

MIKRODATORSYSTEM MED INTEL 8080
FÖR REGLERAPPLIKATIONER

PER-OLOF SJÖBERG

Inst. för Reglerteknik
Lunds Tekniska Högskola
Oktober 1976

MIKRODATORSYSTEM MED INTEL 8080
FÖR REGLERAPPLIKATIONER

Examensarbete av Per-Olof Sjöberg

Handledare: Leif Andersson

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Oktober 1976

INNEHÅLLSFÖRTECKNING

1.1.1	Abstract - sammanfattning.
2.1.1	Inledning, beskrivning.
3.1.1	Instruktionstider.
4.1.1	CPU-kortet.
4.2.1	SINGLE STEP - STOP - RUN.
4.3.1	Avbrottsfunktionen.
4.4.1	HOLD-DMA.
4.5.1	LED-display.
4.6.1	Reset.
5.1.1	Minnesdesign.
5.2.1	Navelenheten.
5.3.1	PROM-kortet.
5.4.1	RAM-kortet.
6.1.1	A/D omvandlaren.
7.1.1	D/A omvandlaren.
8.1.1	Testprogram.
9.1.1	Kostnader.
10.1.1	Användarinstruktion.
11.1.1	Referenser.

RITNINGAR

R 2.1	64-pol kontaktdon till rack.
R 3.1, R 3.2	Pulsscheman.
R 4.1, L 4.1	CPU-kortet.
R 5.1, L 5.1	Navelenheten.
R 5.2, L 5.2	PROM-kortet.
R 5.3, L 5.3	RAM-kortet.
R 6.1, L 6.1	A/D omvandlaren.
R 7.1, L 7.1	D/A omvandlaren.

ABSTRACT

This report describes a possible solution for how to use the Intel 8080 micro-processor in a microcomputersystem.

The basis for the design has been to create a hardware suitable for the studying of the implementations of different kinds of regulation algorithms on a microcomputer, especially self tuning regulators. This has been achieved by a "general purpose" design which also makes other microcomputer applications possible. In order to communicate with the 8080-system a microcomputer - Intel 8008 - is used.

SAMMANFATTNING

I detta examensarbete presenteras ett sätt på vilket en microdator kan byggas upp med utgångspunkt från Intels CPU-krets 8080.

Förutsättningarna för utformningen har varit att skapa hårdvara passande för studier av implementeringen av olika styralgoritmer på microdator, speciellt självinställande regulatorer. Detta har uppnåtts med en "generell" design som även tillåter andra microdatorapplikationer.

För kommunikation med 8080-systemet används en microdator - Intel 8008.

INLEDNING

8080 microdatoren är uppbyggd kring intelkretsarna 8080 (processorn), 8224 (clockkretsen) och 8228 (controllern).

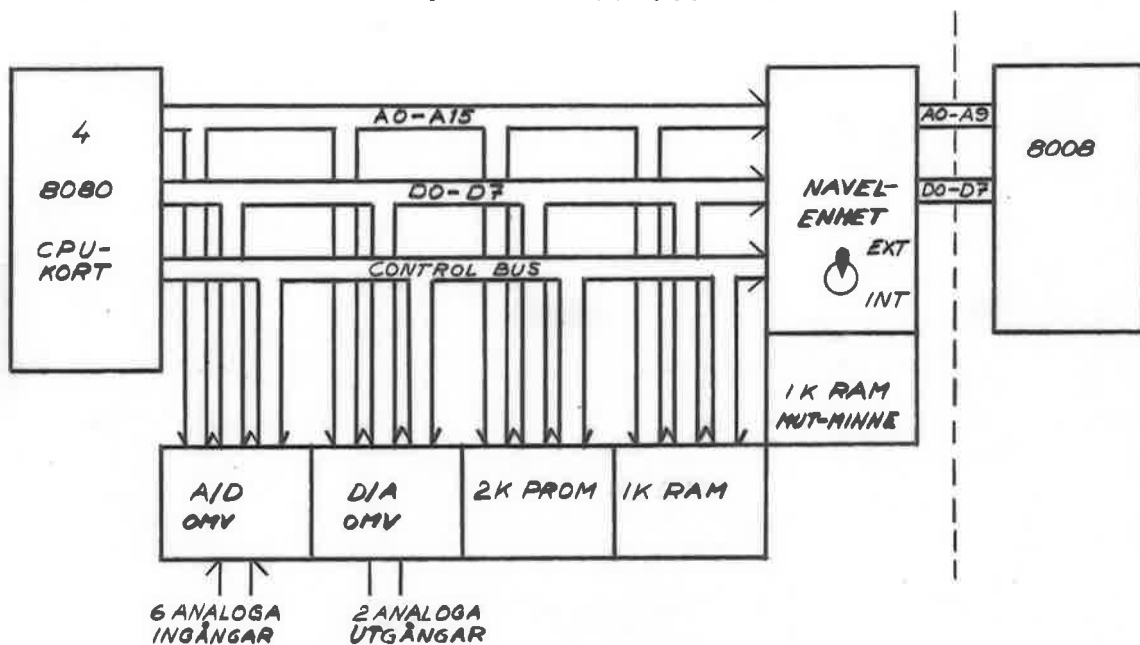
Logikkretsarna i kringelektroniken är hämtade ur TTL-familjens 74-serie.

I kapitel 5.2 och 10.1.1 beskrivs hur en intel 8008 (transintro) microdator användes för kommunikation med 8080-systemet.

Eftersom 8080-datorn är uppbyggd på ett antal kretskort som kan placeras i ett rack, kan systemet byggas ut eller modifieras efter användarens önskemål och krav.

Beskrivning av systemet

Blockschemat visar systemets uppbyggnad.



Varje block till vänster om den streckade linjen representerar ett kretskort.

Systemet har: 16 bitars adressbuss
 8 bitars databuss
 DMA-threestate buffrade adress- och databussar
 1 avbrottsnivå
 SINGLE STEP - stegvis exekvering
 9 MHz oscillatorfrekvens \Rightarrow 1 state - } μ s
 minneskapasitet 2 K PROM, 1 K RAM *
 6 analoga ingångar, 0 - 10V
 2 analoga utgångar, 0 - 10V

* Systemet utbyggbart till sammanlagt 64 K.

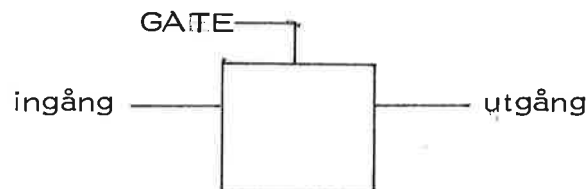
2.1.2

Adressbussen har dimensionerats för att kunna belastas med upp till 16 mA, dvs 10 st unit loads.

Databussen. Några av de enheter som skall leverera information till databussen har låg drivkapacitet (ca 1,9 mA). Därför skall databussen belastas varsamt, vilket bäst görs med CMOS-kretsar.

DMA. Direct Memory Access kan göras eftersom data- och adressbussen är threestate-buffrade. En threestate komponent har den egenskapen att dess utgångar kan anta tre tillstånd, nämligen: 1. logisk "nolla" 2. logisk "etta" 3. högimpedansläge.

Symbolen för en threestate-buffert visas i fig nedan:

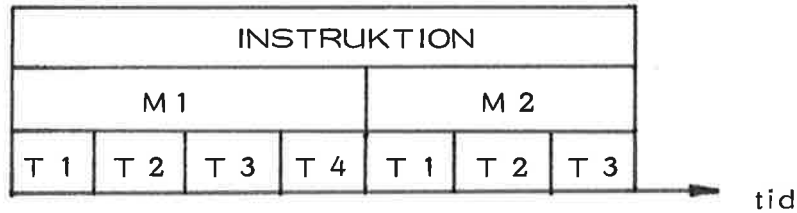


Minneskapaciteten är 2 K PROM (Programmable Read Only Memory) och 1 K RAM (Random Access Memory, dvs läs- och skrivminne). På 8008 microdatorn finns en PROM-programmerare avsedd för PROM-kretsarna 8702. Beskrivning finns i ref /4/.

3.1.1

INSTRUKTIONSTIDER - I/O-instruktioner.

En 8080 assemblerinstruktion är uppdelad i 1 - 5 st maskininstruktioner. Varje maskininstruktion är i sin tur uppdelad i 3 - 5 states. Clockpulsoscillatorn i 8224 styrs av en kristall som ger en clockpulsfrekvens ($\phi 1$) på 1 MHz. En state tar då 1 us. Nedanstående figur visar en fiktiv instruktion som innehåller två maskincykler.



Av speciellt stort intresse är timingen för in- och utinstruktionernas (dvs IN och OUT) tredje maskincykel.

Ritn R 3.1 och R 3.2 visar en sammanställning över dels ovanstående pulser, dels en del pulser som kan tänkas ingå i en "basic instruction" (jfr fig 2.5 i /1/). Det bör observeras att det i figurerna ej har tagits hänsyn till pulsernas stig- och falltider.

CPU-KORTET

CPU-kortet är uppbyggt enl fig 4.1

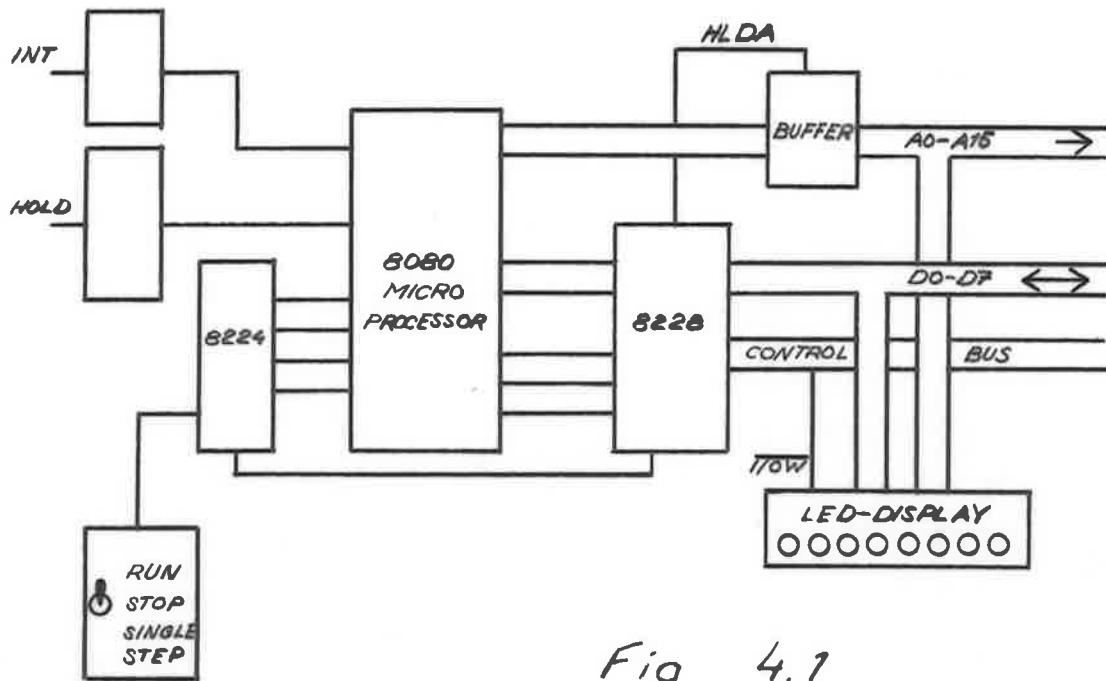


Fig 4.1

Adressbussen A0 - A15 (där A0 är LSB) har en three-state output. Adressbussens ordlängd gör det möjligt att adressera antingen maximalt 64 K bytes minne eller 256 in-enheter och 256 ut-enheter (i det senare fallet utnyttjas endast de 8 minst signifikanta bitarna i adressen). Drivkapaciteten för processorns adressbitar är 1.9 mA. För att höja denna buffras bussen med 74125-TTL-three-state buffertkretsar, vilka ger en fan out som är tillräcklig för att driva 10 st TTL unit loads (dvs ca 16 mA).

"Gatarna" (dessa är inverterade) på buffertarna är anslutna till HLDA på 8080 (via buffertarna E1-2, E3-4, E5-6). Då HOLD-REQUEST-signalen går låg, går HLDA hög och lägger 74125-buffertarna i högimpedansläge, vilket innebär att kontrollen över adressbussen kan övertas av en yttre enhet. Detta gör DMA (Direct Memory Access) möjligt.

4.1.2

Databussen har en ordlängd på 8 bitar. Ett ord kallas ett byte. Bussen går från 8080 till kontrollkretsen 8228, vilken buffrar dubbelriktat (three-state). Databussen har som primär uppgift att ta hand om informationsflödet till och från processorn. Dessutom bär databussen statusinformation, dvs den information som anger processorns tillstånd. Genom att avkoda databussen i 8228 kan man därför tillsammans med kontrollsignaler från 8080 och 8224 få fram styrsignaler som styr systemets yttre enheter. Dessa avkodade signaler utgör kontrollbussen.

SINGLE STEP - STOP - RUN

Genom att lägga processorns READY-ingång låg kan man få CPU:n att anta ett wait-tillstånd. Detta gör det möjligt att medelst extern elektronik få processorn att t.ex stoppa i en TW-state eller på ett manuellt kommando exekvera precis en maskincykel.

På CPU-kortets frontpanel sitter en switch som är märkt:

RUN
STOP
SINGLE STEP

Vid normal "kontinuerlig" exekvering skall switchen stå i läge RUN. Då switchen sätts i STOP avbryts exekveringen. I läget SINGLE STEP exekveras en maskincykel. Switchen är återfjädrande från SINGLE STEP till STOP.

Nedanstående figur visar switchkopplingen.

