

Den styrlogik som kontrollerar dataöverföringen mellan datorn och drivenheterna, består av en Z80 CPU med ett styrprogram, som ligger lagrat i ett 2 kbyte EPROM. CPU:n har också till sitt förfogande ett statistiskt RAM på 1 kbyte, för lagring av egna variabler samt mellanlagring av data vid dataöverföring mellan datorn och drivenheterna.

Via de I/O-kretsar, som är anslutna till CPU:ns adress-, kontroll- och databuss, kommer alla dataöverföringar att ske mellan datorn och driv enheterna.

Innan datorn kan börja kommunicera med controllerkortet ska kortet aktiveras. Aktiveringen görs med hjälp av Card Select-blocket.

Vid överföring av data mellan dator och controllerkortets CPU, används port A på PIO-kretsen, som via en dubbelriktad buffertkrets är ansluten till datorenhetens databuss.

Data som överförs TILL controllerkortet via bufferten och port A kan vara kommandon som ska tolkas av CPU eller data som ska lagras på flexskivorna.

Data som kommer FRÅN controllerkortet via port A och bufferten till datorenheten, är data som har varit lagrade på flexskivorna eller en felkod (om fel inträffar).

Statusporten läses av datorenheten och innehållet indikerar olika statuslägen på controllerkortet. Statusvärdet skrivs dit av CPU:n.

I/O-kontroll används av datorenheten för att avge RESET-signal samt för att sätta controllerkortet i kommando-mod.

När CPU:n ska lagra eller hämta data från flexskivorna, använder den sig av tre I/O-kretsar för att generera kontroll-, läs- och skrivsignaler samt för att läsa statusvärden.

PIO-kretsens port B är programmerad i bitmode så att en del bitar är utsignaler och en del insignaler. Dessa används för att generera kontrollsignaler samt även för att läsa in vilket format data är lagrat med på skivan.

Drive kontroll-porten är enkelriktad och används för att lägga ut olika kontrollsignaler.

Den sista I/O-kretsen är en "Floppy Disc Controller/Formatter", FD 1791. FD 1791 är mycket komplex krets, konstruerad för en speciell uppgift, nämligen att genererara och läsa kontrollsignaler till/från drivenheter, översätta data från CPU:n till ett seriellt format som passar för lagring på flexskivor och vid läsning av data från flexskivan omvandla dessa till ett format som CPU:n kan läsa.

Kretsen fungerar så att CPU:n bara behöver ge den övergripande instruktioner, sedan sköter den själv generering och läsning av kontrollsignaler och dataöverföring till och från drivenheterna.

För att öka datasäkerheten (minska antalet felläsningar) låter man signalen WR Data från 1791, som innehåller information, vid skrivning passera en krets där man gör en förkompensering av skrivsignalen, innan den läggs ut till drivenheterna. Vid läsning av lagrade data kommer Read Data-signalen från drivenheten att passera en Data Separator (Data Recovery) - krets, med vilken man återskapar en klockpuls som är fastlåst till den inkommande dataströmmen. Från kretsen kopplas signalerna RCLK och RDATA in till 1791:n och omvandlas till parallella data som CPU:n kan läsa.

6.2 Anpassningslogik mot datorenhet, kommando och dataöverföring

I den 64-pol busskontakten på datorenheten finns det ett antal strobsignaler; OUT 1-5 samt INP 0-1. Dessa strobar aktiveras när datorenheten exekverar en IN eller en OUT-instruktion. Ex. (Basic instr.) OUT 1,45 kommer att aktivera OUT 1-stroben samtidigt som data 45 finns tillgängligt på databussen.

När ett I/O-kort konstrueras enligt 4680-standard används dessa strobar för speciella funktioner.

INSTRUKTION (Basic)	FUNKTION
OUT 1,Data	Gör kortval där data=kortadress.
OUT 0,Data	Skriver data till I/O-kort.
OUT 2,Data	Fri funktion. På controllerkortet sätter den CPU:n i kommando-mode.
OUT 3,Data	Fri funktion.
OUT 4,Data	Fri funktion. På controllerkortet RESET:ar den CPU samt alla I/O-portar.
Out 5,Data	Fri funktion.
INP(0)	Läser in data från I/O-kort.
INP(1)	Läser in status på I/O-kort.
INP(7)	Gör RESET på alla anslutna I/O-kort

När datorenheten vill kommunicera med controllerkortet börjar den med att lägga ut instruktionen OUT 1,45, vilket gör att OUT1-stroben (stift 23A i den 64-pol kontakten) blir aktiv låg och påverkar Three State-ingången på ett av buffertstegen i kretsen LS 125. Utgången på buffertsteget aktiverar signalen CS (Card Select), som är kopplad till klockingången på kretsen DM 8131.

DM 8131 innehåller sex XOR-grindar med gemensam utgång. En av ingångarna till varje grind är ansluten till databussen medan den andra ingången på varje grind är ansluten antingen till låg eller till hög nivå (ger binärt 45).

När CS blir aktiv finns värdet 45 (OUT 1,45) på databussen. Värdet jämförs i kretsen med det förinställda värdet på de andra ingångarna. När CS-signalen går från låg till hög nivå kommer utgången stift 9 att bli aktiv låg. Signalen från stift 9 är kopplad till ingångarna på fem buffertkretsar, som läggs på låg nivå. Samtidigt tänds också lysdioden som finns på kortet och som indikerar att kortet är selekterat.

