                     OUT    CMD
489   03B7   CD 03C6                   CALL   STATUS
490   03BA   C9                        RET
491
492
493                          ;;;;;;; VIDEO, skifter til video, når en tast aktiveres ;;;;;;
494
495   03BB   DB CB             VIDEO:  IN     0CBH     ; Afsøg tastaturet indtil
496   03BD   FE FF                     CPI    0FFH     ; en vilkårlig tast er aktiv.
497   03BF   CA 03BB                   JZ     VIDEO
498   03C2   AF                        XRA    A        ; Skift til video.
499   03C3   D3 B0                     OUT    176
500   03C5   C9                        RET
501
502
503                          ;;;;;;; STATUS, læser GP's statusregister indtil GP er klar ;;;
504
```

```
  505    03C6   DB A0              STATUS: IN      160     ; Læs GP's statusregister
  506    03C8   E6 04                      ANI     04H     ; Lad kun bit 2 passere.
  507    03CA   FE 04                      CPI     04H
  508    03CC   C2 03C6                    JNZ     STATUS  ; Gentag aflæsning indtil bit 2 = 1.
  509    03CF   C9                         RET
  510
  511
  512                                ;;;;;;; TABEL, ASCII karakterer til tekst ;;;;;;;;;;;;;;;;;;;;;;
  513
  514    03D0                              ASEG
  515                                      ORG     0900H
  516
  517    0900   4B 6F 6F 72         TABEL:  DB 4BH,6FH,6FH,72H,64H,69H,6EH,61H,74H
  518    0904   64 69 6E 61
  519    0908   74
  520    0909   73 79 73 74               DB 73H,79H,73H,74H,65H,6DH
  521    090D   65 6D
  522
  523                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
  524
  525                                      END

Macros:

Symbols:
00A0    CMD             00A3    CSIZE           00A1    CTRL1
00A2    CTRL2           00A5    DELTAX          00A7    DELTAY
031C    FUNK1           033B    FUNK2           037C    FUNK3
02C0    FX              0310    GENTAG          01CC    REP1
0206    REP2            1000    STACK           0100    START
03C6    STATUS          0900    TABEL           02F4    TEKST
01EF    UNTIL1          0229    UNTIL2          03BB    VIDEO
02A1    X               00A8    XHIGH           00A9    XLOW
019D    XPIL            01BB    XPUNKT          022F    XTAL
012D    XYAKS           00AA    YHIGH           00AB    YLOW
017F    YPIL            01F5    YPUNKT          0266    YTAL


No Fatal error(s)

C
```

```
1
2                                .Z80
3                                .COMMENT*
4                                                    IEC.COM
5                                 ------------------------------------------------
6                                !                                                !
7                                ! Programmet sender/modtager koder fra et instrument !
8                                !  tilkoblet IEC 625 BUS'en.                     !
9                                !                                                !
10                               ! Instrumentadr. indtastes (0-31H)               !
11                               !                                                !
12                               ! SENDE MODE aktiveres ved at taste (S)          !
13                               !  Data indtastes nu og afsluttes med: (CNT)C (RET) !
14                               !                                                !
15                               ! RECEIVE MODE aktiveres ved at taste (R)        !
16                               !  Der indtastes ordre til instrument, om hvad det !
17                               !  skal sende, som afsluttes med: (RET)          !
18                               !  Programmet modtager nu data fra instrumentet  !
19                               !  som skrives ud på skærmen.                    !
20                               !                                                !
21                               ! EXIT til CP/M aktiveres ved at taste (E)       !
22                               !                                                !
23                                ------------------------------------------------
24                               *
25
26
27   0010                          BOUT    EQU     10H     ;Byte out reg. (W)
28   0010                          DIN     EQU     10H     ;Data in reg. (R)
29   0011                          INTM1   EQU     11H     ;Interupt maske reg. 1 (W)
30   0011                          INTS1   EQU     11H     ;Interupt status reg. 1 (R)
31   0012                          INTM2   EQU     12H     ;Interupt maske reg. 2 (W)
32   0014                          ADRM    EQU     14H     ;Adr. mode reg. (W)
33   0015                          AUXM    EQU     15H     ;Aux.mode reg. (W) C.com.
34   0016                          ADR01   EQU     16H     ;Adr. select reg. 0/1 (W)
35   0017                          EOS     EQU     17H     ;End of string meddelelse (W)
36   0005                          BDOS    EQU     0005H   ;Indgang til CP/M rutiner
37   0600                          BUFFER  EQU     0600H   ;Buffer for data til/fra IEC-BUS
38   05FD                          ANTBUF  EQU     05FDH   ;Buffer for antal byte hentet fra IEC-BUS
39   05FF                          IADRBUF EQU     05FFH   ;Buffer for instrument adr. tilkoblet
40                                                           ; IEC-BUS
41
42
43   0000'                         ASEG
44                                 ORG     100H
45
46
47                                 ;;;;;;;;;;  Rutine henter instrumentadr. fra keyboard  ;;;;;;;;;
48
49   0100   CD 02D3                INSTADR:CALL    CLRCRT          ;Clear skærm
50   0103   11 0400                       LD      DE,TABEL1       ;Tabeladr. til DE
51   0106   0E 09                        LD      C,09H           ;Adr. på PRINT STRING
52   0108   CD 0005                      CALL    BDOS
53   010B   21 0600                      LD      HL,BUFFER       ;BUFFER -) pointer
54   010E   36 03                        LD      (HL),03H        ;Max antal byte i streng = 3
55   0110   0E 0A                        LD      C,0AH           ;Adr. på READ CONSOLE BUFFER
56   0112   11 0600                      LD      DE,BUFFER
```

```
 57    0115   CD 0005              CALL   BDOS·          ;Hent karakterer fra key
 58    0118   21 0602              LD     HL,BUFFER+2    ;Læs første ASCII
 59    011B   7E                   LD     A,(HL)         ;Konverter til HEX nip.
 60    011C   E6 0F                AND    0FH
 61    011E   CB 07                RLC    A
 62    0120   CB 07                RLC    A
 63    0122   CB 07                RLC    A
 64    0124   CB 07                RLC    A
 65    0126   47                   LD     B,A            ;Gem nip. i B reg.
 66    0127   2C                   INC    L              ;Peg på næste ASCII
 67    0128   7E                   LD     A,(HL)         ;Læs næste ASCII
 68    0129   E6 0F                AND    0FH            ;Konverter til HEX nip.
 69    012B   B0                   OR     B              ;Sammensæt instr.adr.
 70    012C   21 05FF              LD     HL,IADRBUF     ;Gem adr. i instr.adr.buffer
 71    012F   77                   LD     (HL),A
 72    0130   C3 0133              JP     VAELG          ;Hop til udvælgelse
 73
 74
 75                        ;;;;;;;;;;  Udvælgelsesrutine: SEND / RECEIVE / EXIT  ;;;;;;;;;;
 76
 77    0133   CD 02D3      VAELG:  CALL   CLRCRT         ;Clear skærm
 78    0136   11 046E              LD     DE,TABEL2      ;Sender meddelelse til skærm
 79    0139   CD 0005              CALL   BDOS
 80    013C   0E 01        KEY:    LD     C,01H          ;Function 1 i BDOS (Console Input)
 81    013E   CD 0005              CALL   BDOS
 82    0141   FE 53                CP     53H            ;Er der tastet: S ?
 83    0143   CA 017D              JP     Z,HSEND        ;Hop til hovedrutine for se.mode
 84    0146   FE 52                CP     52H            ;Er der tastet: R ?
 85    0148   CA 0153              JP     Z,HRECIV       ;Hop til hovedrutine for re.mode
 86    014B   FE 45                CP     45H            ;Er der tastet: E ?
 87    014D   CA 0310              JP     Z,EXIT         ;Hop til EXIT
 88    0150   C3 013C              JP     KEY            ;Nej læs igen
 89
 90
 91                        ;;;;;;;;;;;;;;;;  Hovedrutine for receive mode  ;;;;;;;;;;;;;;;;
 92
 93    0153   CD 0198      HRECIV: CALL   INIT           ;Initialiser 7210
 94    0156   CD 01BF              CALL   REN            ;Set instrument i remote
 95    0159   CD 01C8              CALL   IFC            ;Send Interf. Clear til instrument
 96    015C   CD 01D8              CALL   SLTADR         ;Send TAD=Contr. / LAD=instrument
 97    015F   CD 0215              CALL   DEL            ;Def. del.=INS  (US) til instrument
 98    0162   CD 0244              CALL   SORDRE         ;Ordre til instrument om sende
 99    0165   CD 0256              CALL   RLTADR         ;Send TAD=Instrument / LAD=Contr.
100    0168   CD 027F              CALL   DRECIV         ;Modtag data fra IEC-BUS og send
101                                                      ;til databuffer
102    016B   CD 02D3              CALL   CLRCRT         ;Clear skærm
103    016E   11 04D5              LD     DE,TABEL3      ;Send meddelelse til skærm
104    0171   CD 0005              CALL   BDOS
105    0174   CD 02A5              CALL   CRTBUF         ;Udskriv modtaget streng til skærm
106    0177   CD 01C8              CALL   IFC            ;Clear Interface
107    017A   C3 0133              JP     VAELG          ;Hop til udvælgelsesrutine
108
109
110                        ;;;;;;;;;;;;;;;;;;  Hovedrutine for sendemode  ;;;;;;;;;;;;;;;;;;
111
112    017D   CD 02D3      HSEND:  CALL   CLRCRT         ;Clear skærm
```

```
113   0180   11 0524                LD     DE,TABEL4      ;Send meddelelse til skærm
114   0183   CD 0005                CALL   BDOS
115   0186   CD 0198                CALL   INIT           ;Initialiser 7210
116   0189   CD 01BF                CALL   REN            ;Set instrument i remote
117   018C   CD 01C8                CALL   IFC            ;Send Interf. Clear til instrument
118   018F   CD 01D8                CALL   SLTADR         ;Send TAD=Contr./LAD=instrument
119   0192   CD 01F5                CALL   RDBUF          ;Læs keystreng og send til instrument
120   0195   C3 0133                JP     VAELG          ;Hop til udvælgelsesrutine
121
122
123                          ;;;;;;;;;;;;;;;;;;  Initialisering af 7210  ;;;;;;;;;;;;;;;;;;
124
125   0198   3E 02          INIT:  LD     A,02H          ;Chip reset 7210
126   019A   D3 15                 OUT    (AUXM),A
127
128   019C   3E 00                 LD     A,00H          ;Disable Interupt 1
129   019E   D3 11                 OUT    (INTM1),A
130   01A0   D3 12                 OUT    (INTM2),A      ;Disable Interupt 2
131
132   01A2   3E 31                 LD     A,31H          ;Adr.mode 1 / T/R-mode 3
133   01A4   D3 14                 OUT    (ADRM),A
134
135   01A6   3E 00                 LD     A,00H          ;My adr.=0 -) adr.reg.0
136   01A8   D3 16                 OUT    (ADR01),A
137
138   01AA   3E E0                 LD     A,0E0H         ;Disable adr.reg.1
139   01AC   D3 16                 OUT    (ADR01),A
140
141   01AE   3E 03                 LD     A,03H          ;ETX = EOS (^C)
142   01B0   D3 17                 OUT    (EOS),A        ;NB: INSTRUMENT AFHÆNGIG
143
144   01B2   3E A4                 LD     A,0A4H         ;High speed trans. INT=akt.Hi
145   01B4   D3 15                 OUT    (AUXM),A
146
147   01B6   3E 24                 LD     A,24H          ;4 MHz clock
148   01B8   D3 15                 OUT    (AUXM),A
149
150   01BA   3E 8C                 LD     A,8CH          ;Normal handshake mode
151   01BC   D3 15                 OUT    (AUXM),A
152
153   01BE   C9                    RET
154
155
156                          ;;;;;;;;;;;;  Comand senderutine, sætter instrument i remote  ;;;;;;;;
157
158   01BF   3E 00           REN:   LD     A,00H          ;PON reset af 7210
159   01C1   D3 15                 OUT    (AUXM),A
160
161   01C3   3E 1F                 LD     A,1FH          ;Instrument i remote
162   01C5   D3 15                 OUT    (AUXM),A
163
164   01C7   C9                    RET
165
166
167                          ;;;;;;;;;;;;;;;;;  Rutine sender IFC til instrument  ;;;;;;;;;;;;;;;;;
168
```

```
169   01C8   3E 00          IFC:   LD     A,00H          ;PON til 7210
170   01CA   D3 15                 OUT    (AUXM),A
171
172   01CC   3E 1E                 LD     A,1EH          ;Send IFC til instrument
173   01CE   D3 15                 OUT    (AUXM),A
174
175   01D0   CD 02E6               CALL   DELAY          ;Vent
176
177   01D3   3E 16                 LD     A,16H          ;Reset IFC
178   01D5   D3 15                 OUT    (AUXM),A
179
180   01D7   C9                    RET
181
182
183                        ;;;;;;;;;;;;;;;  LAD/TAD - rutine for sendemode  ;;;;;;;;;;;;;;;
184
185   01D8   CD 02EC       SLTADR: CALL   COCHECK        ;Klar til com. out ?
186
187   01DB   3E 3F                 LD     A,3FH          ;Send UNL
188   01DD   D3 10                 OUT    (BOUT),A
189
190   01DF   CD 02EC               CALL   COCHECK        ;Klar til com.out ?
191
192   01E2   21 05FF               LD     HL,IADRBUF     ;Instrumentadr. til A
193   01E5   7E                    LD     A,(HL)         ;Hent instr.adr.=LAD
194   01E6   F6 20                 OR     20H            ;Form LAD
195   01E8   D3 10                 OUT    (BOUT),A       ;Send instr.adr.=LAD
196
197   01EA   CD 02EC               CALL   COCHECK
198
199   01ED   3E 40                 LD     A,40H          ;Send contr. adr.=TAD
200   01EF   D3 10                 OUT    (BOUT),A
201
202   01F1   CD 02EC               CALL   COCHECK
203
204   01F4   C9                    RET
205
206
207
208                        ; Læse/senderutine,læser fra keyboard og sender til instrument ;
209
210   01F5   3E 10         RDBUF:  LD     A,10H          ;7210 stand by (Data mode)
211   01F7   D3 15                 OUT    (AUXM),A
212   01F9   11 0600               LD     DE,BUFFER      ;Buffer adr. til DE
213   01FC   0E 0A                 LD     C,0AH          ;Start adr. på "READ CNOSOL"
214   01FE   21 0600               LD     HL,BUFFER      ;BUFFER til HL
215   0201   36 FF                 LD     (HL),0FFH      ;Max antal Byte til streng
216   0203   CD 0005               CALL   BDOS           ;Call READ CONSOLE BUFFER
217   0206   21 0601               LD     HL,BUFFER+1    ;Antal byte i streng til C
218   0209   4E                    LD     C,(HL)
219   020A   23           LOOP:   INC    HL             ;Næste byte
220   020B   CD 02F3               CALL   DOCHECK        ;Klar til data out ?
221   020E   7E                    LD     A,(HL)         ;Byte til A reg.
222   020F   D3 10                 OUT    (BOUT),A       ;Send byte til IEC-BUS
223   0211   0D                    DEC    C              ;Resterende antal byte - 1
224   0212   20 F6                 JR     NZ,LOOP        ;Streng slut ?
```

```
225   0214   C9                    RET                 ;Ja retur
226
227
228
229                        ;;;;;;;;;;;;;;;  Def. ny del. til instrument (INS)  ;;;;;;;;;;;;;;;
230
231                                  ;NB: DETTE MODUL ER INSTRUMENTAFHÆNGIG !
232
233   0215   3E 10          DEL:    LD      A,10H       ;7210 Stand by (datamode)
234   0217   D3 15                  OUT     (AUXM),A
235
236   0219   CD 02F3                CALL    DOCHECK     ;Klar til data out ?
237
238   021C   3E 1B                  LD      A,1BH
239   021E   D3 10                  OUT     (BOUT),A    ;Send: ESC -) instrument
240
241   0220   CD 02F3                CALL    DOCHECK
242   0223   3E 30                  LD      A,30H
243   0225   D3 10                  OUT     (BOUT),A    ;Send: 0 -) instrument
244
245   0227   CD 02F3                CALL    DOCHECK
246   022A   3E 44                  LD      A,44H
247   022C   D3 10                  OUT     (BOUT),A    ;Send: D -) instrument
248
249   022E   CD 02F3                CALL    DOCHECK
250   0231   3E 44                  LD      A,44H
251   0233   D3 10                  OUT     (BOUT),A    ;Send: D -) instrument
252
253   0235   CD 02F3                CALL    DOCHECK
254   0238   3E 30                  LD      A,30H
255   023A   D3 10                  OUT     (BOUT),A    ;Send: 0 -) instrument
256
257   023C   CD 02F3                CALL    DOCHECK
258   023F   3E 1F                  LD      A,1FH
259   0241   D3 10                  OUT     (BOUT),A    ;Send: (INS) -) instrument
260
261   0243   C9                    RET
262
263
264                        ;;;;;;;;;;;  Ordre til instrument om at sende data  ;;;;;;;;;;;;
265
266   0244   CD 02D3       SORDRE: CALL    CLRCRT      ;Clear skærm
267   0247   0E 09                  LD      C,09H       ;Adr. på PRINT STRING
268   0249   11 057A                LD      DE,TABEL5   ;Tabeladr. til DE
269   024C   CD 0005                CALL    BDOS        ;Skriv tekst på skærm
270   024F   CD 01F5                CALL    RDBUF       ;Læs keystreng og send til instr.
271   0252   CD 02F3                CALL    DOCHECK     ;Klar ?
272   0255   C9                    RET
273
274
275                        ;;;;;;;;;;;;;;;  LAD/TAD rutine for recive mode  ;;;;;;;;;;;;;;;
276
277   0256   3E 11          RLTADR: LD      A,11H       ;Take contr. async. -) 7210
278   0258   D3 15                  OUT     (AUXM),A
279
280   025A   CD 02EC                CALL    COCHECK     ;Klar til CO out ?
```

```
281
282   025D   3E 3F                    LD      A,3FH           ;UNL til instrument
283   025F   D3 10                    OUT     (BOUT),A
284
285   0261   CD 02EC                  CALL    CDCHECK
286
287   0264   21 05FF                  LD      HL,IADRBUF      ;Instrumentadr. til A
288   0267   7E                       LD      A,(HL)          ;Hent instr.adr.=TAD
289   0268   F6 40                    OR      40H             ;Form TAD
290   026A   D3 10                    OUT     (BOUT),A        ;Send TAD
291
292   026C   CD 02EC                  CALL    CDCHECK
293
294   026F   3E 20                    LD      A,20H           ;Send contr.adr.=LAD
295   0271   D3 10                    OUT     (BOUT),A
296
297   0273   CD 02EC                  CALL    CDCHECK
298
299   0276   3E 13                    LD      A,13H           ;7210 ... listen
300   0278   D3 15                    OUT     (AUXM),A
301
302   027A   3E 1A                    LD      A,1AH           ;Take contr. synk. on end
303   027C   D3 15                    OUT     (AUXM),A
304
305   027E   C9                       RET
306
307
308                          ;;;;;;;;;;;;;  Modtagerutine fra IEC-BUS til buffer  ;;;;;;;;;;;
309
310   027F   3E 10           DRECIV:  LD      A,10H           ;Go to stand by (data mode)
311   0281   D3 15                    OUT     (AUXM),A
312
313   0283   21 0600                  LD      HL,BUFFER       ;Bufferadr. til HL
314   0286   16 00                    LD      D,00H           ;Clear D reg.
315   0288   0E FF           DDD:     LD      C,0FFH          ;Antal "ventetid" til C reg.
316   028A   06 FF           AAA:     LD      B,0FFH          ;Ventetid til B reg.
317
318   028C   DB 11           BBB:     IN      A,(INTS1)       ;Læs interrupt status 1
319   028E   CB 47                    BIT     0,A             ;Klar til data ind ? (DI)
320   0290   20 0C                    JR      NZ,CCC          ;Ja - hent data
321
322   0292   10 F8                    DJNZ    BBB             ;Nej - vent
323   0294   0D                       DEC     C
324   0295   20 F3                    JR      NZ,AAA
325   0297   21 05FD                  LD      HL,ANTBUF       ;Gem antal byte modtaget
326   029A   72                       LD      (HL),D          ;-fra IEC-BUS i ANTBUF
327   029B   23                       INC     HL
328   029C   73                       LD      (HL),E
329   029D   C9                       RET                     ;Tid udløbet - retur
330
331   029E   DB 10           CCC:     IN      A,(DIN)         ;Ja -) læs data fra IEC-BUS
332   02A0   77                       LD      (HL),A          ;Gem værdi i BUFFER
333   02A1   23                       INC     HL              ;Næste plads
334   02A2   13                       INC     DE              ;Resterende antal byte -1
335   02A3   18 E3                    JR      DDD             ;Hent næste Byte
336
```

```
337
338                              ;;;;;;;;;;;  Rutine sender "BUFFER"-streng til Skærm  ;;;;;;;;;;
339
340     02A5    21 05FD          CRTBUF: LD      HL,ANTBUF       ;Hent antal byte modtaget
341     02A8    56                       LD      D,(HL)          ;Ant.byte -) DE
342     02A9    23                       INC     HL
343     02AA    5E                       LD      E,(HL)
344     02AB    21 0600                  LD      HL,BUFFER       ;Bufferadr. til HL
345     02AE    7A                       LD      A,D             ;Hent antal ASCII
346     02AF    FE 00                    CP      00H             ;Antal = 0 ?
347     02B1    20 06                    JR      NZ,CBUF
348     02B3    7B                       LD      A,E
349     02B4    FE 00                    CP      00H
350     02B6    20 01                    JR      NZ,CBUF
351     02B8    C9                       RET                     ;Ja -retur
352     02B9    0E 02            CBUF:   LD      C,02H           ;Adr. på CONSOLE OUTPUT rutine
353     02BB    D5                       PUSH    DE
354     02BC    5E                       LD      E,(HL)
355     02BD    E5                       PUSH    HL
356     02BE    CD 0005                  CALL    BDOS            ;Skriv karakter på skærm
357     02C1    E1                       POP     HL
358     02C2    23                       INC     HL              ;Peg på næste ASCII
359     02C3    D1                       POP     DE
360     02C4    1B                       DEC     DE              ;Antal -1
361     02C5    7A                       LD      A,D             ;Færdig ?
362     02C6    B3                       OR      E
363     02C7    20 F0                    JR      NZ,CBUF         ;Nej -hent næste
364     02C9    0E 01            CR:     LD      C,01H           ;Adr. på CONSOLE INPUT rutine
365     02CB    CD 0005                  CALL    BDOS
366     02CE    FE 0D                    CP      0DH             ;Er der tastet (CR)
367     02D0    C8                       RET     Z               ;Ja retur
368     02D1    18 F6                    JR      CR              ;Nej igen
369
370
371                              ;;;;;;;;;;;;;;;;;;;  Rutine clear'er skærm  ;;;;;;;;;;;;;;;;;;;
372
373     02D3    0E 09            CLRCRT: LD      C,09H           ;Adr. på PRINT STRING rutine
374     02D5    21 F000                  LD      HL,0F000H       ;Start adr. på skærm RAM
375     02D8    36 00            LOOPCL: LD      (HL),00H        ;Clear skærm (SP)
376     02DA    23                       INC     HL              ;Næste plads
377     02DB    7C                       LD      A,H             ;Skærm slut ?  (F780)
378     02DC    FE F7                    CP      0F7H
379     02DE    20 F8                    JR      NZ,LOOPCL       ;Nej -igen
380     02E0    7D                       LD      A,L
381     02E1    FE 80                    CP      80H
382     02E3    20 F3                    JR      NZ,LOOPCL
383     02E5    C9                       RET                     ;Ja -retur
384
385
386                              ;;;;;;;;;;;;;;;;;;;;;;;;  Delayrutine  ;;;;;;;;;;;;;;;;;;;;;;;;
387
388     02E6    3E FF            DELAY:  LD      A,0FFH          ;Delay ca. 0,1mSek
389     02E8    3D                       DEC     A
390     02E9    20 FD                    JR      NZ,(DELAY+2)
391     02EB    C9                       RET
392
```

```
393
394                                ;;;;;;;;;;;;;;  Control out check-rutine  ;;;;;;;;;;;;;;;;;;;;;;;;
395
396     02EC    DB 12              COCHECK:IN      A,(INTM2)       ;Klar til at sende (CO)
397     02EE    CB 5F                   BIT     3,A
398     02F0    28 FA                   JR      Z,(COCHECK)     ;Nej -vent
399     02F2    C9                      RET                     ;Ja -retur
400
401
402
403                                ;;;;;;;;;;;;;;;;;;;;  Data out check rutine  ;;;;;;;;;;;;;;;;;;;;
404
405     02F3    DB 11              DOCHECK:IN      A,(INTM1)       ;Læs INTM1
406     02F5    CB 4F                   BIT     1,A             ;Klar til at sende ? (DO)
407     02F7    28 FA                   JR      Z,DOCHECK       ;Nej -vent
408     02F9    C9                      RET                     ;Ja -retur
409
410
411                                ;;;;;  Rutine sender Selektiv Device Clear til instrument  ;;;;;
412
413     02FA    3E 11              SDC:    LD      A,11H           ;7210 take control async.
414     02FC    D3 15                   OUT     (AUXM),A
415     02FE    CD 01D8                 CALL    SLTADR          ;Send LAD = instr./ TAD = contr.
416     0301    3E 04                   LD      A,04H           ;Send SDC til instrument
417     0303    D3 10                   OUT     (BOUT),A
418     0305    CD 02EC                 CALL    COCHECK         ;Klar til comand out ?
419     0308    3E 03                   LD      A,03H           ;Send (ETX) del.
420     030A    D3 10                   OUT     (BOUT),A
421     030C    CD 02EC                 CALL    COCHECK         ;Klar ?
422     030F    C9                      RET
423
424
425                                ;;;;;;;;;;;;;;  Afslutter program og går til CP/M  ;;;;;;;;;;;;;;
426
427     0310    CD 02FA            EXIT:   CALL    SDC             ;Send Sel.Div.Clear
428     0313    CD 02D3                 CALL    CLRCRT          ;Clear skærm
429     0316    C3 0000                 JP      0000H           ;Hop til CP/M
430
431
432                                ;;;;;;;;;;;;;;  Tabel for meddelelser til skærm  ;;;;;;;;;;;;;;
433
434                                ORG     400H
435
436     0400    0D                 TABEL1: DEFB 0DH
437     0401    20 20 20 20             DEFM "      IEC 625 SENDE/MODTAGE RUTINE:"
438     0405    20 20 49 45
439     0409    43 20 36 32
440     040D    35 20 53 45
441     0411    4E 44 45 2F
442     0415    4D 4F 44 54
443     0419    41 47 45 20
444     041D    52 55 54 49
445     0421    4E 45 3A
446     0424    0A 0A 0D                DEFB 0AH,0AH,0DH
447     0427    20 20 20 20             DEFM "      INDTAST INSTRUMENTETS KALDE-ADR."
448     042B    20 20 49 4E
```

```
449   042F   44 54 41 53
450   0433   54 20 49 4E
451   0437   53 54 52 55
452   043B   4D 45 4E 54
453   043F   45 54 53 20
454   0443   4B 41 4C 44
455   0447   45 2D 41 44
456   044B   52 2E
457   044D   0A 0A 0D                  DEFB 0AH,0AH,0DH
458   0450   20 20 20 20               DEFM "      AFSLUT MED: (RET)"
459   0454   20 20 41 46
460   0458   53 4C 55 54
461   045C   20 4D 45 44
462   0460   3A 20 3C 52
463   0464   45 54 3E
464   0467   0A 0A 0A 0A               DEFB 0AH,0AH,0AH,0AH,07H,0DH,24H
465   046B   07 0D 24
466
467   046E   0D                  TABEL2: DEFB 0DH
468   046F   20 20 20 20               DEFM "      VÆLG: SEND / RECEIVE / EXIT"
469   0473   20 20 56 5B
470   0477   4C 47 3A 20
471   047B   53 45 4E 44
472   047F   20 2F 20 52
473   0483   45 43 45 49
474   0487   56 45 20 2F
475   048B   20 45 58 49
476   048F   54
477   0490   0A 0A 0D                  DEFB 0AH,0AH,0DH
478   0493   20 20 20 20               DEFM "      S = SEND MODE"
479   0497   20 20 53 20
480   049B   3D 20 53 45
481   049F   4E 44 20 4D
482   04A3   4F 44 45
483   04A6   0A 0D                     DEFB 0AH,0DH
484   04A8   20 20 20 20               DEFM "      R = RECEIVE MODE"
485   04AC   20 20 52 20
486   04B0   3D 20 52 45
487   04B4   43 45 49 56
488   04B8   45 20 4D 4F
489   04BC   44 45
490   04BE   0A 0D                     DEFB 0AH,0DH
491   04C0   20 20 20 20               DEFM "      E = EXIT"
492   04C4   20 20 45 20
493   04C8   3D 20 45 58
494   04CC   49 54
495   04CE   0A 0A 0A 0A               DEFB 0AH,0AH,0AH,0AH,07H,0DH,24H
496   04D2   07 0D 24
497
498   04D5   0D                  TABEL3: DEFB 0DH
499   04D6   20 20 20 20               DEFM "      RECEIVE MODE:"
500   04DA   20 20 52 45
501   04DE   43 45 49 56
502   04E2   45 20 4D 4F
503   04E6   44 45 3A
504   04E9   0A 0A 0D                  DEFB 0AH,0AH,0DH
```

```
505     04EC    20 20 20 20             DEFM "      AFSLUT MED (RET)"
506     04F0    20 20 41 46
507     04F4    53 4C 55 54
508     04F8    20 4D 45 44
509     04FC    20 3C 52 45
510     0500    54 3E
511     0502    0A 0A 0D                DEFB 0AH,0AH,0DH
512     0505    20 20 20 20             DEFM "      DATA FRA IEC-BUS:"
513     0509    20 20 44 41
514     050D    54 41 20 46
515     0511    52 41 20 49
516     0515    45 43 2D 42
517     0519    55 53 3A
518     051C    0A 0A 0A 0A             DEFB 0AH,0AH,0AH,0AH,07H,0DH,24H
519     0520    0A 07 0D 24
520
521     0524    0D              TABEL4: DEFB 0DH
522     0525    20 20 20 20             DEFM "      SEND MODE:"
523     0529    20 20 53 45
524     052D    4E 44 20 4D
525     0531    4F 44 45 3A
526     0535    0A 0A 0D                DEFB 0AH,0AH,0DH
527     0538    20 20 20 20             DEFM "      INDTAST DATASTRENG"
528     053C    20 20 49 4E
529     0540    44 54 41 53
530     0544    54 20 44 41
531     0548    54 41 53 54
532     054C    52 45 4E 47
533     0550    0A 0A 0D                DEFB 0AH,0AH,0DH
534     0553    20 20 20 20             DEFM "      AFSLUT MED: (CNT) C  (RET)"
535     0557    20 20 41 46
536     055B    53 4C 55 54
537     055F    20 4D 45 44
538     0563    3A 20 3C 43
539     0567    4E 54 3E 20
540     056B    43 20 20 3C
541     056F    52 45 54 3E
542     0573    0A 0A 0A 0A             DEFB 0AH,0AH,0AH,0AH,07H,0DH,24H
543     0577    07 0D 24
544
545     057A    0D              TABEL5: DEFB 0DH
546     057B    20 20 20 20             DEFM "      RECEIVE MODE:"
547     057F    20 20 52 45
548     0583    43 45 49 56
549     0587    45 20 4D 4F
550     058B    44 45 3A
551     058E    0A 0A 0D                DEFB 0AH,0AH,0DH
552     0591    20 20 20 20             DEFM "      INDTAST ORDRE TIL INSTRUMENT OM"
553     0595    20 20 49 4E
554     0599    44 54 41 53
555     059D    54 20 4F 52
556     05A1    44 52 45 20
557     05A5    54 49 4C 20
558     05A9    49 4E 53 54
559     05AD    52 55 4D 45
560     05B1    4E 54 20 4F
```

```
 561    05B5   4D
 562    05B6   0A 0A 0D                    DEFB 0AH,0AH,0DH
 563    05B9   20 20 20 20                 DEFM "      AT SENDE DATA, AFSLUT MED: (RET)"
 564    05BD   20 20 41 54
 565    05C1   20 53 45 4E
 566    05C5   44 45 20 44
 567    05C9   41 54 41 2C
 568    05CD   20 41 46 53
 569    05D1   4C 55 54 20
 570    05D5   4D 45 44 3A
 571    05D9   20 3C 52 45
 572    05DD   54 3E
 573    05DF   0A 0A 0A 0A                 DEFB 0AH,0AH,0AH,0AH,07H,0DH,24H
 574    05E3   07 0D 24
 575
 576
 577
 578                              END
```

```
MACRO-80 3.43  27-Jul-81      PAGE    S

Macros:

Symbols:
028A    AAA             0016    ADR01           0014    ADRM
05FD    ANTBUF          0015    AUXM            028C    BBB
0005    BDOS            0010    BOUT            0600    BUFFER
02B9    CBUF            029E    CCC             02D3    CLRCRT
02EC    COCHECK         02C9    CR              02A5    CRTBUF
0288    DDD             0215    DEL             02E6    DELAY
0010    DIN             02F3    DOCHECK         027F    DRECIV
0017    EOS             0310    EXIT            0153    HRECIV
017D    HSEND           05FF    IADRBUF         01C8    IFC
0198    INIT            0100    INSTADR         0011    INTM1
0012    INTM2           0011    INTS1           013C    KEY
020A    LOOP            02D8    LOOPCL          01F5    RDBUF
01BF    REN             0256    RLTADR          02FA    SDC
01D8    SLTADR          0244    SORDRE          0400    TABEL1
046E    TABEL2          04D5    TABEL3          0524    TABEL4
057A    TABEL5          0133    VAELG

No Fatal error(s)