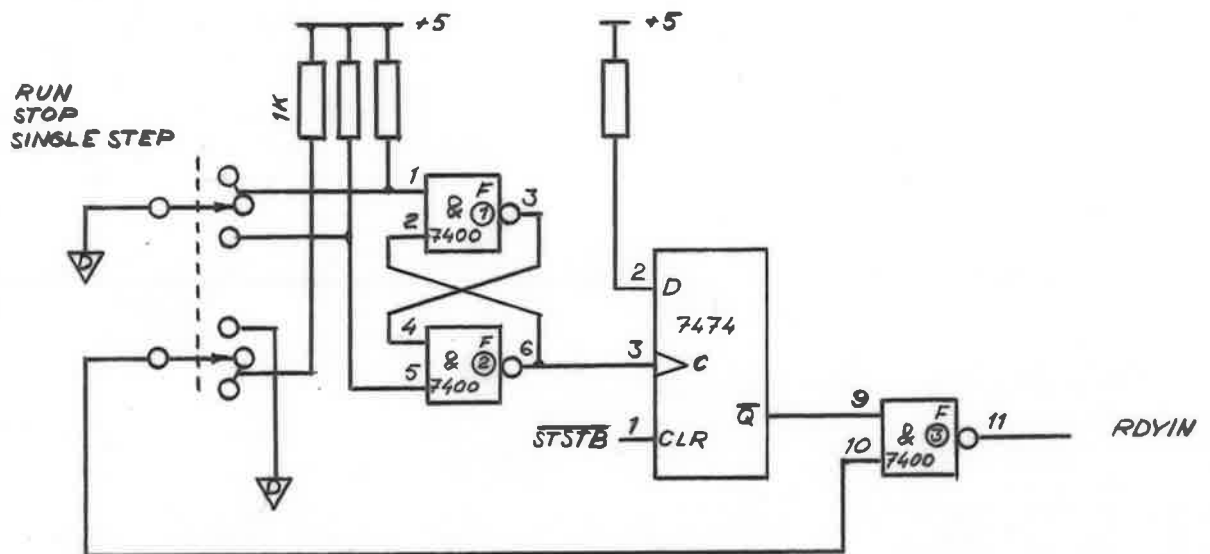


Fig 4.2

Vippan som består av F① och F② är till för att ge ett snabbt distinkt omslag, och för att eliminera kontaktstudsar.

RDYIN-signalen som fås på kopplingens utgång kommer att upp-träda assynkront och måste därför "timas". Detta görs i clock-kretsen 8224, där RDYIN synkroniseras och därvid bildar READY-signalen som styr processorn. Fig 4.3 visar hur RDYIN kommer att se ut.

Funktion:

Antag att switchen står i mittläge (= STOP)

Tillstånden är då:

- utgången på F① = 1
- utgången på F② = 0
- utgången på F③ = 0

RDYIN (=F③) är alltså låg och processorns arbete är stoppat. Då switchen slås ned till SINGLE STEP-läget kommer F① - F② vippan att kantra över och lägga F②:s utgång på en logisk etta. Denna signal går till clock-ingången på D-vippan, vilken triggas på positiv flank. En logisk etta (D-ingången kopplas via motstånd till +5V) kommer alltså att slussas ut till D-vippans Q-utgång. \overline{Q} -utgången läggs då på en nolla. Denna signalen går in på NAND-grinden F③:s ena ingång vilket betyder att F③:s utgång lägger sig hög. RDYIN-signalen ligger nu hög och tar 8080 ut ur wait-tillståndet. Eftersom ett wait-tillstånd alltid kommer efter T2 i en maskincykel, kommer nu CPU:n att fortsätta sitt arbete i T3.

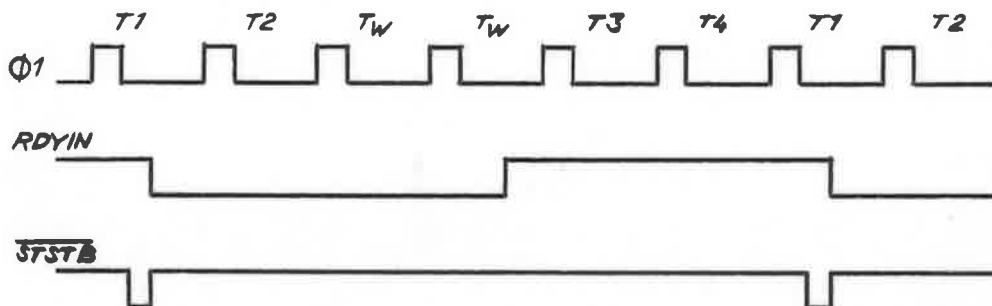


Fig 4.3

Den resterande delen av den påbörjade maskincykeln avslutas och exekveringen fortsätter med state T1 i nästa maskincykel. Då T1 påbörjas ger \overline{STSTB} -utgången på 8224 en kort puls som styr CLR-ingången på D-vippan. Båda utgångarna på D-vippan ett-ställes nu och speciellt medför ett-ställningen på \overline{Q} att F③:s utgång går till 0. RDYIN blir noll och 8080 intar åter ett wait-tillstånd efter nästkommande T2.

Exekveringen av den aktuella maskincykeln från TW till T2 tar endast 4 - 5 us medan SINGLE STEP-switchen hålles i SINGLE STEP-läget i storleksordningen sekunder (beroende på operatörens fingerflinkhet). Detta skulle kunna ge en felaktig RDYIN-signal då

4.2.3

SINGLE STEP-switchen återgår till STOP-läget, men eftersom F① - F② -vippan ger en negativ flank till D-vippan, kommer dennas tillstånd ej att förändras.

I switchens tredje läge -RUN- kommer ingång 10 på NAND-grunden F③ att ligga på logisk nolla. Detta lägger RDYIN på ett-nivå, och medför att processorn exekverar kontinuerligt.

AVBROTTSFUNKTIONEN - 1 AVBROTTSNIVÅ

Genom att $\overline{\text{INTA}}$ på 8228 är ansluten till +12V via ett 1 kohms motstånd, får man ett komplett avbrottssystem med en avbrottsnivå. Då avbrott begärs och accepteras av processorn kommer en RST7 instruktion att exekveras, vilket betyder att programräknarens innehåll sparas i stacken, varefter avbrottsrutinen startar i cell 70_g. Om registerinnehåll eller flaggstatus skall sparas, måste detta göras av avbrottsrutinen.

Avbrott kan göras antingen manuellt - med int-knappen på CPU-kortets frontpanel - eller automatiskt genom att en pulsgenerator kopplas till $\overline{\text{INT}}$ -ingången.

Fig 4:5 visar kopplingen som handhar avbrotten, medan fig 4:6 visar vilket krav som ställs på INT-pulsen (dvs den puls som processorn känner).

I CPU:n finns en interrupt enable (INTE) - vippa, vars tillstånd avgör huruvida ett avbrott skall accepteras eller ej. Denna vippa kan styras programmässigt med instruktionerna:

```

EI          . Interrupt enable (= set "INTE"-vippan.
            . Avbrott kan accepteras.)

DI          . Disable interrupt (= reset "INTE"-vippan.
            . Avbrott tillåts ej.)

```

En reset-beordring går till INTE-vippan då ett avbrott accepteras. (INTE-vippans tillstånd finns som utsignal från CPU:n).

Med ett interrupt är det möjligt att få processorn att lämna ett halttillstånd. Ett program som skall upprepas periodiskt kan det därför vara lämpligt att förse med följande "ram":

```

(START)      DI
              - - -
              - - - Program
              - - -
              EI
              HLT

```

Då programmet har genomlöpts kommer processorn att stanna i ett halttillstånd. En ny exekvering startas med ett interrupt. Eftersom inga avbrott tillåts mellan instruktionerna DI och EI kan man låta cykeltiden för interrupt-signalen understiga exekverings-

tiden för programmet. Om interruptfrekvensen då är tillräckligt hög kommer exekveringstiden att bestämma den högsta möjliga samplingshastigheten.

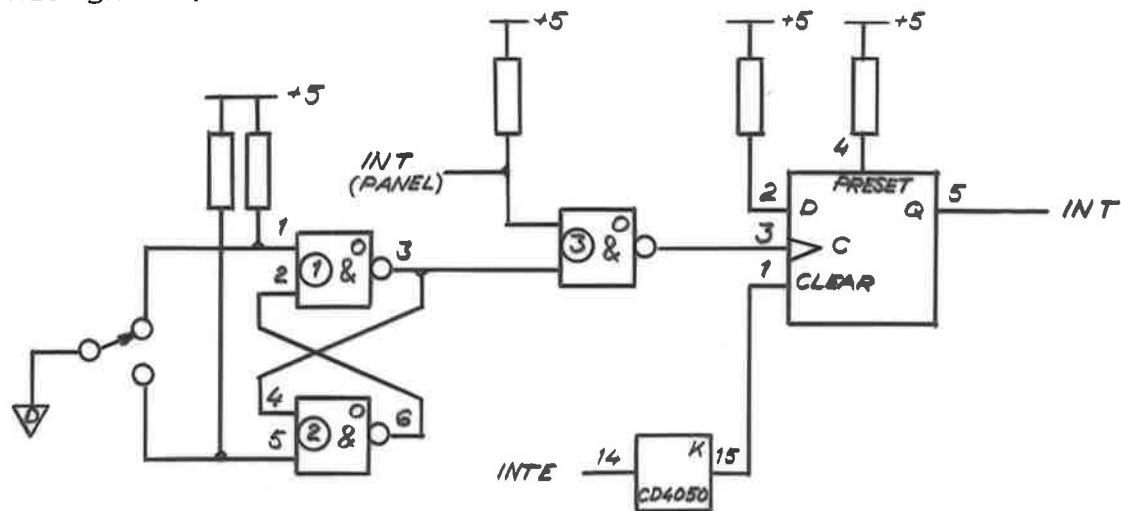


Fig 4.5

Normalt står O① - O② -vippans utgång i 1-läge. På ingångarna till NAND-grinden O③ finns då ett och en logisk nolla ligger på D-vippans clockingång. Om INT-knappen trycks in eller INT-ingången jordas, kommer en positivt gående puls att uppträda på O③ utgången. Då D-vippan triggar på positiv flank, kommer D-ingångens "letta" att slussas till Q-utgången, som anger att ett avbrott begärs. Om nu interrupt-vippan är nollställd, kommer avbrottet att accepteras. INT-vippan ställs och avbrottet tillkännages genom att INTE-signalen går låg. Eftersom avbrottet har accepterats kan D-vippan återställas, vilket görs med INTE-signalen. INTE-signalen buffras med hjälp av CMOS-bufferten K14-15.

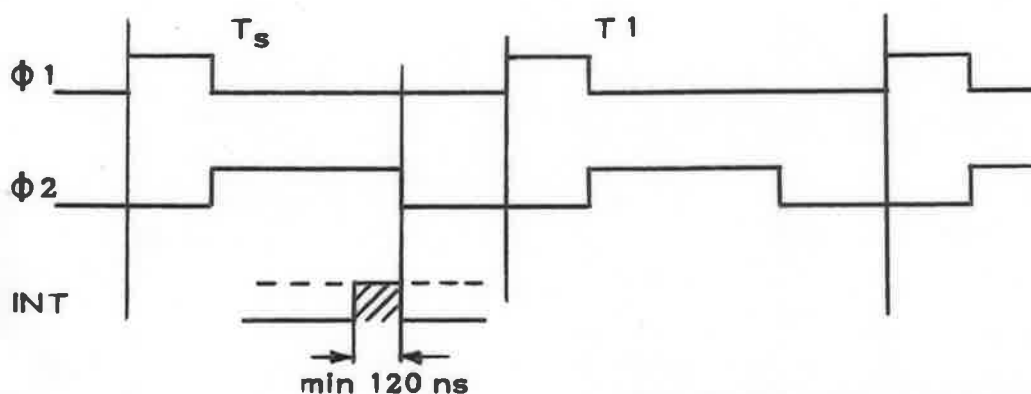


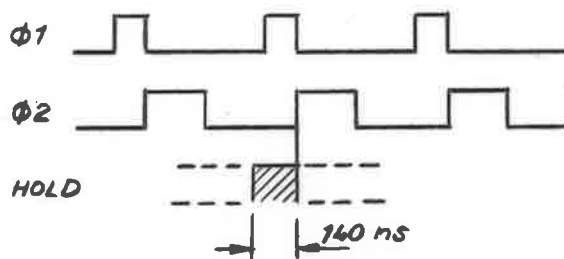
Fig 4.6 T_s betyder sista state i en godtycklig instruktion. För att avbrottet skall behandlas fördras att INT-signalen är hög i minst 120 ns före nedåtgående flank på $\Phi 2$.

HOLD-DMA

En HOLD-request till 8080 innebär att processorn avslutar den pågående exekveringen och lägger adress och databuss i three-state högimpedansläge. En HOLD-request accepteras om något av följande villkor är uppfyllt:

1. CPU:n är i halt-tillstånd
2. CPU:n är i T2 eller TW och READY-signalen är hög.

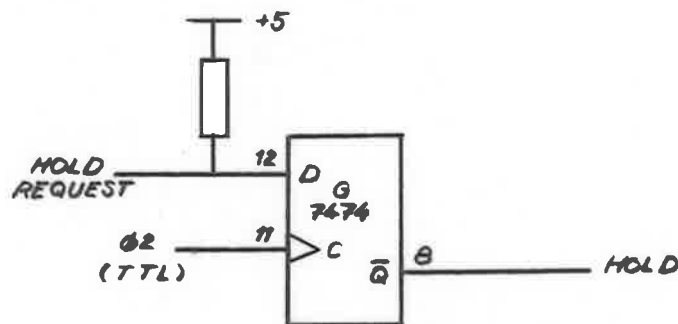
HOLD-signalen synkroniseras internt i CPU:n. Det krav som ställs på HOLD är att den är stabilt ett-ställd minst 140 ns före $\phi 2$ -pulsens början i antingen T2 eller TW.



HOLD-signalen skall vara hög så länge som HOLD-mode önskas.

HOLD-request signalen (pinne 60 i racket) går till en D-vippa enligt fig nedan. HOLD-request signalen är aktiv låg. En request medför att en nolla slussas in på D-vippan. \bar{Q} -utgången blir då hög, och efter ett antal clockcykler kommer processorn att inta HOLD-mode.

Då HOLD-request åter går hög, kommer HOLD-signalen att bli låg, och processorn lämnar sitt HOLD-tillstånd.



4.4.2

Att processorn antagit HOLD-mode tillkännages genom utsignalen HLDA. HLDA buffras med E1-2, E3-4, E5-6 varefter den får gå till gatarna på adressbussens buffertkretsar. I HOLD-mode är HLDA aktiv hög vilket medför att threestate buffertarnas "gatar" måste vara inverterade. HLDA styr även kontrollkretsens datautgångar.

Observera att inget av villkoren 1 och 2 är uppfyllt då CPU-kortets switch står i STOP (READY-signalen är låg). Därför måste SINGLE STEP göras efter en "hold request".

LED^{*)}-DISPLAY.