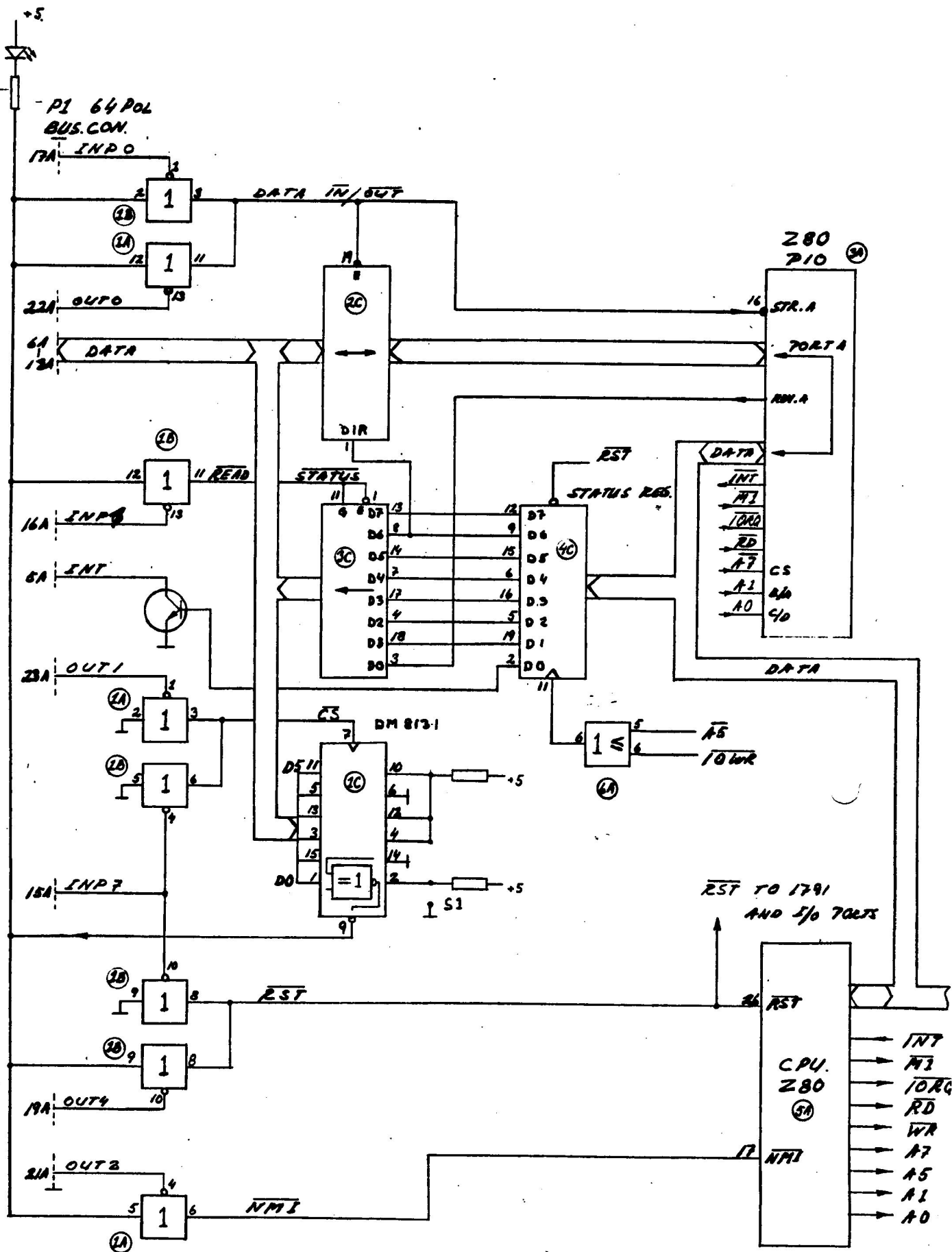


Fig 6.2 Anpassingslogik mot datorn

Controllerkortet kommer nu att vara inkopplat tills en av följande saker inträffar:

- o En OUT 1-instruktion med en annan adress än 45 läggs ut av datorenheten, vilken kommer att göra signalen på stift 9 hög.
- o En INP(7) instruktion som aktiverar INP7-stroben (stift 15A i busskontakten), som i sin tur via Three State-ingången till en annan buffertkrets, aktiverar CS-signalen. Eftersom det är en INP instruktion som utförs finns inga giltiga data på bussen vilket gör att signalen på stift 9 blir hög även i detta fall.

INP7-stroben påverkar samtidigt ytterligare en buffertkrets som via sin utgång kommer att avge RST-signal till CPU och alla I/O-portar på controllerkortet.

INP7-stroben är en signal som RESET:ar alla I/O-kort som är anslutna till busskontakten. Signalen aktiveras när datorenheten får RESET-signal, men kan också genereras med instruktionen INP(7).

OUT4-stroben (stift 19A i busskontakten) aktiveras med INP(4) och används för att göra en intern RST på controllerkortet. OUT4-stroben kommer att avge RST-signal om ingången till den buffertkrets som stroben påverkar är låg. Insignalen till bufferten kommer från CS-kretsens utgång och är låg när kortet är selekterat.

OUT2-stroben (stift 21A) aktiverar på samma sätt NMI-ingången på CPU:n via den buffertkrets utgång som är ansluten till NMI-ingången på CPU-kretsen.

När en NMI-signal läggs in till CPU:n kommer denna att ställa sig i kommando-mode. Detta innebär att CPU:n går till en speciell inmatningsrutin där den programmerar PIO-kretsens port A för att läsa in data från den dubbelriktade bufferten i pos 2C. Buffertens DIREktion-ingång sätts hög av CPU:n via statusregistrets databit 6 och som riktar bufferten in mot port A.

