

===== MIKRODATAMATTEKNIK, trin 3 - FEJLFINDING =====

-----INDLEDNING-----

Lærebogen anvendes i undervisningen på kurset "Mikrodatamatteknik, trin 3 - fejlfinding", kursuskode 5820.

Metalindustriens Efteruddannelsesudvalg har foranlediget lærebogen optrykt.

Lærebogen er tilrettelagt af faglærerne Bent Bertelsen og Sven Dyhr fra Sønderborg tekniske skole i samarbejde med Jernindustriens Forlag.

Lærebogen "Mikrodatamatteknik - Fejlfinding" er delt op i instruktioner og øvelser.

Instruktionsdelen omfatter følgende emner:

- Signaturanalyse
- Logik state analyse
- Brug af oscilloscop i computer
- Logik prober
- Testprogrammer
- In-circuit emulator
- Komponentfejl
- PIV-1 systembeskrivelse

Øvelsesdelen omfatter øvelser i følgende emner:

- Systemkendskab
- MUART, 8256A analyse
- Program analyse
- Beregning af absolutte adresser
- Måling med Logik state analysator
- Optagelse af signaturen
- Måling med oscilloscop
- Måling med In-circuit emulator
- Fejllokalisering med logikprober

Sønderborg, april 1983
Håndværkerskolen i Sønderborg

----- Kursus mål -----

Kursets mål er at give kursusedtagerene kendskab til de teknikker og det måleudstyr der anvendes ved fejlfinding på dataudstyr, indeholdende mikrocomputere, herunder one-chip computere.

----- Kursus indhold -----

Kursusedtagerene vil ved teori og praktiske øvelser opnå en sådan viden, at de er i stand til at diagnosticere, fejlrette og funktionsprøve et fejlbehæftet rimeligt dokumenteret mikroprocessorbaseret system.

----- Indholdsfortegnelse -----

Emne:	afs.:
Indledning	1
Signaturanalyse	2
Logik state analyse	3
Oscilloscopmåling	4
Logik prober	5
Testprogrammer	6
In-circuit emulator	7
Komponentfejl	8
PIV-1 systembeskrivelse	9
Måleøvelser	10

----- Signatur analyse -----

Med fremkomsten af signatur analyse som fejlfindings princip i digitale systemer, har man i virkeligheden indført det fra analoge systemer kendte hjælpemiddel med at forsyne diagrammerne med spændinger og oscillogrammer som serviceteknikeren kan referere til under fejlretningen og derved spore en fejl tilbage til kilden.

Inden fremkomsten af signatur analyse var det meste fejlfinding på mikrocomputere baseret på instrumenter der i virkeligheden er måleudstyr for udviklingsingeniøren, som for eksempel logik state analysatoren. LSA er et meget effektivt instrument når det drejer sig om at undersøge i detaljer hvad der sker i computeren, men til fejlsøgning kan det være meget tidskrævende at sammenligne de opsamlede data med dokumentationen der skal foreligge som en assemblerudskrift.

Da det meste program i en computer er udformet som en større eller mindre sløjfe der gentages, drejer det sig i virkeligheden om at kontrollere om bitstrømmene på de forskellige bus- og kontrolledninger er korrekte i forhold til en kendt korrekt reference og fra gennemløb til gennemløb. Problemet er blot at frembringe en nemt overskuelig repræsentation af bitstrømmene, samt at en fejlagtig strøm udpeges med stor sikkerhed.

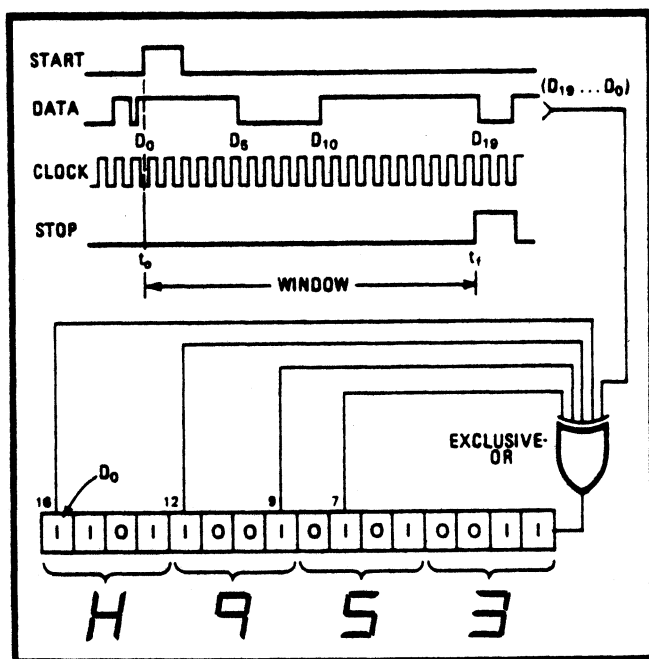
Der er i tidens løb anvendt flere forskellige løsninger på ovenstående problem. Man kan tælle antallet af "1" mellem en start- og en stoppuls. Denne løsning er dog meget usikker idet to vidt forskellige bitstrømme nemt kan indeholde samme antal "1". En bedre løsning findes i den såkaldte "transition counting" der går ud på at tælle antallet af skift fra "0" til "1" og omvendt mellem en start- og en stoppuls. Dette princip giver afhængig af bitstrømmens længde og fejlsens art (single- eller multibitfejl) en sandsynlighed for at finde en fejlbehæftet bitstrøm på mellem 50% og 99%.

En endnu større sikkerhed for at detektere en defekt bitstrøm kan opnås med signatur analyse. Med signatur menes et "ord" på fire karakterer der er unikt for den pågældende bitstrøm.

Signatur analysatorens princip:

Signatur analysatoren består af et 16 bit skifte-
register med liniær tilbagekobling fra 4 forskellige
udgange.

billede af skifteregister med tilbagekobling



Når der skal udledes en signatur af en bitstrøm, tilføres denne skifteregistrets indgang via en modul-2 additionskobling (5 input X-OR), og skifteregistret forsynes med en clockpuls der er synkron med den strøm der skal analyseres. For at få den samme signatur fra analyse til analyse skal indspilningen styres af en start og en stoppuls der udvælger et bestemt stykke af bitstrømmen. På grund af tilbagekoblingen vil indputtet til skifteregistret være et produkt af både det næste bit der skal skiftes ind, samt den samlede bitstrøm indtil da. Det vil sige at hvis der har været en fejl et sted i strømmen vil den uanset sin placering blive "fanget" og påvirke det bitmønster der til slut er i skifteregistret.

Med den viste kobling som anvendes i alle HP's signatur analysatorer, samt analysatorer med HP signaturer, er der 100% garanti for at finde enkelbitfejl og 99.998% chance for at finde multibitfejl.

Display:

Skifteregistrets indhold vises på et 4-cifret 7-segment display. Da signaturen ikke er et tal, men bitstrømmens "fingeraftryk" kan man vælge symbolerne ud fra læselighed hvorfor den normale hexadecimal symbolrække er blevet modificeret til følgende udseende:

0 1 2 3 4 5 6 7 8 9 A C F H P U

De udlæste signaturer vil være kombinationer af disse symboler som for eksempel:

5F29, HP74, 8585, UUUU, 0003, CF15, osv

----- HP 5005A signatur multimeter -----

5005A er en signatur analysator med mulighed for at "qualifye" datainputtet hvilket giver nem adgang til at tage signaturer på f.eks. indholdet i de enkelte ROM'er en en computer, herom senere.

Ud over dette kan instrumentet anvendes til:

DC spændingsmåling til 250 V
Deltaspændingsmåling i forhold til målt DC spænding
Positiv og negativ peakspænding
Frekvensmeter i KHz
Pulstæller mellem start og stop
Tidsmåling i msek.
Ohmmeter i K ohm.

For yderligere oplysninger henvises til manualen.

----- Signatur analyse i praksis -----

Muligheden for signatur analyse kan enten være designet ind i opstillingen eller man må selv forberede den medens det pågældende apparat er OK. I førstnævnte tilfælde følges det pågældende apparats servicemanual, medens man i sidste tilfælde selv må forberede apparatet til fejlsøgning med signatur analyse således at man har et sæt OK signaturer at referere til den dag apparatet fejler.

Pod 0 Low adresse + data

Pod 1 High adresse

Pod 2 Control signaler

Data S states max 64 K

delay S states max 64 K

Delay knap ud stopper ISA på triggerord

$S = \text{states} \leftarrow \begin{matrix} \text{Aktive klokflanker} \\ \text{dvs opsamling af delay af} \\ \text{clock pulser} \end{matrix}$

$T = \text{Trigger opsamling af delay på trigger ord}$

Ved delay ud over 256 clockpulser findes triggerord ikke i layer og kan derfor ikke displayes

triggerord styrer clocken

$DBY = 00126 T$ antal installerede linier hvor
Triggerord findes for dataopsamling
stopper

10101 0101 55

1010 1010 AA

CKO \Leftarrow AIE kun Adresse

CKO \Leftarrow RD + WR + INTA

med kun 1 klokke via Disassembler
ikke

Quasi = filnavnelsesvis, næsten

----- ADVARSEL -----

INDEN CPU'en SÆTTES I FREE-RUN BØR MAN FRAKOBLER
ALLE KREDSLØB DER KAN TAGE SKADE AF IKKE AT BLIVE KORREKT
INITIALISERET, F.EKS. THERMOPRINTERE, PROGRAMMERBARE
STRØMFORSYNINGER, MOTORER O.S.V.

Bitstrømmene på adressebussen:

Med CPU'en i free-run vil den tælle sin
programcounter op i det uendelige hvorfor der nu vil være
reproducerbare bitstrømme på adressebussen samt chip-
select logikkens udgange.

Bitstrømmene på databussen:

I takt med at CPU'en tæller igennem hele
adresseområdet fra 0000H til FFFFH selectes de
forskellige hukommelseskredse og porte hvorved de sender
deres indhold på databussen som en seriebitstrøm. Af
disse kredse er det kun ROM'erne (PROM og EPROM) der har
et kontrolleret indhold på hvilket der kan genereres
signaturer, medens RAM-lageret og portene med deres til-
fældige indhold vil give forskellige signaturer fra gang
til gang. Virkningen af dette kan undgås ved enten at
deselecte RAM og porte så de er forhindret i at "gå på
bussen", eller ved at kvalifere den opsamlede bitstrøm ved
hjælp af chip-selecten fra de kredse (ROM) der sender
stabile data på databussen.

----- Signaler til signatur analysatoren -----

Signatur analysatoren skal have følgende signaler
fra det undersøgte apparat:

Clockpuls.....: Skal være synkron med den undersøgte
bitstrøm, den aktive flanke vælges
på analysatoren.
Ved 8085 anvendes ALE ved opsamling
af adresser, og RD ved opsamling af
data.

Startsignal.....: Bestemmer starttidspunktet for opsam-
ling af bit, den aktive flanke vælges
på analysatoren.
Med CPU'en i free-run vælges i
almindelighed A15, faldende flanke
som startsignal idet dataopsamlingen
da starter når programcounteren
"flyder" over til 0000H.

Stopsignal.....: Bestemmer sluttidspunktet for opsamlingen af bit, den aktive flanke vælges på analysatoren. Med CPU'en i free-run vælges i almindelighed A15, faldende flanke idet dataopsamlingen da stopper når programcounteren flyder over til 0000H. Derved kommer dataopsamlingen til at dække et helt adressegennemløb.

Qualifyer.....: Ved hjælp af qualifyer inputtet kan man gate datatilførslen til signatur analysatoren inden for en START - STOP periode. I "QUAL-mode" bliver den oprindelige START input til et kombineret ST-SP input, og STOP inputtet får funktion af QUALIFYER input. Det aktive niveau for QUAL-signalet vælges på analysatoren. Måling i QUAL-mode er særlig nyttig ved optagelse af signaturer på databussen, idet man ved at anvende de enkelte ROM'ers CS-signaler som qualifyer efter tur, kan få et sæt signaturer for hver uden at tage dem ud af fatningerne.

Grundlæggende signaturer:

Når CPU'en er sat i free run, og analysatoren koblet til skal der på et typisk system optages følgende signaturer.

- Signatur for forsyningsspændingen. Denne bruges til at kontrollere om der er det rigtige antal clockpulser mellem start- og stoppulsen.
- Signaturer for alle adresseledninger.
- Signaturer for alle chip-select signaler.
- Signaturer for ROM'erne, samlet og individuelt ved enten at anvende qualifyer, eller delecte alt undtagen een ROM af gangen.

----- Signatur analyse på RAM-lager og porte -----

Signaturer på RAM-lageret og portene skal behandles som et afsnit for sig. Begge disse kredsløb lider af en mangel på konstant indhold fra tilslutning til tilslutning, hvorfor det ikke uden videre lader sig gøre at optage brugbare signaturer ved en simpel free-run test. Til både RAM- og porttest må der ligge nogle testrutiner i computerens ROM-lager der enten startes ved at udskifte en ROM, eller vælges med en serviceomskifter. Efter starten af en testrutine vil der være en repeterende bitstrøm på busserne hvorpå der kan tages signaturer.

Eksempel på algoritme til RAM-test:

```
-----  
Fyld RAM-lageret med skiftevis AAH og 55H  
-----  
Læs alle RAM adresser  
-----  
Fyld RAM-lageret med skiftevis 55H og AAH  
-----  
Læs alle RAM adresser  
-----  
Generer en start/stop puls  
-----  
Start forfra  
-----
```

RAM-test til signaturanalyse i 8085 assemblerkode:

```
;-----  
; Testprogrammet starter med at fylde det aktuelle  
; RAM-areal op med skiftevis AAH og 55H. Derefter  
; læses alle pladser. Nu fyldes RAM'en med skiftevis  
; 55H og AAH og det læses igen. For hvert totalt  
; gennemløb sendes en puls ud på CPU'ens pin 4, SOD.  
; START: pin 4, pos. flanke.  
; STOP: pin 4, neg. flanke  
; CLOCK: pin 32, pos. flanke, RD  
;-----
```

Fortsættes på næste side.....

```
RAMBOT EQU 4000H      ;RAM begyndelse
RAMTOP EQU 43FFH      ;RAM slut

CSEG
MVI A,40H             ;Reset SOD, pin 4, st./sp.
SIM
MVI B,0H              ;Omgangstæller
LOOP3: MVI C,0AAH      ;Startmønster = AAH
LOOP4: LXI H,RAMBOT
LOOP1: MOV A,C          ;Fyld RAM'en med skaktern
      MOV A,M
      CMA              ;Inverter testmønstret
      MOV C,A
      INX H            ;Find næste plads
      MOV A,L          ;Er vi færdige?
      CPI LOW(RAMTOP+1)
      JNZ LOOP1
      MOV A,H
      CPI HIGH(RAMTOP+1)
      JNZ LOOP1

      LXI H,RAMTOP      ;Læs de bit der er i RAM'en
LOOP2: MOV A,M
      INX H
      MOV A,L          ;Er vi færdige?
      CPI LOW(RAMTOP+1)
      JNZ LOOP2
      MOV A,H
      CPI HIGH(RAMTOP+1)
      JNZ LOOP2

      INR B            ;Er det et lige/ulige ?
      MOV A,B
      ANI 01H
      JZ LOOP3         ;Lige
      MVI C,55H        ;Nyt startmønster
      MVI A,0COH       ;Puls SOD, pin 4
      SIM
      MVI A,40H
      SIM
      JMP LOOP4        ;Ulige

END
```

Eksempel på algoritme til test P1 og P2 på MUART 8256A:

Inden programkørslen sammenkobles portene med en til formålet fremstillet testadaptor således at man kan sende data ud af den ene port og ind i den anden og omvendt.

```

-----
Generer startpuls
-----
Initialiser P1 som output og P2 som input
-----
Send efter tur AAH og 55H til P1 og læs P2
-----
Byt om på input og output
-----
Send efter tur AAH og 55H til P2 og læs P1
-----
Generer stoppuls
-----
Begynd forfra
-----

```

Porttest i 8085 assemblerkode:

```

;-----;
; Programmet afprøver P1 og P2 ved at sende data ;
; skiftevis fra P1 mod P2 og omvendt. ;
; START: SOD, pin 4, pos. flanke ;
; STOP: SOD, pin 4, neg. flanke ;
; CLOCK: RD , pin 32, pos. flanke ;
;-----;

```

```

MODER EQU OF3H ;P2 mode register
CMDRP1 EQU OF4H ;P1 kontrol register
PORT1 EQU OF8H ;PORT 1
PORT2 EQU OF9H ;PORT 2

```

CSEG

```

START: MVI A,40H ;Generer startpuls
        SIM ;Analysator starter på
LOOP: MVI A,0COH ;stigende flanke fra
        SIM ;SOD, pin 4

        MVI A,OFFH ;P1=output, P2=input.
        OUT CMDRP1 ;P1
        MVI A,0
        OUT MODER ;P2

```

```
MVI    A,0AAH      ;Send og læs AAH
OUT     PORT1       ;Ud på P1
IN      PORT2       ;Ind fra P2
MVI     A,55        ;Send og læs 55
OUT     PORT1       ;Ud på P1
IN      PORT2       ;Ind fra P2

MVI     A,0         ;P1=input, P2=output.
OUT     CMDR1       ;P1
MVI     A,03        ;
OUT     MODER       ;P2

MVI     A,0AAH      ;Send og læs AAH
OUT     PORT2       ;Ud på P2
IN      PORT1       ;Ind fra P1
MVI     A,55        ;Send og læs 55
OUT     PORT2       ;Ud på P2
IN      PORT1       ;Ind fra P1

MVI     A,40        ;Generer stoppuls
SIM     SIM         ;Stop på faldende flanke fra
JMP     LOOP        ;SOD, pin 4.

END
```

Under kørslen af ovenstående program kan der tages signaturer både på databussen og på "ydtersiden" af portene.

----- Dokumentation -----

For at have nogen fornøjelse af signaturanalyse som fejlfindingsværktøj, er det meget vigtigt at have/fremstille en fuldstændig dokumentation med samtlige oplysninger om hvorledes signaturerne er optaget.

Disse kan for eksempel være:

1) Hvorledes skal apparatet forberedes til testen:

- Kortslutningsbøjler der skal isættes/flyttes.
- Testomskiftere der skal indstilles.
- Forpladekontroller der skal indstilles.
- Om apparatet skal resettes.
- o.s.v.

- 2) Hvorfra skal styresignalerne til signatur analy-
satoren tages, og hvorledes skal denne indstilles.
 - START-signal; hvorfra og aktiv flanke.
 - STOP- signal; hvorfra og aktiv flanke.
 - CLOCKPULS; Hvorfra og aktiv flanke.
 - MODE; NORMAL eller QUAL, og da aktivt niveau.
 - Evt. hvilken logikfamilie skal analysatoren
indstilles til, TTL, ECL, CMOS, eller anden.
- 3) Specielle forholdsregler.
 - Pull up modstande der skal holdes mod probetip-
pen for at få en stabil signatur.

Dokumentationen kan være udført på flere forskellige måder der hver har deres fordele og ulemper. De fire almindeligste måder er:

- Signaturerne er påført diagrammet.
- Signaturerne er opført i tabeller.
- Signaturerne er påført en komponentpladserings-
tegning.
- Signaturerne indgår i et fejlfindingstræ.

Signaturer i diagrammet har den fordel at det er nemt at orienterer sig under fejlsøgningen, samt at det umiddelbart fremgår hvor den enkelte signatur hører til. Af ulemper må nævnes at det kan blive noget gnidret, specielt hvis der skal være plads til flere sæt signaturer der hver hører til sin test.

Signaturer på tabelform giver en mere flexibel dokumentation, idet det er nemmere at rette en liste til end at lave ændringer i et diagram og der er bedre plads til at vise flere signaturer for de samme punkter. I fejlfindingssituationen er der den ulempe at man måske samtidig skal have diagram og pladseringstegning for at finde rundt i apparatet.

Signaturer på en komponentpladseringstegning kaldes et signatur map. Fordelen ved dette princip er at man fra et og samme diagram får målepunktets pladsering og den tilhørende signatur. Princippet kan udbygges med vejledende pile der viser serviceteknikeren signalvejen på printpladen. Som en ekstra mulighed kan kortet udføres i gennemsigtigt materiale til at lægge på printet. Ulem-
pen ved princippet er at der nemt bliver trængsel så de enkelte signaturer bliver svære at skelne fra hinanden.

Fejlfindings træer er som bekendt et almindeligt brugt, og effektivt middel til at støtte en fejlfinding, men kombineres fejlfindingstræet med signaturer analyse vil det ofte være muligt at lokalisere en fejl med et væsentligt mindre antal målinger end man ellers ville have foretaget. For at dette kan lykkes, skal der ved fremstillingen af dokumentationen foretages et omfattende analyse og planlægningsarbejde, hvor "nøglesignaturerne" vælges ud og lægges til grund for spørgsmålene i fejlfindings træet.

Eksempler på dokumentation findes i "Operating and service manual" til 5005A signature multimeter, og HP application note 222 "A designers guide to signature analysis".

----- HP Microprocessor exerciser -----

Som det fremgår af det foregående, skal anvendelsen af signatur analyse som fejlfindingsværktøj forbedres både med hensyn til ændringer i hardwaren som med optagelse af signaturer. Forberedelserne af hardwaren kan man forenkle, og i tilgift få adgang til et stort antal effektive testprogrammer ved at anvende HP's Microprocessor Exerciser i forbindelse med signatur analysatoren. Det eneste der kræves af det undersøgte apparat, er at CPU'en sidder i en fatning så den kan flyttes over i Exerciseren, og erstattes med exerciserens 40-pin stik.

Microprocessor Exerciser, opbygning:

Når der sættes en CPU (8085 for 5001C) i 40-pin fatningen på exerciserens overside er den en komplet microcomputer med adgang til 56 forskellige testprogrammer. Exerciseren forbindes til det undersøgte apparat via et 40 leder bündelkabel med stik der passer til CPU-fatningen. Ved hjælp af buffere er Exerciserens systembus skilt fra busserne i det undersøgte apparat, således at en fejl i dette ikke vil hindre afviklingen af exerciserens testprogrammer. Det er derfor muligt at generere test mønstre på hvilke man kan tage signaturer selvom der er alvorlige fejl i det undersøgte apparat.

Indbyggede testprogrammer:

Ved hjælp af de indbyggede testprogrammer kan der tages signaturer til lokalisering af fejl inden for følgende områder i hardwaren:

- Test af microprocessorens instruktionssæt.
- Free run test af adresse og databus.
- ROM læse test.
- RAM skrive læse test.
- Output port test mønstre (simple porte)
- Input port test mønstre (simple porte)

Hvis der skal tages signaturer i forbindelse med programmerbare interfacekredse er det nødvendigt at skrive et testprogram som initialiserer kredsen og derefter genererer de nødvendige testmønstre. Testprogrammet skal indeholdes i en EPROM type 2716 som sættes i en fatning på Exerciserens overside, hvorefter man med en omskifter vælger at få kørt programmet i denne externe ROM.

Sammenkobling af Exerciser-Analysator-målesystem:

Som allerede nævnt sker sammenkoblingen mellem Exerciser og målsystem via et 40 leder bündelkabel med stik der sættes med i CPU fatningen. Gennem dette kabel får Exerciseren sin forsyningsspænding, hvilket dog forudsætter at det undersøgte apparats power kan leverer de 500 mA ekstra som Exerciseren bruger. Hvis dette ikke er tilfældet kan Exerciseren forsynes fra en selvstændig 5V spændingsforsyning.

Sammenkoblingen mellem Signatur analysatoren og Exerciseren sker via analysatorens START/STOP/ og CLOCK indgange, idet de en gang for alle sættes i de dertil hørende bøsninger, og forbliver her under afviklingen af alle testprogrammerne. Der er således ikke nødvendigt at finde velegnede kontrolsignaler i det undersøgte apparat.

Understøttede mikroprocessorer:

I skrivende stund (nov. 82) findes Microprocessor Exerciseren til 6800, 8085 og Z80.

Litteraturhenvisning:

- HP, A designer's guide to signature analysis. Note 222
- HP, An index to signature analysis
publications. Note 222-0
- HP, Implementing signature analysis for
production testing. Note 222-1
- HP, Application articles on
signature analysis. Note 222-2
- HP, A managers guide to signature analysis. Note 222-3
- HP, Guidelines for signature analysis.
Understanding the signature measurement. Note 222-4

----- Logik State Analysator, LSA -----

Måleprincip:

Logik State Analysatoren er et REGISTRERENDE instrument med hvilket man kan opsamle og vise hvad der SKETE i den computer der er under observation. I en fejlfindingssituation skal opsamlingen af data efterfølges af en analyse af det opsamlede, idet analysatoren ikke direkte fortæller hvad årsagen til den eventuelle fejl er, men kun følgen af den. For at anvende analysatoren effektivt er det nødvendigt at have software-dokumentationen til det fejlramte udstyr, og denne bør foreligge i assemblerkode.

Dataopsamlingen foregår SYNKRONT med hændelserne i det undersøgte udstyr, idet clockpulsen til analysatoren tages i dette. Det vil sige at det der samles op, er de niveauer (logiske states, deraf navnet logic state analyser) der var i de undersøgte punkter på tidspunktet for clockpulsens aktive flanke.

Ved måling i en microcomputer opsamler man almindeligvis data fra databussen, adressebussen samt vigtige kontrolsignaler som chipselect- og interruptsignaler.

----- Philips Logic Scope PM3543 -----

Blokdiagram:

Blokdiagram og beskrivelse er oversat og bearbejdet fra Operating manual til PM3543.

Microprocessor system:

Analysatoren styres af en 8085 microprocessor. Denne CPU har en multiplexet adresse-data bus. De otte mindst betydende adressebit hentes fra den multiplexede bus via adresselatchen. Hukommelsen består af en PROM som indeholder analysatorens styreprogram, samt et RAM-lager der bruges til midlertidig lagring af data, f. eks. fra en tidligere dataopsamling.