CPU-kortets frontpanel är försedd med en display bestående av 8 lysdioder. På denna display presenteras ackumulatorns innehåll då OUT 377 exekverats. Varje lysdiod representerar en bit på databussen, och är kopplade enl följande figur:

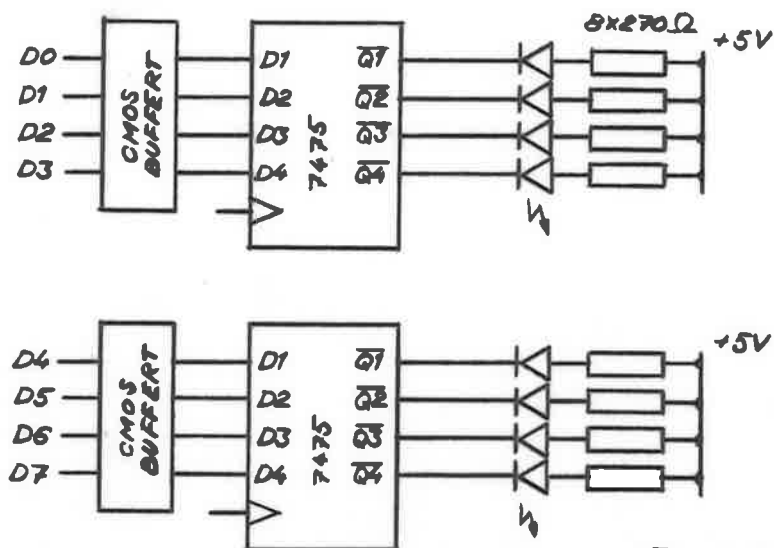


Fig 4.9

Instruktionen OUT upptar i minnet två konsekutiva celler enl nedanstående programsekvens i maskinkod:

```

1    11 010 011 (= 323g)
2    11 111 111 (= 377g)

```

Byte nummer ett är instruktionsdelen medan byte 2 anger adressen till den avsedda OUTPUT-porten, vilken i det här fallet är 377.

Då instruktionen exekveras händer följande:

1. Ackumulatorns innehåll lägges ut på databussen.
2. Adressen till den avsedda OUTPUT-porten (här adr 377) kommer att läggas ut på adressbussen.
3. Ett $\overline{I/O}$ write commando skickas ut (aktiv låg).

Adressbussens 8 minst signifikanta bitar avkodas tillsammans med $\overline{I/O}$ writesignalen och en positiv puls fås som clockar in databussens innehåll på en 8 bitars latch. Latchens utgångar driver direkt de 8 lysdioderna på displayen.

*) Light Emission Diode

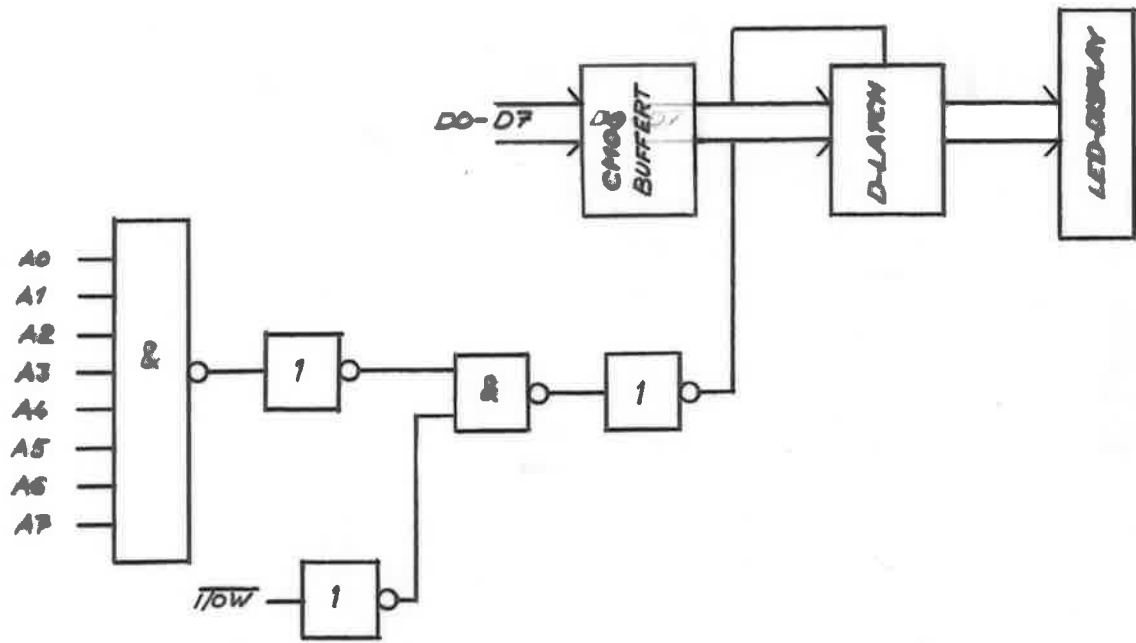
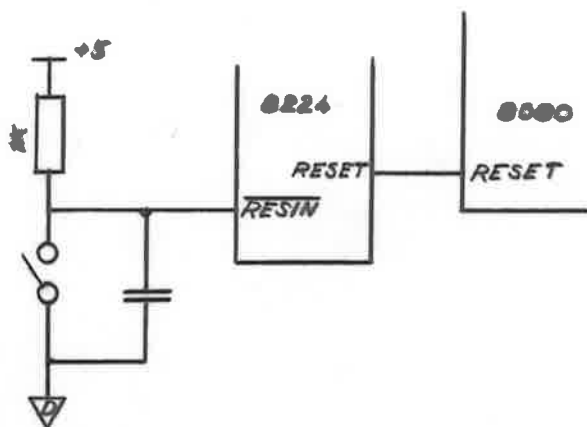


Fig 4.10

CMOS bufferten som driver LATCHEN gör att databussen ej belastas med mer än högst någon μA .

RESET

Då RESET-knappen på CPU-kortets frontpanel trycks in, kommer programräknaren att nollställas. När knappen släpps upp startar processorn att exekvera från cell 000 bank 0. För att CPU:n skall hinna avsluta aktuell instruktion fordras att RESET-signalen är minst tre clockpulsintervaller lång. Synkronisering mellan \overline{RESIN} -signalen och RESET-signalen sker i 8224. Följande fig visar kopplingen:



Observera att varken flaggor, ackumulator, stackpekare eller register nollställs vid RESET.

MINNESDESIGN

Minnesarean innehåller 2 K PROM och 0,5 K RAM. En typisk minnesreferensinstruktion visas i följande figur:

LDA-adr	00	111	010
	LOW ORDER ADDR		
	HIGH ORDER ADDR		

LOW ORDER ADDR (LOA) och HIGH ORDER ADDR (HOA) läggs ut på adressbitarna A0 - A7 resp A8 - A15.

Med LOA kan man alltså adressera 256_{10} ($=400_8$) ord. Därför är det lämpligt att tänka sig en uppdelning av minnet i banker, där varje bank innehåller 400_8 ord. HOA kommer då att ange banknumret som alltså skulle kunna omfatta banker från 000 till 377_8 . Nedanstående figur visar minnesareans utseende:

000 000	PROM	- PROM-kortet
:		
:		
007 377		
010 000	RAM	- RAM-kortet
:		
:		
013 377		

Då man testar program (tex subrutiner) behövs en minnesarea där man kan lägga sina program och där man enkelt kan göra ändringar. I samma area skall 8080-CPU:n kunna exekvera. För att klara av detta finns det en navelenhet som innehåller 1 K RAM (kallat MUT-minne). Då navelenheten är inkopplad skall PROM-kortets 4 lägsta banker kopplas bort och ersättas med MUT-minnet.

Fortfarande kan dock PROM-kortet (adress 004 000 - 007 377) användas liksom RAM-kortet (adress 010 000 - 013 000).

NAVELKORTET (figur på sid 5. 2. 3)

Navelkortet är en kommunikationsenhet som gör det möjligt för både 8080 och 8008 att få access till MUT-minnets 4 bankers RAM. På navelenhetens frontpanel sitter en switch märkt EXT-INT. I läget INT kan 8080-CPU:n exekvera i MUT-minnet, vars innehåll då utgör bankerna 0 - 3. EXT-läget innebär i princip att DMA görs i 8080-systemet varvid det möjliggörs att genom transintron skriva och läsa i MUT-minnet.

Adressbuss, databuss och styrsignalerna $\overline{\text{MEMW}}$, $\overline{\text{CE}}$ från 8008 är anslutna till navelkortet via en "navelsträng". I "8008-änden" av navelsträngen sitter en anordning som passar in i de chassi kontakter som normalt är avsedda för 8008:ans egna minneskort. Om kontakten placeras i PCH-hacket, kommer MUT-minnet att adresseras som bankerna 30 - 33.

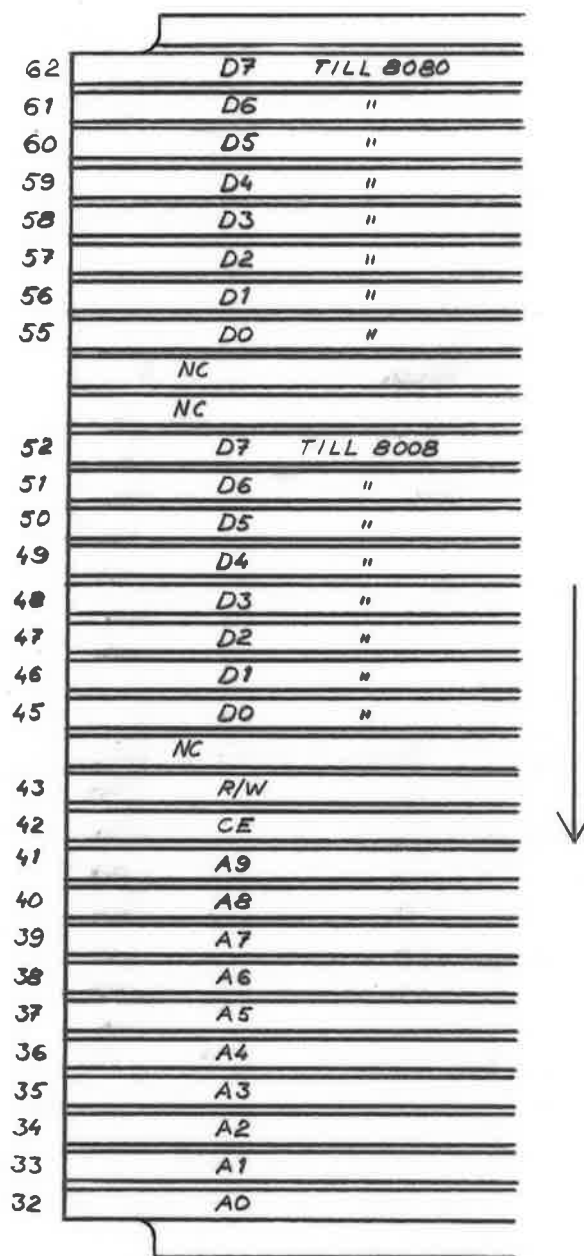
$M\textcircled{3}$ och $\overline{\text{CE}}_{8008}$ (från 8008) går till en AND-funktionskoppling som består av två open collector kretsar ($I\textcircled{1}$ och $I\textcircled{2}$) med ett gemensamt collectormotstånd. AND-kopplingens utgång är ansluten till $\overline{\text{CE}}$ (aktiv låg) på MUT-minnet. $M\textcircled{3}$ styr gatarna på ABC och CZ. Dessa buffertar kommer alltså att vara öppna i EXT-läget. Då 8008 vill läsa ur minnet sänder den ut en $\overline{\text{CE}}_{8008}$ -puls och en adress, vilket får MUT-minnets datautgång att lämna information som via HI-bufferten går till transintron. (OBS! FG-bufferten är stängd). $\overline{\text{MEMW}}_{8008}$ styr R/W-ingången på minnet. Det betyder att då man vill skriva i MUT-minnet sänds $\overline{\text{CE}}_{8008}$ och $\overline{\text{MEMW}}_{8008}$ ut från 8008. Data går då via DE buffertar och skrivs in i minnet.

Då 8080 skall ha access till MUT-minnet skall switchen stå i läget INT. $M\textcircled{3}$ och $M\textcircled{4}$ får då tillstånden 0 resp 1. Buffertarna ABC, DE och $C\textcircled{2}$ kommer då att läggas i högimpedansläge, medan $C\textcircled{1}$ öppnas av $M\textcircled{4}$. AND-kopplingen $I\textcircled{1}$ - $I\textcircled{2}$:s utgång ligger hela tiden låg och gör MUT-minnet "enable".