Datorenheten kan sedan skriva data till port A med instruktionen OUT 0,Data. Instruktionen påverkar OUT0-stroben (stift 22A) som via en buffertkrets aktiverar Enable-ingången på databufferten och öppnar denna så att data kopplas fram till port A. OUT0-stroben aktiverar samtidigt STrobe.A-ingången på PIO-kretsen och data läses in till port A. När data lästs in, läggs ReaDY.A-utgången på låg nivå för att markera att dataregistret i port A innehåller data. RDY.A-signalen sätts hög igen när CPU:n har läst data från port A. De data som CPU:n läser in i det här läget tolkar den som kommandon och beroende på vilket kommando datorenheten har skickat kommer olika aktiviteter att äga rum.

Statusregistret styrs av CPU:n som en I/O-port med I/O adress 223. Adressbit A5 tillsammans med signalen IOWR aktiverar klockingången och läser in data från bussen. Statusregistrets data-utgångar D7-D1 är kopplade till ingångarna på buffertkretsen i pos 3C. Via denna buffert läser datorenheten med instruktionen INP(1) controllerkortets status. INP1-stroben aktiverar G- och E-ingångarna och statusregistrets data läggs ut på bussen.

När CPU:n har fått ett kommando för att överföra data från controllerkortet till datorenheten, kommer den först att programera om PIO:ns port A så att den riktar ut mot bussen samtidigt som den också ändrar riktningen på databufferten via D6 i statusregistret. Datorenheten kan sedan läsa data från port A med INP(0)-instruktionen, som aktiverar INP0-stroben (stift 17A) och via buffertkretsens utgång gör Enable på databufferten samt aktiverar STR:A-signalen på PIO:n.

I det här fallet fungerar STR.A och RDY:A signalerna på följande sätt: när RDY.A blir hög indikerar den att port A har blivit laddad med data och när sedan STR.A-ingången, efter att data har lästs, går från låg till hög nivå kommer den att återställa RDY.A-signalen till låg nivå.

RDY.A och STB.A är handskakningssignaler mellan datorenheten och controllerkortets CPU. RDY.A-signalen är framdragen till Statuslatchen så att datorenheten kan läsa om data finns att hämta vid läsning eller vid skrivning att det är klart att lägga ut nya data till port A.

PIO-kretsens port A kontrolleras av CPU med I/O-adresserna 125 för kontroll och 124 för dataöverföring. Vid dataöverföring mellan dator och controllerkort används interrupt-funktionen hos PIO-kretsen, där STR.A kommer att aktivera INT-signalen.

Via statusregistrets databit 0 kan en INT till datorenheten avges. Denna funktion utnyttjas inte i samband med ABC-datorerna.

6.5 WAIT-logik

När CPU:n exekverar en I/O instruktion aktiveras signalen IORQ tillsammans med RD eller WR beroende på om det är en läs- eller skrivinstruktion. Signalen MREQ aktiveras på samma sätt tillsammans med RD eller WR vid läsning eller skrivning i minnet.

I bägge fallen kan man genom att aktivera WAIT-ingången på CPU:n få den att förlänga den pågående läs- eller skrivcykeln lika länge som WAIT-ingången är aktiv.

På controllerkortet används möjligheten att generera WAIT till CPU:n bara vid dataöverföring till och från controller-kretsen FD 1791.

Logiken består av ett antal grindor och en J-K vippa, som avkodar ett flertal insignaler.