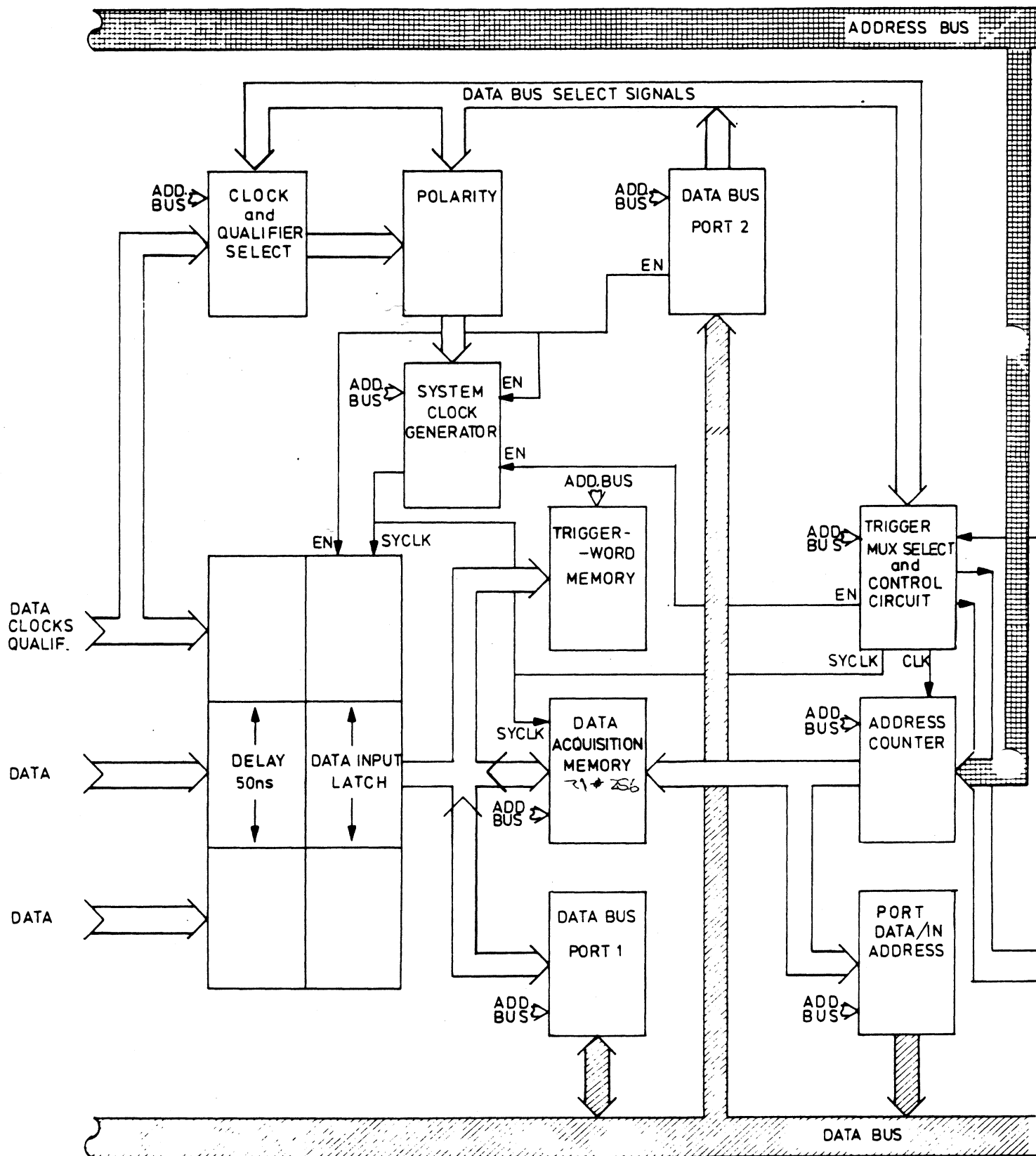
Control switches (på forpladen):

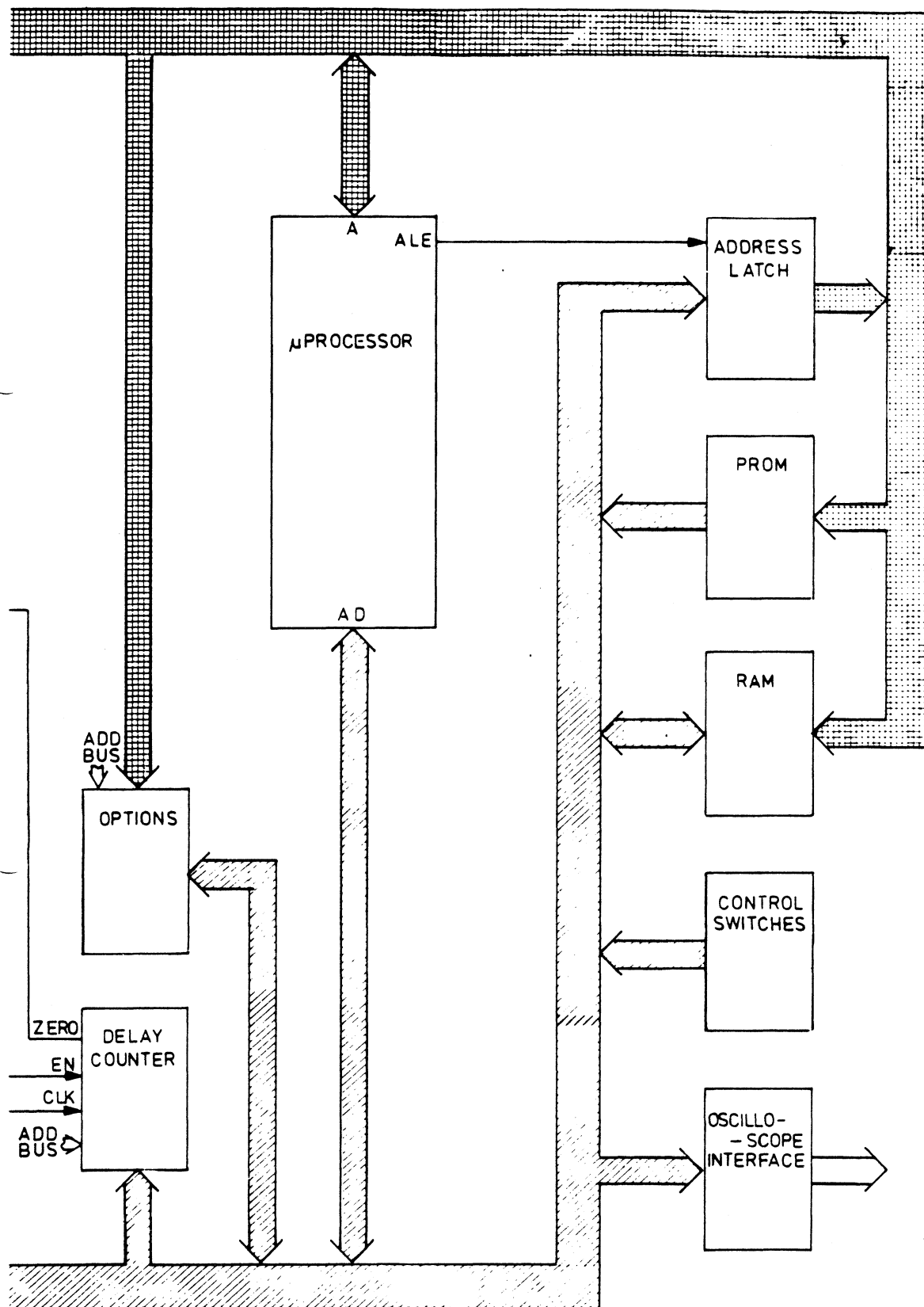
Informationen om forpladekontrollernes stilling og betjening læses ind til microprocessoren, som derefter foretager indstillingen af analysatoren i overensstemmelse med indtastningerne, f.eks. indgivelse af triggerord, delay, triggermode o.s.v.

Oscilloscop Interface:

Dette kredsløb styrer, under kontrol af microprocessoren, den lodrette og vandrette afbøjning på katode-

Blokdiagram af PM3543's logikanalysator sektion.





strålerøret, samt karaktergeneratoren. Via dette kredsløb tilføres oscilloscopsektionen det der skal vises.

Pod's:

Data-, clock- og kvalifyerinputtet til analysatoren sker via tre pod's. Disse er forsynet med buffere, således at det system der måles på ikke bliver for hårdt belastet. Inputtærskelsspændingen er enten TTL-kompatibel, eller en af to ander spændinger der kan indstilles manuelt og tilpasses ikke TTL-kompatibel logik.

Delay (input delay):

For at opnå en data hold tid på nul, det vil sige at data må forsvinde på samme tid som den aktive flanke på clockpuls, er det nødvendigt at forsinke data i forhold til clock- og kvalifyer.

Data input latches:

Data fra Pod'ene bliver fanget i input latchene for at sikre at de er stabile når de skal læses ind i data opsamlings hukommelsen (data acquisition memory).

Data Acquisition memory (data opsamlings hukommelse):

De opsamlede data flyttes over i data opsamlings hukommelsen. Det er data opsamlings hukommelsens maksimale arbejdshastighed der bestemmer hvor højt analysatoren kan gå op i frekvens (og dens pris!). For PM3543 er den maksimale clockfrekvens 10 MHz.

Data Bus Port 1:

De opsamlede data i data opsamlings hukommelsen kan læses af microprocessoren via data bus port 1. Det er også muligt at overføre data til data opsamlings hukommelsen af denne vej.

Address counter:

Adressetælleren holder kontrol med hvor i dataopsamlings hukommelsen de opsamlede data skal læses ind. Da analysatoren arbejder i "negativ tid", det vil sige opsamler data kontinuert når den er startet og går i stå når triggerordet findes, vil adresetælleren tælle "rundt" og derved overskrive tidligere opsamlede data indtil analysatoren stopper. Det bevirker at data opsamlings hukommelsen kommer til at virke som et skifteregister på 255 x 21 bit der indeholder de sidste 255 indspilninger. Via adressebussen kan microprocessoren sætte adresse counteren til en vilkårlig værdi.

Address Pointer (port data/in address):

Adresse counterens indhold kan læses af microprocessoren via adresse pointeren, således at microprocessoren, efter dataopsamlingen er stoppet, kan konstatere hvor i data opsamlings hukommelsen den sidste indlæsning fandt sted.

Triggerword Memory:

Inden der startes en dataopsamling, skal der vælges et eller to triggerord. De(t) valgte triggerord lagres i triggerord hukommelsen under kontrol af micro-processoren idet de(t) overføres fra kontroltasterne på forpladen via databussen og Data Bus Port 1. Når dataopsamlingen er startet, og der findes et triggerord, giver Triggerword Memory'en et signal til Trigger-mode Select der bestemmer hvad der videre skal ske.

Trigger-mode Select and Control:

Trigger-mode Select leverer kontrolsignaler til Delay Counteren, Adresse counteren og system clock generatoren afhængigt af den valgte triggermode. Der kan vælges mellem Parallel, Quasi-parallel og sequentiell trigning. Se senere om de forskellige triggermodes. Efter opdagelsen af et triggerord skal delay counteren, hvis den er indkoblet, enables og tilføres clockpulser så den kan tælle ned til nul. Når delay counteren er nået til nul sender den et signal tilbage til Trigger-mode Select and Control der afhængig af den valgte mode styrer det videre forløb, f.eks. stopper dataopsamlingen.

Delay Counter:

Inden dataopsamlingen startes, presettes Delay Counteren til en værdi mellem 0 og 65535. Når analysatoren tændes er delayet automatisk sat til 126. Delay Counteren starter nedtællingen når triggerordet er fundet, og bestemmer hvor længe dataopsamlingen skal fortsætte derefter. Nedtællingen kan foregå enten med system-clockpulsen der i store træk svarer til den clockpuls der kommer fra det undersøgte udstyr, eller med outputtet fra Triggerword Memoryen. I første tilfælde taler man om et clock-delay, og i andet tilfælde om et trig-delay, hvor triggerordet skal findes at antal gange inden analysatoren stopper dataopsamlingen.

Data Bus Port 2:

Via Data Bus Port 2 styres valget af aktive clockflanke(r), kvalifyerpolaritet samt styring af system clock generator og data input latch. Ligeledes sættes trigger-moden af denne vej.

Clock and Qualifyer selection:

I dette kredsløb bestemmes på hvilke flanke af clockpulsen der skal læses data ind i analysatoren, samt polariteten af kvalifyerene.

System Clock Generator:

System Clock Generatoren leverer den nødvendige interne clockpuls til logikanalysatoren. Denne er afledt af input-clockpulsen og timet i overensstemmelse med den valgte trigger-mode.

----- Indgange - POD 0, POD 1, POD 2 -----

Tilslutningen af analysatoren til testobjectet foregår via 3 POD's. I hver POD er der 8 dataindgange og en stelledning.

ffffffffffff HUSK at tilslutte stel i hver POD ffffffffff

Indgangene i de tre POD's har følgende funktion:

Ledning Nr.	0	1	2	3	4	5	6	7
POD 0:	DO	D2	D2	D3	D4	D5	D6	D7
POD 1:	D8	D9	D10	D11	D12	D13	D14	D15
POD 2:	D16	D17	D18	D19	D20	Q0	CK1	CK0
ELLER:	D16	D17	D18	Q2	Q1	Q0	CK1	CK0

Som det fremgår kan D19 og D20 i POD 2 enten bruges som databit, eller som kvalifyerinput.

----- Tilslutning af PM5343 og 8085 microprocessor -----

POD 0 input: forbindes til 8085:

ground	stel
data 0-7	ADO - AD7

POD 1 input:

ground	stel
data 0-7	A8 - A15

POD 2 input:

ground	stel	
data 0	ledig	
data 1	ledig	
data 2	IO/M	
data 3	S1	Q2 = X (dont' care)
data 4	S0	Q1 = X (dont' care)
data 5	HLDA	Q0 = 0
data 6	RD	CK1= pos. flanke
data 7	ALE	CK0= neg. flanke

----- PM3543's dual clock fasilitet -----

Som det fremgår af tabellen øverst på side 3-6, har PM3543 21 datainput. Da en "normal" 8 bit CPU uden multiplexet bus kræver 16 + 8 kanaler må man i forbindelse med en sådan give afkald på 3 af adresseledningerne. Når det drejer sig om en CPU med multiplexet bus, og tilhørende adresse- og datastrobe kan man udnytte PM3543's to clockindgange, idet CK0 tilføres adressestroben (ALE) og CK1 datastroben (RD eller WR). Ved hjælp af de to clockindgange er analysatoren istand til at demultiplexe den multiplexede bus fra CPU'en, og med den store fordel at der kun skal forbindes en probeledning til hvert busledning. Derved udvides de 21 indgange faktisk til 37, idet POD 0 og POD 1 først clockes ind med CK0 og derefter med CK1, medens POD 2 kun clockes ind med CK0.

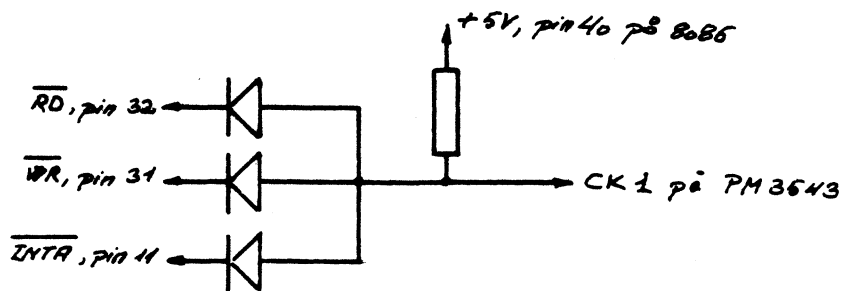
Processorer med separat RD- og WR-clock.

Da der kun er een clock 1 indgang på analysatoren kan der umiddelbart kun tilføres een data-clock. Det betyder at der kommer til at mangle data i read- eller write-cycles alt efter hvilken data-clock der er valgt. Dette problem kan omgås ved at gate de forskellige clockpulser sammen inden de tilføres analysatoren som et kombineret signal.

På 8085 er RD og WR signalerne negerede (aktiv low) hvorfor der skal/kan anvendes en indgangsnegeret NOR-gate til sammengatningen.

Ifølge d'Morgan: $\overline{\overline{RD}} + \overline{\overline{WR}} = RD \cdot WR$

Altså en AND-gate. Denne gate kan realiseres med en 74SXX eller 74LSXX TTL-kreds, idet der skal tages hensyn til den forsinkelse der indføres. En nemmere løsning er at fremstille gaten som en diodegate som følgende skitse viser.



Materialer: dioder = AA119, modstand = 10 Kohm.

----- PM3543's kontroltaster -----

Kontroltasterne er delt op i 5 hovedgrupper med funktion som følger:

NAVN	FUNKTION

Control keys (de tre øverste)	

LSA/OSC	LSA eller oscilloscop mode <i>35 MHz</i>
AUTO/MAN	Automatisk eller manuel start af LSA
START/STOP	Start dataopsamling/manuel trigger

Trigger control keys

SEQ	dataopsamling med een clockpuls: OUT: et triggerord (TRIG) IN: To triggerord (ARM + TRIG)
	dataopsamling med to clockpulser: OUT: Quasiparallel trigning (TRIG+TRIG) IN: To triggerord, med ARM-ordet på CKO, og TRIG-ordet på enten CKO eller CK1.
DELAY	OUT: Delay udkoblet. IN: Delay indkoblet.
FALSE	OUT: Trigning på triggerordet (sand). IN: Trigning på andet end det andet triggerord (falsk). Kun ved to triggerord.

Display control keys

SEQ	IN: De opsamlede data vises i den rækkefølge (sekvens) de er opsamlet, med angivelse af den clock de er opsamlet med. OUT: De på CKO og efterfølgende CK1 opsamlede data vises side om side.
COMP	IN: Indkobler compare-mode.
GRAPH	IN: Indkobler graph-mode display.
EXCHANGE	Når HEX er ude, kan displayet formateres til blandet HEX-BIN bestemt den horiontale coursers plads.
HEX	IN: Display i rent HEX-format.

NAVN	FUNKTION

SPECL opsætning Selection/Execute keys	IN: Indkobler disassembler option.

MENU	IN: Menu-mode. OUT: Run- og display-mode.
TRIGW	IN: Muliggør overførsel af et nyt sæt triggerord fra det aktuelle display (de to øverste linier) ved indtrykning af X/EXCH-keyen.
DATA	IN: Indholdet af compare-hukommelsen vises. Hvis X/EXCH-keyen indtrykkes overføres de opsamlede data til compare-hukommelsen.
1, INC, RCL	MENU inde: Sætter "1" ind i binære tal og tæller decimale og hexadecimale tal op. MENU ude: Genfremkalder tidligere blandede kanaler.
0, DECR, BLANK	MENU inde: Sætter "0" ind i binære tal og tæller decimale og hexadecimale tal ned. MENU ude: Blanker uønskede kanaler sammen med den horisontale courser .
X, EXCH, INV	MENU inde: Sætter X (dont' care) ind i binære tal, og skifter bogstaver, f.eks. S til T. MENU ude: Inverterer den søjle i data-listen som den horisontale courser ligger over.

----- Trigger-modes -----

PM3543 har tre grundlæggende triggermodes, disse er:

- 1) Parallel trigning med 1 clockpuls.

Triggerordet, med en bredde på max. 21 bit, skal findes på clock 0, derefter skal delayet tælle ned til nul hvorefter dataopsamlingen stopper, og de opsamlede data vises på skærmen.

2) Quasiparallel trigning med 2 clockpulser.

Der er to triggerord, et for hver clockpuls. Det første triggerord skal findes på clock 0, og på den næste clockpuls der SKAL være clock 1 skal det andet triggerord findes. Findes det andet triggerord ikke på det rigtige tidspunkt leder analysatoren efter det første ord igen. Når begge ord er fundet tæller delayet ned til nul og dataopsamlingen stopper. Denne triggermode er "født" til multiplexede processorer, idet det første triggerord typisk vil være adressen der skal findes på CK0 (ALE), og det andet ord er de tilhørende data der skal findes på CK1 (RD). I denne mode er den maximale bredde på triggerordet 42 bit. Til mange anvendelser vil man sætte det andet triggerord til dont' care, og trigge på adressen alene.

3) Sequentiel trigning med delay og en eller to clockpulser.

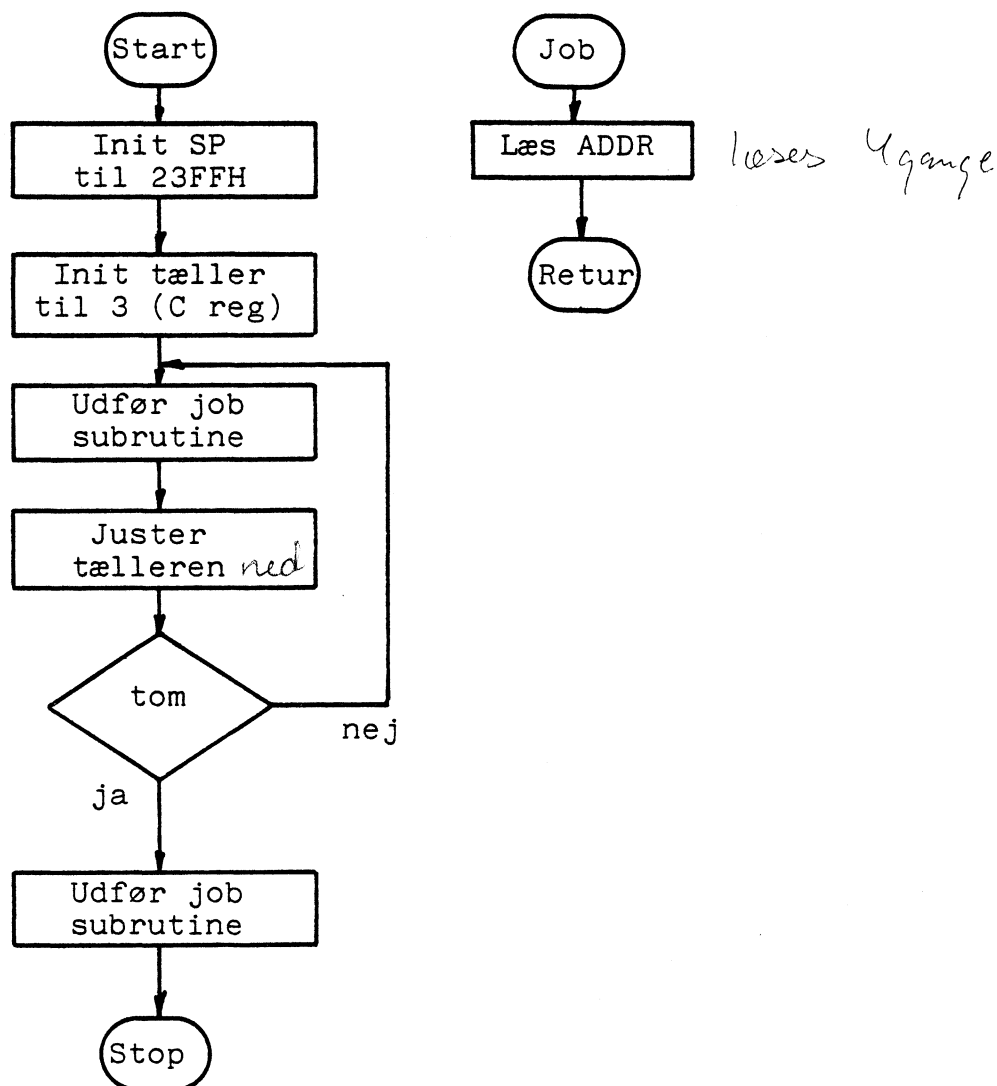
Ved sequentiel trigning skal analysatoren finde en sekvens af 2 triggerord. Det første af disse skal altid findes på CK0 og kaldes ARM-ordet (armeringsordet). Når dette er fundet leder analysatoren efter det andet ord, TRIG-ordet der kan sættes til at falde på CK0 eller CK1. Delayet kan lægges efter triggerordet, eller mellem ARM og TRIG-ordet. I sidste tilfælde begynder analysatoren først at lede efter TRIG-ordet når delayet har talt ned til nul. BEMÆRK at delayet tæller på TRIG-ordets clock!

Eksempler: I det følgende er der eksempler på anvendelsen af de forskellige triggermodes illustreret med et simpelt program, se næste side.

Side 3-12 Parallel trig. m. 1 cp. Data=S Delay=S
" 3-13 Quasiparallel trig. m. 2 cp. Data=S Delay=S
" 3-14 Sequentiel listning.
" 3-15 Sequentiel trig. m. 2 cp. Data=S Delay=S
" 3-16 Sequentiel trig. m. 2 cp. Delay mel. TRIG-ord.
" 3-17 Quasiparallel trig. m. 2. cp. Delay=T Data=S
" 3-18 Sequentiel trig. m. 1 cp. Z-delay mel. T-ord.
" 3-19 Quasiparallel trig. m. 2 cp. Data=T Delay=T
" 3-20 Quasiparallel mode med FALSE trig.
" 3-21 DATA-memory og COMPARE-mode.

S = STATES, d.v.s. opsamling og delay på clockpulser.
T = TRIGGERS, d.v.s. opsamling og delay på triggerord.

Programeksempel til belysning af de forskellige triggermodes. Programmet består af to sløjfer, og en subrutine der kaldes i og efter den første sløjfe. Programmet stopper i en endeløs sløjfe.



Assemblerliste:

2000	3123FF	START:	LXI	SP,23FFH	;Init SP
2003	212200		LXI	H,2200H	;Init HL som RAM point
2006	0E03	IGEN:	MVI	C,3H	;Init tæller
2008	3655	HOP:	MVI	M,55H	;Udfør jobbet med 55
200A	CD2019		CALL	JOB	;Hop til subrutinen
200D	0D	JUSTE:	DCR	C	;Tæller -1
200E	C22008		JNZ	HOP	;Een gang til
2011	36AA		MVI	M,AAH	;Udfør jobbet med AA
2013	CD2019		CALL	JOB	;Hop til subrutinen
2016	C32016	LOOP:	JMP	LOOP	;Stop
2019	7E	JOB:	MOV	A,M	;Opdater accumulatoren
201A	C9		RET		;Job slut

Parallel trigning med 1 clockpuls og Delay=S, Data=S

Assemblerliste:

```

2000* 3123FF  START: LXI  SP,23FFH ;Init SP
2003  212200          LXI  H,2200H ;Init HL som RAM point
2006  0E03      IGEN: MVI  C,3H    ;Init tæller
2008  3655      HOP:  MVI  M,55H    ;Udfør jobbet med 55
200A  CD2019          CALL JOB      ;Hop til subrutinen
200D  0D          JUSTE: DCR  C      ;Tæller -1
200E  C22008          JNZ  HOP      ;Een gang til
2011  36AA          MVI  M,AAH      ;Udfør jobbet med AA
2013  CD2019          CALL JOB      ;Hop til subrutinen
2016  C32016  LOOP:  JMP  LOOP      ;Stop

2019  7E          JOB:  MOV  A,M      ;Opdater accumulatoren
201A  C9          RET                ;Job slut

```

* = Triggerordets pladsering

CK0.Q0.Q1.Q2=0XXX CK1.Q0.Q1.Q2=XXXX
 TRIG= X XXXX 20 00 D CK0

DATA=S DLY=00253S SPECL

LINE NO	POD#2	#1	#0	CK
- 1	1 10..	02	3A	#0
TRIG 1	1 10..	20	00	#0
2	0 10..	20	01	#0
3	0 10..	20	02	#0
4	1 10..	20	03	#0
5	0 10..	20	04	#0
6	0 10..	20	05	#0
7	1 10..	20	06	#0
8	0 10..	20	07	#0
9	1 10..	20	08	#0
10	0 10..	20	09	#0
11	1 00..	22	00	#0
12	1 10..	20	0A	#0
13	0 10..	20	0B	#0
14	0 10..	20	0C	#0
15	1 00..	23	FE	#0
16	1 00..	23	FD	#0
17	1 10..	20	19	#0
18	0 10..	22	00	#0
19	1 10..	20	1A	#0
20	0 10..	23	FD	#0

Triggerordet 2000H findes på CK0 (ALE) hvorefter analysatoren yderligere opsamler 253 linier og stopper. Da der kun arbejdes med en clockpuls, og det er adressestroben (ALE), er der kun opsamlet adresser. Bemærk stakoperationen i linie 14, 15 som følge af CALL instruktionen. Via PODf2 er i følgende rækkefølge fra venstre indspillet, S0, S1 og IO/M. Displayet er formateret til HEX i PODf0 og f1, samt de to laveste bit i PODf2 blanket.

Quasiparallel trigning med 2 clockpuls, Delay=S, Data = S

Assemblerliste:

```

2000 3123FF START: LXI SP,23FFH ;Init SP
2003 212200      LXI H,2200H ;Init HL som RAM point
2006 0E03      IGEN: MVI C,3H ;Init tæller
2008* 3655      HOP: MVI M,55H ;Udfør jobbet med 55
200A CD2019      CALL JOB ;Hop til subrutinen
200D 0D      JUSTE: DCR C ;Tæller -1
200E C22008      JNZ HOP ;Een gang til
2011 36AA      MVI M,AAH ;Udfør jobbet med AA
2013 CD2019      CALL JOB ;Hop til subrutinen
2016 C32016 LOOP: JMP LOOP ;Stop

2019 7E      JOB: MOV A,M ;Opdater accumulatorens
201A C9      RET ;Job slut

```

Der skal triggles når programmet første gang går ind i den sløjfe der starter på adr. 2008H

```

CK0.Q0.Q1.Q2=00XX CK1.Q0.Q1.Q2=10XX
TRIG= XX 20 08 Z CK0
TRIG= XX 20 36 D CK1
DATA=S DLY=001265 DISA 8085

```

```

LINE POD#2 #1 #0 #0
NO CK #0 #0 #0 #1
^
- 1 03

```

TRIG	18 20 08	36 MVI M	
1	08 20 09	55	
2	10 22 00	55	
3	18 20 0A	CD CALL	
4	08 20 0B	19	
5	08 20 0C	20	
6	10 23 FE	20	
7	10 23 FD	0D	
8	18 20 19	7E MOV A,M	
9	08 22 00	55	
10	18 20 1A	C9 RET	
11	08 23 FD	0D	
12	08 23 FE	20	
13	18 20 0D	0D DCR C	
14	18 20 0E	C2 JNZ	
15	08 20 0F	08	
16	08 20 10	20	
17	18 20 08	36 MVI M	
18	08 20 09	55	
19	10 22 00	55	

Ved Quasiparallel trigning skal de to triggerord findes på henholdsvis CK0 og CK1. CK0-ordet er adressen, og CK1-ordet er den øverste halvdel af adressen og data. Delayet løber på CK1 der er et sammengated RD- og WR-signal fra CPU'en. Uden denne sammenkobling af RD og WR, ville der mangle data i de linier hvor CPU'en skriver, idet man normalt vil vælge at bruge RD-signalet til CK1. Tabellen er vist med disassembleren i funktion.