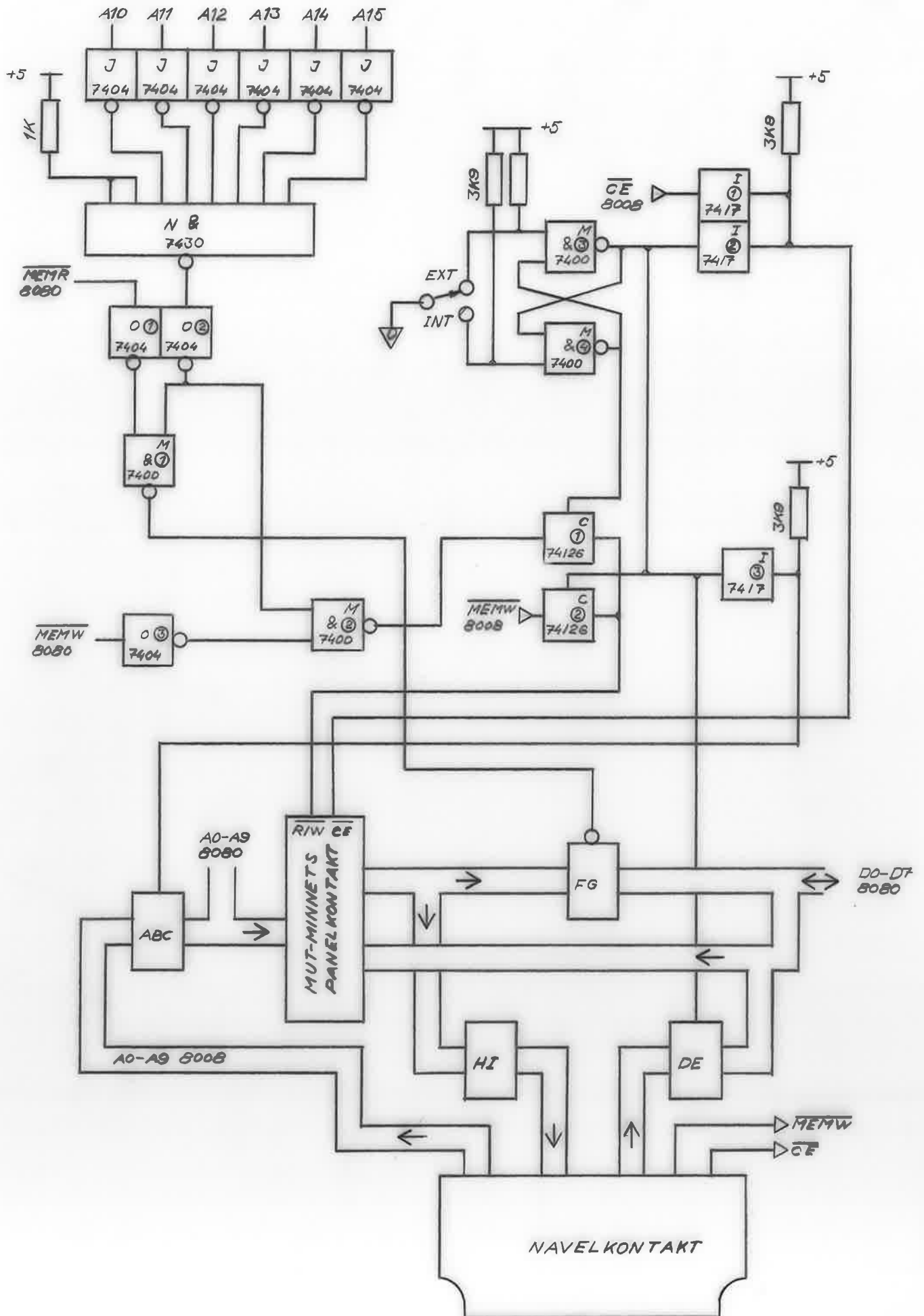
MUT-kortet är alltså aktiverat och adressbitarna A0 - A10 ligger hela tiden "till". På MUT-minnets utgångar uppträder därför alltid information. Datautgångarna är emellertid skilda från 8080-systemets databus genom FG-bufferten. FG-bufferten skall öppnas vid läsning i någon av bankerna 0 - 3. Därför decodas adressbitarna i A10 - A15 tillsammans med $\overline{\text{MEMR}}$ och ger på $M\textcircled{1}$:s utgång en aktiv låg utsignal. Därvid öppnas FG och informationen från MUT-minnet går ut på 8080-databussen.

5.2.2

Vid skrivning i minnet skall R/W- ingången få en aktiv låg skrivpuls. A0 - A9 adresserar det önskade ordet i minnet, medan skrivpulsen erhålles då A10 - A15 och $\overline{\text{MEMW}}$ decodas och ger en 0-puls på M②:s utgång.



NAVELKONTAKTEN



PROM-KORTET

PROM-kortet har 2 K ords minnsutrymme, dvs 8 banker. Varje bank utgörs av en 8702-2 MOS-krets. Fig 5.1 visar en principskiss över kortets uppbyggnad.

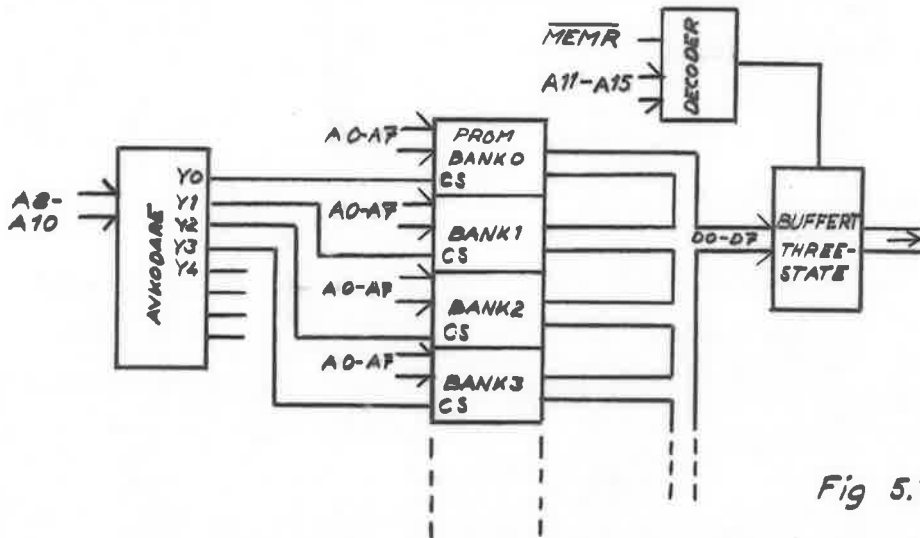


Fig 5.1

Adressbitarna A0 - A7 går direkt till minneskapslarna, medan bitarna A8 - A10 går till en avkodare (74138) som avgör vilken av de 8 kretsarna som skall aktiveras, dvs vilken krets som skall få sin \overline{CE} -ingång låg. Ett problem uppstår eftersom \overline{CE} styrs enbart av adressbussen. Man kommer att få en utsignal från någon av kretsarna även om man inte gör en minnesreferens till PROM-kortet. En OUT- eller IN-instruktion ger samma fel. Problemet har lösts genom att minneskretsarna har fått en gemensam three-state databussbuffert. Gatarna på buffertarna (aktiva låga) styrs av en decoder vars utgång går låg då A11 - A15 är låga samtidigt med \overline{MEMR} .

Då PROM-kortets banker 0 - 3 skall ersättas av navelkortet (och MUT-minnet) måste en modifiering göras. I normala fall är ingång nr 11 på N-kapseln ansluten till +5V via ett motstånd. Då navelenheten används skall istället ingång 11 anslutas till A10 på adressbussen.

RAM-KORTET. (ritning nr 5.3)

1 K RAM finns i bankerna 10 - 13. Varje bank består av två parallellt kopplade 8111-2 (8111-2 är ett 256x4-organiserat RAM). Det är möjligt att läsa eller skriva i RAM endast då \overline{CE} är låg på något av RAM-kretsaren. Med adressbitarna A11 - A15 adresseras RAM-kortet. Som resultat av denna dekodning får man ut en aktiv låg signal (B-8), som är en enable-signal till 3/8 avkodaren C(74138). De 8 utgångarna på C går till \overline{CE} -ingångarna på var sin minnesbank. Eftersom RAM-kortet endast har 1 K minne (4 banker), behöver 4 av avkodarutgångarna användas. Kortet är alltså lätt utbyggbart till 2 K RAM.

OD (=Output Disable) och R/W (=Read/Write) ingångarna är aktivt låga. Därför kan de direkt styras av \overline{MEMR} resp \overline{MEMW} . Fig 5.2 visar läs- och skrivcykelspecifikationer för 8111-2.

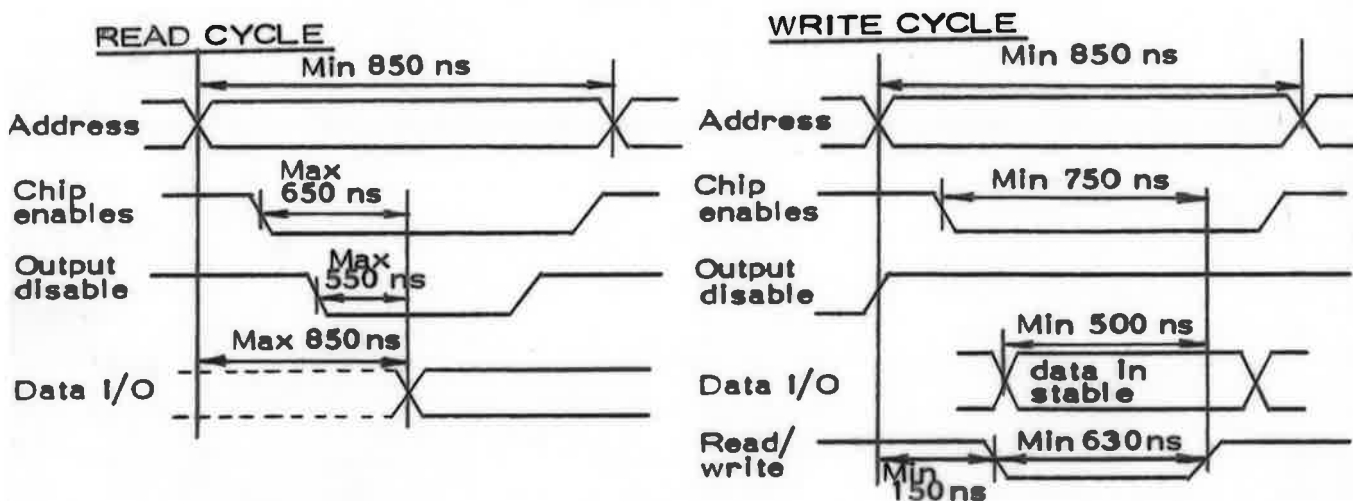
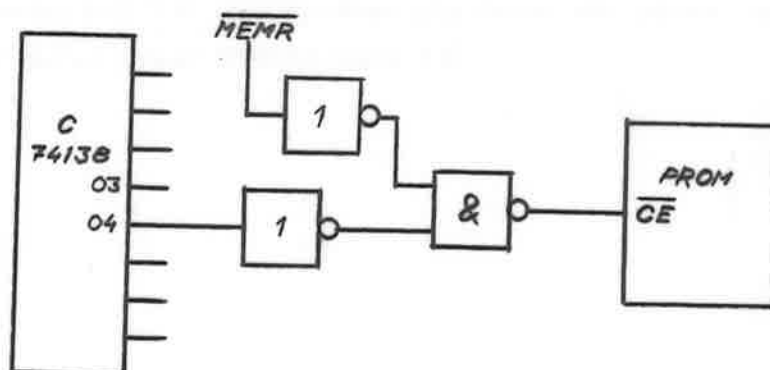


Fig 5.2

Om man skulle vilja utnyttja det nu icke använda utrymmet på RAM-kortet till 1 K PROM låter sig detta lätt göras, som fig 5.3 visar (figuren visar PROM-bank 14).



A/D-OMVANDLAREN

A/D-omvandlaren har 6 ingångar. Var och en av dessa är via en emitterföljare (återkopplad OP-förstärkare) och ett RC-filter kopplat till en multiplexer. RC-filtret har en tidskonstant på ca 5 ms, vilket gör att signalen in på omvandlaren varierar maximalt 1% under omvandlingstiden.

Med multiplexern väljes en av de 8 kanalerna. Multiplexerns utgång är ansluten till A/D-omvandlaren via en följare.

I A/D-omvandlaren konverteras den utvalda analoga signalen till ett 8-bitars digitalt ord, som kan läsas in i processorns accumulatorregister.

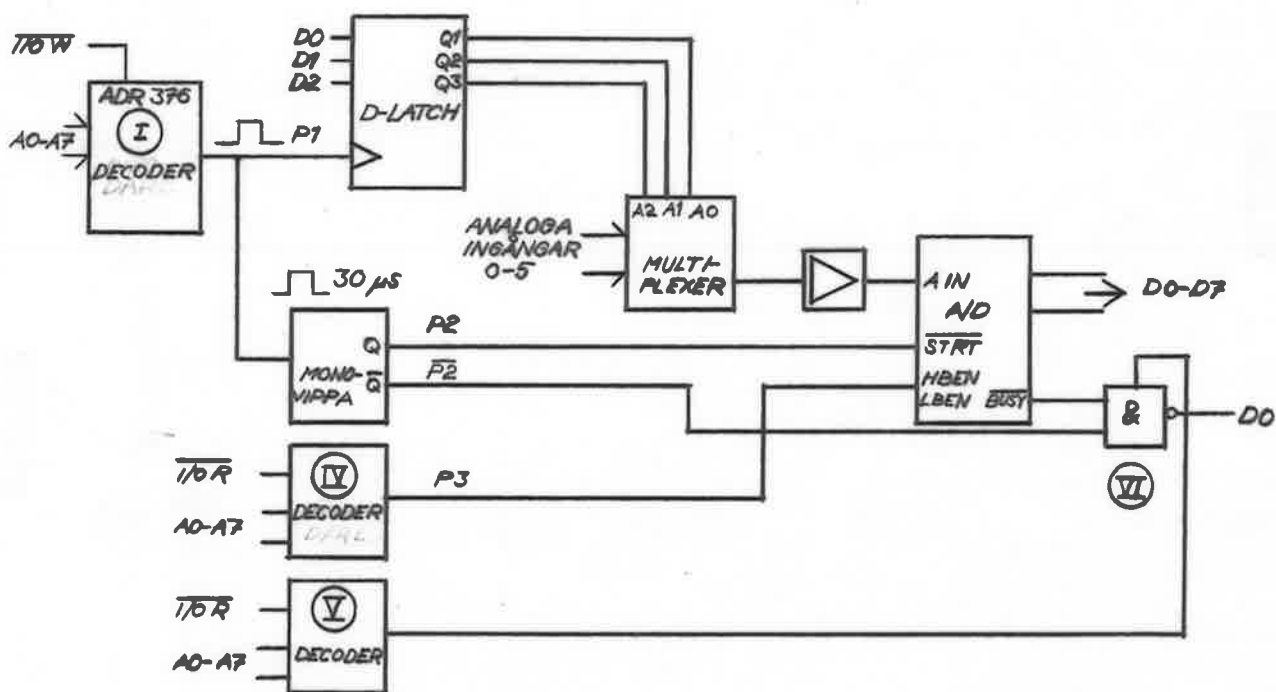


Fig 6.1

En konvertering startar man i sitt program med kommandot OUT 376.

I instruktionens tredje maskencykel skickas då ut:

1. Adressen 376
2. Accumulatorinnehållet till databussen
3. En $\overline{I/O\overline{W}}$ -puls

Adressen och $\overline{I/O\overline{W}}$ -pulsens decodas i I, och som resultat fås en aktiv hög puls P1. Pulsens skickas in på D-latchens clockingång, och slussar därvid över databussens (accumulators) innehåll till multiplexerns ingångar A0-A2. Därmed är den analoga ingångs-

kanalen vald. Vid val av multiplexerkanal användes endast accumulatorns tre minst signifikanta bitar enl följande:

Databussen	Kanal
00 000 000	0
00 000 001	1
00 000 010	2
00 000 011	3
00 000 100	4
00 000 101	5

Efter ett antal us finns den förstärkta och filtrerade analoga insignalen på A/D-omvandlarens ingång.