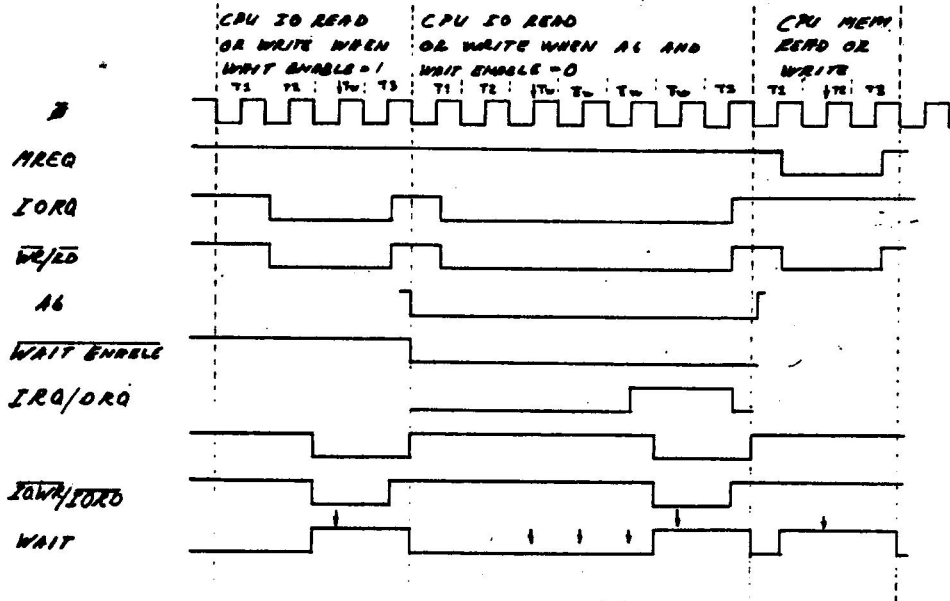
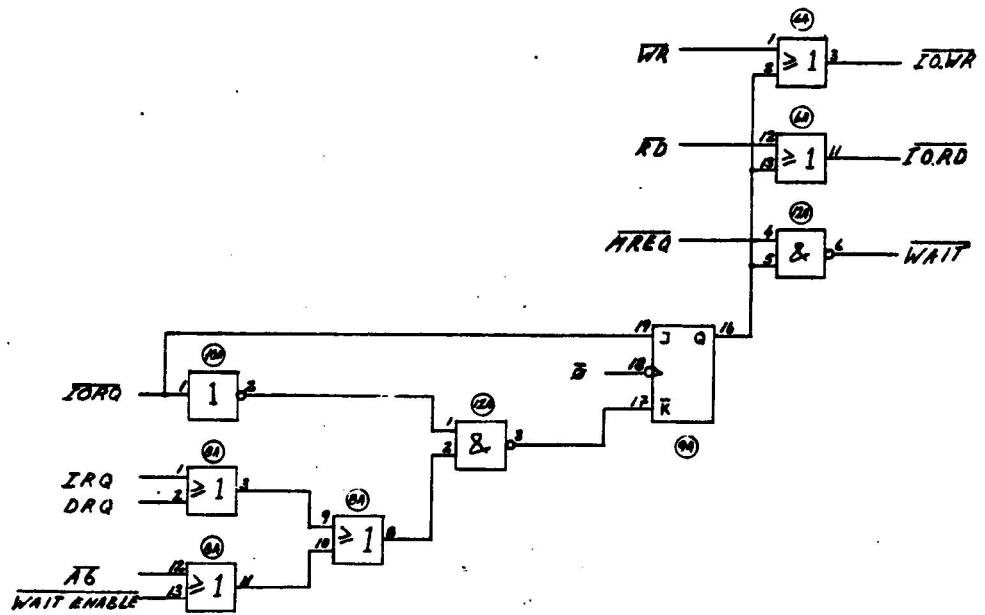


Fig 6.5 WAIT-logik

Insignaler som avkodas är de som nämnts tidigare, dvs IORQ, MREQ, RD, WR från CPU:n samt två signaler från controllerkretsen FD1791: DRQ och IRQ (Data Request och Interrupt Request). De två senare signalerna har en aktiv hög nivå.

DRQ-signalen från 1791:n indikerar att data finns att hämta i dataregistret. Vid skrivning av data till skivan indikerar DRQ att dataregistret är tomt. Signalen deaktiveras så snart CPU:n har läst eller skrivit in data.

IRQ-signalen indikerar, när den är aktiv, att ett kommando har slutförts eller att ett fel har inträffat vid exekveringen av kommandot. Signalen återställs genom att CPU:n läser av statusregistret i 1791:an eller skriver ett kommando i kommandoregistret.

Signalen WAIT ENABLE kommer från en I/O-port som används av CPU:n för att generera olika kontrollsignaler, varav en är WAIT ENABLE. WAIT ENABLE-signalen är aktiv låg och först när den är aktiv kan WAIT genereras till CPU:n.

Adressbit 6 från CPU:n avkodas och indikerar att det är FD1791 som adresseras. FD1791 adresseras som en I/O enhet där A6 låg ger CS (Chip Select) signal.

När WAIT ENABLE har hög nivå kommer den ena av ingångarna på NAND-grinden (pos 12A, stift 2) också att läggas hög via två OR-grindar. De tre andra insignalerna DRQ, IRQ och A6, som har möjlighet att påverka samma ingång, kommer i detta läge inte att kunna göra det.

Som en andra insignal har samma NAND-grind IORQ via en inverterare. Om ingen I/O-instruktion utförs av CPU:n, kommer utgången på grinden att vara hög. Den efterföljande J-K vippans Q-utgång kommer att vara hög eftersom J och K-ingångarna har hög nivå. Q-utgången är ansluten till tre grindar, en NAND-grind och två OR-grindar. Genom att Q-utgången är hög, kommer OR-grindarnas utsignaler inte att aktiveras när RD eller WR signalerna är aktiva vid minnesoperationer. MREQ, som indikerar minnesoperation, kommer då via NAND-grinden att göra WAIT-signalen till CPU:n inaktiv under den tid som CPU:n känner av läget på sin WAIT-ingång.

När IORQ aktiveras vid en I/O läs eller skrivinstruktion (WAIT ENABLE fortfarande hög), kommer Q-utgången att gå låg och WAIT-signalen från NAND-grinden går hög, för att inte generera WAIT till CPU:n. Samtidigt öppnas de båda OR-grindarna så att signalerna IOWR och IORD kan genereras av respektive insignal. Dessa signaler blir en kombination av IORQ och WR eller RD. Signalerna används för att aktivera I/O portarna vid läsning eller skrivning.

Om WAIT ENABLE aktiveras kommer A6 att bestämma nivån på stift 2 på NAND-grinden i pos 12A. För alla I/O-adresser, som har A6 hög, kommer förloppet att bli som beskrivits ovan. Men om controllerkretsen adresseras går A6 låg och ingången till NAND-grinden blir låg. När sedan IORQ blir aktiv kommer grindens utgång inte att ändra värde utan K-ingången blir hög och J-ingången blir låg. När sedan vippans klockas kommer inget omslag att ske på Q-utgången, utan denna ligger kvar på hög nivå vilket gör att när CPU:n känner av läget på WAIT-ingången så är den aktiv och s.k. Wait States adderas till och förlänger instruktionscykeln tills någon av de två signalerna DRQ eller IRQ blir aktiva. Därmed får K-ingången låg nivå. Q-utgången slår om till låg nivå. WAIT-signalen blir hög och IORD/IOWR kan avges samt CPU:n kan slutföra instruktionscykeln.