C
```

KOMPENDIE TIL EFTERUDDANNELSKURSUS

ME 5821

MICROCOMPUTER INTERFACE-TEKNIK

INDGANGSMODUL

Type 601F2600 + 601F5100/601F5000

Indgangsmodulerne danner overgangen fra følere til den centrale del i styresystemet, computersiden.
Hvert modul er forsynet med 16 indgange og ved hjælp af optokobler sikres der en galvanisk adskillelse mellem computerside og processide.
Hver indgangskreds har tilsluttet sin egen lysdiode, der ved lys indikerer signal tilstede (højt spændingsniveau).

Indgangsmodulet er forberedt til et spændingsområde dækkende fra 15 volt til 60 volt jævn- eller vekselspænding, og gennem specielkonstruktion begrænses indgangsstrømmens størrelse til max. 20 mA ved overspænding.

Foruden overspændingsbeskyttelse er indgangene polaritetssikrede, det vil sige plus- og minusledere kan tilsluttes vilkårligt.

Sammen med dette kort skal anvendes klemrækkemodul 601F5100 ved spændinger fra 15-60V. Ved 220V AC/DC anvendes klemrækkemodul 601F5000.

DATA:

Indgangsspænding	:	15-60 volt AC/DC	50Hz
Indgangsstrøm, typisk	:	10-20 mA	
Isolationsspænding	:	2 KV	DC/AC P.P.
Antal indgange	:	16 stk.	
Støjfilter	:	10 m.sec.	

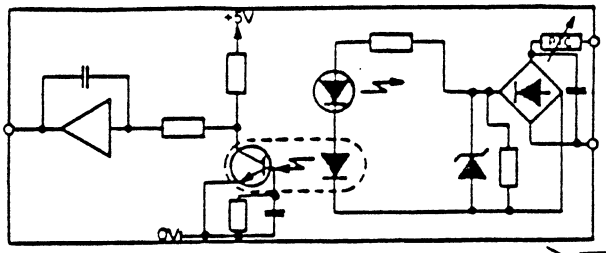
DIMENSIONER:

Printhøjde 233,4 mm. dobbelt Europa-modul
Printdybde 160 mm.
Modulbredde 20 mm.

Nesleo

DIGITAL INPUT 16 KANALER
601 F 2600

12-60V DC/AC



601 F 2600-01

601 F 2600

STIK 1

c1 +5V
a1
c32 0V DIGITAL
a32

c14 A15
a14 A14
c13 A13
a13 A12
c12 A11
a12 A10
c11 A9
a11 A8

ADRESSE-DEKODN.

A13 = 0
A12 = 1
A11 = 0
A10 = 0
A9 = 0
A8 = 0

c10 A7
a10 A6
c9 A5
a9 A4
c30 KA7
a30 KA6
c29 KA5
a29 KA4

COMPARATOR

SIGN.GR.	a29 KA4	c29 KA5	a30 KA6	c30 KA7
0-15	0	0	0	0
16-31	1	0	0	0
32-47	0	1	0	0
48-63	1	1	0	0
64-79	0	0	1	0
80-95	1	0	1	0
96-111	0	1	1	0
112-127	1	1	1	0

a28 $\overline{\text{DIS FUNC.}}$
c8 A3
a8 A2
c7 A1
a7 A0
a21 $\overline{\text{RD}}$
c21 $\overline{\text{WR}}$
c28 $\overline{\text{OUT EN}}$

FUNKTIONS-DEKODNING
A3=0 → SIGN 0-7
A3=1 → SIGN 8-15

A2	A1	A0	FUNKT.
0	0	0	IN/OUT K.
0	0	1	DATA
0	1	X	NO
1	X	X	NO

c31
a31
I/O-KODNING

c6 DB7
a6 DB6
c5 DB5
a5 DB4
c4 DB3
a4 DB2
c3 DB1
a3 DB0

DIR/EN

BI-DIRECTIONAL DRIVER

INPUT/OUTPUT KODE

INPUT:
DB0 = 0
DB1 = 1

8

(A3=1) INPUT LATCH SIGNAL 8-15

(A3=0) INPUT LATCH SIGNAL 0-7

8

STIK 2

16	d6	c6
	z6	a6
15	b6	c 7
	d 14	a 7
14	d8	c8
	z8	a8
13	b8	c 9
	b4	a 9
12	d10	c 10
	z10	a 10
11	b10	c 11
	z14	a 11
10	d12	c 12
	z12	a 12
9	b12	c 13
	b14	a 13
8	b24	c 25
	d24	a 25
7	d26	c 26
	z26	a 26
6	b26	c 27
	b22	a 27
5	d28	c 28
	z28	a 28
4	b28	c 29
	z24	a 29
3	d30	c 30
	z30	a 30
2	b30	c 31
	b32	a 31
1	d32	c 32
	z32	a 32

22 11 79	28 11 80	BLOKDIAGRAM
	NESELCO DANMARK	601 F 2605

2.7 UDGANGSMODULER JÆVNSTRØM

Type 601F2500 + 601F5100, 601F2550 + 601F4200

Udgangsmodulet tjener som overgang fra computersiden i F2002, og afgiver de udgangsspændinger, der aktiverer maskinens/processens aktuatorer (kontaktorer, relæer, magnet ventiler) og indikatorer (lamper, m.v.).

Hvert udgangsmodul er forsynet med 16 udgange, der ved hjælp af optokoblere sikrer, at logikkredse i det centrale styresystem adskilles galvanisk fra maskine/proces.
Hver udgang er tilsluttet en lysdiode, der ved lys indikerer aktiveret udgang.

Det er muligt at vælge mellem forskellige udgangsmoduler, alt efter behovet for størrelsen af strømstyrke.

Udgangsmodulerne for jævnspænding indeholder som udgangstrin en transistor for hver kanal. Transistoren fungerer desuden som sikring, og beskytter derfor systemet mod kortslutninger.

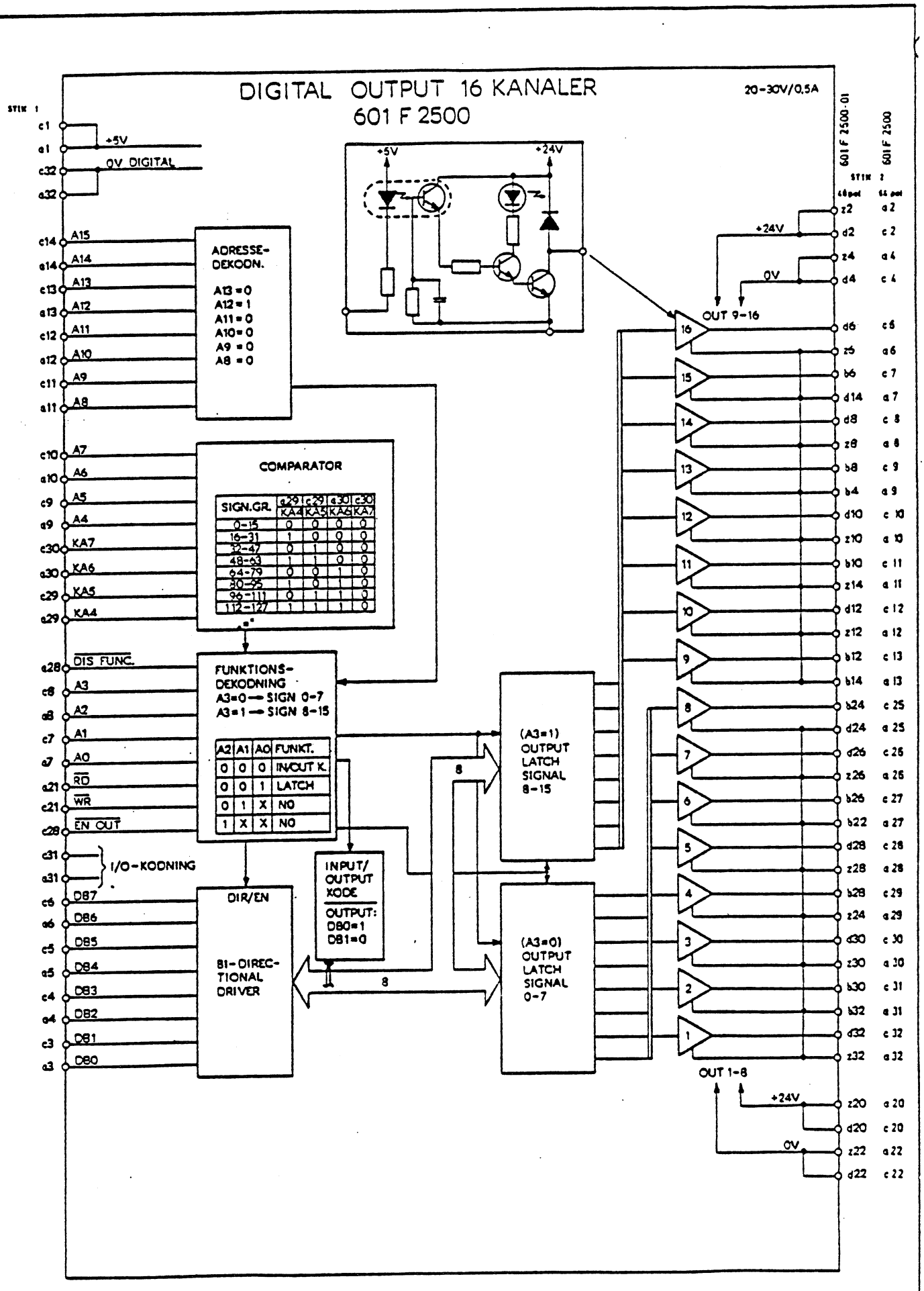
Udgangstransistoren bryder spændingsforsyningens nul.
Sammen med udgangsmodulerne anvendes klemrækkemodulerne 601F5100 ved 0,5A og 601F4200 ved 2A udgange.

DATA:

601F2500		
Udgangsspænding	:	24 volt DC +/-20%
Udgangsstrøm max.	:	0,5A ohmsk/induktiv
601F2550		
Udgangsspænding	:	24 volt DC +/- 20%
Udgangsstrøm max.	:	2A ohmsk/induktiv

DIMENSIONER:

601F2500		
Printhøjde	:	233,4 mm.
Printdybde	:	160 mm.
Modulbredde	:	20 mm.
601F2550:		
Printhøjde	:	233,4 mm.
Printdybde	:	160 mm.
Modulbredde	:	40 mm.



21 11 19	26 11 90	BLOKDIAGRAM	
GK			
SS	NERELCO	601 F 2505	

2.8 UDGANGSMODUL - VEKSELSTRØM

Type 601F2500 + 601F6400

Udgangsmodulet er et standard jævnstrømsmodul type 601F2500, hvortil der kan tilkobles et klemrækkemodul type 601F6400. Dette modul er monteret med 16 triac og den nødvendige elektronik for at kunne styre disse. Forbindelser mellem udgangsmodulet og triacmodulet foretages ved hjælp af et 64-leder fladkabel type 601F6500. Fladkablet er monteret med stik i begge ender.

Triacmodulet type 601F6400 forsyner udgangsmodulet type 601F2500 med den nødvendige eksterne spænding. Der er to interne hjælpe-strømforsyninger. Den ene er forbundet til fase.

NB: Ved tilslutning af 220V AC skal fase og nul tilsluttes de respektive klemmer.

Triacmodulet trækker til fase.

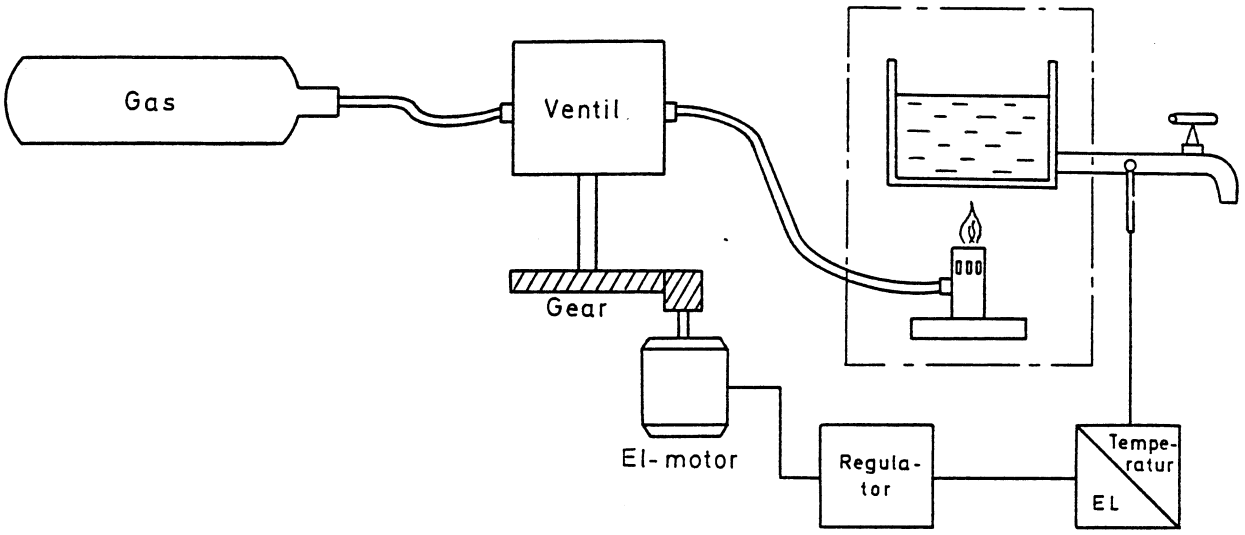
DATA:

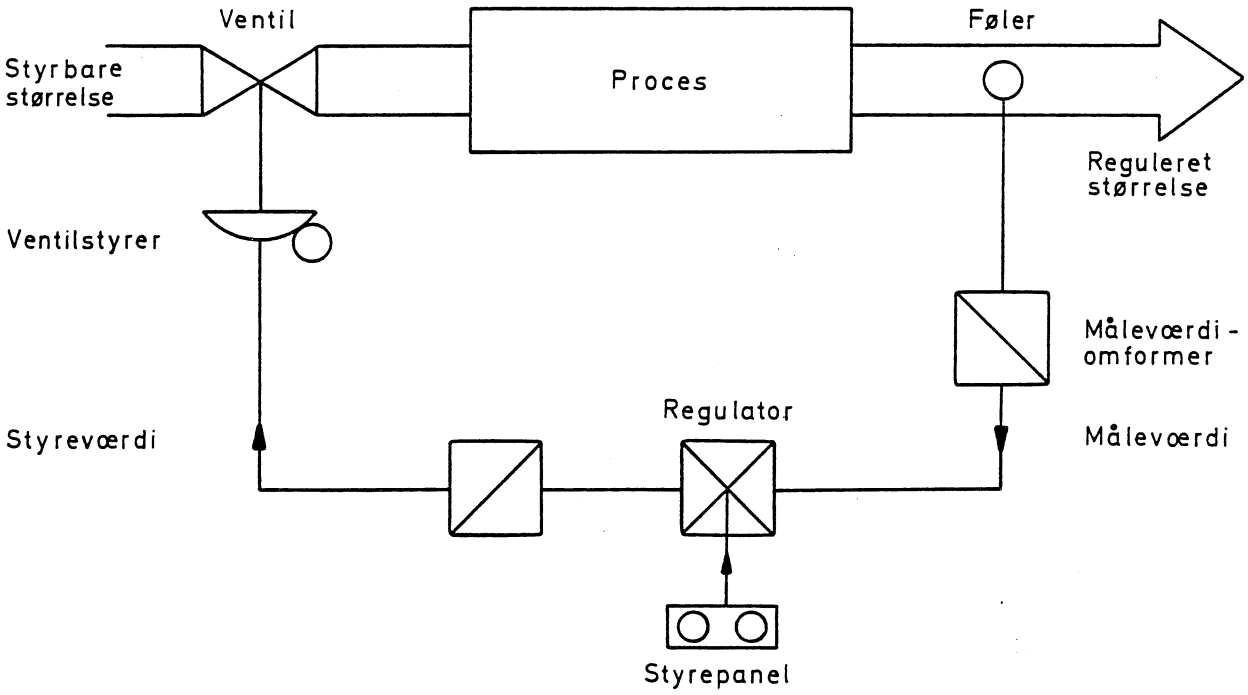
Forsyningsspænding:		220V AC +/-20%, 50Hz
Sikring	:	Max. 6A
Last/kanal	:	Max. 0,5A ohm ohmsk/induktiv

DIMENSIONER:

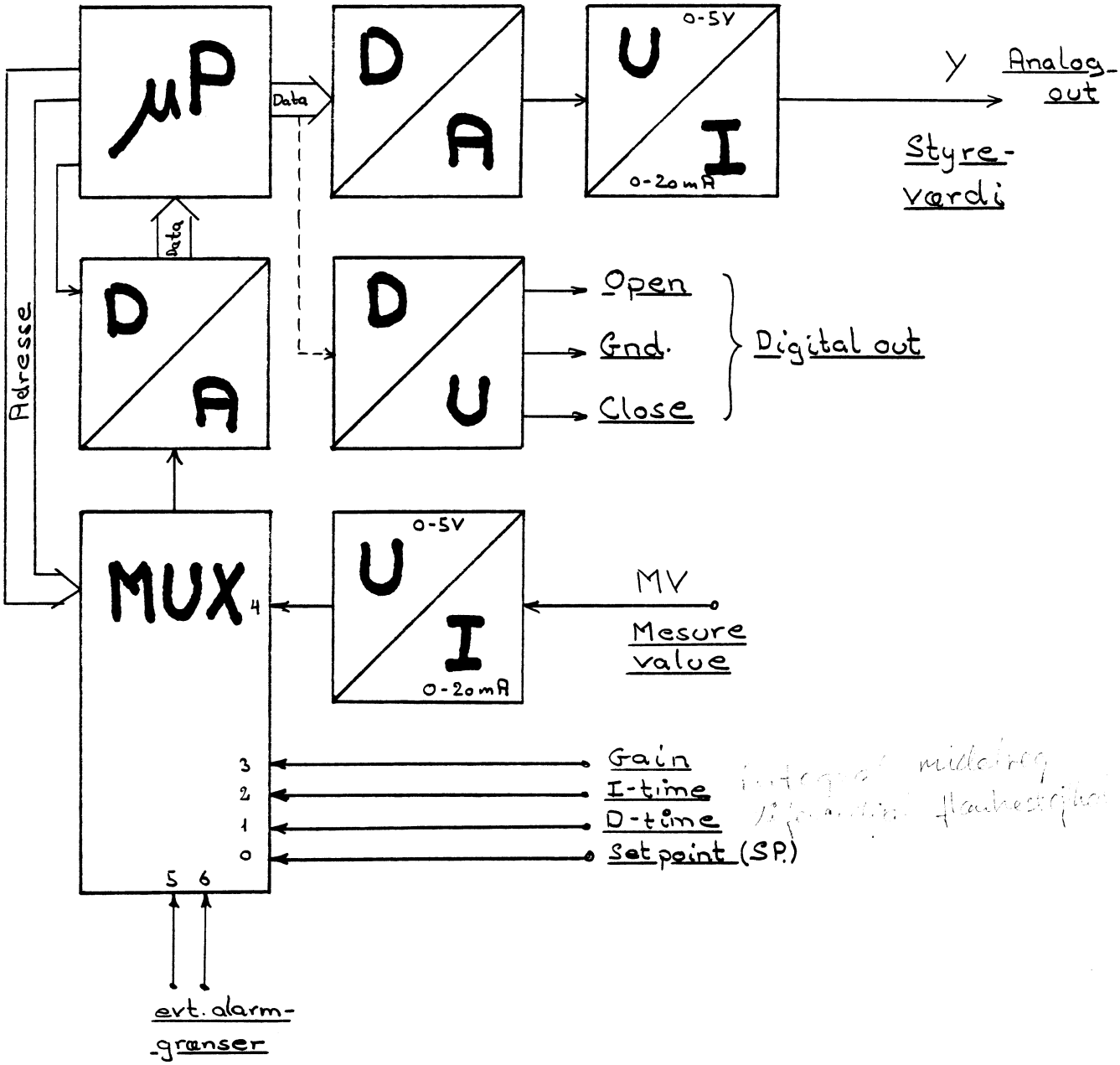
Printhøjde	:	233,4 mm
Printdybde	:	160 mm
Modulbredde	:	20 mm

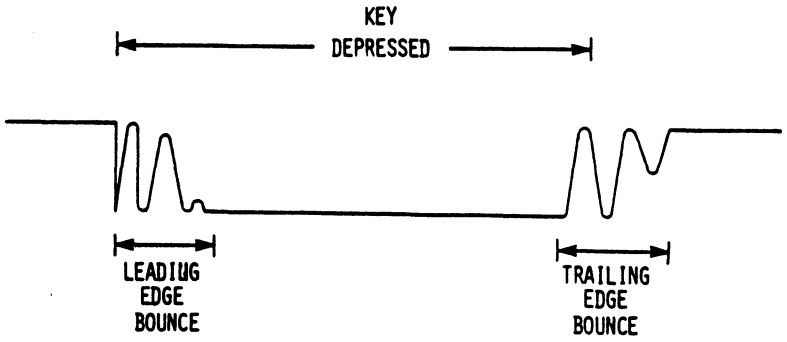
Eksempel på et enkelt regule-
ringssystem





REGULATOR MED μ-PROCESSOR





- BOUNCE IS 10-20 MSEC
- HARDWARE SOLUTION: R-C FILTER
- SOFTWARE SOLUTION: VERIFY KEY STATUS FOR 20 MS

Fig. 4-0: Key Bounce

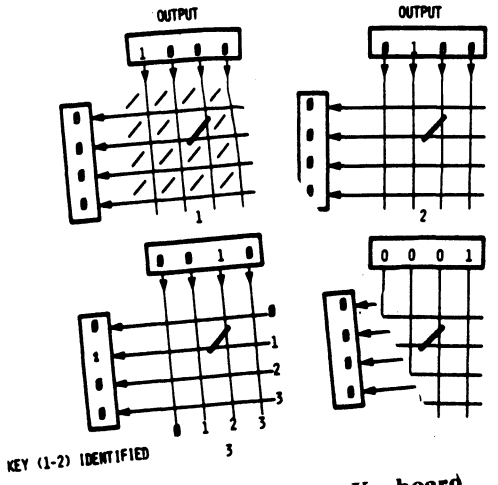


Fig. 4-3: Walking Ones Keyboard ode

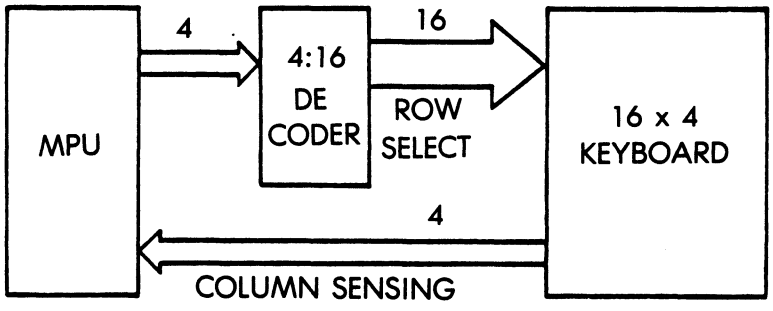


Fig. 4-4: 4-to-16-Line Decoder with Keyboard

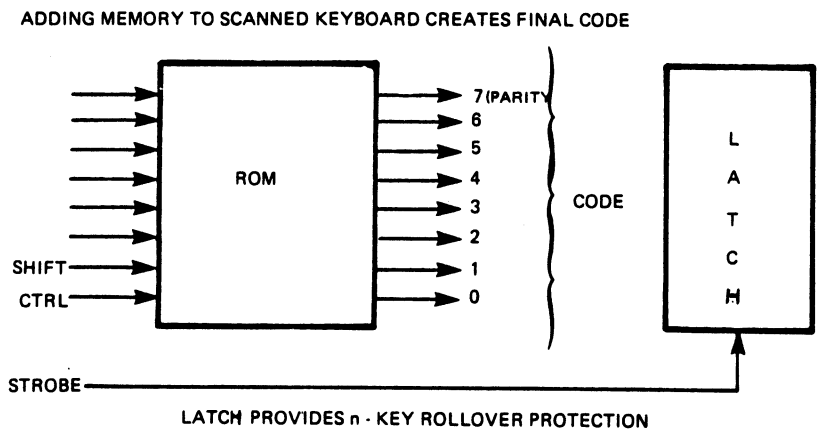


Fig. 4-9: ROM and Latch

BIT NUMBERS								0	0	0	0	1	1	1	1
								0	0	1	1	0	0	1	1
								0	1	0	1	0	1	0	1
b_7	b_6	b_5	b_4	b_3	b_2	b_1	HEX 1 / HEX 0	0	1	2	3	4	5	6	7
			0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
			0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
			0	0	1	0	2	STX	DC2	"	2	B	R	b	r
			0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
			0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
			0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
			0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
			0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
			1	0	0	0	8	BS	CAN	(	8	H	X	h	x
			1	0	0	1	9	HT	EM	)	9	I	Y	i	y
			1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
			1	0	1	1	11	VT	ESC	+	;	K	[	k	{
			1	1	0	0	12	FF	FS	,	<	L	\	l	|
			1	1	0	1	13	CR	GS	-	=	M	]	m	}
			1	1	1	0	14	SO	RS	.	>	N	^	n	~
			1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Fig. 4-14: ASCII Table

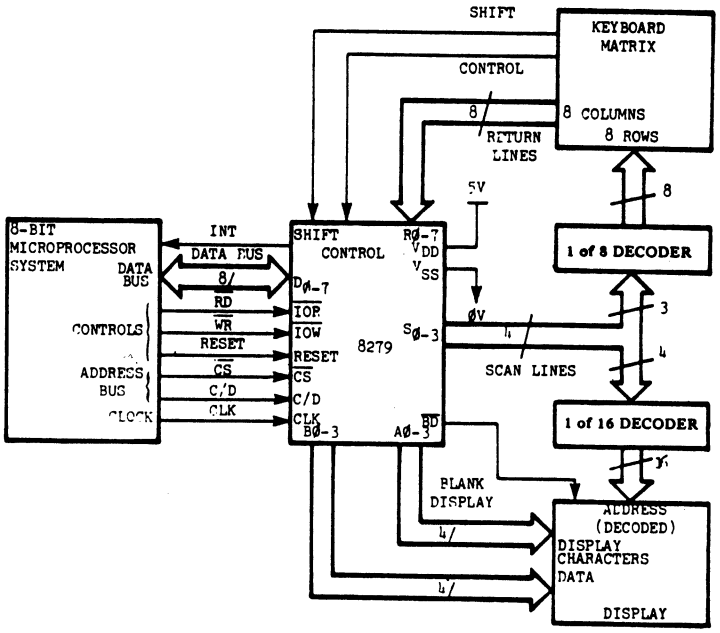
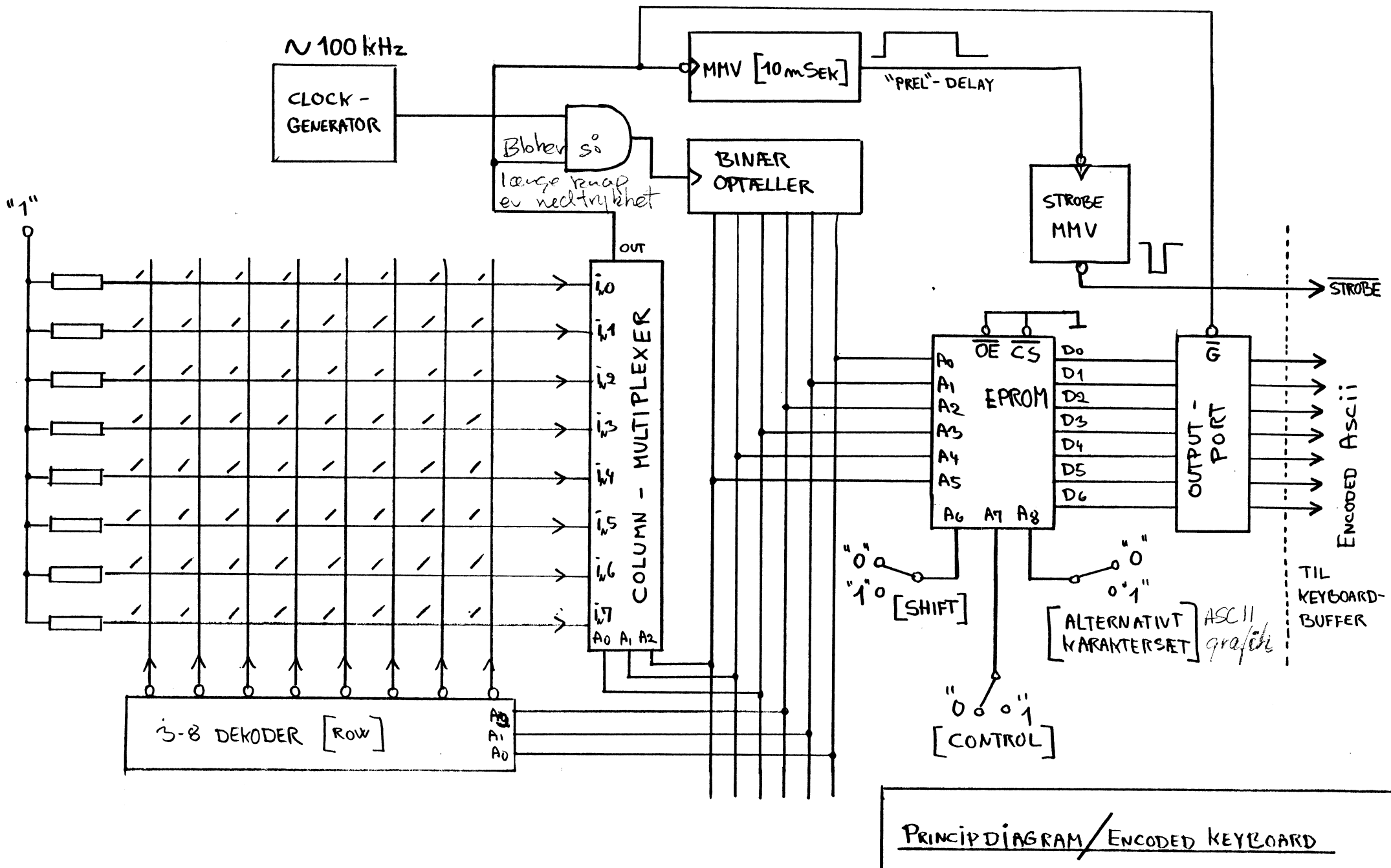


Fig. 4-12: 8279 Keyboard Display Controller



PRINCIPDIAGRAM / ENCODED KEYBOARD

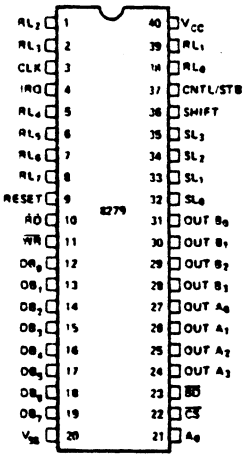
8279/8279-5
PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

- MCS-85™ Compatible 8279-5
- Simultaneous Keyboard Display Operations
- Scanned Keyboard Mode
- Scanned Sensor Mode
- Strobed Input Entry Mode
- 8-Character Keyboard FIFO
- 2-Key Lockout or N-Key Rollover with Contact Debounce
- Dual 8- or 16-Numerical Display
- Single 16-Character Display
- Right or Left Entry 16-Byte Display RAM
- Mode Programmable from CPU
- Programmable Scan Timing
- Interrupt Output on Key Entry

The Intel® 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel® microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the hall effect and ferrite variety. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries are debounced and strobed in an 8-character FIFO. If more than 8 characters are entered, overrun status is set. Key entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has 16X8 display RAM which can be organized into dual 16X4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

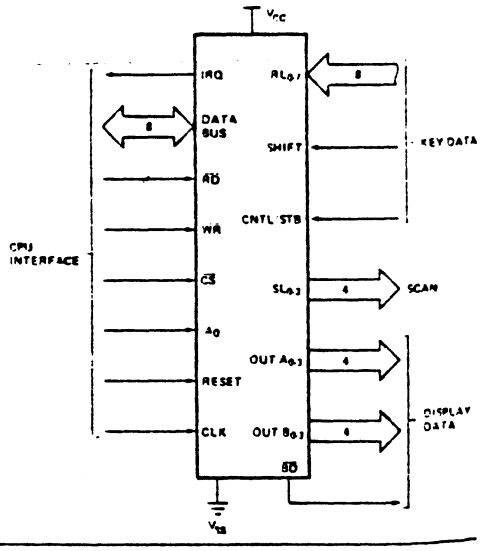
PIN CONFIGURATION



PIN NAMES

DB0-7	I/O	DATA BUS (BI-DIRECTIONAL)
CLK	I	CLOCK INPUT
RESET	I	RESET INPUT
CS	I	CHIP SELECT
RD	I	READ INPUT
WR	I	WRITE INPUT
A0	I	BUFFER ADDRESS
IRQ	O	INTERRUPT REQUEST OUTPUT
SL0-3	O	SCAN LINES
RL0-7	I	RETURN LINES
SHIFT	I	SHIFT INPUT
CNTL/STB	I	CONTROL/STROBE INPUT
OUT A0-3	O	DISPLAY A0 OUTPUTS
OUT B0-3	O	DISPLAY B0 OUTPUTS
BD	O	BLANK DISPLAY OUTPUT

LOGIC SYMBOL



HARDWARE DESCRIPTION

The 8279 is packaged in a 40 pin DIP. The following is a functional description of each pin.

No. Of Pins	Designation	Function
8	DB_0-DB_7	Bi-directional data bus. All data and commands between the CPU and the 8279 are transmitted on these lines.
1	CLK	Clock from system used to generate internal timing.
1	RESET	A high signal on this pin resets the 8279. After being reset the 8279 is placed in the following mode: 1) 16 8-bit character display —left entry. 2) Encoded scan keyboard—2 key lockout. Along with this the program clock prescaler is set to 31.
1	$\overline{CS}$	Chip Select. A low on this pin enables the interface functions to receive or transmit.
1	A_0	Buffer Address. A high on this line indicates the signals in or out are interpreted as a command or status. A low indicates that they are data.
2	$\overline{RD}$, $\overline{WR}$	Input/Output read and write. These signals enable the data buffers to either send data to the external bus or receive it from the external bus.
1	IRQ	Interrupt Request. In a keyboard mode, the interrupt line is high when there is data in the FIFO/Sensor RAM. The interrupt line goes low with each FIFO/Sensor RAM read and returns high if there is still information in the RAM. In a sensor mode, the interrupt line goes high whenever a change in a sensor is detected.
2	V_{SS}, V_{CC}	Ground and power supply pins.
4	SL_0-SL_3	Scan Lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 16) or decoded (1 of 4)
8	RL_0-RL_7	Return line inputs which are connected to the scan lines through the keys or sensor switches. They have active internal pullups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the Strobed Input mode.
1	SHIFT	The shift input status is stored along with the key position on key closure in the Scanned Keyboard modes. It has an active internal pullup to keep it high until a switch closure pulls it low.
1	CNTL/STB	For keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into the FIFO in the Strobed Input mode. (Rising Edge). It has an active internal pullup to keep it high until a switch closure pulls it low.
4	OUT A_0-OUT A_3	These two ports are the outputs for the 16 x 4 display refresh registers. The data from these outputs is synchronized to the scan lines (SL_0-SL_3) for multiplexed digit displays. The two 4 bit ports may be blanked independently. These two ports may also be considered as one 8 bit port.
4	OUT B_0-OUT B_3	
1	$\overline{BD}$	Blank Display. This output is used to blank the display during digit switching or by a display blanking command.

PRINCIPLES OF OPERATION

The following is a description of the major elements of the 8279 Programmable Keyboard/Display interface device. Refer to the block diagram in Figure 1.

I/O Control and Data Buffers

The I/O control section uses the $\overline{CS}$, A_0, $\overline{RD}$ and $\overline{WR}$ lines to control data flow to and from the various internal registers and buffers. All data flow to and from the 8279 is enabled by $\overline{CS}$. The character of the information, given or desired by the CPU, is identified by A_0. A logic one means the information is a command or status. A logic zero means the information is data. $\overline{RD}$ and $\overline{WR}$ determine the direction of data flow through the Data Buffers. The Data Buffers are bi-directional buffers that connect the internal bus to the external bus. When the chip is not selected ($\overline{CS}$ = 1), the devices are in a high impedance state. The drivers input during $\overline{WR} \bullet \overline{CS}$ and output during $\overline{RD} \bullet \overline{CS}$.

Control and Timing Registers and Timing Control

These registers store the keyboard and display modes and other operating conditions programmed by the CPU. The modes are programmed by presenting the proper command on the data lines with A_0 = 1 and then sending a $\overline{WR}$. The command is latched on the rising edge of $\overline{WR}$.

FUNCTIONAL DESCRIPTION

Since data input and display are an integral part of many microprocessor designs, the system designer needs an interface that can control these functions without placing a large load on the CPU. The 8279 provides this function for 8-bit microprocessors.

The 8279 has two sections: keyboard and display. The keyboard section can interface to regular typewriter style keyboards or random toggle or thumb switches. The display section drives alphanumeric displays or a bank of indicator lights. Thus the CPU is relieved from scanning the keyboard or refreshing the display.

The 8279 is designed to directly connect to the microprocessor bus. The CPU can program all operating modes for the 8279. These modes include:

Input Modes

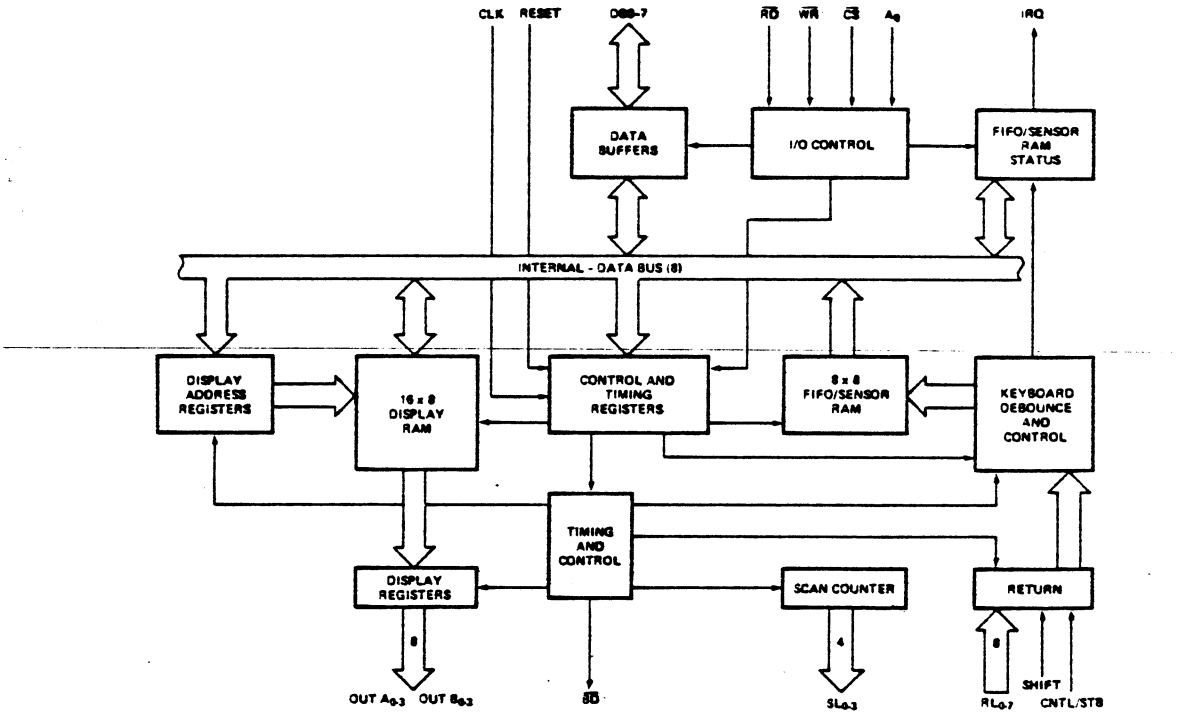
- Scanned Keyboard — with encoded (8 x 8 key keyboard) or decoded (4 x 8 key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key rollover.
- Scanned Sensor Matrix — with encoded (8 x 8 matrix switches) or decoded (4 x 8 matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.
- Strobed Input — Data on return lines during control line strobe is transferred to FIFO.

Output Modes

- 8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit ($B_0 = D_0$, $A_3 = D_7$).
- Right entry or left entry display formats.

Other features of the 8279 include:

- Mode programming from the CPU.
- Clock Prescaler
- Interrupt output to signal CPU when there is keyboard or sensor data available.
- An 8 byte FIFO to store keyboard information.
- 16 byte internal Display RAM for display refresh. This RAM can also be read by the CPU.



The command is then decoded and the appropriate function is set. The timing control contains the basic timing counter chain. The first counter is a ÷ N prescaler that can be programmed to yield an internal frequency of 100 kHz which gives a 5.1 ms keyboard scan time and a 10.3 ms debounce time. The other counters divide down the basic internal frequency to provide the proper key scan, row scan, keyboard matrix scan, and display scan times.

Scan Counter

The scan counter has two modes. In the encoded mode, the counter provides a binary count that must be externally decoded to provide the scan lines for the keyboard and display. In the decoded mode, the scan counter decodes the least significant 2 bits and provides a decoded 1 of 4 scan. Note than when the keyboard is in decoded scan, so is the display. This means that only the first 4 characters in the Display RAM are displayed.

In the encoded mode, the scan lines are active high outputs. In the decoded mode, the scan lines are active low outputs.

Return Buffers and Keyboard Debounce and Control

The 8 return lines are buffered and latched by the Return Buffers. In the keyboard mode, these lines are scanned, looking for key closures in that row. If the debounce circuit detects a closed switch, it waits about 10 msec to check if the switch remains closed. If it does, the address of the switch in the matrix plus the status of SHIFT and CONTROL are transferred to the FIFO. In the scanned Sensor Matrix modes, the contents of the return lines is directly transferred to the corresponding row of the Sensor RAM (FIFO) each key scan time. In Strobed Input mode, the contents of the return lines are transferred to the FIFO on the rising edge of the CNTL/STB line pulse.

FIFO/Sensor RAM and Status

This block is a dual function 8 x 8 RAM. In Keyboard or Strobed Input modes, it is a FIFO. Each new entry is written into successive RAM positions and each is then read in order of entry. FIFO status keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or writes will be recognized as an error. The status can be read by an $\overline{RD}$ with $\overline{CS}$ low and A_0 high. The status logic also provides an IRQ signal when the FIFO is not empty. In Scanned Sensor Matrix mode, the memory is a Sensor RAM. Each row of the Sensor RAM is loaded with the status of the corresponding row of sensor in the sensor matrix. In this mode, IRQ is high if a change in a sensor is detected.

Display Address Registers and Display RAM

The Display Address Registers hold the address of the word currently being written or read by the CPU and the two 4-bit nibbles being displayed. The read/write addresses are programmed by CPU command. They also can be set to auto increment after each read or write. The Display RAM can be directly read by the CPU after the correct mode and address is set. The addresses for the A and B nibbles are automatically updated by the 8279 to match data entry by the CPU. The A and B nibbles can be entered independently or as one word, according to the mode that is set by the CPU. Data entry to the display can be set to either left or right entry. See Interface Considerations for details.

SOFTWARE OPERATION

8279 commands

The following commands program the 8279 operating modes. The commands are sent on the Data Bus with $\overline{CS}$ low and A_0 high and are loaded to the 8279 on the rising edge of $\overline{WR}$.

Keyboard/Display Mode Set

	MSB							LSB
Code:	0	0	0	D	D	K	K	K

Where DD is the Display Mode and KKK is the Keyboard Mode.

DD

0	0	8 8-bit character display — Left entry
0	1	16 8-bit character display — Left entry*
1	0	8 8-bit character display — Right entry
1	1	16 8-bit character display — Right entry

For description of right and left entry, see Interface Considerations. Note that when decoded scan is set in keyboard mode, the display is reduced to 4 characters independent of display mode set.

KKK

0	0	0	Encoded Scan Keyboard — 2 Key Lockout*
0	0	1	Decoded Scan Keyboard — 2-Key Lockout
0	1	0	Encoded Scan Keyboard — N-Key Rollover
0	1	1	Decoded Scan Keyboard — N-Key Rollover
1	0	0	Encoded Scan Sensor Matrix
1	0	1	Decoded Scan Sensor Matrix
1	1	0	Strobed Input, Encoded Display Scan
1	1	1	Strobed Input, Decoded Display Scan

Program Clock

Code:	0	0	1	P	P	P	P	P

All timing and multiplexing signals for the 8279 are generated by an internal prescaler. This prescaler divides the external clock (pin 3) by a programmable integer. Bits PPPPP determine the value of this integer which ranges from 2 to 31. Choosing a divisor that yields 100 kHz will give the specified scan and debounce times. For instance, if Pin 3 of the 8279 is being clocked by a 2 MHz signal, PPPPP should be set to 10100 to divide the clock by 20 to yield the proper 100 kHz operating frequency.

Read FIFO/Sensor RAM

Code:	0	1	0	AI	X	A	A	A	X = Don't Care

The CPU sets up the 8279 for a read of the FIFO/Sensor RAM by first writing this command. In the Scan Key-

*Default after reset.

board Mode, the Auto-Increment flag (AI) and the RAM address bits (AAA) are irrelevant. The 8279 will automatically drive the data bus for each subsequent read ($A_0 = 0$) in the same sequence in which the data first entered the FIFO. All subsequent reads will be from the FIFO until another command is issued.

In the Sensor Matrix Mode, the RAM address bits AAA select one of the 8 rows of the Sensor RAM. If the AI flag is set (AI = 1), each successive read will be from the subsequent row of the sensor RAM.

Read Display RAM

Code:

0	1	1	AI	A	A	A	A

The CPU sets up the 8279 for a read of the Display RAM by first writing this command. The address bits AAAA select one of the 16 rows of the Display RAM. If the AI flag is set (AI = 1), this row address will be incremented after each following read *or write* to the Display RAM. Since the same counter is used for both reading and writing, this command sets the *next read or write* address and the sense of the Auto-Increment mode for both operations.

Write Display RAM

Code:

1	0	0	AI	A	A	A	A

The CPU sets up the 8279 for a write to the Display RAM by first writing this command. After writing the command with $A_0 = 1$, all subsequent writes with $A_0 = 0$ will be to the Display RAM. The addressing and Auto-Increment functions are identical to those for the Read Display RAM. However, this command does not affect the source of subsequent Data Reads; the CPU will read from whichever RAM (Display or FIFO/Sensor) which was last specified. If, indeed, the Display RAM was last specified, the Write Display RAM will, nevertheless, change the next Read location.

Display Write Inhibit/Blanking

				A	B	A	B
Code:	1	0	1	X	IW	IW	BL	BL

The IW Bits can be used to mask nibble A and nibble B in applications requiring separate 4-bit display ports. By setting the IW flag (IW = 1) for one of the ports, the port becomes marked so that entries to the Display RAM from the CPU do not affect that port. Thus, if each nibble is input to a BCD decoder, the CPU may write a digit to the Display RAM without affecting the other digit being displayed. It is important to note that bit B_0 corresponds to bit D_0 on the CPU bus, and that bit A_3 corresponds to bit D_7.

If the user wishes to blank the display, the BL flags are available for each nibble. The last Clear command issued determines the code to be used as a "blank." This code defaults to all zeros after a reset. Note that both BL flags must be set to blank a display formatted with a single 8-bit port.

Clear

Code:

1	1	0	C_D	C_D	C_D	C_F	C_A

The C_D bits are available in this command to clear all rows of the Display RAM to a selectable blanking code as follows:

C_D C_D C_D

0	X	All Zeros (X = Don't Care)
1	0	AB = Hex 20 (0010 0000)
1	1	All Ones

Enable clear display when = 1 (or by C_A = 1)

During the time the Display RAM is being cleared (~160 μs), it may not be written to. The most significant bit of the FIFO status word is set during this time. When the Display RAM becomes available again, it automatically resets.

If the C_F bit is asserted ($C_F = 1$), the FIFO status is cleared and the interrupt output line is reset. Also, the Sensor RAM pointer is set to row 0.

C_A, the Clear All bit, has the combined effect of C_D and C_F; it uses the C_D clearing code on the Display RAM and also clears FIFO status. Furthermore, it resynchronizes the internal timing chain.

End Interrupt/Error Mode Set

Code:

1	1	1	E	X	X	X	X

X = Don't care.

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM. (The IRQ line would have been raised upon the detection of a change in a sensor value. This would have also inhibited further writing into the RAM until reset).

For the N-key rollover mode — if the E bit is programmed to "1" the chip will operate in the special Error mode. (For further details, see Interface Considerations Section.)

Status Word

The status word contains the FIFO status, error, and display unavailable signals. This word is read by the CPU when A_0 is high and $\overline{CS}$ and $\overline{RD}$ are low. See Interface Considerations for more detail on status word.

Data Read

Data is read when A_0, $\overline{CS}$ and $\overline{RD}$ are all low. The source of the data is specified by the Read FIFO or Read Display commands. The trailing edge of $\overline{RD}$ will cause the address of the RAM being read to be incremented if the Auto-Increment flag is set. FIFO reads always increment (if no error occurs) independent of AI.

Data Write

Data that is written with A_0, $\overline{CS}$ and $\overline{WR}$ low is always written to the Display RAM. The address is specified by the latest Read Display or Write Display command. Auto-Incrementing on the rising edge of $\overline{WR}$ occurs if AI set by the latest display command.

INTERFACE CONSIDERATIONS

Scanned Keyboard Mode, 2-Key Lockout

There are three possible combinations of conditions that can occur during debounce scanning. When a key is depressed, the debounce logic is set. Other depressed keys are looked for during the next two scans. If none are encountered, it is a single key depression and the key position is entered into the FIFO along with the status of CNTL and SHIFT lines. If the FIFO was empty, IRQ will be set to signal the CPU that there is an entry in the FIFO. If the FIFO was full, the key will not be entered and the error flag will be set. If another closed switch is encountered, no entry to the FIFO can occur. If all other keys are released before this one, then it will be entered to the FIFO. If this key is released before any other, it will be entirely ignored. A key is entered to the FIFO only once per depression, no matter how many keys were pressed along with it or in what order they were released. If two keys are depressed within the debounce cycle, it is a simultaneous depression. Neither key will be recognized until one key remains depressed alone. The last key will be treated as a single key depression.

Scanned Keyboard Mode, N-Key Rollover

With N-key Rollover each key depression is treated independently from all others. When a key is depressed, the debounce circuit waits 2 keyboard scans and then checks to see if the key is still down. If it is, the key is entered into the FIFO. Any number of keys can be depressed and another can be recognized and entered into the FIFO. If a simultaneous depression occurs, the keys are recognized and entered according to the order the keyboard scan found them.

Scanned Keyboard — Special Error Modes

For N-key rollover mode the user can program a special error mode. This is done by the "End Interrupt/Error Mode Set" command. The debounce cycle and key-validity check are as in normal N-key mode. If during a single debounce cycle, two keys are found depressed, this is considered a simultaneous multiple depression, and sets an error flag. This flag will prevent any further writing into the FIFO and will set interrupt (if not yet set). The error flag could be read in this mode by reading the FIFO STATUS word. (See "FIFO STATUS" for further details.) The error flag is reset by sending the normal CLEAR command with $CF = 1$.

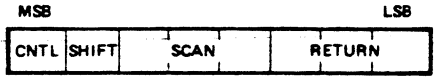
Sensor Matrix Mode

In Sensor Matrix mode, the debounce logic is inhibited. The status of the sensor switch is inputted directly to the Sensor RAM. In this way the Sensor RAM keeps an image of the state of the switches in the sensor matrix. Although debouncing is not provided, this mode has the advantage that the CPU knows how long the sensor was closed and when it was released. A keyboard mode can only indicate a validated closure. To make the software easier, the designer should functionally group the sensors by row since this is the format in which the CPU will read them. The IRQ line goes high if any sensor value change is detected at the end of a sensor matrix scan. The IRQ line is cleared by the first data read operation if the Auto-Increment flag is set to zero, or by the End Interrupt command if the Auto-Increment flag is set to one.

Note: Multiple changes in the matrix Addressed by ($SL_{0-3} = 0$) may cause multiple interrupts. ($SL_0 = 0$ in the Decoded Mode). Reset may cause the 8279 to see multiple changes.

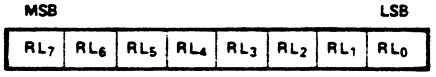
Data Format

In the Scanned Keyboard mode, the character entered into the FIFO corresponds to the position of the switch in the keyboard plus the status of the CNTL and SHIFT lines (non-inverted). CNTL is the MSB of the character and SHIFT is the next most significant bit. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to which return line the key was connected.

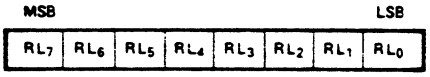


SCANNED KEYBOARD DATA FORMAT

In Sensor Matrix mode, the data on the return lines is entered directly in the row of the Sensor RAM that corresponds to the row in the matrix being scanned. Therefore, each switch postion maps directly to a Sensor RAM position. The SHIFT and CNTL inputs are ignored in this mode. Note that switches are not necessarily the only thing that can be connected to the return lines in this mode. Any logic that can be triggered by the scan lines can enter data to the return line inputs. Eight multiplexed input ports could be tied to the return lines and scanned by the 8279.



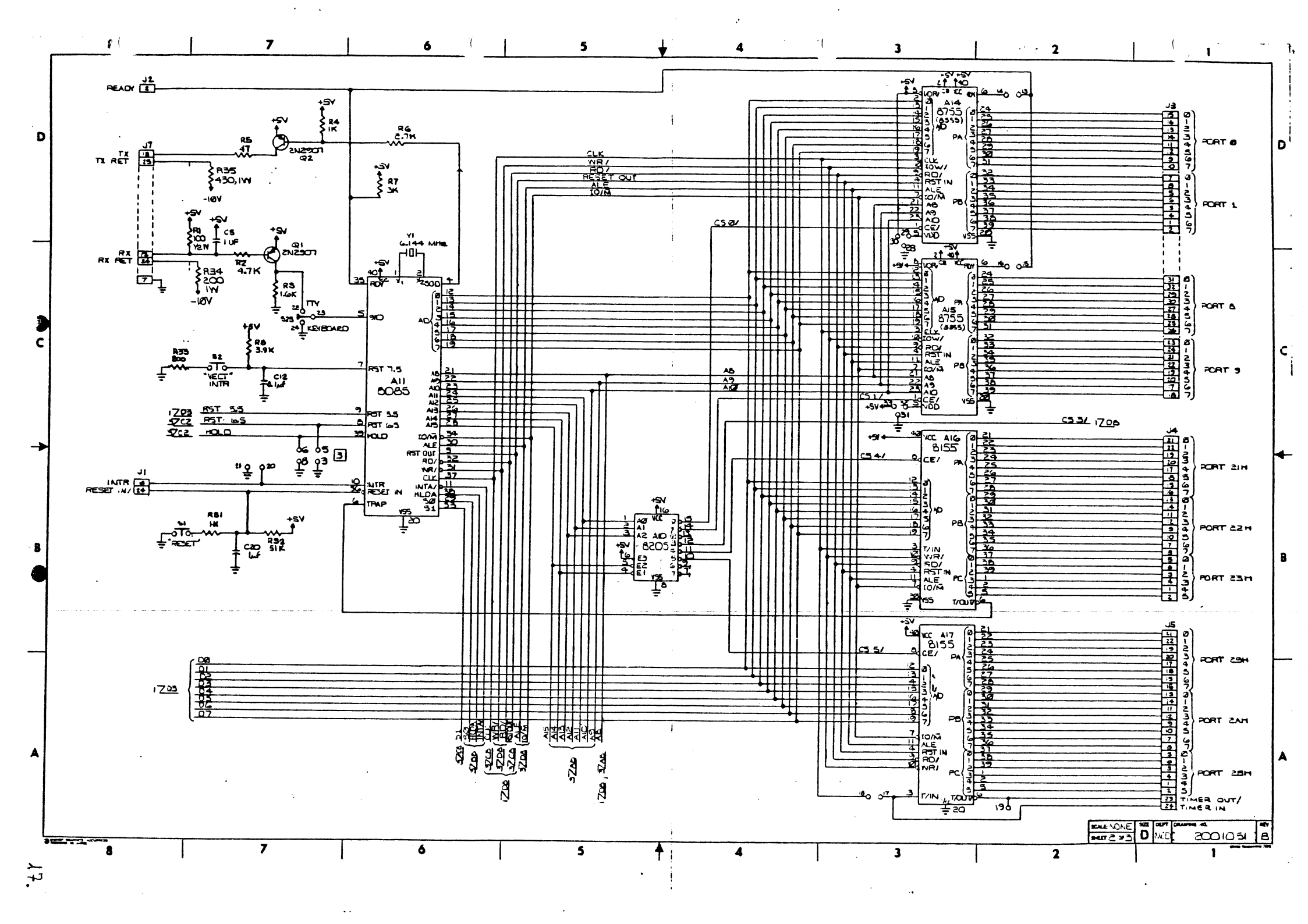
In Strobed Input mode, the data is also entered to the FIFO from the return lines. The data is entered by the rising edge of a CNTL/STB line pulse. Data can come from another encoded keyboard or simple switch matrix. The return lines can also be used as a general purpose strobed input.

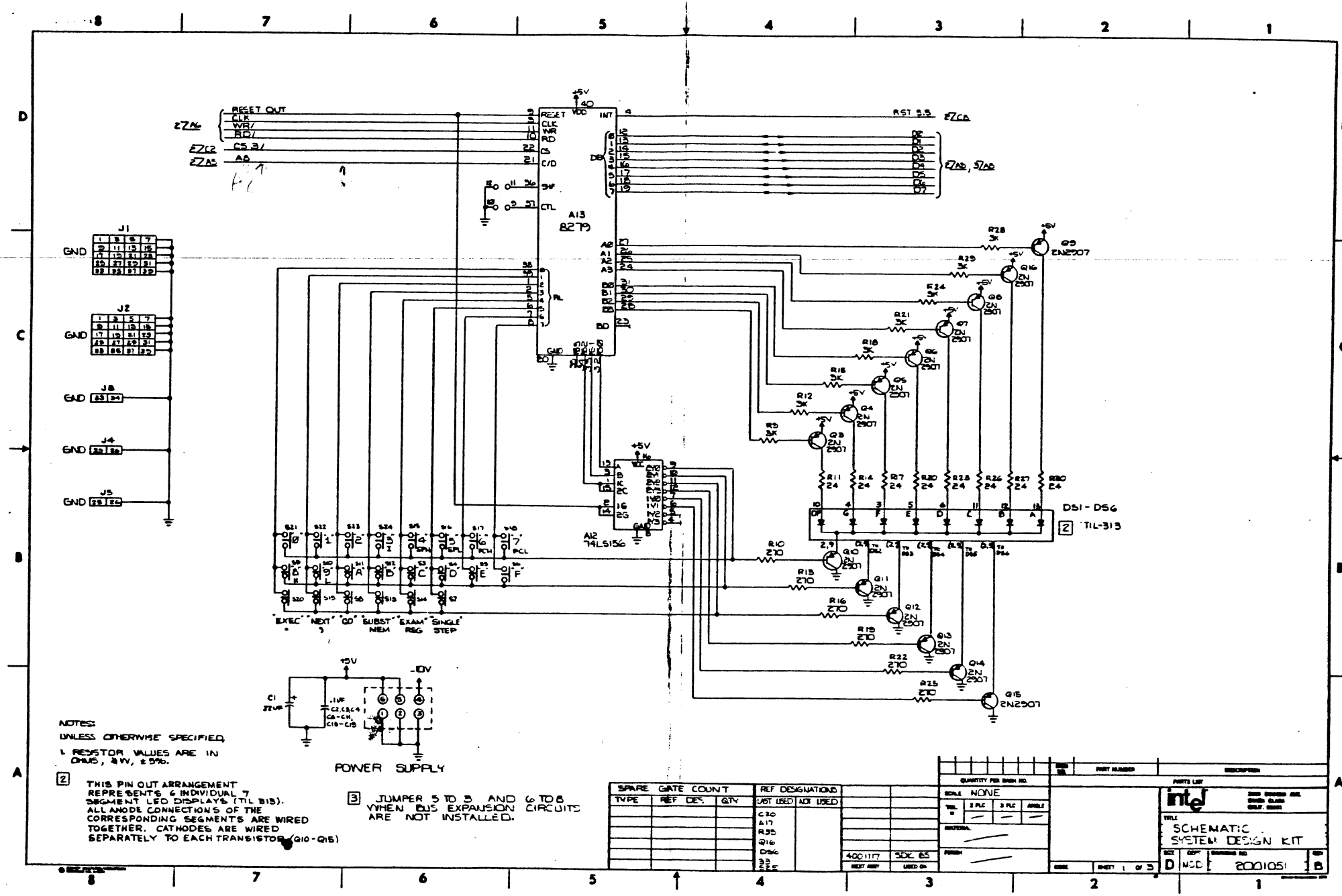


Display

Left Entry

Left Entry mode is the simplest display format in that each display position directly corresponds to a byte (or nibble) in the Display RAM. Address 0 in the RAM is the left-most display character and address 15 (or address 7 in 8 character display) is the right most display character. Entering characters from position zero causes the display to fill from the left. The 17th (9th) character is entered back in the left most position and filling again proceeds from there


READY
J2
+5V
R4 1K
R6 2.7K
R7 1K
R5 47
2N2907 Q2
TX
TX RET
J7
R35 430, 1W
-10V
R3 100 1/2W
C5 1uF
Q1 2N2907
RX
RX RET
R34 200 1W
R2 4.7K
R5 1.6K
TTY
KEYBOARD
S2S
S2
R8 3.9K
R33 820
"VECT INTR"
C12 1uF
Y1 6.144 MHz
RST 7.5
RST 5.5
RST 6.5
HOLD
A11 8085
INTR
RESET IN
TRAP
J1
RESET
R31 1K
R32 51K
C20 1uF
CLK
WR/
RD/
RESET OUT
ALE
IO/M
CS 0/
CS 1/
CS 3/ 1Z05
CS 4/
CS 5/
A14 8755 (8355)
A15 8755 (8355)
A16 8155
A17 8155
A8
A9
A10
A11
A12
A13
A14
A15
AD
A8 A9 A10
8205
PORT 0
PORT 1
PORT A
PORT 9
PORT 21H
PORT 22H
PORT 23H
PORT 29H
PORT 2AH
PORT 2BH
TIMER OUT/
TIMER IN
1Z03
1Z04
1Z05
1Z06
1Z07
1Z08
1Z09
1Z10
1Z11
1Z12
1Z13
1Z14
1Z15
1Z16
1Z17
1Z18
1Z19
1Z20
1Z21
1Z22
1Z23
1Z24
1Z25
1Z26
1Z27
1Z28
1Z29
1Z30
1Z31
1Z32
1Z33
1Z34
1Z35
1Z36
1Z37
1Z38
1Z39
1Z40
1Z41
1Z42
1Z43
1Z44
1Z45
1Z46
1Z47
1Z48
1Z49
1Z50
1Z51
1Z52
1Z53
1Z54
1Z55
1Z56
1Z57
1Z58
1Z59
1Z60
1Z61
1Z62
1Z63
1Z64
1Z65
1Z66
1Z67
1Z68
1Z69
1Z70
1Z71
1Z72
1Z73
1Z74
1Z75
1Z76
1Z77
1Z78
1Z79
1Z80
1Z81
1Z82
1Z83
1Z84
1Z85
1Z86
1Z87
1Z88
1Z89
1Z90
1Z91
1Z92
1Z93
1Z94
1Z95
1Z96
1Z97
1Z98
1Z99
1Z100
1Z101
1Z102
1Z103
1Z104
1Z105
1Z106
1Z107
1Z108
1Z109
1Z110
1Z111
1Z112
1Z113
1Z114
1Z115
1Z116
1Z117
1Z118
1Z119
1Z120
1Z121
1Z122
1Z123
1Z124
1Z125
1Z126
1Z127
1Z128
1Z129
1Z130
1Z131
1Z132
1Z133
1Z134
1Z135
1Z136
1Z137
1Z138
1Z139
1Z140
1Z141
1Z142
1Z143
1Z144
1Z145
1Z146
1Z147
1Z148
1Z149
1Z150
1Z151
1Z152
1Z153
1Z154
1Z155
1Z156
1Z157
1Z158
1Z159
1Z160
1Z161
1Z162
1Z163
1Z164
1Z165
1Z166
1Z167
1Z168
1Z169
1Z170
1Z171
1Z172
1Z173
1Z174
1Z175
1Z176
1Z177
1Z178
1Z179
1Z180
1Z181
1Z182
1Z183
1Z184
1Z185
1Z186
1Z187
1Z188
1Z189
1Z190
1Z191
1Z192
1Z193
1Z194
1Z195
1Z196
1Z197
1Z198
1Z199
1Z200
1Z201
1Z202
1Z203
1Z204
1Z205
1Z206
1Z207
1Z208
1Z209
1Z210
1Z211
1Z212
1Z213
1Z214
1Z215
1Z216
1Z217
1Z218
1Z219
1Z220
1Z221
1Z222
1Z223
1Z224
1Z225
1Z226
1Z227
1Z228
1Z229
1Z230
1Z231
1Z232
1Z233
1Z234
1Z235
1Z236
1Z237
1Z238
1Z239
1Z240
1Z241
1Z242
1Z243
1Z244
1Z245
1Z246
1Z247
1Z248
1Z249
1Z250
1Z251
1Z252
1Z253
1Z254
1Z255
1Z256
1Z257
1Z258
1Z259
1Z260
1Z261
1Z262
1Z263
1Z264
1Z265
1Z266
1Z267
1Z268
1Z269
1Z270
1Z271
1Z272
1Z273
1Z274
1Z275
1Z276
1Z277
1Z278
1Z279
1Z280
1Z281
1Z282
1Z283
1Z284
1Z285
1Z286
1Z287
1Z288
1Z289
1Z290
1Z291
1Z292
1Z293
1Z294
1Z295
1Z296
1Z297
1Z298
1Z299
1Z300
1Z301
1Z302
1Z303
1Z304
1Z305
1Z306
1Z307
1Z308
1Z309
1Z310
1Z311
1Z312
1Z313
1Z314
1Z315
1Z316
1Z317
1Z318
1Z319
1Z320
1Z321
1Z322
1Z323
1Z324
1Z325
1Z326
1Z327
1Z328
1Z329
1Z330
1Z331
1Z332
1Z333
1Z334
1Z335
1Z336
1Z337
1Z338
1Z339
1Z340
1Z341
1Z342
1Z343
1Z344
1Z345
1Z346
1Z347
1Z348
1Z349
1Z350
1Z351
1Z352
1Z353
1Z354
1Z355
1Z356
1Z357
1Z358
1Z359
1Z360
1Z361
1Z362
1Z363
1Z364
1Z365