Monovippan triggas av pulsen P1. En ca 30 μ s lång 1-puls - P2 - fås, vilken går till $\overline{\text{STRT}}$ på A/D convertern. $\overline{\text{STRT}}$ triggas på nedåtgående flank, vilket betyder att konvertering börjar ca 30 μ s efter det att den har beordrats. $\overline{\text{BUSY}}$ -signalen talar om A/D-converters status:

- 1 betyder omvandling färdig
- 0 betyder räknearbete pågår.

En konvertering tar maximalt en 100 us. För att undersöka om konverteringen är färdig skickar man ut instruktionen IN 377. Instruktionen andra byte avcodas tillsammans med $\overline{\text{I/OR}}$ i avcodare V. En positiv puls på dennas utgång får grinden VI:s status att lägga sig på D0. IN 377-instruktionen ombesörjer att D0 tas in i accumulatorn. Pulserna in till denna logiken ser ut enl nedan:

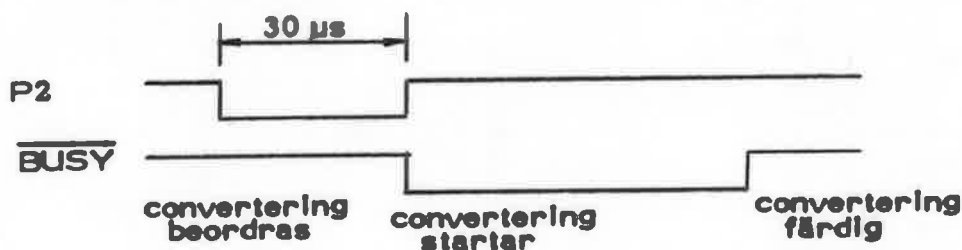


Fig 6.2

Då konverteringen är färdig skall alltså D0 vara 0. Efter det att en konvertering har konstaterats "färdig" skall A/D-omvandlarens utgångsvärden läggas ut på databussen. Detta görs med instruktionen IN 376. Adresserna och $\overline{\text{I/OR}}$ decodas i IV och pulsen P3 fås. Denna puls går till HBEN och LBEN vilka öppnar A/D-omvandlarens three-state utgångar till databussen. Databussens innehåll tas genom instruktionens försorg in i accumulatorn.

Detta ovan beskrivna är alltså en omvandlingscykel.

6.1.3

Rutinen som sköter en omvandling skall ha följande struktur:

```

                                MVI
                                OUT 376
LOOP                            IN 377
                                ANI 001
                                CPI 000
                                JZ UT
                                J LOOP
UT                               IN 376
```

A/D-omvandlaren skall ha en analog insignal 0 - 10V. Frekvenser högre än ca 1 kHz filtreras bort före converteringen och signalen omvandlas till ett 8-bits ord i binär offset, dvs en följande tabell:

9,98	11 111 111
5	10 000 000
0	00 000 000

således erhålles en upplösning på $\frac{10}{256}$ 0,039V. Detta är dock inte den effektiva upplösningen då man i oturliga fall kan få störningar på de analoga signalerna som kan uppgå till 50 - 100 mV.

Eftersom 8080-CPU räknar i 2-komplement måste ordet i binär offset omvandlas. Detta göres programmässigt med en rutin som inverterar MSB.

D/A-OMVANDLAREN

Ett 8 bitars ord i binär offset, omvandlas i D/A-omvandlaren till en analog signal 0 - 10V.

D/A-omvandlaren har två kanaler - kanal 0 och kanal 1 - vilka har adresserna 374 resp 375. Låt oss studera en konvertering i kanal 0.

Blockschema:

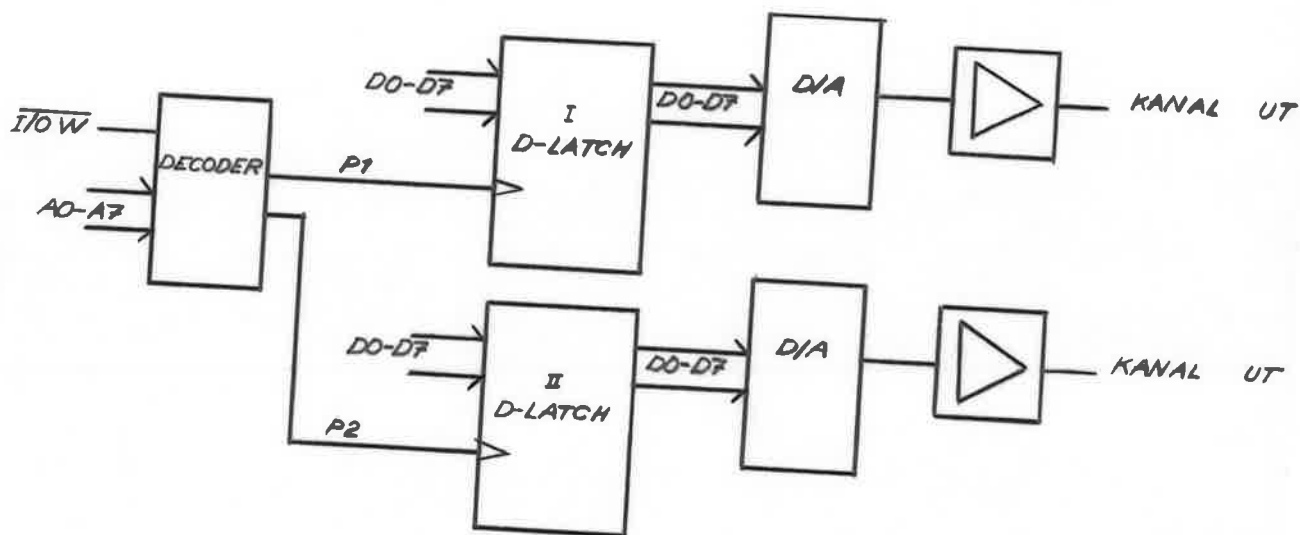


Fig 7.1

Man initierar en omvandling med instruktionen OUT 375. Instruktionens andra byte (dvs I/O portens adress) läggs ut på databussens 8 LSB och decodas tillsammans med $\overline{I/O W}$. En aktiv hög puls fås ut från decodern och går vidare till D-latch I:ts clockpuls-ingång. D-latchen triggas på pulsens främre flank, och slussar därvid över accumulatorns innehåll till Q-utgångarna. Detta 8-bits ordet går direkt vidare till D/A-omvandlaren, som efter ca $5 \mu s$ ger den önskade analoga utsignalen. Eftersom D/A-kretsen arbetar kontinuerligt och dess in-information bestäms av D-latchens utsignaler, kommer den analoga signalen att ligga konstant, ända tills nästa OUT 375 exekveras. Genom att D-latchens D-ingångar är buffrade med CMOS-buffertar, kommer databussen ej att belastas.

TESTPROGRAM

För att testa olika funktioner i systemet kan de program som är beskrivna nedan vara till hjälp. För enkelhets skull har programmen dels skrivits i INTEL 8080:s mnemokod dels i oktal kod.

Program 1 "GRUNDTTEST AV CPU- OCH MINNESKORTET (text
MUT-minnet)

Programmet gör att informationen i accumulator-registret räknas upp med ett för varje loop som penetreras. Efter varje gång genomgången loop stannar processorn i halttillstånd. Genom ett "interrupt" kommer processorn att fortsätta exekveringen från cell 70g.

Program:

0	XRA A	257	.nollställ accumulatorn A
1	INR A	074	.A + 1 → A
2	OUT 377	323	.lägg ut A:s innehåll på .LED display
3		377	
4	EI	373	
5	HLT	166	
70	JMP 1	303	.hoppa ovillkorligt till .cell 1
		001	
		000	

8.1.2

Program 2 TEST AV A/D- OCH D/A-OMVANDLARNA.

En signal in till A/D-omvandlaren omvandlas och tas in i accumulatorn, läggs ut på LED-displayer varefter den åter omvandlas tillbaka till analog signal i D/A-omvandlaren.

0	MVI	A, 000	076	. byte 2 anger A/D kanal
1			000	
2	OUT	376	323	. starta omvandling. MUX-
				. kanal
3			376	. enl A
4	IN	377	333	. test om conv färdig
5			377	
6	ANI	001	346	. nollställ bitarna 1 - 7
7			001	
10	CPI	000	376	. Z=1 om A=0
11			000	
12	JZ	30	312	
13			030	
14			000	
15	JNP	004	303	
16			004	
17			000	
30	IN	376	333	. läs A/D-omvandlaren
31			376	
32	OUT	377	323	. A → LED-display
33			377	
34	OUT	375	323	. A → D/A-omvandlaren
35			375	
36	JMPO		303	
37			000	
38			000	

Program 3. TEST AV RAM-KORTET.

20₈ st ord i bank 1 (med start från cell 0) överföres till RAM bank 10 (start cell 0). Därefter överföres de till bank 10 överförda orden till bank 3 (start cell 0).

Test av RAM-kortet:

0	MVI B, 000	006	.nollställ B
1		000	
2	LXI DE, 010 000	021	.010 →D 000 →E
3		000	
4		010	
5	LXI HL, 001 000	041	.001 → H 000 → L
6		000	
7		001	
10	MOV A, M	176	.((H), (L)) → A
11	XCHG	353	.(H) ↔ (D) (L) ↔ (E)
12	MOV M, A	167	.((H), (L)) ↔ A
13	INX DE	023	
14	XCHG	353	
15	INX DE	023	
16	INR B	004	
17	MOV A, B	170	.(B)→ A för test
20	CPI 020	376	
21		020	.A= 20?
22	JNZ 10	302	.om nej, hoppa till 10
23		010	
24		000	
25	NOP	000	.här börjar transfer (bank10)→ .→ bank 3
26	MVI B, 000	006	
27		000	
30	LXI DE, 003 000	021	
31		000	
32		003	
33	LXI HL, 010 000	041	
34		000	
35		010	
36	MOV A, M	176	
37	XCHG	353	
40	MOV M, A	167	
41	INX DE	023	
42	XCHG	353	
43	INX DE	023	
44	INR B	004	
45	MOV A, B	170	
46	CPI 020	376	
47		020	
50	JNZ 36	302	
51		036	
52		000	
53	OUT	323	
54		377	
55	HLT	166	

KOSTNADER FÖR HÅRDVARAN

Av olika anledningar är det svårt att ange ett definitivt pris på den enskilda komponenterna som 8080-systemet är uppbyggt av. Anledningarna kan vara grossistens specialerbjudanden, rabatter vid storinköp etc. Nedanstående sammanställning är en mycket grov skattning som baseras på katalogernas styckeprislistor, aktuella i februari 1976. Det är möjligt att få upp till 30 % rabatt på vissa komponenter vid stormängdsinköp. Kostnaderna för motstånd, vintråd osv utgör blott en ringa del av det totala priset för systemet, och redovisas därför ej. Kretskort + panel + europakontakt behandlas i klump som "chassi".

	<u>Krets</u>	<u>Antal</u>	<u>Pris/st</u>	<u>Total</u>
<u>CPU-KORT</u>				
TTL	7400	2	2:70	
	7404	1	3:--	
	7430	1	2:70	
	7474	2	4:30	
	7475	2	4:30	
	74125	4	5:70	
CMOS	4050	2	6:70	
	8080	1	372:--	
	8228	1	58:--	
	2224	1	52:--	
	CHASSI		70:--	
	SOCKLAR	17	5:--	
	LED	8	1:--	
	SWITCHAR	2	8:--	
				725:--
<u>NAVELKONTAKT</u>				
TTL	7400	1	2:70	
	7404	2	3:--	
	7430	1	2:70	
	74125	2	5:70	
	74126	5	5:70	
	7417	2	3:70	
	CHASSI		70:--	
	SOCKLAR	15	5:--	
				204:--
<u>PROM</u>				
TTL	7404	1	3:--	
	7430	1	2:70	
	7125	2	5:70	
	74138	1	8:--	
	CHASSI		70:--	
	SOCKLAR	13	5:--	
MINNE	8702	8	117:--	
				1 096:--

		trpt		2 025:--

	<u>Krets</u>	<u>Antal</u>	<u>Pris/st</u>	<u>Total</u>
			trpt	2 025:--
<u>RAM</u>				
TTL	7400	1	2:70	
	7404	2	3:--	
	7430	1	2:70	
	74138	1	8:--	
	CHASSI		70:--	
	SOCKLAR	15	5:--	
MINNE	8111-2	8	<u>24:--</u>	356:--
<u>D/A-OMVANDLARE</u>				
TTL	7400	1	2:70	
	7404	1	3:--	
	7430	1	2:70	
CMOS	4042	4	14:70	
	AD7520	2	70:--	
	uA741	3	4:--	
	CHASSI		70:--	
	SOCKLAR	10	<u>5:--</u>	339:--
<u>A/D-OMVANDLARE</u>				
TTL	7400	1	2:70	
	7404	2	3:--	
	7430	2	2:70	
	74121	1	8:--	
CMOS	4042	1	14:70	
	AD311	1	15:--	
	AD7501	1	100:--	
	AD7570	1	200:--	
	uA741	8	4:--	
	CHASSI		70:--	
	SOCKLAR	16	<u>5:--</u>	534:--
<u>Summa totalt</u>				<u>3 254:--</u>

ANVÄNDARINSTRUKTION

Mikrodatorsystemet kan köras i två moder:

1. Programtestningsmod, där det är möjligt att kommunicera med 8080-systemet med hjälp av en 8008.
2. Arbetsmod, där alla program ligger i PROM och ej kan nås av operatören.

I programtestningsmoden har både 8080 och 8008 access till en fyra bankers RAM-area , MUT-minnet. Skrivning och läsning kan göras i detta minne antingen med teletype eller genom 8008:ans switchregister. Tillgänglig minneskapacitet är 2K RAM och 1 K PROM med adresser enl följande figur:

000 000	MUT-RAM
:	1 K
:	
003 377	
004 000	PROM
:	1 K
:	
007 377	
010 000	RAM
:	
:	
013 000	

För programutveckling kan PDP-15 användas som host computer. Man stansar då först en remsa i 8080:s assemblerkod med hjälp av teletype. Remsan matas därefter in på PDP-15 där programmet kan editeras (och även lagras på bandminne). På PDP-15 utföres assemblering och laddning. Assemblern översätter programmet till maskinkod och laddaren länkar subrutiner. Därefter trycks resultatet ut på hålremsa. Med teletype kan remsan läsas in på MUT-minnet via 8008. Förfarandet finns beskrivet i /3/ (se ref!).

Färdiga, uttestade rutiner kan läggas på PROM, där de kan anropas från program i MUT-minnet (se ref. /4/)

GÖR SÅ HÄR!