WAIT genereringen används t.ex. när ett datablock ska skrivas ut på skivan. En byte från RAM-minnet ska då överföras till FD1791:ns dataregister varje gång som DRQ går hög. Om man då använder sig av instruktionen OUTI (block överförings instr.) kommer WAIT-signalen att vara låg när FD1791 adresseras (A6 låg) och CPU:n kommer då att generera Wait States tills DRQ går hög och CPU:n kan överföra data till 1791:ans dataregister. CPU försöker sedan överföra nästa byte och förloppet upprepas.

IRQ måste kännas av för att CPU:n ska komma ur WAIT läget om 1791 indikerar ett fel.

6.6 Anpassning och kontrollsignaler till drivenheterna

Vid kommunikation med drivenheterna använder sig CPU:n av tre I/O-kretsar, PIO-kretsens port A, controllerkretsen FD1791 samt en LS 273, som används som en I/O-port.

PIO-kretsens port B används av CPU:n för att hålla reda på dataformat på drivenheterna och status på några kontrollsignaler från FD 1791. PIO:ns port B adresseras av CPU:n med I/O-adress 127 för kontroll och adress 126 för dataöverföring.

Port B är programmerad för bit mode vilket innebär att vissa bitar är riktade in mot CPU:n och en del riktade ut från CPU:n. Dom bitar som är riktade in mot CPU:n används för att läsa status på följande signaler:

- Bit D4 Radial/Binär Select. Indikerar om drive nr som skrivs till controllerkortet ska översättas till motsvarande select-signal eller om värdet ska läggas ut binärt på Select-utgångarna. Hög nivå för Radial Select
Signal återverkan på Select 0,1,2.
- Bit D0 Enkel/Dubbel Sida. Används för att indikera om driven, som är ansluten till kortet, kan läsa/skriva på bägge sidor av flexskivan. Låg nivå för enkelsidiga drivar.
Signal återverkan på Side Select
- Bit D1 Dubbel/Enkel densitet. Indikerar dataformat för respektive drive, via switch 1,2 och NAND-grunden i pos 12A, stift 9 och 10.
Låg för enkel densitet.
Signal återverkan på Dubbel DENSitet-signalen till FD 1791.
- Bit D6 HDLD. Indikerar om Head Load-signalen från FD1791 till drivenheten är aktiv eller ej.
- Bit D7 IRQ. Interrupt Request från FD1791 blir aktiv hög när FD1791 har slutfört ett kommando eller när ett fel inträffar.
PIO-kretsen kan programmeras för att avge interrupt till CPU när IRQ blir aktiv. Eller CPU:n kan läsa port B och känna av värdet på IRQ för att få information om när FD1791 är klar.

Utsignaler

- Bit D5 HLT. Head Load Timing-signal till FD1791.
När FD1791 har aktiverat HDLD-signalen till drivenheten tar det en stund för driven att hinna göra Head Load. FD1791 känner av nivån på HLT-ingången och när den sätts hög antar FD1791 att Head Load har utförts i drivenheten.
Tiden det tar för att göra Head Load är olika mellan olika drive fabrikat. Genom att HLT kontrolleras av CPU:n är det lätt att anpassa tiden för olika drivenheter genom att ändra i styrprogrammet.
- Bit D3 DDEN, Dubble Density. Kontrollsignal till FD1791 som ställer om för dubbel densitet eller enkel densitet.
Kontrolleras av användaren via switch 1,2 för respektive drive.
Låg nivå ger Dubble Density-format.

Kretsen LS 273 adresseras av CPU:n med I/O adress 239. Via denna krets genereras de kontrollsignaler till drivenheterna som FD1791 inte kan generera. Kretsen klockar data från bussen när A4 och IOWR går från aktiv låg till hög nivå.

RST-signalen är kopplad till Clear-ingången så att alla Q-utgångarna blir låga när kortet RESET:as

Kontrollsignaler som genereras av respektive databit är följande:

- Bit D7 MR. Master Reset till FD 1791.
- Bit D6 WAIT ENABLE. En aktiv låg nivå på den här signalen gör att WAIT till CPU:n kommer att genereras vid läsning eller skrivning till FD 1791 (beskrivet tidigare).
- Bit D5 PRECOMP ON. Den här signalen är kopplad till kretsen ULA 2C140:s Write Precompensation-block och används för att stänga av eller slå på förkompenseringen av skrivdata som sänds till drivenheten. Signalen sätts aktiv av CPU:n vid dubbel densitet och ett spår över 24 adresseras.
- Bit D4 SIDE. Väljer sida om en drivenhet med dubbla läs och skrivhuvuden används.
- Bit D3 MOT ON. Motor On-signal till drivenheten som startar motorn som driver flexskivan. Signalen är aktiv låg och kopplas via efterföljande buffertsteg till samtliga drivenheter.
- Bit D2 SEL 2. Selektar respektive drivenhet. Signalen
- Bit D1 SEL 1. måste vara aktiv för att kontrollsignalerna
- Bit D0 SEL 0. ska kunna påverka respektive drive.
Signalerna är här aktivt höga men inverteras i efterföljande buffertsteg till aktivt låg nivå.