1Z366
1Z367
1Z368
1Z369
1Z370
1Z371
1Z372
1Z373
1Z374
1Z375
1Z376
1Z377
1Z378
1Z379
1Z380
1Z381
1Z382
1Z383
1Z384
1Z385
1Z386
1Z387
1Z388
1Z389
1Z390
1Z391
1Z392
1Z393
1Z394
1Z395
1Z396
1Z397
1Z398
1Z399
1Z400
1Z401
1Z402
1Z403
1Z404
1Z405
1Z406
1Z407
1Z408
1Z409
1Z410
1Z411
1Z412
1Z413
1Z414
1Z415
1Z416
1Z417
1Z418
1Z419
1Z420
1Z421
1Z422
1Z423
1Z424
1Z425
1Z426
1Z427
1Z428
1Z429
1Z430
1Z431
1Z432
1Z433
1Z434
1Z435
1Z436
1Z437
1Z438
1Z439
1Z440
1Z441
1Z442
1Z443
1Z444
1Z445
1Z446
1Z447
1Z448
1Z449
1Z450
1Z451
1Z452
1Z453
1Z454
1Z455
1Z456
1Z457
1Z458
1Z459
1Z460
1Z461
1Z462
1Z463
1Z464
1Z465
1Z466
1Z467
1Z468
1Z469
1Z470
1Z471
1Z472
1Z473
1Z474
1Z475
1Z476
1Z477
1Z478
1Z479
1Z480
1Z481
1Z482
1Z483
1Z484
1Z485
1Z486
1Z487
1Z488
1Z489
1Z490
1Z491
1Z492
1Z493
1Z494
1Z495
1Z496
1Z497
1Z498
1Z499
1Z500
1Z501
1Z502
1Z503
1Z504
1Z505
1Z506
1Z507
1Z508
1Z509
1Z510
1Z511
1Z512
1Z513
1Z514
1Z515
1Z516
1Z517
1Z518
1Z519
1Z520
1Z521
1Z522
1Z523
1Z524
1Z525
1Z526
1Z527
1Z528
1Z529
1Z530
1Z531
1Z532
1Z533
1Z534
1Z535
1Z536
1Z537
1Z538
1Z539
1Z540
1Z541
1Z542
1Z543
1Z544
1Z545
1Z546
1Z547
1Z548
1Z549
1Z550
1Z551
1Z552
1Z553
1Z554
1Z555
1Z556
1Z557
1Z558
1Z559
1Z560
1Z561
1Z562
1Z563
1Z564
1Z565
1Z566
1Z567
1Z568
1Z569
1Z570
1Z571
1Z572
1Z573
1Z574
1Z575
1Z576
1Z577
1Z578
1Z579
1Z580
1Z581
1Z582
1Z583
1Z584
1Z585
1Z586
1Z587
1Z588
1Z589
1Z590
1Z591
1Z592
1Z593
1Z594
1Z595
1Z596
1Z597
1Z598
1Z599
1Z600
1Z601
1Z602
1Z603
1Z604
1Z605
1Z606
1Z607
1Z608
1Z609
1Z610
1Z611
1Z612
1Z613
1Z614
1Z615
1Z616
1Z617
1Z618
1Z619
1Z620
1Z621
1Z622
1Z623
1Z624
1Z625
1Z626
1Z627
1Z628
1Z629
1Z630
1Z631
1Z632
1Z633
1Z634
1Z635
1Z636
1Z637
1Z638
1Z639
1Z640
1Z641
1Z642
1Z643
1Z644
1Z645
1Z646
1Z647
1Z648
1Z649
1Z650
1Z651
1Z652
1Z653
1Z654
1Z655
1Z656
1Z657
1Z658
1Z659
1Z660
1Z661
1Z662
1Z663
1Z664
1Z665
1Z666
1Z667
1Z668
1Z669
1Z670
1Z671
1Z672
1Z673
1Z674
1Z675
1Z676
1Z677
1Z678
1Z679
1Z680
1Z681
1Z682
1Z683
1Z684
1Z685
1Z686
1Z687
1Z688
1Z689
1Z690
1Z691
1Z692
1Z693
1Z694
1Z695
1Z696
1Z697
1Z698
1Z699
1Z700
1Z701
1Z702
1Z703
1Z704
1Z705
1Z706
1Z707
1Z708
1Z709
1Z710
1Z711
1Z712
1Z713
1Z714
1Z715
1Z716
1Z717
1Z718
1Z719
1Z720
1Z721
1Z722
1Z723
1Z724
1Z725
1Z726
1Z727
1Z728
1Z729
1Z730
1Z731
1Z732
1Z733
1Z734
1Z735
1Z736
1Z737
1Z738
1Z739
1Z740
1Z741
1Z742
1Z743
1Z744
1Z745
1Z746
1Z747
1Z748
1Z749
1Z750
1Z751
1Z752
1Z753
1Z754
1Z755
1Z756
1Z757
1Z758
1Z759
1Z760
1Z761
1Z762
1Z763
1Z764
1Z765
1Z766
1Z767
1Z768
1Z769
1Z770
1Z771
1Z772
1Z773
1Z774
1Z775
1Z776
1Z777
1Z778
1Z779
1Z780
1Z781
1Z782
1Z783
1Z784
1Z785
1Z786
1Z787
1Z788
1Z789
1Z790
1Z791
1Z792
1Z793
1Z794
1Z795
1Z796
1Z797
1Z798
1Z799
1Z800
1Z801
1Z802
1Z803
1Z804
1Z805
1Z806
1Z807
1Z808
1Z809
1Z810
1Z811
1Z812
1Z813
1Z814
1Z815
1Z816
1Z817
1Z818
1Z819
1Z820
1Z821
1Z822
1Z823
1Z824
1Z825
1Z826
1Z827
1Z828
1Z829
1Z830
1Z831
1Z832
1Z833
1Z834
1Z835
1Z836
1Z837
1Z838
1Z839
1Z840
1Z841
1Z842
1Z843
1Z844
1Z845
1Z846
1Z847
1Z848
1Z849
1Z850
1Z851
1Z852
1Z853
1Z854
1Z855
1Z856
1Z857
1Z858
1Z859
1Z860
1Z861
1Z862
1Z863
1Z864
1Z865
1Z866
1Z867
1Z868
1Z869
1Z870
1Z871
1Z872
1Z873
1Z874
1Z875
1Z876
1Z877
1Z878
1Z879
1Z880
1Z881
1Z882
1Z883
1Z884
1Z885
1Z886
1Z887
1Z888
1Z889
1Z890
1Z891
1Z892
1Z893
1Z894
1Z895
1Z896
1Z897
1Z898
1Z899
1Z900
1Z901
1Z902
1Z903
1Z904
1Z905
1Z906
1Z907
1Z908
1Z909
1Z910
1Z911
1Z912
1Z913
1Z914
1Z915
1Z916
1Z917
1Z918
1Z919
1Z920
1Z921
1Z922
1Z923
1Z924
1Z925
1Z926
1Z927
1Z928
1Z929
1Z930
1Z931
1Z932
1Z933
1Z934
1Z935
1Z936
1Z937
1Z938
1Z939
1Z940
1Z941
1Z942
1Z943
1Z944
1Z945
1Z946
1Z947
1Z948
1Z949
1Z950
1Z951
1Z952
1Z953
1Z954
1Z955
1Z956
1Z957
1Z958
1Z959
1Z960
1Z961
1Z962
1Z963
1Z964
1Z965
1Z966
1Z967
1Z968
1Z969
1Z970
1Z971
1Z972
1Z973
1Z974
1Z975
1Z976
1Z977
1Z978
1Z979
1Z980
1Z981
1Z982
1Z983
1Z984
1Z985
1Z986
1Z987
1Z988
1Z989
1Z990
1Z991
1Z992
1Z993
1Z994
1Z995
1Z996
1Z997
1Z998
1Z999
1Z1000
SCALE: NONE
SHEET 2 of 3
D
MCD
2001051
B



NOTES:
UNLESS OTHERWISE SPECIFIED

1. RESISTOR VALUES ARE IN OHMS, ¼W, ±5%.

[2] THIS PIN OUT ARRANGEMENT REPRESENTS 6 INDIVIDUAL 7 SEGMENT LED DISPLAYS (TIL 313). ALL ANODE CONNECTIONS OF THE CORRESPONDING SEGMENTS ARE WIRED TOGETHER. CATHODES ARE WIRED SEPARATELY TO EACH TRANSISTOR (Q10-Q15)

[3] JUMPER 5 TO 3 AND 6 TO 8 WHEN BUS EXPANSION CIRCUITS ARE NOT INSTALLED.

SPARE GATE COUNT		
TYPE	REF DES.	QTY

REF DESIGNATIONS		
LIST USED	NOT USED	
C30		
A17		
R35		
Q16		
D86		
J3		

400117	SDK 85

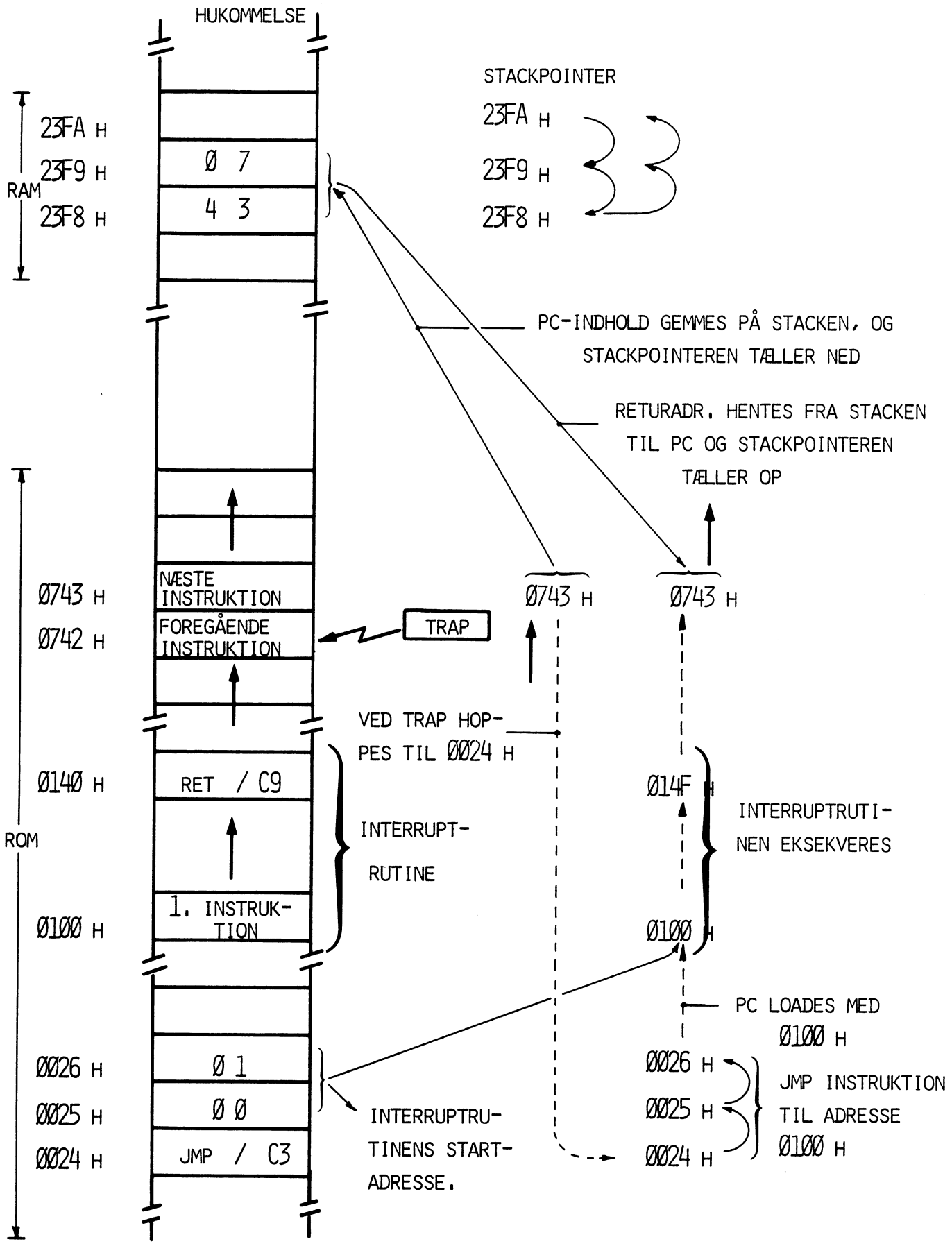
SCALE NONE

intel

SCHEMATIC
SYSTEM DESIGN KIT

SIZE D | DEPT MCD | 9000105 | REV B

SHEET 1 OF 3


HUKOMMELSE
STACKPOINTER
23FA H
23F9 H
23F8 H
RAM
23FA H
23F9 H
23F8 H
Ø 7
4 3
PC-INDHOLD GEMMES PÅ STACKEN, OG STACKPOINTEREN TÆLLER NED
RETURADR. HENTES FRA STACKEN TIL PC OG STACKPOINTEREN TÆLLER OP
ROM
Ø743 H
Ø742 H
NÆSTE INSTRUKTION
FOREGÅENDE INSTRUKTION
TRAP
Ø743 H
Ø743 H
VED TRAP HOP-PES TIL ØØ24 H
Ø140 H
RET / C9
INTERRUPT-RUTINE
Ø14F H
INTERRUPTRUTI-NEN EKSEKVERES
Ø100 H
1. INSTRUK-TION
Ø100 H
PC LOADES MED Ø100 H
ØØ26 H
ØØ25 H
ØØ24 H
Ø 1
Ø Ø
JMP / C3
INTERRUPTRU-TINENS START-ADRESSE.
ØØ26 H
ØØ25 H
ØØ24 H
JMP INSTRUKTION TIL ADRESSE Ø100 H

SW-LØSNING:

Bit 7 = 1			
PORT 1	PORT 2	PORT 3	PORT 4
PROCES 1	PROCES 2	PROCES 3	PROCES 4

SW/HW-LØSNING

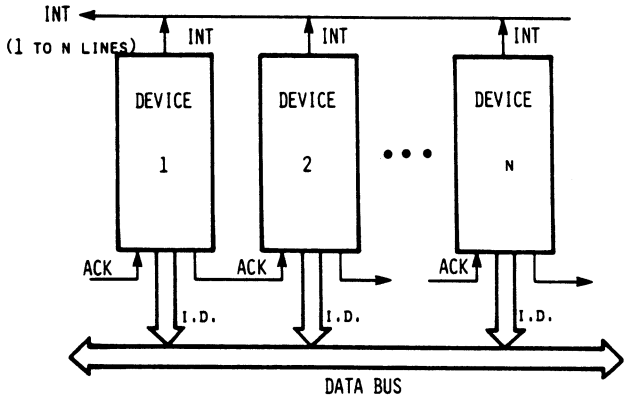


Fig. 3-37: Daisy Chain Technique

HW-LØSNING

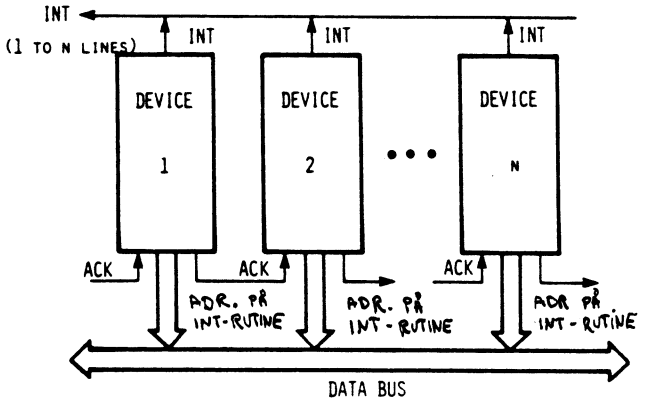


Fig. 3-37: Daisy Chain Technique

Z8400
Z80® CPU Central Processing Unit

Zilog

Product Specification

September 1983

Features

- The instruction set contains 158 instructions. The 78 instructions of the 8080A are included as a subset; 8080A software compatibility is maintained.
- Eight MHz, 6 MHz, 4 MHz and 2.5 MHz clocks for the Z80H, Z80B, Z80A, and Z80 CPU result in rapid instruction execution with consequent high data throughput.
- The extensive instruction set includes string, bit, byte, and word operations. Block searches and block transfers together with indexed and relative addressing result in the most powerful data handling capabilities in the microcomputer industry.
- The Z80 microprocessors and associated family of peripheral controllers are linked by a vectored interrupt system. This system may be daisy-chained to allow implementation of a priority interrupt scheme. Little, if any, additional logic is required for daisy-chaining.
- Duplicate sets of both general-purpose and flag registers are provided, easing the design and operation of system software through single-context switching, background-foreground programming, and single-level interrupt processing. In addition, two 16-bit index registers facilitate program processing of tables and arrays.
- There are three modes of high-speed interrupt processing: 8080 similar, non-Z80 peripheral device, and Z80 Family peripheral with or without daisy chain.
- On-chip dynamic memory refresh counter.

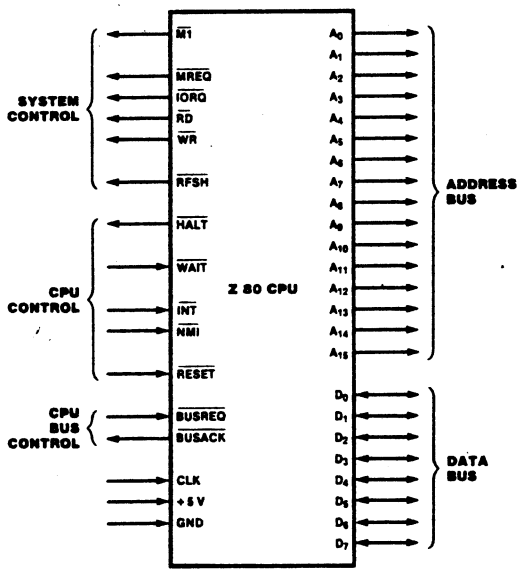


Figure 1. Pin Functions

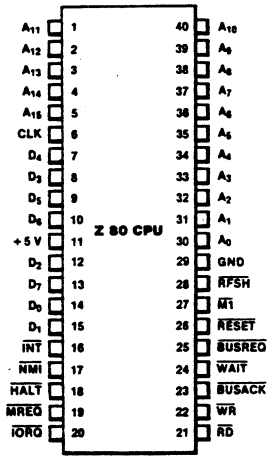


Figure 2. Pin Assignments

Interrupts: General Operation (Continued)

Non-Maskable Interrupt ($\overline{NMI}$). The non-maskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU. $\overline{NMI}$ is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shut-down after power failure has been detected. After recognition of the $\overline{NMI}$ signal (providing $\overline{BUSREQ}$ is not active), the CPU jumps to restart location 0066H. Normally, software starting at this address contains the interrupt service routing.

Maskable Interrupt ($\overline{INT}$). Regardless of the interrupt mode set by the user, the Z80 response to a maskable interrupt input follows a common timing cycle. After the interrupt has been detected by the CPU (provided that interrupts are enabled and $\overline{BUSREQ}$ is not active) a special interrupt processing cycle begins. This is a special fetch ($\overline{M1}$) cycle in which $\overline{IORQ}$ becomes active rather than $\overline{MREQ}$, as in normal $\overline{M1}$ cycle. In addition, this special $\overline{M1}$ cycle is automatically extended by two $\overline{WAIT}$ states, to allow for the time required to acknowledge the interrupt request.

Mode 0 Interrupt Operation. This mode is similar to the 8080 microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus. This is normally a Restart instruction, which will initiate a call to the selected one of eight restart locations in page zero of memory. Unlike the 8080, the Z80 CPU responds to the Call instruction with only one interrupt acknowledge cycle followed by two memory read cycles.

Mode 1 Interrupt Operation. Mode 1 operation is very similar to that for the $\overline{NMI}$. The principal difference is that the Mode 1 interrupt has a restart location of 0038H only.

Mode 2 Interrupt Operation. This interrupt mode has been designed to utilize most effectively the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address of the interrupt service routine. It does this by placing an 8-bit vector on the data bus during the interrupt acknowledge cycle. The CPU forms a pointer using this byte as the lower 8-bits and the contents of the I register as the upper 8-bits. This points to an entry in a table of addresses for interrupt service routines. The CPU then jumps to the routine at that address. This flexibility in selecting the interrupt service routine address allows the peripheral device to use several different types of service routines. These routines may be located at any available location in memory. Since the interrupting device supplies the low-order byte of the 2-byte vector, bit 0 (A_0) must be a zero.

Interrupt Priority (Daisy Chaining and Nested Interrupts). The interrupt priority of each peripheral device is determined by its physical location within a daisy-chain configuration. Each device in the chain has an interrupt enable input line (IEI) and an interrupt enable output line (IEO), which is fed to the next lower priority device. The first device in the daisy chain has its IEI input hardwired to a High level. The first device has highest priority, while each succeeding device has a corresponding lower priority. This arrangement permits the CPU to select the highest priority interrupt from several simultaneously interrupting peripherals.

The interrupting device disables its IEO line to the next lower priority peripheral until it has been serviced. After servicing, its IEO line is raised, allowing lower priority peripherals to demand interrupt servicing.

The Z80 CPU will nest (queue) any pending interrupts or interrupts received while a selected peripheral is being serviced.

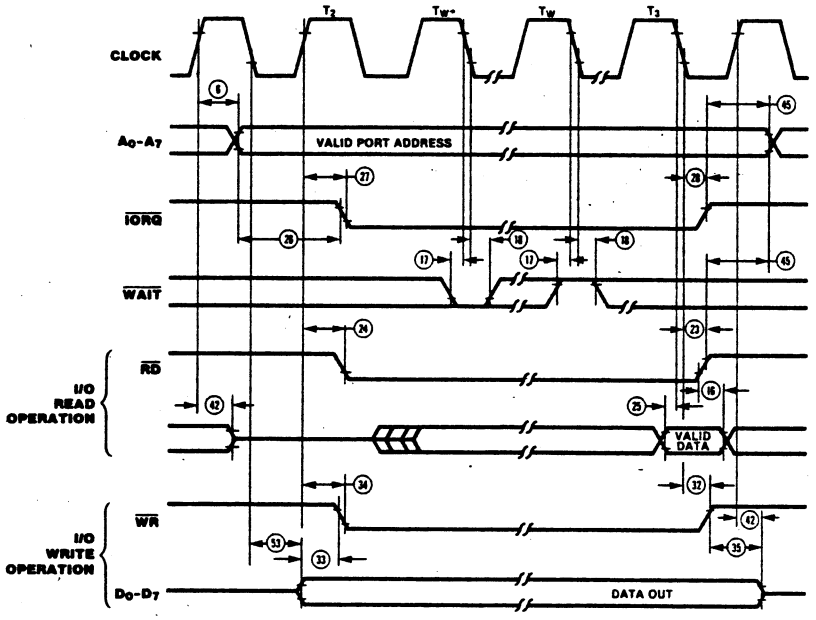
Interrupt Enable/Disable Operation. Two flip-flops, IFF_1 and IFF_2, referred to in the register description are used to signal the CPU interrupt status. Operation of the two flip-flops is described in Table 2. For more details, refer to the *Z80 CPU Technical Manual* and *Z80 Assembly Language Manual.*

Action	IFF_1	IFF_2	Comments
CPU Reset	0	0	Maskable interrupt $\overline{INT}$ disabled
DI instruction execution	0	0	Maskable interrupt $\overline{INT}$ disabled
EI instruction execution	1	1	Maskable interrupt $\overline{INT}$ enabled
LD A,I instruction execution	•	•	IFF_2 → Parity flag
LD A,R instruction execution	•	•	IFF_2 → Parity flag
Accept $\overline{NMI}$	0	IFF_1	IFF_1 → IFF_2 (Maskable interrupt $\overline{INT}$ disabled)
RETN instruction execution	IFF_2	•	IFF_2 → IFF_1 at completion of an $\overline{NMI}$ service routine.

Table 2. State of Flip-Flops

CPU Timing (Continued)

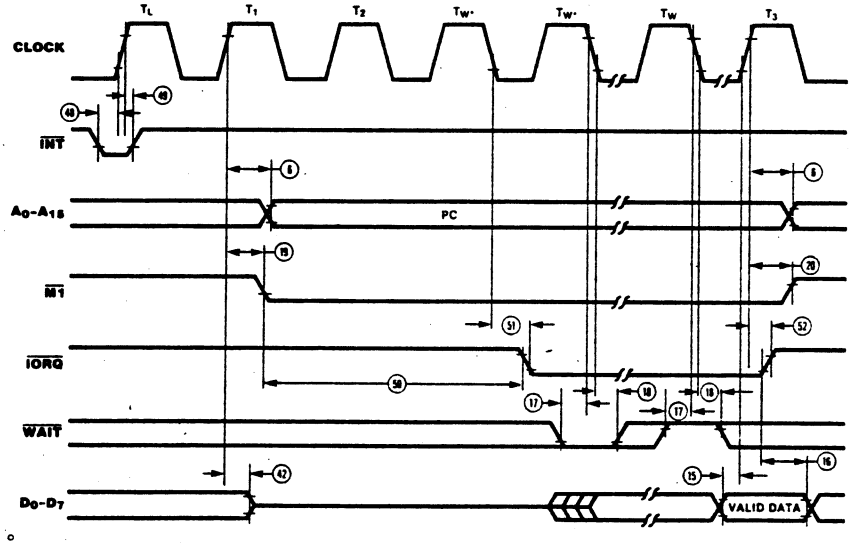
Input or Output Cycles. Figure 7 shows the timing for an I/O read or I/O write operation. During I/O operations, the CPU automatically inserts a single Wait state (T_W). This extra Wait state allows sufficient time for an I/O port to decode the address from the port address lines.



NOTE: T_W* = One Wait cycle automatically inserted by CPU.

Figure 7. Input or Output Cycles

Interrupt Request/Acknowledge Cycle. The CPU samples the interrupt signal with the rising edge of the last clock cycle at the end of any instruction (Figure 8). When an interrupt is accepted, a special $\overline{M1}$ cycle is generated. During this $\overline{M1}$ cycle, $\overline{IORQ}$ becomes active (instead of $\overline{MREQ}$) to indicate that the interrupting device can place an 8-bit vector on the data bus. The CPU automatically adds two Wait states to this cycle.



NOTE: 1) T_L = Last state of previous instruction. 2) Two Wait cycles automatically inserted by CPU(*).

Figure 8. Interrupt Request/Acknowledge Cycle

Z8420
Z80® PIO Parallel Input/Output Controller

Zilog

Product Specification

September 1983

Features

- Provides a direct interface between Z-80 microcomputer systems and peripheral devices.
- Both ports have interrupt-driven handshake for fast response.
- Four programmable operating modes: byte input, byte output, byte input/output (Port A only), and bit input/output.
- Programmable interrupts on peripheral status conditions.
- Standard Z-80 Family bus-request and prioritized interrupt-request daisy chains implemented without external logic.
- The eight Port B outputs can drive Darlington transistors (1.5 mA at 1.5 V).

General Description

The Z-80 PIO Parallel I/O Circuit is a programmable, dual-port device that provides a TTL-compatible interface between peripheral devices and the Z-80 CPU. The CPU configures the Z-80 PIO to interface with a wide range of peripheral devices with no other external logic. Typical peripheral devices that are compatible with the Z-80 PIO include most keyboards, paper tape readers and punches, printers, PROM programmers, etc.

One characteristic of the Z-80 peripheral controllers that separates them from other interface controllers is that all data transfer between the peripheral device and the CPU is accomplished under interrupt control. Thus, the interrupt logic of the PIO permits full use of the efficient interrupt capabilities of the Z-80 CPU during I/O transfers. All logic necessary to implement a fully nested interrupt structure is included in the PIO.

Another feature of the PIO is the ability to interrupt the CPU upon occurrence of specified status conditions in the peripheral device. For example, the PIO can be programmed to interrupt if any specified peripheral alarm conditions should occur. This interrupt capability reduces the time the processor must spend in polling peripheral status.

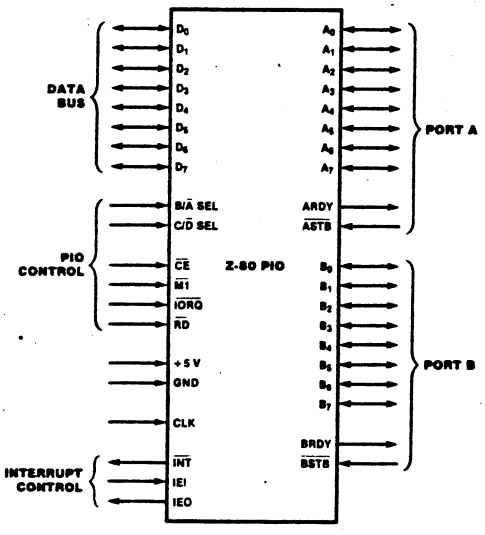


Figure 1. Pin Functions

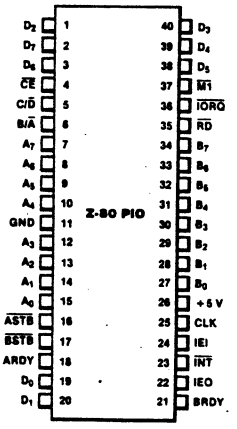


Figure 2. Pin Assignments

General Description (Continued)

The Z-80 PIO interfaces to peripherals via two independent general-purpose I/O ports, designated Port A and Port B. Each port has eight data bits and two handshake signals, Ready and Strobe, which control data transfer. The Ready output indicates to the peripheral that the port is ready for a data transfer. Strobe is an input from the peripheral that indicates when a data transfer has occurred.

Operating Modes. The Z-80 PIO ports can be programmed to operate in four modes: byte output (Mode 0), byte input (Mode 1), byte input/output (Mode 2) and bit input/output (Mode 3).

In Mode 0, either Port A or Port B can be programmed to output data. Both ports have output registers that are individually addressed by the CPU; data can be written to either port at any time. When data is written to a port, an active Ready output indicates to the external device that data is available at the associated port and is ready for transfer to the external device. After the data transfer, the external device responds with an active Strobe input, which generates an interrupt, if enabled.

In Mode 1, either Port A or Port B can be configured in the input mode. Each port has an input register addressed by the CPU. When the CPU reads data from a port, the PIO sets the Ready signal, which is detected by the external device. The external device then places data on the I/O lines and strobes the I/O port, which latches the data into the Port Input Register, resets Ready, and triggers the Interrupt Request, if enabled. The CPU can read the input data at any time, which again sets Ready.

Mode 2 is bidirectional and uses Port A, plus the interrupts and handshake signals from both ports. Port B must be set to Mode 3 and masked off. In operation, Port A is used for both data input and output. Output operation is similar to Mode 0 except that data is allowed out onto the Port A bus only when $\overline{ASTB}$ is Low. For input, operation is similar to Mode 1, except that the data input uses the Port B handshake signals and the Port B interrupt (if enabled).

Both ports can be used in Mode 3. In this mode, the individual bits are defined as either input or output bits. This provides up to eight separate, individually defined bits for each port. During operation, Ready and Strobe are not used. Instead, an interrupt is generated if the condition of one input changes, or if all inputs change. The requirements for generating an interrupt are defined during the programming operation; the active level is specified as either High or Low, and the logic condition is specified as either one input active (OR) or all inputs active (AND). For example, if the port is programmed for active Low inputs and the logic function is AND, then all inputs at the specified port must go Low to generate an interrupt.

Data outputs are controlled by the CPU and can be written or changed at any time.

- Individual bits can be masked off.
- The handshake signals are not used in Mode 3; Ready is held Low, and Strobe is disabled.
- When using the Z-80 PIO interrupts, the Z-80 CPU interrupt mode must be set to Mode 2.

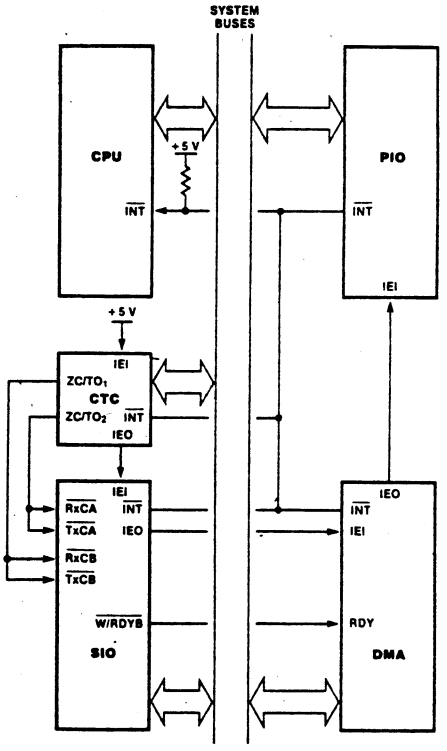


Figure 3. PIO in a Typical Z80 Family Environment

Programming

Mode 0, 1, or 2. *(Byte Input, Output, or Bidirectional).* Programming a port for Mode 0, 1, or 2 requires two words per port. These words are:

A Mode Control Word. Selects the port operating mode (Figure 6). This word may be written any time.

An Interrupt Vector. The Z-80 PIO is designed for use with the Z-80 CPU in interrupt Mode 2 (Figure 7). When interrupts are enabled, the PIO must provide an interrupt vector.

Mode 3. *(Bit Input/Output).* Programming a port for Mode 3 operation requires a control word, a vector (if interrupts are enabled), and three additional words, described as follows:

I/O Register Control. When Mode 3 is selected, the mode control word must be followed by another control word that sets the I/O control register, which in turn defines which port lines are inputs and which are outputs (Figure 8).

Interrupt Control Word. In Mode 3, handshake is not used. Interrupts are generated as a logic function of the input signal levels. The interrupt control word sets the logic conditions and the logic levels required for generating an interrupt. Two logic conditions or functions are available: AND (if all input bits change to the active level, an interrupt is triggered), and OR (if any one of the input bits changes to the active level, an interrupt is triggered). Bit D_6 sets the logic function, as shown in Figure 9. The active level of the input bits can be set either High or Low. The active level is controlled by Bit D_5.

Mask Control Word. This word sets the mask control register, allowing any unused bits to be masked off. If any bits are to be masked, then D_4 must be set. When D_4 is set, the next word written to the port must be a mask control word (Figure 10).

Interrupt Disable. There is one other control word which can be used to enable or disable a port interrupt. It can be used without changing the rest of the interrupt control word (Figure 11).

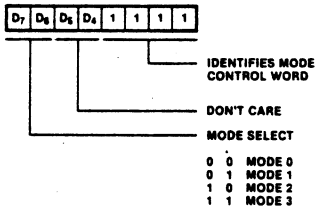


Figure 6. Mode Control Word

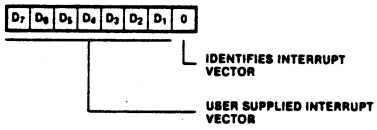


Figure 7. Interrupt Vector Word

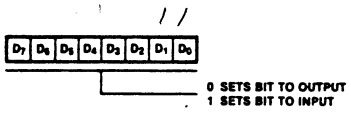


Figure 8. I/O Register Control Word

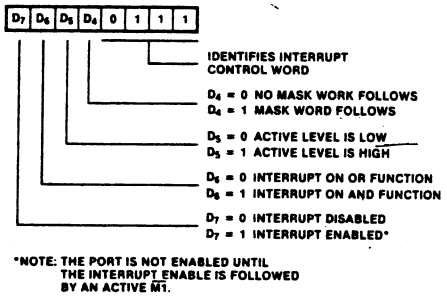


Figure 9. Interrupt Control Word

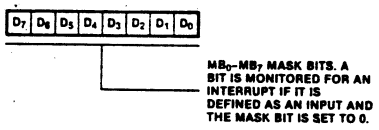


Figure 10. Mask Control Word

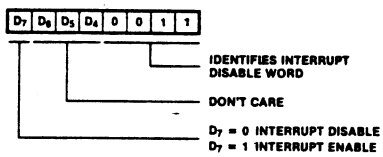


Figure 11. Interrupt Disable Word

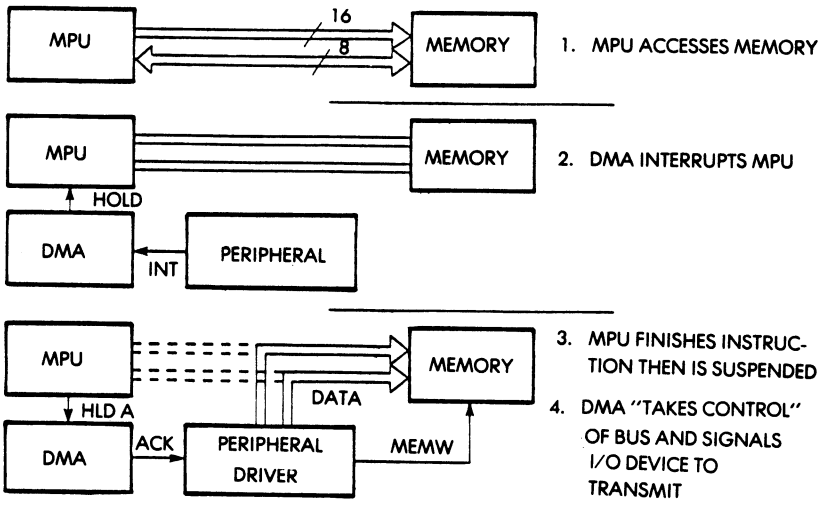


Fig. 3-43: DMA Controller Operation

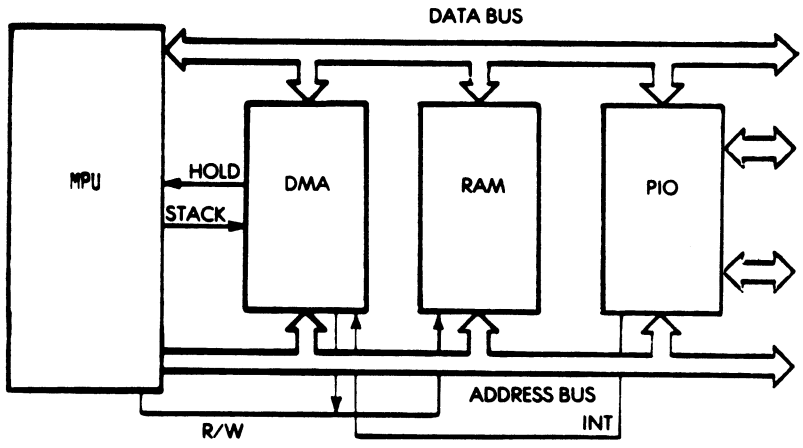


Fig. 3-49: DMA Block Diagram

Z8410
Z80® DMA Direct Memory Access Controller

Zilog

Product Specification

September 1983

Features

- Transfers, searches and search/transfers in Byte-at-a-Time, Burst or Continuous modes. Cycle length and edge timing can be programmed to match the speed of any port.
- Dual port addresses (source and destination) generated for memory-to-I/O, memory-to-memory, or I/O-to-I/O operations. Addresses may be fixed or automatically incremented/decremented.
- Next-operation loading without disturbing current operations via buffered starting-address registers. An entire previous sequence can be repeated automatically.
- Extensive programmability of functions. CPU can read complete channel status.
- Standard Z-80 Family bus-request and prioritized interrupt-request daisy chains implemented without external logic. Sophisticated, internally modifiable interrupt vectoring.
- Direct interfacing to system buses without external logic.

General Description

The Z-80 DMA (Direct Memory Access) is a powerful and versatile device for controlling and processing transfers of data. Its basic function of managing CPU-independent transfers between two ports is augmented by an array of features that optimize transfer speed and control with little or no external logic in systems using an 8- or 16-bit data bus and a 16-bit address bus.

Transfers can be done between any two ports (source and destination), including memory-to-I/O, memory-to-memory, and I/O-to-I/O. Dual port addresses are automatically generated for each transaction and may be either fixed or incrementing/decrementing. In addition, bit-maskable byte searches can be performed either concurrently with transfers or as an operation in itself.

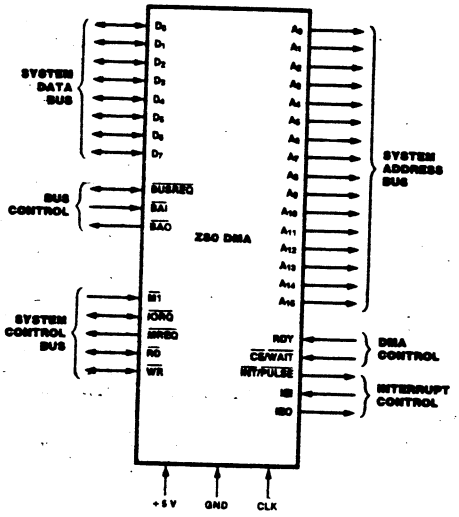


Figure 1. Pin Functions

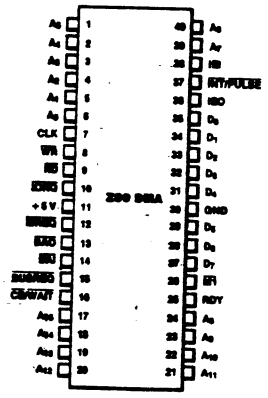


Figure 2. Pin Assignments

General Description (Continued)

The Z-80 DMA contains direct interfacing to and independent control of system buses, as well as sophisticated bus and interrupt controls. Many programmable features, including variable cycle timing and auto-restart, minimize CPU software overhead. They are especially useful in adapting this special-purpose transfer processor to a broad variety of memory, I/O and CPU environments.

The Z-80 DMA is an n-channel silicon-gate depletion-load device packaged in a 40-pin plastic or ceramic DIP. It uses a single +5 V power supply and the standard Z-80 Family single-phase clock.

Functional Description

Classes of Operation. The Z-80 DMA has three basic classes of operation:

- Transfers of data between two ports (memory or I/O peripheral)
- Searches for a particular 8-bit maskable byte at a single port in memory or an I/O peripheral
- Combined transfers with simultaneous search between two ports

Figure 4 illustrates the basic functions served by these classes of operation.

During a transfer, the DMA assumes control of the system address and data buses. Data is read from one addressable port and written to the other addressable port, byte by byte. The ports may be programmed to be either system main memory or peripheral I/O devices. Thus, a block of data may be written from one peripheral to another, from one area of main memory to another, or from a peripheral to main memory and vice versa.

During a search-only operation, data is read from the source port and compared byte by byte with a DMA-internal register containing a programmable match byte. This match byte may optionally be masked so that only certain bits within the match byte are compared. Search rates up to 1.25M bytes per second can be obtained with the 2.5 MHz Z-80 DMA or 2M bytes per second with the 4 MHz Z-80A DMA.

In combined searches and transfers, data is transferred between two ports while simultaneously searching for a bit-maskable byte match.

Data transfers or searches can be programmed to stop or interrupt under various conditions. In addition, CPU-readable status bits can be programmed to reflect the condition.

Modes of Operation. The Z-80 DMA can be programmed to operate in one of three transfer and/or search modes:

- *Byte-at-a-Time:* data operations are performed one byte at a time. Between each byte operation the system buses are released to the CPU. The buses are requested again for each succeeding byte operation.
- *Burst:* data operations continue until a port's Ready line to the DMA goes inactive. The DMA then stops and releases the system buses after completing its current byte operation.
- *Continuous:* data operations continue until the end of the programmed block of data is reached before the system buses are released. If a port's Ready line goes inactive before this occurs, the DMA simply pauses until the Ready line comes active again.

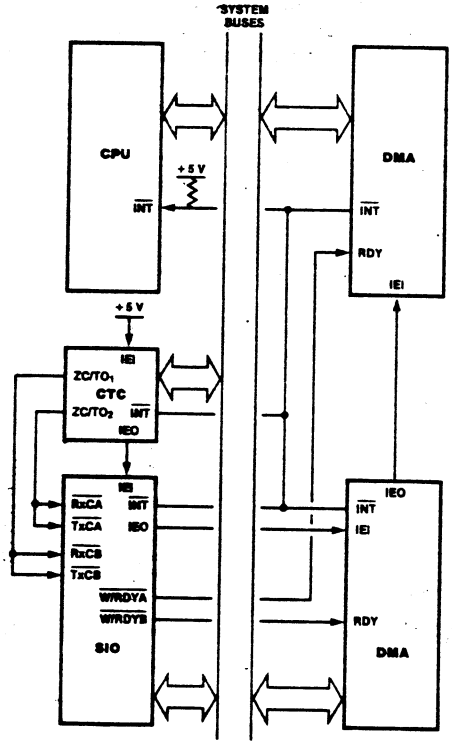


Figure 3. Typical Z-80 Environment

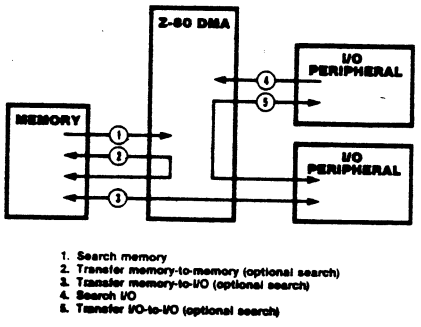


Figure 4. Basic Functions of the Z-80 DMA

Functional Description (Continued)

In all modes, once a byte of data is read into the DMA, the operation on the byte will be completed in an orderly fashion, regardless of the state of other signals (including a port's Ready line).

Due to the DMA's high-speed buffered method of reading data, operations on one byte are not completed until the next byte is read in. This means that total transfer or search block lengths must be two or more bytes, and that block lengths programmed into the DMA must be one byte less than the desired block length (count is N-1 where N is the block length).

Commands and Status. The Z-80 DMA has several writable control registers and readable status registers available to the CPU. Control bytes can be written to the DMA whenever the DMA is not controlling the system buses, but the act of writing a control byte to the DMA disables the DMA until it is again enabled by a specific command. Status bytes can also be read at any such time, but writing the Read Status Byte command or the Initiate Read Sequence command disables the DMA.

Control bytes to the DMA include those which effect immediate command actions such as enable, disable, reset, load starting-address buffers, continue, clear counters, clear status bits and the like. In addition, many mode-setting control bytes can be written, including mode and class of operation, port configuration, starting addresses, block length, address counting rule, match and match-mask byte, interrupt conditions, interrupt vector, status-affects-vector condition, pulse counting, auto restart, Ready-line and Wait-line rules, and read mask.

Readable status registers include a general status byte reflecting Ready-line, end-of-block, byte-match and interrupt conditions, as well as 2-byte registers for the current byte count, Port A address and Port B address.

Variable Cycle. The Z-80 DMA has the unique feature of programmable operation-cycle length. This is valuable in tailoring the DMA to the particular requirements of other system components (fast or slow) and maximizes the data-transfer rate. It also eliminates external logic for signal conditioning.

There are two aspects to the variable cycle feature. First, the entire read and write cycles (periods) associated with the source and destination ports can be independently programmed as 2, 3 or 4 T-cycles long (more if Wait cycles are used), thereby increasing or decreasing the speed with which all DMA signals change (Figure 5).

Second, the four signals in each port specifically associated with transfers of data (I/O Request, Memory Request, Read, and Write) can each have its active trailing edge terminated one-half T-cycle early. This adds a further dimension of flexibility and speed, allowing such things as shorter-than-normal Read or Write signals that go inactive before data starts to change.

Address Generation. Two 16-bit addresses are generated by the Z-80 DMA for every transfer operation, one address for the source port and another for the destination port. Each address can be either variable or fixed. Variable addresses can increment or decrement from the programmed starting address. The fixed-address capability eliminates the need for separate enabling wires to I/O ports.

Port addresses are multiplexed onto the system address bus, depending on whether the DMA is reading the source port or writing to the destination port. Two readable address counters (2 bytes each) keep the current address of each port.

Auto Restart. The starting addresses of either port can be reloaded automatically at the end of a block. This option is selected by the Auto Restart control bit. The byte counter is cleared when the addresses are reloaded.

The Auto Restart feature relieves the CPU of software overhead for repetitive operations such as CRT refresh and many others. Moreover, when the CPU has access to the buses during byte-at-a-time or burst transfers, different starting addresses can be written into buffer registers during transfers, causing the Auto Restart to begin at a new location.

Interrupts. The Z-80 DMA can be programmed to interrupt the CPU on three conditions:

- Interrupt on Ready (before requesting bus)
- Interrupt on Match
- Interrupt on End of Block

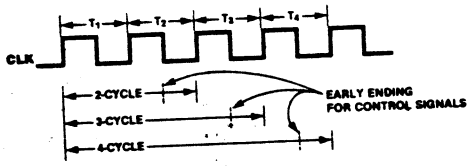


Figure 5. Variable Cycle Length

Functional Description (Continued)

Any of these interrupts cause an interrupt-pending status bit to be set, and each of them can optionally alter the DMA's interrupt vector. Due to the buffered constraint mentioned under "Modes of Operation," interrupts on Match at End of Block are caused by matches to the byte just prior to the last byte in the block.

The DMA shares the Z-80 Family's elaborate interrupt scheme, which provides fast interrupt service in real-time applications. In a Z-80 CPU environment, the DMA passes its internally modifiable 8-bit interrupt vector to the CPU, which adds an additional eight bits to form the memory address of the interrupt-routine table. This table contains the address of the beginning of the interrupt routine itself. In this process, CPU control is transferred directly to the interrupt routine, so that the next instruction executed after an interrupt acknowledge is the first instruction of the interrupt routine itself.

Pulse Generation. External devices can keep track of how many bytes have been transferred by using the DMA's pulse output, which provides a signal at 256-byte intervals. The interval sequence may be offset at the beginning by 1 to 255 bytes.

The Interrupt line outputs the pulse signal in a manner that prevents misinterpretation by the CPU as an interrupt request, since it only appears when the Bus Request and Bus Acknowledge lines are both active.

Pin Description

A_0-A_{15}. *System Address Bus* (output, 3-state). Addresses generated by the DMA are sent to both source and destination ports (main memory or I/O peripherals) on these lines.

$\overline{BAI}$. *Bus Acknowledge In* (input, active Low). Signals that the system buses have been released for DMA control. In multiple-DMA configurations, the $\overline{BAI}$ pin of the highest priority DMA is normally connected to the Bus Acknowledge pin of the CPU. Lower-priority DMAs have their $\overline{BAI}$ connected to the $\overline{BAO}$ of a higher-priority DMA.