1. Se till att CPU-kortet, NAVEL-enheten, PROM- och RAM-korten (ev också A/D och D/A-omvandlarna) finns på plats i racken. Anslut jordförbindelse mellan 8080 och 8008.
 2. Se till att ett av transintrons RAM-kort är placerat i panelkontakten på navelenheten och att navelkontakten sitter på rätt sätt i transintrons minnesdel (dvs kontaktens pilar skall vara riktade mot transintrons frontpanel. Läget PCO motsvarar bankerna 30 - 33).
 3. Slå till spänningen på 8008 och 8080.
 4. Sätt navelswitchen i EXT. Gör SINGLE STEP.
 5. Ställ in adressen 30 000 med knapparna. Kontrollera att det går att läsa och skriva i minnet!*)
 6. Skriv in programmet (max 1 K) i minnet antingen genom att
 - a) "knappa" in.
 - b) ladda med hjälp av remsläsare (TTY). Följande "uppstartning" skall då göras:
 - b:1) ställ TTY:s omkopplare i läge LINE
 - b:2) ställ in instruktionen RESTART 0 (005_g) på switchregistret.
 - b:3) tryck på RESTART
 - b:4) tryck på RUN och datorn skriver ut en asterisk var- efter den väntar på kommando från tangentbordet. *)
- INLÄSNING b:5) placera remsan i läsaren med omkopplaren i läge STOP.
- b:6) skriv kommandot LOADB: (stoppadr: startadr:), ställ omkopplaren i läge START och slå ned mellanstegstangenten.
- DUMPNING b:5) ge kommandot DUMP: med stopp- och startadress för oktalt utformat.
7. Exekvering. Ställ navelkortets switch i läge INT. Tryck på RESET. Exekvera antingen i RUN eller stega fram en maskininstruktion i taget med SINGLE STEP.
 8. Om ändringar eller tillägg skall göras i programmet
 - a) sätt CPU-switchen i läge STOP
 - b) sätt navelswitchen i läge EXT. Gör SINGLE STEP
 - c) gör ändringarna med transintrons knappsats.

*) Se beskrivning av 8008.

BESKRIVNING AV 8008

Frontpanelens utseende visas på nästa sida.

MEMORY ADDRESS och MEMORY DATA är LED-displayer som representerar adress resp data. Om navelkontakten placeras i läget PCO, kommer navelenhetens MUT-minneskort att representera bankerna 30 - 33 dvs adresserna 030 000 - 033 377.

Tryckknapparna omedelbart under switch-registret har följande funktion:

LOAD ADDRESS: Vid tryckning överförs innehållet i switch-registret till registret MEMORY ADDRESS vars innehåll visas med 14 lampor. MEMORY ADDRESS pekar ut en cell i minnet vars innehåll direkt visas på de 8 lamporna kallade MEMORY DATA.

LOAD MEMORY: Vid tryckning överföres innehållet i switch-registrets bitar 0 - 7 till den minnescell som utpekats av MEMORY ADDRESS. Det nya minnesinnehållet visas på lamporna MEMORY DATA.

STEP ADDRESS: Värdet på MEMORY ADDRESS ökas med ett vid tryckning på knappen. Innehållet i den nya adressen visas på MEMORY DATA.

Då teletype är ansluten till 8008 används nedanstående två kommandon, vilka skrives från tangentbordet:

1. Laddning av MUT-minnet:

LOADB: Detta kommando används för att läsa binära data från remsa till konsekutiva minnesceller. Start och stopp ges i oktalt form från tangentbordet.

ex: stoppadr. startadr.
LOADB: 030:020: 030:000:

2. Dumpning av MUT-minnets innehåll.

DUMP: Kommandot ger utskrift och stansning i oktalt form av minnesinnehållet mellan den start- och stoppadress som anges oktalt efter kommandot.

ex: stoppadr. startadr.
DUMP: 030:011: 030:000:

ger utskriften:

030 000 076: 000: 323: 377: 074: 303: 000: 035:
030 010 001: 003:

8192 4096 2048 1024 512 256 128 64 32 16 8 4 2 1

	8192	4096	2048	1024	512	256	128	64	32	16	8	4	2	1
MEMORY ADDRESS	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
MEMORY DATA							<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
SWITCH REGISTER	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>

RUN
 HALT
 STOP

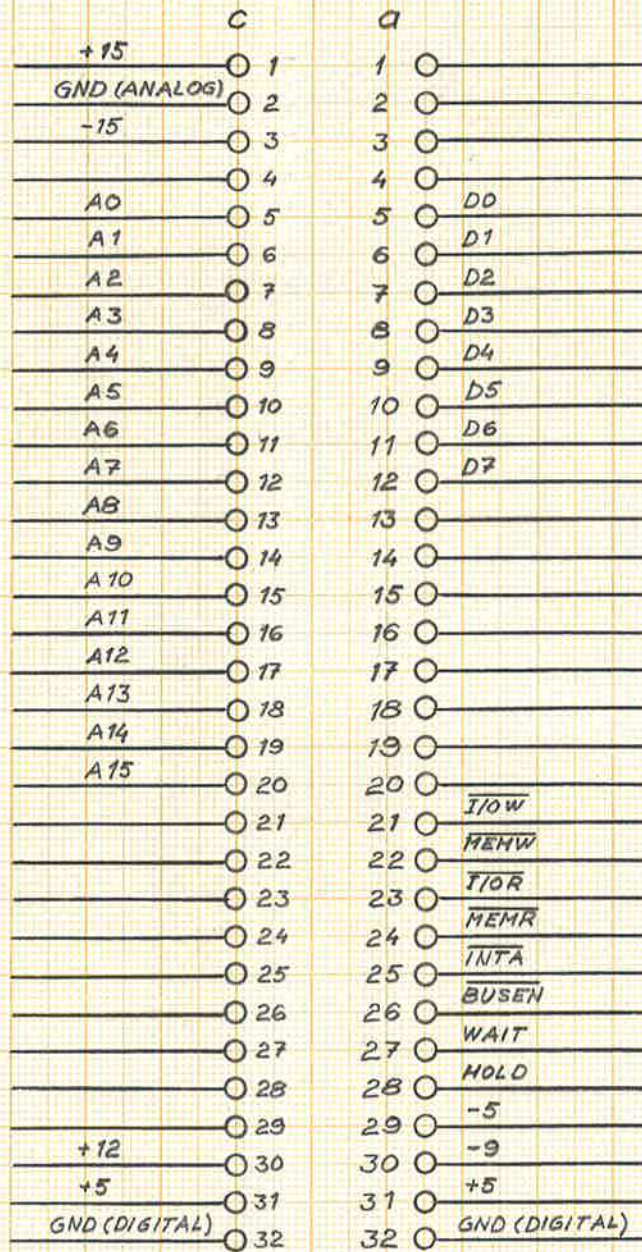
INT
 SINGLE CYCLE
 RESTART

LOAD MEMORY ADDRESS
 LOAD ADDRESS
 RUN
 STOP

STEP ADDRESS
 ADDRESS
 STOP

REFERENSER

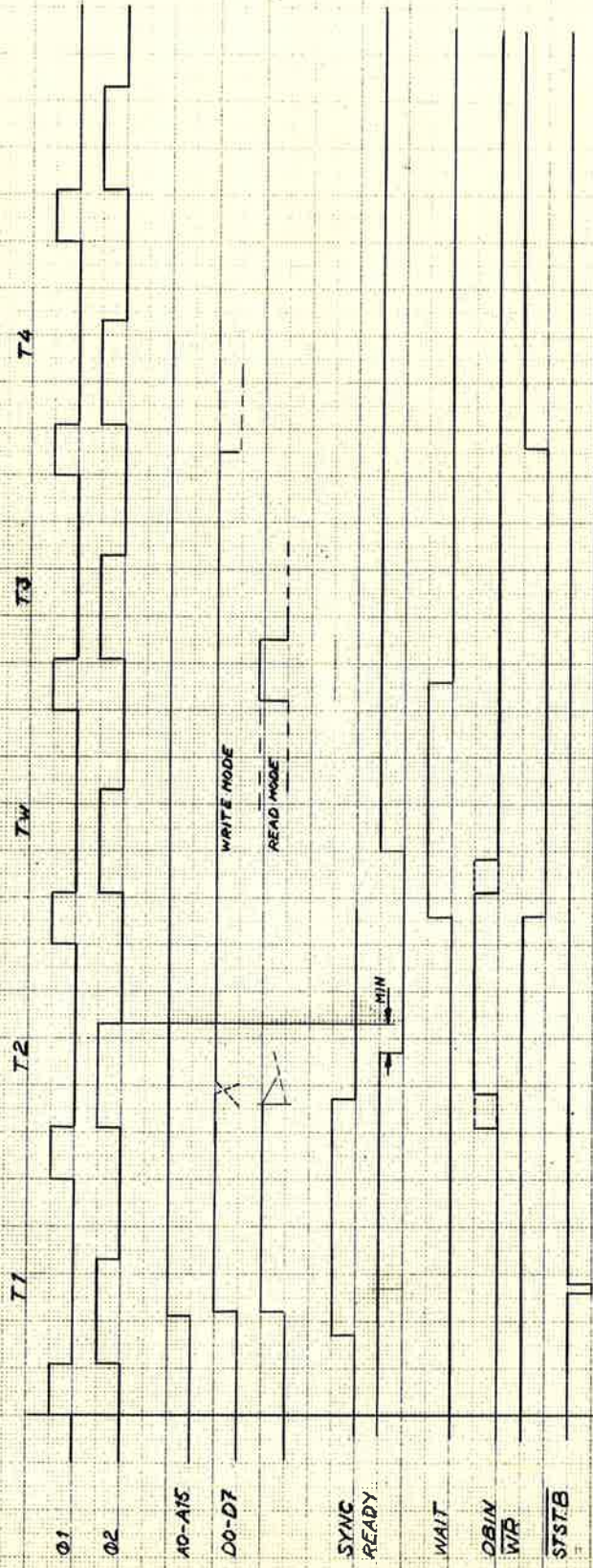
- /1/ Intel 8080 Microcomputer Systems User's Manual.
- /2/ George Kiziroglu: Implementation of a Self-Tuning Regulator on a Microprocessor, Department of Automatic control, LUND Institute of Technology.
- /3/ Leif Andersson: Cross Assembly and relocation of programs for the Intel microprocessors, using a PDP-15 as a host computer, Report 7602, Division of Automatic control, LUND Institute of Technology.
- /4/ TRANSINTRO, 7260 Transdata, Teknisk beskrivning.
- /5/ Hallberg-Danielsson: Digital teknik, Del 5-6