Sequentiel listning, knap DISPLAY, SEQ indtrykket

Nedenstående tabel har samme indhold som eksemplet på foregående side, blot er de opsamlede data vist i den rækkefølge (sequens) de er spillet ind i analysatoren. Til højre for tabellen er angivet med hvilke clock-puls den pågældende linie er opsamlet. Den sequentielle liste kan være nyttig til at få et overblik over det tidsmæssige forløb af hændelserne i computeren.

```

CK0.Q0.Q1.Q2=00XX  CK1.Q0.Q1.Q2=10XX
TRIG=  XX 20 08  Z CK0
TRIG=  XX 20 36  D CK1
DATA=S          DLY=00126S  DISA 8085
LINE  POD#2 #1 #0
NO
      ^
- 2      08 20 03      #1
TRIG      18 20 08      #0

TRIG      18 20 36 MVI M      #1
  1      08 20 09      #0
  2      08 20 55      #1
  3      10 22 00      #0
  4      10 22 55      #1

  5      18 20 0A      #0
  6      18 20 CD CALL      #1
  7      08 20 0B      #0
  8      08 20 19      #1
  9      08 20 0C      #0

10      08 20 20      #1
11      10 23 FE      #0
12      10 23 20      #1
13      10 23 FD      #0
14      10 23 0D      #1

15      18 20 19      #0
16      18 20 7E MOV A,M      #1
17      08 22 00      #0
18      08 22 55      #1
19      18 20 1A      #0

20      18 20 C9 RET      #1
21      08 23 FD      #0
22      08 23 0D      #1
23      08 23 FE      #0
24      08 23 20      #1

25      18 20 0D      #0
26      18 20 0D DCR C      #1
27      18 20 0E      #0
28      18 20 C2 JNZ      #1
29      08 20 0F      #0

```

Sequentiel trigning med 2 clockpulser, Delay=S, Data=S

Assemblerliste:

```

2000 3123FF START: LXI SP,23FFH ;Init SP
2003 212200      LXI H,2200H ;Init HL som RAM point
2006 0E03      IGEN: MVI C,3H ;Init tæller
2008 3655      HOP: MVI M,55H ;Udfør jobbet med 55
200A CD2019      CALL JOB ;Hop til subrutinen
200D 0D      JUSTE: DCR C ;Tæller -1
200E C22008      JNZ HOP ;Een gang til
2011 36AA      MVI M,AAH ;Udfør jobbet med AA
2013f CD2019      CALL JOB ;Hop til subrutinen
2016 C32016 LOOP: JMP LOOP ;Stop

2019* 7E      JOB: MOV A,M ;Opdater accumulatoren
201A C9      RET ;Job slut

```

Der skal triggles på subrutinen, men kun når den kaldes fra adresse 2013H. f = ARM-ord, * = TRIG-ord.

```

CK0.Q0.Q1.Q2=00XX CK1.Q0.Q1.Q2=10XX
ARM = XX 20 13 - CK0
TRIG= XX 20 19 D CK0
DATA=S DLY=00120S DISA 8085

```

```

LINE POD#2 #1 #0 #0
NO CK #0 #0 #0 #1
^

```

- 7	08 20 12	AA	Ved sequentiel trigning arbejder analysatoren med at armeringsord (ARM) og et triggerord (TRIG). ARM-ordet skal altid findes på CK0. Efter start søger analysatoren efter ARM-ordet. Når det findes, armeres TRIG-ordet der kan vælges til at skulle findes på CK0 eller CK1. Når trigordet findes tælles delayet ned af TRIG-ordets clock og dataopsamlingen går i stå.
- 6	10 22 00	AA	
ARM	18 20 13	CD CALL	
- 4	08 20 14	19	
- 3	08 20 15	20	
- 2	10 23 FE	20	
- 1	10 23 FD	16	
TRIG	18 20 19	7E MOV A,M	
1	08 22 00	AA	
2	18 20 1A	C9 RET	
3	08 23 FD	16	
4	08 23 FE	20	
5	18 20 16	C3 JMP	
6	08 20 17	16	
7	08 20 18	20	
8	18 20 16	C3 JMP	
9	08 20 17	16	
10	08 20 18	20	
11	18 20 16	C3 JMP	

Sequentiel trigning med 2 cp, delay mellem ARM og TRIG

Assemblerliste:

```

2000 3123FF START: LXI SP,23FFH ;Init SP
2003 212200 LXI H,2200H ;Init HL som RAM point
2006 0E03 IGEN: MVI C,3H ;Init tæller
2008 3655 HOP: MVI M,55H ;Udfør jobbet med 55
200A CD2019 CALL JOB ;Hop til subrutinen
200D* 0D JUSTE: DCR C ;Tæller -1
200E C22008 JNZ HOP ;Een gang til
2011 36AA MVI M,AAH ;Udfør jobbet med AA
2013 CD2019 CALL JOB ;Hop til subrutinen
2016 C32016 LOOP: JMP LOOP ;Stop

2019f 7E JOB: MOV A,M ;Opdater accumulatorens
201A C9 RET ;Job slut

```

Jeg vil trigge anden gang subrutinen kaldes fra sløjfen.

```

CK0.Q0.Q1.Q2=00XX CK1.Q0.Q1.Q2=10XX
ARM = XX 20 19 D CK0 223C
TRIG= XX 20 0D - CK0 2220
DATA=S DLY=000055 DISA 8085

```

LINE	POD#2	#1	#0	#0	#1
NO	CK	#0	#0	#0	#1
- 23	10	23	FD	0D	
ARM	18	20	19	7E	MOV A, M
- 21	08	22	00	55	
- 20	18	20	1A	C9	RET
- 19	08	23	FD	0D	
- 18	08	23	FE	20	
- 17	18	20	0D	0D	DCR C
- 16	18	20	0E	C2	JNZ
- 15	08	20	0F	08	
- 14	08	20	10	20	
- 13	18	20	08	36	MVI M
- 12	08	20	09	55	
- 11	10	22	00	55	
- 10	18	20	0A	CD	CALL
- 9	08	20	0B	19	
- 8	08	20	0C	20	
- 7	10	23	FE	20	
- 6	10	23	FD	0D	
- 5	18	20	19	7E	MOV A, M
- 4	08	22	00	55	
- 3	18	20	1A	C9	RET
- 2	08	23	FD	0D	
- 1	08	23	FE	20	
TRIG	18	20	0D		

Analysatoren armeres i starten af subrutinen på adr. 2019H, og trigges på den adresse der returneres til fra subrutinen (200DH). Delayet sættes til en sådan værdi, at TRIG-ordet først bliver aktivt efter første gennemløb af subrutinen. Derved forsinkes trigningen til anden gang 200DH forekommer. Da delayet anvendes mellem ARM- og TRIG-ordet vil der udelukkende være opsamlet "negative" data.

Quasiparallel trigning med TRIG-delay, Data=S

Assemblerliste:

```

2000 3123FF  START: LXI  SP,23FFH ;Init SP
2003 212200      LXI  H,2200H ;Init HL som RAM point
2006 0E03      IGEN: MVI  C,3H ;Init tæller
2008 3655      HOP:  MVI  M,55H ;Udfør jobbet med 55
200A CD2019      CALL JOB ;Hop til subrutinen
200D 0D      JUSTE: DCR  C ;Tæller -1
200E C22008      JNZ  HOP ;Een gang til
2011 36AA      MVI  M,AAH ;Udfør jobbet med AA
2013 CD2019      CALL JOB ;Hop til subrutinen
2016 C32016  LOOP: JMP  LOOP ;Stop

2019 7E      JOB:  MOV  A,M ;Opdater accumulatorens
201A* C9      RET ;Job slut

```

Jeg ønsker at trigge anden gang subrutinen returnerer til sløjfen.

```

CK0.Q0.Q1.Q2=00XX  CK1.Q0.Q1.Q2=10XX
TRIG=  XX 20 1A  Z CK0
TRIG=  XX XX C9  D CK1
DATA=S          DLY=00001T  DISA 8085

```

LINE	POD#2	#1	#0	#0	#1
NO	CK	#0	#0	#0	#1
- 2	18	20	19		7E MOV A,M
- 1	08	22	00		55
TRIG	18	20	1A		C9 RET
1	08	23	FD		0D
2	08	23	FE		20
3	18	20	0D		0D DCR C
4	18	20	0E		C2 JNZ
5	08	20	0F		08
6	08	20	10		20
7	18	20	08		36 MVI M
8	08	20	09		55
9	10	22	00		55
10	18	20	0A		CD CALL
11	08	20	0B		19
12	08	20	0C		20
13	10	23	FE		20
14	10	23	FD		0D
15	18	20	19		7E MOV A,M
16	08	22	00		55
17	18	20	1A		C9 RET

Som TRIG-ord vælges den adresse på hvilke RET-instruktionen ligger, nemlig 201AH. Da jeg ønsker at trigge anden gang subrutinen gennemløbes, indskydes et TRIG-delay på 1T. Det vil sige at analysatoren skal finde TRIG-ordet 1+delayet gange inden den stopper. I listen markeres den første gang TRIG-ordet optræder med "TRIG", og de efterfølgende linier angives positive!.

Sequentiel trigning med Zero-delay mellem ARM og TRIG

Assemblerliste:

```

2000 3123FF START: LXI SP,23FFH ;Init SP
2003 212200 LXI H,2200H ;Init HL som RAM point
2006 0E03 IGEN: MVI C,3H ;Init tæller
2008 3655 HOP: MVI M,55H ;Udfør jobbet med 55
200A CD2019 CALL JOB ;Hop til subrutinen
200D 0D JUSTE: DCR C ;Tæller -1
200E C2F2008 JNZ HOP ;Een gang til
2011* 36AA MVI M,AAH ;Udfør jobbet med AA
2013 CD2019 CALL JOB ;Hop til subrutinen
2016 C32016 LOOP: JMP LOOP ;Stop

2019 7E JOB: MOV A,M ;Opdater accumulatoren
201A C9 RET ;Job slut

```

Der skal triggles når programmet forlader sløjfen.

CK0.Q0.Q1.Q2 =00XX CK1.Q0.Q1.Q2=XXXX

ARM = XX 20 0F Z CK0 2232

TRIG= XX 20 11 D CK0 2234

DATA=S DLY=00250S SPECL

LINE NO	POD#2	#1	#0	CK
- 4	08	23	FE	#0
- 3	18	20	0D	#0
- 2	18	20	0E	#0
ARM	08	20	0F	#0
TRIG	18	20	11	#0
1	08	20	12	#0
2	10	22	00	#0
3	18	20	13	#0
4	08	20	14	#0
5	08	20	15	#0
6	10	23	FE	#0
7	10	23	FD	#0
8	18	20	19	#0
9	08	22	00	#0
10	18	20	1A	#0
11	08	23	FD	#0
12	08	23	FE	#0
13	18	20	16	#0
14	08	20	17	#0

Når programmet forlader sløjfen, er det karakteriseret ved at processoren kun henter den anden byte i JNZ-instruktionene inden den fortsætter til næste instruktion. Det vil sige at adr. 200FH vil komme umiddelbart inden 2011H. I denne mode kan der kun arbejdes med een clockpuls. Derfor bliver der kun opsamlet adresser hvilket giver denne mode nogen begrænsninger i sin anvendelse.

Quasiparallel trigning, opsamling af triggere, delay=T.

Assemblerliste:

```
2000 3123FF START: LXI SP,23FFH ;Init SP
2003 212200      LXI H,2200H ;Init HL som RAM point
2006 0E03      IGEN: MVI C,3H ;Init tæller
2008 3655      HOP: MVI M,55H ;Udfør jobbet med 55
200A CD2019      CALL JOB ;Hop til subrutinen
200D 0D      JUSTE: DCR C ;Tæller -1
200E C22008      JNZ HOP ;Een gang til
2011 36AA      MVI M,AAH ;Udfør jobbet med AA
2013 CD2019      CALL JOB ;Hop til subrutinen
2016 C32016      LOOP: JMP LOOP ;Stop

2019 7E      JOB: MOV A,M ;Opdater accumulatoren
201A C9      RET ;Job slut
```

Jeg ønsker at følge data til og fra adr 2200H.

```
CK0.Q0.Q1.Q2=00XX CK1.Q0.Q1.Q2=10XX
TRIG= XX 22 00 Z CK0 2200
TRIG= XX XX XX D CK1
DATA=T DLY=00007T DISA 8085
LINE POD#2 #1 #0 #0
NO CK #0 #0 #0 #1
^
```

- 10
- 9
- 8
- 7
- 6

- 5
- 4
- 3
- 2
- 1

TRIG	10	22	00	55
1	08	22	00	55
2	10	22	00	55
3	08	22	00	55
4	10	22	00	55
5	08	22	00	55
6	10	22	00	AA
7	08	22	00	AA

Denne mode kan bruges til at følge dataflowet til og fra en adresse, eller et adresseområde ved at lade TRIG-ordet indeholde dont' cares (x). Delayet tælles ned af triggerord. Ved at se på statusordet kan man se hvilken retning data har i de enkelte linier, idet 10H er en WR-cycle og 08H er en RD-cycle. Hvis delayet er sat større end antallet af triggerord melder analysatoren "NO TRIG-WORD(S)", og analysatoren skal stoppes manuelt.

Quasiparallel mode med FALSE trigning.

Assemblerliste:

```

2000 3123FF START: LXI SP,23FFH ;Init SP
2003 212200      LXI H,2200H ;Init HL som RAM point
2006 0E03      IGEN: MVI C,3H ;Init tæller
2008 3655      HOP: MVI M,55H ;Udfør jobbet med 55
200A CD2019      CALL JOB ;Hop til subrutinen
200D 0D      JUSTE: DCR C ;Tæller -1
200E C22008      JNZ HOP ;Een gang til
2011 36AA      MVI M,AAH ;Udfør jobbet med AA
2013 CD2019      CALL JOB ;Hop til subrutinen
2016 C32016      LOOP: JMP LOOP ;Stop

2019 7E      JOB: MOV A,M ;Opdater accumulatorens
201A C9      RET ;Job slut

```

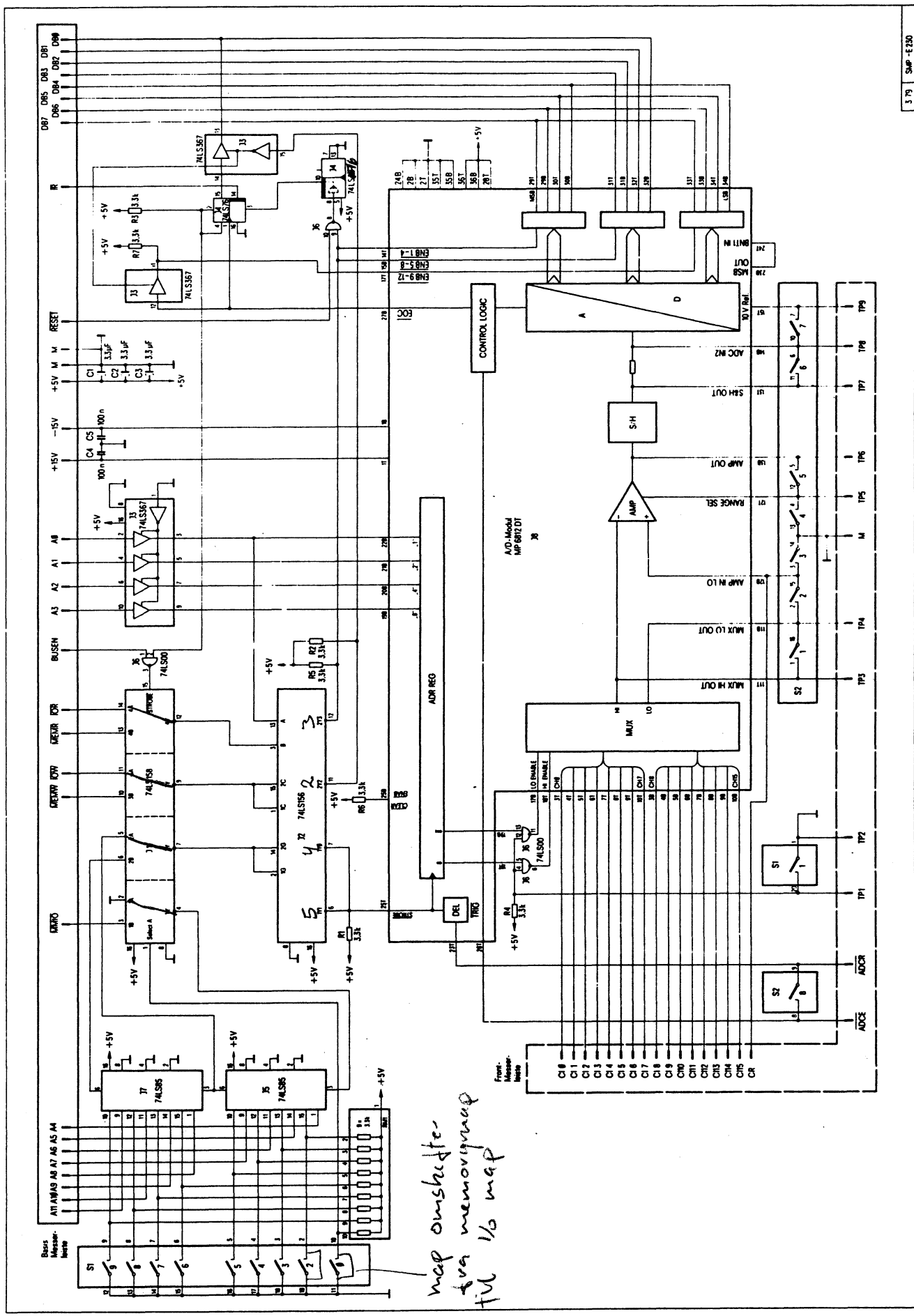
Jeg vil trigge hvis der i forbindelse med adresse 2200H forekommer noget andet end 55H.

```

CK0.Q0.Q1.Q2=00XX CK1.Q0.Q1.Q2=10XX
TRIG= XX 22 00 Z CK0
TRIG=F XX XX 55 D CK1
DATA=S DLY=00122S DISA 8085
LINE POD#2 #1 #0 #0
NO CK #0 #0 #0 #1
^

```

- 5		0D DCR C	FALSE trigning
- 4	18 20 0E	C2 JNZ	anvendes hvor man har
- 3	08 20 0F	08	mistanke om at der fore-
- 2	18 20 11	36 MVI M	kommer forkerte data i
- 1	08 20 12	AA	forbindelse med en be-
			stemt adresse. Som det
TRIG	10 22 00	AA	første TRIG-ord vælges
1	18 20 13	CD CALL	den adresse der skal
2	08 20 14	19	undersøges, og som det
3	08 20 15	20	andet TRIG-ord indsættes
4	10 23 FE	20	de korrekte data. Derefter
			indtrykkes FALSE i
5	10 23 FD	16	TRIGGER-gruppen. F'et
6	18 20 19	7E MOV A,M	foran det andet TRIG-ord
7	08 22 00	AA	viser at FALSE trigning
8	18 20 1A	C9 RET	er indkoblet. Analysato-
9	08 23 FD	16	ren vil nu trigge hvis
			der forekommer noget
10	08 23 FE	20	andet end 55H sammen med
11	18 20 16	C3 JMP	adressen 2200H.
12	08 20 17	16	
13	08 20 18	20	
14	18 20 16	C3 JMP	



Addr
EO-
EF

C register kanal nr
DATA 1 HL register

Out EO til EF

Vent IN EO

ANI 01 klar Z=0 ikke klar = 1

JZ Vent

IN EO

MOV L, A

IN EI

MOV H, A

A/D converter

16 indgange multiplexer 12 bit output
 konverterings tid under 50 μ sek
 multiplexer settling time ca 7 μ sek
 mon også høre 10V \pm 10V og \pm 5V

- 1/ Vælg Kanal aktiver strobe med negativ flanke når den er færdig aktiver
- 2/ End of conversion b
- 3 Hent Buffer data
 ENB 1-4, ENB 5-8, ENB 9-12

	IOW	IOR	A0	
	C	B	A	
Start F0-EF	1	0	0	Næste kanal/ Aktiver strobe
Loeseqdd E0 (lige adr)	0	1	0	Konvertering klar EOC på LSB
Loese E0	0	1	0	Henter MSB <div style="border: 1px solid black; display: inline-block; padding: 2px;"> MSB 1 </div> EOC
Loese E1 (ulige adr)	0	1	1	Henter 1 bit 8 og resetter og starter ny konvertering

DATA-memory og COMPARE-mode.

Foruden dataopsamlingsshukommelsen indeholder analysatoren en "baggrundshukommelse" i hvilken der kan gemmes en dataliste fra en tidligere dataopsamling.

Ved at indtrykke DATA i SELECTION/execute feltet vises indholdet af baggrundshukommelsen på skærmen. Indtrykkes samtidigt X/EXCH overføres indholdet af dataopsamlingshukommelsen til baggrundshukommelsen. Det er på den måde muligt at gemme en tidligere indspilning til sammenligning med nye indspilninger.

Denne sammenligning kan analysatoren selv foretage ved at indtrykke knappen COMP (compare) i DISPLAY gruppen. Derved sammenlignes indholdet af de to hukommelser og resultatet meddeles med et EQ (equal) eller NEQ (not equal) samt ved øget lysstyrke i tabellen hvor forskellene findes.

Analysatoren kan sættes i en "baby-sitter mode" hvor den kontinuært starter dataopsamling og sammenligner med referencen. Så længe der er lighed restarter analysatoren blot, men stopper når der registreres en forskel. Denne mode indkobles ved at indtrykke knapperne COMP og AUTO/MAN og til slut START.

Når alt svigter:

[illegible]

Børge Callesen

Logic State Analyser

50 kanaler

Multiplex

antal klokkes

7 stk Digital

Philips

HP

21 kanaler

50 kanaler

2 klokkes

4 klokkes

Hukommelse

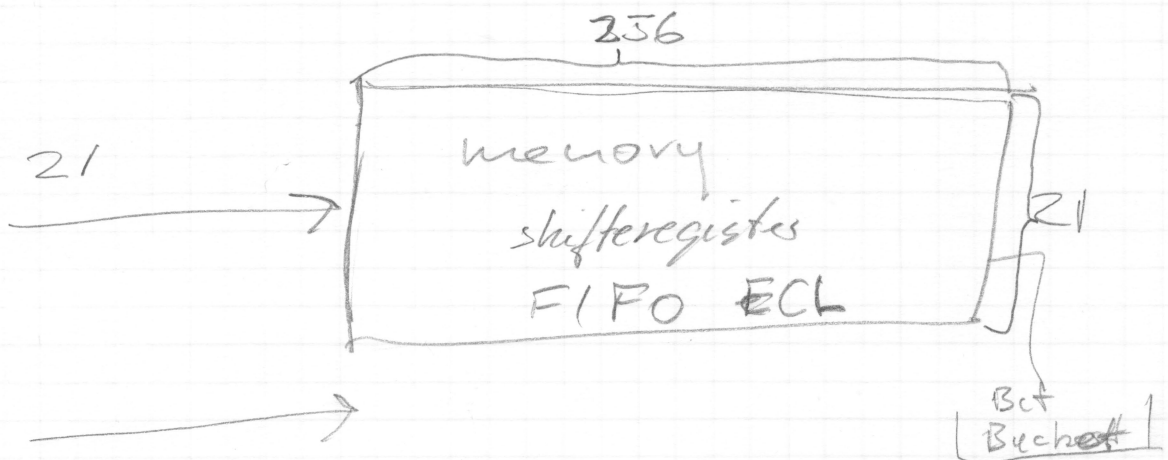
Dokumentation af software

Brug LSA

til udprintning af software

hvis der er

desimulator



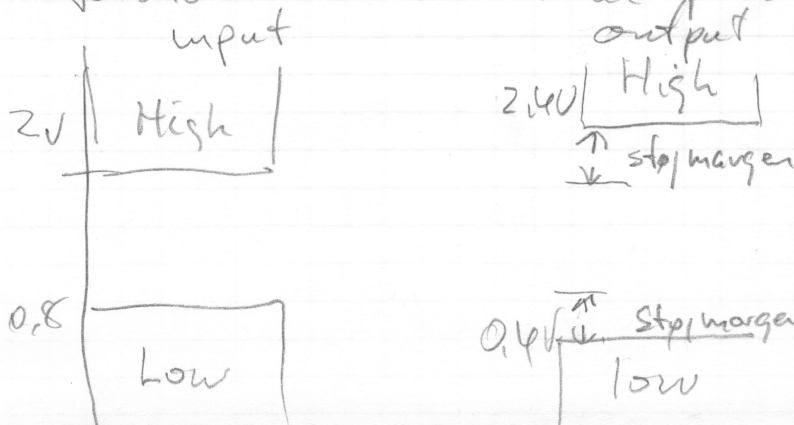
trig stopper indlæsning

trig delay antal clock pulser før stop

Ca 4M μ s/bp SUT System Under Test
UUT Unit

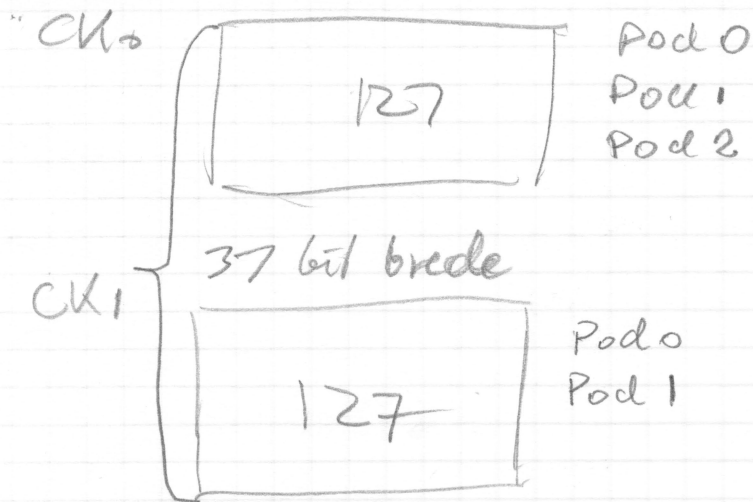
sovere de til fan in på ca 2

forbuds omrøde skal possesses på under 20ns



1,4V Analy.