$\overline{BAO}$. *Bus Acknowledge Out* (output, active Low). In a multiple-DMA configuration, this pin signals that no other higher-priority DMA has requested the system buses. $\overline{BAI}$ and $\overline{BAO}$ form a daisy chain for multiple-DMA priority resolution over bus control.

$\overline{BUSREQ}$. *Bus Request* (bidirectional, active Low, open drain). As an output, it sends requests for control of the system address bus, data bus and control bus to the CPU. As an input, when multiple DMAs are strung together in a priority daisy chain via $\overline{BAI}$ and $\overline{BAO}$, it senses when another DMA has requested the buses and causes this DMA to refrain from bus requesting until the other DMA is finished. Because it is a bidirectional pin, there cannot be any buffers between this DMA and any other DMA. It can, however, have a buffer between it and the CPU because it is unidirectional into the CPU. A pull-up resistor is connected to this pin.

$\overline{CE}$/$\overline{WAIT}$. *Chip Enable and Wait* (input, active Low). Normally this functions only as a $\overline{CE}$ line, but it can also be programmed to serve a $\overline{WAIT}$ function. As a $\overline{CE}$ line from the CPU, it becomes active when $\overline{WR}$ and $\overline{IORQ}$ are active and the I/O port address on the system address bus is the DMA's address, thereby allowing a transfer of control or command bytes from the CPU to the DMA. As a $\overline{WAIT}$ line from memory or I/O devices, after the DMA has received a bus-request acknowledge from the CPU, it causes wait states to be inserted in the DMA's operation cycles thereby slowing the DMA to a speed that matches the memory or I/O device.

CLK. *System Clock* (input). Standard Z-80 single-phase clock at 2.5 MHz (Z-80 DMA) or 4.0 MHz (Z-80A DMA). For slower system clocks, a TTL gate with a pullup resistor may be adequate to meet the timing and voltage level specification. For higher-speed systems, use a clock driver with an active pullup to meet the V_{IH} specification and risetime requirements. In all cases there should be a resistive pullup to the power supply of 10K ohms (max) to ensure proper power when the DMA is reset.

D_0-D_7. *System Data Bus* (bidirectional, 3-state). Commands from the CPU, DMA status, and data from memory or I/O peripherals are transferred on these lines.

IEI. *Interrupt Enable In* (input, active High). This is used with IEO to form a priority daisy chain when there is more than one interrupt-driven device. A High on this line indicates that no other device of higher priority is being serviced by a CPU interrupt service routine.

IEO. *Interrupt Enable Out* (output, active High). IEO is High only if IEI is High and the CPU is not servicing an interrupt from this DMA. Thus, this signal blocks lower-priority devices from interrupting while a higher-priority device is being serviced by its CPU interrupt service routine.

Pin Description (Continued)

$\overline{INT}$/PULSE. *Interrupt Request* (output, active Low, open drain). This requests a CPU interrupt. The CPU acknowledges the interrupt by pulling its $\overline{IORQ}$ output Low during an $\overline{M1}$ cycle. It is typically connected to the $\overline{INT}$ pin of the CPU with a pullup resistor and tied to all other $\overline{INT}$ pins in the system. This pin can also be used to generate periodic pulses to an external device. It can be used this way only when the DMA is bus master (i.e., the CPU's $\overline{BUSREQ}$ and $\overline{BUSACK}$ lines are both Low and the CPU cannot see interrupts).

$\overline{IORQ}$. *Input/Output Request* (bidirectional, active Low, 3-state). As an input, this indicates that the lower half of the address bus holds a valid I/O port address for transfer of control or status bytes from or to the CPU, respectively; this DMA is the addressed port if its $\overline{CE}$ pin and its $\overline{WR}$ or $\overline{RD}$ pins are simultaneously active. As an output, after the DMA has taken control of the system buses, it indicates that the 8-bit or 16-bit address bus holds a valid port address for another I/O device involved in a DMA transfer of data. When $\overline{IORQ}$ and $\overline{M1}$ are both active simultaneously, an interrupt acknowledge is indicated.

$\overline{M1}$. *Machine Cycle One* (input, active Low). Indicates that the current CPU machine cycle is an instruction fetch. It is used by the DMA to decode the return-from-interrupt instruction (RETI) (ED-4D) sent by the CPU. During two-byte instruction fetches, $\overline{M1}$ is active as each opcode byte is fetched. An interrupt acknowledge is indicated when both $\overline{M1}$ and $\overline{IORQ}$ are active.

$\overline{MREQ}$. *Memory Request* (output, active Low, 3-state). This indicates that the address bus holds a valid address for a memory read or write operation. After the DMA has taken control of the system buses, it indicates a DMA transfer request from or to memory.

$\overline{RD}$. *Read* (bidirectional, active Low, 3-state). As an input, this indicates that the CPU wants to read status bytes from the DMA's read registers. As an output, after the DMA has taken control of the system buses, it indicates a DMA-controlled read from a memory or I/O port address.

RDY. *Ready* (input, programmable active Low or High). This is monitored by the DMA to determine when a peripheral device associated with a DMA port is ready for a read or write operation. Depending on the mode of DMA operation (Byte, Burst or Continuous), the RDY line indirectly controls DMA activity by causing the $\overline{BUSREQ}$ line to go Low or High.

$\overline{WR}$. *Write* (bidirectional, active Low, 3-state). As an input, this indicates that the CPU wants to write control or command bytes to the DMA write registers. As an output, after the DMA has taken control of the system buses, it indicates a DMA-controlled write to a memory or I/O port address.

Internal Structure

The internal structure of the Z-80 DMA includes driver and receiver circuitry for interfacing with an 8-bit system data bus, a 16-bit system address bus, and system control lines (Figure 6). In a Z-80 CPU environment, the DMA can be tied directly to the analogous pins on the CPU (Figure 7) with no additional buffering, except for the $\overline{CE}$/$\overline{WAIT}$ line.

The DMA's internal data bus interfaces with the system data bus and services all internal logic and registers. Addresses generated from this logic for Ports A and B (source and destination) of the DMA's single transfer channel are multiplexed onto the system address bus.

Specialized logic circuits in the DMA are dedicated to the various functions of external bus interfacing, internal bus control, byte matching, byte counting, periodic pulse generation, CPU interrupts, bus requests, and address generation. A set of twenty-one writable control registers and seven readable status registers provides the means by which the CPU governs and monitors the activities of these logic circuits. All registers are eight bits wide, with double-byte information stored in adjacent registers. The two address counters (two bytes each) for Ports A and B are buffered by the two starting addresses.

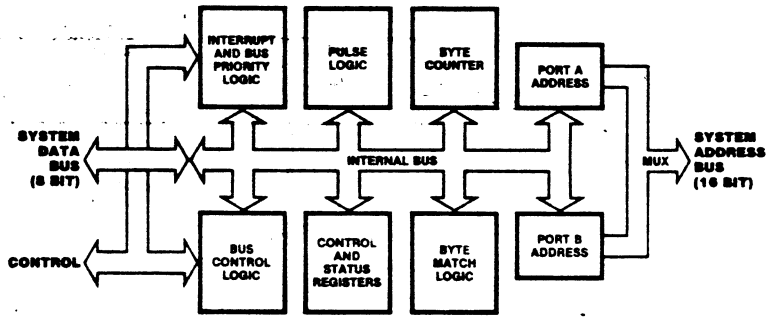


Figure 6. Block Diagram

Internal Structure (Continued)

The 21 writable control registers are organized into seven base-register groups, most of which have multiple registers. The base registers in each writable group contain both control/command bits and pointer bits that can be set to address other registers within the group. The seven readable status registers have no analogous second-level registers.

The registers are designated as follows, according to their base-register groups:

WR0-WR6 — Write Register groups 0 through 6 (7 base registers plus 14 associated registers)
RR0-RR6 — Read Registers 0 through 6

Writing to a register within a write-register group involves first writing to the base register, with the appropriate pointer bits set, then writing to one or more of the other registers within the group. All seven of the readable status registers are accessed sequentially according to a programmable mask contained in one of the writable registers. The section entitled "Programming" explains this in more detail.

A pipelining scheme is used for reading data in. The programmed block length is the number of bytes compared to the byte counter, which increments at the end of each cycle. In searches, data byte comparisons with the match byte are made during the read cycle of the next byte. Matches are, therefore, discovered only after the next byte is read in.

In multiple-DMA configurations, interrupt-request daisy chains are prioritized by the order in which their IEI and IEO lines are connected (Zilog Application Note 03-0041-01, *The Z-80 Family Program Interrupt Structure*). The system bus, however, may not be pre-empted. Any DMA that gains access to the system bus keeps the bus until it is finished.

Write Registers	
WR0	Base register byte Port A starting address (low byte) Port A starting address (high byte) Block length (low byte) Block length (high byte)
WR1	Base register byte Port A variable-timing byte
WR2	Base register byte Port B variable-timing byte
WR3	Base register byte Mask byte Match byte
WR4	Base register byte Port B starting address (low byte) Port B starting address (high byte) Interrupt control byte Pulse control byte Interrupt vector
WR5	Base register byte
WR6	Base register byte Read mask

Read Registers	
RR0	Status byte
RR1	Byte counter (low byte)
RR2	Byte counter (high byte)
RR3	Port A address counter (low byte)
RR4	Port A address counter (high byte)
RR5	Port B address counter (low byte)
RR6	Port B address counter (high byte)

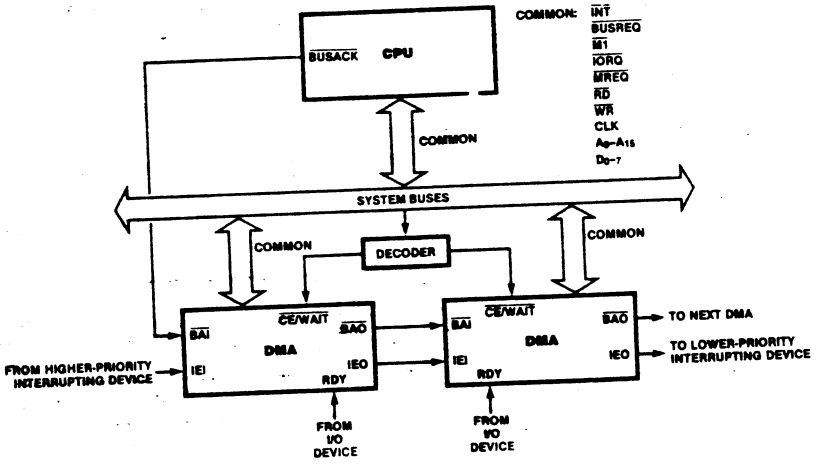


Figure 7. Multiple-DMA Interconnection to the Z-80 CPU

Programming

The Z-80 DMA has two programmable fundamental states: (1) an enabled state, in which it can gain control of the system buses and direct the transfer of data between ports, and (2) a disabled state, in which it can initiate neither bus requests nor data transfers. When the DMA is powered up or reset by any means, it is automatically placed into the disabled state. Program commands can be written to it by the CPU in either state, but this automatically puts the DMA in the disabled state, which is maintained until an enable command is issued by the CPU. The CPU must program the DMA in advance of any data search or transfer by addressing it as an I/O port and sending a sequence of control bytes using an Output instruction (such as OTIR for the Z-80 CPU).

Writing. Control or command bytes are written into one or more of the Write Register groups (WR0-WR6) by first writing to the base register byte in that group. All groups have base registers and most groups have additional associated registers. The associated registers in a group are sequentially accessed by first writing a byte to the base register containing register-group identification and pointer bits (1's) to one or more of that base register's associated registers.

This is illustrated in Figure 8b. In this figure, the sequence in which associated registers within a group can be written to is shown by the vertical position of the associated registers. For example, if a byte written to the DMA contains the bits that identify WR0 (bits D0, D1 and D7), and also contains 1's in the bit positions that point to the associated "Port A Starting Address (low byte)" and "Port A Starting Address (high byte)," then the next two bytes written to the DMA will be stored in these two registers, in that order.

Reading. The Read Registers (RR0-RR6) are read by the CPU by addressing the DMA as an I/O port using an Input instruction (such as INIR for the Z-80 CPU). The readable bytes contain DMA status, byte-counter values, and port addresses since the last DMA reset. The registers are always read in a fixed sequence beginning with RR0 and ending with RR6. However, the register read in this sequence is determined by programming the Read Mask in WR6. The sequence of reading is initialized by writing an Initiate Read Sequence or Set Read Status command to WR6. After a Reset DMA, the sequence must be initialized with the Initiate Read Sequence command or a Read Status command. The sequence of reading all registers that are not excluded by the Read Mask register must be completed before a new Initiate Read Sequence or Read Status command.

Fixed-Address Programming. A special circumstance arises when programming a destination port to have a fixed address. The load command in WR6 only loads a fixed address to a port selected as the source, not to a port selected as the destination. Therefore, a fixed destination address must be loaded by temporarily declaring it a fixed-source address and subsequently declaring the true source as such, thereby implicitly making the other a destination.

The following example illustrates the steps in this procedure, assuming that transfers are to occur from a variable-address source (Port A) to a fixed-address destination (Port B):

1. Temporarily declare Port B as source in WR0.
2. Load Port B address in WR6.
3. Declare Port A as source in WR0.
4. Load Port A address in WR6.
5. Enable DMA in WR6.

Figure 9 illustrates a program to transfer data from memory (Port A) to a peripheral device (Port B). In this example, the Port A memory starting address is 1050_H and the Port B peripheral fixed address is 05_H. Note that the data flow is 1001_H bytes—one more than specified by the block length. The table of DMA commands may be stored in consecutive memory locations and transferred to the DMA with an output instruction such as the Z-80 CPU's OTIR instruction.

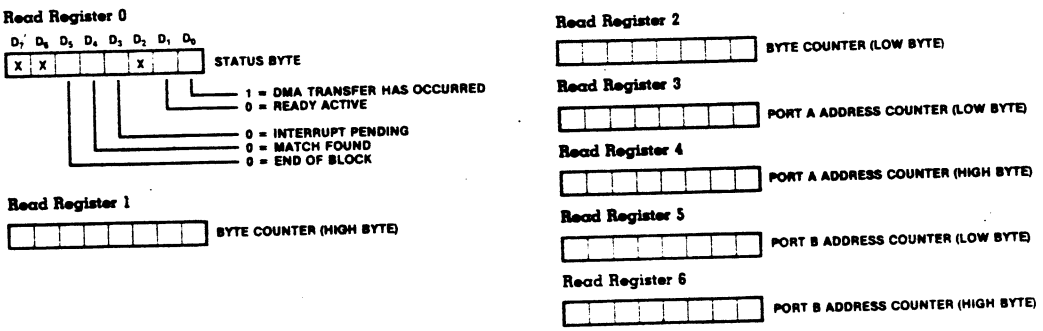


Figure 8a. Read Registers

Programming
(Continued)

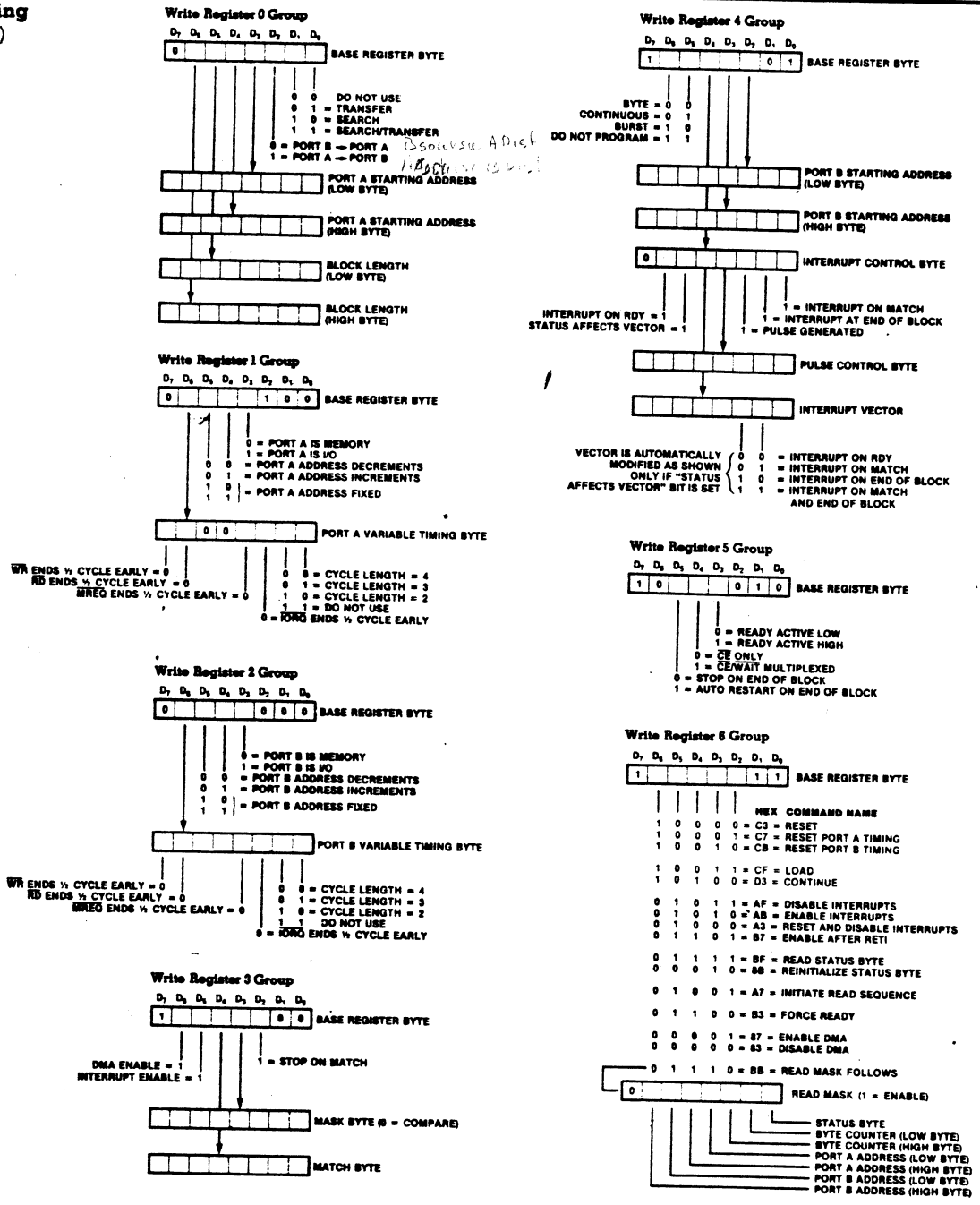


Figure 8b. Write Registers

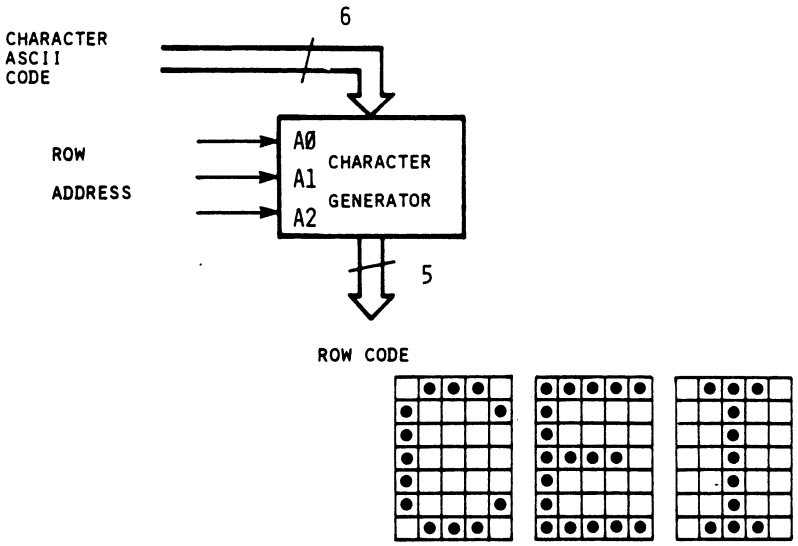


Fig. 4-63: 5X7-Dot Matrix

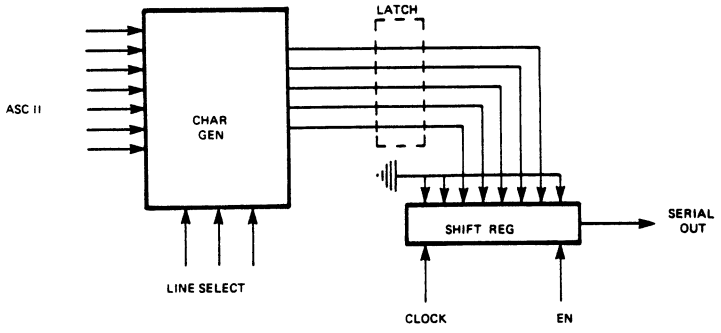


Fig. 4-65: Shift Register Serializes Characters

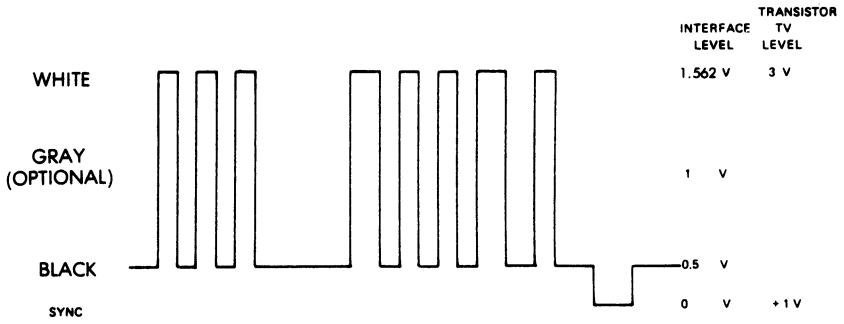


Fig. 4-67: Composite Video and Sync

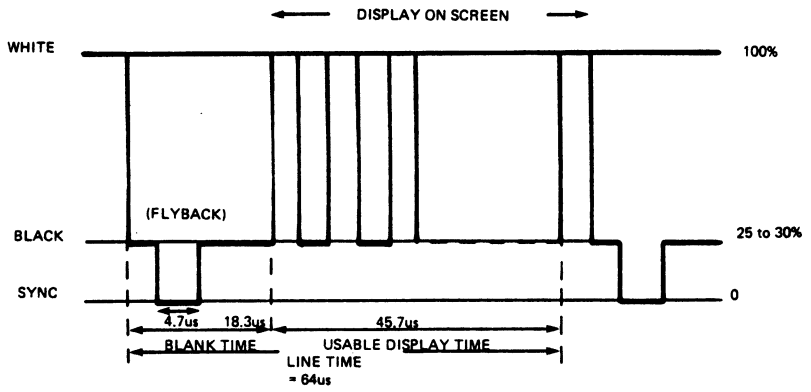


Fig. 4-68: TV Timing

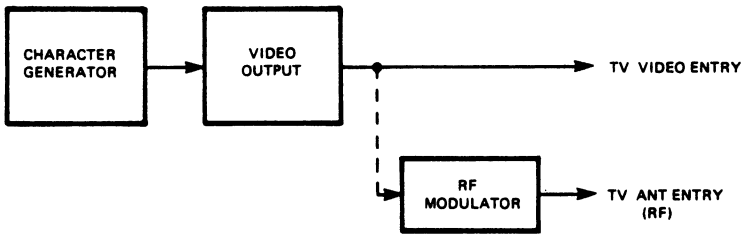
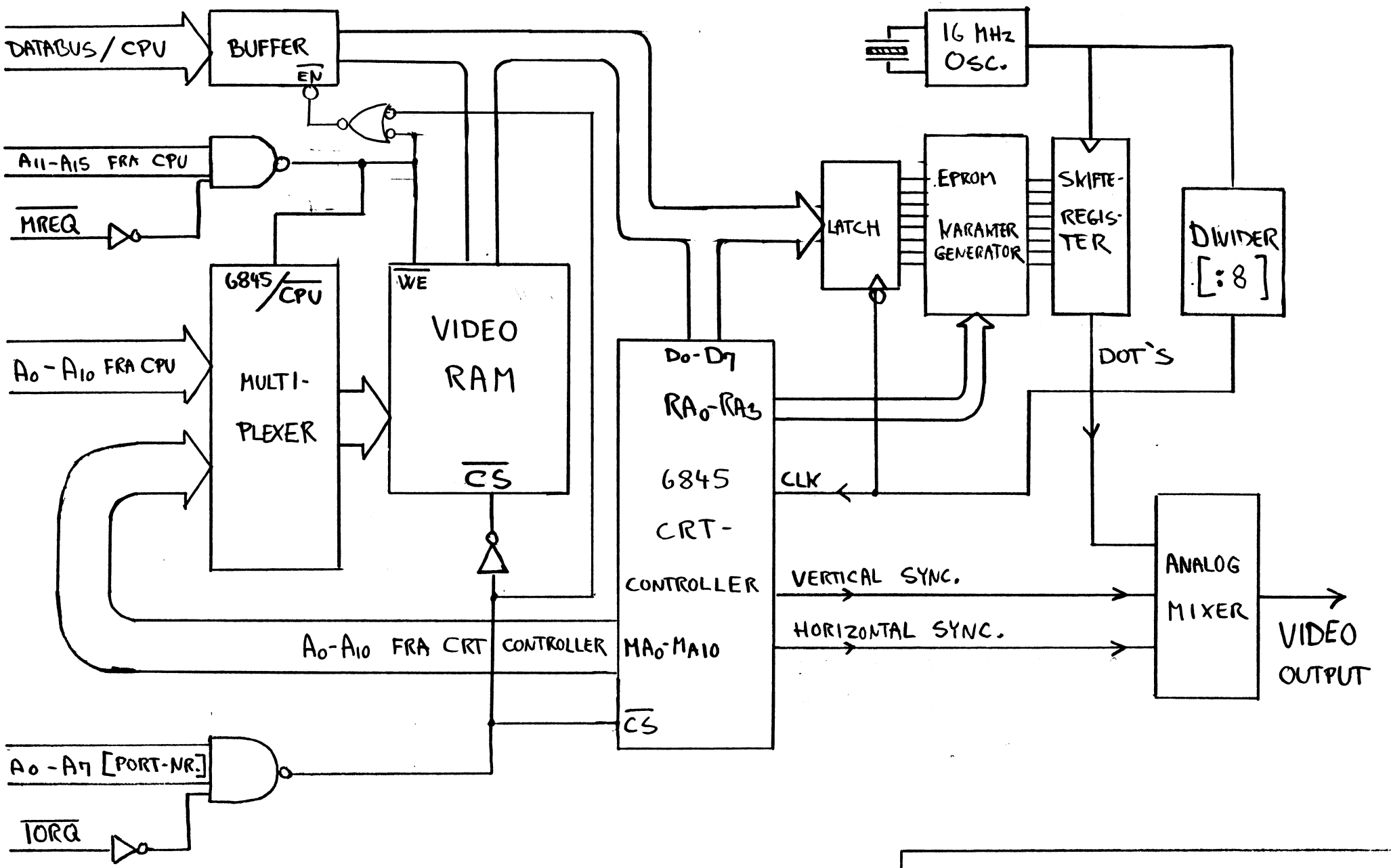


Fig. 4-69: Video vs RF Entry



PRINCIPDIAGRAM / VIDEO-KREDSLØB

MOTOROLA

MC6845 (1.0 MHz)	**MC6845☆1** (1.0 MHz)
MC68A45 (1.5 MHz)	**MC68A45☆1** (1.5 MHz)
MC68B45 (2.0 MHz)	**MC68B45☆1** (2.0 MHz)

CRT CONTROLLER (CRTC)

The MC6845 CRT Controller performs the interface between an MPU and a raster-scan CRT display. It is intended for use in MPU-based controllers for CRT terminals in stand-alone or cluster configurations.

The CRTC is optimized for the hardware/software balance required for maximum flexibility. All keyboard functions, reads, writes, cursor movements, and editing are under processor control. The CRTC provides video timing and refresh memory addressing.

- Useful in Monochrome or Color CRT Applications
- Applications Include "Glass-Teletype," Smart, Programmable, Intelligent CRT Terminals; Video Games, Information Displays
- Alphanumeric, Semi-Graphic, and Full-Graphic Capability
- Fully Programmable Via Processor Data Bus. Timing May Be Generated for Almost Any Alphanumeric Screen Format, e.g., 80 x 24, 72 x 64, 132 x 20
- Single +5 V Supply
- M6800 Compatible Bus Interface
- TTL-Compatible Inputs and Outputs
- Start Address Register Provides Hardware Scroll (by Page, Line, or Character)
- Programmable Cursor Register Allows Control of Cursor Format and Blink Rate
- Light Pen Register
- Refresh (Screen) Memory May be Multiplexed Between the CRTC and the MPU Thus Removing the Requirements for Line Buffers or External DMA Devices
- Programmable Interlace or Non-Interlace Scan Modes
- 14-Bit Refresh Address Allows Up to 16K of Refresh Memory for Use in Character or Semi-Graphic Displays
- 5-Bit Row Address Allows Up to 32 Scan-Line Character Blocks
- By Utilizing Both the Refresh Addresses and the Row Addresses, a 512K Address Space is Available for Use in Graphics Systems
- Refresh Addresses are Provided During Retrace, Allowing the CRTC to Provide Row Addresses to Refresh Dynamic RAMs
- Programmable Skew for Cursor and Display Enable (DE)
- Pin Compatible with the MC6835

MAXIMUM RATINGS

Rating	Symbol	Value	Unit
Supply Voltage	V_{CC}*	−0.3 to +7.0	V
Input Voltage	V_{in}*	−0.3 to +7.0	V
Operating Temperature Range MC6845, MC68A45, MC68B45 MC6845C, MC68A45C, MC68B45C	T_A	T_L to T_H 0 to 70 −40 to +85	°C
Storage Temperature Range	T_{stg}	−55 to +150	°C

THERMAL CHARACTERISTICS

Characteristic	Symbol	Value	Rating
Thermal Resistance Plastic Package Cerdip Package Ceramic Package	θ_{JA}	 100 60 50	°C/W

*This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{CC}$

MOS

(N-CHANNEL, SILICON-GATE)

CRT CONTROLLER (CRTC)

L SUFFIX
CERAMIC PACKAGE
CASE 715

S SUFFIX
CERDIP PACKAGE
CASE 734

P SUFFIX
PLASTIC PACKAGE
CASE 711

☆ = Package Suffix

FIGURE 1 — PIN ASSIGNMENTS

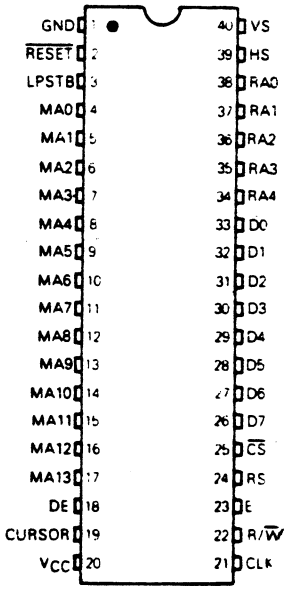
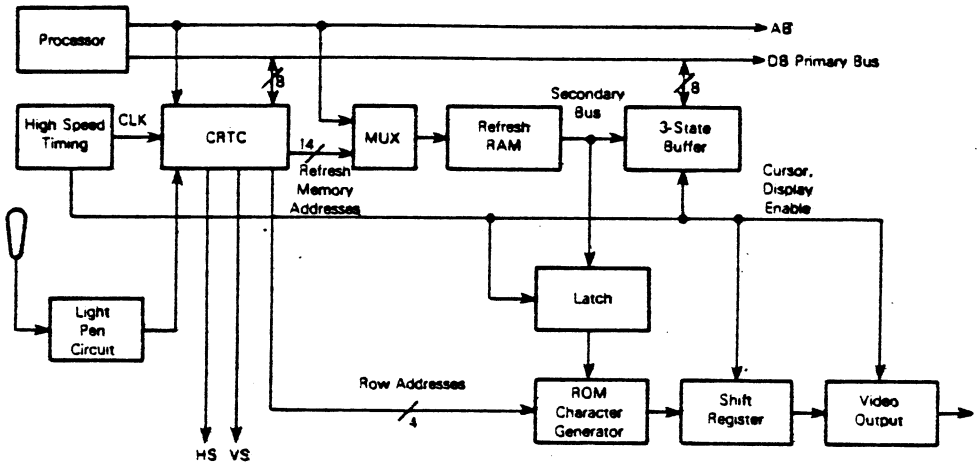


FIGURE 2 – TYPICAL CRT CONTROLLER APPLICATION



RECOMMENDED OPERATING CONDITIONS

Characteristics	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.75	5.0	5.25	V
Input Low Voltage	V_{IL}	−0.3	–	0.8	V
Input High Voltage	V_{IH}	2.0	–	V_{CC}	V

POWER CONSIDERATIONS

The average chip-junction temperature, T_J, in °C can be obtained from:

$$T_J = T_A + (P_D \bullet \theta_{JA}) \quad (1)$$

Where:

T_A = Ambient Temperature, °C

θ_{JA} = Package Thermal Resistance, Junction-to-Ambient, °C/W

$P_D = P_{INT} + P_{PORT}$

$P_{INT} = I_{CC} \times V_{CC}$, Watts – Chip Internal Power

P_{PORT} = Port Power Dissipation, Watts – User Determined

For most applications $P_{PORT} \ll P_{INT}$ and can be neglected. P_{PORT} may become significant if the device is configured to drive Darlington bases or sink LED loads.

An approximate relationship between P_D and T_J (if P_{PORT} is neglected) is:

$$P_D = K \div (T_J + 273°C) \quad (2)$$

Solving equations 1 and 2 for K gives:

$$K = P_D \bullet (T_A + 273°C) + \theta_{JA} \bullet P_D^2 \quad (3)$$

Where K is a constant pertaining to the particular part. K can be determined from equation 3 by measuring P_D (at equilibrium) for a known T_A. Using this value of K the values of P_D and T_J can be obtained by solving equations (1) and (2) iteratively for any value of T_A.

CRTC INTERFACE SYSTEM DESCRIPTION

The CRT controller generates the signals necessary to interface a digital system to a raster scan CRT display. In this type of display, an electron beam starts in the upper left hand corner, moves quickly across the screen and returns. This action is called a horizontal scan. After each horizontal scan the beam is incrementally moved down in the vertical direction until it has reached the bottom. At this point one frame has been displayed, as the beam has made many horizontal scans and one vertical scan.

Two types of raster scanning are used in CRTs, interlace and non-interlace, shown in Figures 7 and 8. Non-interlace scanning consists of one field per frame. The scan lines in Figure 7 are shown as solid lines and the retrace patterns are indicated by the dotted lines. Increasing the number of frames per second will decrease the flicker. Ordinarily, either a 50 or 60 frame per second refresh rate is used to minimize beating between the CRT and the power line frequency. This prevents the displayed data from weaving.

Interlace scanning is used in broadcast TV and on data monitors where high density or high resolution data must be displayed. Two fields, or vertical scans are made down the screen for each single picture or frame. The first field (even field) starts in the upper left hand corner; the second (odd field) in the upper center. Both fields overlap as shown in Figure 8, thus interlacing the two fields into a single frame.

In order to display the characters on the CRT screen the frames must be continually repeated. The data to be displayed is stored in the refresh (screen) memory by the MPU controlling the data processing system. The data is usually written in ASCII code, so it cannot be directly displayed as characters. A character generator ROM is typically used to convert the ASCII codes into the "dot" pattern for every character.

The most common method of generating characters is to create a matrix of dots "x" dots (columns) wide and "y" dots (rows) high. Each character is created by selectively filling in the dots. As "x" and "y" get larger a more detailed character may be created. Two common dot matrices are 5 x 7 and 7 x 9. Many variations of these standards will allow Chinese, Japanese, or Arabic letters instead of English. Since characters require some space between them, a character block larger than the character is typically used, as shown in Figure 9. The figure also shows the corresponding timing and levels for a video signal that would generate the characters.

Referring to Figure 2, the CRT controller generates the refresh addresses (MA0-MA13), row addresses (RA0-RA4), and the video timing (vertical sync – VS, horizontal sync – HS, and display enable – DE). Other functions include an internal cursor register which generates a cursor output when its contents compare to the current refresh address. A light pen strobe input signal allows capture of the refresh address in an internal light pen register.

All timing in the CRTC is derived from the CLK input. In alphanumeric terminals, this signal is the character rate. The video rate or "dot" clock is externally divided by high-speed logic (TTL) to generate the CLK input. In alphanumeric terminals, this signal is the character rate. The video rate or "dot" clock is externally divided by high-speed logic (TTL) to generate the CLK signal. The high-speed logic must also generate the timing and control signals necessary for the shift register, latch, and MUX control.

The processor communicates with the CRTC through an 8-bit data bus by reading or writing into the 19 registers.

The refresh memory address is multiplexed between the processor and the CRTC. Data appears on a secondary bus separate from the processor's bus. The secondary data bus concept in no way precludes using the refresh RAM for other purposes. It looks like any other RAM to the processor. A number of approaches are possible for solving contentions for the refresh memory:

1. Processor always gets priority. (Generally, "hash" occurs as MPU and CRTC clocks are not synchronized.)
2. Processor gets priority access anytime, but can be synchronized by an interrupt to perform accesses only during horizontal and vertical retrace times.
3. Synchronize the processor with memory wait cycles (states).
4. Synchronize the processor to the character rate as shown in Figure 10. The M6800 processor family works very well in this configuration as constant cycle lengths are present. This method provides no overhead for the processor as there is never a contention for a memory access. All accesses are transparent.

The present version of the CRTC is being upgraded to improve functionality. This data sheet contains the information describing both the MC6845 (present CRTC) and the MC6845★1 (upgraded CRTC). Complete compatibility between both versions is maintained by programming all register bits, which are undefined/unused, in the MC6845 with zero's.

FIGURE 7 – RASTER SCAN SYSTEM (NON-INTERLACE)

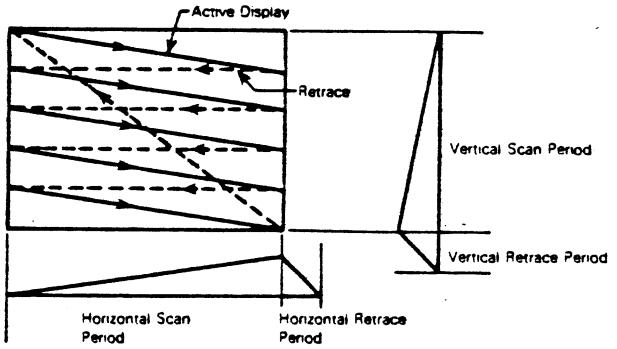
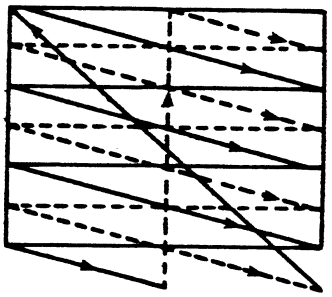


FIGURE 8 — RASTER SCAN SYSTEM (INTERLACE)



Even Number Field (First)
Odd Number Field (Second)

FIGURE 9 — CHARACTER DISPLAY ON THE SCREEN AND VIDEO SIGNAL

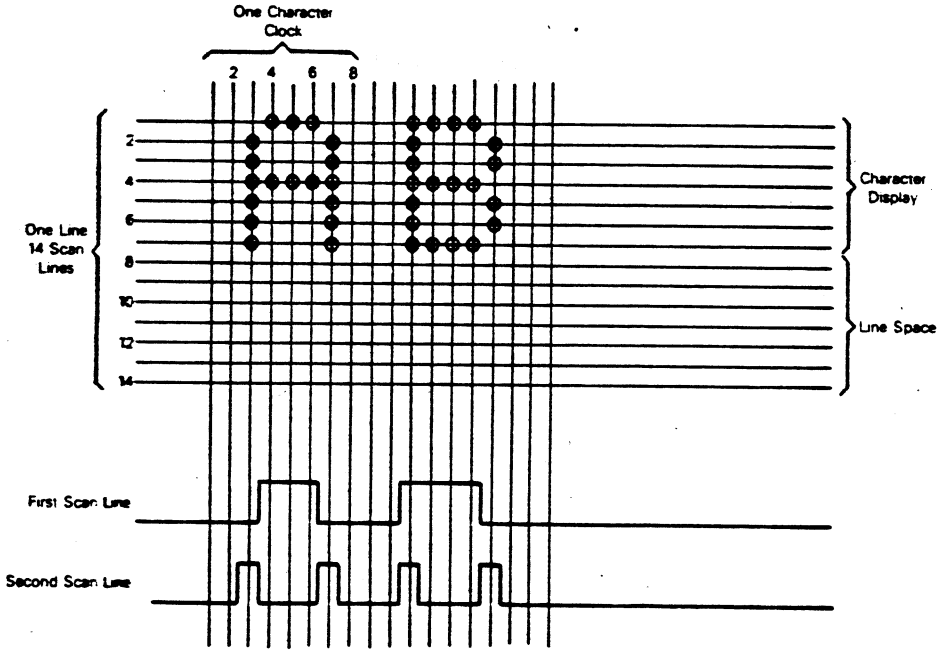
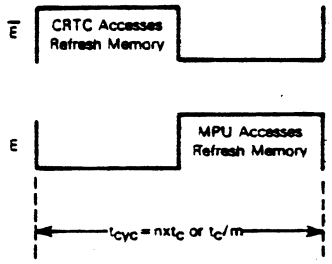


FIGURE 10 – TRANSPARENT REFRESH MEMORY CONFIGURATION TIMING USING M6800 FAMILY MPU



Where: m, n are integers; t_c is character period

TABLE 1 – CRTC OPERATING MODE

RESET	LPSTB	Operating Mode
0	0	Reset
0	1	Test Mode
1	0	Normal Mode
1	1	Normal Mode

The test mode configures the memory addresses as two independent 7-bit counters to minimize test time.

PIN DESCRIPTION

PROCESSOR INTERFACE

The CRTC interfaces to a processor bus on the bidirectional data bus (D0-D7) using $\overline{CS}$, RS, E, and R/$\overline{W}$ for control signals.

Data Bus (D0-D7) – The bidirectional data lines (D0-D7) allow data transfers between the internal CRTC register file and the processor. Data bus output drivers are high-impedance state until the processor performs a CRTC read operation.

Enable (E) – The Enable signal is a high-impedance TTL/MOS compatible input which enables the data bus input/output buffers and clocks data to and from the CRTC. This signal is usually derived from the processor clock. The high-to-low transition is the active edge.

Chip Select ($\overline{CS}$) – The CS line is a high-impedance TTL/MOS compatible input which selects the CRTC, when low, to read or write to the internal register file. This signal should only be active when there is a valid stable address being decoded from the processor.

Register Select (RS) – The RS line is a high-impedance TTL/MOS compatible input which selects either the address register (RS = "0") or one of the data register (RS = "1") of the internal register file.

Read/Write (R/$\overline{W}$) – The R/$\overline{W}$ line is a high-impedance TTL/MOS compatible input which determines whether the internal register file gets written or read. A write is defined as a low level.

CRT CONTROL

The CRTC provides horizontal sync (HS), vertical sync (VS), and display enable (DE) signals.

NOTE

Care should be exercised when interfacing to CRT monitors, as many monitors claiming to be "TTL compatible" have transistor input circuits which require the CRTC or TTL devices buffering signals from the CRTC/video circuits to exceed the maximum-rated drive currents.

Vertical Sync (VS) and Horizontal Sync (HS) – These TTL-compatible outputs are active high signals which drive the monitor directly or are fed to the video processing circuitry to generate a composite video signal. The VS signal determines the vertical position of the displayed text while the HS signal determines the horizontal position of the displayed text.

Display Enable (DE) – This TTL-compatible output is an active high signal which indicates the CRTC is providing addressing in the active display area.

REFRESH MEMORY/CHARACTER GENERATOR ADDRESSING

The CRTC provides memory addresses (MA0-MA13) to scan the refresh RAM. Row addresses (RA0-RA4) are also provided for use with character generator ROMs. In a graphics system, both the memory addresses and the row addresses would be used to scan the refresh RAM. Both the memory addresses and the row addresses continue to run during vertical retrace thus allowing the CRTC to provide the refresh addresses required to refresh dynamic RAMs.

Refresh Memory Addresses (MA0-MA13) – These 14 outputs are used to refresh the CRT screen with pages of data located within a 16K block of refresh memory. These outputs are capable of driving one standard TTL load and 30 pF.

Row Addresses (RA0-RA4) – These five outputs from the internal row address counter are used to address the character generator ROM. These outputs are capable of driving one standard TTL load and 30 pF.

OTHER PINS

Cursor – This TTL-compatible output indicates a valid cursor address to external video processing logic. It is an active high signal.

Clock (CLK) – The CLK is a TTL/MOS-compatible input used to synchronize all CRT functions except for the processor interface. An external dot counter is used to derive this signal which is usually the character rate in an alphanumeric CRT. The active transition is high-to-low.

Light Pen Strobe (LPSTB) – A low-to-high transition on this high-impedance TTL/MOS-compatible input latches the current Refresh Address in the light pen register. The latching of the refresh address is internally synchronized to the character clock (CLK).

V_{CC}, V_{SS} – These inputs supply +5 Vdc ±5% to the CRTC.

$\overline{RESET}$ – The $\overline{RESET}$ input is used to reset the CRTC. A low level on the $\overline{RESET}$ input forces the CRTC into the following state:

(a) All counters in the CRTC are cleared and the device stops the display operation.

(b) All the outputs are driven low.

(c) The control registers of the CRTC are not affected and remain unchanged.

Functionality of $\overline{RESET}$ differs from that of other M6800 parts in the following functions:

(a) The $\overline{RESET}$ input and the LPSTB input are encoded as shown in Table 1.

(b) After $\overline{RESET}$ has gone low and (LPSTB = "0"), MA0-MA13 and RA0-RA4 will be driven low on the falling edge of CLK. $\overline{RESET}$ must remain low for at least one cycle of the character clock (CLK).

(c) The CRTC resumes the display operation immediately after the release of $\overline{RESET}$. DE is not active until after the first VS pulse occurs.

CRTC DESCRIPTION
(Figure 11 CRTC Block Diagram)

The CRTC consists of programmable horizontal and vertical timing generators, programmable linear address register, programmable cursor logic, light pen capture register, and control circuitry for interface to a processor bus.

All CRTC timing is derived from CLK, usually the output of an external dot rate counter. Coincidence (CO) circuits continuously compare counter contents to the contents of the programmable register file, R0-R17. For horizontal timing generation, comparisons result in: 1) horizontal sync pulse (HS) of a frequency, position, and width determined by the registers, 2) horizontal display signal of a frequency, position, and duration determined by the registers.

The horizontal counter produces H clock which drives the scan line counter and vertical control. The contents of the Raster Counter are continuously compared to the maximum scan line address register. A coincidence resets the raster counter and clocks the vertical counter.

Comparisons of vertical counter contents and vertical registers result in: 1) vertical sync pulse (VS) of a frequency, width and position determined by the registers; 2) vertical display of a frequency and position determined by the registers.

The vertical control logic has other functions.

1. Generate row selects, RA0-RA4, from the raster count for the corresponding interlace or non-interlace modes.
2. Extend the number of scan lines in the vertical total by the amount programmed in the vertical total adjust register.

The linear address generator is driven by CLK and locates the relative positions of characters in memory with their positions on the screen. Fourteen lines, MA0-MA13, are available for addressing up to four pages of 4K characters, 8 pages of 2K characters, etc. Using the start address register, hardware scrolling through 16K characters is possible. The linear address generator repeates the same sequence of addresses for each scan line of a character row.

The cursor logic determines the cursor location, size, and blink rate on the screen. All are programmable.

The light pen strobe going high causes the current contents of the address counter to be latched in the light pen register. The contents of the light pen register are subsequently read by the processor.

Internal CRTC registers are programmed by the processor through the data bus, D0-D7, and the control signals – R/$\overline{W}$, $\overline{CS}$, RS, and E.