PULSSCHEMA , SKALA: 1RVTA=80ns

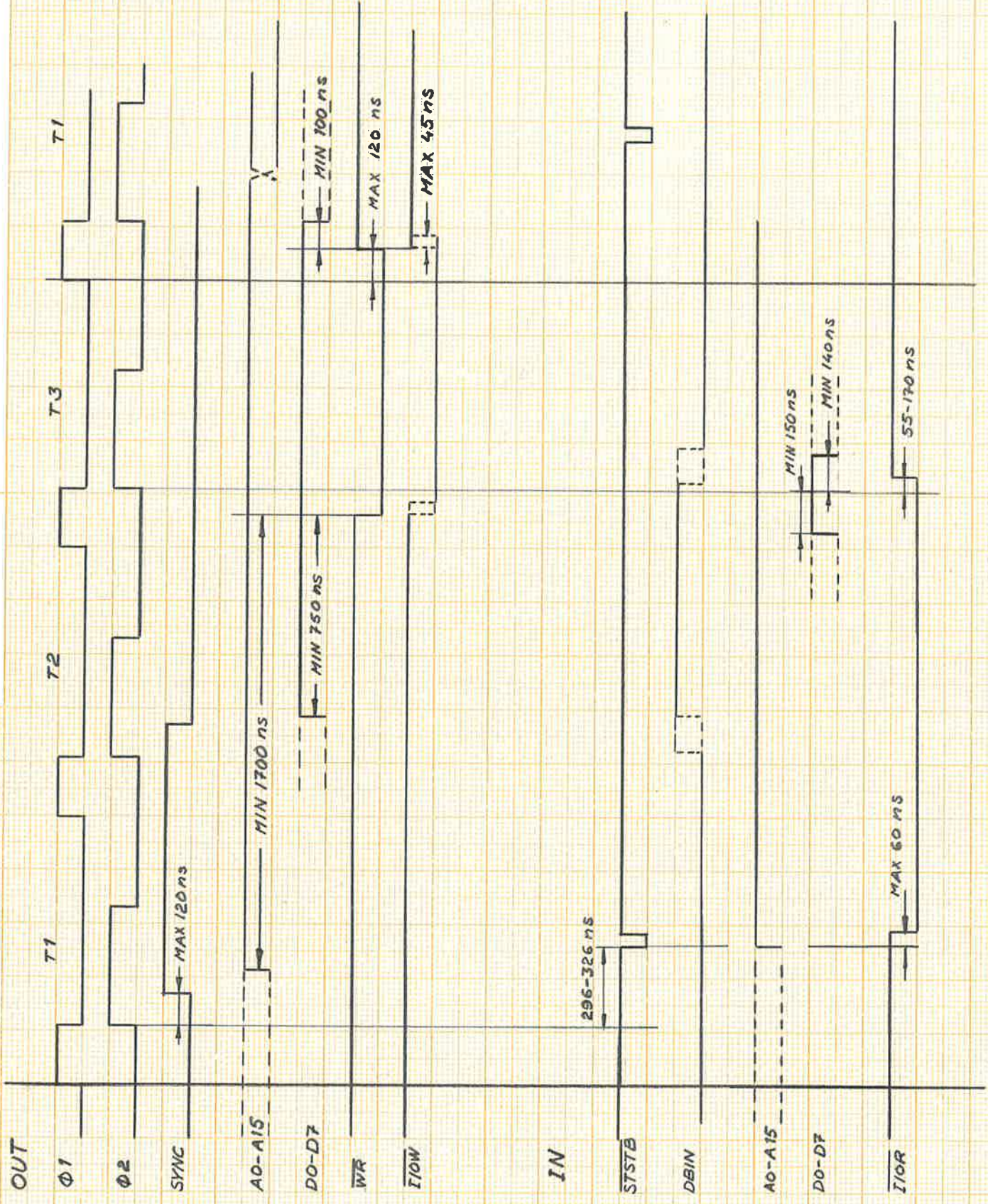
R3.1

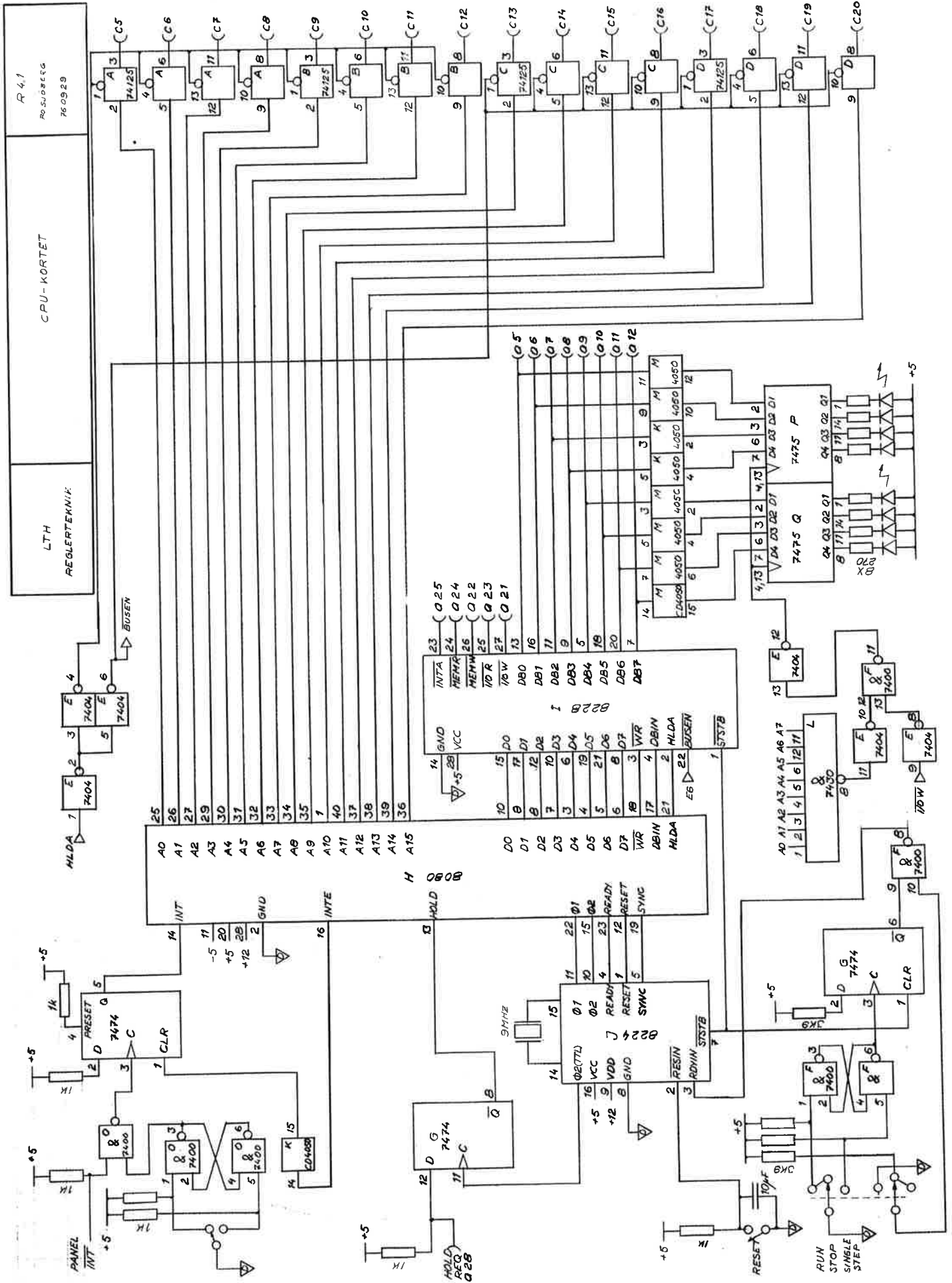
76 0929



R 3.2

Figuren visar OUT-
resp IN instruktionens
bredje maskinckel.
SKALA 1 RUTA = 20 NS

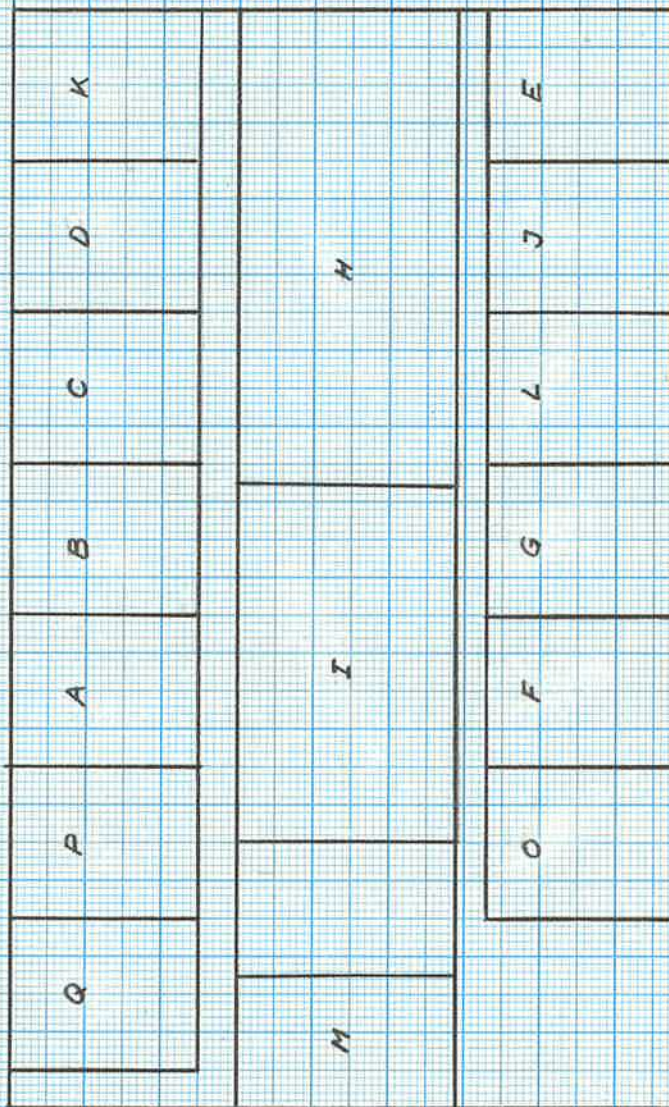


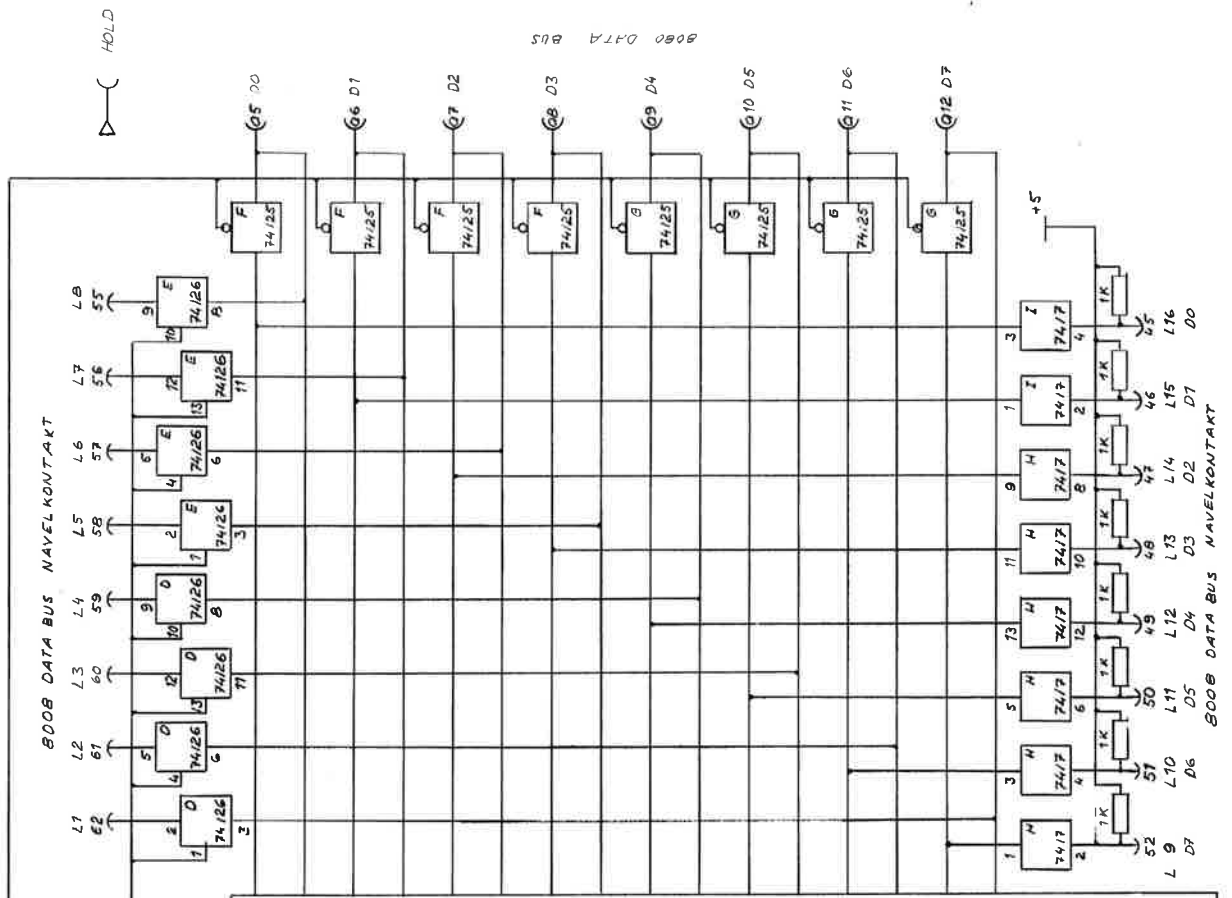
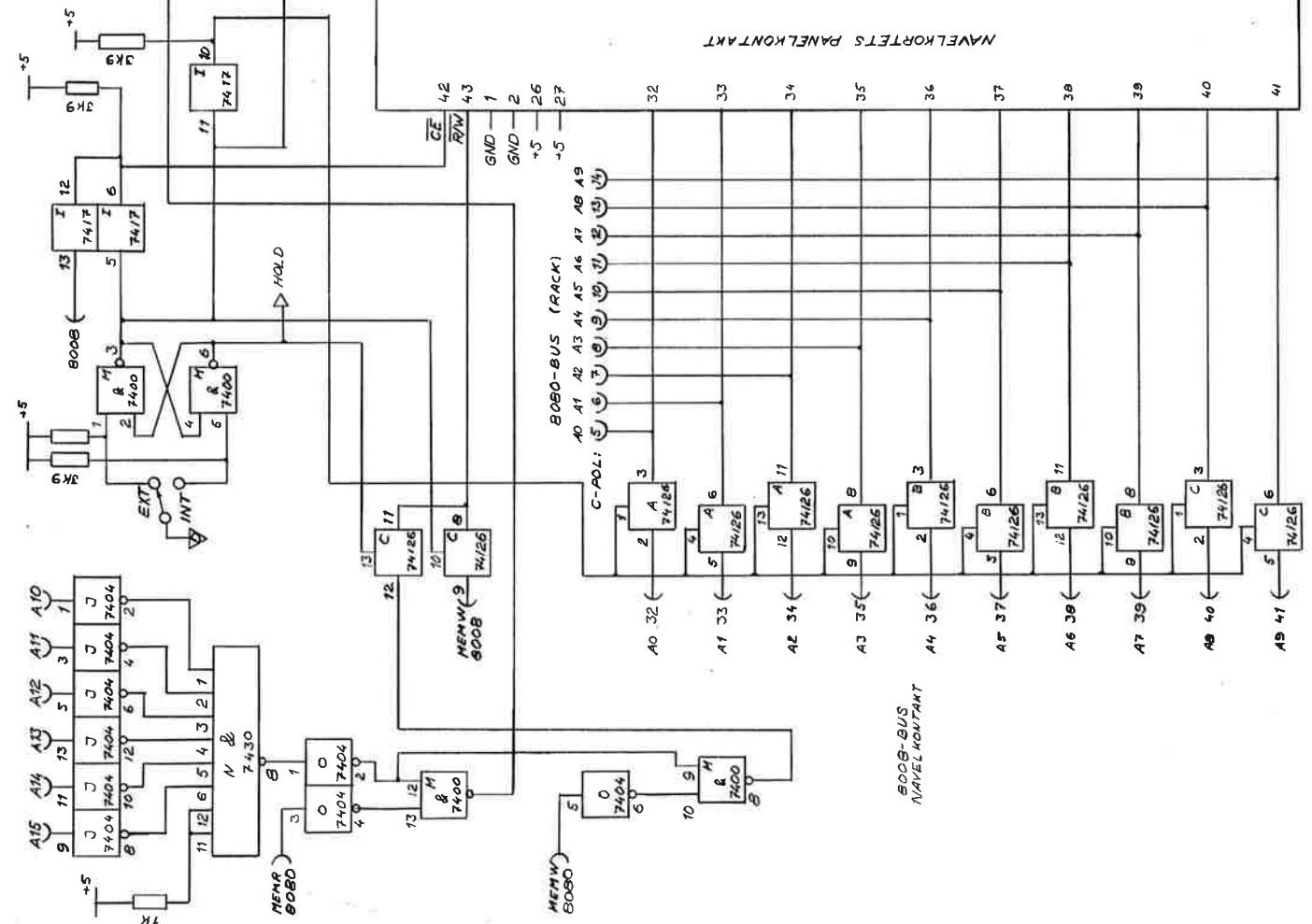