Övriga kontrollsignaler som behövs till drivenheterna genererar controllerkretsen FD 1791. För en komplett beskrivning av FD 1791, se separat beskrivning.

FD 1791 styrs och kontrolleras av CPU:n som en I/O-enhet. Signaler som används vid kommunikation med CPU:n är WR och RD som aktiveras av IOWR respektive IORD signalerna. Dessa är en kombination av CPU:ns IORQ och RD eller WR. CE (Chip Enable) ingången kontrolleras av A6 och är aktiv låg vid varje adressering av kretsen, medan A0 och A1 väljer vilket register i FD 1791 som avses.

I/O-adresser som används för FD1791 är följande:

Adress	read reg.	write reg.
188	Status	Command
189	Track	Track
190	Sector	Sector
191	Data	Data

IRQ känns av på olika sätt av CPU:n för att tala om när ett kommando är slutfört eller ett fel har inträffat. Signalen återställs så snart ett nytt kommando skrivs i kommandoregistret eller att statusregistret läses.

DRQ används bara vid dataöverföring för att tala om att data finns att hämta eller ska skrivas i dataregistret. Signaleringen till CPU:n går alltid via Wait-logiken.

Utsignaler till drivenheten som FD1791 genererar är följande:

HOLD, signalen är aktiv hög men inverteras till en låg nivå innan den läggs in till selekterad drive och gör att ett relä drar som pressar skivan mot läs och skrivhuvudet. 1791 känner sedan av läget på HLT-ingången och kommer inte att fullfölja kommandot om inte HLT blir aktiv hög.

DIR, Direction-signalen kopplas till stegmotorn i selekterad drive, för att ange om skrivhuvudet ska stegas mot spår 39 (aktiv låg) eller spår 0 (aktiv hög). Signalen inverteras i efterföljande buffertsteg.

STEP, steg pulser till stegmotorn i selekterad drive, en puls stegar fram ett spår. Signalen inverteras i buffersteget.

WG, Write Gate kopplas till skrivelektroniken i drive enheten, signalen måste vara aktiv hög för att skrivning ska kunna ske på flexskivan. WG aktiveras innan data sänds ut på WD-utgången. WG inverteras i buffertsteget.

Signalerna WD, EARLY och LATE kopplas till ett block i kretsen ULA 2C140 som kallas Write Precompensation (Skriv förkompensering).

WD, Write Data innehåller data som CPU:n lagt in i parallell form och som 1791 har transformerat om till seriell form för lagring på flexskivan. Formatet på data är frekvens modulerat (enkel densitet) eller modifierat frekvens modulerat (dubbel densitet).

EARLY och LATE är signaler som talar om för förkompenserings kretsen om datapulserna ska skiftas mer eller mindre isär.

När man skriver in data på skivan kommer avståndet mellan data /klockbitarna att bli tätast när man skriver på dom innersta spåren. Detta ger upphov till någonting som brukar kallas pulsträngsel, vilket gör att när data läses in kan avståndet mellan pulserna ha skiftats så mycket att en fel-detektering sker. För att motverka det här gör man en kompensering i motsatt riktning mot troligt fasskift i förkompenseringskretsarna när man skriver ut data. Avståndet mellan bitarna i WD-signalen är beroende av om det sänds flera ettor eller nollor i rad. Detta indikeras med signalerna EARLY och LATE.

Till förkompenseringskretsen läggs två andra signaler in. SHCLK används för utskiftning av data och PRECOMP ON stänger av eller slår på förkompenseringen, vilket görs efter spår 20. WD-signalen läggs fram till förkompenseringskretsen via en J-K vipa som klockas med 2 Mhz för att få ett fastläge som stämmer med SHCLK-signalen.

Insignaler till FD 1791 är följande:

IP. Indexpulser från selekterad drivenhet indikerar att en skiva är isatt i drivenheten. Om ingen indexpuls finns avbryts kommandot och en IRQ genereras.

TROO. Track noll indikerar att skriv/läs huvudet befinner sig över spår noll.

RDY. Ready indikerar att drivenheten är selekterad samt att skivan i drivenheten har kommit upp i ca:70% av sin nominella hastighet. Om signalen inte är aktiv (hög), avbryter FD1791 pågående skriv- eller läsoperation och genererar IRQ.

WRPT. Write Protect indikerar att skivan i selekterad drivenhet är skriv-skyddad. Innan FD1791 gör en skrivning känner den av nivån och om den inte är aktiv (låg), avbryts instruktionen och IRQ genereras.

RAWD och RCLK kommer från den andra delen i kretsen ULA 2C140 som benäms Data Recovery Block. RAWD är de data som läses från skivan och RCLK är en klocksignal som är fastlåst till RAWD-signalen via Data Recovery-blocket och en VCO-krets.

Signalen Read Data från drivenheten läggs via en buffertkrets fram till en OR-grind. OR-grinden har en andra insignal från FD1791 som benäms VFOE. När en read-instruktion utförs kommer FD1791 att lägga signalen aktivt låg för att släppa fram data till Recovery-kretsen. Signalen kommer indirekt att se till att VFO-kretsen är frisvängande när inte read-instruktionen utförs, eftersom inga pulser släpps fram till recovery-kretsen så länge som signalen är inaktiv.

Data Recovery-kretsen har som insignal en 4 MHz klocksignal från en VFO krets MC 4024. Kretsens frekvens kan ändras genom att variera spänningen på stift 2. 4 MHz läggs in till ett frekvensdelarsteg som delar ned frekvensen olika mycket beroende på signalen DDEN (stift 1). Från stift 15 läggs den sedan in till RCLK-ingången på FD1791. Internt i kretsen kopplas också signalen till en fasjämförare.