REGISTER FILE DESCRIPTIONS

The nineteen registers of the CRTC may be accessed through the data bus. Only two memory locations are required as one location is used as a pointer to address one of the remaining eighteen registers. These eighteen registers control horizontal timing, vertical timing, interlace operation, row address operation, and define the cursor, cursor address, start address, and light pen register. The register addresses and sizes are shown in Table 2.

ADDRESS REGISTER

The address register is a 5-bit write-only register used as an "indirect" or "pointer" register. It contains the address of one of the other eighteen registers. When both RS and $\overline{CS}$ are low, the address register is selected. When $\overline{CS}$ is low and RS is high, the register pointed to by the address register is selected.

TIMING REGISTERS R0-R9

Figure 12 shows the visible display area of a typical CRT monitor giving the point of reference for horizontal registers as the left most displayed character position. Horizontal registers are programmed in character clock time units with respect to the reference as shown in Figure 13. The point of reference for the vertical registers is the top character position displayed. Vertical registers are programmed in scan line times with respect to the reference as shown in Figure 14.

Horizontal Total Register (R0) – This 8-bit write-only register determines the horizontal sync (HS) frequency by defining the HS period in character times. It is the total of the displayed characters plus the non-displayed character times (retrace) minus one.

FIGURE 11 — CRTC BLOCK DIAGRAM

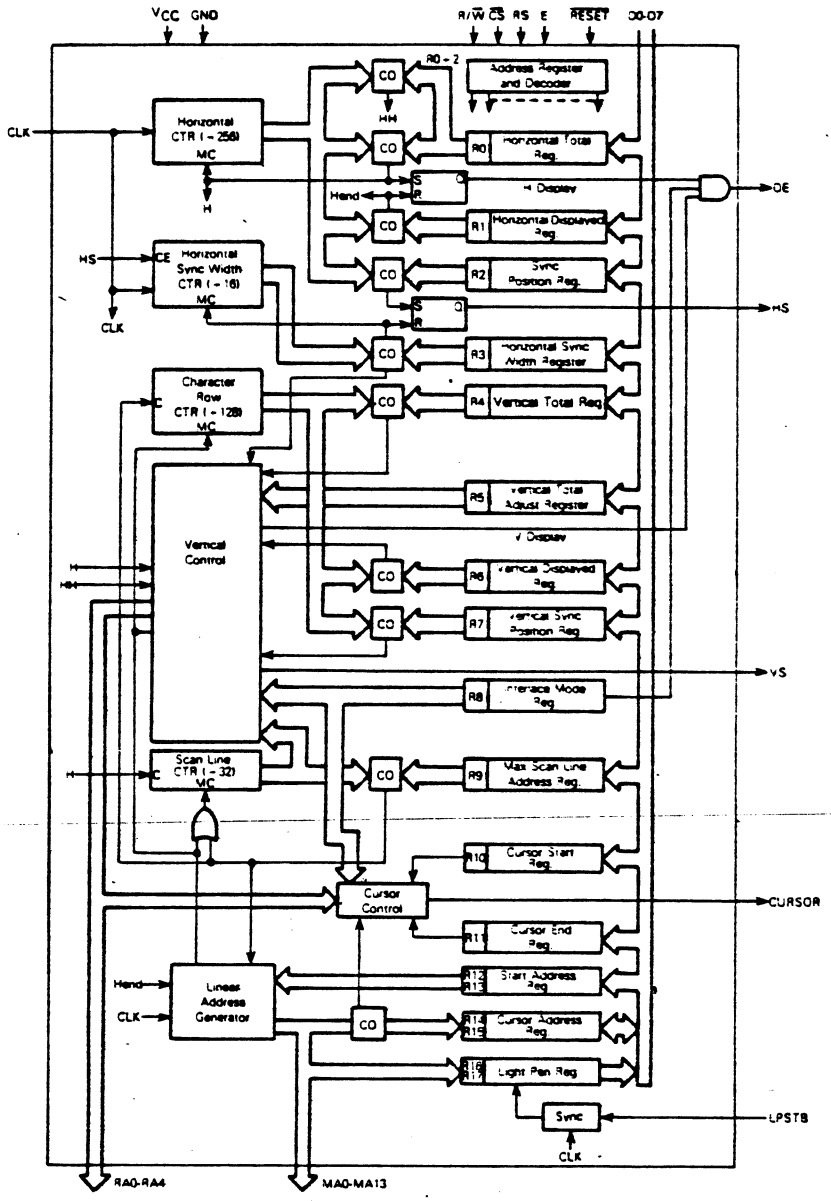
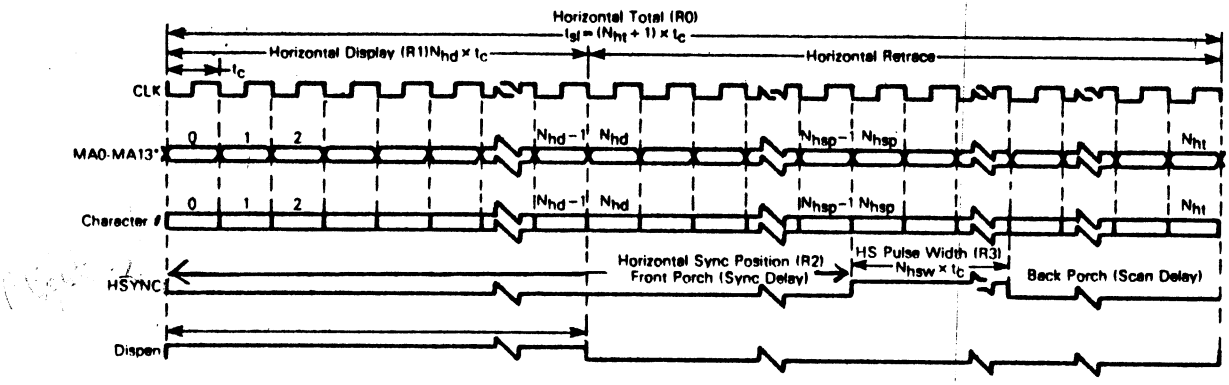


TABLE 3 — CURSOR AND DE SKEW CONTROL

Value	Skew
00	No Character Skew
01	One Character Skew
10	Two Character Skew
11	Not Available

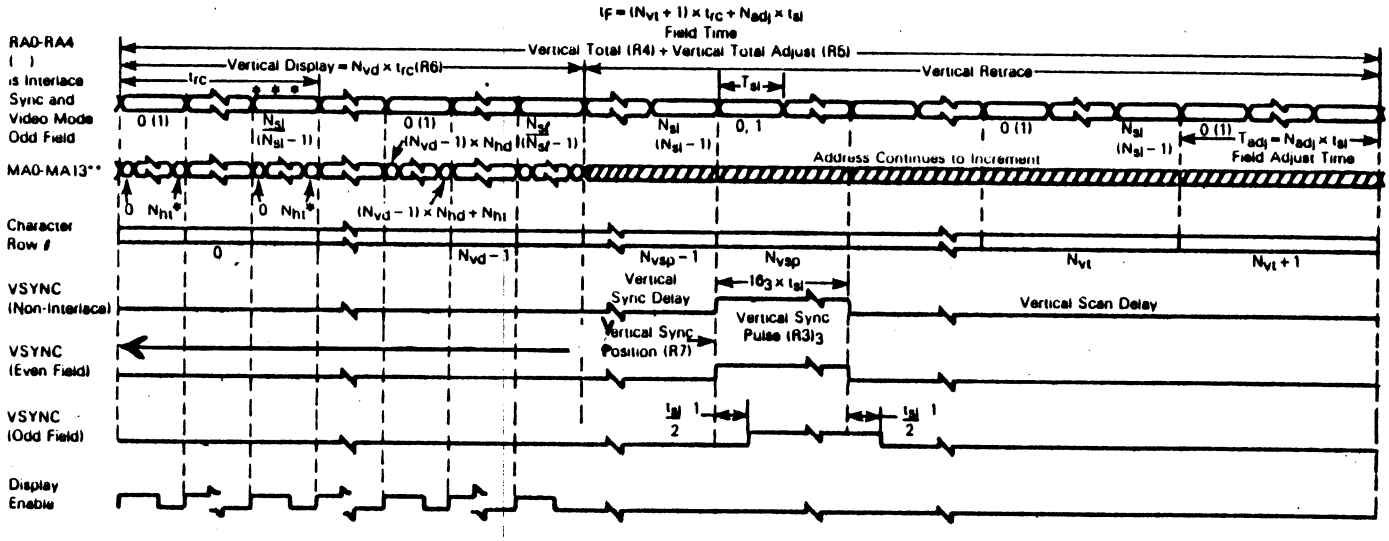
FIGURE 13 — CRTC HORIZONTAL TIMING



*Timing is shown for first displayed scan row only. See Chart in Figure 16 for other rows. The initial MA is determined by the contents of Start Address Register, R12/R13. Timing is shown for R12/R13 = 0.

Note 1: Timing values are described in Table 8

FIGURE 14 — CRTC VERTICAL TIMING



*N_{ht} must be an odd number for both interlace modes

**Initial MA is determined by R12/R13 (Start Address Register), which is zero in this timing example.

***N_{sl} must be an odd number for Interlace Sync and Video Mode.

NOTES

1. Refer to Figure 8 — The Odd Field is offset ½ horizontal scan time.
2. Timing values are described in Table 8.
3. Vertical Sync Pulse width may be programmed from 1 to 16 scan line times for the MC6845☆1.

Horizontal Displayed Register (R1) – This 8-bit write-only register determines the number of displayed characters per line. Any 8-bit number may be programmed as long as the contents of R0 are greater than the contents of R1.

Horizontal Sync Position Register (R2) – This 8-bit write-only register controls the HS position. The horizontal sync position defines the horizontal sync delay (Front Porch) and the horizontal scan delay (Back Porch). When the programmed value of this register is increased, the display on the CRT screen is shifted to the left. When the programmed value is decreased the display is shifted to the right. Any 8-bit number may be programmed as long as the sum of the contents of R2, and R3 are less than the contents of R0.

Sync Width Register (R3) – This 8-bit write-only register determines the width of the vertical sync (VS) pulse and the horizontal sync (HS) pulse for the MC6845★1 CRTC. The vertical sync pulse width is fixed at 16 scan-line times for the MC6845 and the upper four bits of this register are treated as "don't cares."

The MC6845★1 allows control of the VS pulse width for 1-to-16 scan-line times. Programming the upper four bits for 1-to-15 will select pulse widths from 1-to-15 scan-line times. Programming the upper four bits as zeros will select a VS pulse width of 16 scan-line times, allowing compatibility with the MC6845.

For both the MC6845 and the MC6845★1, the HS pulse width may be programmed from 1-to-15 character clock periods thus allowing compatibility with the HS pulse width specifications of many different monitors. If zero is written into this register then no HS is provided.

Horizontal Timing Summary (Figure 13) – The difference between R0 and R1 is the horizontal blanking interval. This interval in the horizontal scan period allows the beam to return (retrace) to the left side of the screen. The retrace time is determined by the monitor's horizontal scan components. Retrace time is less than the horizontal blanking interval. A good rule of thumb is to make the horizontal blanking about 20% of the total horizontal scanning period for a CRT. In inexpensive TV receivers, the beam overscans the display screen so that aging of parts does not result in underscanning. Because of this, the retrace time should be about ½ the horizontal scanning period. The horizontal sync delay, HS pulse width, and horizontal scan delay are typically programmed with a 1:2:2 ratio.

Vertical Total Register (R4) and Vertical Total Adjust Register (R5) – The frequency of VS is determined by both R4 and R5. The calculated number of character line times is usually an integer plus a fraction to get exactly a 50 or 60 Hz vertical refresh rate. The integer number of character line times minus one is programmed in the 7-bit write-only vertical total register (R4). The fraction of character line times is programmed in the 5-bit write-only vertical total adjust register (R5) as a number of scan-line times.

Vertical Displayed Register (R6) – This 7-bit write-only register specifies the number of displayed character rows on the CRT screen, and is programmed in character row times. Any number smaller than the contents of R4 may be programmed into R6.

Vertical Sync Position (R7) – This 7-bit write-only register controls the position of vertical sync with respect to the reference. It is programmed in character row times. The value programmed in the register is one less than the number of computed character-line times. When the programmed value of this register is increased, the display position of the CRT screen is shifted up. When the programmed value is decreased the display position is shifted down. Any number equal to or less than the vertical total (R4) may be used.

Interlace Mode and Skew Register (R8) – The MC6845 only allows control of the interlace modes as programmed by the low order two bits of this write-only register. The MC6845-1 controls the interlace modes and allows a programmable delay of zero-to-two character clock times for the DE (display enable) and cursor outputs. Table 3 describes operation of the cursor and DE skew bits. Cursor skew is controlled by bits 6 and 7 of R8 while DE skew is controlled by bits 4 and 5. Table 4 shows the interlace modes available to the user. These modes are selected using the two low order bits of this 6-bit write-only register.

In the normal sync mode (non-interlace) only one field is available as shown in Figures 7 and 15a. Each scan line is refreshed at the VS frequency (e.g., 50 or 60 Hz).

Two interlace modes are available as shown in Figures 8, 15b, and 15c. The frame time is divided between even and odd alternating fields. The horizontal and vertical timing relationship (VS delayed by ½ scan line time) results in the displacement of scan lines in the odd field with respect to the even field.

In the interlace sync mode the same information is painted in both fields as shown in Figure 15b. This is a useful mode for filling in a character to enhance readability.

In the interlace sync and video mode, shown in Figure 15c, alternating lines of the character are displayed in the even field and the odd field. This effectively doubles the given bandwidth of the CRT monitor.

Care must be taken when using either interlace mode to avoid an apparent flicker effect. This flicker effect is due to the doubling of the refresh time for all scan lines since each field is displayed alternately and may be minimized with proper monitor design (e.g., longer persistence phosphors).

In addition, there are restrictions on the programming of the CRTC registers for interlace operation:

1. For the MC6845:
   a. The horizontal total register value, R0, must be odd (i.e., an even number of character times).
   b. For interlace sync and video mode only, the maximum scan-line address, R9, must be odd (i.e., an even number of scan lines).
   c. For interlace sync and video mode only, the vertical displayed register (R6) must be even. The programmed number Nvd, must be ½ the actual number required. The even numbered scan lines are displayed in the even field and the odd numbered scan lines are displayed in the odd field.
   d. For interlace sync and video mode only, the cursor start register (R10) and cursor end register (R11) must both be even or both odd depending on which field the cursor is to be displayed in.
2. For the MC6845★1:
   a. The horizontal total register value, R0, must be odd (i.e., an even number of character times).
   b. For the interlace sync and video mode only, the vertical displayed register (R6) must be even. The programmed number, Nvd, must be ½ the actual number required.

TABLE 4 — INTERLACE MODE REGISTER

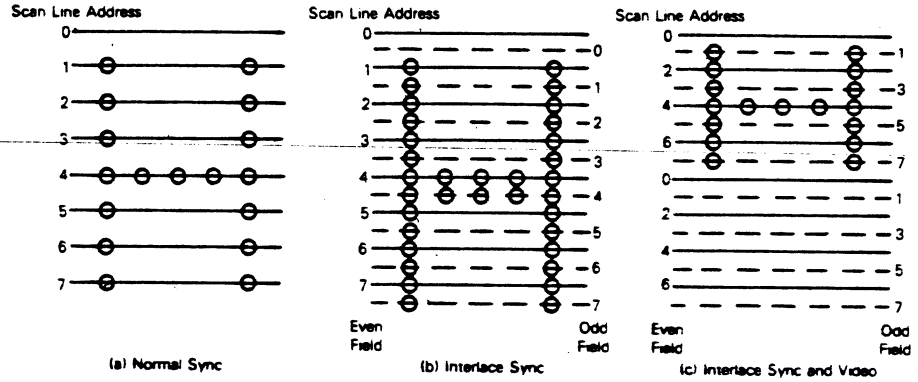
Bit 1	Bit 0	Mode
0	0	Normal Sync Mode (Non-Interlace)
1	0	
0	1	Interlace Sync Mode
1	1	Interlace Sync and Video Mode

TABLE 5 — CURSOR START REGISTER

Bit 6	Bit 5	Cursor Display Mode
0	0	Non-Blink
0	1	Cursor Non-Display
1	0	Blink, 1/16 Field Rate
1	1	Blink, 1/32 Field Rate

Example of Cursor Display Mode

FIGURE 15 — INTERLACE CONTROL



(a) Normal Sync (b) Interlace Sync (c) Interlace Sync and Video

Maximum Scan Line Address Register (R9) — This 5-bit write-only register determines the number of scan lines per character row including the spacing; thus, controlling operation of the row address counter. The programmed value is a maximum address and is one less than the number of scan lines.

CURSOR CONTROL

Cursor Start Register (R10) and Cursor End Register (R11) — These registers allow a cursor of up to 32 scan lines in height to be placed on any scan line of the character block as shown in Figure 16. R10 is a 7-bit write-only register used to define the start scan line and the cursor blink rate. Bits 5 and 6 of the cursor start address register control the cursor operation as shown in Table 5. Non-display, display, and two blink modes (16 times or 32 times the field period) are available. R11 is a 5-bit write-only register which defines the last scan line of the cursor.

When an external blink feature on characters is required, it may be necessary to perform cursor blink externally so that both blink rates are synchronized. Note that an invert/non-invert cursor is easily implemented by programming the CRTC for a blinking cursor and externally inverting the video signal with an exclusive-OR gate.

Cursor Register (R14-H, R15-L) — This 14-bit read/write register pair is programmed to position the cursor anywhere in the refresh RAM area; thus, allowing hardware paging and scrolling through memory without loss of the original cursor position. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register.

OTHER REGISTERS

Start Address Register (R12-H, R13-L) — This 14-bit write-only register pair controls the first address output by the CRTC after vertical blanking. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register. The start address register determines which portion of the refresh RAM is displayed on the CRT screen. Hardware scrolling by character, line, or page may be accomplished by modifying the contents of this register.

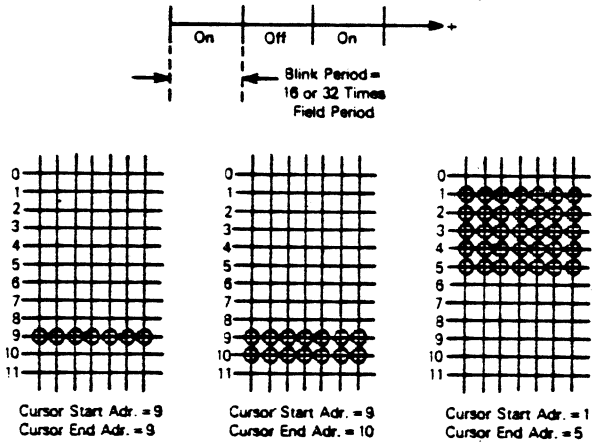
Light Pen Register (R16-H, R17-L) — This 14-bit read-only register pair captures the refresh address output by the CRTC on the positive edge of a pulse input to the LPSTB pin. It consists of an 8-bit low order (MA0-MA7) register and a 6-bit high order (MA8-MA13) register. Since the light pen pulse is asynchronous with respect to refresh address timing an internal synchronizer is designed into the CRTC. Due to delays (Figure 3) in this circuit, the value of R16 and R17 will need to be corrected in software. Figure 17 shows an interrupt driven approach although a polling routine could be used.

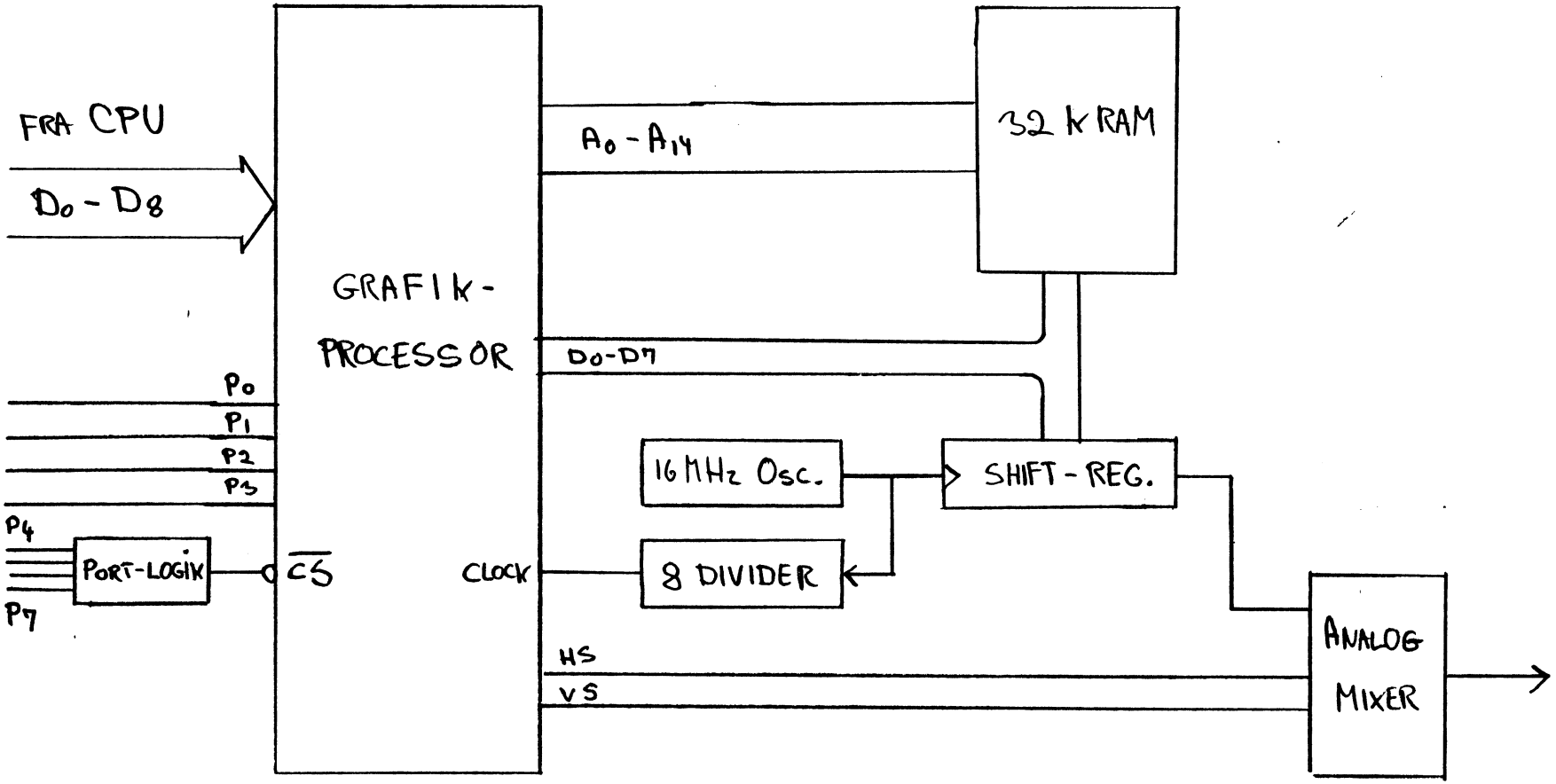
CRTC INITIALIZATION

Registers R0-R15 must be initialized after the system is powered up. The processor will normally load the CRTC register file from a firmware table. The worksheet of Table 6 is extremely useful in computing proper register values for the CRTC. Table 7 shows the worksheet filled out for an 80 × 24 configuration using a 7 × 9 character generator and Figure 18 shows an M6800 program which could be used to program the CRT controller. The programmed values allow use of either an MC6845 or MC6845☆1 CRTC.

The CRTC registers will have an initial value at power up. When using a direct drive monitor (sans horizontal oscillator) these initial values may result in out-of-tolerance operation. CRTC programming should be done immediately after power up especially in this type of system.

FIGURE 16 — CURSOR CONTROL





PRINCIPDIAGRAM / GRAFIKMODUL

CMD COMMAND REGISTER (Address : 0_{16})

The CMD register is an 8-bit write-only register. Each write operation in this register causes a command to be executed, upon completion of the time necessary for synchronizing the microprocessor access and the GDP's CK clock.

Several types of command are available :

- vector plotting
- character plotting
- screen erase
- light pen circuitry setting
- access to the display memory through an external circuitry.
- indirect modification of the other registers (commands that make it possible for the X, Y, DELTAX, DELTAY, CTRL1, CTRL2 and CSIZE registers to be amended or scratched).

STATUS REGISTER (Address 0_{16})

The STATUS register is an 8-bit read-only register. It is used to monitor the status of the executing statements entered into the circuit, and more specifically to avoid the need for modifying a register that is already used for the command currently executing.

Bit 0 : When low, this bit indicates that a light pen sequence is currently executing.
When high, it indicates that no light pen sequence is currently executing.

Bit 1 : This bit is high during vertical blanking. It is the VB signal recopy.

Bit 2 : When low, this bit indicates that a command is currently executing.
When high, this bit indicates that the circuit is ready for a new command.

Bit 3 : When low, this bit indicates that the X and Y registers point within the display window.
When high, this bit indicates that the X and Y registers are pointing outside the memory display.
This bit is the logic OR of the unused MSBs of the X and Y registers.

Bit 4 : When high, this bit indicates that an interrupt has been initiated by the completion of a light pen running sequence. Such an interrupt is enabled by bit 4 in CTRL1 register.

Bit 5 : When high, this bit indicates that an interrupt has been initiated by vertical blanking. Such an interrupt is enabled by bit 5 in CTRL1 register.

Bit 6 : When high, this bit indicates that an interrupt has been initiated by the completion of execution of a command. Such an interrupt is enabled by bit 6 in CTRL1 register.

Bit 7 : When high, this bit indicates that an interrupt has been initiated. It is the logic OR of bits 4, 5 and 6 in STATUS register. The $\overline{IRQ}$ output state is always the opposite of the status of this bit.

Note : Bits 4, 5, 6 and 7 are reset low by a read of the STATUS register.

XLP AND YLP REGISTERS (Addresses C_{16} and D_{16})

The XLP and YLP registers are read-only registers, with 7 and 8 bits respectively. Upon completion of a light pen running sequence, they contain the display address sampled by the first edge appearing rising on the LPCK input. The use of such registers is discussed in section : **Use of light pen circuitry.**

NOTES :

1. All internal registers may be read or written at any time by the microprocessor. However, the precautions outlined below should be observed :

   – Do not write into the CMD register if execution of the previous command is not completed (bit 2 of STATUS register).

   – Do not alter any register if it is used as an input parameter for the internal hardwired systems (e. g. : modifying the DELTAX register while a vector plotting sequence is in progress).

   – Do not read a register that is being asynchronously modified by the internal hardwired systems (e. g. : reading the X register while a vector plotting sequence is in progress may be erroneous if CK and $\overline{E}$ are asynchronous).

2. On powering up, the writing devices may have any status. Before entering a command for the first time, it is necessary to wait until all functions currently underway are completed, which information can be derived from the STATUS register.

REGISTER DESCRIPTION

X AND Y REGISTERS (Addresses : 8_{16}, 9_{16}, A_{16}, 8_{16})

The X and Y registers are 12-bit read-write registers. They indicate the position of the next dot to be written into the display memory. They have no connection at all with the video signal generating scan, but they point the write address, in the same way as the pen address on a plotter.

These 2 registers are incremented or decremented, prior to each write operation into the display memory, by the internal vector and character generators, or they may be directly positioned by the microprocessor.

This 2 x 12 bit write address covers a 4096 x 4096 point addressing space. Only the LSBs are used here, since the maximum definition of the picture actually stored is 512 x 512 pixels (picture elements).

The MSBs are either ignored or used to inhibit writing where the actual screen is regarded as being a window within a 4096 x 4096 space.

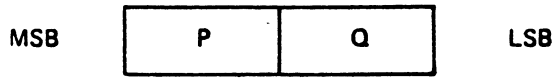
The above features along with the relative mode description of all picture component elements make it possible to automatically solve the great majority of edge cut-off problems.

DELTAX AND DELTAY REGISTERS (Addresses : 5_{16}, 7_{16}).

The DELTAX and DELTAY registers are 8-bit read-write registers. They indicate to the vector generator the projections of the next vector to be plotted, on the X and Y axes respectively. Such values are unsigned integers. The plotting of a vector is initiated by a write operation in the command register (CMD).

CSIZE REGISTER (Address : 3_{16})

The CSIZE register is an 8-bit read-write register. It indicates the scaling factors of X and Y registers for the symbols and characters. 98 characters are generated from a 5 x 8 pixel matrix defined by an internal ROM. In the standard version, it contains the alphanumeric characters in the ASCII code which may be printed, together with a number of special symbols.



Each symbol can be increased by a factor P(X) or Q(Y). These factors are independent integers which may each vary from 1 to 16 and which are defined by the CSIZE register. The symbol generation sequence is started after writing the ASCII code of the symbol to be represented in the CMD register.

CTRL1 REGISTER (Address : 1_{16}).

The CTRL1 register is a 7-bit read-write register, through which the general circuit operation may be fed with the required parameters.

Bit 0 : When low, this bit inhibits writing in display memory (equivalent to pen or eraser up).
When high, this bit enables writing in display memory (pen or eraser down).
This bit controls the $\overline{DW}$ output.

Bit 1 : When low, this bit selects the eraser.
When high, this bit selects the pen.
This bit controls the DIN output.

Bit 2 : When low, this bit selects normal writing mode (writing apart from the display and refresh periods, which are a requirement for the dynamic storages) in display memory.
When high, this bit selects the high speed writing mode : the display periods are deleted. Only the dynamic storage refresh periods are retained.

Bit 3 : When low, this bit indicates that the 4096 x 4096 space is being used (the 12 X and Y bits are significant)
When high, this bit selects the cyclic screen operating mode.

Bit 4 : When low, this bit inhibits the interrupt triggered by the light pen sequence completion.
When high, this bit enables the interrupt.

Bit 5 : When low, this bit inhibits the interrupt release by vertical blanking.
When high, this bit enables the interrupt.

Bit 6 : When low, this bit inhibits the interrupt indicating that the system is ready for a new command.
When high, this bit enables the interrupt.

Bit 7 : Not used. Always low in read mode.

CTRL2 REGISTER (Address : 2_{16})

The CTRL2 register is a 4-bit read/write register, through which the plotting of vectors and characters may be denoted by parameters.

Bit 0, 1 : These 2 bits define 4 types of lines (continuous, dotted, dashed, dash-dotted).

Bit 2 : When low, this bit defines straight writing.
When high, it defines tilted characters.

Bit 3 : When low, this bit defines writing along an horizontal line.
When high, this bit defines writing along a vertical line.

Bit 4, 5, 6, 7 : Not used. Always low in read mode.

HARDWIRED WRITE PROCESSOR OPERATION IN DISPLAY MEMORY

The hardwired write processors are sequenced by the master clock CK. They receive their parameters from the microprocessor bus. They control the X, Y write address, and the DIN, $\overline{DW}$, $\overline{MFREE}$ and $\overline{IRQ}$ outputs.

These harwired processors operate in continuous mode. In the event of conflicting access to the display memory, the display and refresh processors have priority.

Since command decoding is synchronous with the CK master clock, any write operation into the (CMD) command register triggers a synchronizing mechanism which engages the circuit for a maximum of 2 CK cycles when the $\overline{E}$ input returns high. The circuit remains engaged throughout command execution.

No further command should be entered as long as bit 2 in STATUS register is low.

VECTOR PLOTTING

The internal vector generator makes it possible to modify, within the display memory, all the dots which form the approximation of a straight line segment. All vectors plotted are described by the origin dot and the projections on the axes.

The starting point co-ordinates are defined by the X, Y register value, prior to the plotting operation.
Projections onto the axes are defined as absolute values by the DELTAX and DELTAY registers, with the sign in the command byte that initiates the vector plotting process.

The vector approximation achieved here is that established by J. F. BRESENHAM ("Algorithm for computer control of a digital plotter"). This algorithm is executed by a hardwired processor which allows for a further vector component dot to be written in each CK clock cycle.

During plotting, the display memory is addressed by the X, Y registers, which are incremented or decremented.

On completion of vector plotting, they point to the end of this vector.

All vectors may be plotted using any of the following line patterns : continuous, dotted, dashed, dash-dotted, according to the 2 LSBs in register CTRL2.

Irrespective of such patterns, the plotting speed remains unchanged. The "pen down-pen up" statement required for plotting non-continuous lines is controlled by the $\overline{DW}$ output.

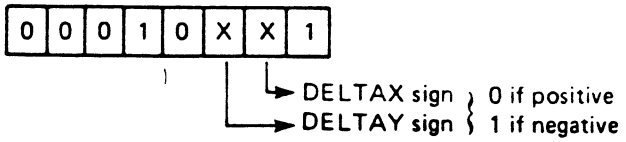
For a specified non-continuous line plotted vector, defined by DELTAX, DELTAY, CTRL2, CMD, the $\overline{DW}$ sequencing during the plotting process is always the same, irrespective of vector origin and of the nature of previous plots. This feature guarantees that a specified vector can be deleted by plotting it again after moving X and Y to the starting point, and complementing bit 1 in register CTRL1.

Since the vector plotting initiation command defines the sign of the projections onto the axes, all vectors may be plotted using 4 different commands.

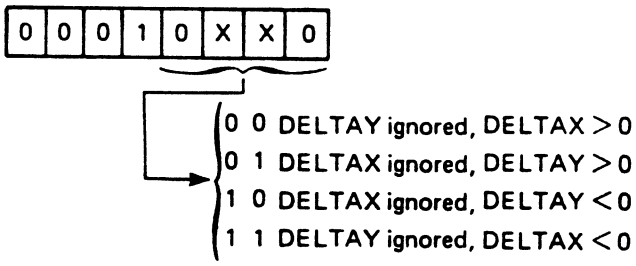
For increased programming flexibility, the system incorporates 16 different commands, supplemented by a set of 128 commands which make it possible to plot small size vectors by ignoring the DELTAX and DELTAY registers.

Such commands are as follows :

- Basic commands

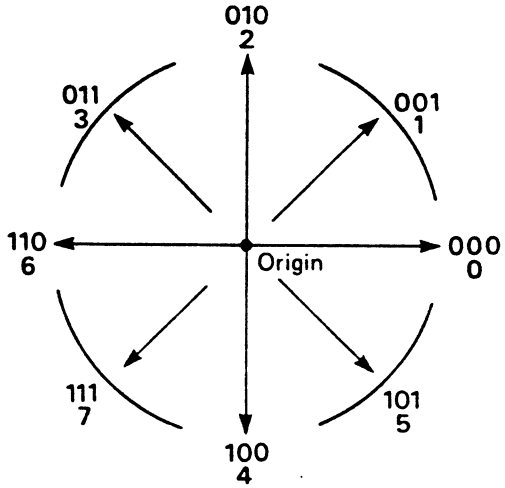


- Commands which allow ignoring the DELTAX or DELTAY registers by considering them as of zero value.

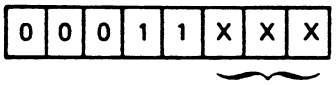


Note : Bits 1 and 2 always have the same sign meaning.

These 8 codes may be summarized by the following diagram :

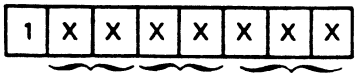


- Commands which allow ignoring the smaller of the two DELTAX and DELTAY registers, by considering it as being equal to the larger one, which is the same as plotting vectors parallel to the axes or diagonals, using a single DELTA register.



Same direction codes as above.

- Commands in which the two registers DELTAX and DELTAY may be ignored by specifying the projections through the CMD register (0 to 3 steps for each projection).



EXAMPLE : PLOTTING A DOTTED VECTOR

Origin : $\begin{cases} X = 47_{10} \\ Y = 75_{10} \end{cases}$

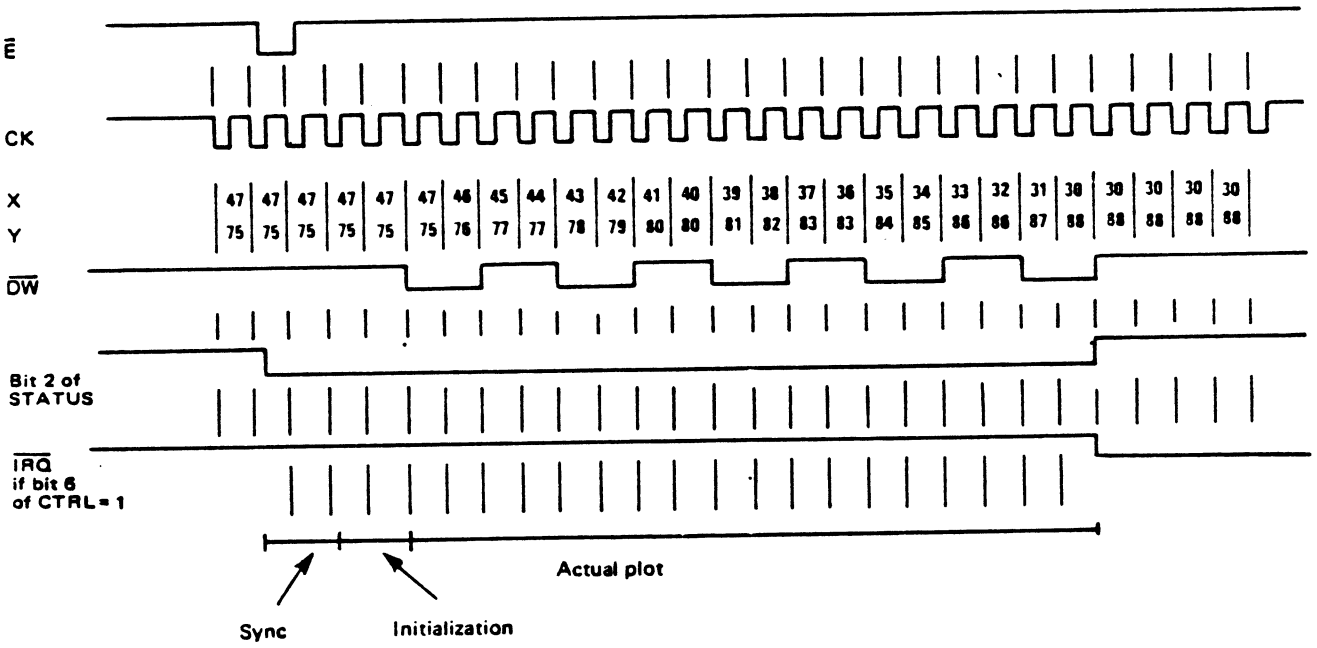
Projections: $\begin{cases} DELTAX = 17_{10} \\ DELTAY = 13_{10} \end{cases}$

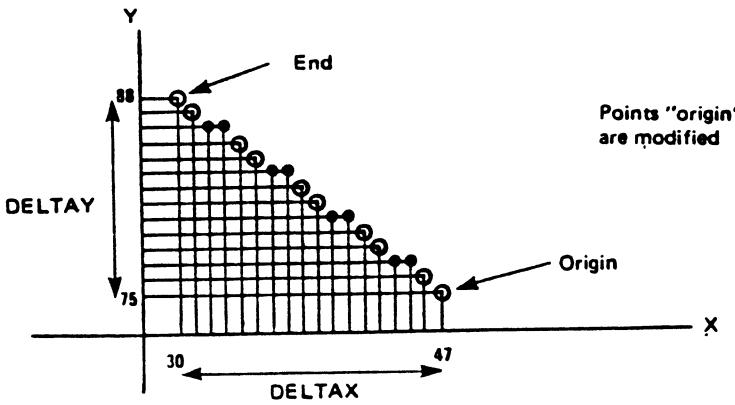
$CMD = 13_{16}$ — Corresponding to
- Basic command,
- DELTAX < 0
- DELTAY > 0

$CTRL1 = 03_{16}$ — Pen down

$CTRL2 = 1_{16}$ — Dotted vector :
2 dots on,
2 dots off.

Plotting cycle sequence : (It is assumed that the vector generator is not interrupted by the display or refresh cycle).





Points "origin" and "end" are modified

Display shows : ● unmodified dot
o dot on

Note :

Plotting a vector with DELTAX = DELTAY = 0 writes the point X, Y in memory. It occupies the vector generator for synchronization, initialization and one write cycle.

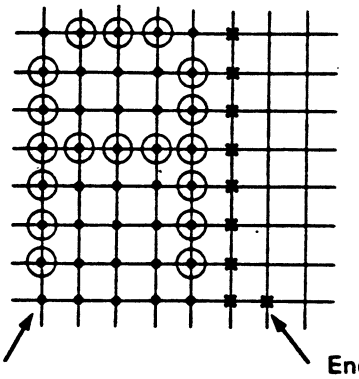
CHARACTER AND SYMBOL GENERATOR

The character generator operates in the same way as the vector generator, i.e. through incrementing or decrementing the X, Y registers, in conjunction with a $\overline{DW}$ output control.

It receives parameters from the CSIZE, CTRL2 and CMD registers. The characters plotted are selected, according to the CMD value, out of 98 matrices (97 8-dot high x 5-dot wide rectangular matrices, and one 4 dot x 4 dot matrix) defined in an internal ROM. Two scaling factors may be applied to the characters plotted using X and Y defined by the CSIZE register. The characters may be tilted, according to the content of register CTRL2.

Basic matrix

Upon completion of a character writing process, the X and Y registers are positioned for writing a further character next to the previous one, with a 1 dot spacing, i.e. Y is restored to its original value and X is incremented by 6.



• Unchanged
⊙ Altered dots
} if CMD = 41_{16} (in the ROM standard version)

× Computed dots, not defined into the ROM (not modifiable).

Scaling factors

Each individual dot in the 5 x 8 basic matrix may be replaced by a P x Q size block.

P : X co-ordinate scaling factor
Q : Y co-ordinate scaling factor

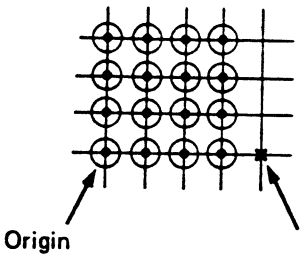
The character size becomes 5P x 8Q. Upon completion of the writing process, X is incremented by 6P. The CK clock cycle count required is 6P x 8Q.

P and Q may each take values from 1 through 16. They are defined by the CSIZE register. Each value is encoded on 4 bits, value 16 being encoded as 0_{16}.

In register CSIZE, P is encoded on the 4 MSBs and Q on the 4 LSBs.

Among the 97 rectangular matrices available in the standard ROM, 96 correspond to CMD values ranging from 20_{16} to $7F_{16}$, and the 97th matrix to $0A_{16}$. In the standard version, these values correspond to the 96 printable characters in the ASCII set. The 97th character is a 5P x 8Q block which may be used for deleting the other characters.

The 98th code ($0B_{16}$) is used to plot a 4P x 4Q graphic block. It locates X, Y, without spacing for the next symbol. Such a block makes it possible to pad uniform areas on the screen.



⊙ Modified dots
× Computed dot not defined in ROM (not modifiable)

Tilted characters

All characters may be modified to produce tilted characters or to mark the vertical co-ordinate with straight or tilted type symbols. Such changes may be achieved using bits 2 and 3 in register CTRL2.

Note : Scaling factors P and Q are always applied within the co-ordinates of the character before conversion.

Character deletion

A character may be deleted using either the same command code or command code $0A_{16}$. In either case, bit 1 in register CTRL1 should be inverted, the origin should be the same as prior to a character plotting operation, as should the scaling factors.

Note : Vector generator and character generator operate in similar ways :

	VECTOR	CHARACTER
Dimensions	DELTAX, DELTAY	CSIZE, tilting
$\overline{DW}$ modulation	Type of line	Character code

USE OF LIGHT PEN CIRCUITRY

A rising edge on the LPCK input is used to sample the current display address in the XLP and YLP registers, provided that this edge is present in the frame immediately following loading of the 08_{16} or 09_{16} code into the CMD register.

Here, the frame origin is counted starting with the VB falling edge. With code 08_{16}, the $\overline{WHITE}$ output recopies the BLK signal from the frame origin up to the rising edge on the LPCK input, or when VB starts rising again, if the LPCK input remains low for the entire frame. With code 09_{16}, the $\overline{WHITE}$ output is not activated.

The YLP address is 8-bit coded since there are 256 display lines in each frame. The XLP address is 6-bit coded since there are 64 display cycles in each line.

These 6 bits are left justified in the XLP register. XLP and YLP register contents match the write address if FMAT is low (or for the EF9366), but should be multiplied by 2 if FMAT is high, so as to be able to match the write address.

The address sampled into XLP corresponds to the current memory cycle. Dots detected by the light pen were addressed in the memory during the previous cycle. Hence, 1 should be subtracted from bit 2 in XLP register where the light pen electronic circuitry does not produce any additional delay.

If the rising edge on input LPCK occurs while VB is low, then the LSB in XLP is set high. This bit acts as a status signal which is reset to the low state by reading register XLP or YLP.

The rising edge first received (LPCK or VB) sets bit 0 in STATUS register high. An interrupt is initiated if bit 4 in CTRL1 is high.

When commands 08_{16} or 09_{16} have been decoded, bit 2 of the status register goes high (circuit ready for any further command) and bit 0 goes low (light pen operating sequence underway).

SCREEN BLANKING COMMANDS

Three commands (04_{16}, 06_{16}, 07_{16}) will set the whole display memory to a status corresponding to a "black display screen" condition. Another command ($0C_{16}$) may be used to set the whole memory to a status other than black (this condition being determined by bit 1 in register CTRL1).