LTH
REGLERTEKNIK

CPU-KORTET

L 4.1
PO SJÖBERG
760929

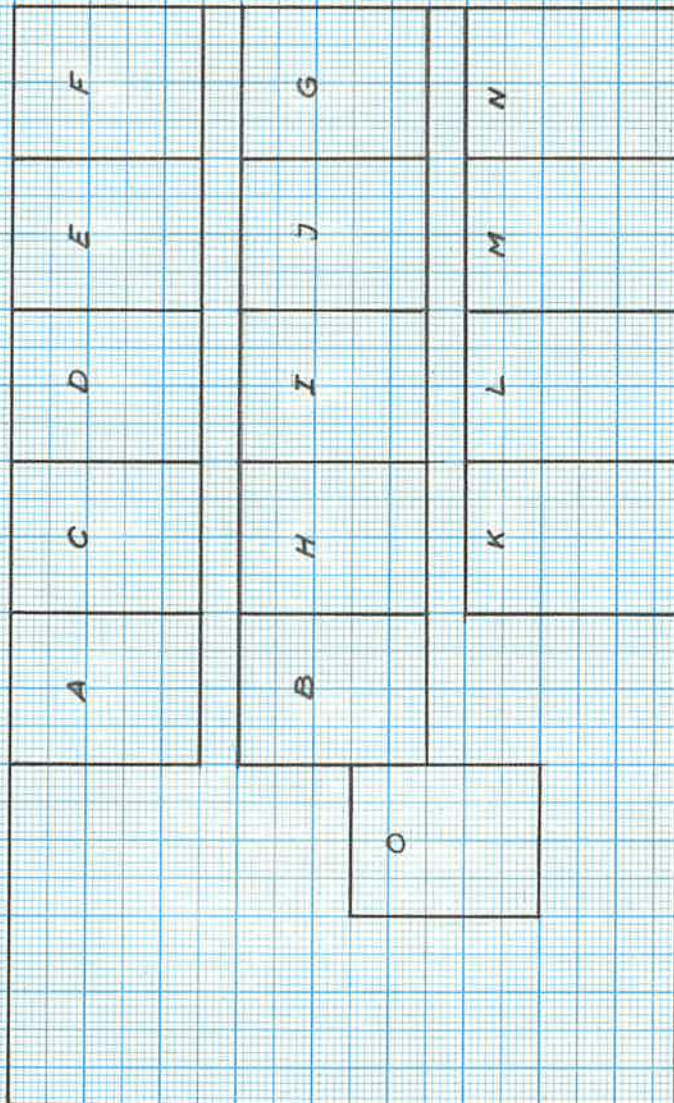


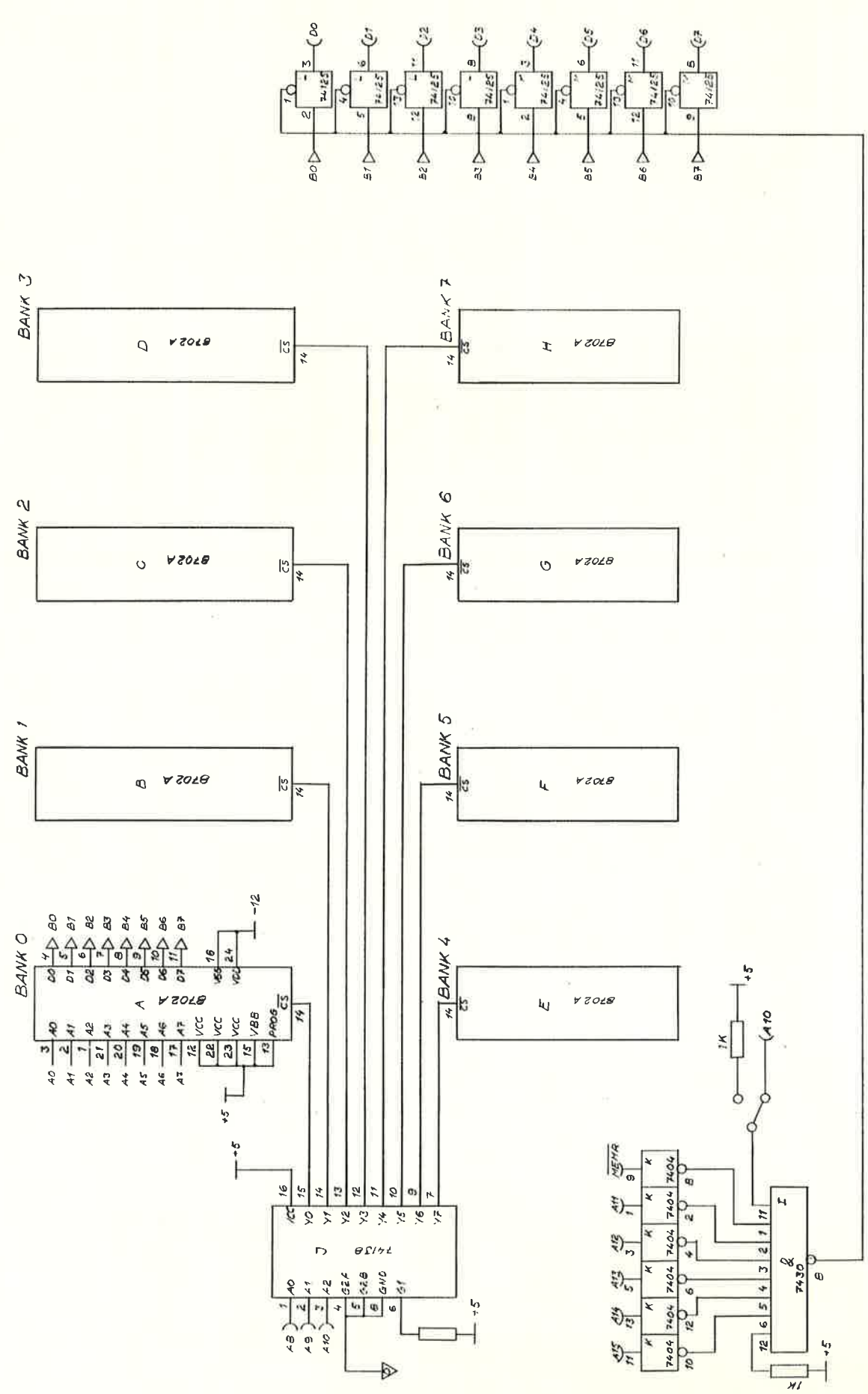


HOLD

BOBO DATA BUS

BOBO DATA BUS NAVELKONTAKT

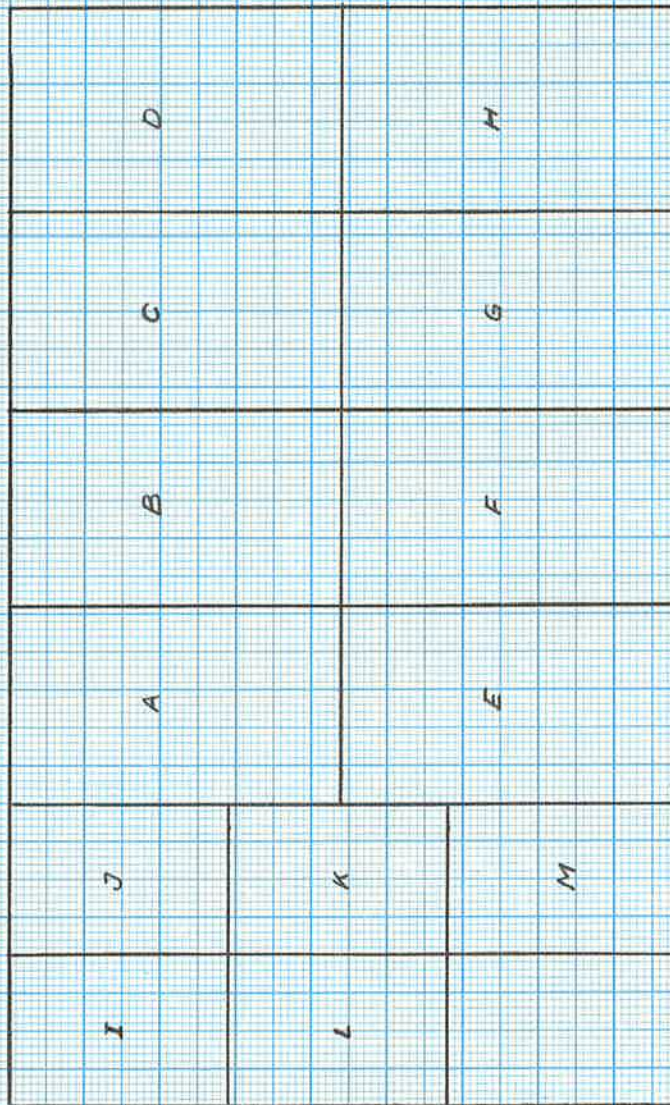




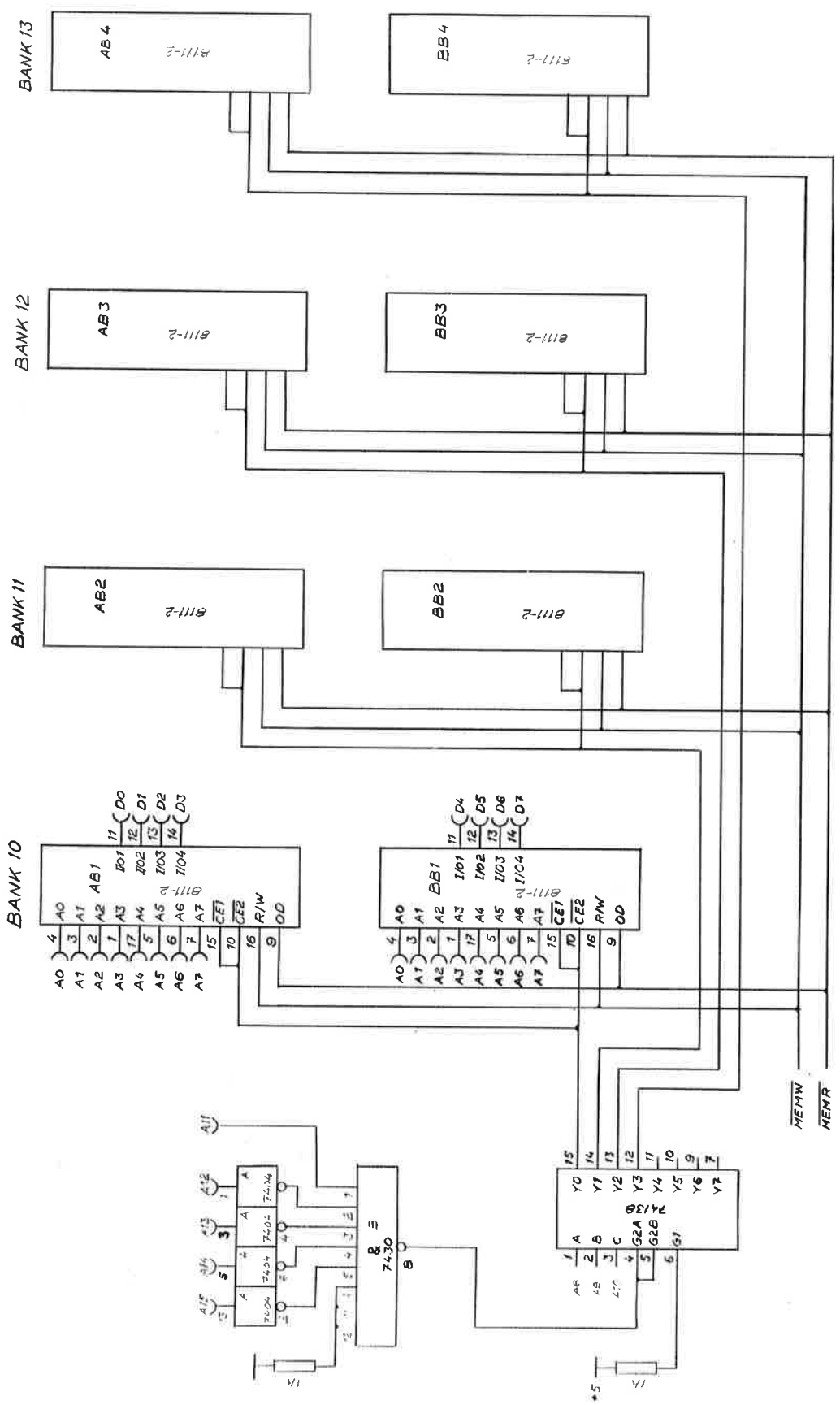
LTH
REGLERTEKNIK

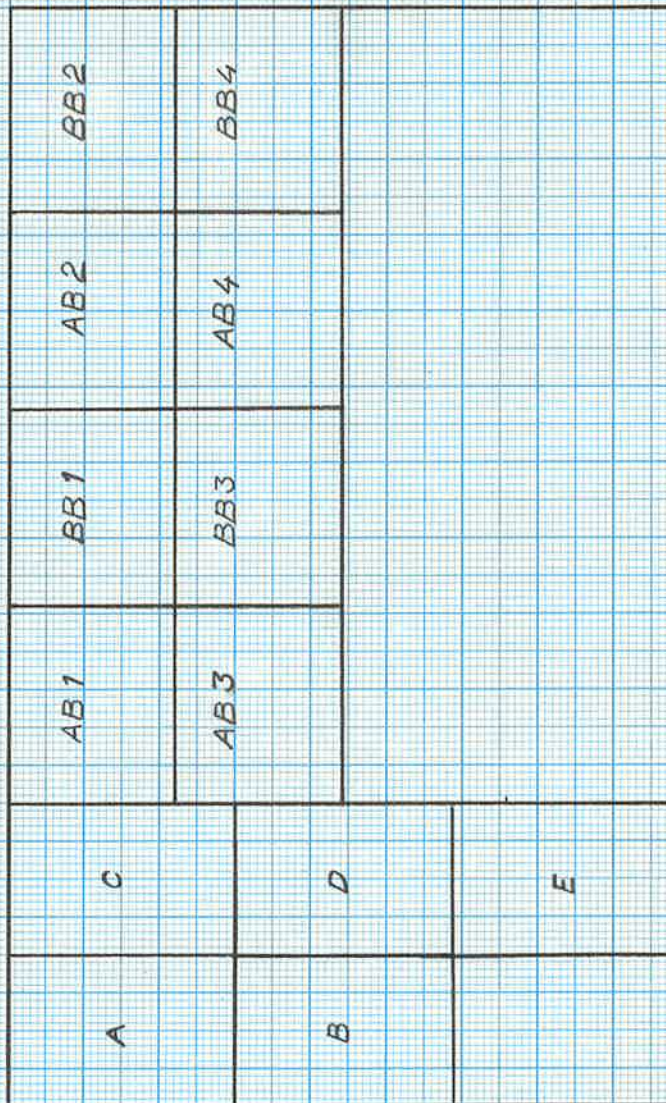
2K PROM

L 5,2
P-O SJÖBERG
760929



4 TH REGLERTEKNIK	1 K RAM	R 5.3 P.O. SJÖBERG 76 0929
----------------------	---------	----------------------------------





I.T.H
REGLETERIJK

2 CHANNEL DIA CONVERTER

R 7.1