Fasjämföraren har också signalen Read Data som insignal, via ett steg som också klockar ut data i takt med RCLK till 1791:ns RAWD-ingång. De båda insignalernas frekvens och fas jämförs och ger upphov till pulser på UP- och Down-utgångarna. UP-signalen innehåller positiva pulser när frekvensen på oscillatoren är lägre än Read Datas och på Down kommer negativa pulser när förhållandet är det omvända. Pulserna är kopplade till + ingången på en komparatorkrets, vilken även har en filterkrets inkopplad. Genom att lägga fram positiva pulser från UP-utgången, kan man öka spänningen på + ingången vilket gör att även utgångens spänningsnivå ökar. Detta innebär i sin tur att frekvensen ökar på oscillatoren. Genom att pulser sänds ut på DOWN-utgången kommer vi att sänka spänningen på + ingången och utspänningen sjunker och därmed oscillator frekvensen.

Frekvensen på RCLK-signalen vid enkel densitet ska vara 120 kHz och vid dubbel densitet 240 kHz, med oscillatoren frisvängande. Frekvensen justeras med trimkondensatorn till stift 3 och 4 på MC 4024.

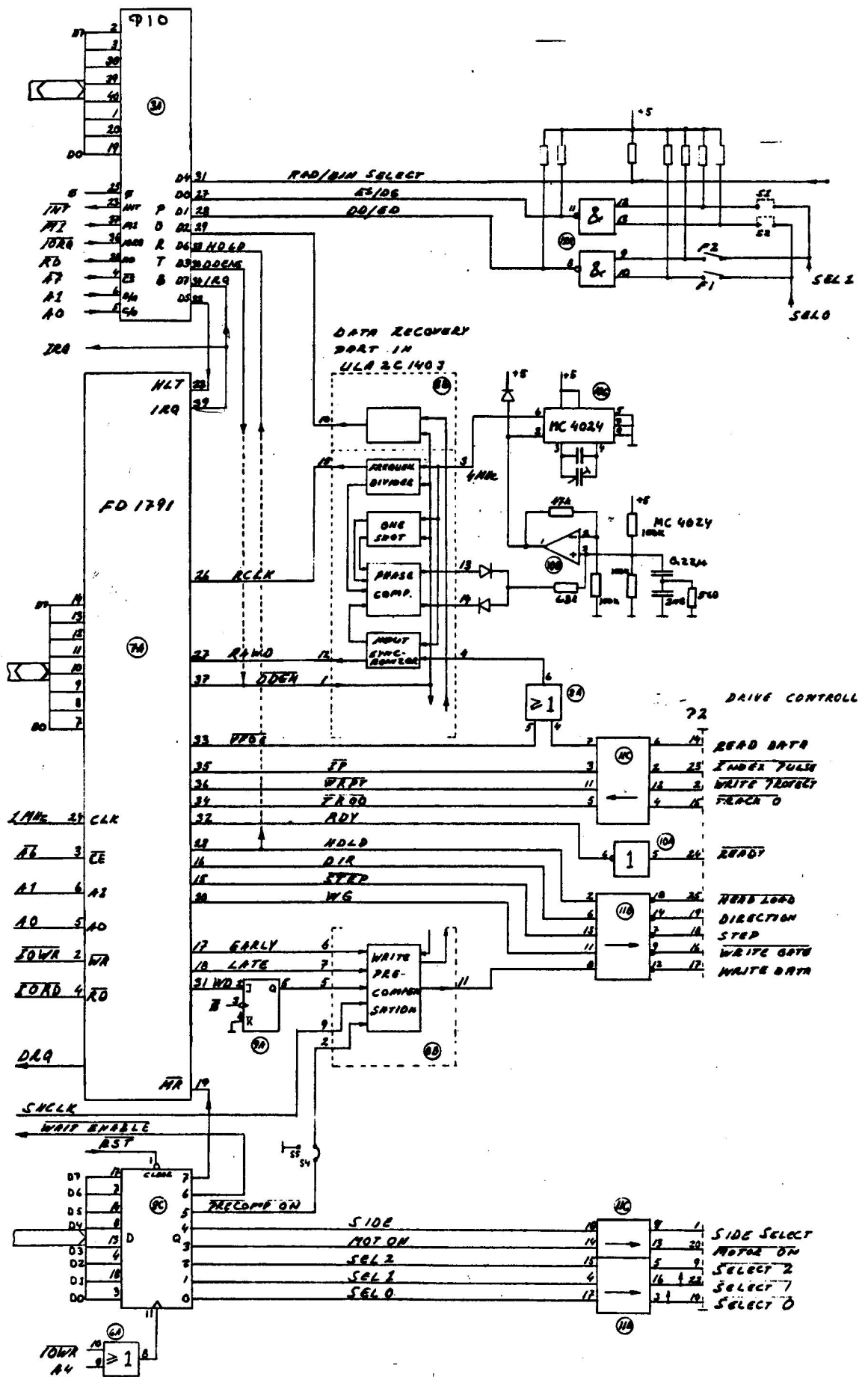


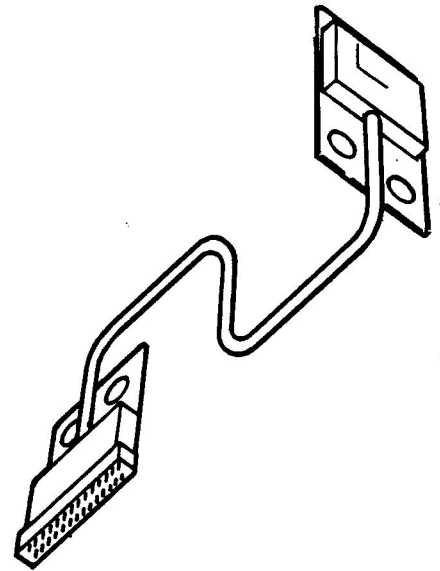
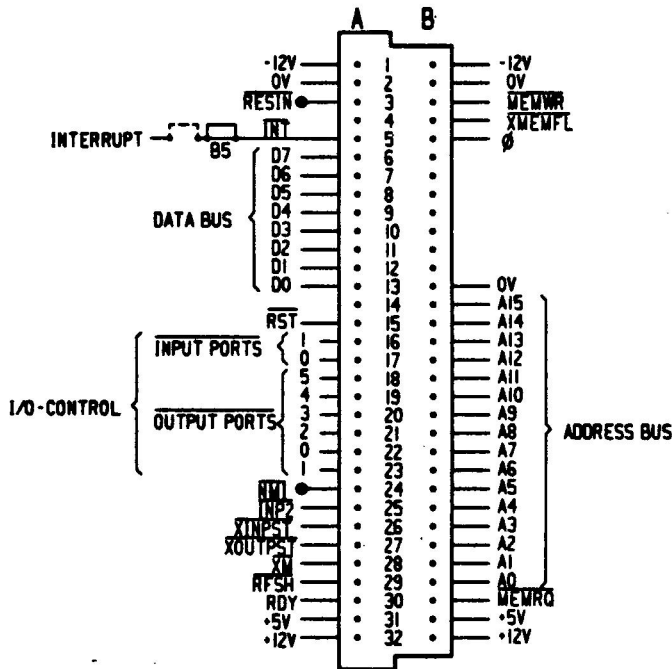
Fig 6.6 Anpassning mot drivenheterna

7. KONTAKTER OCH KABLAR

BUSSKONTAKT ABC 890

BUSSKABEL ABC 890

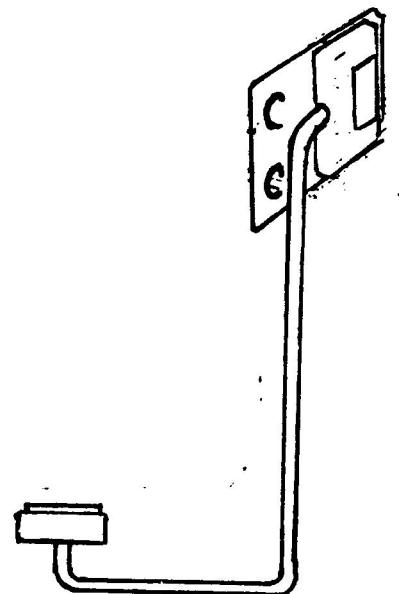
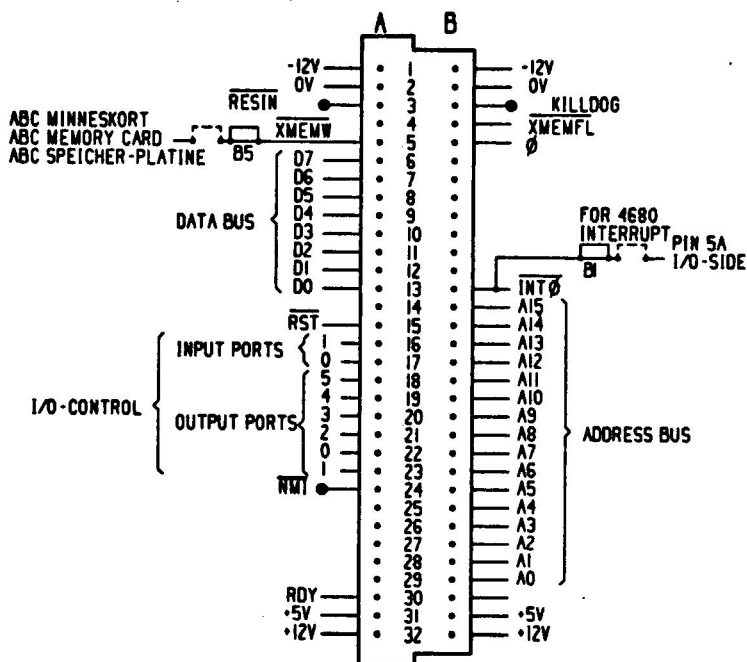
ART. NR 43 71419-01



BUSSKONTAKT ABC 80

BUSSKABEL ABC 80

ART. NR 43 71355-01



5. MINNESKORT

Minneskortet behöver bara användas i ABC 80-system med flexskiveenhet ABC.

Minneskortet ansluts till expansionsenhetens minnessida. Kortet har byglingar för basadress och för ROM/EPROM-typ. Kortet är förbyglat till basadress 24 k och EPROM-typ 2708. Minnesarean från 24 k till 28 k upptas av en ROM-krets (2332) innehållande DOS. Från 28 k till 31 k kan tre 1 kbyte EPROM anslutas. Både data- och adressledningarna är buffrade för att ge ökad drivförmåga. Observera att då vissa typer av EPROM kräver -5 V, finns det en serieregulator (79 M05) på minneskortet.

BLOCKSCHEMA