The 4 commands outlined above use the planned scanning of the memory addresses achieved by the display stage. The X and Y registers are not affected by commands 04_{16} and $0C_{16}$. Hence, the time required is that corresponding to one frame (EF9366 or FMAT low) or two frames (FMAT high). The time corresponding to the completion of the frame currently executing when the CMD register is loaded, should be added to the above time.

For the screen blanking process, the frame origin is counted starting with the VB falling edge.
The only signals affected here are the $\overline{DW}$ output, which remains low when VB is low, and the DIN output which is forced high where the 04_{16}, 06_{16} and 07_{16} commands are entered.

Such commands are activated without requiring action by WO input or bit 2 in register CTRL1. While these commands are executing, bit 2 in STATUS register remains low.

EXTERNAL REQUEST FOR DISPLAY MEMORY ACCESS ($\overline{MFREE}$ OUTPUT)

On writing code $0F_{16}$ into the CMD register, the $\overline{MFREE}$ output is set low by the circuitry, during the next free memory cycle.

Apart from the display and refresh periods, this cycle is the first complete cycle that occurs after input $\overline{E}$ is reset high.

During this cycle, those addresses output on DAD and MSL correspond to the X and Y register contents : $\overline{DW}$ is high, $\overline{ALL}$ is high.

Should the memory be engaged in a display or refresh operation, (which is the case when $\overline{ALL}$ is low), then this cycle is postponed to be executed after $\overline{ALL}$ is reset high. The maximum waiting time is thus 64 cycles.

The $\overline{MFREE}$ signal may be used e. g. for performing a read or write operation into a register located between the display memory and the microprocessor bus.

INTERRUPTS OPERATION

An interrupt may be initiated by three situations denoted by internal signals :

- Circuit ready for a further command
- Vertical blanking signal
- Light pen sequence completed.

These three signals appear in real time in the STATUS register (bits 0, 1, 2). Each signal is cross-referenced to a mask bit in the register CTRL1 (bits 4, 5, 6).

If the mask bit is high, the first rising edge that occurs on the interrupt initiating signal sets the related interrupt flip-flop circuit high.

The outputs from these three flip-flop circuits appear in the STATUS register (bits 4, 5, 6). If one flip-flop circuit is high, bit 7 in the STATUS register is high, and pin $\overline{IRQ}$ is forced low.

A read operation in the STATUS register resets its 4 MSBs low, after input $\overline{E}$ is reset high.

The three interrupt control flip-flops are duplicated to prevent the loss of an interrupt coming during a read cycle of the STATUS register.

The status of bits 4, 5 and 6 corresponds to the interrupt control flip-flop circuit output, before input $\overline{E}$ goes low.

An interrupt coming during a read cycle of the STATUS register does not appear in bits 4, 5 and 6 during this read sequence, but during the following one. However, it may appear in bits 0, 1, 2 or on pin $\overline{IRQ}$.

TABLE 1 – REGISTER ADDRESS

ADDRESS REGISTER					REGISTER FUNCTIONS		Number of bits
Binary				Hexa	Read R/$\overline{W}$ = 1	Write R/$\overline{W}$ = 0	
A3	A2	A1	A0				
0	0	0	0	0	STATUS	CMD	8
0	0	0	1	1	CTRL 1 (Write control and interrupt control)		7
0	0	1	0	2	CTRL 2 (Vector and symbol type control)		4
0	0	1	1	3	CSIZE (Character size)		8
0	1	0	0	4	Reserved		–
0	1	0	1	5	DELTAX		8
0	1	1	0	6	Reserved		–
0	1	1	1	7	DELTAY		8
1	0	0	0	8	X MSBs		4
1	0	0	1	9	X LSBs		8
1	0	1	0	A	Y MSBs		4
1	0	1	1	B	Y LSBs		8
1	1	0	0	C	XLP (Light-pen)	Reserved	7
1	1	0	1	D	YLP (Light-pen)	Reserved	8
1	1	1	0	E	Reserved		–
1	1	1	1	F	Reserved		–

Reserved : These addresses are reserved for future versions of the circuit. In read mode, output buffers D0-D7 force a high state on the data bus.

TABLE 2 – COMMAND REGISTER

b3 b2 b1 b0		b7 b6 b5 b4 = 0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0 0 0	0	Set bit 1 of CTRL1 : Pen selection	Vector generation (for b2, b1, b0 see small vector definition)	SPACE	0	@	P	`	p								
0 0 0 1	1	Clear bit 1 of CTRL 1 : Eraser selection		!	1	A	Q	a	q								
0 0 1 0	2	Set bit 0 of CTRL1 : Pen/Eraser down selection		"	2	B	R	b	r								
0 0 1 1	3	Clear bit 0 of CTRL 1 : Pen/Eraser up selection		#	3	C	S	c	s								
0 1 0 0	4	Clear screen		$	4	D	T	d	t								
0 1 0 1	5	X and Y registers reset to 0		%	5	E	U	e	u								
0 1 1 0	6	X and Y reset to 0 and clear screen		&	6	F	V	f	v								
0 1 1 1	7	Clear screen, set CSIZE to code "minsize" All other registers reset to 0 (except XLP, YLP)		'	7	G	W	g	w								
1 0 0 0	8	Light-pen initialization (WHITE forced low)	Special direction vectors (for b2, b1, b0 see small vector definition)	(	8	H	X	h	x								
1 0 0 1	9	Light-pen initialization		)	9	I	Y	i	y								
1 0 1 0	A	5 x 8 block drawing (size according to CSIZE)		*	:	J	Z	j	z								
1 0 1 1	B	4 x 4 block drawing (size according to CSIZE)		+	;	K	[	k	{								
1 1 0 0	C	Screen scanning : Pen or Eraser as defined by CTRL1		,	<	L	\	l	¦								
1 1 0 1	D	X register reset to 0		–	=	M	]	m	}								
1 1 1 0	E	Y register reset to 0		.	>	N	↑	n	~								
1 1 1 1	F	Direct image memory access request for the next free cycle.		/	?	O	←	o	▒								

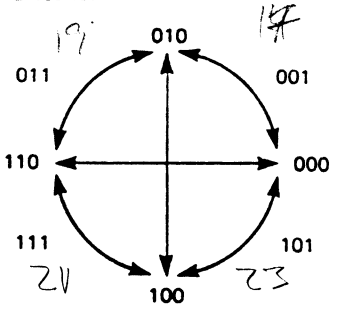
SMALL VECTOR DEFINITION :

b7	b6 b5	b4 b3	b2 b1 b0
1	|ΔX|	|ΔY|	Direction

Dimension

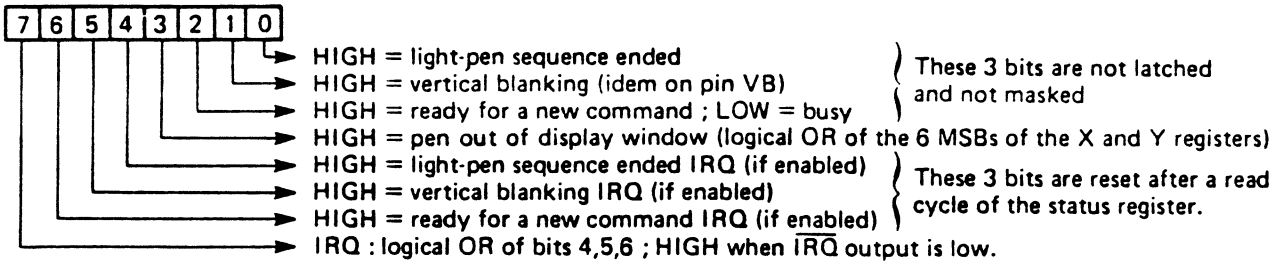
ΔX or ΔY	Vector length
0 0	0 step
0 1	1 step
1 0	2 steps
1 1	3 steps

Direction

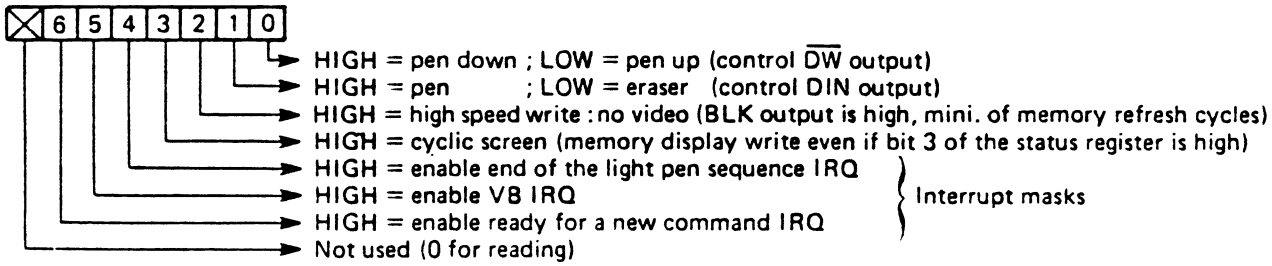


OTHER REGISTERS

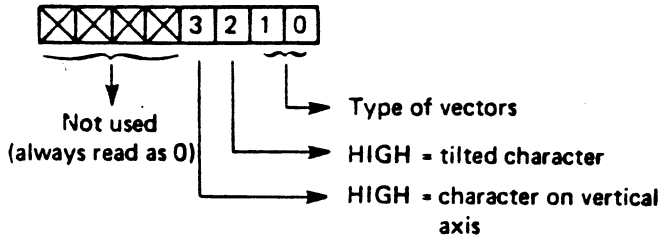
STATUS REGISTER (Read only)



CONTROL REGISTER 1 (Read/Write)

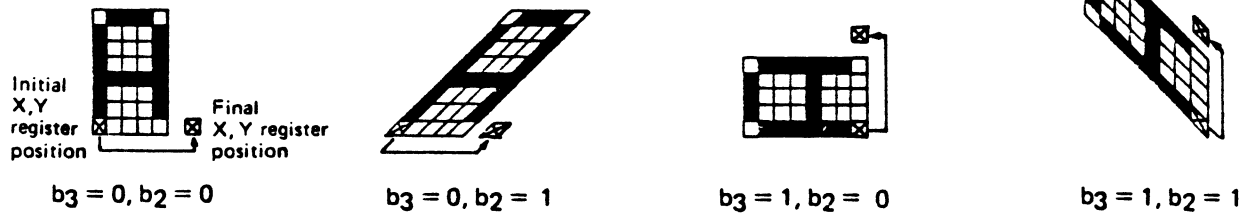


CONTROL REGISTER 2 (Read/Write)



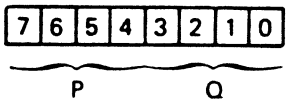
b1	b0	Type of vectors	
0	0	——— continuous	
0	1	------- dotted	2 dots on, 2 dots off
1	0	— — — dashed	4 dots on, 4 dots off
1	1	—·—· dotted-dashed	10 dots on, 2 dots off, 2 dots on, 2 dots off.

Types of character orientations



$b_3 = 0, b_2 = 0$ $b_3 = 0, b_2 = 1$ $b_3 = 1, b_2 = 0$ $b_3 = 1, b_2 = 1$

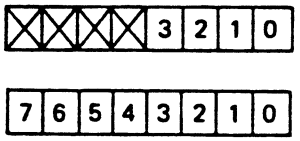
C-SIZE REGISTER (Read/Write)



P : Scaling factor on X axis
Q : Scaling factor on Y axis

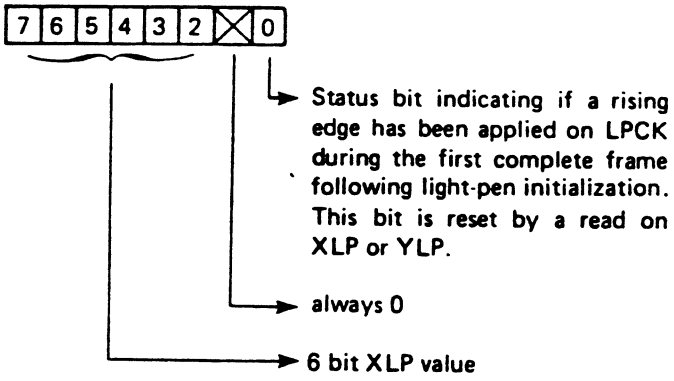
P and Q may take any value between 1 and 16. This value is given by the leftmost or rightmost 4 bits for P and Q respectively. Binary value (0) means 16.

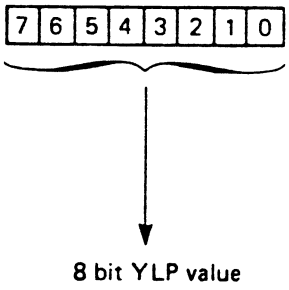
X AND Y REGISTERS (Read/Write)



The 4 leftmost MSBs are always 0.

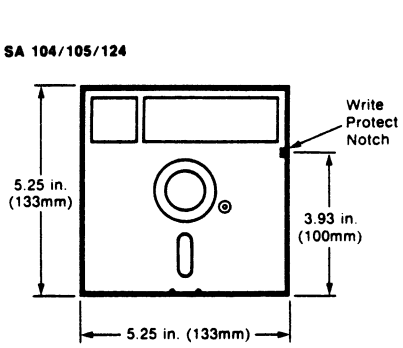
XLP and YLP REGISTERS





ASCII CHARACTER GENERATOR (5 x 8 matrix)

b7	0	0	0	0	0	0
b6	0	0	1	1	1	1
b5	1	1	0	0	1	1
b4	0	1	0	1	0	1
b3 b2 b1 b0						
0 0 0 0						
0 0 0 1						
0 0 1 0						
0 0 1 1						
0 1 0 0						
0 1 0 1						
0 1 1 0						
0 1 1 1						
1 0 0 0						
1 0 0 1						
1 0 1 0						
1 0 1 1						
1 1 0 0						
1 1 0 1						
1 1 1 0						
1 1 1 1						



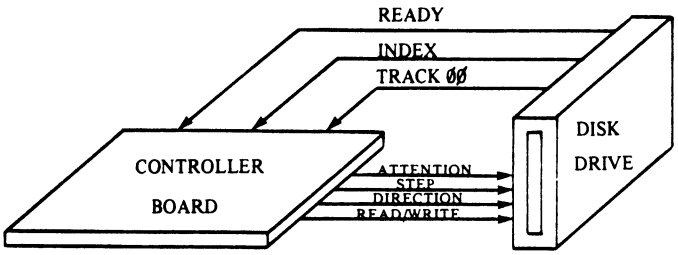


Fig. 4-90: Basic Floppy Drive Signals

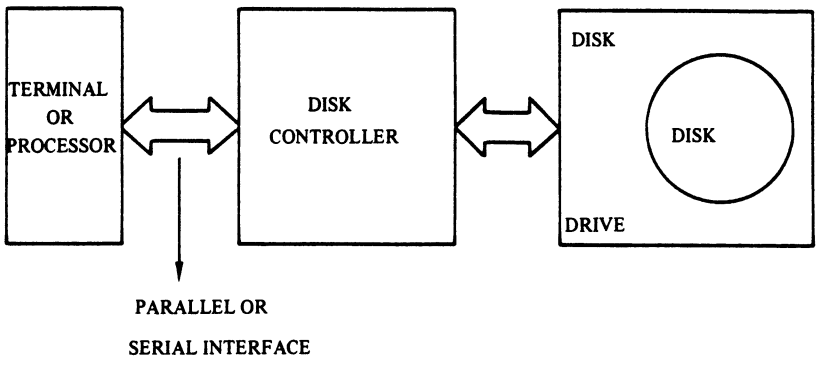


Fig. 4-95: Disk Controller Interfaces Drive to Processor

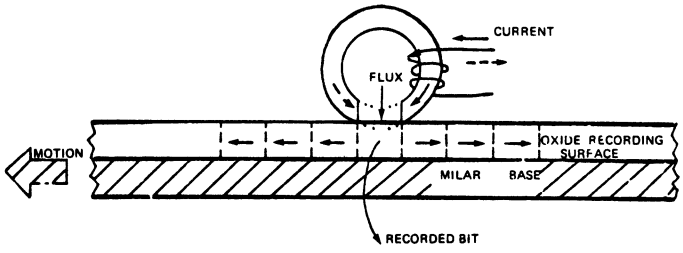


Fig. 4-96: Recording a Bit on a Disk

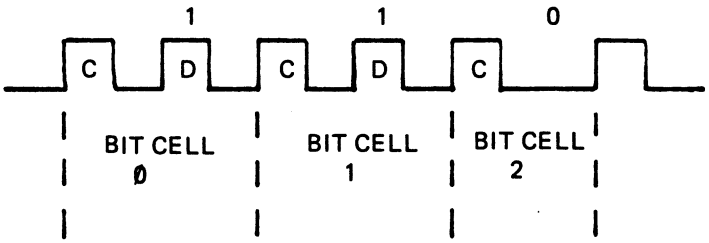


Fig. 4-97: Representing Clock and Data

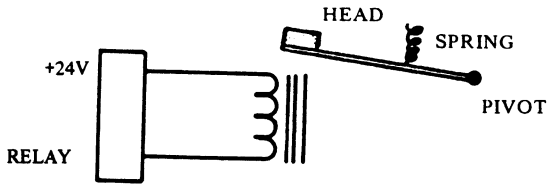
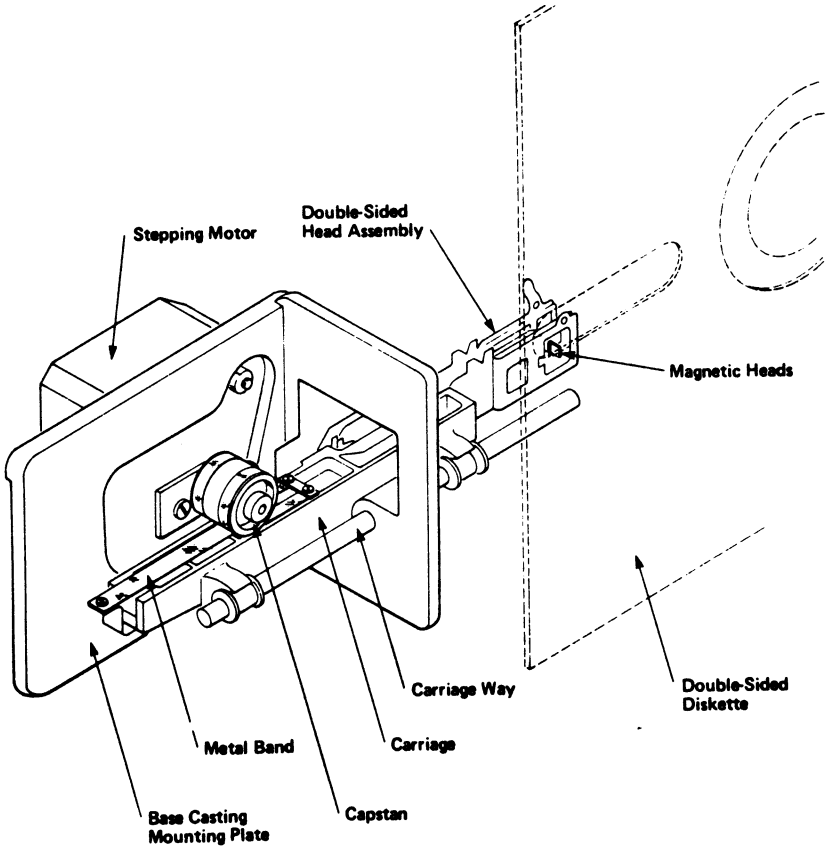
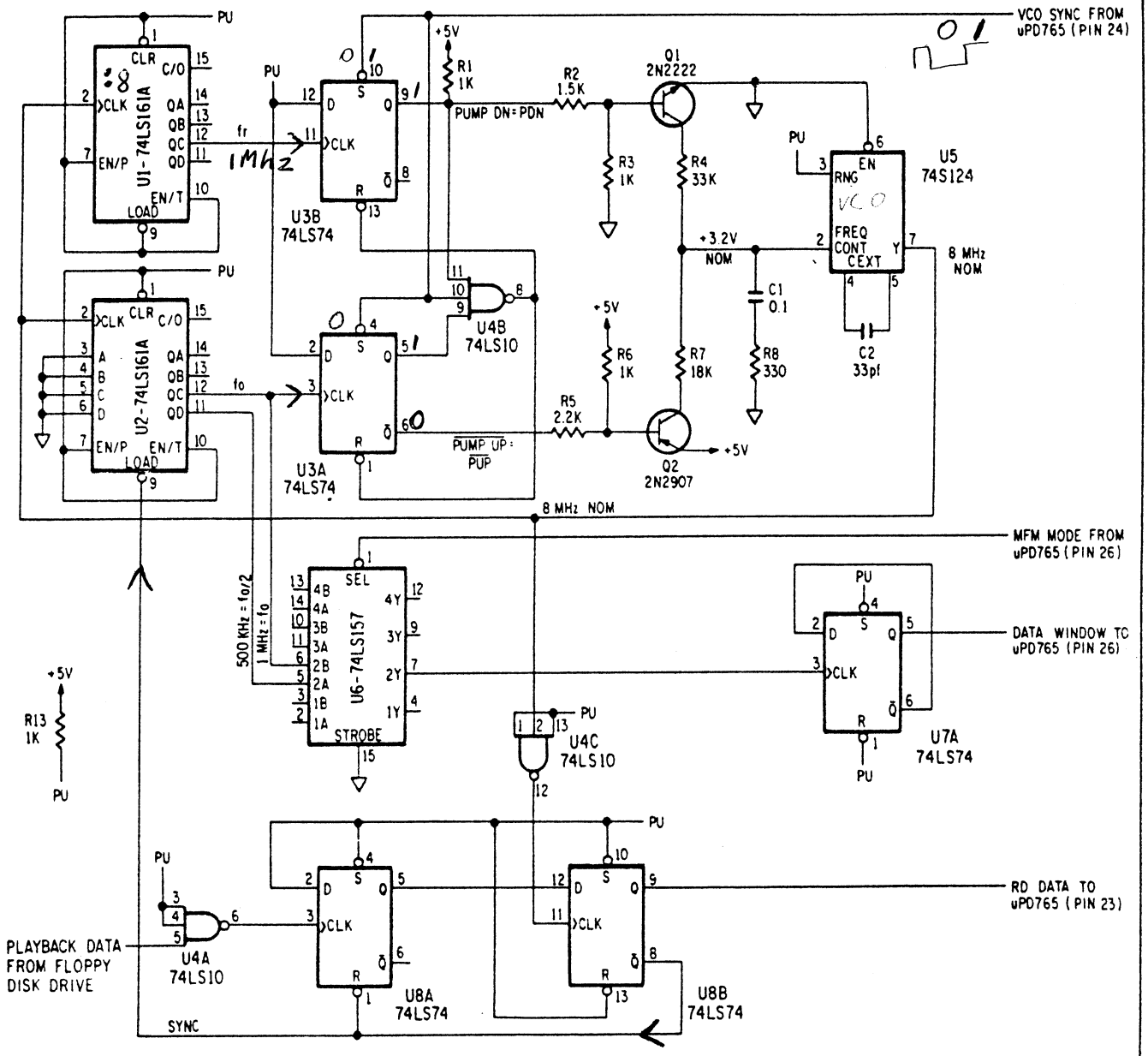


Fig. 4-83: Disk Drive: Head Load Assembly



Phase Locked Loop For FM/MFM Encoded Data

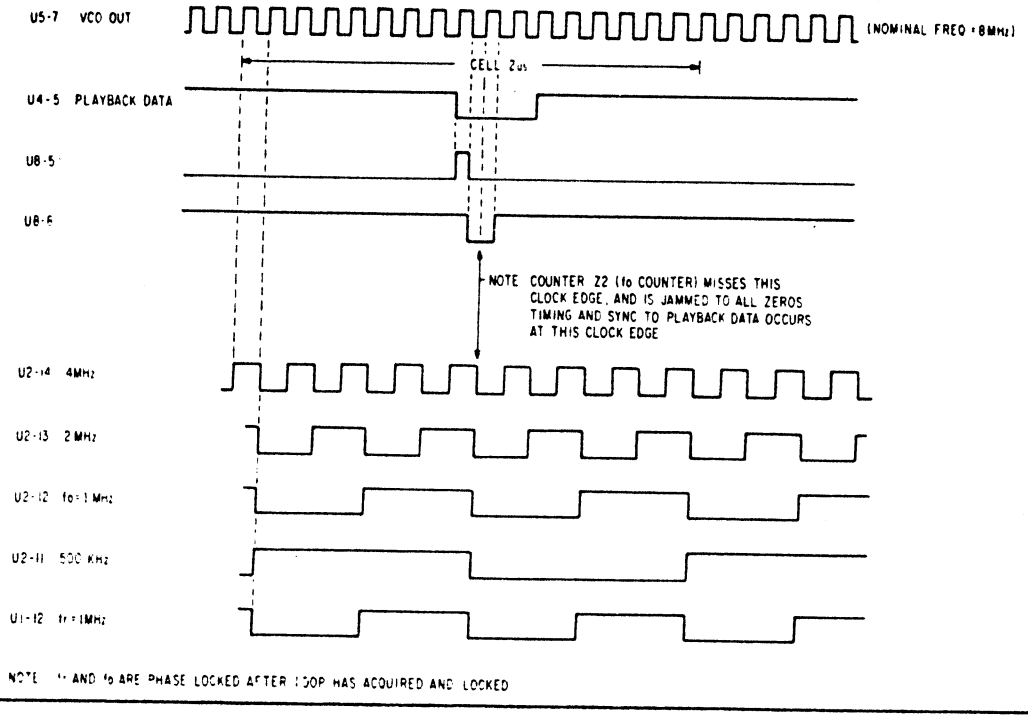
Figure 13



Figures 14 and 15 show the timing relationship between these signals under both ideal and worst-case bit patterns. Note in Figure 15 how the RD Data pulse moves around within the Data Window under worst-case peak shift conditions. With this circuit, peak-shifts of up to 375 ns (500 ns theoretical maximum) may be tolerated in MFM encoded data before read errors will be encountered.

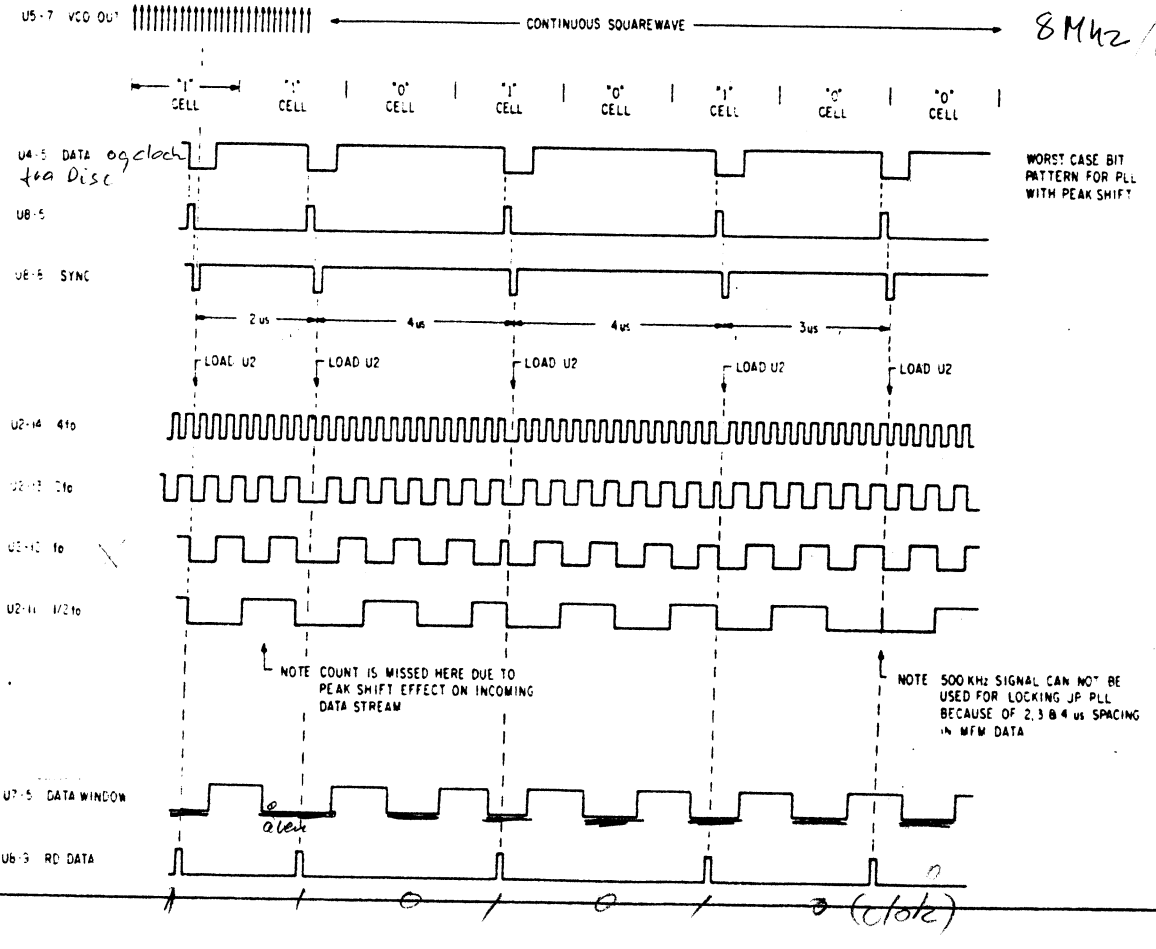
Synchronization of fo Counter with FDD Data

Figure 14



fo Counter Sync with Peak Shift

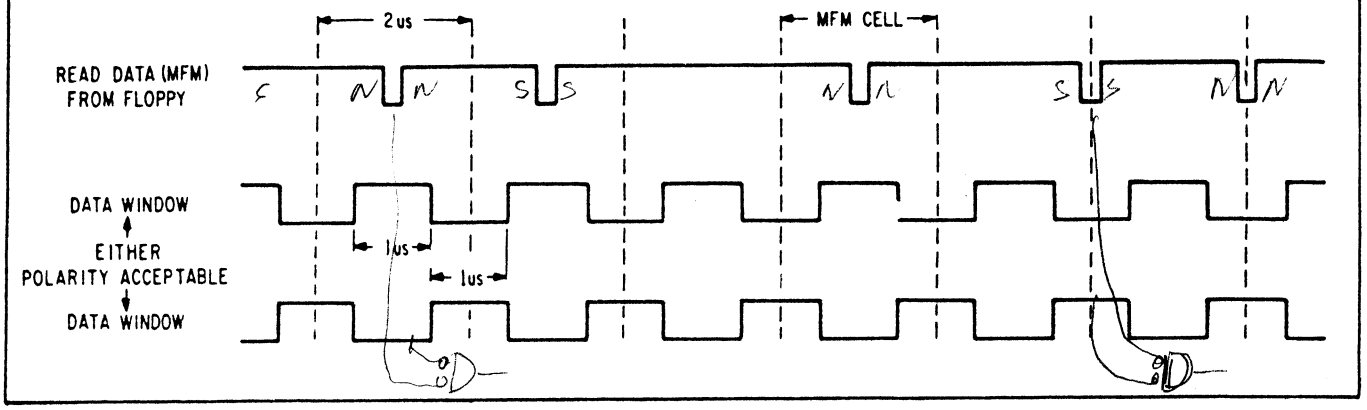
Figure 15



The MFM mode signal is high (logic 1) when the unit has been programmed to receive MFM (double density) data and is low (logic 0) for FM (single density) data. This signal allows the data recovery circuit to switch between single and double density modes of operation.

The μPD765 requires a Data Window signal (pin 22) which brackets the data coming from the drive. Figure 12 shows the timing relationship between these signals. Note how the Data Window brackets both the beginning and center of the cell. Also note that either polarity Data Window signal is acceptable to the μPD765. When single density (FM) data is to be processed, times shown in Figure 12 must be multiplied by two. For minifloppies the times shown in Figure 12 must be multiplied by two (for double density minifloppies) and by four (for single density minifloppies).

Data Window Signal Required by μPD765 Figure 12



CAUTION: The PLL described here is meant as an example, and may not prove entirely adequate. For an improved version please refer to Application Note #10, and the specification of your drive.

A convenient way to generate this Data Window signal is with a Phase Locked Loop (PLL). The PLL tracks the playback signal from the drive and performs the function of an adaptive low-pass filter. Many excellent technical articles have been written on the virtues of the PLL in this application, and it is only necessary to say that the PLL will work very well for data recovery of floppy disk data if two key factors are remembered:

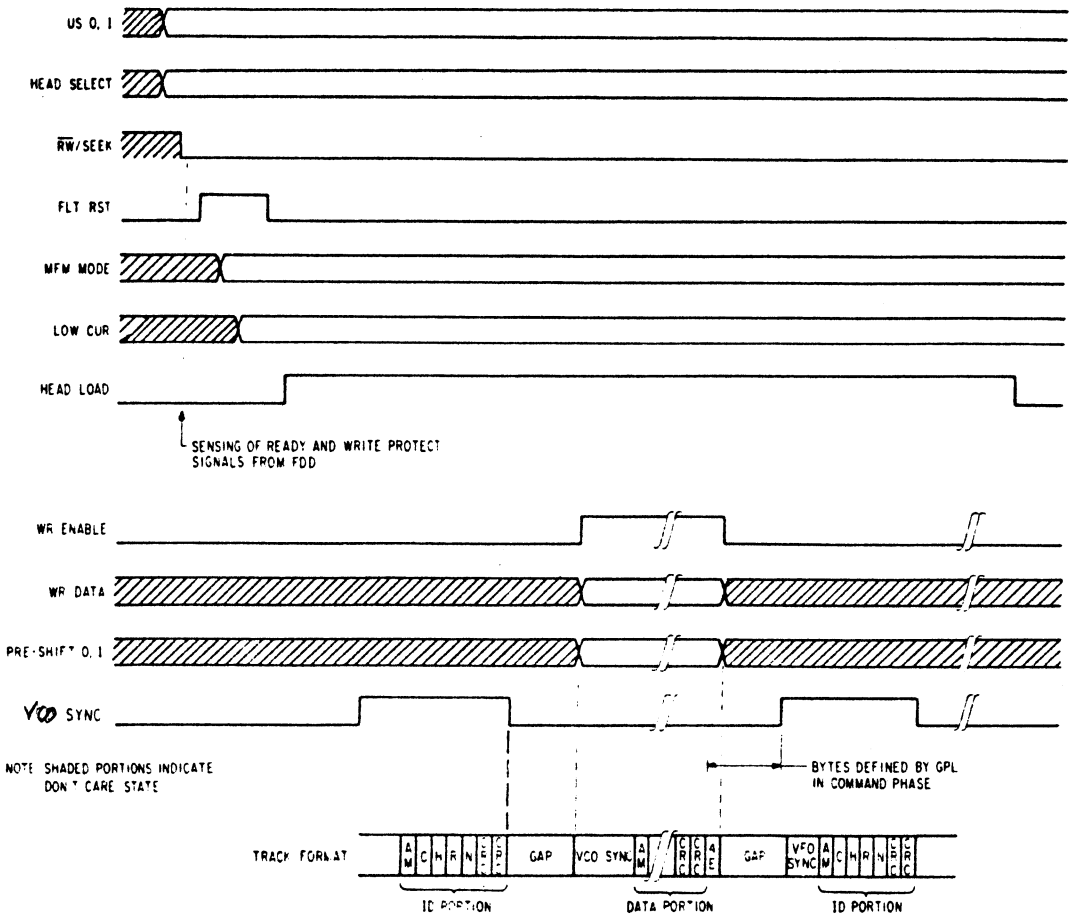
- The data must be periodic in nature. (FM and MFM data exhibit this characteristic if they are adequately pre- or post-compensated.)
- Adequate allowances must be made for both the VCO Sync period and the discontinuities on the diskette which result when the Write current was turned on or off.

Phase Locked Loop Data Recovery Circuit

Figure 13 shows a design for a phase locked loop data recovery circuit interfaced to the μPD765. The circuit operates from a single +5V power supply and is implemented primarily with digital I.C.'s. The only analog circuitry is the low pass filter circuit (R4, R7, R8 and C1) and two transistors (Q1 and Q2) which interface to it. I.C.'s U1 and U2 are the heart of the PLL circuit; these generate two 1 MHz signals fr (reference) and fo (signal). The two counter outputs U1-12 and U2-12 are phase locked after acquisition and lock have occurred in the loop. Counter U1 is a free-running counter and simply counts down the VCO output (U5-7). Counter U2 is pre-loaded to all zeros every time a flux reversal is read from the floppy disk drive. U8-5 and U8-8 perform the function of synchronizing the playback data from the drive to the VCO output, as well as generating the load pulse to counter U2-9.

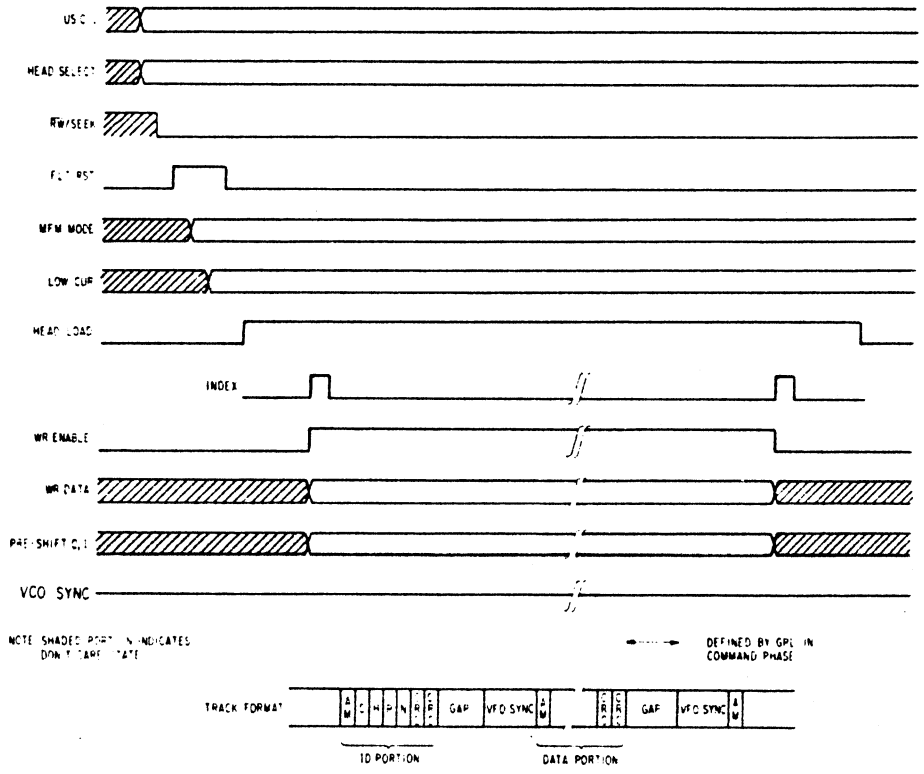
Write Data and Write Deleted Data Instructions

Figure 8



Timing at FDD Interface

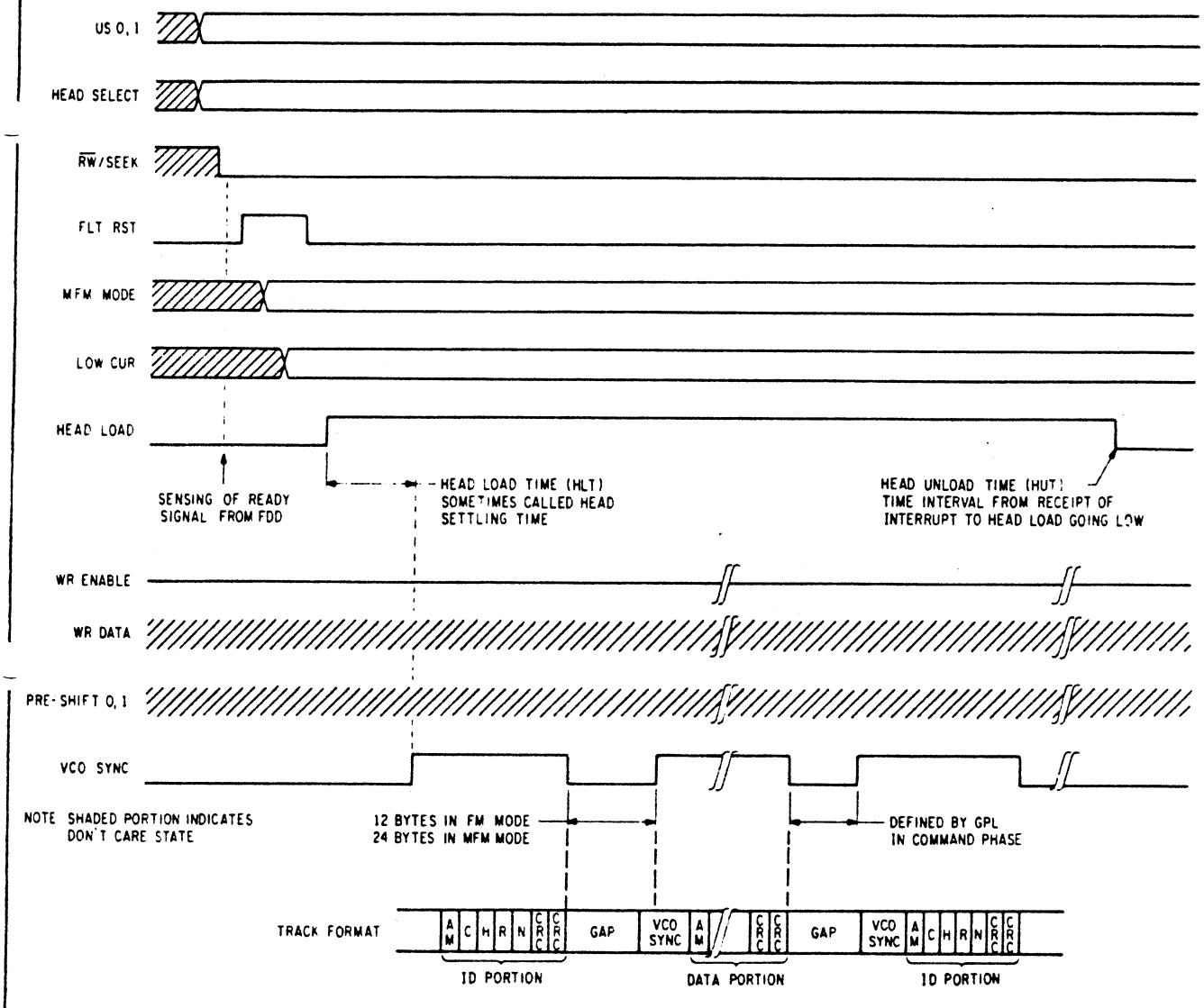
Figure 9



The schematic in Figure 18 shows most of the configurations which are possible with both standard and minifloppies. However, it is highly unlikely that anyone would want all the optional features shown. As a result one should read the notes at the bottom of the schematic carefully and eliminate the necessary IC's before starting to build this controller.

Figures 7 through 11 show the sequence of events which occur at the μPD765/drive interface for each of the instructions. Detailed timing information for each of these signals is provided in the μPD765 data sheet.

Read Data, Read Deleted Data, Read A Track, Scan and Read ID Instructions Figure 7

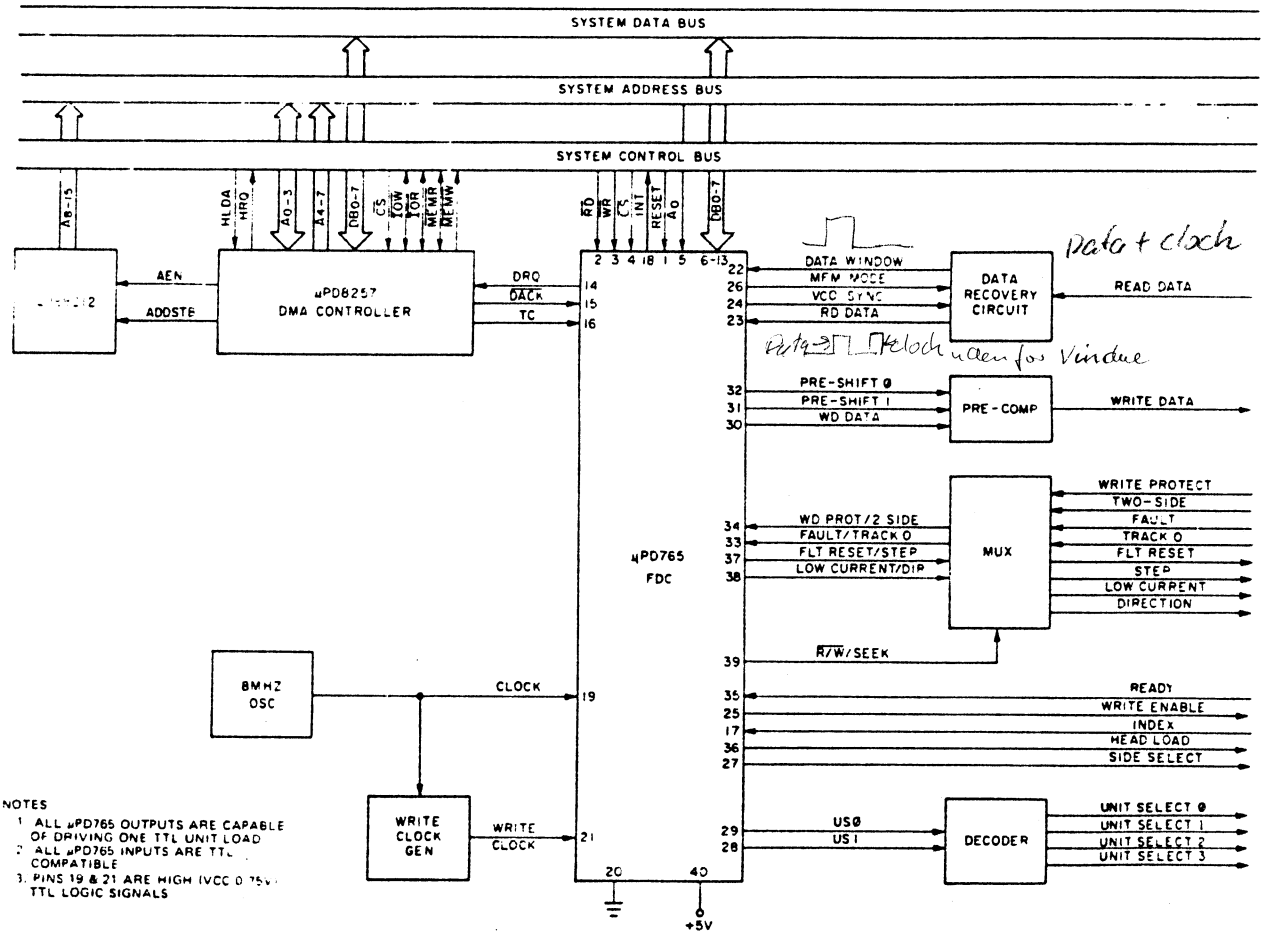


System Block Diagram

Figure 1 shows how the μPD765 may be configured in a typical system. The μPD765 may be easily interfaced to any of the popular microprocessors such as μPD8080A, μPD8085A, μPD780 (Z80™), 6800, 6502, etc. If the FDC is to be interrupt-driven instead of DMA-driven, then the μPB8212 and the μPD8257 (or other hardware associated with the DMA Controller) may be eliminated and their functions will be taken over by the processor.

Two clocks are required by the μPD765. The clock on pin 18 (CLK) is 8 MHz for 8" drive and 4 MHz for 5 1/4" drive and may be totally asynchronous with the processor or system clock. The Write Clock on pin 21 (WCK) controls the rate at which data is sent to the drive electronics (for writing onto the diskette) and is either 500 KHz or 1 MHz, depending upon whether single or double density data is to be written (see page 5 of the μPD765 data sheet). There are 19 lines which interface to the floppy disk drive. Several of these signals may be connected directly to the drive. Other signals require demultiplexing, decoding or signal conditioning. The external circuits that perform these functions require typically only 5.5 TTL I.C.'s. The DMA interface consists of only three signals: DRQ (DMA Request, pin 14), $\overline{\text{DACK}}$ (DMA Acknowledge, pin 15) and TC (Terminal Count, pin 16). These three signals interface directly to the μPD8257 DMA Controller as well as many other popular DMA Controller I.C.'s.

System Block Diagram Figure 1



TM: Z80 is a registered trademark of Zilog, Inc.

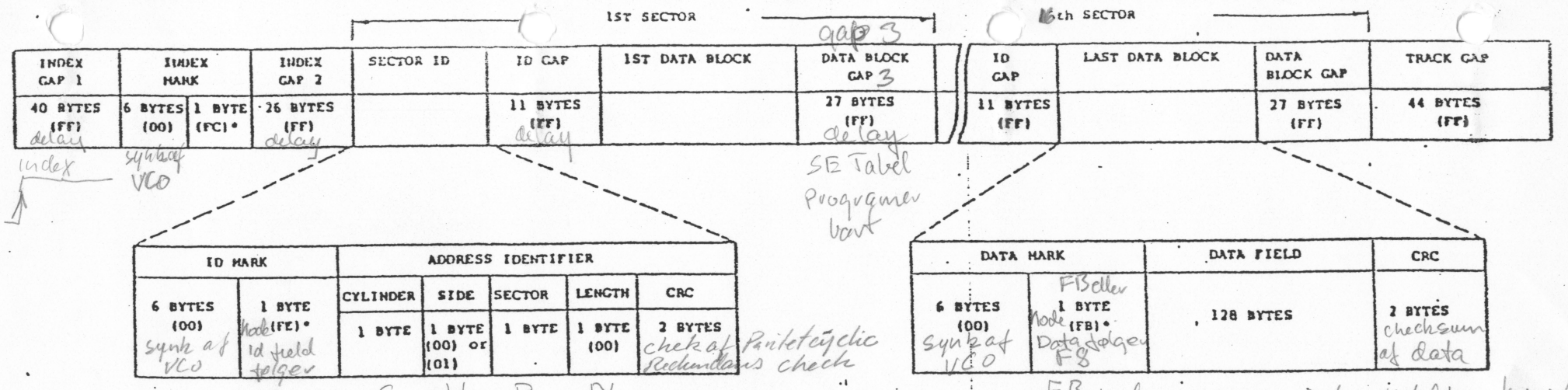


Figure 9. FM format example

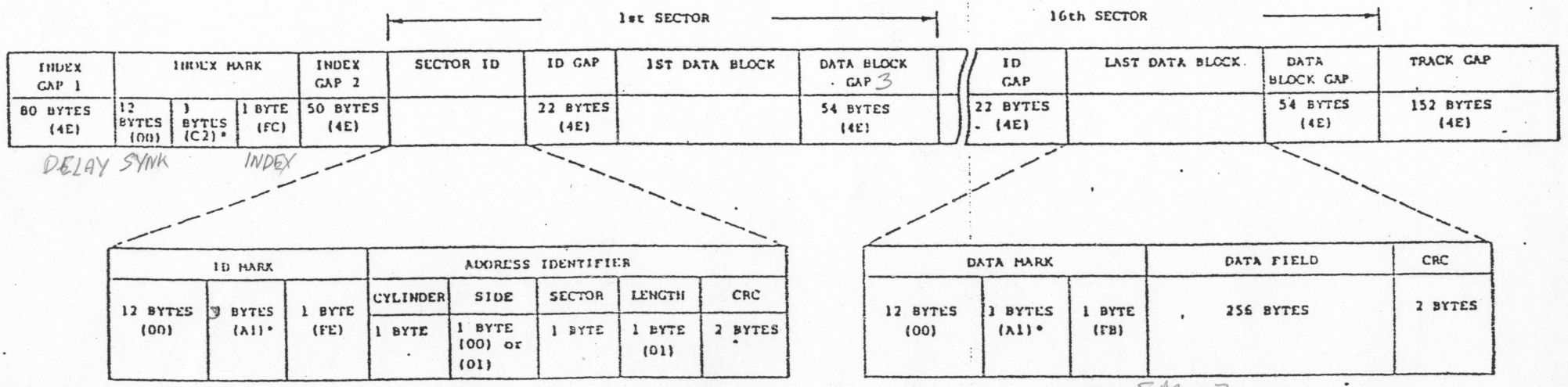


Figure 10 MFM format example

Notes:
1. Codes given in brackets are of hex notation.
2. The clock for (FC)* is D7(H).
3. The clock for (FE)* is C7(H).
4. The clock for (FB)* is C7(H).
5. (A1)* is a special MFM in which the change between bits 3 and 4 is missing.
6. (C2)* is a special MFM in which the change between bits 4 and 5 is missing.

RECORDING FORMAT

The recording format is dependent upon requirements of the controller. The track and sector organization of data is dependent on the format.

Encoding Scheme

The drive allows double-density or single-density encoding schemes. In double-density recording, each bit cell is 2 microseconds wide, in single-density recording, each bit cell is 4 microseconds wide (see Figure 1-3).

Track Format

The floppy disk contains 77 tracks. The first (outside) track is track 00, and the last (inside) track is 76. The top read/write head is for tracks offset from the bottom read/write head. The offset direction is toward the spindle. During the write operation, an erase coil in the read/write head erases the outside edges of the data just written, narrowing the data track. In this manner, a guard band is

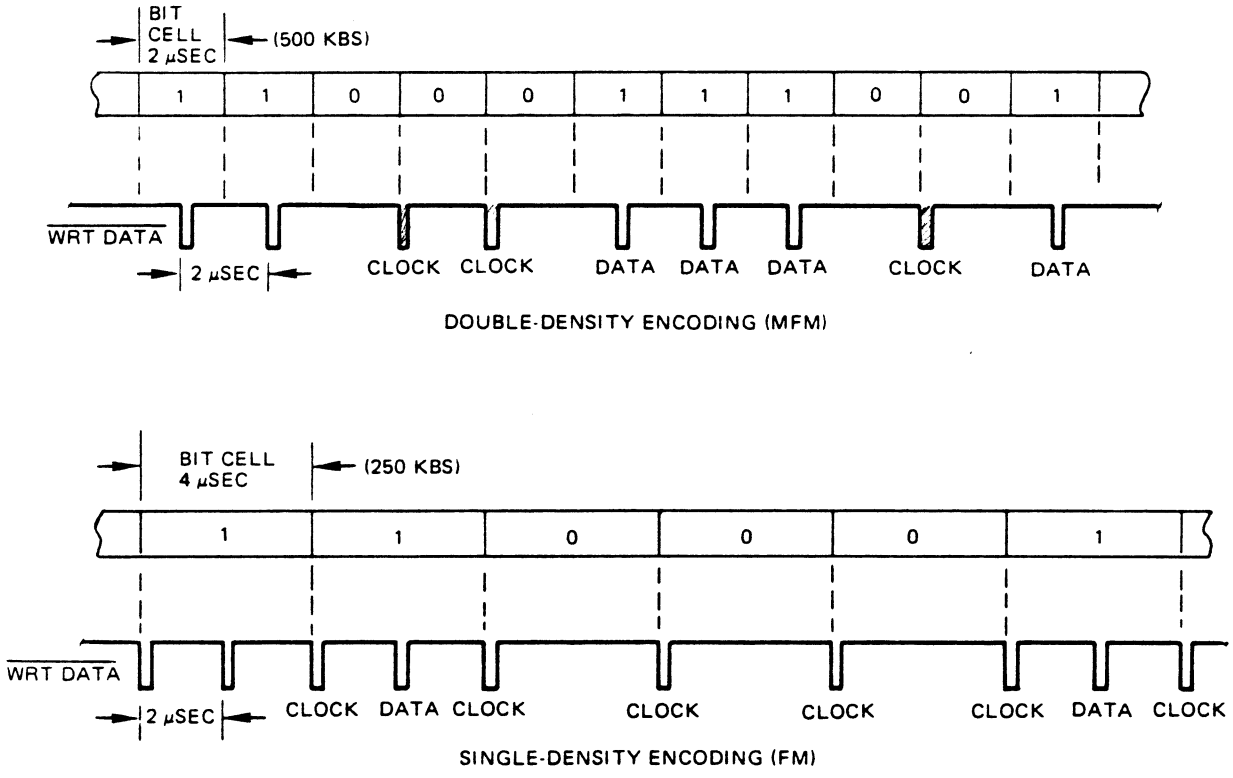


Figure 1-3. Single and Double Density Encoding

established to protect the data from adjacent track crosstalk when reading. The erase coil will be either a tunnel or straddle type. The tunnel erase method requires time delays since it is 34 milliseconds behind the R/W coil. The straddle erase head does not require any time delays.

Sector Format

The number of sectors in each track is determined by the application, and can range from 1 to 32, depending on whether the soft-sector or hard-sector floppy disk is being used.

When soft sector operation is required, only one index hole is punched in the floppy disk. With this disk, the controller uses the index pulse to define the sectors. When hard sector operation is required, the floppy disk used contains the index hole plus 32 sector holes spaced equidistant around the disk (see Figure 1-2).

The index hole is punched midway between sector holes 31 and 0. The double-pulse of sector 31 and index alerts the controller that the next pulse starts sector 0. The index and sector holes are sensed photoelectrically, providing the pulses supplied to the controller.

Sector Content

The format of each sector is determined by the application. Normally, preambles and postambles containing a stream of coded bytes are written at the beginning and end of each sector, to provide data synchronization. Following the preamble of each new track, an identification (ID) field is written containing the track and sector numbers. Following the ID field, data bytes are written.

32-Sector Format

This format is not the most efficient OEM format due to the number of gaps required between data records. A typical 32-sector format is shown in Figure 1-4.

IBM 3740 Format

There are two IBM 3740 formats; Data Set Label and Track. The disk drive is compatible to both formats.

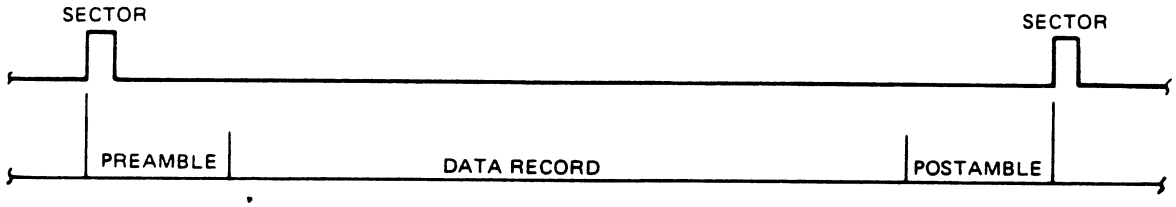


Figure 1-4. 32-Sector Format

Track 00 contains only Data Set Labels that identify the type of information stored in tracks 01 through 76. Tracks 01 through 73, 75, and 76 are allocated 26 sectors, each containing 128 data bytes. A data set may be one or more sectors, including overflow to each on-line disk drives. In the drive, only tracks 01 through 73 are normally used. Track 74 and 75 are reserved as spares to be used when other tracks become flawed, and track 76 is not used. The IBM 3740 format is shown in Figure 1-5. For detailed information on the IBM 3740 data format and initialization, refer to IBM Publication GA21-9190.

DISK DRIVE ASSEMBLY

The disk drive assembly can be installed in a standard 19-inch RETMA rack; two horizontally, or four vertically.

The disk drive comprises three major assemblies:

- Printed Circuit Board (Electronics)
- Main Deck Assembly
- Carrier Assembly

Printed Circuit Board

All electronic circuitry required to convert the digital data input and output to and from analog data for the read/write heads and head positioning information is contained on one circuit board. Interface and DC connectors can be provided. Logic is TTL with selected discrete and IC Components. The electronics perform the following functions:

- Read Chain
- Write Chain
- Ready Generation

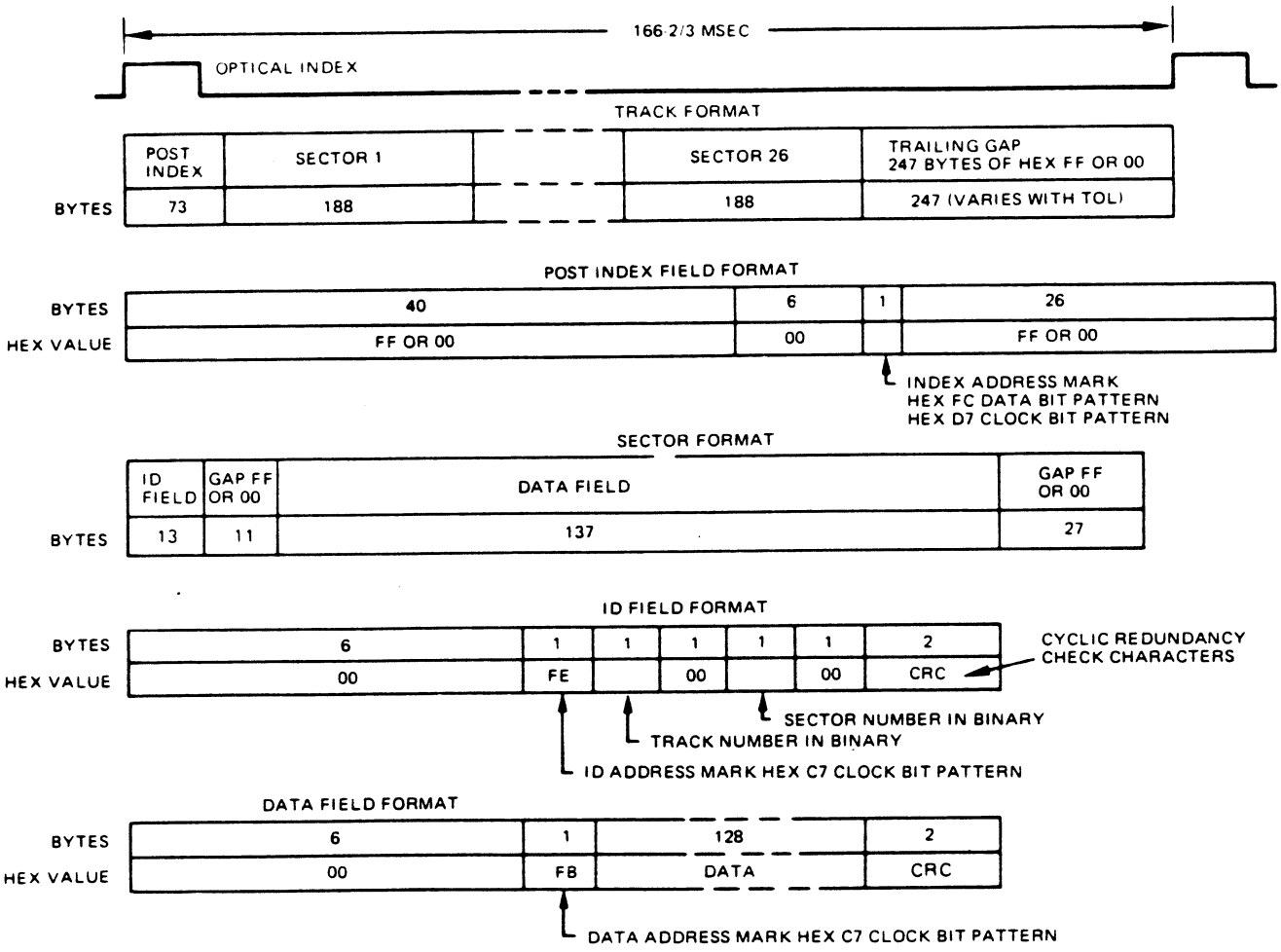


Figure 1-5. IBM 3740 Format

- Index Detection
- Stepper Motor Control
- Interface Drivers and Receivers
- Write-Protect
- Index/Sector Separator (Option)
- FM Data Separator (Option)
- FM Data Separator for Missing Clock Patterns (Optional)*
- Binary Select (Option)

*This option can be used to detect address marks.

TOSHIBA

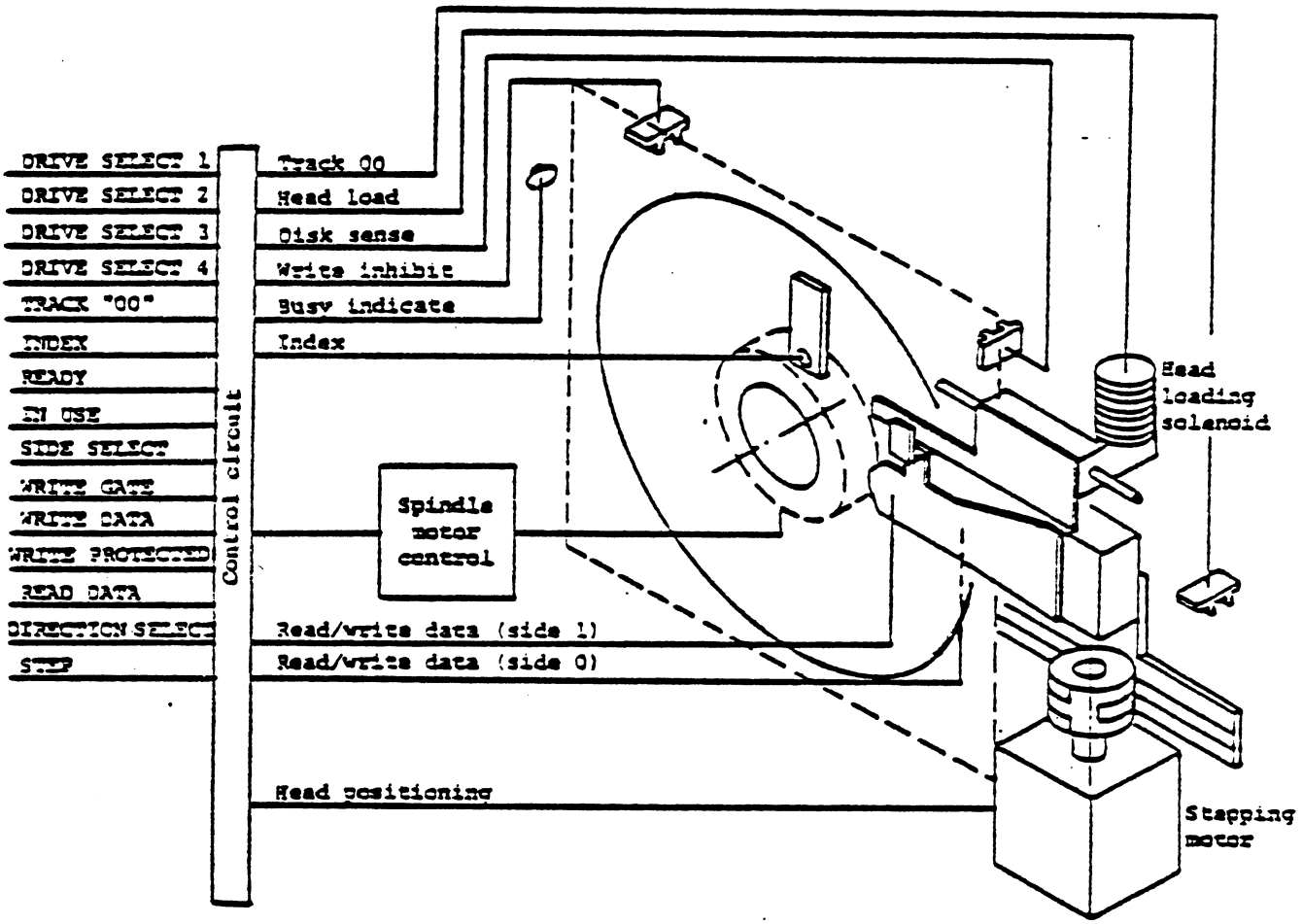


Figure 3 Device configuration

6. INTERFACE

Interface lines consist of signal and power supply lines.

6.1 SIGNAL LINES

Signal lines consist of input and output signal lines.
Their logic level specifications are as follows:

Low = 0.0 to +0.4 V = logic 1 (true)

High = +2.5 to +5.25 V = logic 0 (false)

6.1.1 Input Signal Lines

The input signal lines to the ND-06D consist of the following eleven signal lines:

(1) Drive Select 1-4

When one of the four Drive Select lines is set to logic 1, the corresponding drive becomes ready to respond to other input signal lines and opens its output signal gates.
Up to four drives are controllable with these four Drive Select lines.
Which drive responds to Drive Select lines 1-4 is determined by the jumper connection (using shorting plugs) within each drive.

(2) Step

The Step signal line is used to move the heads. It is used with the Direction Select signal line. The trailing edge (transition from 1 to 0) of a pulse on the Step line causes the heads to move across one track. However, if the drive is in Write mode, no head traverse takes place.
The head stabilizes 18 ms after the trailing edge of the last step pulse and the drive becomes ready for data read/write operations.

(3) Direction Select

This input signal line determines the direction of head movement which is caused by step pulses on the Step line. When the Direction Select line is set to 1, step pulses on the Step line cause the heads to step in. When it is set to 0, step pulses cause the heads to step out.
The logic level on the Direction Select line must be valid at least 1 μs before the trailing edge of a step pulse.

(4) Side Select

Active state of the Side Select line selects the head on side 1; inactive state of the line selects the head on side 0. Side 0 corresponds the recording side of a single-sided disk.
Side select is completed 100 μs after a transition of the Side Select line, and the drive becomes ready for read/write operations.

(5) Write Gate

When set to 1, the Write Gate line enables the write circuit of the drive. When reset to 0, the Write Gate line places the drive in Read mode. It takes 1.3 ms before valid read data appears on the interface after the end of write operation.

(6) Write Data

This input line is used to transfer write data to the drive. Data signals modulated with clock pulses are carried on this Write Data line. Data is written onto the disk at the leading edge (transition from 0 to 1) of the pulses on this line.
Write Pre-compensation is recommended for MFM recording.

Minimum;	100 ns	for all tracks
Optimum;	100 ns	for tracks from 00 to 39
	300 ns	for tracks from 40 to 79

(7) In Use

If the In Use line is set to 1, the LED indicator on the front panel of the drive comes on. This signal line is valid independently of the Drive Select line. Refer to 7.3 for detailed usage of LED indicator.

(8) Motor On

The Motor On line is used to control the spindle motor.
If this line is set to 1, the spindle motor starts rotating.
A start-up time is approximately 0.8 seconds.
The Motor On line is valid independently of the Drive Select line.

6.1.2 Output Signal Lines

The output signal lines of the ND-06D consist of the following five lines:

(1) Index

This line is used to index the beginning of each track.
An active pulse is sent on this line for each rotation of the disk. The index timing is shown in Figure 19.

(2) Read Data

This line is used to transfer data read from the disk. A mixture of data pulses and clock pulses are carried on this line. The timing is shown in Figure 14.

(3) Track 00

Active state (1) of this signal line indicates that the heads are located on track 00 and also indicates that a specific phase of the step motor is energized.

(4) Ready

Active state of this line indicates that a disk is loaded, the door is closed, the disk has reached the specified rotational speed, and the drive is ready for read/write operation.

(5) Write Protected

Active state of this line indicates that the inserted disk is write protected (the write enable notch is covered with a label). Write operation to the disk is ignored.

6.1.3 Input Signal Line Termination

Since the ND-06D is designed to be used with daisy chain connection to its host system, seven input lines out of eleven must be terminated on the furthest drive on the chain. The remaining four input lines, Drive Select lines 1-4, are terminated on each drive.
When the drives are used in a radial configuration, all input lines should be terminated on each drive.
For example, if the drives are attached to the host in the daisy chain configuration as shown in Figure 20, the seven input lines must be terminated on drive 3. If the drives are in the radial configuration as shown in Figure 21, the input lines should be terminated on each drive.
The terminating resistors are of DIP resistor modules which are designed to be mounted on IC sockets (mounted when shipped). For the drives requiring no termination, the resistor modules should be removed.

NEC

μPD765A/μPD7265
SINGLE/DOUBLE DENSITY FLOPPY DISK CONTROLLER

Description

The μPD765A is an LSI Floppy Disk Controller (FDC) chip which contains the circuitry and control functions for interfacing a processor to 4 floppy-disk drives. It is capable of supporting either IBM 3740 Single Density format (FM), or IBM System 34 Double Density format (MFM) including double-sided recording. The μPD765A provides control signals which simplify the design of an external phase locked loop and write precompensation circuitry. The FDC simplifies and handles most of the burdens associated with implementing a floppy-disk interface.

The μPD7265 is an addition to the FDC family that has been designed specifically for the Sony Micro Floppydisk® drive. The μPD7265 is pin-compatible and electrically equivalent to the 765A but utilizes the Sony recording format. The μPD7265 can read a diskette that has been formatted by the μPD765A.

Hand-shaking signals are provided in the μPD765A μPD7265 which make DMA operation easy to incorporate with the aid of an external DMA Controller chip, such as the μPD8257. The FDC will operate in either the DMA or non-DMA mode. In the non-DMA mode the FDC generates interrupts to the processor every time a data byte is to be transferred. In the DMA mode, the processor need only load the command into the FDC and all data transfers occur under control of the FDC and DMA controllers.

There are 15 commands which the μPD765A μPD7265 will execute. Each of these commands requires multiple 8-bit bytes to fully specify the operation which the processor wishes the FDC to perform. The following commands are available:

READ DATA	SCAN HIGH OR EQUAL	WRITE DELETED DATA
READ ID	SCAN LOW OR EQUAL	SEEK
SPECIFY	READ DELETED DATA	RECALIBRATE
READ TRACK	WRITE DATA	SENSE INTERRUPT STATUS
SCAN EQUAL	FORMAT TRACK	SENSE DRIVE STATUS

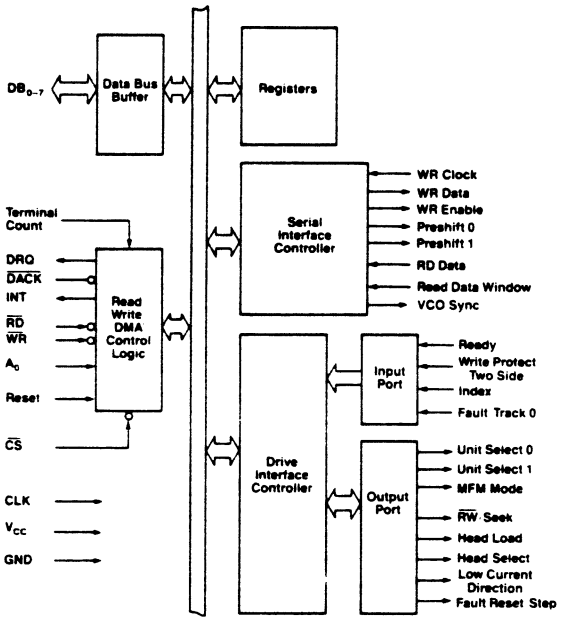
Features

Address Mark detection circuitry is internal to the FDC which simplifies the phase locked loop and read electronics. The track stepping rate, head load time, and head unload time are user-programmable. The μPD765A/μPD7265 offers additional features such as multitrack and multiside read and write commands and single and double density capabilities.

- ☐ Sony (EMCA) Compatible Recording Format (μPD7265)
- ☐ IBM-compatible Format (Single and Double Density) (μPD765A)
- ☐ Multisector and Multitrack Transfer Capability
- ☐ Drive Up to 4 Floppy or Micro Floppydisk® Drives
- ☐ Data Scan Capability—Will scan a single sector or an entire cylinder comparing byte-for-byte host memory and disk data
- ☐ Data Transfers in DMA or Non-DMA Mode
- ☐ Parallel Seek Operations on Up to Four Drives
- ☐ Compatible with μPD8080/85, μPD8086/88 and μPD780 (Z80™) Microprocessors
- ☐ Single Phase Clock (8 MHz)
- ☐ +5V Only
- ☐ 40-Pin Plastic Package

™Z80 is a registered trademark of Zilog Inc.
Micro Floppydisk® is a registered trademark of Sony Corporation.

Block Diagram



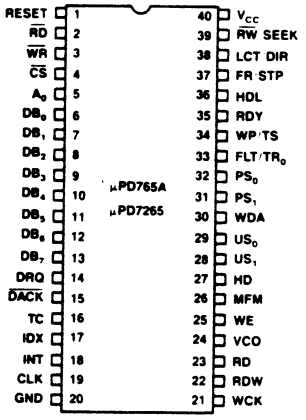
Absolute Maximum Ratings*

T_a = 25°C	
Operating Temperature	−10°C to +70°C
Storage Temperature	−40°C to +125°C
All Output Voltages	−0.5 to +7V
All Input Voltages	−0.5 to +7V
Supply Voltage V_{CC}	−0.5 to +7V
Power Dissipation	1W

*COMMENT: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

μPD765A/7265

Pin Configuration



DC Characteristics

T_a = −10°C to +70°C;
V_{CC} = +5V ± 5% unless otherwise specified

Parameter	Symbol	Limits Min	Typ ①	Max	Unit	Test Conditions
Input Low Voltage	V_{IL}	−0.5		0.8	V	
Input High Voltage	V_{IH}	2.0		V_{CC} + 0.5	V	
Output Low Voltage	V_{OL}			0.45	V	I_{OL} = 2.0 mA
Output High Voltage	V_{OH}	2.4		V_{CC}	V	I_{OH} = −200 μA
Input Low Voltage (CLK + WR Clock)	$V_{IL}(\phi)$	−0.5		0.65	V	
Input High Voltage (CLK + WR Clock)	$V_{IH}(\phi)$	2.4		V_{CC} + 0.5	V	
V_{CC} Supply Current	I_{CC}			150	mA	
Input Load Current	I_{LI}			10	μA	V_{IN} = V_{CC}
(All Input Pins)				−10	μA	V_{IN} = 0V
High Level Output Leakage Current	I_{LOH}			10	μA	V_{OUT} = V_{CC}
Low Level Output Leakage Current	I_{LOL}			−10	μA	V_{OUT} = +0.45V

Note: ① Typical values for T_a = 25 C and nominal supply voltage

Capacitance

T_a = 25°C; f_c = 1MHz; V_{CC} = 0V

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Clock Input Capacitance	$C_{IN(\phi)}$			20	pF	All pins except pin under test tied to AC Ground
Input Capacitance	C_{IN}			10	pF	
Output Capacitance	C_{OUT}			20	pF	

Pin Identification

Pin No.	Symbol	Name	I/O	Connects To	Function
1	RST	Reset	I	Processor	Places FDC in idle state. Resets output lines to FDD to 0 (low). Does not affect SRT, HUT or HLT in Specify command. If RDY pin is held high during Reset, FDC will generate an interrupt within 1.024 msec. To clear this interrupt use Sense Interrupt Status command.
2	$\overline{RD}$	Read	I	Processor	Control signal for transfer of data from FDC to Data Bus, when 0 (low).
3	$\overline{WR}$	Write	I	Processor	Control signal for transfer of data to FDC via Data Bus, when 0 (low).
4	$\overline{CS}$	Chip Select	I	Processor	IC selected when 0 (low), allowing RD and WR to be enabled.
5	A_0	Data Status Reg Select	I	Processor	Selects Data Reg (A_0 = 1) or Status Reg (A_0 = 0) contents of the FDC to be sent to Data Bus.
6–13	DB_0–DB_7	Data Bus	I/O	Processor	Bidirectional 8-bit Data Bus.
14	DRQ	Data DMA Request	O	DMA	DMA Request is being made by FDC when DRQ = 1.
15	$\overline{DACK}$	DMA Acknowledge	I	DMA	DMA cycle is active when 0 (low) and controller is performing DMA transfer.
16	TC	Terminal Count	I	DMA	Indicates the termination of a DMA transfer, when 1 (high). It terminates data transfer during Read/Write/Scan command in DMA or Interrupt mode.
17	IDX	Index	I	FDD	Indicates the beginning of a disk track.
18	INT	Interrupt	O	Processor	Interrupt Request generated by FDC.
19	CLK	Clock	I		Single phase 8 MHz square-wave clock.
20	GND	Ground			DC power return.
21	WCK	Write Clock	I		Write data rate to FDD. FM = 500 KHz, MFM = 1 MHz, with a pulse width of 250 ns for both FM and MFM.
22	RDW	Read Data Window	I	Phase Lock Loop	Generated by PLL, and used to sample data from FDD.
23	RDD	Read Data	I	FDD	Read data from FDD, containing clock and data bits.
24	VCO/Sync	VCO/Sync	O	Phase Locked Loop	Inhibits VCO in PLL when 0 (low), enables VCO when 1.
25	WE	Write Enable	O	FDD	Enables write data into FDD.
26	MFM	MFM Mode	O	Phase Lock Loop	MFM mode when 1, FM mode when 0.
27	HD	Head Select	O	FDD	Head 1 selected when 1 (high), Head 0 selected when 0 (low).
28, 29	US_0, US_1	Unit Select	O	FDD	FDD Unit selected.
30	WDA	Write Data	O	FDD	Serial clock and data bits to FDD.
31, 32	PS_1, PS_0	Precompensation (preshift)	O	FDD	Write precompensation status during MFM mode. Determines early, late, and normal times.
33	FLT/TR_0	Fault/Track 0	I	FDD	Senses FDD fault condition in Read/Write mode; and Track 0 condition in Seek mode.
34	WP/TS	Write Protect/Two Side	I	FDD	Senses Write Protect status in Read/Write mode; and Two-Side Media in Seek mode.
35	RDY	Ready	I	FDD	Indicates FDD is ready to send or receive data.
36	HDL	Head Load	O	FDD	Command which causes read/write head in FDD to contact diskette.
37	FR/STP	Flt Reset/Step	O	FDD	Resets fault FF in FDD in Read/Write mode; contains step pulses to move head to another cylinder in Seek mode.
38	LCT/DIR	Low Current/Direction	O	FDD	Lowers Write current on inner tracks in Read/Write mode; determines direction head will step in Seek mode. A fault reset pulse is issued at the beginning of each Read or Write command prior to the occurrence of the Head Load signal.
39	$\overline{RW}$/SEEK	Read/Write Seek	O	FDD	When 1 (high) Seek mode selected and when 0 (low) Read/Write mode selected.
40	V_{CC}	+5V			DC power.

Note: ① Disabled when $\overline{CS}$ = 1

AC Characteristics

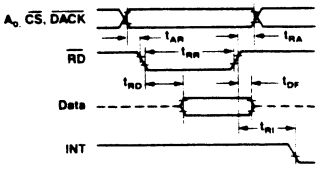
T_a = −10°C to +70°C;
V_{CC} = +5V ±5% unless otherwise specified

Parameter	Symbol	Limits Min	Typ	Max	Unit	Test Conditions
Clock Period	φCY	120	125	500	ns	8" FDD
			125			5¼" FDD
			250			3½" Sony
			125			
Clock Active (High)	φ0	40			ns	
Clock Rise Time	φr			20	ns	
Clock Fall Time	φf			20	ns	
A_0, CS, DACK Setup Time to RD ↓	t_{AR}	0			ns	
A_0, CS, DACK Hold Time from RD ↑	t_{RA}	0			ns	
RD Width	t_{RR}	250			ns	
Data Access Time from RD ↓	t_{RD}			200	ns	C_L = 100 pf
DB to Float Delay Time from RD ↑	t_{DF}	20		100	ns	C_L = 100 pF
A_0, CS, DACK Setup Time to WR ↓	t_{AW}	0			ns	