Z = Zero delay mellem Clock 0 og Clock 1
adr data

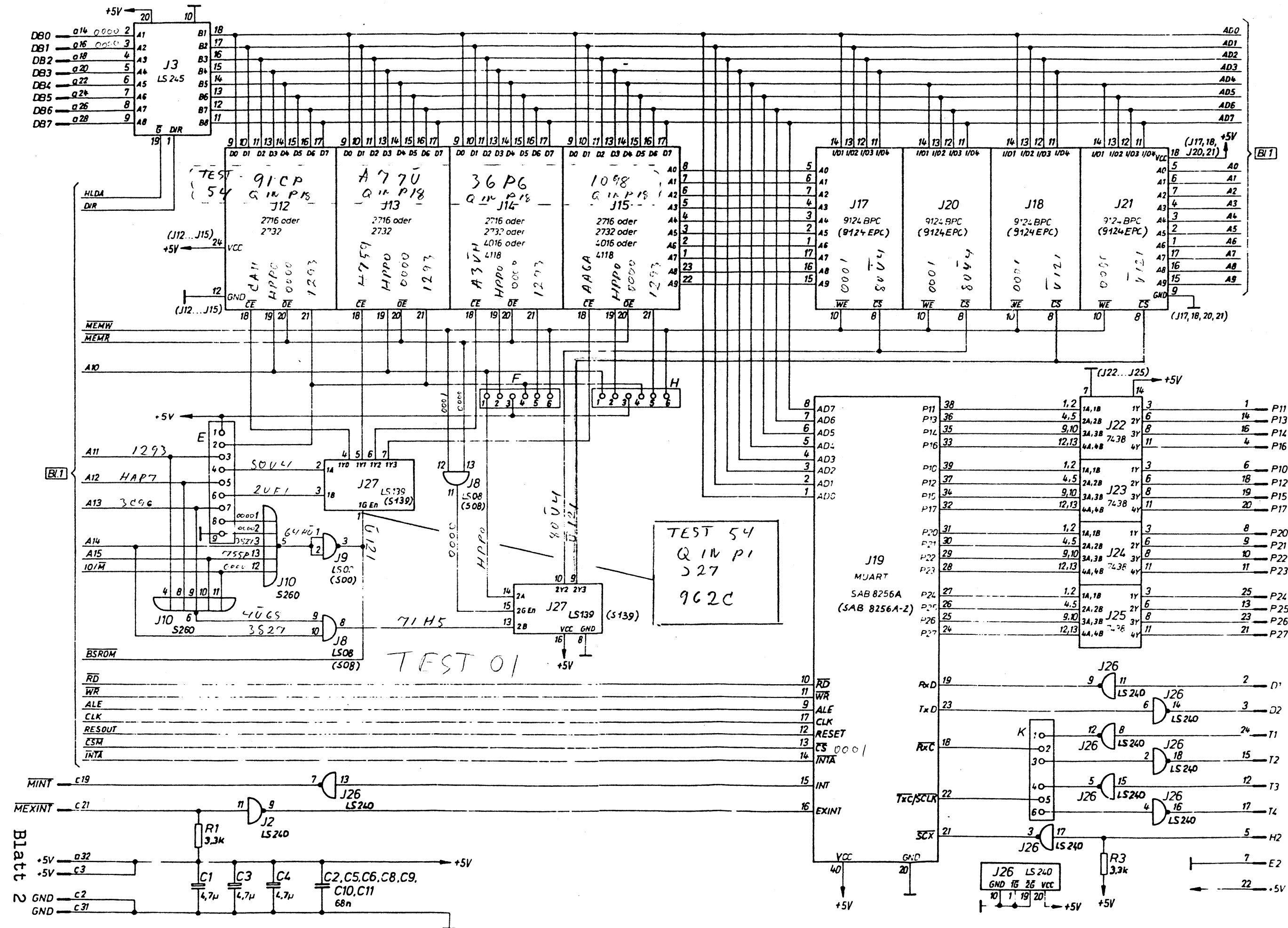


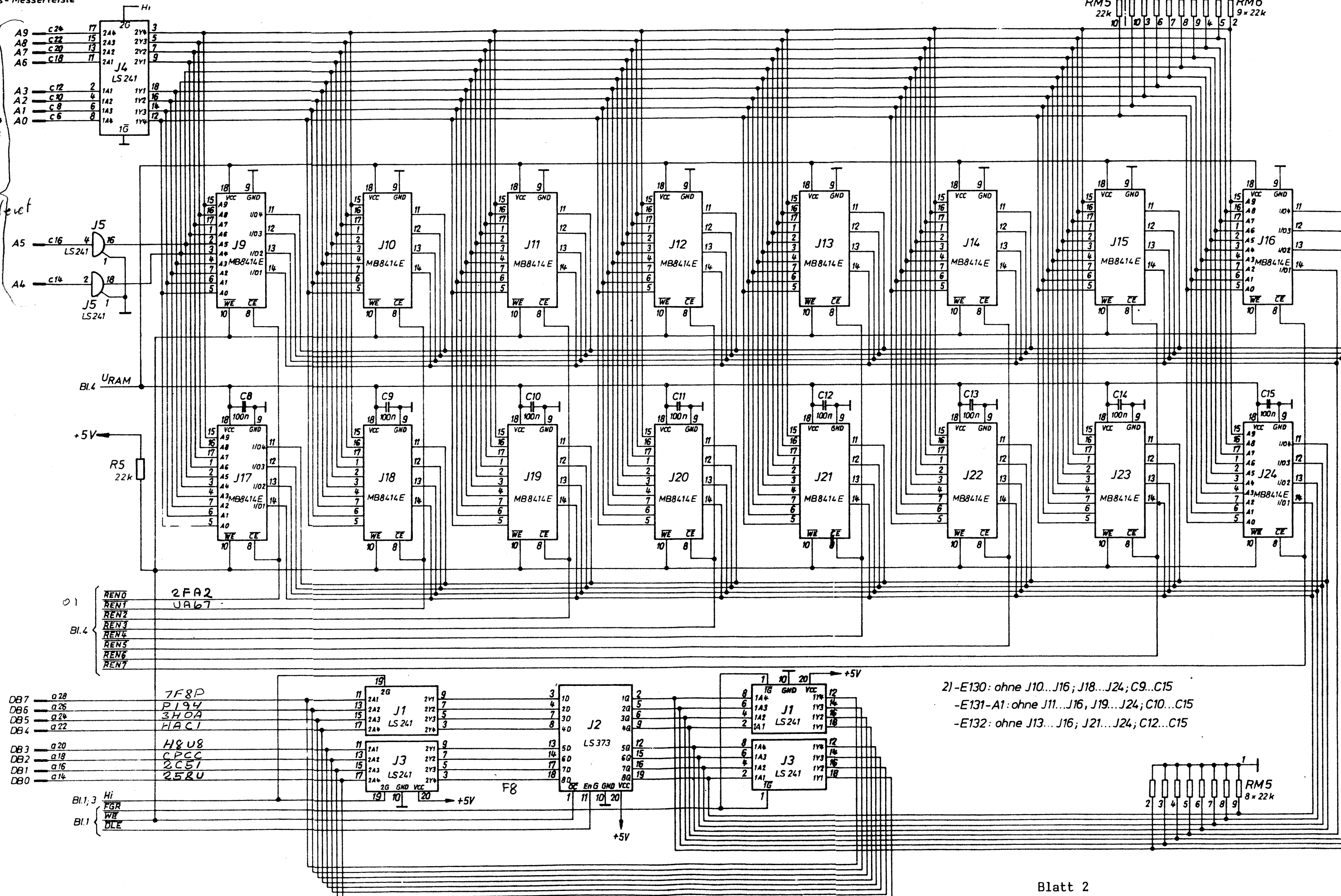
D = delay efter Clock 1
Sequentielt

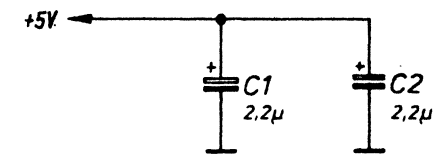
- ingen delay lige gyldig hvor langt til
der er T mellem de to adresser

Z = Zero delay mellem Clock 0 og Clock 0
clock skal være ved
arm og Trigger skal være umiddel-
bart efter hinanden

kan trigge på falske data

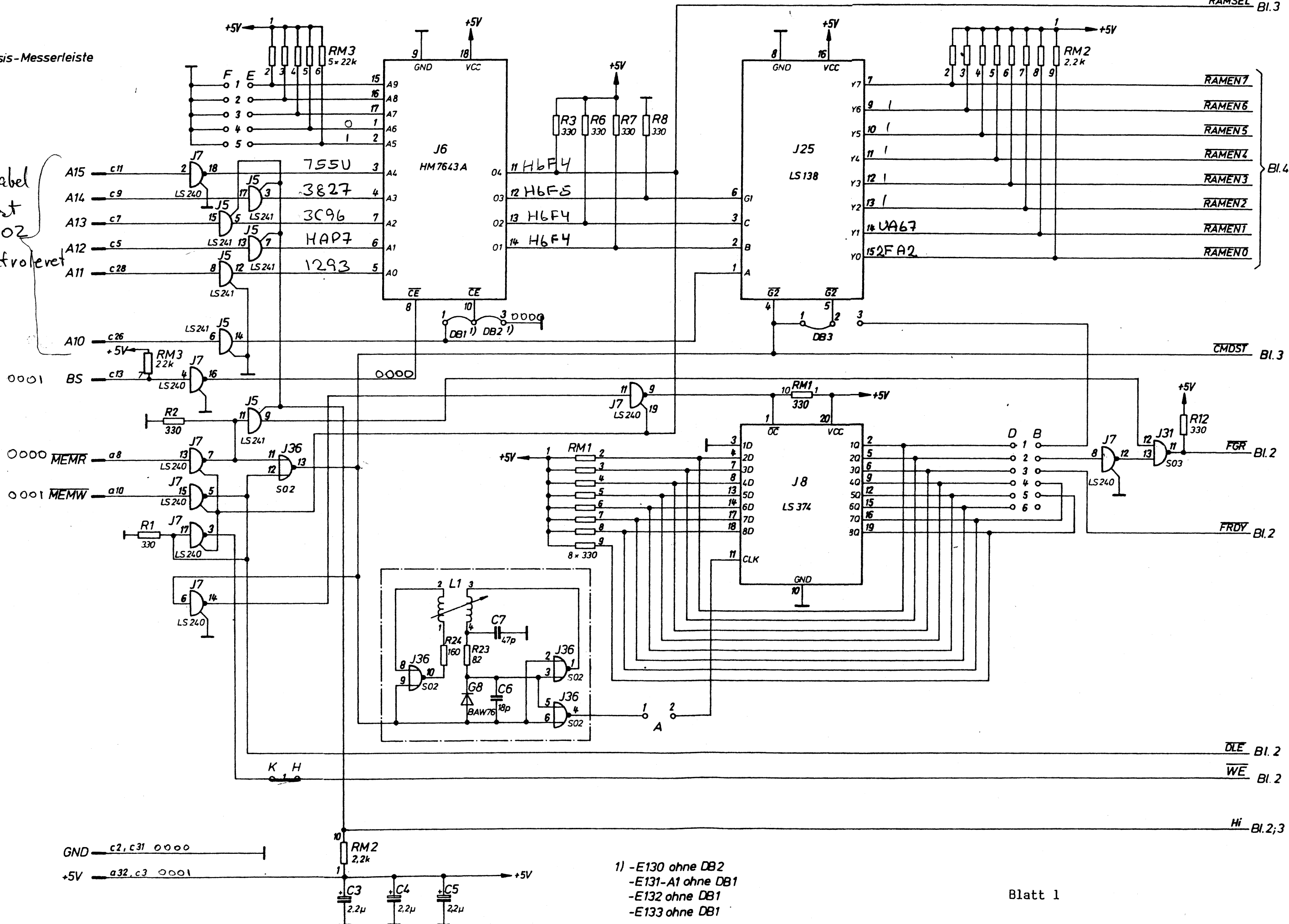




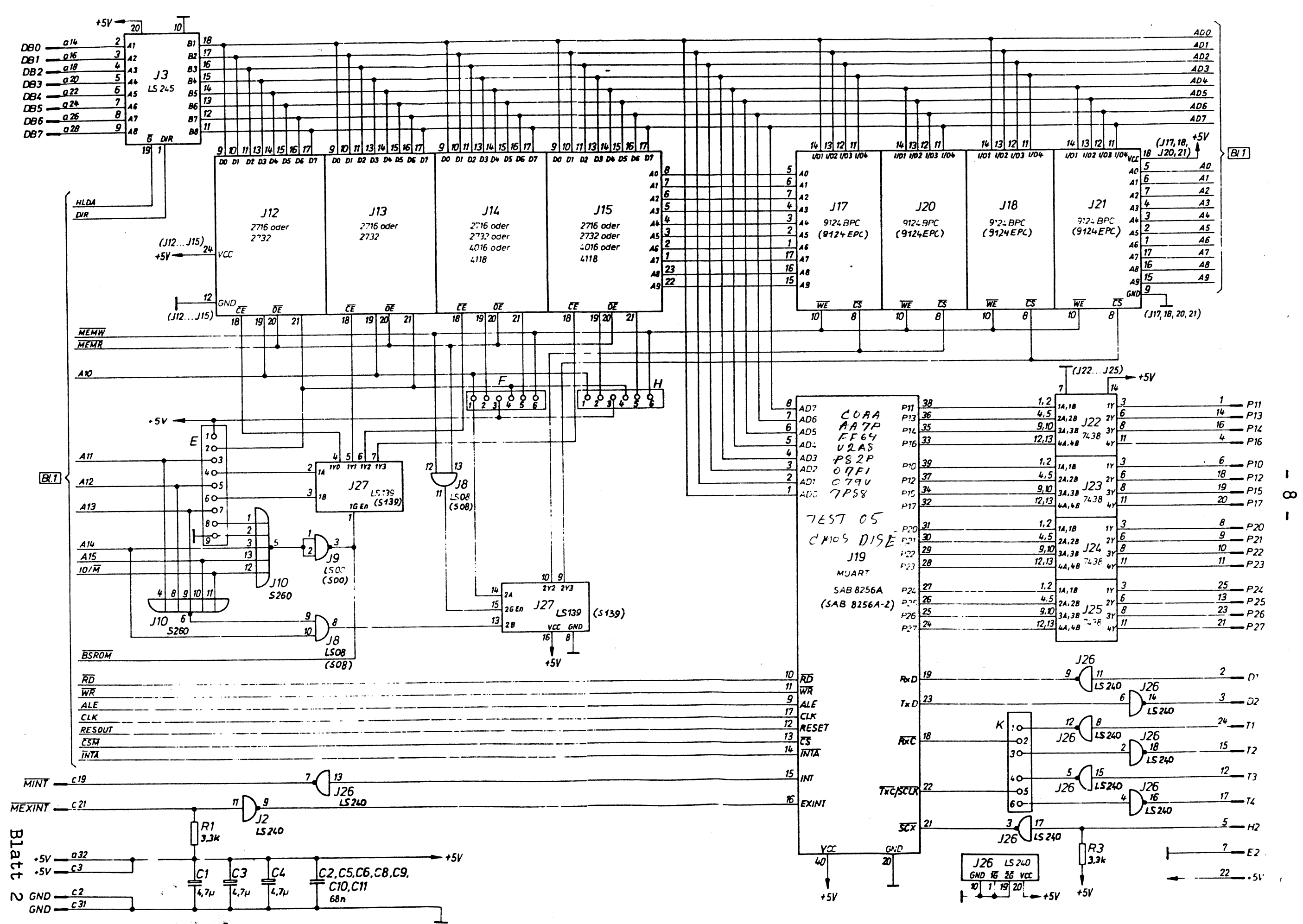


Basis-Messerleiste

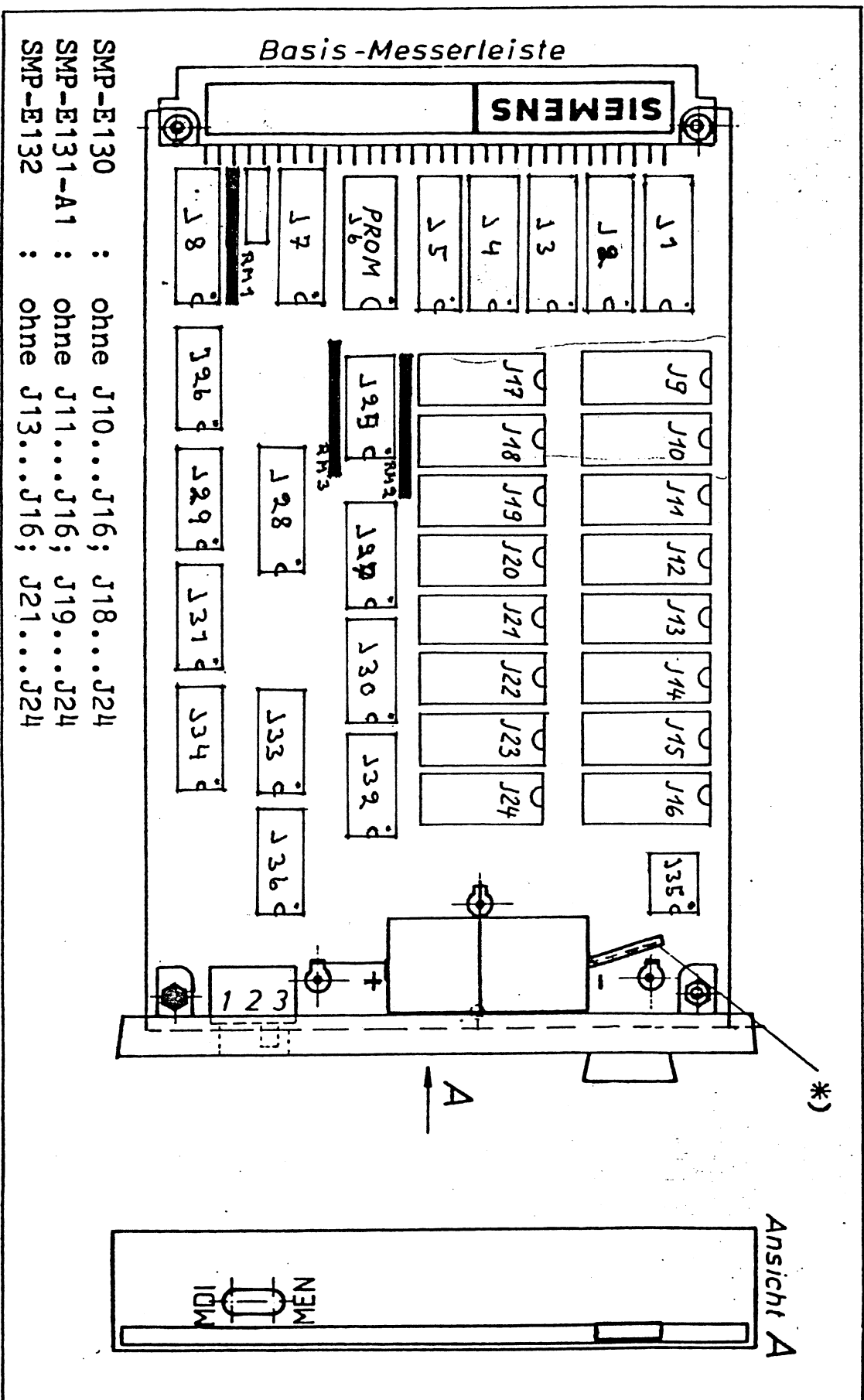
setabel
Test
01 02
Kontrollere
09
0



- 1) -E130 ohne DB2
- E131-A1 ohne DB1
- E132 ohne DB1
- E133 ohne DB1



Lage der Bausteinfassungen und der Batterie auf der Baugruppe



*) Vor Inbetriebnahme Isolierschlauch abziehen und Drahtende an Lötöse anlöten.

----- Oscilloscop målinger -----

Inden for den "almindelige" digitalteknik er oscilloscopet et uundværligt måleinstrument. Det kan bruges til kontrol af tider og spændinger, men med den betingelse at der er tale om et reppeterende signal af ikke for lang varighed. Det er netop på dette punkt der kan vise sig problemer når man vil anvende oscilloscopet til målinger omkring en mikroprocessor, idet signalernes reppetitionstid kan være meget lang, og tilmed varierende. Alfa og omega ved oscilloscopmålinger i en mikrocomputer er således at finde et velegnet triggersignal til oscilloscopet.

Almindeligt oscilloscop:

Et almindeligt oscilloscop er anvendeligt til at få et indtryk af aktiviteten og spændingsniveauerne på de forskellige busser og kontrolledninger. Ud over dette er det mest egnet til at måle på reppeterende forløb som for eksempel clocksignalet eller små programsløjfer. Ved lange programsløjfer hvor man ønsker at iagttage et bestemt programstyret fænomen, er det ofte et problem at finde et signal til trigning af oscilloscopet, samt at lysstyrken kan blive temmelig lille ved en lav reppetitionsfrekvens. Hvis oscilloscopet er udstyret med variabel hold-off tid, kan denne i mange tilfælde bruges til at opnå et entydigt billede.

Storage oscilloscop:

Med et storage oscilloscop, enten traditionelt med storagerør eller digital storage, kan man overkomme en del af de problemer der er forbundet med at anvende oscilloscopet til måling på programstyrede hændelser, idet man med storage oscilloscopet ikke behøver et reppetitivt forløb for at foretage en måling. Derimod består der de samme problemer med trigningen som ved et almindeligt oscilloscop. En speciel feature ved det digitale storage oscilloscop gør det muligt at flytte triggerpunktet ind på skærmen hvilket bevirker at man kan iagttage fænomener der ligger inden triggertidspunktet.

Delayed sweep:

Hvis man har adgang til et oscilloscop med delayed sweep kan det med fordel benyttes til gennemførelse af målinger på microcomputere. Ved at anvende delayed sweep får man større frihed med hensyn til valg af triggerpuls, idet denne kan ligge langt før det punkt i datastrømmen man ønsker at iagttage. Ydermere har man mulighed for, med en simpel justering af delayet, at flytte billedet frem og tilbage i datastrømmen.

----- Trigning af oscilloscopet -----

For at opnå et stabilt billede, samt et billede af det man forventer at se er det vigtigt at vælge/finde et velegnet signal til trigning af oscilloscopet. I det følgende er forslag til triggersignaler hvis det man vil iagttage har relation til maskin-cycles, eller program-styrede hændelser som I/O ,chip-select og interrupt.

- CHIP-SELECT signal, eventuelt gated sammen med et READ eller WRITE signal. Denne triggerpuls er nyttig ved kontrol af dataoverførslen mellem CPU'en og hukommelse eller portkredse. Måske findes det sammengatede signal allerede, ellers må man fremstille et lille kredsløb der udfører den nødvendige logiske funktion, f. eks. ved hjælp af wire wrap på et stykke weroboard.

- ADRESSE-BIT, kan anvendes til trigning, idet man finder en adresseledning (hvis den findes) der udfører et skift på det tidspunkt hvor man ønsker at måle, eller kort forinden. For at få så veldefineret trigning som muligt skal man så vidt muligt finde et adressebit med det højest mulige nummer, idet antallet af skift i almindelighed vil aftage jo højere man kommer op i adresseledningerne. Hvis det viser sig at den adresseledning der skal anvendes er en af de otte laveste, må man tage triggersignalet på udgangen af en eventuel adresselatch. Her skal man dog være opmærksom på at der ofte forekommer en del hazardpulser som vil forårsage trigning i utide. Se senere i eksempel 3.

-INT, INTA, TRAP og RST. Ved kontrol af interruptstyrede hændelser, vil det være naturligt at anvende interruptsignalet til trigning af oscilloscopet. Problemet ved iagttagelse af et interruptforårsaget fænomen, er at tiden der går fra CPU'en har modtaget INT-signalet, til den reagerer med et INTA-signal, eller ved at hoppe til restart adressen er bestemt af længden af den instruktion der tilfældigvis bliver eksekveret når interrupten kommer. Problemet løses mest effektivt ved at anvende et storageoscilloscop.

----- Digital trigning -----

Da hændelserne i en computer er nært knyttede til programmet og dermed adresserne fra programcounteren, vil det være naturligt at anvende disse til trigning af oscilloscopet. Dertil skal der bruges en digital comparator, der sammenligner adressen på adressebussen med en forudindstillet værdi, og genererer en triggerpuls til oscilloscopet når der er overensstemmelse mellem de to input til comparatoren. På denne måde kan man få oscilloscopet trigget af programforløbet. En sådan comparator kaldes også en "Ordgenkender" eller en "Triggeranalysator". Som bekendt indeholder en logik state analysator som en væsentlig funktion en triggeranalysator, hvorfor LSA'ere da også i almindelighed er udstyret med et tilgængeligt output fra triggeranalysatoren der kan bruges til at trigge et oscilloscop.

----- Philips Logic Scope, PM3543 -----

I PM3543 kan oscilloscopet "ordtrigges" ved blot at trykke på knappen mærket "WORD" i oscilloscopets time-baseafsnit. Derved tages triggerpulsen fra triggeranalysatorens output. Når man vil anvende ordtrigning, starter man med at sætte analysatoren i stilling "LSA". Derefter sætter man det ord op der ligger een clockpuls før det man ønsker at iagttage, idet triggerpulsen genereres een clockpuls efter triggerordet er fundet. Delayet fjernes. Skiftes der nu til stilling "OSC" vil oscilloscopet blive trigget hver gang det valgte ord findes, samt eventuelle kvalifysere er opfyldt. Hvis der er langt mellem at det valgte triggerord forekommer kan det blive et problem at få lys nok på skærmen. Hvis man har adgang til et storageoscilloscop kan dette anvendes, idet triggerpulsen er tilgængelig via et BNC-stik på PM3543's bagside.

TRIG OUT: TTL low til high skift.

----- Eksempler -----

I det følgende er vist nogle eksempler på målinger foretaget på PIV-1 ved hjælp af oscilloscop. Målingerne er foretaget med PM3543 og Philips digitale storageoscilloscop PM3310, idet oscillogrammerne er tegnet på en X-Y skriver styret fra X-Y udgangene på PM3310.

På grund af PM3310's begrænsede opløsning forekommer der nogen forvrængning ved gengivelsen af clocksignalet på 3,072 MHz, ligesom stige- og faldetiderne er fortegnede.

Eks. 1 Kontrol af READ-cycle på RAM-hukommelse.

Se diagram af CPU-kort. CHIP-SELECT signalet til RAM-hukommelsen er gated med MEMR og MEMW og decodet i J27. Den kritiske tid er overlappet mellem RD pulsen målt på CPU'en og CS-pulsen målt på RAM-hukommelsen. Ifølge databladet for 8085A skal data være stabile i mindst 100 nsek inden stigende flanke af RD, hvilket medfører at overlappet mellem RD og CS mindst skal være summen af RAM-hukommelsens accesstid + 100 nsek.

$$\text{Overlap RD/CS} = 450 \text{ n} + 100 \text{ n} = 550 \text{ n sek}$$

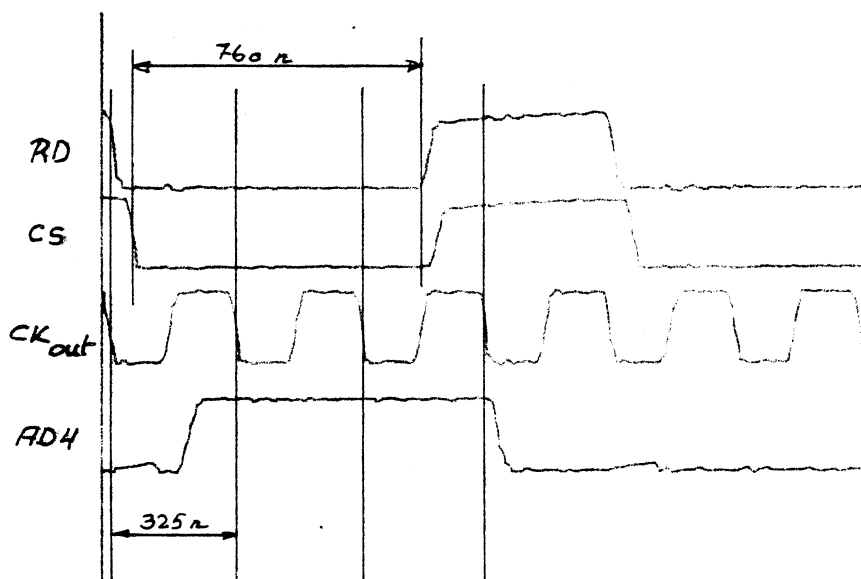
Målingen blev udført med PM3543. Programsløjfen var 5 instruktioner lang, og indeholdt 4 readcycler på RAM-hukommelsen, se eksempel 5 linie 1 og 10 i LSA listen.

Kanal A: RD fra CPU, pin 32.

Kanal B: CS på RAM-hukommelsen, J17 pin 8.

Trigger: Kanal B, neg.

Til sammenligning er CLK85 bagplan a4, og AD4 CPU pin 16 vist i oscillogrammerne.



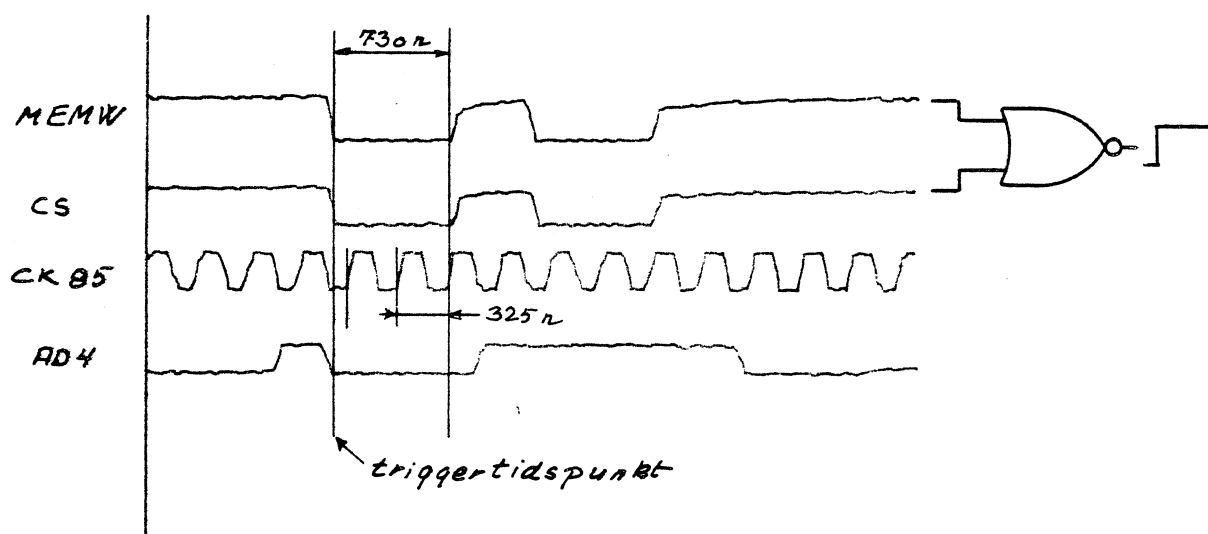
Eks. 2 Kontrol af WRITE-cycle på RAM-hukommelse.

Se diagram af CPU-kort. Den kritiske tid ved en WRITE-cycle er længden af CS/WR overlappet samt forholdet mellem dette og data fra CPU'en. For de aktuelle RAM-kredse (Am 9124 BPC) er overlappet, t_W opgivet til min 250 nsek, og data skal være stabiliseret mindst 200 nsek inden afslutningen af en write-cycle.

Målingen blev udført med PM3543 og dokumenteret på PM3310. På grund af sidstnævntes mulighed for negativt delay ligger triggerpunktet ca. 3 div. inde på skærmen.