A_0, CS, DACK Hold Time to WR ↑	t_{WA}	0			ns	
WR Width	t_{WW}	250			ns	
Data Setup Time to WR ↑	t_{DW}	150			ns	
Data Hold Time from WR ↑	t_{WD}	5			ns	
INT Delay Time from RD ↓	t_{RI}			500	ns	
INT Delay Time from WR ↑	t_{WI}			500	ns	
DRQ Cycle Time	t_{MCY}	13			μs	
DACK ↓ → DRQ ↓ Delay	t_{AM}			200	ns	
DRQ ↑ → DACK ↓ Delay	t_{MA}	200			ns	ϕ_{CY} = 125 ns
DACK Width	t_{AA}	2			ϕ_{CY}	
TC Width	t_{TC}	1			φCY	
Reset Width	t_{RST}	14			φCY	
WCK Cycle Time	t_{CY}		4		μs	MFM = 0 5¼"
			2			MFM = 1 5¼"
			2			MFM = 0 8"
			1			MFM = 1 8"
			2			MFM = 0 3½" ③
			1			MFM = 1 3½" ③
WCK Active Time (High)	t_0	80	250	350	ns	
WCK Rise Time	t_r			20	ns	
WCK Fall Time	t_f			20	ns	
Preshift Delay Time from WCK ↑	t_{CP}	20		100	ns	
WCK ↑ → WE ↑ Delay	t_{CWE}	20		100	ns	
WDA Delay Time from WCK ↑	t_{CD}	20		100	ns	
RDD Active Time (High)	t_{RDD}	40			ns	
Window Cycle Time	t_{WCY}		4		μs	MFM = 0 5¼
			2			MFM = 1 5¼
			2			MFM = 0 8
			1			MFM = 1 8
			2			MFM = 0 3½
			1			MFM = 1 3½
Window Hold Time to/from RDD	t_{RDW} t_{WRD}	15			ns	
$US_{0,1}$ Hold Time to RW/SEEK ↑	t_{US}	12			μs	
RW/SEEK Hold Time to LOW CURRENT/DIRECTION ↑	t_{SD}	7			μs	
LOW CURRENT/DIRECTION Hold Time to FAULT RESET STEP ↑	t_{DST}	1.0			μs	
$US_{0,1}$ Hold Time from FAULT RESET/STEP ↑	t_{STU}	5.0			μs	8 MHz Clock Period ④
STEP Active Time (High)	t_{STP}	6	7	8	μs	④
STEP Cycle Time	t_{SC}	33	②	②	μs	④
FAULT RESET Active Time (High)	t_{FR}	8.0		10	μs	④
Write Data Width	t_{WDD}	T_0−50			ns	
$US_{0,1}$ Hold Time After SEEK	t_{SU}	15			μs	8 MHz Clock Period
Seek Hold Time from DIR	t_{DS}	30			μs	
DIR Hold Time after STEP	t_{STD}	24			μs	
Index Pulse Width	t_{IDX}	10			φCY	
RD ↓ Delay from DRQ	t_{MR}	800			μs	
WR ↓ Delay from DRQ	t_{MW}	250			μs	8 MHz Clock Period
WE or RD Response Time from DRQ ↑	t_{MRW}			12	μs	

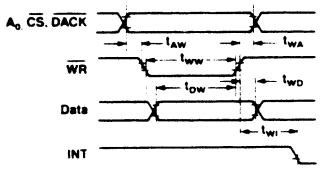
Notes: ① Typical values for T_a = 25°C and nominal supply voltage.
② Under Software Control. The range is from 1 ms to 16 ms at 8 MHz clock period, and 2 to 32 ms at 4 MHz clock period.
③ Sony Micro Floppydisk® 3½" drive.
④ Double these values for a 4 MHz clock period.

Timing Waveforms

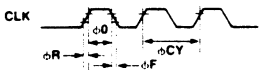
Processor Read Operation



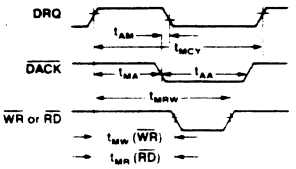
Processor Write Operation



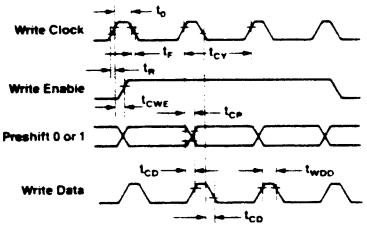
Clock



DMA Operation



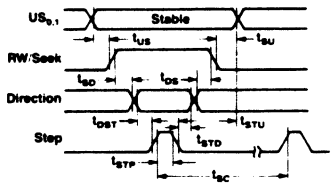
FDD Write Operation



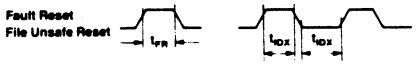
	Preshift 0	Preshift 1
Normal	0	0
Late	0	1
Early	1	0
Invalid	1	1

Timing Waveforms (Cont.)

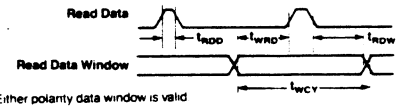
Seek Operation



FLT Reset *Index*



FDD Read Operation



Note: Either polarity data window is valid

Terminal Count *Reset*

Internal Registers

The μPD765A μPD7265 contains two registers which may be accessed by the main system processor: a Status Register and a Data Register. The 8-bit Main Status Register contains the status information of the FDC, and may be accessed at any time. The 8-bit Data Register (which actually consists of several registers in a stack with only one register presented to the data bus at a time), stores data, commands, parameters, and FDD status information. Data bytes are read out of, or written into, the Data Register in order to program or obtain the results after a particular command. Only the Status Register may be read and used to facilitate the transfer of data between the processor and μPD765 μPD7265.

The relationship between the Status Data registers and the signals $\overline{RD}$, $\overline{WR}$, and A_0 is shown below.

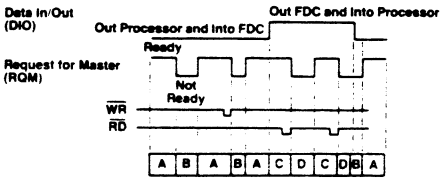
A_0	RD	WR	Function
0	0	1	Read Main Status Register
0	1	0	Illegal
0	0	0	Illegal
1	0	0	Illegal
1	0	1	Read from Data Register
1	1	0	Write into Data Register

Internal Registers (Cont.)

The bits in the Main Status Register are defined as follows:

Bit No.	Name	Symbol	Description
DB_0	FDD 0 Busy	D_0B	FDD number 0 is in the Seek mode. If any of the bits is set FDC will not accept read or write command.
DB_1	FDD 1 Busy	D_1B	FDD number 1 is in the Seek mode. If any of the bits is set FDC will not accept read or write command.
DB_2	FDD 2 Busy	D_2B	FDD number 2 is in the Seek mode. If any of the bits is set FDC will not accept read or write command.
DB_3	FDD 3 Busy	D_3B	FDD number 3 is in the Seek mode. If any of the bits is set FDC will not accept read or write command.
DB_4	FDC Busy	CB	A read or write command is in process. FDC will not accept any other command.
DB_5	Execution Mode	EXM	This bit is set only during execution phase in non-DMA mode. When DB_5 goes low, execution phase has ended and result phase has started. It operates only during non-DMA mode of operation.
DB_6	Data Input Output	DIO	Indicates direction of data transfer between FDC and Data Register. If DIO = 1, then transfer is from Data Register to the processor. If DIO = 0, then transfer is from the processor to Data Register.
DB_7	Request for Master	RQM	Indicates Data Register is ready to send or receive data to or from the processor. Both bits DIO and RQM should be used to perform the hand-shaking functions of "ready" and "direction" to the processor.

The DIO and RQM bits in the Status Register indicate when data is ready and in which direction data will be transferred on the data bus. The maximum time between the last $\overline{RD}$ or $\overline{WR}$ during a command or result phase and DIO and RQM getting set or reset is 12 μs. For this reason every time the Main Status Register is read the CPU should wait 12 μs. The maximum time from the trailing edge of the last $\overline{RD}$ in the result phase to when DB_4 (FDC busy) goes low is 12 μs.



Notes: A — Data register ready to be written into by processor
B — Data register not ready to be written into by processor
C — Data register ready for next data byte to be read by processor
D — Data register not ready to be read by processor

Status Register Identification

Bit No.	Name	Symbol	Description
			Status Register 0
D_7 D_6	Interrupt Code	IC	D_7 = 0 and D_6 = 0 Normal Termination of command, (NT). Command was completed and properly executed.
			D_7 = 0 and D_6 = 1 Abnormal Termination of command, (AT). Execution of command was started but was not successfully completed.
			D_7 = 1 and D_6 = 0 Invalid Command issue, (IC). Command which was issued was never started.
			D_7 = 1 and D_6 = 1 Abnormal Termination because during command execution the ready signal from FDD changed state.
D_5	Seek End	SE	When the FDC completes the SEEK command, this flag is set to 1 (high).
D_4	Equipment Check	EC	If a fault signal is received from the FDD, or if the Track 0 signal fails to occur after 77 step pulses (Recalibrate Command) then this flag is set.
D_3	Not Ready	NR	When the FDD is in the not-ready state and a read or write command is issued, this flag is set. If a read or write command is issued to Side 1 of a single-sided drive, then this flag is set.
D_2	Head Address	HD	This flag is used to indicate the state of the head at Interrupt.
D_1	Unit Select 1	US_1	These flags are used to indicate a Drive Unit Number at Interrupt.
D_0	Unit Select 0	US_0	
			Status Register 1
D_7	End of Cylinder	EN	When the FDC tries to access a sector beyond the final sector of a cylinder, this flag is set.
D_6			Not used. This bit is always 0 (low).
D_5	Data Error	DE	When the FDC detects a CRC① error in either the ID field or the data field, this flag is set.
D_4	Overrun	OR	If the FDC is not serviced by the host system during data transfers within a certain time interval, this flag is set.
D_3			Not used. This bit always 0 (low).
D_2	No Data	ND	During execution of READ DATA, WRITE DELETED DATA or SCAN command, if the FDC cannot find the sector specified in the IDR② Register, this flag is set.
			During execution of the READ ID command, if the FDC cannot read the ID field without an error, then this flag is set.
			During execution of the READ A cylinder command, if the starting sector cannot be found, then this flag is set.
D_1	Not Writable	NW	During execution of WRITE DATA, WRITE DELETED DATA or Format A cylinder command, if the FDC detects a write protect signal from the FDD, then this flag is set.
D_0	Missing Address Mark	MA	If the FDC cannot detect the ID Address Mark after encountering the index hole twice, then this flag is set.
			If the FDC cannot detect the Data Address Mark or Deleted Data Address Mark, this flag is set. Also at the same time, the MD (Missing Address Mark in data field) of Status Register 2 is set.
			Status Register 2
D_7			Not used. This bit is always 0 (low).
D_6	Control Mark	CM	During execution of the READ DATA or SCAN command, if the FDC encounters a sector which contains a Deleted Data Address Mark, this flag is set.
D_5	Data Error in Data Field	DD	If the FDC detects a CRC error in the data field then this flag is set.
D_4	Wrong Cylinder	WC	This bit is related to the ND bit, and when the contents of C③ on the medium is different from that stored in the IDR, this flag is set.
D_3	Scan Equal Hit	SH	During execution of the SCAN command, if the condition of "equal" is satisfied, this flag is set.
D_2	Scan Not Satisfied	SN	During execution of the SCAN command, if the FDC cannot find a sector on the cylinder which meets the condition, then this flag is set.
D_1	Bad Cylinder	BC	This bit is related to the ND bit, and when the contents of C on the medium is different from that stored in the IDR and the contents of C is $FF_{(16)}$, then this flag is set.
D_0	Missing Address Mark in Data Field	MD	When data is read from the medium, if the FDC cannot find a Data Address Mark or Deleted Data Address Mark, then this flag is set.

Status Register Identification (Cont.)

Bit No.	Name	Symbol	Description
			Status Register 3
D_7	Fault	FT	This bit is used to indicate the status of the Fault signal from the FDD.
D_6	Write Protected	WP	This bit is used to indicate the status of the Write Protected signal from the FDD.
D_5	Ready	RY	This bit is used to indicate the status of the Ready signal from the FDD.
D_4	Track 0	T0	This bit is used to indicate the status of the Track 0 signal from the FDD.
D_3	Two Side	TS	This bit is used to indicate the status of the Two Side signal from the FDD.
D_2	Head Address	HD	This bit is used to indicate the status of the Side Select signal to the FDD.
D_1	Unit Select 1	US_1	This bit is used to indicate the status of the Unit Select 1 signal to the FDD.
D_0	Unit Select 0	US_0	This bit is used to indicate the status of the Unit Select 0 signal to the FDD.

Notes: ① CRC = Cyclic Redundancy Check
② IDR = Internal Data Register
③ Cylinder (C) is described more fully in the Command Symbol Descripton on page 7

Command Sequence

The μPD765A/μPD7265 is capable of performing 15 different commands. Each command is initiated by a multibyte transfer from the processor, and the result after execution of the command may also be a multibyte transfer back to the processor. Because of this multibyte interchange of information between the μPD765A/μPD7265 and the processor, it is convenient to consider each command as consisting of three phases:

Command Phase:	The FDC receives all information required to perform a particular operation from the processor.
Execution Phase:	The FDC performs the operation it was instructed to do.
Result Phase:	After completion of the operation, status and other housekeeping information are made available to the processor.

Following are shown the required preset parameters and results for each command. Most commands require 9 command bytes and return 7 bytes during the result phase. The "W" to the left of each byte indicates a command phase byte to be written, and an "R" indicates a result byte.

Instruction Set ① ②

Phase	R/W	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
Read Data										
Command	W	MT	MF	SK	0	0	1	1	0	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			C				→	Sector ID information prior
	W	←			H				→	to command execution. The
	W	←			R				→	4 bytes are commanded
	W	←			N				→	against header on Floppy
	W	←			EOT				→	Disk.
	W	←			GPL				→	
	W	←			DTL				→	
Execution										Data transfer between the FDD and main system
Result	R	←			ST 0				→	Status information after
	R	←			ST 1				→	command execution
	R	←			ST 2				→	
	R	←			C				→	Sector ID information after
	R	←			H				→	command execution
	R	←			R				→	
	R	←			N				→	
Read Deleted Data										
Command	W	MT	MF	SK	0	1	1	0	0	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			C				→	Sector ID information prior
	W	←			H				→	to command execution. The
	W	←			R				→	4 bytes are commanded
	W	←			N				→	against header on Floppy
	W	←			EOT				→	Disk.
	W	←			GPL				→	
	W	←			DTL				→	
Execution										Data transfer between the FDD and main system
Result	R	←			ST 0				→	Status information after
	R	←			ST 1				→	command execution
	R	←			ST 2				→	
	R	←			C				→	Sector ID information after
	R	←			H				→	command execution
	R	←			R				→	
	R	←			N				→	
Write Data										
Command	W	MT	MF	0	0	0	1	0	1	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			C				→	Sector ID information prior
	W	←			H				→	to command execution. The
	W	←			R				→	4 bytes are commanded
	W	←			N				→	against header on Floppy
	W	←			EOT				→	Disk.
	W	←			GPL				→	
	W	←			DTL				→	
Execution										Data transfer between the main system and FDD
Result	R	←			ST 0				→	Status information after
	R	←			ST 1				→	command execution
	R	←			ST 2				→	
	R	←			C				→	Sector ID information after
	R	←			H				→	command execution
	R	←			R				→	
	R	←			N				→	

Notes: ① Symbols used in this table are described at the end of this section
② A_0 should equal binary 1 for all operations
③ X = Don't care, usually made to equal binary 0

Instruction Set

Phase	R/W	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
Write Deleted Data										
Command	W	MT	MF	0	0	1	0	0	1	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			C				→	Sector ID information prior
	W	←			H				→	to command execution. The
	W	←			R				→	4 bytes are commanded
	W	←			N				→	against header on Floppy
	W	←			EOT				→	Disk.
	W	←			GPL				→	
	W	←			DTL				→	
Execution										Data transfer between the FDD and main system
Result	R	←			ST 0				→	Status information after
	R	←			ST 1				→	command execution
	R	←			ST 2				→	
	R	←			C				→	Sector ID information after
	R	←			H				→	command execution
	R	←			R				→	
	R	←			N				→	
Read A Track										
Command	W	0	MF	SK	0	0	0	1	0	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			C				→	Sector ID information prior
	W	←			H				→	to command execution
	W	←			R				→	
	W	←			N				→	
	W	←			EOT				→	
	W	←			GPL				→	
	W	←			DTL				→	
Execution										Data transfer between the FDD and main system. FDC reads all data fields from index hole to EOT.
Result	R	←			ST 0				→	Status information after
	R	←			ST 1				→	command execution
	R	←			ST 2				→	
	R	←			C				→	Sector ID information after
	R	←			H				→	command execution
	R	←			R				→	
	R	←			N				→	
Read ID										
Command	W	0	MF	0	0	1	0	1	0	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
Execution										The first correct ID information on the cylinder is stored in Data Register.
Result	R	←			ST 0				→	Status information after
	R	←			ST 1				→	command execution
	R	←			ST 2				→	
	R	←			C				→	Sector ID information read
	R	←			H				→	during Execution phase from
	R	←			R				→	Floppy Disk.
	R	←			N				→	
Format A Track										
Command	W	0	MF	0	0	1	1	0	1	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			N				→	Bytes Sector
	W	←			SC				→	Sectors Track
	W	←			GPL				→	Gap 3
	W	←			D				→	Filler byte
Execution										FDC formats an entire track.
Result	R	←			ST 0				→	Status information after
	R	←			ST 1				→	command execution
	R	←			ST 2				→	
	R	←			C				→	In this case, the ID
	R	←			H				→	information has no meaning.
	R	←			R				→	
	R	←			N				→	
Scan Equal										
Command	W	MT	MF	SK	1	0	0	0	1	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			C				→	Sector ID information prior
	W	←			H				→	to command execution
	W	←			R				→	
	W	←			N				→	
	W	←			EOT				→	
	W	←			GPL				→	
	W	←			STP				→	
Execution										Data compared between the FDD and main system
Result	R	←			ST 0				→	Status information after
	R	←			ST 1				→	command execution
	R	←			ST 2				→	
	R	←			C				→	Sector ID information after
	R	←			H				→	command execution
	R	←			R				→	
	R	←			N				→	

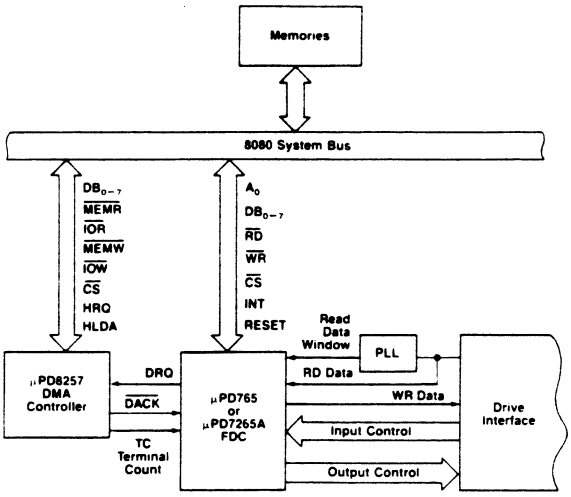
Instruction Set (Cont.)

Phase	R/W	Data Bus D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Remarks
Scan Low or Equal										
Command	W	MT	MF	SK	1	1	0	0	1	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			C				→	Sector ID information prior to command execution
	W	←			H				→	
	W	←			R				→	
	W	←			N				→	
	W	←			EOT				→	
	W	←			GPL				→	
	W	←			STP				→	
Execution										Data compared between the FDD and main system
Result	R	←			ST 0				→	Status information after command execution
	R	←			ST 1				→	
	R	←			ST 2				→	
	R	←			C				→	Sector ID information after command execution
	R	←			H				→	
	R	←			R				→	
	R	←			N				→	
Scan High or Equal										
Command	W	MT	MF	SK	1	1	1	0	1	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			C				→	Sector ID information prior to command execution
	W	←			H				→	
	W	←			R				→	
	W	←			N				→	
	W	←			EOT				→	
	W	←			GPL				→	
	W	←			STP				→	
Execution										Data compared between the FDD and main system
Result	R	←			ST 0				→	Status information after command execution
	R	←			ST 1				→	
	R	←			ST 2				→	
	R	←			C				→	Sector ID information after command execution
	R	←			H				→	
	R	←			R				→	
	R	←			N				→	
Recalibrate										
Command	W	0	0	0	0	0	1	1	1	Command Codes
	W	X	X	X	X	X	0	US_1	US_0	
Execution										Head retracted to Track 0
Sense Interrupt Status										
Command	W	0	0	0	0	1	0	0	0	Command Codes
Result	R	←			STO				→	Status information about the FDC at the end of seek operation
	R	←			PCN				→	
Specify										
Command	W	0	0	0	0	0	0	1	1	Command Codes
	W	← SRT			→	←		HUT →		
	W	←	HLT			→		ND		
Sense Drive Status										
Command	W	0	0	0	0	0	1	0	0	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
Result	R	←			ST 3				→	Status information about FDD
Seek										
Command	W	0	0	0	0	1	1	1	1	Command Codes
	W	X	X	X	X	X	HD	US_1	US_0	
	W	←			NCN				→	
Execution										Head is positioned over proper cylinder on diskette.
Invalid										
Command	W	←			Invalid Codes				→	Invalid Command Codes (NoOp — FDC goes into Standby state.)
Result	R	←			ST 0				→	ST 0 = $80_{(16)}$

Command Symbol Description

Symbol	Name	Description
A_0	Address Line 0	A_0 controls selection of Main Status Register (A_0 = 0) or Data Register (A_0 = 1)
C	Cylinder Number	C stands for the current/selected cylinder (track) numbers 0 through 76 of the medium.
D	Data	D stands for the data pattern which is going to be written into a sector.
D_7–D_0	Data Bus	8-bit Data Bus, where D_7 stands for a most significant bit, and D_0 stands for a least significant bit.
DTL	Data Length	When N is defined as 00, DTL stands for the data length which users are going to read out or write into the sector.
EOT	End of Track	EOT stands for the final sector number on a cylinder. During Read or Write operations, FDC will stop data transfer after a sector number equal to EOT.
GPL	Gap Length	GPL stands for the length of Gap 3. During Read/Write commands this value determines the number of bytes that VCOs will stay low after two CRC bytes. During Format command it determines the size of Gap 3.
H	Head Address	H stands for head number 0 or 1, as specified in ID field.
HD	Head	HD stands for a selected head number 0 or 1 and controls the polarity of pin 27. (H = HD in all command words.)
HLT	Head Load Time	HLT stands for the head load time in the FDD (2 to 254 ms in 2 ms increments).
HUT	Head Unload Time	HUT stands for the head unload time after a Read or Write operation has occurred (16 to 240 ms in 16 ms increments).
MF	FM or MFM Mode	If MF is low, FM mode is selected, and if it is high, MFM mode is selected.
MT	Multitrack	If MT is high, a Multitrack operation is performed. If MT = 1 after finishing Read/Write operation on side 0, FDC will automatically start searching for sector 1 on side 1.
N	Number	N stands for the Number of data bytes written in a sector.
NCN	New Cylinder Number	NCN stands for a New Cylinder Number which is going to be reached as a result of the Seek operation. Desired position of head.
ND	Non-DMA Mode	ND stands for operation in the Non-DMA mode.
PCN	Present Cylinder Number	PCN stands for the cylinder number at the completion of Sense Interrupt Status command. Position of Head at present time.
R	Record	R stands for the sector number which will be read or written.
R/W	Read/Write	R/W stands for either Read (R) or Write (W) signal.
SC	Sector	SC indicates the number of Sectors per Cylinder.
SK	Skip	SK stands for Skip Deleted Data Address mark.
SRT	Step Rate Time	SRT stands for the Stepping Rate for the FDD (1 to 16 ms in 1 ms increments). Stepping Rate applies to all drives (F = 1 ms, E = 2 ms, etc.).
ST0 ST1 ST2 ST3	Status 0 Status 1 Status 2 Status 3	ST 0–3 stands for one of four registers which store the status information after a command has been executed. This information is available during the result phase after command execution. These registers should not be confused with the main status register (selected by A_0 = 0). ST 0–3 may be read only after a command has been executed and contains information relevant to that particular command.
STP		During a Scan operation, if STP = 1, the data in contiguous sectors is compared byte by byte with data sent from the processor (or DMA); and if STP = 2, then alternate sectors are read and compared.
US0, US1	Unit Select	US stands for a selected drive number 0 or 1.

System Configuration



Processor Interface

During Command or Result phases the Main Status Register (described earlier) must be read by the processor before each byte of information is written into or read from the Data Register. After each byte of data read or written to the Data Register, CPU should wait for 12μs before reading Main Status Register. Bits D6 and D7 in the Main Status Register must be in a 0 and 1 state, respectively, before each byte of the command word may be written into the μPD765A μPD7265. Many of the commands require multiple bytes and, as a result, the Main Status Register must be read prior to each byte transfer to the μPD765A μPD7265. On the other hand, during the Result phase, D6 and D7 in the Main Status Register must both be 1's (D6 = 1 and D7 = 1) before reading each byte from the Data Register. Note that this reading of the Main Status Register before each byte transfer to the μPD765A/μPD7265 is required only in the Command and Result phases, and *not* during the Execution phase.

During the Execution phase, the Main Status Register need not be read. If the μPD765A/μPD7265 is in the non-DMA mode, then the receipt of each data byte (if μPD765A μPD7265 is reading data from FDD) is indicated by an Interrupt signal on pin 18 (INT = 1). The generation of a Read signal ($\overline{RD}$ = 0) or Write signal ($\overline{WR}$ = 0) will clear the Interrupt as well as output the data onto the data bus. If the processor cannot handle Interrupts fast enough (every 13 μs for the MFM mode and 27 μs for the FM mode), then it may poll the Main Status Register and bit D7 (RQM) functions as the Interrupt signal. If a Write command is in process then the $\overline{WR}$ signal negates the reset to the Interrupt signal.

Note that in the non-DMA mode it is necessary to examine the Main Status Register to determine the cause of the interrupt, since it could be a data interrupt or a command termination interrupt, either normal or abnormal.

If the μPD765A μPD7265 is in the DMA mode, no Interrupts are generated during the Execution phase. The μPD765A/μPD7265 generates DRQs (DMA Requests) when each byte of data is available. The DMA Controller responds to this request with both a $\overline{DACK}$ = 0 (DMA Acknowledge) and an $\overline{RD}$ = 0 (Read signal). When the DMA Acknowledge signal goes low ($\overline{DACK}$ = 0), then the DMA Request is cleared (DRQ = 0). If a Write command has been issued then a $\overline{WR}$ signal will appear instead of $\overline{RD}$. After the Execution phase has been completed (Terminal Count has occurred) or the EOT sector read/written, then an Interrupt will occur (INT = 1). This signifies the beginning of the Result phase. When the first byte of data is read during the Result phase, the Interrupt is automatically cleared (INT = 0).

The $\overline{RD}$ or $\overline{WR}$ signals should be asserted while $\overline{DACK}$ is true. The $\overline{CS}$ signal is used in conjunction with $\overline{RD}$ and $\overline{WR}$ as a gating function during programmed I/O operations. $\overline{CS}$ has no effect during DMA operations. If the non-DMA mode is chosen, the $\overline{DACK}$ signal should be pulled up to V_{CC}.

It is important to note that during the Result phase all bytes shown in the Command Table must be read. The Read Data command, for example, has seven bytes of data in the Result phase. All seven bytes must be read in order to successfully complete the Read Data command. The μPD765A μPD7265 will not accept a new command until all seven bytes have been read. Other commands may require fewer bytes to be read during the Result phase.

The μPD765A μPD7265 contains five Status Registers. The Main Status Register mentioned above may be read by the processor at any time. The other four Status Registers (ST0, ST1, ST2, and ST3) are available only during the Result phase and may be read only after completing a command. The particular command that has been executed determines how many of the Status Registers will be read.

The bytes of data which are sent to the μPD765A μPD7265 to form the Command phase and are read out of the μPD765A μPD7265 in the Result phase must occur in the order shown in the Command Table. That is, the Command Code must be sent first and the other bytes sent in the prescribed sequence. No foreshortening of the Command or Result phases is allowed. After the last byte of data in the Command phase is sent to the μPD765A μPD7265, the Execution phase automatically starts. In a similar fashion, when the last byte of data is read out in the Result phase, the command is automatically ended and the μPD765A μPD7265 is ready for a new command.

Polling Feature of the μPD765A/μPD7265

After Reset has been sent to the μPD765A μPD7265, the Unit Select lines US_0 and US_1 will automatically go into a polling mode. In between commands (and between step pulses in the Seek command) the μPD765A μPD7265 polls all four FDDs looking for a change in the Ready line from any of the drives. If the Ready line changes state (usually due to a door opening or closing), then the μPD765A μPD7265 will generate an interrupt. When Status Register 0 (ST0) is read (after Sense Interrupt Status is issued), Not Ready (NR) will be indicated. The polling of the Ready line by the μPD765A/μPD7265 occurs continuously between commands, thus notifying the processor which drives are on or off line. Each drive is polled every 1.024 ms except during the Read/Write commands. When used with a 4 MHz clock for interfacing to minifloppies, the polling rate is 2.048 ms.

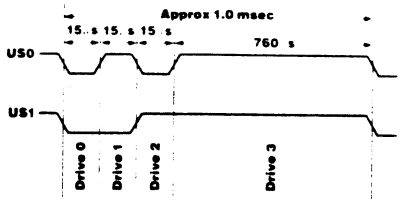


figure 1. (polling feature)

Read Data

A set of nine (9) byte words are required to place the FDC into the Read Data Mode. After the Read Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the Specify Command), and begins reading ID Address Marks and ID fields. When the current sector number ("R") stored in the ID Register (IDR) compares with the sector number read off the diskette, then the FDC outputs data (from the data field) byte-to-byte to the main system via the data bus.

After completion of the read operation from the current sector, the Sector Number is incremented by one, and the data from the next sector is read and output on the data bus. This continuous read function is called a "Multi-Sector Read Operation." The Read Data Command may be terminated by the receipt of a Terminal Count signal. TC should be issued at the same time that the $\overline{DACK}$ for the last byte of data is sent. Upon receipt of this signal, the FDC stops outputting data to the processor, but will continue to read data from the current sector, check CRC (Cyclic Redundancy Count) bytes, and then at the end of the sector terminate the Read Data Command. The amount of data which can be handled with a single command to the FDC depends upon MT (multitrack), MF (MFM FM), and N (Number of Bytes Sector). Table 1 below shows the Transfer Capacity.

Transfer Capacity

Multi-Track MT	MFM/FM MF	Bytes/Sector N	Maximum Transfer Capacity (Bytes/Sector) (Number of Sectors)	Final Sector Read from Diskettes
0	0	00	(128) (26) = 3,328	26 at Side 0
0	1	01	(256) (26) = 6,656	or 26 at Side 1
1	0	00	(128) (52) = 6,656	26 at Side 1
1	1	01	(256) (52) = 13,312	
0	0	01	(256) (15) = 3,840	15 at Side 0
0	1	02	(512) (15) = 7,680	or 15 at Side 1
1	0	01	(256) (30) = 7,680	15 at Side 1
1	1	02	(512) (30) = 15,360	
0	0	02	(512) (8) = 4,096	8 at Side 0
0	1	03	(1024) (8) = 8,192	or 8 at Side 1
1	0	02	(512) (16) = 8,192	8 at Side 1
1	1	03	(1024) (16) = 16,384	

The "multi-track" function (MT) allows the FDC to read data from both sides of the diskette. For a particular cylinder, data will be transferred starting at Sector 1, Side 0 and completing at Sector L, Side 1 (Sector L = last sector on the side). Note, this function pertains to only one cylinder (the same track) on each side of the diskette.

When N = 0, then DTL defines the data length which the FDC must treat as a sector. If DTL is smaller than the actual data length in a Sector, the data beyond DTL in the Sector, is not sent to the Data Bus. The FDC reads (internally) the complete Sector performing the CRC check, and depending upon the manner of command termination, may perform a Multi-Sector Read Operation. When N is non-zero, then DTL has no meaning and should be set to FF Hexidecimal.

At the completion of the Read Data Command, the head is not unloaded until after Head Unload Time Interval (specified in the Specify Command) has elapsed. If the processor issues another command before the head unloads then the head settling time may be saved between subsequent reads. This time out is particularly valuable when a diskette is copied from one drive to another.

If the FDC detects the Index Hole twice without finding the right sector, (indicated in "R"), then the FDC sets the ND (No Data) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

After reading the ID and Data Fields in each sector, the FDC checks the CRC bytes. If a read error is detected (incorrect CRC in ID field), the FDC sets the DE (Data Error) flag in Status Register 1 to a 1 (high), and if a CRC error occurs in the Data Field the FDC also sets the DD (Data Error in Data Field) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

If the FDC reads a Deleted Data Address Mark off the diskette, and the SK bit (bit D5 in the first Command Word) is not set (SK = 0), then the FDC sets the CM (Control Mark) flag in Status Register 2 to a 1 (high), and terminates the Read Data Command, after reading all the data in the Sector. If SK = 1, the FDC skips the sector with the Deleted Data Address Mark and reads the next sector. The CRC bits in the deleted data field are not checked when SK = 1.