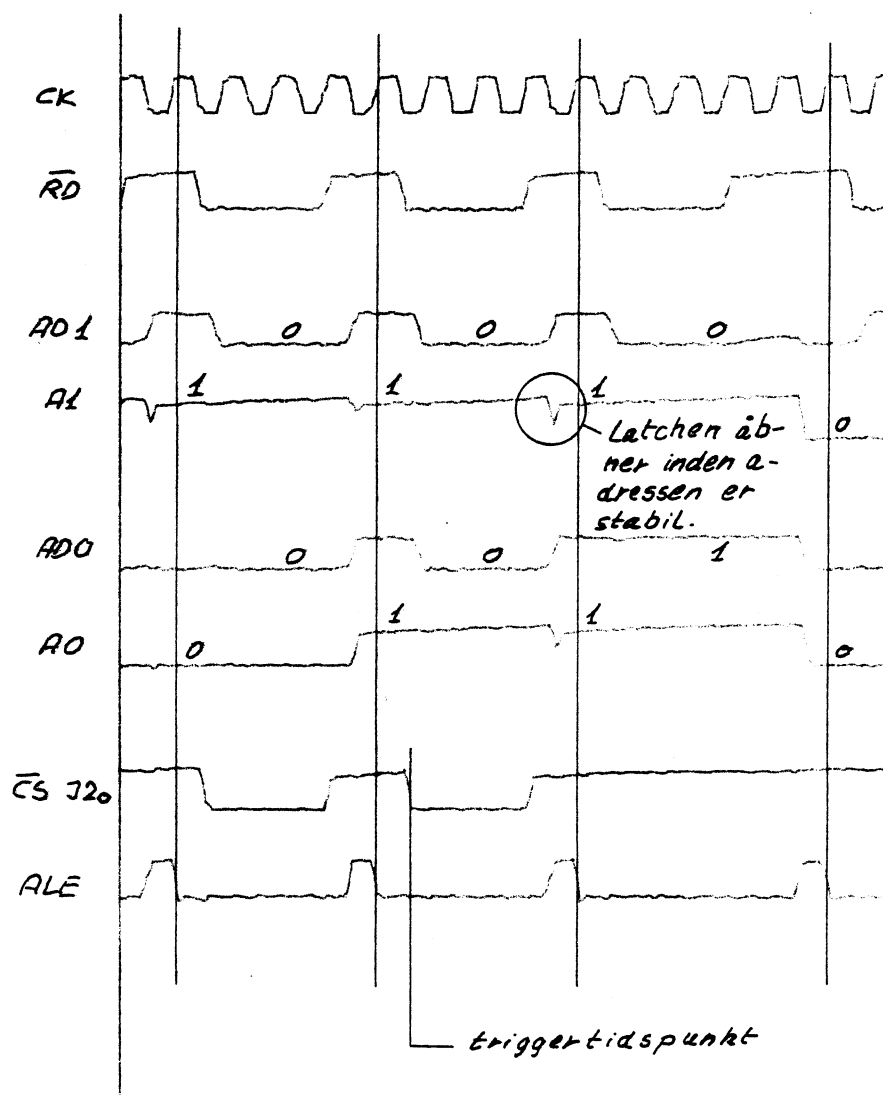
Kanal A: MEMW på RAM, J17 pin 10.
Kanal B: CS på RAM, J17 pin 8.
Kanal C: CLK85, bagplan a4
Kanal D: AD4, CPU pin 16

Trigger: For med sikkerhed at trigge oscilloscopet på en write-cycle på RAM-hukommelsen blev oscilloscopet triggeret med et signal der var gated sammen af MEMW og CS. Som gate blev anvendt SN74S27, og der blev triggeret på den pos. flanke.



Eks. 3 Kontrol af Adresselatchens funktion.

Se diagram af CPU-kort. Målingen blev foretaget medens PIV-1 kørte en programsøjle på 5 instruktioner. Se LSA listen til eks. 5. Der blev triggeret ved hjælp af CS-signalet til J20 pin 8. Målingen blev foretaget med PM3543, idet kanal A konstant viste ALE, medens kanal B blev flyttet mellem de øvrige signaler. De viste oscillogrammer er optaget med PM3310, med triggerpunktet liggende 4 div. inde på skærmen. På X-aksen dækker oscillogrammerne fra linie -2 til 1 i LSA listen til eks. 5.



Eks. 4 Måling af forsinkelse mellem INT og INTA.

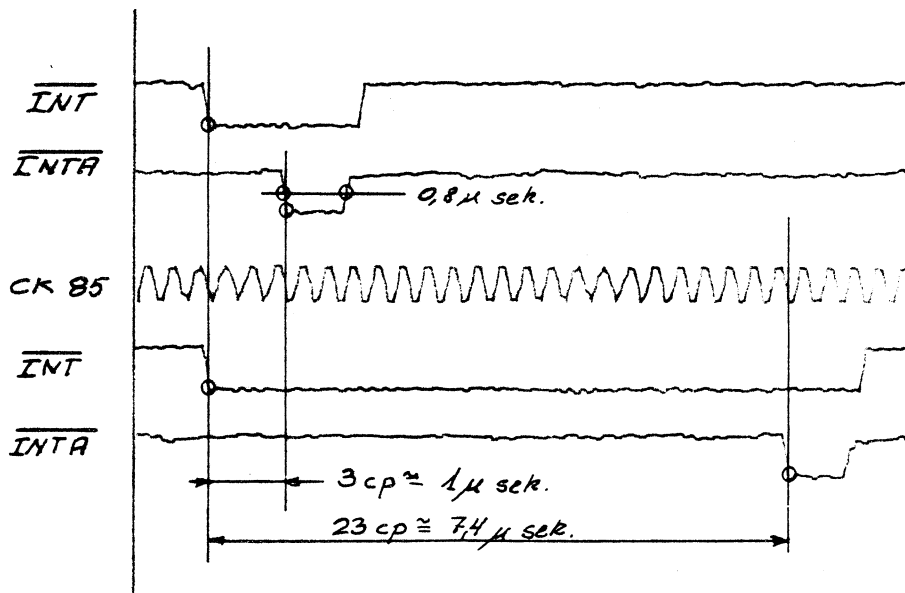
Målingen blev foretaget ved hjælp af PM3310 digitalt storage oscilloscop, idet reppetitionsfrekvensen er så lav at der ikke vil være lys nok på billedet hvis målingen gennemføres med et almindeligt oscilloscop. Hvis det er muligt at generere et interrupt kunstigt med en højere frekvens, kan målingen gennemføres med et normalt oscilloscop, dog med et noget ustabilt billede til resultat idet forsinkelsen mellem INT og INTA er bestemt af længden af den instruktion der tilfældigvis eksekveres.

Målingen viser hvor lang tid der kan gå fra MUART 8256 har genereret en INT, til CPU'en svarer med INTA.

Kanal A: INT i bagplan nr. a19.
Kanal B: INTA i bagplan nr. a23.
Kanal C: CLK85
Kanal D: INT i bagplan nr. a19.
Kanal E: INTA i bagplan nr. a23.

Trigger: INT neg. flanke, delay -1

De øverste spor viser den hurtigste respons på en INT, og de nederste spor den langsomste respons på en INT.



Eks. 5 Kontrol af adresse- og dataflow på AD-bussen.

Målingen blev gennemført med PM3543, idet der blev anvendt ordtrigning (WORD). PIV-1 kørte i en relativ kort programsøjle som det ses af LSA-listen, linie -5 til 9. Ved optagelsen af oscillogrammerne var LSA'en sat op som følger:

CK0 : ALE faldende flanke
 CK1 : RD stigende flanke
 Mode : quasi-parallel
 Delay : Ingen (deselected)
 Ord : 03AB H

Ved at sammenligne oscillogrammerne og LSA-listen kan man se at trigningen af oscilloscopet finder sted en clockperiode (ALE) senere end fremkomsten af triggerordet, hvilket man skal have i tankerne ved valg af triggerord.

LSA-liste:

67 18 03 B0
 VERSION 4301-S01-DA01

CK0.Q0.Q1.Q2=00XX CK1.Q0.Q1.Q2=10XX
 TRIG= XX 03 AB Z CK0
 TRIG= XX XX XX D CK1
 DATA=S DLY=00120S DISA B0B5

LINE	POD#2	#1	#0	#0
NO	CK	#0	#0	#1

- 8				E5 PUSH H
- 7	10	47	F9	
- 6	10	47	F8	
- 5	18	03	A8	2A LHLD
- 4	08	03	A9	E6
- 3	08	03	AA	40
- 2	08	40	E6	F0
- 1	08	40	E7	40

TRIG				
	18	03	AB	7D MOV A,L
1	18	03	AC	2A LHLD
2	08	03	AD	EA
3	08	03	AE	40
4	08	40	EA	F0
5	08	40	E9	40
6	18	03	AF	BD CMP L
7	18	03	B0	CA JZ
8	08	03	B1	A8
9	08	03	B2	03
10	18	03	A8	2A LHLD

LSA-triggerord

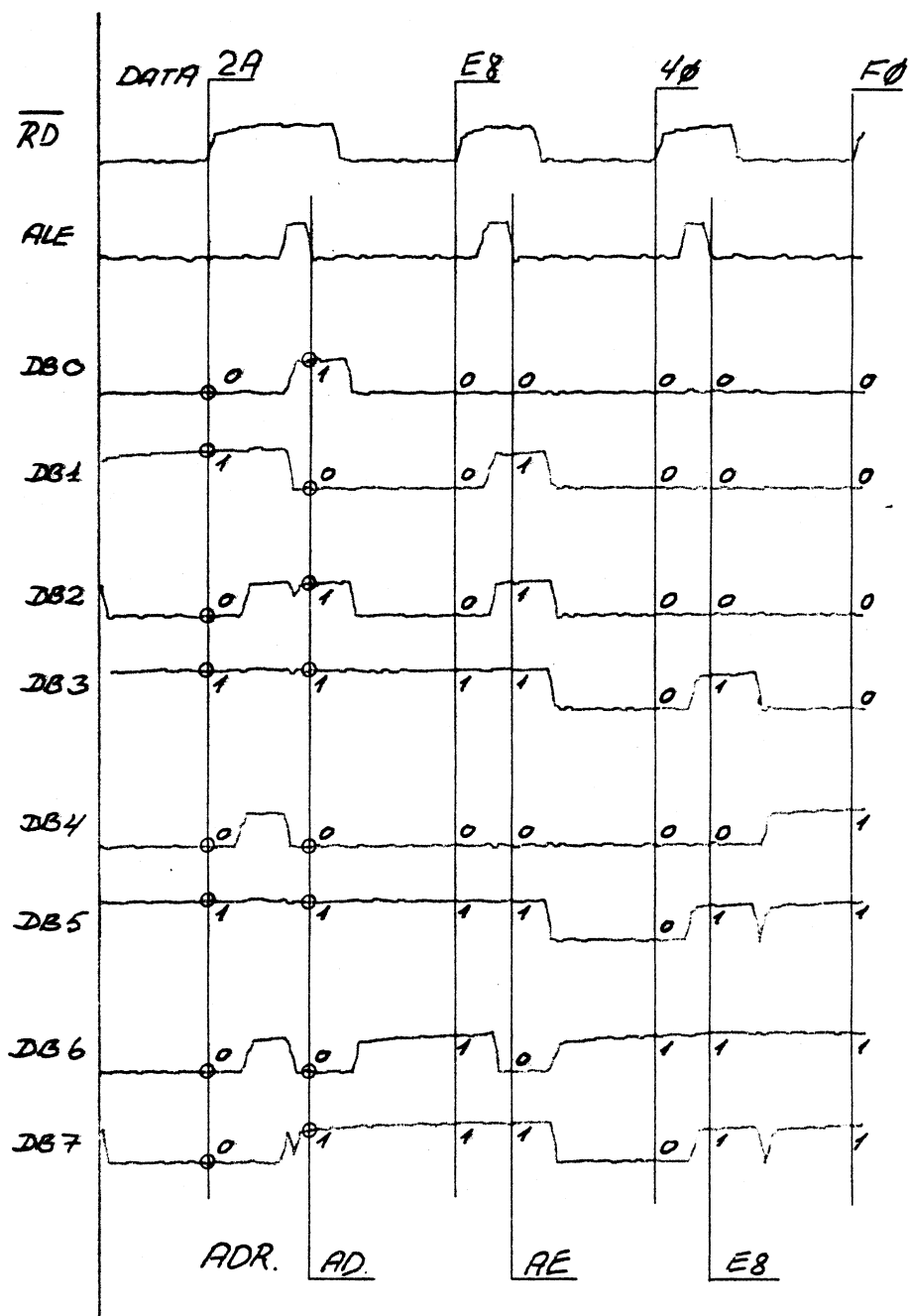
Første adr. på skærmen, idet oscilloscopet triggeres på CK0 (ALE) efter triggerordet er fundet.

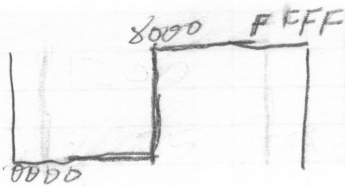
Oscillogrammer:

Kanal A: MEMR, bagplan a8

Kanal B: ALE, bagplan a7

Kanal C - J: DB0 - DB7, bagplan a14, 16, 18, 20,
22, 24, 26, 28.





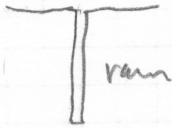
Brug jne ram sofhed

test af chip select logik

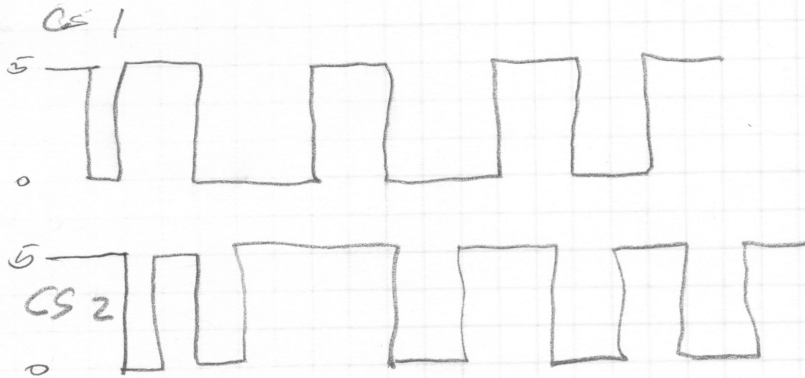
ved hjælp af oscilloscope

soep hastighed ca 10 kHz

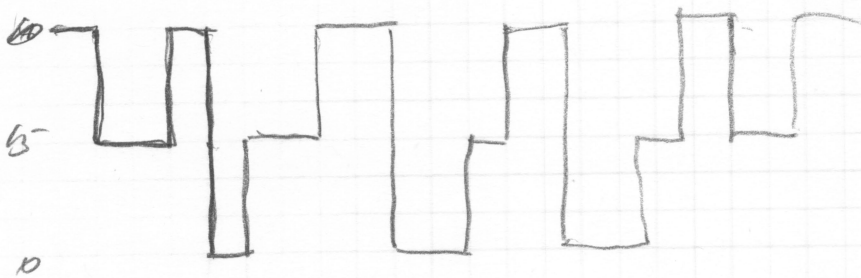
CS 2K taste eprom



Anvend scope uden udfagbar CPU



addres de to
kanaler



ved 0 Volts følger
vil to CS udsiges
samtidig

----- LOGIK PROBER -----

Logik proberne omfatter tre medlemmer der tilsammen udgør et effektivt værktøj til fejlfinding på digitale kredsløb.

Sættet består af:

Logisk niveau probe,.HP type 545A
Logisk pulser,.....HP type 546A
Current tracer,.....HP type 547A

Logik probernes fordel ligger først og fremmest i deres små dimensioner,-at de strømforsynes fra det undersøgte apparat, og måleresultatet fremkommer på målestedet.

----- Logisk niveau-probe, HP 545A -----

Anvendelse:

Ved hjælp af den logiske niveauprobe kan man kontrollere om spændingerne i det undersøgte kredsløb overholder de niveauer der er gældende for den pågældende logiske familie. Dette forudsætter dog at den anvendte logikprobe kan omkobles til at dække flere logikfamilier. Den her behandlede probe, HP 545A kan anvendes til TTL- og CMOS-kredse.

Som alternativ til voltmeter eller oscilloscop har logikproben den fordel at måleresultatet præsenteres direkte på målestedet i form af mere eller mindre lys i en indikatorlampe.

Tilkobling:

Logikproben skal tilsluttes en forsyningsspænding der ligger mellem 4,5v og 15v DC ved anvendelse på TTL logik, og 3v til 18 v ved CMOS-logik. Det nemmeste er at tilslutte proben til forsyningsspændingen på det apparat der måles på, under forudsætning af at dette kan klare en ekstra belastning på max 70 mA.

Hvis man vælger at anvende en separat spændingsforsyning skal der forbindes en stelledning mellem logikproben og det undersøgte apparats stel.

Ved tilslutning til et kredsløb der indeholder CMOS-kredse vil det nemmeste være at anvende det undersøgte apparats forsyningsspænding idet probens tærskel-spændinger da automatisk vil være korrekte.

Tærskelspændinger:

	TTL	CMOS
Logisk 1	2.0 +0.4/-0.2Vdc	0.7xVss +/-0.5Vdc
Logisk 0	0.8 +0.2,-0.3Vdc	0.3xVss +/-0.5Vdc

Funktion:

Sammenhæng mellem inputsignal og indikation:

INPUT	INDIKATION
Spænding under low-level	Indikatorlampe slukket
Spænding mellem low-level og high level.	Indikatorlampe lyser med halv styrke.
Spænding over high-level	Indikatorlampe lyser med fuld styrke
Input er i 3-state	Indikatorlampe lyser med halv styrke.
Inputfrekvens lave-end 10 Hertz	Indikatorlampen blinker i takt med inputsignalet.
Inputfrekvens høje-re end 10 Hertz	Indikatorlampen blinker med 10 Hertz takt.
Enkeltpuls ned til 10 n sek	Indikatorlampen tænder/slukker i 50 m sek.
Åben gateindgang	Indikatorlampe lyser med halv styrke.

Pulshukommelse:

Proben er forsynet med pulshukommelse, således at den kan indikere eet skift mellem high og low eller omvendt, samt et skift fra et gyldigt logisk niveau til

undefineret og tilbage igen.

Pulshukommelsen resettes ved at trykke på pulshukommelsens indikatorlampe, idet denne er en kombineret indikator og resetkontakt.

Stelforbindelse:

Hvis der skal indikeres pulser med en bredde ned til 10 n sek, skal proben have en direkte stelforbindelse der tilsluttes mellem probens stelttilslutning og et stel-punkt der ligger så tæt på målepunktet som muligt. For eksempel stelbenet på den IC der måles på. Uden stelforbindelse (kun forsyningstilslutning) er den mindste pulsbredde der kan måles typisk 20 n sek.

Typiske anvendelsesområder:

Kontrol af Vcc og Vss

Kontrol af clocksignal, "er der liv?"

Kontrol af vigtige CPU-input READY, HOLD og TRAP

Kontrol af "liv" i chip-select logikken.

Kontrol af den logiske funktion af gates.

osv.

----- Logisk pulser, HP 546A -----

Anvendelse:

Ved hjælp af den logiske pulser kan man stimulere et logisk kredsløb uden at foretage nogen indgreb i det. Dette kan bruges til at afprøve gates medens de sidder i kredsløbet, resette tællere, flip-flops eller CPU'er, generere interrupts og undersøge ledninger for kortslutninger.

Tilslutning:

Pulseren skal tilsluttes en forsyningsspænding på 5Vdc ved TTL logik, og 3-18Vdc ved CMOS logik. Som ved proben, er det lettest at forsyne pulseren fra det kredsløb der måles på, eller alternativt en extern spændingsforsyning. Hvis den sidste løsning anvendes skal man huske at forbinde pulserens stelforbindelse til det undersøgte kredsløb.

Output:

Når pulseren er neutral har den en udgangs-impedans der er $>1\text{megohm}$, hvorfor den ikke vil påvirke det kredsløb den sættes i forbindelse med. I den aktive tilstand falder impedansen til $<2\text{ ohm}$ hvorved pulseren overtager kontrollen med det logiske niveau i det punkt der pulses, også selvom dette er en udgang.

Pulsforløbet starter med at probetippen går til logisk 0 i typisk 10 u sek, derefter skiftes til logisk 1 i andre 10 u sek hvorefter outputtet igen bliver højimpedant. Den maksimale udgangsstrøm er begrænset således at der sker en afkortning i udgangspulsen ved stigende strøm. Den tilførte energi bliver således holdt på et sikkert niveau.

Pulseren kan enten triggere manuelt til at give en puls, eller programmeres til automatisk at afgive forskellige pulstog, ialt 5 forskellige. Pulsafgivelsen indikeres på den indbyggede indikatorlampe, hvilket er nødvendigt i de automatiske modes når pulseren anvendes sammen med logik proben.

Typiske anvendelser:

Som enkeltstående instrument kan pulseren anvendes hvorsomhelst man ønsker at pulse det logiske niveau i et punkt, idet det jævner det ovenstående kortvarigt vil antage det modsatte logiske niveau af hvad der var inden pulseren blev trigget.

Eksempler:

- Sette/resette tællere og registre
- Sette/resette flip-flops
- resette CPU'er
- Generere interrupts

Måling med pulser og logik probe:

Ved hjælp af pulser og probe kan man afprøve funktionen af gates, tællere og registre medens disse sidder i det undersøgte kredsløb.

Afprøvning af gates:

Gatens input pulser og dens output kontrolleres med logikproben. Hvis der er reaktion er gaten øjensynlig i orden. Hvis der ingen reaktion er, flyttes logikproben om på gatens indgang sammen med pulseren, og det kontrolleres at indgangen kan pulseres.

Kan indgangen ikke pulseres må der ligge en kortslutning til Vss eller Vcc. Denne fejl kan enten ligge i gatens indgang eller må søges i det kredsløb der styrer indgangen.

Kan indgangen pulseres flyttes pulser og probe om til gatens udgang og det kontrolleres om udgangen kan pulseres.

Kan udgangen pulseres må gaten være defekt. I modsat fald er der en kortslutning til Vss eller Vcc. Denne fejl må søges i gatens udgang eller i det kredsløb den styrer. Vær i denne forbindelse opmærksom på loddebroer i printet.

Afprøvning af tællere og skifteregistre:

Først bringes kredsløbet i en sådan tilstand at det kan styres med pulseren ved f. eks. at stoppe en eventuel clockoscillator. Derefter kan tælleren/registret resettes og clockes frem med pulseren, og resultatet kontrolleres med logikproben.

----- Current tracer HP 547A -----

Anvendelser:

Ved hjælp af current traceren kan man følge en pulserende strøm i et printspor. Det er således muligt at konstatere om en gate, hvis udgangsniveau ikke ændrer sig, faktisk sender strøm ud i en kortslutning eller den er defekt, og hvis den genererer en strøm, hvor denne løber hen!

Specielt til fejlfinding på bussystemer, hvor mange udgange og indgange har forbindelse til den samme leder er current traceren effektiv, idet den som enkeltstående instrument eller sammen med pulseren kan spore lavimpedante forbrugere på en ledning og dermed afsløre en defekt gate. Uden current tracer vil en sådan fejlsøgning ofte nødvendiggøre udlodning af kredse og afbrydelse af printbaner inden fejlen er lokaliseret.

Måleprincip:

Måleprincippet beror på den ændring der opstår i magnetfeltet omkring en leder hvis strømmen i denne ændres. Denne ændring vil kunne inducere en spænding i en spole, som efter en passende forstærkning kan bringe en indikator til at lyse.

Current traceren består af en pickupspole viklet på en lille ferritkerne, efterfulgt af en forstærker og pulsformer der driver en indikatorlampe. Forstærkeren er forsynet med en kontrol til regulering af følsomheden således at denne kan tilpases den strøm der skal traces.

Tilslutning:

Current traceren skal forsynes fra en spænding mellem 4.5V og 18V dc, som tages fra det udstyr der måles på eller fra en separat spændingsforsyning.

Følsomhed:

Current tracerens følsomhed afhænger af stigetiden på den strømpuls der følges. Med en stigetid på 200 n sek er en strømandring på 1 mA tilstrækkelig, medens en stigetid på 200 u sek kræver en strømandring på 1 A. Da stigetiderne i en mikrocomputer normalt ligger væsentligt under 200 n sek er følsomheden tilstrækkelig til at vise normalt forekommende strømme i et kredsløb med MOS og CMOS kredse.

Ved brugen af current traceren er det vigtigt at følsomheden indstilles til netop sikker indikation af den strøm der skal følges og ikke mere, idet overhøringen fra nærliggende printbaner ellers vil vanskeliggøre sporingen. Specielt nærliggende strømforsynings- og stilledere kan give problemer.

Måleteknik:

Current traceren kan anvendes til enten at følge strømmene i et arbejdende kredsløb, eller en strøm der er induceret i et spændingsløst kredsløb ved hjælp af pulseren.

Ved måling på et aktivt kredsløb skal current traceren kalibreres direkte på den udgang der driver den mistænkte printbane. Ved denne kalibrering kan det konstateres om der overhovedet sendes noget strøm ud af den pågældende udgang.

Lad os som eksempel tage en adressebusbuffer, hvor det med logikproben er konstateret at der er aktivitet på indgangen, men udgangen ligger konstant på low. Er bufferen defekt, eller er der en kortslutning til stel et sted på den pågældende adresseledning?

Først kalibreres current traceren på bufferens udgang, hvorved det konstateres at denne sender strøm ud i adresseledningen; bufferen er altså i orden. Derefter følges printsporet bort fra bufferen, idet det påses at markeringerne på probespidsen er rettet ind efter printbanen, således at proben drejes hvis printbanen skifter retning. Så længe indikatoren lyser med samme styrke er man på sporet af strømmen. Deler banen sig og indikatoren slukker er det fordi strømmen løb af den anden gren som man derefter følger. På denne måde vil man til slut være fremme ved den skyldige som enten viser sig at være en IC eller en kortskutning i printet.

Hvis ikke kredsløbet selv leverer den fornødne strøm til målingen kan denne induceres ved hjælp af pulseren. Pulseren tilsluttes i "generator enden" af det printspor der skal følges, og når den er programmeret til et velegnet puls-output kalibreres current traceren HELT oppe på pulserens output pin, hvorefter strømmen følges. Hvis først det er besluttet at anvende pulseren sammen med current traceren, vil det ofte være en fordel at slukke for det kredsløb der arbejdes på, og tilslutte pulseren direkte mellem de printspor der er under mistanke. Derved generes man mindst muligt af overhøring fra nærliggende ledere. Man skal dog være opmærksom på at nogen fejl i gateudgange først viser sig når der er forsyningsspænding tilsluttet.

Hvis en afkoblingskondensator i forsyningsspændingen er kortsluttet kan den lokaliseres med pulseren og current traceren. Der pulser ind mellem stel og den kortsluttede forsyningsledning hvorefter strømmen følges. Hvis der ikke er tale om en total kortslutning kan det være nødvendigt at udlodde eventuelle store elektrolytkondensatorer idet de vil opføre sig som kortslutninger og dermed sløre fejlstedet.

Litteraturhenvisning:

HP, Techniques of digital troubleshooting. Note 163-1
HP, The IC troubleshooters. Note 163-2

Logisk prøver

Ved meget korte pulser skal man bruge stor på
proben

logisk Pulser



puls ca. 10 μ s ved stor belastning
for kortes puls tiden

Milli ohm meter

ca 2000 Ω

----- Testprogrammer -----

Indledning:

Et testprogram er et program der tester og afprøver dele af computersystemet eller hele systemet, hvis det er muligt. Alt afhængig af computersystemets anvendelse kan programmet være udformet på forskellig måde. Nogle programmer starter hver gang computeren tændes, dette anvendes ofte i måleinstrumenter. Andre testprogrammer starter når computeren ikke har andet at lave, dette princip anvendes i computere der aldrig slukkes. De to foregående programtyper er indeholdt som en del af den software der er indeholdt i computeren. En tredje type program startes ved et startsignal, det kan være i form af en interrupt eller en speciel kode på en port. Programmerne kan også anvendes i forbindelse med en emulator dette instrument anvendes dog ofte først når computeren ikke længere fungerer normalt eller computeren aldrig før har hvirket (sluttet i produktion).

Programmerne kan teste RAM, ROM, I/O enheder samt CPU, dog således at fatale fejl kan stoppe computeren så programmet slet ikke kan eksekveres, dette gælder dog ikke i forbindelse med emulator. Outputet fra testen kan være meget varieret, nogle programmer indikerer OK v.h.a nogle LED, andre skriver en fejlliste ud, hvis ikke lige netop den del er defekt.

CPU-test:

Selvdetektering af fejl er effektiv i mikrodata-mater, fordi CPU'en sammen med et testprogram repræsenterer en intelligens, som det ikke er muligt at opnå med konventionel logik. For at testprogrammet imidlertid kan eksekveres, må CPU-sættet (processor, clock og busbuffer) samt lagermediet for testprogrammet (PROM) nødvendigvis fungere. Fejl i disse komponenter kan være mangeartede, og visse typer af fejl giver ikke umiddelbart anledning til fejl i programeksekveringen, så som fejl i ALU'en eller de interne registre.

Betragter man systemet fra serviceteknikerens side, er CPU-fejl ret frusterende, idet teknikkeren efter at have initieret testproceduren ser, at datamaten absolut intet foretager sig. Derfor er det vigtigt med en indikation af, om datamaten overhovedet har eksekveret instruktioner fra testprogrammet. De første instruktioner i testprogrammet bør sætte lamper eller LED's på passende vis.

Hvis disse signaler ikke er korrekte, testes de vitale dele med andre midler, f.eks. logikanalysator, signaturanalyse, udskiftning af kredse og/eller print-

kort.

Testprogrammet bør derefter teste hele CPU'en. De interne registre testes grundigt, bl.a. for mønstersølsomhed. Der undersøges, om indirekte adresseringer ved hjælp af registrene er muligt, hvorefter ALU'en testes ved, at man lader CPU'en foretage en kompleks beregning på nogle udvalgte data. Dette kan udmærket ske ved kald til underprogrammer i den "normale" software, f.eks multiplikations- og divisionsrutiner. Her skal det lige bemærkes, at RAM-lageret bør testes, før man eksekverer programmer, der gør brug af det.

RAM-test:

I praksis er det ikke muligt at opstille en algoritme, hvormed der inden for acceptabel tid kan testes et RAM-lager fuldstændigt igennem. Dette ville kræve, at hele lageret blev testet igennem for samtlige mulige kombinationer af information. Et "lille" RAM-lager på f.eks. 1 k-byte har 2^{1024} eller 10^{308} mulige kombinationer, og for hver kombination skal der foretages en komplet analyse af hele lageret, som vil vare adskillige sekunder.

Ved valg af initial RAM-test er især tidsforbruget altovervejende, da brugeren næppe vil acceptere mere end et par sekunders "opvarmningstid" for systemet.

En algoritme med et rimeligt tidsforbrug kan se således ud:

```
/* RAM-området ligger mellem RAMBOT og RAMTOP */
```

```
CHECK: PROCEDURE;  
  PAT = 01H;  
  DO PTR = RAMBOT TO RAMTOP;  
    RAM (PTR) = PAT;  
  END;  
  DO I = 0 TO 6;  
    PAT = PAT * 2;  
    DO PTR = RAMBOT TO RAMTOP;  
      RAM (PTR) = RAM (PTR) * 2;  
      IF RAM (PTR) ≠ PAT THEN  
        DO;  
          OUTPUT (PORT 0) = LOW (PTR);  
          OUTPUT (PORT 1) = HIGH (PTR);  
          HALT;  
        END;  
      END;  
    END;  
  END;  
END CHECK;
```

Med den viste test fås med rimelig sandsynlighed testet, at:

- 1) hver bit kan huske både 0 og 1.
- 2) adresseringen er unik (eentydig), d.v.s. ingen kortslutninger eller afbrydelser blandt adressebits.
- 3) de enkelte bytes ikke "smitter af" på hverandre, således at information skrevet i een celle påvirker indholdet i en anden.

Der skal bemærkes, at man ved programmeringen bør gå ud fra, at hele RAM-lageret er defekt fra starten, da det jo ikke er afprøvet endnu. Dette indebærer for de fleste CPU-enheder vedkommende, at man ikke kan arbejde med underprogrammer, da kald af disse medfører stack-operationer i RAM-lageret. ligeledes må man have alle variable data (tæller og bitmønstre) i CPU'ens interne registre. Derfor skal programmet skrives i assemblerkode og ikke i højniveausprog. Den viste algoritme vil i assemblerkode til 8085 se således ud:

```
EXTRN BYTE    PORT0
EXTRN BYTE    PORT1
RAMBOT    EQU 2000H
RAMTOP     EQU 2FFFH
```

CSEG

```
CHECK:    MVI B,01H                ;PAT = 01H
          LXI H,RAMBOT            ;DO PTR = RAMBOT TO RAMTOP
C1:        MOV M,B                ; RAM(PTR) = PAT
          INX H
          MOV A,L                ;END
          CPI LOW(RAMTOP+1)
          JNZ C1
          MOV A,H
          CPI HIGH(RAMTOP+1)
          JNZ C1
          MVI C,07H              ;DO I= 7 TO 1
C2:        MOV A,B                ; PAT = PAT * 2
          RLC
          MOV B,A
          LXI H,RAMBOT            ; DO PTR =RAMBOT TO RAMTOP
C3:        MOV A,M                ; RAM(PTR) = RAM(PTR)*2
          RLC
          MOV M,A
          CMP B                  ; IF RAM(PTR) ≠ PAT THEN
          JZ C4
          MOV A,L                ; DO
          OUT PORT0              ; OUTPUT(0) = LOW(PTR)
                                   fortsættes...
```



```

        MOV A,H                ;   OUTPUT(1) = HIGH(PTR)
        OUT PORT1              ;
        HLT                    ;   HALT
                                ;   END
                                ;   END
C4:      INX H
        MOV A,L
        CPI LOW(RAMTOP+1)
        JNZ C3
        MOV A,H
        CPI HIGH(RAMTOP+1)
        JNZ C3
        DCR C                  ;END
        JNZ C2
                                ; RAM OK
```

Initial ROM-test:

De fleste mikrodatamater har større eller mindre dele af programmellene liggende fast i ROM. For systemer med baggrundslager drejer det sig f.eks. om såkaldte BOOTSTRAP-programmer, som kan initiere systemet og hente eventuelt basisprogrammet (operativsystem) op i RAM-lageret fra baggrundslageret. For disse systemer er det af tvivlsom værdi at implementere ROM-test, da testprogrammet jo skal ligge i den selv samme ROM, som bliver testet, og hvis korrekte funktion er en forudsætning for testens gennemførelse.

For systemer med hele eller en del af applikationsprogrammellene i ROM kan det (især hvis der anvendes E-PROM, som kan være mindre pålidelige end maske-ROM) betale sig at indbygge check af koden. Den normale procedure er, at man ved programmeringen reserverer 1 plads i hver ROM-komponent, i hvilken man lagrer en såkaldt checksum, således at summen af alle bytes i den pågældende ROM, modulo 256, giver en bestemt forud valgt værdi. Denne værdi må ikke være 0, da $(0 * 256)_{256} = 0$, og $(OFFH * 256)_{256} = 0$. Altså vil en ROM, som p.g.a. en fejl i f.eks. chip-select-logikken altid svarer med 0 eller OFFH, give et korrekt resultat ved en samlet sum på 0. Testalgoritmen er den, at man for hver ROM adderer alle bytes (og negligerer carry) og tester for den valgte korrekte sum.

Svagheden ved ROM-test er som nævnt, at det program, der skal udføre testen, selv er beliggende i ROM.

Initial test af IN-/OUTPUT:

Medens test af memory kan indbygges udelukkende ved at udvide programmet, er det samme sjældent tilfældet med perifere kredse. Da disse kredse er datamaterns forbindelse til omverdenen, er en initialtest af dem af langt større vigtighed, dels fordi en defekt port direkte vil føre til en fejlfunktion, dels fordi interfacekredse er langt mere udsat for støjimpulser mv., som kan være årsag til defekter, end det indre af datamaten.

Der opstår imidlertid det problem, at en eventuel fejl lige så godt kan ligge i den ekstra "fejl-test-HW", som i den del der bliver testet, hvilket gør denne teknik mest brugbar i store systemer, hvor mængden af testhardware vil være lille i forhold til resten.

For at kunne teste porte, parallelle eller serielle, må man kunne afbryde forbindelsen ud til processen. Hvis der sker konvertering til et andet spændingssystem (f.eks. 48 V_{DC} eller 220 V_{AC}) i påbygget interface, kan denne afbrydelse foretages ved at CPU'en slukker for forsyningsspændingen. Hvis outputportene i systemet ikke har veldefinerede tilstande ved power-on, bør dette realiseres ved, at kun en bestemt bitkombination på en port kan tænde det andet spændingssystem.

Samtidig med at forsyningsspændingen afbrydes, kobles input- og outputporte parvis sammen. Datamaten kan da teste portkredsens funktioner ved at skrive et bitmønster på outputporte og aflæse det på inputporte. Det skal her bemærkes, at et interface ofte inkluderer følsomme komponenter, såsom optokoblere etc., og at disse komponenter bør være inkluderet i testen, således at sammenkoblingen sker så langt væk fra datamaten som

muligt.

Samme fremgangsmåde kan også anvendes ved serielle porte.

Interruptindgange kan i princippet testes på samme måde. Softwaresiden bliver dog en del mere omfattende, da interruptsignaler jo forårsager afbrydelser af programeksekveringen. Testen ligger her i at undersøge funktionen af interruptproceduren.

Løbende HW-check:

Begrundelsen for at lave løbende HW-check under kørslen er først og fremmest den, at man ønsker at standse programeksekveringen, før datamaten på grund af fejlen foretager fatale aktioner på processen. Det vil altså næppe være økonomisk fordelagtigt ved anvendelse i f.eks. underholdningselektronik.

Løbende RAM-test:

Forskellige krav må opfyldes af algoritmen for en løbende RAM-test:

- 1) Ingen lagerplads må miste sit indhold under testen.
- 2) Tidsforbruget skal være minimalt.
- 3) Datamaten skal bringes i en veldefineret venteposition ved fejl.

For at man kan opfylde krav nr. 1 kan kun en enkelt eller ganske få lagerpladser afprøves ad gangen.

Krav nr. 2 nødvendiggør, at kun en lille del af den samlede test udføres ad gangen.

Desuden må der findes en speciel fejlrutine, som nulstiller vitale udgange og henleder opmærksomheden på fejltilstanden.

Ethvert program har rutiner, der gennemløbes periodisk, enten i form af en lang sløjfe i hovedprogrammet eller på basis af timer-interrupt. Et sted i en sådan rutine kan der pladseres en programdel, der tester en RAM-celle for hvert gennemløb. Dette kan gøres hurtigst ved at skrive og læse det inverse af cellernes oprindelige bitmønster.

```
        ;Løbende RAM-test i 8085-assembler
        ...
        ...
        LHLD TSTADR          ;TSTADR = TSTADR +1
        INX H
        MOV A,L              ;IF TSTADR = RAMTOP+1 THEN
        CPI LOW(RAMTOP+1)
        JNZ A1
        MOV A,H
        CPI HIGH(RAMTOP+1)
        JNZ A1
        LXI H,RAMBOT         ; TSTADR = RAMBOT
A1:      SHLD TSTADR
        MOV A,M              ;TEMP = NOT RAM(TSTADR)
        CMA
        MOV M,A              ;RAM(TSTADR) ≠ TEMP THEN
        CMP M                ;DO
        JZ A2
        MOV A,L              ; OUTPUT(0) = LOW(TSTADR)
        OUT PORT0
        MOV A,H              ; OUTPUT(1) = HIGH(TSTADR)
        OUT PORT1
        HLT                  ; HALT
                                ;END
A2:      CMA                  ;RAM(TSTADR) = NOT TEMP
        MOV M,A
        ...
        ...
```

Løbende ROM-test:

Her er der de samme kriterier som ved RAM-test.

En løbende sumberegning efter samme princip som ved initial ROM-test kan fortages i samme rutine som udfører RAM-testen. Af tidshensyn bør ROM-testen dog kun teste een ROM-kreds i hvert gennemløb, så ROM'ene bliver testet på skift.

Løbende INPUT/OUTPUT-test:

Det er her vanskeligere at angive generelle retningslinier, idet strategien og dybden af testen i høj grad afhænger af HW-konfigurationen. For systemer med langsomt reagerende omgivelser (mekaniske relæer, lamper etc.) kan princippet fra initial PORT-test dog anvendes, idet et enkelt port-par testes ad gangen, og det derved opståede fejlsignal kun bliver af få mikrosekunders varighed. Man bør sørge for, at testen

(c) HS-Sønderborg Mikrocomputer fejlfinding Afs: Testprogrammer....

foretages med passende store mellemrum.

Litteraturhenvisning:

- 1) "Test og service på mikrodatamatsystemer",
ECR-97, Elektronikcentralen, August 1980.

Test Program

7K ram $1028 \times 8 = 8192$ celler
 2^{8192} forskellige kombinations muligheder
 sequence $1,098 \times 10^{2466}$

Test 5 Exerciser HP signatur

4000 SS ud på alle pladser der efter Aft

adresse 4000 40 afhængig data

↓
 41
 40
 42
 40
 43

7FFF

7FFE FF
 7FFF 7E

førstetest kontrollere at alle udgange kan

go 1 og 0 AO er testet

anden test giver test på bopi i ram

tester ikke AO vil give forkert signatur

på data ben der vises hvilket adresse ben

der er defekt

Data setup

CPU 100 n sek

LSA 35 n sek

SA 10 n sek

Niveau

CPU TTL

LSA (i TTL) $\frac{1}{A} \times 1,4V$

SA TTL første ben i stop

VAR1 > sammenligner
 VAR2

sidste side april 3 - 2/

----- Microcomputer emulator -----

1. Måleprincip:

En emulator er et måleinstrument, hvormed der kan gribes aktivt ind i programafviklingen i et microcomputersystem. Måleinstrumentet erstatter systemets CPU og kan derved styre og overvåge alle BUSsystemer, eksekvere systemets programmer mens der opsamles måleresultater, samt eksekvere testprogrammer der afprøver hele eller dele af systemet.

Under programafprøvningen kan der sættes nogle break pointes, hvor programafviklingen stoppes så der er muligt at undersøge om CPU registrene har det forventede indhold, samt at der er det forventede indhold i RAM lageret. For at have glæde af denne facilitet kræves en god programdokumentation.

Krav til SUT: *System Under Test*

I systemet der skal testes, her kaldet SUT (System Under Test), skal CPU'en sidde i en fatning, så den kan tages ud og erstattes med stik fra emulatoren. CPU'en skal udover at være i fatning også være pladseret så det er muligt at føre et multilederkabel fra emulatoren til fatningen. Dette er nogle praktiske ting systemdesigneren skal planlægge, hvis det skal være muligt at emulere på systemet.

----- Millennium Microsystem Analyzer (uSA) -----

2. Introduktion:

uSA er et selvstændigt testsystem til test af mikrocomputer baserede systemer og printkort. uSA kan udføre fejldiagnoser på alle niveauer, system-, printkort- og komponentniveau, fordi uSA anvender to diagnosteteknikker - IN CIRCUIT EMULERING (ICE) og SIGNATURE ANALYSIS (SA). Ved anvendelse af forskellige PLUG-IN PERSONALITY-moduler kan uSA'en omstilles til at teste forskellige mikrocomputersystemer.

Ved at anvende ICE kan uSA udføre funktionel GO/NO GO-test via CPU-fatningen i SUT. Ved at anvende SA kan der findes fejl til komponentniveau. Med samme probe, som anvendes til SA, kan der måles puls-bredde, logisk-niveau, tids-interval, frekvens og tælles pulser.



Tilslutning:

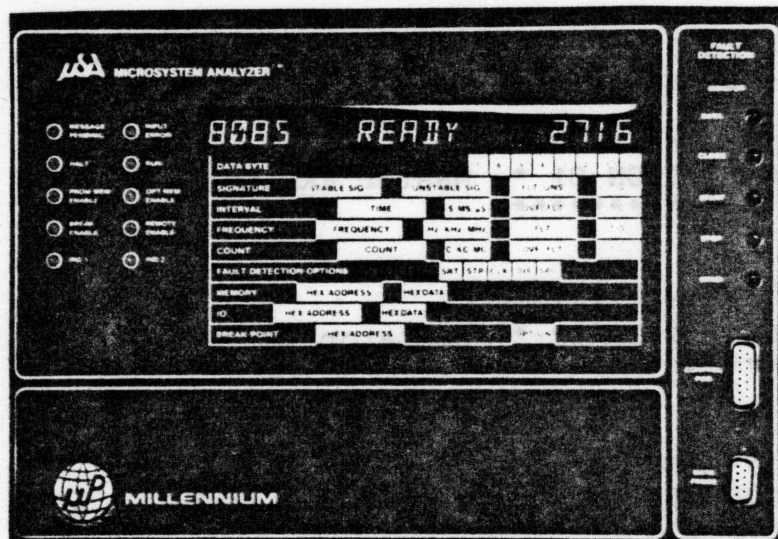
Uden spænding på UUT og uSA anbringes emulatorstikket i UUT med pin 1 vent rigtigt. Husk at kontrollere at det er den rigtige emulator probe der anvendes d.v.s. den skal passe til den processor den skal emulere, og proben skal vende rigtigt 'ben1 til ben1'.

Tænd først for UUT og derefter for uSA, denne rækkefølge skal følges, idet en forkert rækkefølge kan bevirke varig skade på emulatorproben.

Efter gennemløb af en selvtest vil uSA'en skrive på displayet

```
*****
* 8085 READY 2716 *
*****
```

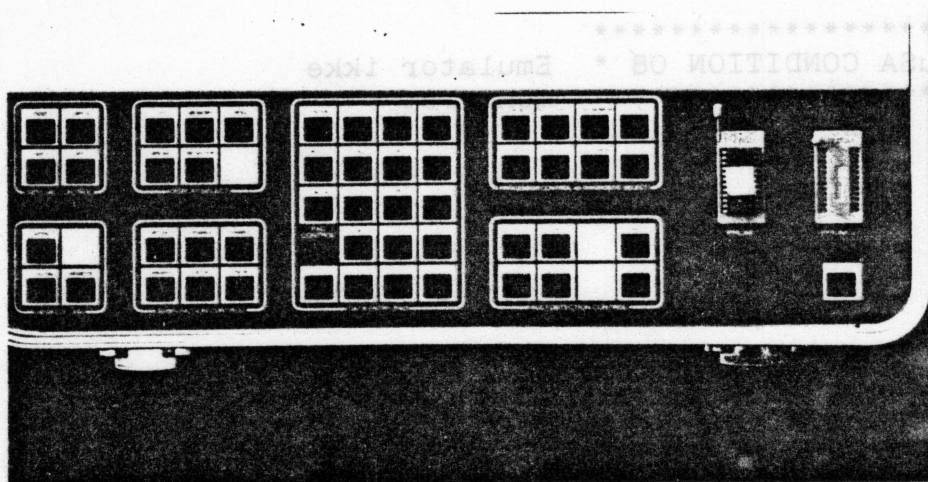
Dersom UUT clocken ikke svinger vil uSA indikere det ved at tænde HALT indikatoren, samtidig med at der gives akustisk signal. Ved OK clock tændes HALT ikke og uSA'en er klar til at emulere UUT'en.



3. Tastaturet

Keyboardet er opdelt i følgende områder:

- (1) RESTART
- (2) PROCESSOR CONTROL-DIAGNOSTIC SELECT
- (3) FAULT DETECTION CONTROL
- (4) DISPLAY SELECT
- (5) DATA ENTRY PAD
- (6) SUBFUNCTION CONTROL
- (7) SPECIAL CONTROL
- (8) REMOTE LINK
- (9) UUT DIAGNOSTIC PROMS



RESTART

Med RESTART bringes uSA tilbage til en power on reset og test af master RAM, emulator ROM, master ROM samt signature ROM udføres. Alle aktive funktioner bliver reset og følgende indformation vil blive vist på displayet for at indikere at uSA'en er klar til at modtage kommandoer:

```
*****  
* 8085 READY 2716 *  
*****
```

Er der en fejl vil een af følgende meddelelser blive vist på displayet:

```
*****  
* uSA CONDITION 01 * Master RAM fejl  
*****
```

```
*****  
* uSA CONDITION 02 * Emulator ROM fejl  
*****
```

```
*****  
* uSA CONDITION 03 * Master ROM fejl  
*****
```

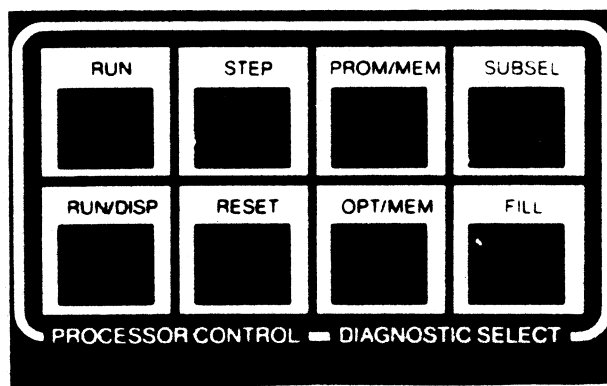
```
*****  
* uSA CONDITION 04 * Emulator ikke aktiv  
*****
```

```
*****  
* uSA CONDITION 05 * Signatur ROM fejl  
*****
```

```
*****  
* uSA CONDITION 06 * Skygge RAM fejl  
*****
```

```
*****  
* uSA CONDITION 08 * Emulator ikke  
monteret
```

PROCESSOR CONTROL



Denne gruppe anvendes til at styre UUT'en, tasterne i denne gruppe er:

RUN
RUN/DISP
STEP
RESET
FILL

RUN

Med denne tast startes eksekveringen på den adresse der er indeholdt i programtæller status register PN. Indholdet af PN registeret kan ændres ved brug af register tasten. Når RUN er aktiv tændes RUN LED indikatoren.

RUN/DISP

Udover RUN funktionen vil RUN/DISP få den valgte memory, register eller I/O udlæst og opdateret ca. ti gange i sekundet. *giver problemer i interrupt styrede systemer PIV*

STEP

Med STEP tasten eksekveres een instruktion af gangen og hvis memory, register eller I/O display er aktiv opdateres det efter hver instruktions eksekvering. Tasten anvendes også til at stoppe RUN eller RUN/DISP funktionen med, hvis de er aktive.

RESET

Denne tast genererer et hardware reset signal til emulator microprocessoren. Der gives et reset signal hver gang tasten påvirkes. RESET tasten påvirker ikke UUT'en. Processor registerne bliver ikke ændret med denne tast, en undtagelse er dog enablede interrupts idet de bliver reset.

FILL

FILL tasten fylder et specificeret RAM adresse område med en bestemt byte. Alle data bliver verificeret mens opfyldningen af lageret finder sted. Alle byte der ikke er korrekt ved verificeringen vil give følgende display:

```
*****
* FILL ERR xxxx=yy *      xxxx= Adresse hvor fejlen
*****                  er fundet.

                               yy= Data læst på adres-
                               sen.
```

Efter tryk på FILL vil uSA'en vise følgende på displayet:

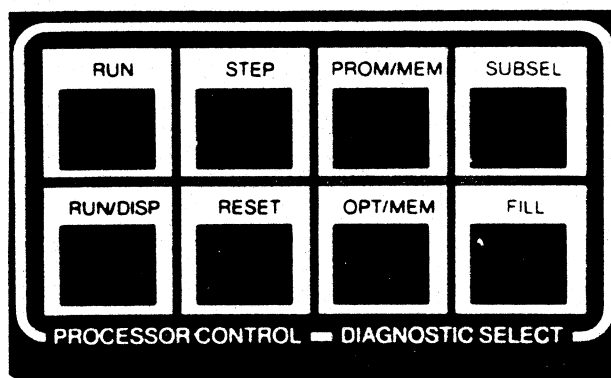
```
*****
* FILL xxxx xxxx xx *
*****

..... Start adresse
. .... Sidste adresse
. .... Data
xxxx xxxx xx
```

og venter på indtastning af startadresse, indtastning afsluttes med ENTER. Derefter venter uSA'en på indtastning af slutadresse. Efter endnu en ENTER venter uSA'en på specificering af de data området skal fyldes op med. Når ENTER igen aktiveres starter den egentlige FILL operation. Udføres det uden læsefejl skrives følgende på displayet:

```
*****
* FILL COMPLETE *
*****
```

DIAGNOSTIC SELECT



Diagnostic select tasterne anvendes til at lægge PROM'en ind i UUT'ens memorymap, samt til at vælge test fra denne PROM. Følgende taster anvendes til dette:

PROM/MEM
SUBSEL
OPT/MEM

PROM/MEM

Med PROM/MEM tasten overføres PROM'en på uSA'ens låg til UUT'ens memory område. PROM MEM ENABLE led'en tændes, uSA'en initialiseres til den første subtest og der skrives på displayet en initialiserings tekst.

```
*****
* INITIAL TEKST      *
*****
```

Er teksten længere end en linie skrives første linie på displayet og med INCR tasten kan resten af linierne hentes frem på displayet, MESSAGE PENDING led'en er tændt så længe der er tekstlinier i samme meddelelse der ikke er vist på displayet. Efter PROM/MEM tasten er aktiveret kan prom'en adresse i UUT'ens område findes ved at se i register PL's indhold.

SUBSEL

Med SUBSEL tasten vælges een ud af op til 99 mulige subtest's. Ved valg af Subtest= 00 vil alle testene blive eksekveret efter hinanden med et opkald. Antal af subtest's i PROM'en sætter grænsen for hvilke subtest's der kan vælges. Vælges der et SUBTEST nummer, der ikke eksisterer, vil det blive opfattet som en

input-fejl og INPUT ERROR led'en vil tændes.

Ved valg af en bestemt test trykkes på SUBSEL tasten og displayet vil vise:

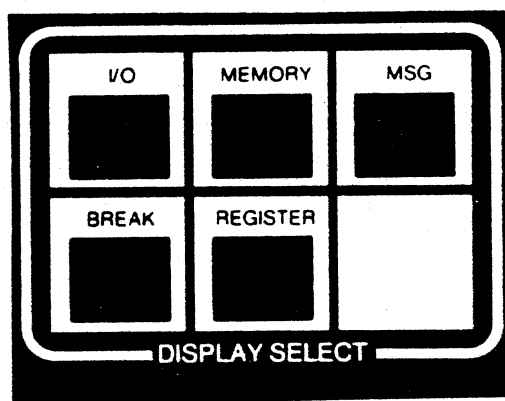
```
*****
* SUBSELECT = dd      * dd= Subtest nummer i
*****                * decimal
```

Med decimal tasterne og ENTER vælges en subtest.

OPT/MEM

Er kun hvirksom når communicationsudstyret er installeret.

DISPLAY SELECT



Display select tasterne er følgende:

I/O
BREAK
MEMORY
REGISTER
MSG

I/O

Denne tast tillader operatøren at læse fra eller skrive til en I/O enhed. Når I/O tasten nedtrykkes, beder uSA'en om en I/O port nummer ved at udlæse:

```
*****
* I/O xx          * xx= I/O port nummer
*****
```

og anmode om indtastning af port nummer med hex tasterne og ENTER tasten. Efter nedtrykning af ENTER tasten udskrives:

```
*****
* I/O xx = yy      *   xx= I/O port nummer
*****             *   yy= data læst på port xx
```

Ved nedtrykning af BINARY tasten bliver data'ene udlæst binært. Ønskes nye data indlæst på porten, indtastes data hvorefter ENTER bruges til at indlæse med.

BREAK

Med BREAK tasten vælges hardware breakpoint's adresser, samt om der skal stoppes ved

```
READ ACCESS
WRITE ACCESS
READ OR WRITE ACCESS
PASS COUNT
```

Ved PASS COUNT forstås at bestemt antal gange adressen skal passeres før der stoppes. Der kan ligeledes vælges mellem et stop eller et spring til en anden adresse, når break'en bliver aktiv. Ved tryk på BREAK vises på displayet den tidligere valgte adresse og option.

```
*****
* BRKA=xxxx OPT=y  *   xxxx= Break adresse
*****             *   y= option i hex format
```

Derefter er det muligt at bruge følgende taster:

```
INCR
HEXADECIMALE TASTER
BINARY TASTER
ENABLE
DISABLE
```

BINARY tasten bruges til at få option'en til at blive udlæst binært. For at break'en overhovedet kan opstå skal der mindst vælges READ- eller WRITE access. Bit'ene i option'en har følgende betydning:

BIT	FUNKTION
3	1= break ved write på den specificerede adresse
2	1= break ved read på den specificerede adresse
1	1= pass count enablet
0	0= stop ved break, 1= spring ved break

Efter editering af de to felter (break adresse og option) vil eksistensen af de efterfølgende display afhænge af den valgte option. Er pass count valgt kan antallet af passeringer angives ved at trykke INCR

```
*****
* PASS COUNT = dddd *      dddd= antallet af passager
*****                    af den valgte adres-
                           se. Området er fra 1
                           til 9999.
```

Hvis spring option er valgt, fremkommer adresse valget ved at trykke på INCR

```
*****
* JUMP ADDR = xxxx *      xxxx= Den adresse hvor uSA
*****                    vil fortsætte efter
                           break.
```

Efter break adresse og option er specificeret skal ENABLE tasten trykkes før en anden hovedfunktion aktiveres, efter det er BREAK ENABLE led'en tændt for at indikere at et breakpoint er aktiv.