During disk data transfers between the FDC and the processor, via the data bus, the FDC must be serviced by the processor every 27 μs in the FM Mode, and every 13 μs in the MFM Mode, or the FDC sets the OR (Overrun) flag in Status Register 1 to a 1 (high), and terminates the Read Data Command.

If the processor terminates a read (or write) operation in the FDC, then the ID information in the Result Phase is dependent upon the state of the MT bit and EOT byte. Table 2 shows the values for C, H, R, and N, when the processor terminates the Command.

Functional Description of Commands

MT	ND	Final Sector Transferred to Processor	ID Information at Result Phase C	H	R	N
0	0	Less than EOT	NC	NC	R+1	NC
	0	Equal to EOT	C+1	NC	R = 01	NC
	1	Less than EOT	NC	NC	R+1	NC
	1	Equal to EOT	C+1	NC	R = 01	NC
1	0	Less than EOT	NC	NC	R+1	NC
	0	Equal to EOT	NC	LSB	R = 01	NC
	1	Less than EOT	NC	NC	R+1	NC
	1	Equal to EOT	C+1	LSB	R = 01	NC

Notes: NC (No Change): The same value as the one at the beginning of command execution.
LSB (Least Significant Bit): The least significant bit of H is complemented.

Write Data

A set of nine (9) bytes is required to set the FDC into the Write Data mode. After the Write Data command has been issued the FDC loads the head (if it is in the unloaded state), waits the specified head settling time (defined in the Specify command), and begins reading ID fields. When all four bytes loaded during the command (C, H, R, N) match

the four bytes of the ID field from the diskette, the FDC takes data from the processor byte-by-byte via the data bus and outputs it to the FDD.

After writing data into the current sector, the sector number stored in "R" is incremented by one, and the next data field is written into. The FDC continues this "Multisector Write Operation" until the issuance of a Terminal Count signal. If a Terminal Count signal is sent to the FDC it continues writing into the current sector to complete the data field. If the Terminal Count signal is received while a data field is being written then the remainder of the data field is filled with zeros.

The FDC reads the ID field of each sector and checks the CRC bytes. If the FDC detects a read error (CRC error) in one of the ID fields, it sets the DE (Data Error) flag of Status Register 1 to a 1 (high) and terminates the Write Data command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

The Write command operates in much the same manner as the Read command. The following items are the same, and one should refer to the Read Data command for details:

- Transfer Capacity
- EN (End of Cylinder) Flag
- ND (No Data) Flag
- Head Unload Time Interval
- ID Information when the processor terminates command
- Definition of DTL when N = 0 and when N ≠ 0

In the Write Data mode, data transfers between the processor and FDC, via the data bus, must occur every 27 μs in the FM mode and every 13 μs in the MFM mode. If the time interval between data transfers is longer than this, then the FDC sets the OR (Overrun) flag in Status Register 1 to a 1 (high) and terminates the Write Data command. (Status Register 0 also has bits 7 and 6 set to 0 and 1 respectively.)

Write Deleted Data

This command is the same as the Write Data command except a Deleted Data Address mark is written at the beginning of the data field instead of the normal Data Address mark.

Read Deleted Data

This command is the same as the Read Data command except that when the FDC detects a Data Address mark at the beginning of a data field (and SK = 0 (low)), it will read all the data in the sector and set the CM flag in Status Register 2 to a 1 (high), and then terminate the command. If SK = 1, then the FDC skips the sector with the Data Address mark and reads the next sector.

Read A Track

This command is similar to the Read Data command except that this is a continuous Read operation where the entire data field from each of the sectors is read. Immediately after sensing the index hole, the FDC starts reading all data fields on the track as continuous blocks of data. If the FDC finds an error in the ID or Data CRC check bytes, it continues to read data from the track. The FDC compares the ID information read from each sector with the value stored in the IDR and sets the ND flag of Status Register 1 to a 1 (high) if there is no comparison. Multitrack or skip operations are not allowed with this command.

This command terminates when the number of sectors read is equal to EOT. If the FDC does not find an ID Address mark on the diskette after it senses the index hole for the second time, it sets the MA (Missing Address mark) flag in Status Register 1 to a 1 (high) and terminates the command. (Status Register 0 has bits 7 and 6 set to 0 and 1 respectively.)

Read ID

The Read ID command is used to give the present position of the recording head. The FDC stores the values from the first ID field it is able to read. If no proper ID Address mark is found on the diskette before the index hole is encountered for the second time, then the MA (Missing Address mark) flag in Status Register 1 is set to a 1 (high), and if no data is found then the ND (No Data) flag is also set in Status Register 1 to a 1 (high). The command is then terminated with bits 7 and 6 in Status Register 0 set to 0 and 1 respectively. During this command there is no data transfer between FDC and the CPU except during the result phase.

Format A Track

The Format command allows an entire track to be formatted. After the index hole is detected, data is written on the diskette: Gaps, Address marks, ID fields and data fields, all per the IBM System 34 (Double Density) or System 3740 (Single Density) format are recorded. The particular format which will be written is controlled by the values programmed into N (Number of bytes sector), SC (Sectors Cylinder), GPL (Gap Length), and D (Data pattern) which are supplied by the processor during the Command phase. The data field is filled with the byte of data stored in D. The ID field for each sector is supplied by the processor; that is, four data requests per sector are made by the FDC for C (Cylinder number), H (Head number), R (Sector number) and N (Number of bytes sector). This allows the diskette to be formatted with nonsequential sector numbers, if desired.

The processor must send new values for C, H, R, and N to the μPD765A μPD7265 for each sector on the track. If FDC is set for the DMA mode, it will issue four DMA requests per sector. If it is set for the Interrupt mode, it will issue four interrupts per sector and the processor must supply C, H, R, and N loads for each sector. The contents of the R register are incremented by 1 after each sector is formatted; thus, the R register contains a value of R when it is read during the Result phase. This incrementing and formatting continues for the whole track until the FDC detects the index hole for the second time, whereupon it terminates the command.

If a Fault signal is received from the FDD at the end of a Write operation, then the FDC sets the EC flag of Status Register 0 to a 1 (high) and terminates the command after setting bits 7 and 6 of Status Register 0 to 0 and 1 respectively. Also the loss of a Ready signal at the beginning of a command execution phase causes bits 7 and 6 of Status Register 0 to be set to 0 and 1 respectively.

Table 1 shows the relationship between N, SC, and GPL for various sector sizes.

Functional Description of Commands (Cont.)

Format	Sector Size	N	SC	GPL ①	GPL ②
		8" Standard Floppy			
FM Mode	128 bytes/sector	00	1A	07	1B
	256	01	0F	0E	2A
	512	02	08	1B	3A
	1024	03	04	47	8A
	2048	04	02	C8	FF
	4096	05	01	C8	FF
MFM Mode ④	256	01	1A	0E	36
	512	02	0F	1B	54
	1024	03	08	35	74
	2048	04	04	99	FF
	4096	05	02	C8	FF
	8192	06	01	C8	FF
		5¼" Minifloppy			
FM Mode	128 bytes/sector	00	12	07	09
	128	00	10	10	19
	256	01	08	18	30
	512	02	04	46	87
	1024	03	02	C8	FF
	2048	04	01	C8	FF
MFM Mode ④	256	01	12	0A	0C
	256	01	10	20	32
	512	02	08	2A	50
	1024	03	04	80	F0
	2048	04	02	C8	FF
	4096	05	01	C8	FF
		3½" Sony Microfloppy			
FM Mode	128 bytes/sector	0	0F	07	1B
	256	1	09	0E	2A
	512	2	05	1B	3A
MFM Mode ④	256	1	0F	0E	36
	512	2	09	1B	54
	1024	3	05	35	74

Table 1

Notes: ① Suggested values of GPL in Read or Write commands to avoid splice point between data field and ID field of contiguous sections.
② Suggested values of GPL in format command.
③ All values except sector size are hexidecimal.
④ In MFM mode FDC cannot perform a Read Write format operation with 128 bytes sector. (N = 00)

Scan Commands

The Scan commands allow data which is being read from the diskette to be compared against data which is being supplied from the main system. The FDC compares the data on a byte-by-byte basis and looks for a sector of data which meets the conditions of $D_{FDD} = D_{Processor}$, $D_{FDD} \leq D_{Processor}$, or $D_{FDD} \geq D_{Processor}$. The hexidecimal byte of FF either from memory or from FDD can be used as a mask byte because it always meets the condition of the comparison. One's complement arithmetic is used for comparison (FF = largest number, 00 = smallest number). After a whole sector of data is compared, if the conditions are not met, the sector number is incremented (R + STP → R), and the scan operation is continued. The scan operation continues until one of the following conditions occur: the conditions for scan are met (equal, low, or high), the last sector on the track is reached (EOT), or the terminal count signal is received.

If the conditions for scan are met, then the FDC sets the SH (Scan Hit) flag of Status Register 2 to a 1 (high) and terminates the Scan command. If the conditions for scan are not met between the starting sector (as specified by R) and the last sector on the cylinder (EOT), then the FDC sets the SN (Scan Not Satisfied) flag of Status Register 2 to a 1 (high) and terminates the Scan command. The receipt of a Terminal Count signal from the processor or DMA controller during the scan operation will cause the FDC to complete the comparison of the particular byte which is in process and then to terminate the command. Table 2 shows the status of bits SH and SN under various conditions of Scan.

Command	Status Register 2 Bit 2 = SN	Status Register 2 Bit 3 = SH	Comments
Scan Equal	0	1	$D_{FDD} = D_{Processor}$
	1	0	$D_{FDD} \neq D_{Processor}$
Scan Low or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} < D_{Processor}$
	1	0	$D_{FDD} > D_{Processor}$
Scan High or Equal	0	1	$D_{FDD} = D_{Processor}$
	0	0	$D_{FDD} > D_{Processor}$
	1	0	$D_{FDD} < D_{Processor}$

Table 2

If the FDC encounters a Deleted Data Address mark on one of the sectors (and SK = 0), then it regards the sector as the last sector on the cylinder, sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) and terminates the command. If SK = 1, the FDC skips the sector with the Deleted Address mark and reads the next sector. In the second case (SK = 1), the FDC sets the CM (Control Mark) flag of Status Register 2 to a 1 (high) in order to show that a Deleted sector had been encountered.

When either the STP (contiguous sectors = 01, or alternate sectors = 02) sectors are read or the MT (Multitrack) is programmed, it is necessary to remember that the last sector on the track must be read. For example, if STP = 02, MT = 0, the sectors are numbered sequentially 1 through 26 and the Scan command is started at sector 21, the following will happen: Sectors 21, 23, and 25 will be read, then the next sector (26) will be skipped and the index hole will be encountered before the EOT value of 26 can be read. This will result in an abnormal termination of the command. If the EOT had been set at 25 or the scanning started at sector 20, then the Scan command would be completed in a normal manner.

During the Scan command, data is supplied by either the processor or DMA Controller for comparison against the data read from the diskette. In order to avoid having the OR (Overrun) flag set in Status Register 1, it is necessary to have the data available in less than 27 μs (FM mode) or 13 μs (MFM mode). If an Overrun occurs, the FDC ends the command with bits 7 and 6 of Status Register 0 set to 0 and 1, respectively.

Seek

The Read/Write head within the FDD is moved from cylinder to cylinder under control of the Seek command. FDC has four independent Present Cylinder Registers for each drive. They are cleared only after the Recalibrate command. The FDC compares the PCN (Present Cylinder Number) which is the current head position with the NCN (New Cylinder Number), and if there is a difference, performs the following operations:

PCN < NCN: Direction signal to FDD set to a 1 (high), and Step Pulses are issued. (Step In)

PCN > NCN: Direction signal to FDD set to a 0 (low), and Step Pulses are issued. (Step Out)

The rate at which Step pulses are issued is controlled by SRT (Stepping Rate Time) in the Specify command. After each Step pulse is issued NCN is compared against PCN, and when NCN = PCN, the SE (Seek End) flag is set in Status Register 0 to a 1 (high), and the command is terminated. At this point FDC interrupt goes high. Bits D_0B-D_3B in the Main Status Register are set during the Seek operation and are cleared by the Sense Interrupt Status command.

)uring the command phase of the Seek operation the 'DC is in the FDC Busy state, but during the execution hase it is in the Nonbusy state. While the FDC is in the lonbusy state, another Seek command may be issued, nd in this manner parallel Seek operations may be done n up to four drives at once. No other command can be ;sued for as long as the FDC is in the process of sending tep pulses to any drive.

an FDD is in a Not Ready state at the beginning of the ommand execution phase or during the Seek operation, ıen the NR (Not Ready) flag is set in Status Register 0 to a (high), and the command is terminated after bits 7 and 6 f Status Register 0 are set to 0 and 1 respectively.

the time to write three bytes of Seek command exceeds 50μs, the timing between the first two step pulses may e shorter than set in the Specify command by as much s 1ms.

lecalibrate

he function of this command is to retract the Read/Write ead within the FDD to the Track 0 position. The FDC lears the contents of the PCN counter and checks the tatus of the Track 0 signal from the FDD. As long as the rack 0 signal is low, the Direction signal remains 0 (low) nd step pulses are issued. When the Track 0 signal goes igh, the SE (Seek End) flag in Status Register 0 is set to 1 (high) and the command is terminated. If the Track 0 ignal is still low after 77 step pulses have been issued, the DC sets the SE (Seek End) and EC (Equipment Check) ags of Status Register 0 to both 1s (highs) and terminates ıe command after bits 7 and 6 of Status Register 0 are set ı 0 and 1 respectively.

he ability to do overlap Recalibrate commands to multiple DDs and the loss of the Ready signal, as described in the eek command, also applies to the Recalibrate command. the Diskette has more than 77 tracks, then Recalibrate ɔmmand should be issued twice, in order to position the ead/Write head to the Track 0.

Sense Interrupt Status

An Interrupt signal is generated by the FDC for one of the following reasons:

1. Upon entering the Result phase of:
   a. Read Data command
   b. Read A Track command
   c. Read ID command
   d. Read Deleted Data command
   e. Write Data command
   f. Format A Cylinder command
   g. Write Deleted Data command
   h. Scan commands
2. Ready Line of FDD changes state
3. End of Seek or Recalibrate command
4. During Execution phase in the non-DMA mode

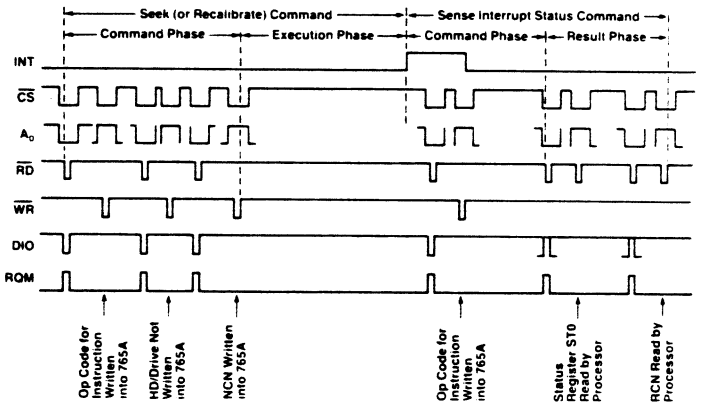
Interrupts caused by reasons 1 and 4 above occur during normal command operations and are easily discernible by the processor. During an execution phase in non-DMA mode, DB5 in the Main Status Register is high. Upon entering the Result phase this bit gets cleared. Reasons 1 and 4 do not require Sense Interrupt Status commands. The interrupt is cleared by Reading/Writing data to the FDC. Interrupts caused by reasons 2 and 3 above may be uniquely identified with the aid of the Sense Interrupt Status command. This command when issued resets the Interrupt signal and via bits 5, 6, and 7 of Status Register 0 identifies the cause of the interrupt.

Seek End	Interrupt Code		
Bit 5	Bit 6	Bit 7	Cause
0	1	1	Ready Line changed state, either polarity
1	0	0	Normal Termination of Seek or Recalibrate command
1	1	0	Abnormal Termination of Seek or Recalibrate command

Table 3

The Sense Interrupt Status command is used in conjunction with the Seek and Recalibrate commands which have no result phase. When the disk drive has reached the desired head position the μPD765A μPD7265 will set the Interrupt line true. The host CPU must then issue a Sense Interrupt Status command to determine the actual cause of the interrupt, which could be Seek End or a change in ready status from one of the drives. A graphic example is shown:

Seek, Recalibrate, and Sense Interrupt Status



Specify

The Specify command sets the initial values for each of the three internal timers. The HUT (Head Unload Time) defines the time from the end of the execution phase of one of the Read/Write commands to the head unload state. This timer is programmable from 16 to 240ms in increments of 16ms (01 = 16ms, 02 = 32ms . . . $0F_{16}$ = 240ms). The SRT (Step Rate Time) defines the time interval between adjacent step pulses. This timer is programmable from 1 to 16 ms in increments of 1 ms (F = 1ms, E = 2ms, D = 3ms, etc.). The HLT (Head Load Time) defines the time between when the Head Load signal goes high and the Read/Write operation starts. This timer is programmable from 2 to 254 ms in increments of 2 ms (01 = 2ms, 02 = 4ms, 03 = 6ms . . . 7F = 254ms).

The time intervals mentioned above are a direct function of the clock (CLK on pin 19). Times indicated above are for an 8MHz clock; if the clock was reduced to 4MHz (minifloppy application) then all time intervals are increased by a factor of 2.

The choice of a DMA or non-DMA operation is made by the ND (Non-DMA) bit. When this bit is high (ND = 1) the Non-DMA mode is selected, and when ND = 0 the DMA mode is selected.

Sense Drive Status

This command may be used by the processor whenever it wishes to obtain the status of the FDDs. Status Register 3 contains the Drive Status information stored internally in FDC registers.

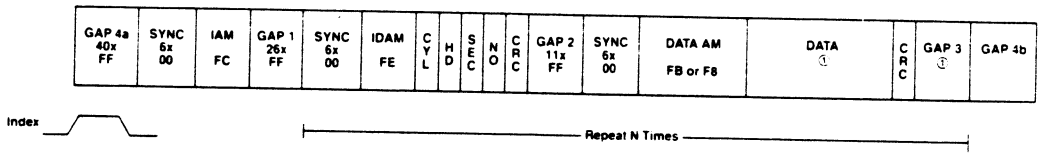
Invalid

If an Invalid command is sent to the FDC (a command not defined above), then the FDC will terminate the command after bits 7 and 6 of Status Register 0 are set to 1 and 0 respectively. No interrupt is generated by the μPD765A/μPD7265 during this condition. Bits 6 and 7 (DIO and RQM) in the Main Status Register are both high (1), indicating to the processor that the μPD765A/μPD7265 is in the Result phase and the contents of Status Register 0 (ST0) must be read. When the processor reads Status Register 0 it will find an 80 hex, indicating an Invalid command was received.

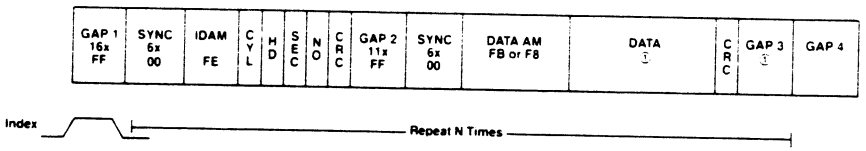
A Sense Interrupt Status command must be sent after a Seek or Recalibrate Interrupt, otherwise the FDC will consider the next command to be an Invalid command.

In some applications the user may wish to use this command as a No-Op command to place the FDC in a standby or No Operation state.

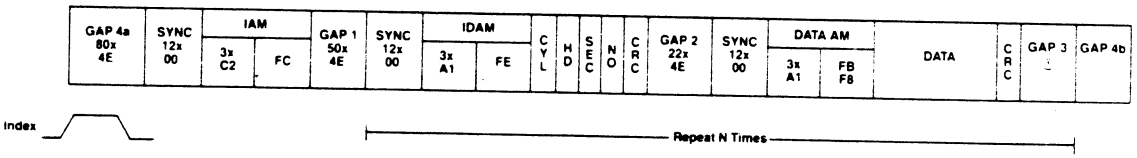
μPD765A (FM Mode)



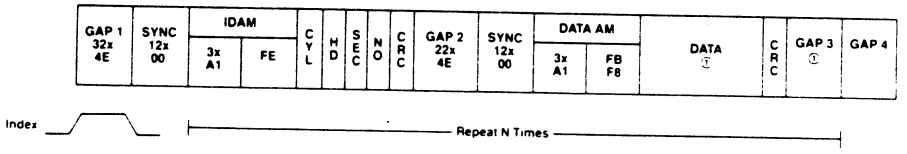
μPD7265 (FM Mode)



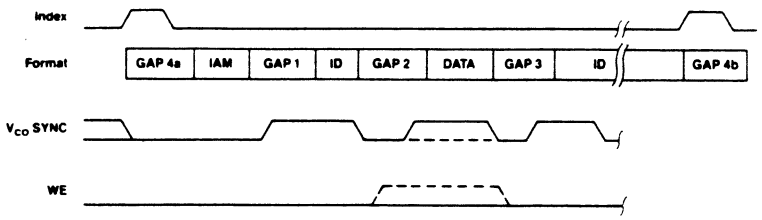
μPD765A (MFM Mode)



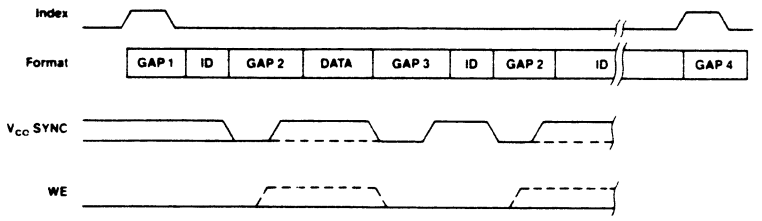
μPD7265 (MFM Mode)



μPD765A



μPD7265



Note: ——— Read
– – – – Write

Notes: It is suggested that the user refer to the following application notes
① #8 — for an example of an actual interface as well as a theoretical data separator
② #10 — for a well documented example of a working phase – locked loop

Package Outlines

For information, see Package Outline Section 7.

Plastic, μPD765AC/7265C
Ceramic, μPD765AD/7265D

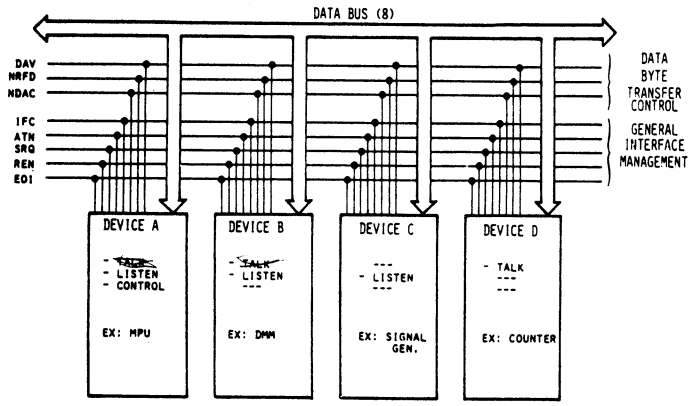


Fig. 6-6: 488 Bus Signals

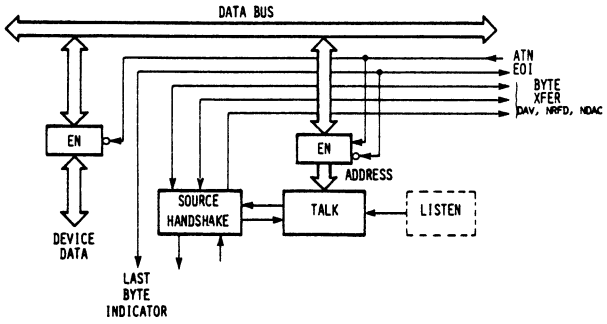


Fig. 6-8: Talker

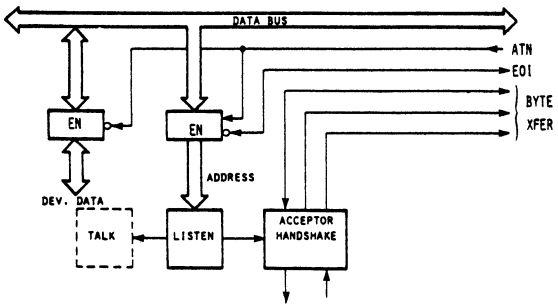


Fig. 6-9: Listener

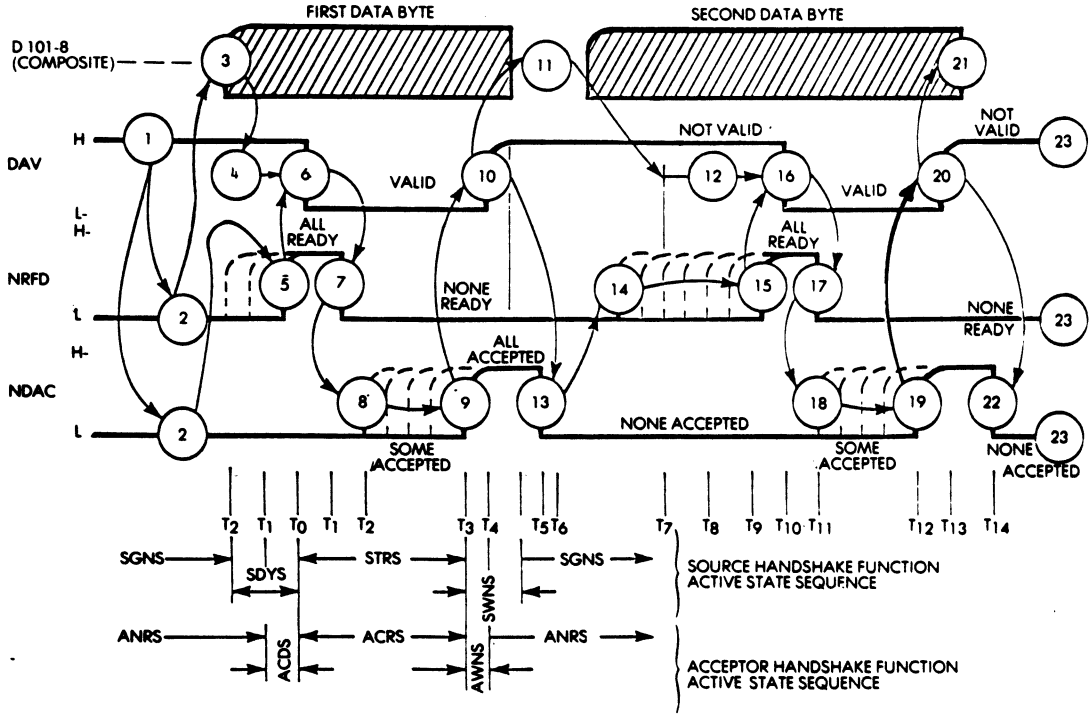
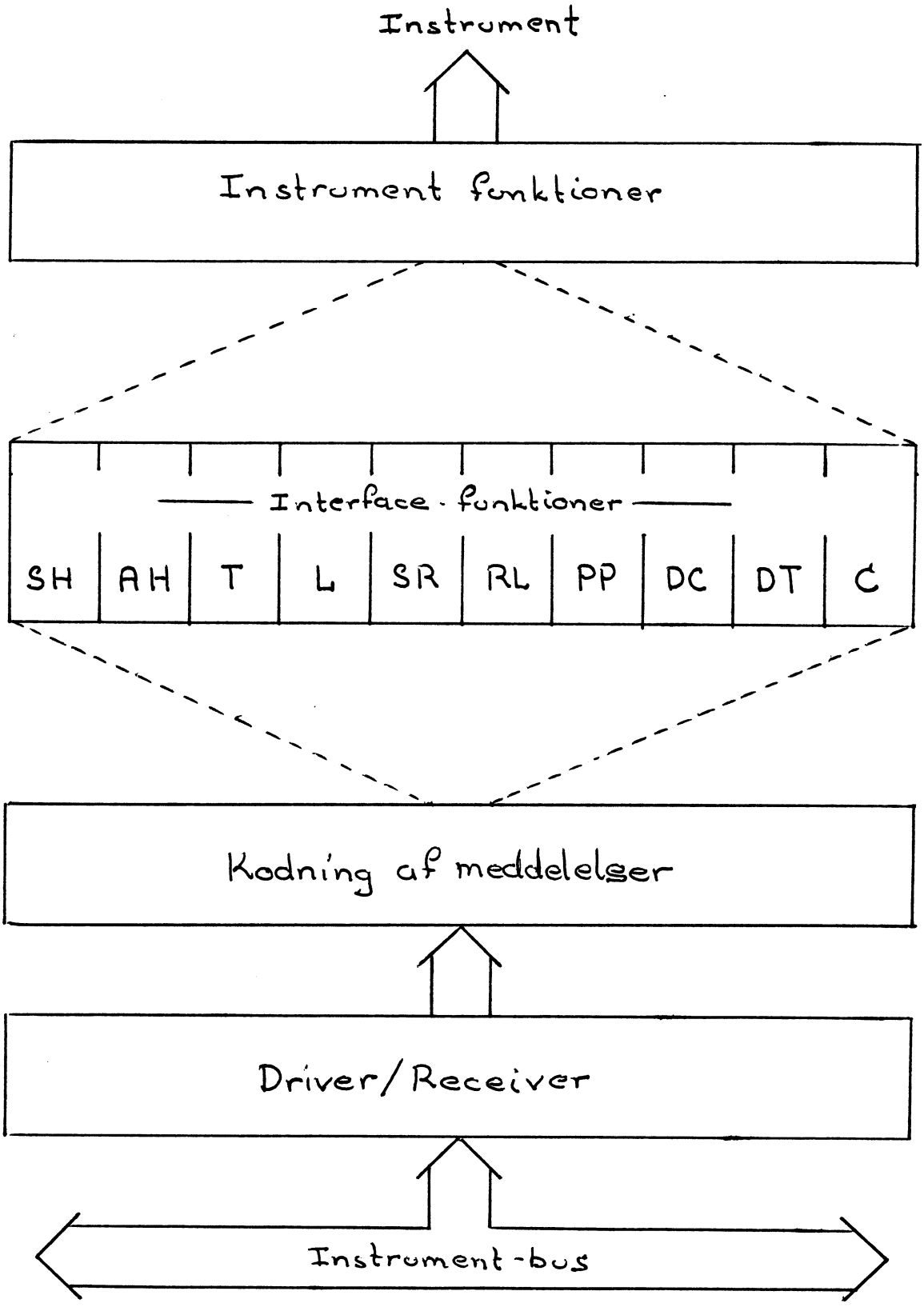
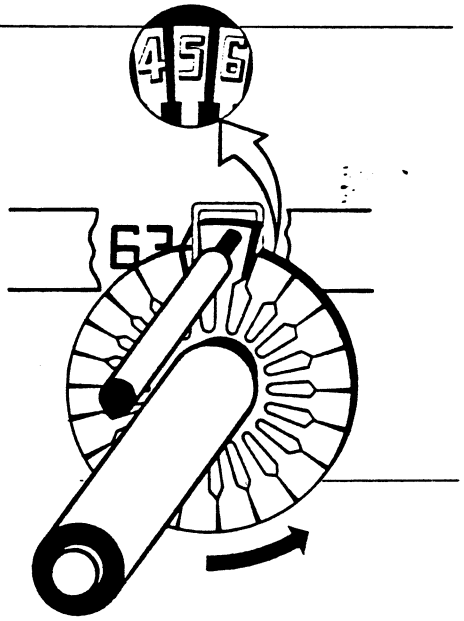


Fig. 6-7: 488 Handshake Timing


Instrument
Instrument funktioner
Interface - funktioner
SH
AH
T
L
SR
RL
PP
DC
DT
C
Kodning af meddelelser
Driver/Receiver
Instrument-bus

THE FIVE TYPES OF PRINT MECHANISM

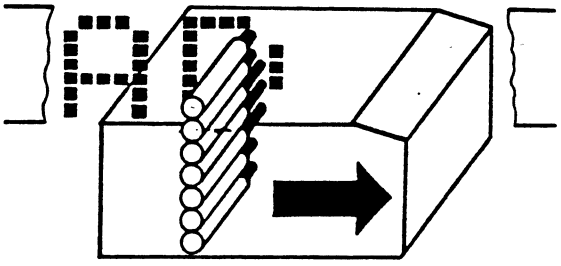
1. Daisywheel



The daisywheel has superseded the golfball on most electronic typewriters and is the most common mechanism on computer printers for producing 'letter' or 'typewriter'-quality output. The type characters all sit at the end of spokes on a 3" diameter plastic wheel. The wheel spins at high speed until the appropriate character is reached when a small hammer strikes the spoke onto a conventional ribbon and prints the characters. Cheap daisywheels can print only 10-15 characters per second (cps) whilst the more expensive can achieve 45-60.

2. Impact Dot-Matrix

This is the most common type of microcomputer printer, and is called an impact mechanism because the image is created by striking a ribbon onto paper. The print head in fact contains a column or needles which can be independently forced onto the ribbon by electro-magnets as it moves across the paper. The resultant text is consequently made up from dots, similar to that on the computer's screen.



It follows that the more needles the print head has and the closer together the dots on the paper, the more readable the text. A 9-pin head, for example, will usually produce characters on a matrix of 9x7 dots, for example.

A recent development is the **multiple-pass** system, where the head passes more than once over the same line of text, offsetting the dots slightly, so that the characters appear to be constructed from solid lines, not dots.

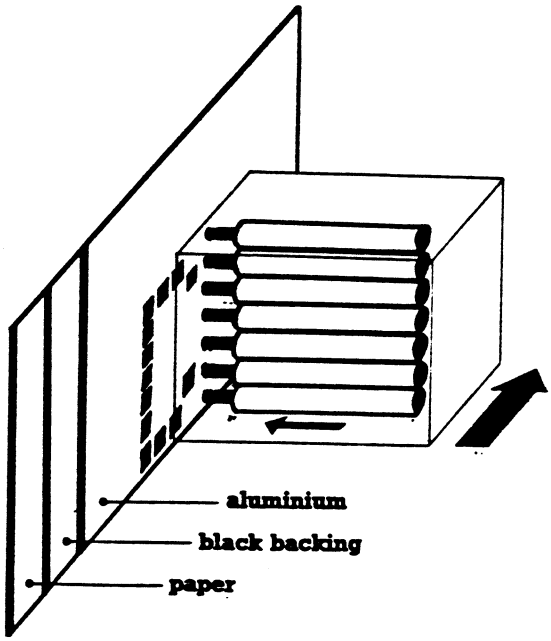
A two-pass print is far more readable than the conventional dot matrix image whilst a three-pass is virtually indistinguishable from a daisywheel's output.

3. Thermal

As with an impact dot-matrix printer the image achieved using a thermal printer is constructed from a matrix of dots, but instead of moving needles, the print head contains a column of miniature heat elements. As the head moves across the paper, these elements heat up and cool down very rapidly according to the requirements of each letter for the presence or absence of appropriate dots in the matrix.

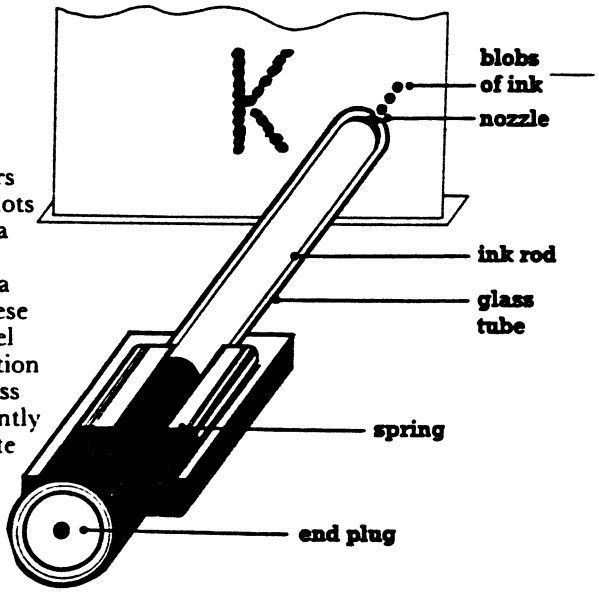
Thermal printers originally required special thermal paper which produced black dots in response to these points of heat. The result looks very similar to impact dot-matrix, though the image is seldom as dark. Although thermal paper is expensive, the operation of the printer is quieter, and in many cases faster than in any other category of printer.

A further recent development in this field is thermal ribbon which achieves the same result on ordinary paper and therefore permits you to use paper bearing your company's letterhead. Many of the new Japanese portable computers feature printers that work either with thermal paper, or thermal ribbon and ordinary paper.



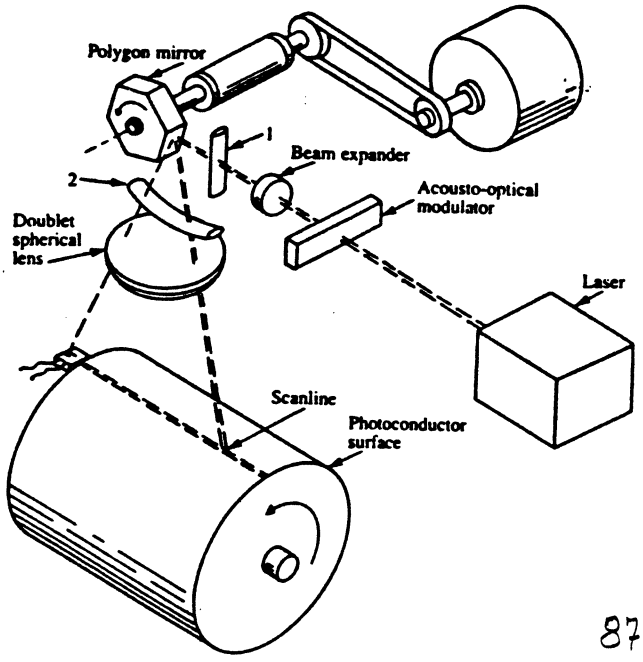
4. Ink Jet

Ink Jet printers also offer an 'almost' silent mode of operation, they construct characters from dots, but employ a very high ratio of dots per inch of paper. The print head contains a reservoir of special ink, a nozzle and a high speed pulsing pump which literally squirts a stream of tiny blobs of ink at the paper. These blobs are deflected up or down as they travel through the air by a high voltage; the variation in this voltage as the print head moves across the paper, forms the image. *Sharp* have recently produced a full colour version using separate red, blue, yellow and black jets.



5. Others

There are other ways of creating an image on paper, but these are either rare and expensive, or new and experimental. The best known is probably the **laser printer** which at anything from £9,000 to £30,000 is strictly for those who produce *very* high volumes of output – a printed page of A4 in one second is typical. Laser printers work on the same basis as electrostatic photocopiers, but using a controlled laser to draw the image onto a charged metal drum. Fine carbon particles stick to the paper where the charge is left and these are baked on hard to form an image.



APPENDIX B INTERFACE SIGNALS

Table B-1 Parallel Interface Signals

Pin No.	Signal	Direction	Description
1	$\overline{\text{DATA STROBE}}$	To printer	Samples input data when changing from low level to high level.
2	DATA BIT 1	To printer	Indicate input data. High level indicates "1" and low level "0".
3	DATA BIT 2		
4	DATA BIT 3		
5	DATA BIT 4		
6	DATA BIT 5		
7	DATA BIT 6		
8	DATA BIT 7		
9	DATA BIT 8		
10	$\overline{\text{ACKNOWL-EDGE}}$	From printer	Indicates character input completion, or function operation end at low level.
11	BUSY	From printer	Indicates data cannot be received at high level. Data can be input at low level.
12	PAPER END	From printer	High level indicates paper end.
13	SELECT	From printer	High level indicates the select (on-line) condition.
14, 16, 33	0 V	–	Signal ground
17	CHASSIS GROUND	–	Frame ground
18	+5 V	From printer	+5 V supply (50 mA maximum)
19 to 30	0 V	–	Twisted pair return (For pins 1 to 11)
31	$\overline{\text{INPUT-PRIME}}$	To printer	Controller is initialized at low level. Pulse width more than 5.0 ms.
32	$\overline{\text{FAULT}}$	From printer	From high to low level when paper runs out.
15, 34, 35, 36		–	Unused

Note: Pin arrangement:

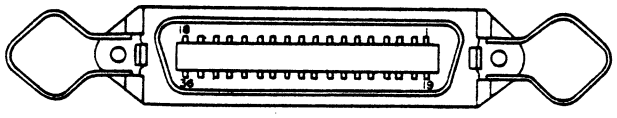


Figure B-1 Connector Pin Arrangement

APPENDIX D BLOCK DIAGRAM

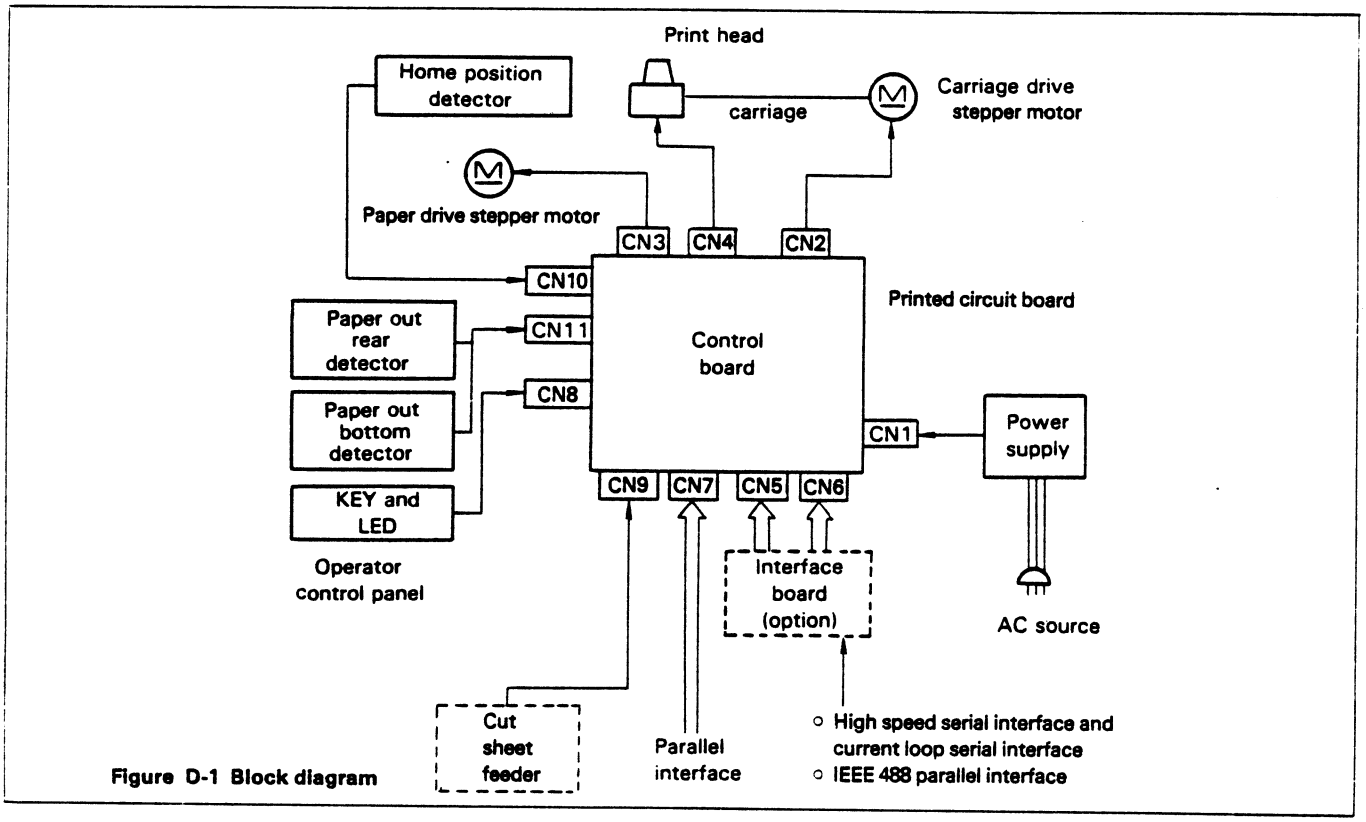


Figure D-1 Block diagram

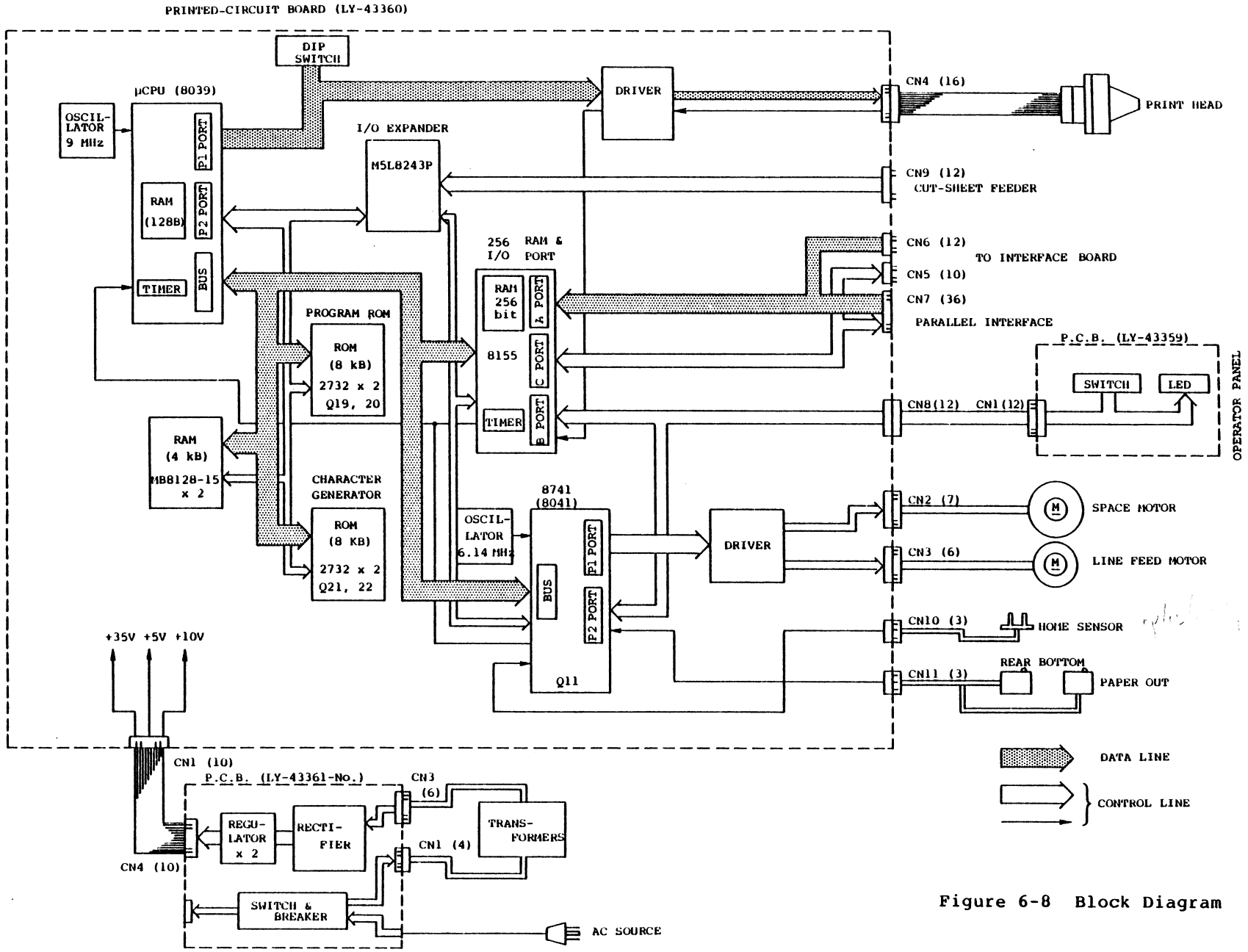


Figure 6-8 Block Diagram

APPENDIX G FUNCTION CODES

Table G-1 Table of Function Codes

Command	Function code		Description
	Decimal	Hexadecimal	
LF	10	0A	Moves paper up one line.
CR	13	0D	Returns carriage.
FF	12	0C	Feeds paper to the first line of next form (TOF: top-of-form)
VT 31H to 3CH	11 49 to 60	0B 31 to 3C	Feeds paper to tab position of same channel number as set in VFU.
HT	9	09	Take space up to the next HT position
BS	8	08	Take one character backspace. Valid in the incremental mode only.
DC1	17	11	Sets the printer in select (on-line) condition.
DC3	19	13	Releases the printer from select condition, and sets it in de-selected (off-line) condition.
DC4	20	14	Loads tab position in VFU.
RS	30	1E	Designates 10 CPI.
GS	29	1D	Designates 12 CPI.
FS	28	1C	Designates 17 CPI.
US	31	1F	Designates wider characters.

Command	Function Code		Description
	Decimal	Hexadecimal	
SO	14	0E	Shifts out character set in case of 7-bit code.
SI	15	0F	Shifts in character set in case of 7-bit code.
CAN	24	18	Clears buffer. Invalid in the incremental mode.
ESC · 0	27 · 48	1B · 30	Designates the ordinary character generator. The printer turns into this mode when the power is turned on or the I-PRIME signal is received.
ESC · 1	27 · 49	1B · 31	Designates the character generator for CQ.
ESC · 2	27 · 50	1B · 32	Designates the character generator for downline loadable character generator.
ESC · 5	27 · 53	1B · 35	Set TOF (first line of printing).
ESC · 6	27 · 54	1B · 36	Designates 6 LPI.
ESC · 8	27 · 56	1B · 38	Designates 8 LPI.
ESC · C	27 · 67	1B · 43	Attaches an underline to the characters after this command.
ESC · D	27 · 68	1B · 44	Prints the characters after this command without underline.

Esc 1B 27 ESC cancel/start
LF 10

Table G-1 (con.)

Command	Function code		Description
	Decimal	Hexadecimal	
ESC · F · 0.0 to 9.9	27 · 70 48 · 48 to 57 · 57	1B · 46 30 · 30 to 39 · 39	Designates the number of lines per page length.
ESC · G 0.0 to 9.9	27 · 71 48 · 48 to 57 · 57	1B · 47 30 · 30 to 39 · 39	Designates the length of line spacing per page. The length is N/2 inches.
ESC · H	27 · 72	1B · 48	Designates emphasized printing for the characters after this command (with 1/144 inch line spacing).
ESC · I	27 · 73	1B · 49	Cancels emphasized printing for the characters after this command.
ESC · J	27 · 74	1B · 4A	Designates superscript printing for the characters after this command.
ESC · K	27 · 75	1B · 4B	Cancels superscript printing for the characters after this command.
ESC · L	27 · 76	1B · 4C	Designates subscript printing for the characters after this command.
ESC · M	27 · 77	1B · 4D	Cancels subscript printing for the characters after this command.

Command	Function code		Description
	Decimal	Hexadecimal	
ESC · N n	27 · 78 · 1 to 11	1B · 4E 01 to 0B	Designates the number of character-to-character spaces for dot expansion line. "n" is a binary number of max. 11 ($0B_H$).
ESC · V	27 · 86	1B · 56	Designates single CSF exhaust.
ESC · S	27 · 83	1B · 53	Designates single CSF inhalation.
ESC · T	27 · 84	1B · 54	Designates emphasized printing for the characters after this command (without 1/144 inch line spacing).
ESC · % · 1 n1 · n2	27 · 37 · 49 n1 · n2	1B · 25 · 31 n1 · n2	Designates start of full graphic printing with half dots. "n1" and "n2" are the numbers of graphic codes.
ESC · % · 2 n1 · n2	27 · 37 · 50 n1 · n2	1B · 25 · 32 n1 · n2	Designates start of full graphic printing without half dots. "n1" and "n2" are the numbers of graphic codes.
ESC · % · 9 n3	27 · 37 · 57 n3	1B · 25 · 39 n3	Designates line spacing of 1/144 inch multipled by "n3." $n3 \leqq 127$
ESC · % · A C	27 · 37 · 65 C	1B · 25 · 41 C	Loads a one-character pattern into the downline loadable CG. "C" is a code in the range of ^ 20 to ^ 5F.

Table G-1 (con.)

Command	Function code		Description
	Decimal	Hexadecimal	
ESC · VT · 0.0 to 9.9	27 · 11 48 · 48 to 57 · 57	1B · 0B 30 · 30 to 39 · 39	Directly skips as many as the designated number of lines.
ESC · HT n · CR	27 · 9 n · 13	1B · 09 n · 0D	Sets a tab in the HT memory. 3 digit number
ESC · O	27 · 79	1B · 4F	Designates incremental printing for the characters after this command.
ESC · P	27 · 80	1B · 50	Cancels incremental printing for the characters after this command.

Cautions on printing mode selection:

1. A dot expansion line means on-line data stored as dot patterns in the 2K-byte RAM when the line includes CQ characters or APA printing.
2. Refer to Table F-3 for the number of printable dots of a dot expansion line.
3. A line including CQ characters of full graphic printing (with half dots) is printed only in the forward direction at a printing speed of 100 CPS.
4. A line including full graphic printing (without half dots) is printed only in the forward direction at a printing speed of 120 CPS.
5. If emphasized printing, special graphic character printing, or superscript/subscript printing is designated, the printing direction becomes forward only.
6. If the incremental mode is designated, the printing direction becomes forward only, and emphasized printing, CQ character printing, superscript/subscript printing, and full graphic printing cannot be designated.
7. In superscript/subscript printing, emphasized printing cannot be designated.
8. Superscript/subscript printing designation and underline designation are reset every time the line is changed.
9. In a dot expansion line, horizontal tab cannot be set.
10. If character generator selection is erroneous, designation of printing mode is not guaranteed.