Når UUT'ens program eksekveres og et breakpoint findes, vises det på displayet:

```
*****
* BREAK POINT =xxxx *      xxxx= Adressen på den in-
*****                    struktion der for
                           årsagede break'en.
```

Et breakpoint slettes ved at trykke BREAK efterfulgt af DISABLE tasten.

MEMery

Denne tast anvendes til at vise eller modificere indholdet af en memory adresse, enten hexadecimalt eller binært. UUT'ens memory indhold bliver altid vist eller modificeret undtagen hvis adressen falder inden for PROM'ens adresse og PROM/MEM er aktiv, er det tilfældet bliver PROM'en eller skygge RAM'en vist. Når MEM tasten aktiveres stoppes emuleringen af UUT'en og auto-editering aktiveres, på displayet vises:

```
*****
* MEM xxxx *
*****
```

og adressen kan indtastes med hex tasterne og ENTER.

Efter ENTER bliver adressens indhold udlæst:

```
*****
* MEM xxxx = yy      *   xxxx = Adresse
*****               yy   = Indholdet af adres-
                        sen.
```

Med INCR og DECR kan adressen tælles een frem eller een tilbage. Med HEX eller BINARY tasterne samt ENTER indlæses nye data på den viste adresse. Hvis operatøren forsøger at modificere ROM eller en defekt RAM celle indikerer uSA'en fejl.

```
*****
* MEM xxxx = yy ERR *   yy = data der er læst fra
*****               adressen der blev for-
                        søgt modificeret.
```

REGister

Med denne tast kan operatøren få vist og modificeret indholdet af registerne i UUT'ens CPU. Med hex tasterne og ENTER ændres indholdet i det viste register eller registerpar, med INCR blades videre til næste sæt registre.

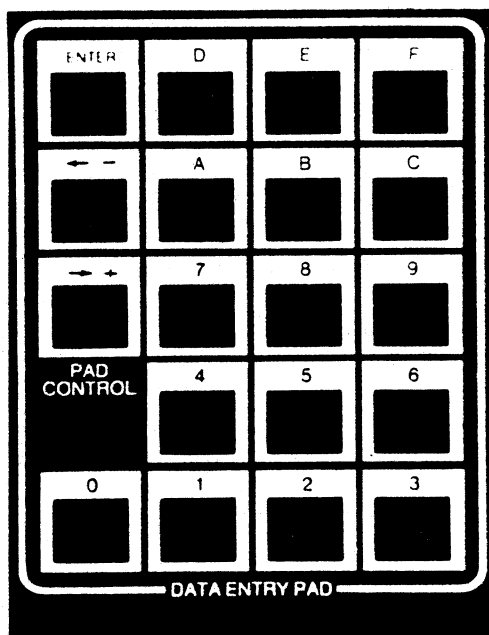
MSG

Med denne tast hentes en meddelelse frem på displayet, hvis der er en meddelelse der venter på at blive vist, dette indikeres ved at MESSAGE PENDING led'en lyser. Er meddelelsen længere end en linie blades efterfølgende linier frem med INCR. Trykkes der på MSG uden at MESSAGE PENDING led'en lyser, så vil

```
*****
* NO MESSAGE      *
*****
```

blive vist.

DATA ENTRY

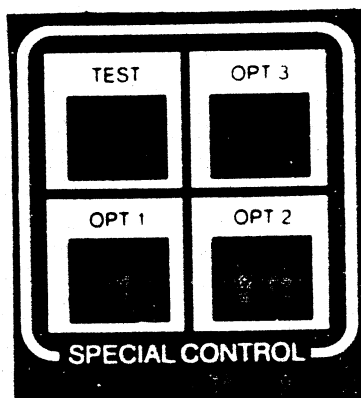


Tasterne i denne gruppe anvendes til at indtaste adresser og data samt forskellige set-up i uSA'en. Tasterne er:

HEX TASTERNE
 PIL TIL HØJRE
 PIL TIL VENSTRE
 ENTER

Med pilene kan der flyttes inden for samme datafelt, hex tasterne anvendes til at indtaste data i datafeltet med, med ENTER afsluttes indtastningen i et datafelt.

SPECIAL CONTROL



Tasterne i denne gruppe er

OPT1
OPT2
OPT3
TEST

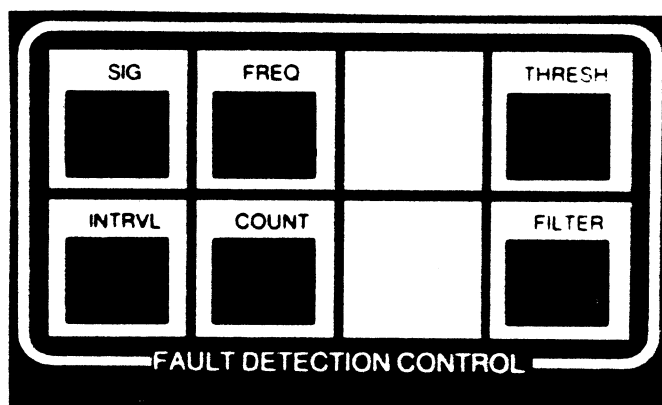
OPT1, OPT2, OPT3

Disse tre taster anvendes kun når uSA'en er udbygget med communicationsudstyr.

TEST

Denne tast starten en SELF-TEST af uSA'en. En hurtig test af RAM, ROM, forpladeindikatorer samt nogle interne funktioner. Testen afsluttes ligesom beskrevet under RESTART.

FAULT DETECTION



Tasterne i denne gruppe anvendes til at starte de specielle målinger med uSA'en.

COUNT
FILTER
FREQ
INTRVL
SIG
THRESH

COUNT

Denne funktion sætter uSA'en i stand til at måle pulser eller transienter. Med OPSEL bestemmes om der skal måles pulser eller transienter samt hvad der skal tælles med (internt signal eller extern dataprobe), der kan ligeledes vælges hvilken adresse der skal eksekveres før målingen påbegyndes, ligeledes kan der opgives en adresse hvor målingen skal stoppe.

FILTER

Med denne tast kan der indsættes et støjfilter i serie med dataproben. Følgende værdier er til rådighed:

VÆRDI	FILTER
0	ingen filter
1	50 nsecs filter
2	100 nsecs filter
3	150 nsecs filter

FREQ

Med denne tast startes frekvensmåling, med OPSEL vælges under hvilke betingelser målingerne foretages.

INTRVL

Interval eller pulsbredde måling. OPSEL vælger under hvilke betingelser målingen foretages.

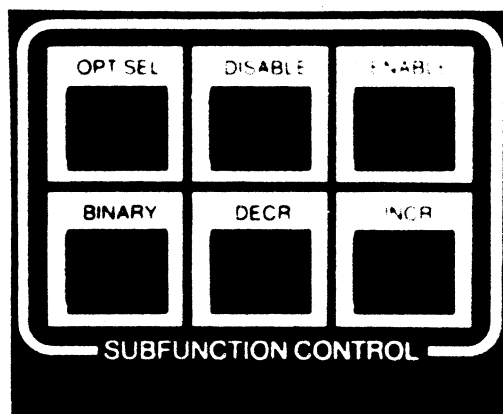
SIG

Signaturmåling med dataproben, start og stopbetingelser vælges med OPSEL.

THRESH

Anvendes ikke endnu.

SUBFUNKTION CONTROL



I denne gruppe er der kun taster der får betydning i forbindelse med andre funktions taster.

BINARY
DECR
DISABLE
ENABLE
INCR
OPTSEL

BINARY

Data byten i det viste display vises som 8 bit binær værdi.

DECR

Memory adresse eller I/O portnummer tælles een ned.

DISABLE

Anvendes kun i forbindelse med BREAK til at sætte aktivt breakpoint ud af funktion.

ENABLE

Anvendes kun sammen med BREAK til at gøre et breakpoint aktivt.

INCR

Tæller memory adresse eller I/O portnummer een frem, blader ny linie frem på displayet ved flereliniet meddelelse, blader frem til næste sæt registre i register display mode.

OPTSEL

Anvendes til at ændre på målebetingelserne for COUNT, FREQ, INTRVL og SIG. Efter valg af en måletype (COUNT, FREQ) trykkes på OPTSEL hvorefter uSA'en viser på displayet det nuværende set-up for målingen.

* fff OPTS = dxxxb *

...	Special kriterie
.	Display kriterie
.	Clock/data kriterie
.	Stop kriterie
.	Start kriterie
.....	Enten SIG, FREQ, COUNT eller INTVL

Operatørem kan nu indtaste værdier der har bestemt betydning i hver af positionerne. Tallenes betydning findes i manualen på siderne 36..41.

USA Micro System Analyzer

In Circuit Emulation (ICE)

Signature Analyzer (SA)

Frequens meter

Interval meter

Signaturer efter HP metode men man ikke sætte EPV i jre rum display er i Hexadecimal som skal converteres til HP code vis man skal sammenligne de to signaturer

Kører programmet i fuld hastighed
se og ændre alle adresser og registre
og t/10

Tænd først for PIV systemet sluk
først for Millennium der efter for PIV
Side 10-15

----- Komponentfejl -----

Hvis (når) en mikrocomputer svigter, vil det i de fleste tilfælde kunne føres tilbage til en fejlende komponent. Det er derfor af betydning for en effektiv fejlfinding at teknikeren har kendskab til typiske komponentfejl og deres konsekvens for computerens funktion.

CPU, normal:

Hvis der opstår fejl på CPU'en vil det normalt være katastrofalt for mikrocomputerens funktion. CPU-fejl kan groft opdeles i adressefejl, datafejl, control-signalfejl, interruptfejl, timingfejl og marginalfejl. De tre første fejltyper vil næsten med sikkerhed føre til øjeblikkeligt stop af programeksekveringen, medens de sidste fejltyper ofte optræder mere sporadisk og derfor tilsvarende vanskelige at finde.

Adressefejl:

Adressefejl kan opstå ved fejl i CPU'ens interne registre som f. eks. programcounteren, eller afbrydelser og kortslutninger fra disse og ud til CPU'ens ben. Det skal erindres at adressefejl også kan have "udvendige" årsager i form af defekte komponenter i forbindelse med adressebussen. Virkningen af en adressefejl er afhængig af om fejlen ligger i de lave eller de høje bit, og om den optræder konstant (kortslutning/afbrydelse), eller kun i forbindelse med bestemte maskincycles eller instruktionstyper. Den almindeligste følge vil dog være at CPU'en kommer ud af "trit" med programmet og begynder at bruge data som instruktioner og omvendt, -den er "gået i skoven". Symtomerne ved adressefejl og andre CPU-fejl vil ofte være en total mangel på liv, og selv ikke testprogrammer i udstyret vil kunne køres.

Kontrol af adressebus med oscilloscop:

Ved at sætte CPU'en i free-run kan man med et oscilloscop kontrollere om programcounteren kan tælle hele adresseområdet igennem, og om den har forbindelse til CPU'ens adresseben. Ved at starte på det mindst betydende adressebit og gå opad, skal frekvensen halvveres for hver bit man kommer op. Det vil på denne måde være let at konstatere afbrydelser i, og kortslutninger mellem adressebit. CPU'er der multiplexer adresse og data ud på en fælles bus giver et specielt problem, idet adressen skal iagttages synkront med en adressestrobe (ALE), eller man kan måle efter en eventuel adresselatch og gå tilbage mod CPU'en hvis der konstateres fejl på enkelte adressebit.

Medens CPU'en er i free-run vil det være naturligt at kontrollere chip-select logikkens funktion. Dette gøres ved at trigge oscilloscopet på det mest betydende adressebit og vælge triggerpolaritet og sweep-tid således at et helt adressegennemløb netop fylder skærmen. Derefter måles på de forskellige kredses chip-selectindgange. Det kan nu konstateres om de forskellige kredse i det hele taget bliver selectet, samt hvor omtrent i adresseområdet de ligger og om hukommelses-kredsene ligger "pænt" efter hinanden.

Adressekontrol med logikanalysator:

Har man mulighed for at iagttage hvad der sker i computeren ved hjælp af en Logik State Analysator, kan man prøve at finde det sted i programmet (det når sjældent langt ved adressefejl) hvor det bryder sammen. Ved at sammenligne med det fejlende udstyrs programdokumentation samt omhyggelig analyse af de opsamlede data er det ofte muligt at lokalisere fejlen. Når jagten gælder "hængende" eller kortsluttede bit i en bus, er det en fordel at formatere analysatorens liste binært idet fejl af ovennævnte type da vil træde tydeligere frem. Findes der ingen sofrwaredokumentation til det fejlramte udstyr, må man holde sig til de oplysninger der kan hentes i CPU'ens datablad. Det er f. eks. restart- og interruptvektorer.

CPU-registerfejl:

Adressefejl kan også opstå som følge af fejl i interne CPU'registre hvis disse indeholder en adresse. Som et eksempel kan nævnes at forfatterene ved en lejlighed har iagttaget en stack-pointer tælle forkert i en Z80. Sådanne fejl lokaliseres med logikanalysator og omhyggelig analyse, eller ved kørsel af testprogrammer der specielt tester CPU'registre for korrekt funktion.

Datafejl:

Fejl i CPU'ens dataport vil medføre at programeksekveringen meget hurtigt bryder sammen som følge af at programmet læses forkert. Problemet er at afprøve CPU'en, idet den i de fleste tilfælde vil reagere helt uforudsigeligt. Datafejl kan også forekomme i forbindelse med at CPU'en afleverer data til hukommelse eller porte. Vær opmærksom på at datafejl kan forårsages af andre kredse med forbindelse til databussen.

Lokaliseringen af fejl i CPU'ens dataport kan ske ved at følge programforløbet med en logikanalysator, idet man kontrollerer at CPU'en eksekverer de enkelte instruktioner korrekt. Det er en betingelse at andre fejlmuligheder kan udelukkes som f. eks. timingfejl (en programhukommelse med for lang accesstid) eller niveaufejl. Man skal erindre at en logikanalysator kun viser

det DEN har opfattet hvilket ikke nødvendigvis er det samme som CPU'en.

Controlssignalfejl:

CPU'ens kontrolsignaler omfatter READ, WRITE og IN/OUT signalerne. Der er nogen variation mellem de forskellige firmaers CPU'er med hensyn til hvor mange af de nævnte signaler der findes, men i sidste ende varetager de de samme opgaver. Hvis kontrolsignalerne mangler, eller er timet forkert, vil det gå ud over CPU'ens kommunikation med hukommelsen og portene. Til kontrol kan man bruge oscilloscop og logikanalysator.

Interruptfejl:

Fejl i CPU'ens interruptsystem kan udarte sig på flere måder. For eksempel reagerer den slet ikke på en indkommende interrupt på trods af at interruptsystemet er enableret hvilket sidste skal være nøje kontrolleret, idet nogen processorer ikke automatisk enabler deres interruptsystem ved returnering fra en interruptrutine. I forbindelse med interrupt, kan det også forekomme at CPU'en hopper til en forkert vektor, og derved kommer ud af trit med programmet. Hvis stackpointeren tæller forkert vil det ikke hindre at en interruptrutine eksekveres korrekt, men det vil gå galt når kontrollen skal returneres til programmet hvori interrupten opstod.

Timingfejl:

Timingfejl vil sige at data og adresser på busserne er ude af trit med kontrolsignalerne. Dette vil (måske) gå ud over signaloverførslen mellem CPU og hukommelse og porte, idet det mange gange kommer an på et sammenspil mellem fejlsens art og hvor meget bedre end databladenes opgivelser de andre kredse i computeren er. Har man mistanke om at der er tale om en timingfejl, bør man først kontrollere CPU'ens oscillatorfrekvens idet den ikke helt sjældent kan være sprunget til 2 eller 3 gange det korrekte. Ellers må man prøve at finde det sted i programmet hvor det går galt, og derefter med et digitalt trigget oscilloscop eller en timinganalysator foretage kontrolmålinger af tiderne.

Marginalfejl:

Marginalfejl vil sige at CPU'ens NÆSTEN overholder sit datablad, det være med hensyn til tider eller spændinger og strømme. En sådan fejl vil kunne gå upåagtet hen i lange tider, lige til den dag hvor man skifter en anden komponent i computeren, eller arbejdsbetingelserne kommer ud mod ydergrænserne. Det giver næsten sig selv at det kan være frustrerende at finde kilden til en sådan fejl!!

Kontrol af CPU'en:

En CPU kan kontrolleres på flere måder, enten indirekte eller direkte. Hvis CPU'en sidder i fatning er det mest nærliggende at prøve med en ny. Man skal dog være klar over at hvis fejlen var forårsaget af en dårlig forbindelse i fatningen er denne ved samme lejlighed fjernet for en tid!!!

Har man adgang til en Microcomputer Emulator kan man lade denne erstatte CPU'en, og ud over verifikation af CPU'ens funktion udføre målinger på hukommelse og porte ved hjælp af uSA'en.

Ved at følge programeksekveringen med en logik-analysator kontrollerer man om CPU'en er i stand til at hente instruktioner fra programhukommelsen samt udføre disse. Man kontrollerer at den kan udføre sit job og slutter heraf at den må være i orden.

Direkte kontrol af CPU'en kan udføres hvis den flyttes over i en "prøvecompute" hvor det ved hjælp af program og hardware afprøves om den er i stand til at eksekvere alle instruktioner, adresseringsformer, interrupts o.s.v. Noget der ligner en sådan test, findes i form af TEST 00 i HP's Microprocessor Exerciser. Ved denne test afprøves hele instruktionssættet med undtagelse af HALT og RST0. Kontrolmålingen foretages ved at kontrollere signaturene på CPU'ens ben. I skrivende stund, jan. 1983 findes Microprocessor Exerciseren til 6800, 6802, 6808, 8085 og Z-80.

A418 21	A8	V _{ss}	20 0000
P8H6 22	A9	AD7	19 8UP2
F910 23	A10	AD6	18 0805
8U45 24	A11	AD5	17 2U33
75F6 25	A12	AD4	16 74A6
480H 26	A13	AD3	15 6A85
0531 27	A14	AD2	14 UUU5
789C 28	A15	AD1	13 5845
1984 29	S0	AD0	12 3P97
0000 30	ALE	INTA	11 P189
P189 31	WR	INTR	10 FU0U
P189 32	RD	RST5.5	09 U4HH
69PC 33	S1	RST6.5	08 C992
HFHC 34	IO/M	RST 7.5	07 A04H
P189 35	READY	TRAP	06 P312
P189 36	RESET IN	SID	05 1336
P189 37	CLK(OUT)	SOD	04 1336
0000 38	HLDA	RESET OUT	03 0000
0000 39	HOLD	X2	02 ----
P189 40	V _{cc}	X1	01 ----

----- Single chip microcomputer -----

De problemer der kan være omkring fejlfinding på single chip computere knytter sig hovedsagelig til to forhold. For det første er der den meget høje grad af integration, hvor hele computeren består af en eller nogen få kredse med mange komplekse funktioner. For det andet findes der så godt som aldrig nogen softwaredokumentation i servicemanualerne hvilket gør brugen af logikanalysator problematisk.

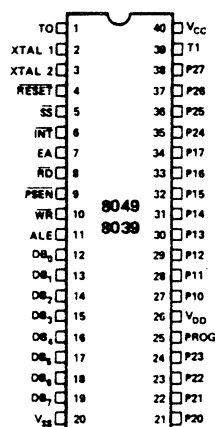
Single chip computerens opbygning:

Single chip computeren adskiller sig fra den normale CPU ved at indeholde alle de elementer der skal til for at opbygge en computer, d.v.s. CPU, programhukommelse, datahukommelse og porte. Som eksempel kan vi se på 8049H og 8039HL fra INTEL.

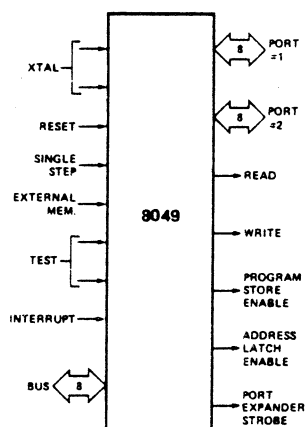
Indbyggede funktioner	8049H	8039HL
8-bit CPU	X	X
Intern Program memory	2K x 8 ROM	ingen
Max program memory	4K	4K
Data memory	128 x 8 RAM	128 x 8 RAM
I/O linier	27	27
8 bit timer/counter	X	X
Reset	X	X
Interrupt	X	X
Clockoscillator	X	X
Power down mode	X	X

Blokdigram og benforbindelser:

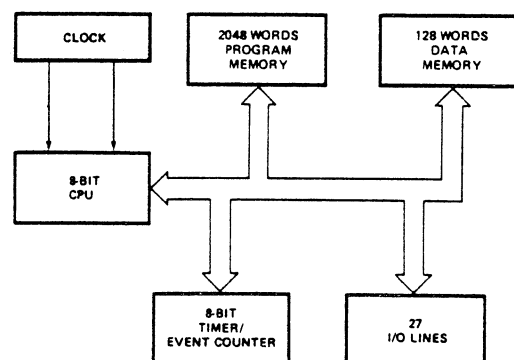
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



Benfunktioner og deres kontrolmuligheder:

Navn	Pin No.	Funktion
Vss	20	Stelben
Vdd	26	Forsyningsspænding til det indbyggede RAM-lager. Low power stand-by.
Vcc	40	Hovedforsyningsspænding, 5V.
PROG	25	Clockpuls der timer data og control-bit i forbindelse med I/O-expander 8243. Hvis computeren er forsynet med en 8243, skal der være aktivitet på PROG.
P10-17 Port 1	27-34	8-bit quasi-bidirectional port.
P20-27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. Hvis der anvendes extern programhukommelse fremkommer de 4 højeste bit af programcounterens 12 bit på P20-P23 under en instruktionsfetch. Hvis I/O expanderen 8243 anvendes kommunikerer den med "CPU"en over P20-23.
DB0-DB7	12-19	Ægte bidirectional port der kan skrives til, og læses fra under kontrol af RD og WR signalerne. Denne port ligner meget AD-bussen på en 8085 CPU. Under en instruktionsfetch fra en extern programhukommelse udsendes programcounterens 8 lave bit via denne port, og under kontrol af ALE'en. Den adreserede instruktion læses under kontrol af PSEN (Program Store ENable) der fungerer som et RD-signal under en extern instruktionsfetch. Ved dataoverførsel til og fra extern RAM eller "almindelige porte" overføres adressen som før under kontrol af ALE'en, medens dataoverførslen sker synkront med RD og WR.

ALE 11 Address Latch Enable. Dette signal forekommer en gang for hver maskincycle med en frekvens der er 1/15 af krystalfrekvensen. ALE'en bruges til at latched de 8 lave bit i adressen ved adressering af extern memory og I/O. Da ALE'en er til stede kontinuerligt kan den bruges til timere og lignende.

Navn	Pin No.	Funktion
PSEN <i>Phlow low 1 opcode fetch</i>	9	Program Store Enable. Kontrolsignal der timer dataoverførslen fra en extern programhukommelse til CPU'en. Det anvendes typisk som chipselect-signal på programhukommelsen.
RD	8	Read strobe. Kontrolsignal der er aktivt under en BUS-read, som for eksempel når der læses data fra et externt RAM-lager eller en port. RD bruges typisk som enable signal.
WR	10	Write strobe. Kontrolsignal der er aktivt under en BUS-write. Anvendes som skrivekontrolsignal på externt RAM-lager og porte.
EA <i>kan bruges til at skifte til testprogram</i>	7	External Access. Niveauet på dette input bestemmer om CPU'en henter instruktioner i den indbyggede programhukommelse, eller fra et externt programlager. For 8039 der ikke har nogen programhukommelse indbygget skal dette ben være lagt til high. Den typiske anvendelse for EA finder sted under programudvikling hvor man kan få eksekveret et program i en extern EPROM inden det overføres til det interne programlager.
RESET	4	Reset input, aktiv low. Dette ben skal holdes lavt mindst 10 m sek. ved power on reset, og mindst 5 maskincycles (9,5 u sek. ved 8 MHz krystalfrekvens) ved en "varm reset".
INT	6	Interrupt input, aktiv low. Dette ben skal holdes lavt i mindst 3 maskincycles for med sikkerhed at blive observeret. Når computeren reagerer på INT gemmes programcounterens indhold i den interne stack, og kontrollen overlades til adresse 3 i programhukommelsen.
SS	5	Single Step. Single step input, der i forbindelse med ALE'en kan bringe processoren til at singlesteppe igennem sit program. Processoren stopper i første maskincycle i en instruktionscyclus hvor adressen på næste instruktion sendes ud.

Navn	Pin No.	Funktion
TO	1	Input ben som kan testes via to betingede hop instruktioner. Kan via en instruktion bringes til at fungere som clockoutput (1/3 krystal-frekvens).
T1	39	Input ben som kan testes via to betingede hop instruktioner. Kan via en instruktion bringes til at fungere som timer/counter input.
XTAL1	2	Crystaloscillator input, og input for extern oscillator.
XTAL2	3	Crystaloscillator, driver output.

For yderligere oplysninger henvises til INTEL "Microcontroller User's Manual".

Fejlfinding omkring en single chip microcomputer:

Ved fejlfinding på en opstilling der er forsynet med en single chip computer må man betragte den som enhver anden kreds der skal have nogen input, og hvis der er i orden skal der være bestemte output, ellers er den defekt!

Signaloversigt for 8049/8039:

Signalerne er nævnt i den rækkefølge forfatteren mener de bør kontrolleres. Hvor et signal er betinget er betingelsen nævnt.

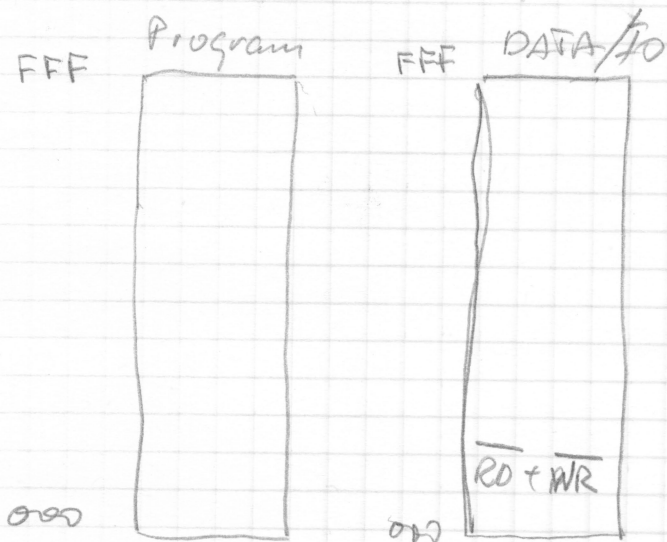
Signal/spænding	Kommentar
Vcc	5 V +/- 10%
Vdd	5 V +/- 10%
Vss	0 V
RESET	Low i mindst 10 m sek. ved power on reset, ellers high.
ALE	Clockoutput, 1/15 af krystal-frekvensen. Hvis den mangler kontrolleres XTAL1 og 2, eller XTAL1 ved extern oscillator.
SS	Skal være high ved normal drift.
EA	Niveauet er afhængig af om der er intern programhukommelse og om denne bruges. Ved 8039 SKAL EA være high.
INT	Normalt high.

Single chip Computere 8-5

ingen program dokumentation

memorymap

Program 0000 - FFF



Alle instruktioner har samme længde
alle altid $1/15$ af clock frekvens

T0 kan programmeres til at være clock output
på $1/3$ af clock frekvensen

T1 kan programmeres til at være timer/counter
input

Bruges til styring af mekanik så som

Video recordere Vaskemaskiner Fotocopi maskiner
Printer

Omskifter til 220 240 skal stå rigtigt
da overstrøms sikring ellers vilke felder ude
og transformer brænder af

Hvis disse signaler og nivauer er i orden, er betingelserne til stede for signaler på følgende ben:

Signal/spænding	Kommentar
PSEN	Hvis der er extern programhukommelse skal dette ben gå low hver gang der hentes en instruktion i denne (en gang for hver byte).
RD	Hvis der er almindelige porte, RAM-lager eller extern hukommelse med data (ikke program, men f. eks. en charactergenerator i en printer), vil RD gå low hver gang der læses i en af disse enheder.
WR	Hvis der findes almindelige porte eller RAM-lager vil WR gå low hver gang der skrives i en af disse enheder.

Hvis der er aktivitet på disse signaler er der god grund til at antage at single chip computeren er i orden, og en fejl skal søges i dens porte eller andet sted i opstillingen.

For yderligere henvises til Elektronikcentralens rapport om Single-chip Microcomputere, ECR-116.

Brug af logikanalysator:

Hvis dele af eller hele programmet ligger i extern ROM, kan man tilslutte en logikanalysator og følge programeksekveringen. Som adresseclock anvendes ALE, og som dataclock PSEN. Med dette setup vil man kun få indspillet instruktionsfetches. Vil man også se executecycles må dataclocken gates sammen af RD, WR og PSEN. Hvis den anvendte analysator er forsynet med disassembler vil denne med fordel kunne anvendes, specielt da der ofte ikke findes nogen softwaredokumentation til det defekte udstyr.

----- Fejl på hukommelser -----

Fejl i en hukommelseskreds vil i de fleste tilfælde medføre at programmet bryder sammen. Hvis fejlen ligger i programhukommelsen vil programmet direkte blive forvansket, medens en fejl i datahukommelsen kan påvirke programafviklingen ved at levere forkerte data til processen. Datahukommelsesfejl i det område hvor CPU'en har sin stack liggende vil dog være katastrofale i forbindelse med afviklingen af subrutiner og interrupt.

Fejl på adresseledninger, generelt:

Fejl på adresseledningerne og den interne adressedecoder vil bevirke at det bliver umuligt at nå bestemte adressers indhold, og andre vil forekomme flere gange. Som eksempel kan tænkes kortslutninger mellem to adresser, og en afbrydelse af en adresseledning. I nedenstående eksempel viser første tabel adressen, anden søjle de korrekte data og tredje og fjerde søjle viser hvad der ville komme ud af data ved henholdsvis en afbrydelse i en adresseledning og en kortslutning mellem to adresseledninger.

-----*				-----*	-----*	-----*	-----*
-- A3	A2	A1	A0	* Data (hex)*	A1 konst.	* A2/A3 korts.	
				* * low *			
-----*				-----*	-----*	-----*	-----*
0	0	0	0	00	00	00	
0	0	0	1	01	01	01	
0	0	1	0	02	00	00	
0	0	1	1	03	01	01	
0	1	0	0	04	04	00	
0	1	0	1	05	05	01	
0	1	1	0	06	04	06	
0	1	1	1	07	05	07	
1	0	0	0	08	08	08	
1	0	0	1	09	09	09	
1	0	1	0	0A	08	08	
1	0	1	1	0B	09	09	
1	1	0	0	0C	0C	08	
1	1	0	1	0D	0D	09	
1	1	1	0	0E	0E	0E	
1	1	1	1	0F	0F	0F	

Ved kortsluttede adresseledninger vil det almindeligvis være således at blot een af dem er low, vil de begge være low.

I forbindelse med test af RAM-lagre hvor det er muligt at indlæse et testmønster inden kontrollæsningen kan førnævnte fejl tydeliggøres. Herunder er vist hvad der vil komme ud af data på de forskellige adresser med henholdsvis A1 afbrudt og A1 og A2 kortsluttet hvis der forinden er indlæst en baggrund af nuller og OFFH på adr. 00H.

-----*				*-----*		*-----*	
-- A3	A2	A1	A0	* Data (hex)*	A1 konst.	* A2/A3 korts.	
				* Se ff	* low	*	
-----*				*-----*		*-----*	
0	0	0	0	FF	FF	FF	
0	0	0	1	00	00	00	
0	0	1	0	00	FF	FF	
0	0	1	1	00	00	00	
0	1	0	0	00	00	FF	
0	1	0	1	00	00	00	
0	1	1	0	00	00	00	
0	1	1	1	00	00	00	
1	0	0	0	00	00	00	
1	0	0	1	00	00	00	
1	0	1	0	00	00	00	
1	0	1	1	00	00	00	
1	1	0	0	00	00	00	
1	1	0	1	00	00	00	
1	1	1	0	00	00	00	
1	1	1	1	00	00	00	

ff: Denne søjle skal forstås således at det er hvad der er forsøgt at lægge ind på de forskellige adresser, men som følge af en eventuel fejl vil der blot ske det samme som når man læser indholdet, nemlig at der skrives til de(n) samme adresse(r) flere gange.

Datafejl:

Ved datafejl kan der skelnes mellem fejl der kun påvirker bit på en enkel eller nogle få adresser og fejl der påvirker et bestemt bit på alle adresser.

Fejl på et bestemt bit på samtlige adresser kan forekomme hvis der opstår fejl i kredsens I/O-kredsløb. En fejl her kan også omfatte alle outputbufferene i kredsens udgang der enten er "døde" eller nægter at gå i 3-state når hukommelsen ikke er chip-selected. Sidstnævnte fejl vil ofte bevirke at computeren stopper idet databussen bliver blokeret af den defekte kreds. En sådan tilstand vil også kunne forårsages af fejl i chip-select-logikken hvis flere kredse bliver selected samtidigt.

Datafejl på RAM-lagre:

Fejl i RAM-lagerets hukommelsesceller kan ytre sig på forskellige måder. De almindeligste er:

- indholdet i en bestemt celle lader sig ikke ændre...
- der er afsmitning mellem celler...
- bestemte mønstre et sted ændrer bit et andet sted...
denne fejl kaldes mønsterfølsomhed og kan også forekomme i CPU'ens interne registre.
- celler mister indholdet efter en tid...

Den sidste fejl kan, som de øvrige forekomme både ved statiske og dynamiske RAM-kredse. Ved Dynamiske RAM-kredse -DRAM'er, skal man mistænke refresh-logikken, idet der normalt skal foretages refresh hver 2 m sek. for ikke at miste indholdet. (Dette gælder hvis kredsene arbejder ved deres maximale omgivelsestemperatur, medens der ved "stuetemperatur" kan gå typisk 100 m sek. mellem hver refresh uden at miste indhold). Hvis fejlen optræder på statiske RAM-kredse er det sket at indholdet først mistes efter flere sekunder, hvilket betyder at en RAM-test ikke med sikkerhed vil finde fejlen med mindre der tages hensyn til netop denne fejltype.

Datafejl på ROM, PROM og EPROM:

Af disse tre hukommelsestyper ses de fleste fejl på PROM'en og EPROM'en. Ud over at fungere som programhukommelse og fastdatahukommelse, f. eks. som kodeconverter vinder TTL-PROM'er indpas som decodere i chip-selectlogikken. Ved tidlige typer er det observeret at de har "repareret" sig selv, idet et bit der var programmeret pludselig kom igen. Ny teknologi har dog gjort denne fejltype mindre hyppig.

EPROM'en er den hukommelse der er mest tilbøjelig til at vise fejl. Det skyldes både den måde bitinformation lagres på, samt den håndtering kredsen er udsat for. Hukommelseseffekten beror på at en "flydende" gate i en MOS-transistor skal forblive op- eller afladet. Flere forskellige mekanismer i kredsen modarbejder dette, specielt i forbindelse med høj drifttemperatur. Har man mistanke til bitfejl i en EPROM, kan man prøve at opvarme (hårdtørre) eller afkøle (kuldespray) kredsen. Hvis fejlen derved forsvinder eller ændrer karakter er kredsen sandsynligvis defekt.

For at reducere chancen for fejl på EPROM'er er det vigtigt at overholde nogle simple regler for brugen:

- Pas på undersletning! Vær sikker på at det anvendte sletteudstyr giver en tilstrækkelig UV-dosis. Kontroller ligeledes at vinduet er fri for limrester....
- Pas på underprogrammering! Kontroller at det anvendte

programmeringsudstyr overholder programmeringsdata for den aktuelle EPROM...

-Pas på lyspåvirkninger efter programmeringen! Sørg for en effektiv afdækning af vinduet, en "syltetøjsmærkat" er ikke god nok, men brug f. eks. sort PVC-tape eller selvklæbende aluminiumfolie.

For yderligere oplysninger henvises til Elektronikcentralens rapporter ECR-92 og ECR-102.

----- Portkredse, simple og programmerbare -----

En stor procentdel af fejlene i et mikrocomputer-system viser sig erfaringsmæssigt at ligge i I/O-kredsløbene. Grunden til dette er at disse kredsløb ofte er hårdt belastede, medens de øvrige kredsløb i computeren arbejder under mere stabile og kontrollerede forhold. Ud over direkte fejl i programmerbare portkredse, kan disse også give problemer ved ikke at blive initialiseret korrekt. Det vil sige at en tilsyneladende uskyldig programfejl kan få helt uventede konsekvenser f. eks. i form af uønskede interrupt.

I I/O-kredsløb med galvanisk adskillelse opnået ved hjælp af photocouplere sker det ret ofte at disse mister overføringsevnen, idet lysdioden har vist sig at være ret sart.

For yderligere henvises til Elektronikcentralens rapporter om photocouplere ERC-61 og ERC-96.

----- Fejl på hjælpe kredse -----

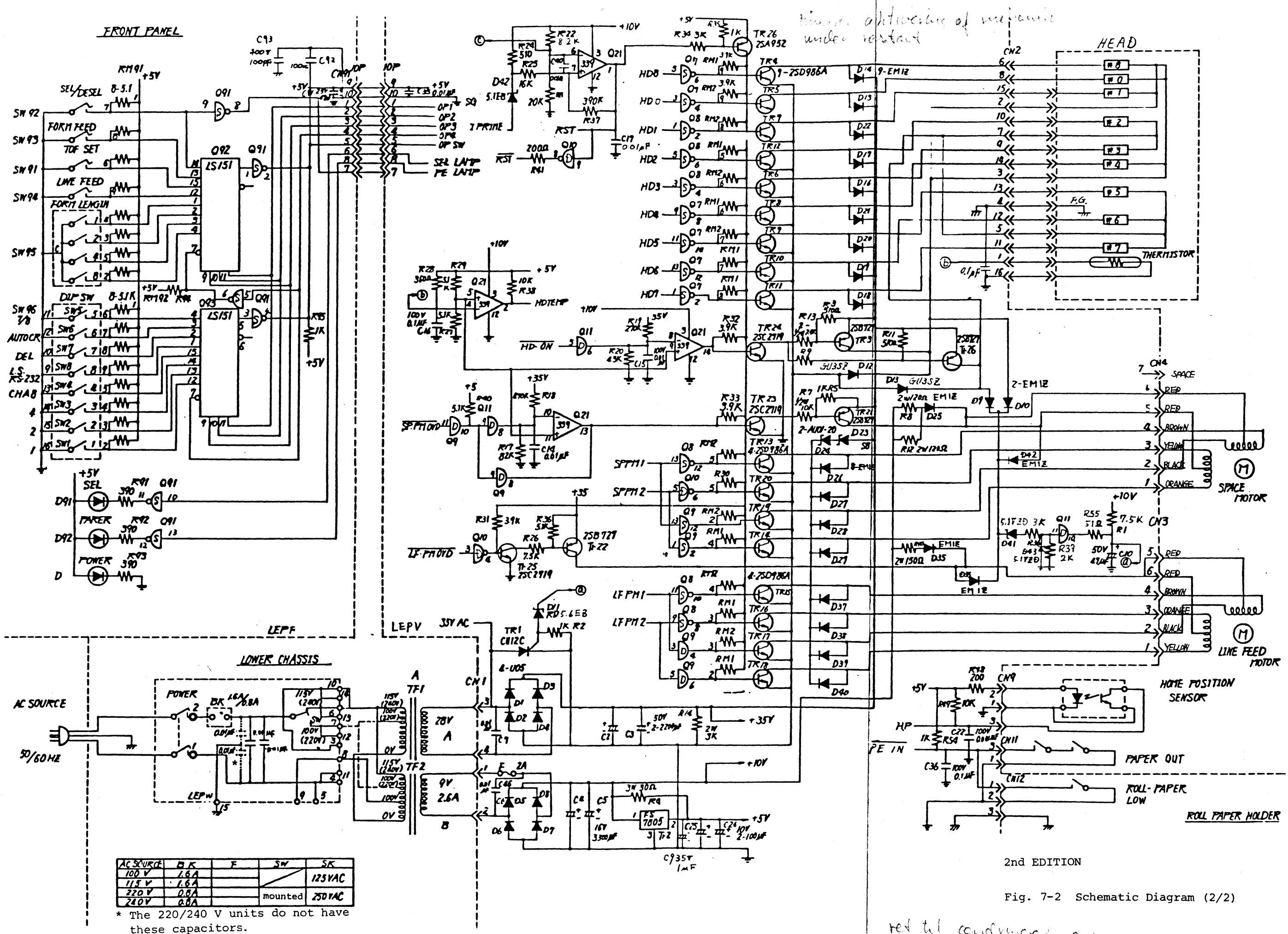
Det største opbud af almindelig logik findes i en typisk mikrocomputer omkring chip-select logikken. Denne er opbygget af diverse decodere, gates og PROM'er samt strappefeller hvor den endelige organisering af memory-mappet finder sted med loddekolbe eller wire-wrap. De anvendte kredse er normalt fra TTL-familien for at få tilstrækkelig høj arbejdshastighed.

En fejl i chip-select logikken kan bevirke at:

-En bestemt kreds bliver ikke selectet...
-En kreds bliver selectet i stedet for en anden...
-To eller flere kredse bliver selected samtidigt...

O.S.V.

Chip-select logikken kan kontrolleres med signaturanalysator og sammenligning mod kendte gode signaturer, eller ved at indspille chip-select signalerne sammen med adresserne i en logikanalysator, og se efter at kun den rigtige kreds selectes i forhold til adresserne.



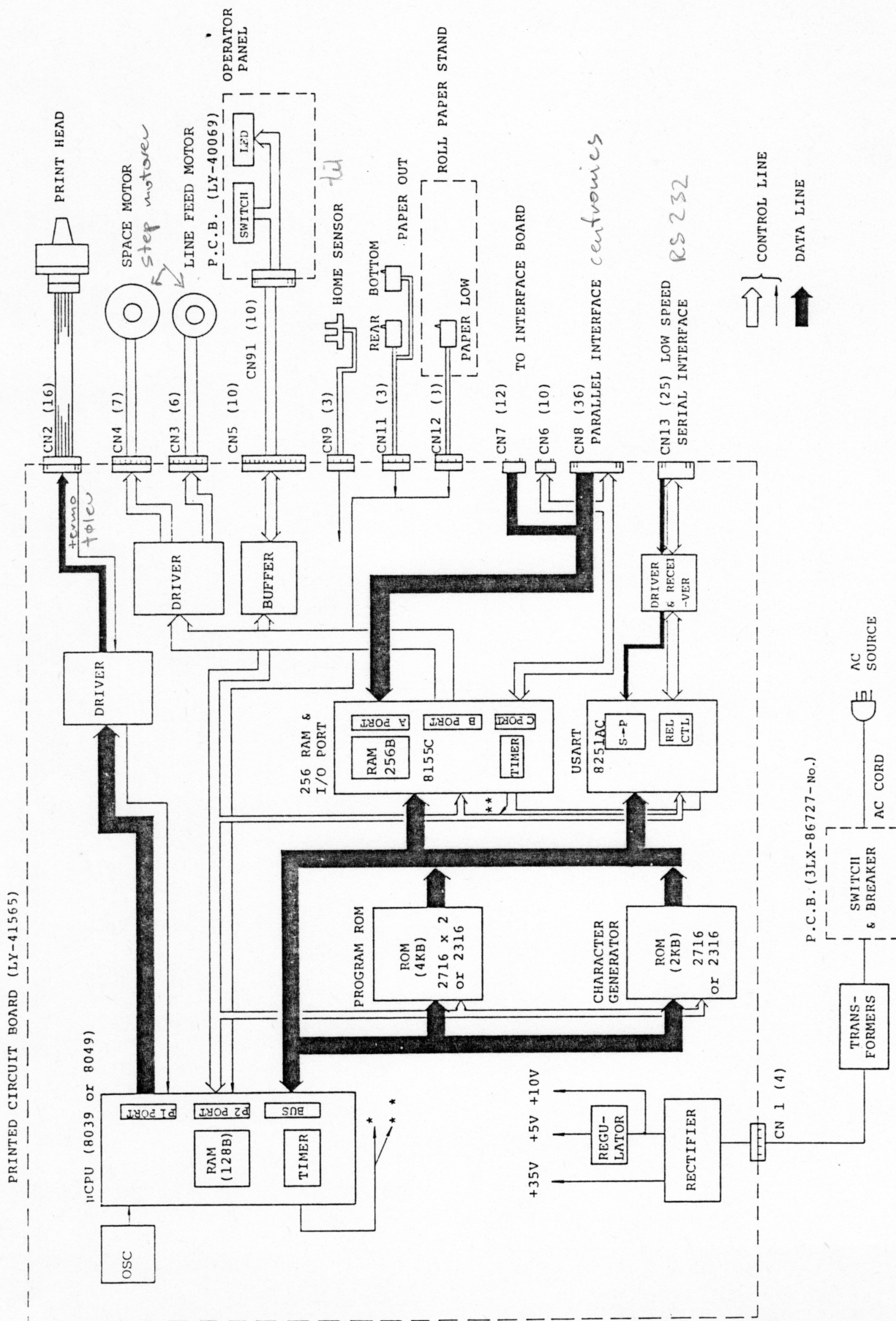


Fig. 5-8 Block Diagram

----- PIV 1 systembeskrivelse -----

PIV 1 står for Programmerbart Interaktivt Voltmeter model 1. PIV 1 er et 16 kanal programmerbart voltmeter. Opstillingen består af en computer med tilhørende hukommelse og en 16 kanal analog/digital converter. Kommunikation og udlæsning sker via en skærmterminal og en matrixprinter. Programmeringen sker på basis af softkeys hvis øjeblikkelige funktion fremgår af den nederste linie på skærmen. Til softkeys bruges den øverste række taster på keyboardet.

Start og programmering:

Efter spændingstilslutning og reset vil der fremkomme følgende skærbillede:

PRINTER:	UR STOPPET

(1)MENU	(3)KANAL (4)SÆT U (5)MÅL

(1)MENU :

Ved indtastning af taltast 1 vælges MENU-mode. I denne mode opsættes voltmetret kanal for kanal idet der skal tages stilling til følgende målebetingelser:

Om den pågældende kanal skal være til eller fra.
Den nominelle spænding for kanalen.
Spændingstolerance for kanalen.

Voltmetret kan måle positive spændinger mellem 0 og 5 volt. Den nominelle spænding for en kanal vælges til at ligge mellem disse to grænser. Overskrides tolerancen for en kanal under målingen vil det give anledning til udprintning af en fejlliste der indeholder oplysning om kanalnummer, den nominelle spænding (reference), den målte værdi samt tidspunktet for målingen hvis uret har været i gang.

(3)KANAL:

I KANAL-mode er PIV 1 sat op som et 1-kanal voltmeter. Ved indtastning af taltast 3 fremkommer følgende skærbillede:

PRINTER:	UR STOPPET

ANGIV KANALNR. = _	

Herefter indtastes et tal mellem 1 og 16 for den kanal man ønsker at måle på, hvorefter den målte spænding løbende vises på skærmen. KANAL-mode afsluttes med ESC.

(4)SÆT U:

Efter reset vil uret være stoppet, hvilket vil fremgå af de udprintede fejlrapporter ved at tidspunktet konstant vises som 00 : 00 : 00. Uret sættes og startes ved at indtaste taltast 4 hvorefter følgende skærbillede fremkommer:

PRINTER:	UR STOPPET

ANGIV AKTUEL TID:	
HH : MM : SS	

Hvis der blot tastes retur vil uret starte på nul og opdateres hvert 5. sekund. Denne fremgangsmåde kan benyttes hvis man ønsker de enkelte målinger tidsangivet relativt til et eventuelt forsøgs begyndelse. Hvis målingerne ønskes angivet med absolut tid indtastes den aktuelle tid inden der tastes retur, hvorefter uret sættes til det indtastede tidspunkt. Om uret er startet eller stoppet, samt starttidspunktet vises øverst på skærmen.

(5)MÅL:

Ved indtastning af taltast 5 starter PIV 1 som voltmeter i overensstemmelse med den menu der er sat op i MENU-mode. Hvis en aktiv kanal overskrider sin tolerance vil det blive vist i form af en fejludskrift til printer eller skærm. Målemode afsluttes med ESC.

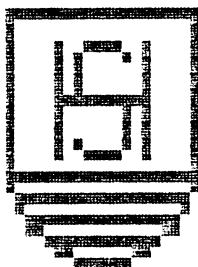
Eksempel på fejlrapport:

Menuen havde følgende udseende:

KANALO1:TIL, VOLT:1.50, TOLLERANCE +/-0.50
 KANALO2:TIL, VOLT:3.00, TOLLERANCE +/-1.00
 KANALO3:TIL, VOLT:4.60, TOLLERANCE +/-0.25
 KANALO4:FRA,

....og alle øvrige FRA.

Uret er startet klokken 12 : 30 : 00.



HÅNDVÆRKERSKOLEN
 SØNDERBORG TEKNISKE SKOLE
 SKOVVEJ 26
 6400 SØNDERBORG

FEJLRAPPORT

Kanalnr	Reference	Målt værdi	Tidspunkt
02	3.00	4.00	12 : 30 : 40
03	4.60	4.34	12 : 30 : 50
03	4.60	4.29	12 : 30 : 50
03	4.60	4.10	12 : 30 : 55
03	4.60	4.07	12 : 30 : 55
03	4.60	4.32	12 : 30 : 55
01	1.50	0.81	12 : 31 : 00
01	1.50	0.00	12 : 31 : 00
01	1.50	2.00	12 : 31 : 10
01	1.50	2.15	12 : 31 : 10
01	1.50	2.95	12 : 31 : 10
01	1.50	4.22	12 : 31 : 15
01	1.50	2.23	12 : 31 : 15
03	4.60	4.85	12 : 31 : 25
03	4.60	4.87	12 : 31 : 30
03	4.60	4.92	12 : 31 : 30
03	4.60	1.12	12 : 31 : 30
02	3.00	1.48	12 : 31 : 30
02	3.00	1.98	12 : 32 : 05
02	3.00	1.91	12 : 32 : 05

----- Øvelses-software i PIV 1 -----

Ud over den software der styrer PIV 1 funktionen, er der tre andre funktioner til rådighed. Disse initialiseres ved hjælp af vippeomskifterene til venstre på forpladen. Efter spændingstilslutning resettes PIV 1 hvorefter man kan vælge en af de tre andre muligheder.

MON:

Siemens Monitor program ved hjælp af hvilket det er muligt at undersøge og ændre hukommelsesindhold, undersøge og ændre registerindhold, disassemblere kode og starte programeksekvierung o.s.v. For nærmere oplysninger henvises til SMP-MON4 V2.0 manualen. MON interrupter CPU'en via RST 7,5. OBS. Efter påvirkning af MON-vippekontakten skal der indtastes 4 x space fra skærmskravlinen til bestemmelse af baud-raten hvorefter monitoren er klar.

TEST:

TEST-program der tester programhukommelsen ved sammenligning med en checksum, foretager en RAM-test, og tester MUART-kredsen på CPU-kortet. Til indikation af resultatet anvendes LED-indikatorerne 2, 3 og P. TEST interrupter CPU'en via TRAP.

LSAØV:

LSA-øvelse initialiserer det programeksempel der er anvendt til illustration af logikanalysator PM3543's triggermodes. Se side 3-11. LSAØV interrupter CPU'en via RST 6,5.

----- LED-indikatorer -----

På interface-modulet findes 4 LED-indikatorer mærket P, 1, 2 og 3.

P: Indikerer overførsel af data til printeren, samt om RAM-testen var OK i TEST-mode.

1: Vagthund, der starter sammen med uret, skal lyse konstant.

2: Indikerer at testen er startet i TEST-mode.

3: Indikerer ROM-test OK.

På powersuplyen er der 5 LED-indikatorer. Den øverste grønne indikerer et netspænding er tilsluttet. De fire gule viser om DC-udgangene er aktive. Disse tændes og slukkes med afbryderen øverst til venstre på PIV 1's bagside og med nøleafbryderen.

hælder CPU hvis ram defekt

----- PIV 1 hardware -----

PIV 1 er bygget op omkring Siemens kredsløbskort og består af følgende enheder:

Rack med bagplan
Powersupply, +5V/10A, -5V/0.5A, +/-15V/1.6A
Centralcomputer med 8085 CPU, 16k PROM og 2k RAM
CMOS RAM-lager, 2k med batteribackup
A/D-converter, 16 kanaler/ 12 bit opløsning

Derudover indeholder racket interfacekredsløb mellem MUART'en i centralcomputeren og printeren og skærmterminalen. Dette kredsløb er opbygget på et universalprint med wirewrap. Ud over interfacekredsløbene omfatter dette modul også diverse LED-indikatorer.

Som input til A/D-converteren findes i racket yderst til venstre 16 potentiometre hvor over der kan udtages 16 spændinger mellem 0 og 5V. Kontakter til generering af restart- og interruptsignaler findes også i dette panel.

===== ADVARSEL =====

På grund af diverse specialsignaler til de forskellige kredsløbsmoduler kan disse IKKE flyttes rundt i racket. En overtrædelse af dette vil i værste fald medføre varige skader på modulerne.

=====

----- Modulbeskrivelser -----

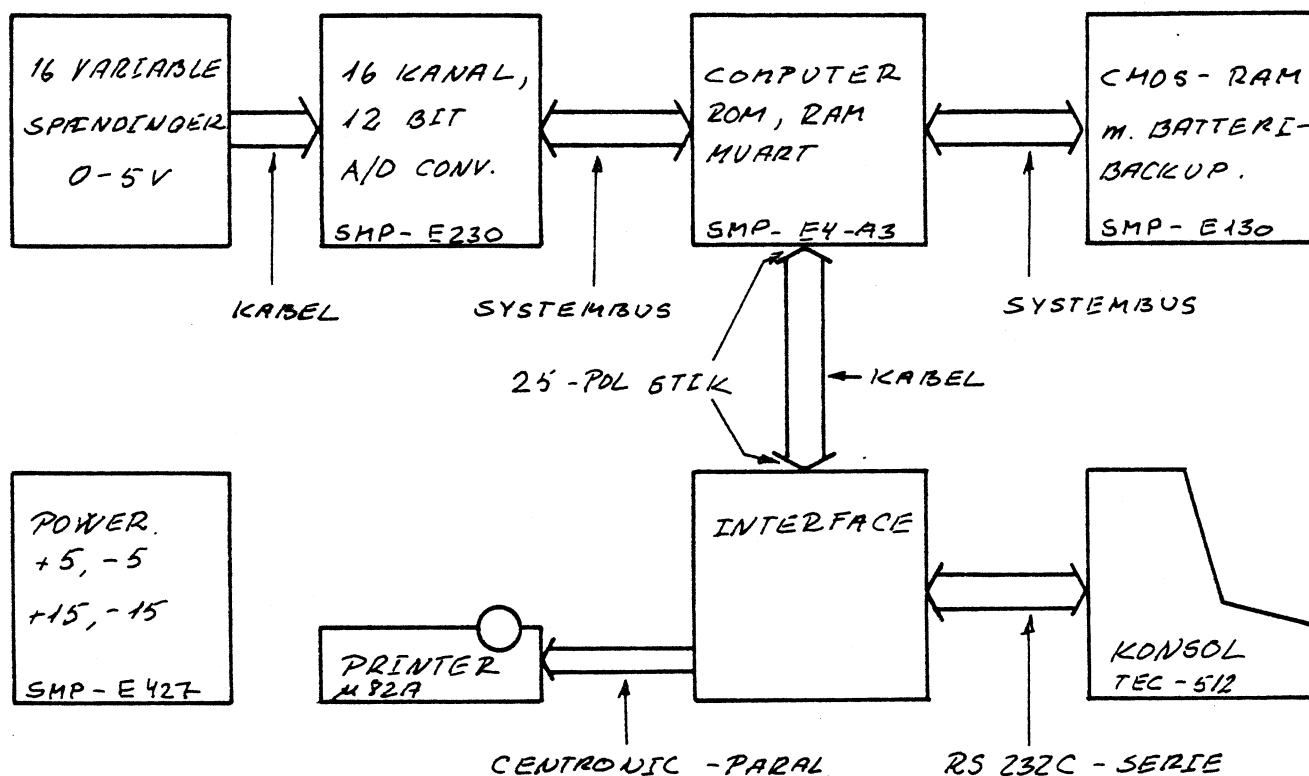
Bagplan:

Bagplan type SMP-S401-A10 kan optage 10 kredsløbskort. Fatningerne er 96-polede, men der er dog kun ført 63 af disse igennem i bagplanet, se dokumentation. I bagplanet ligger følgende spændinger:

a32, c3: +5
a2: -5
a31: +15
a1: -15
c2, c31: gnd.

Bagplanet er konstrueret som et 6-lags print med et gennemgående stelplan som sikrer mod overhøring mellem signalledere samt eksterne forstyrrelser. Standard bussignalerne er afsluttet med modstande for af formindske reflektioner og ringning.

Blokdiagram:



Netdel:

Netdel SMP-E427-A2 er en switch-mode strømforsyning med 4 udgangsspændinger:

+ 5 V / 10 A
- 5 V / 0,5 A
+15 V / 1,6 A
-15 V / 1,6 A

Alle udgange er kortslutningssikre, og strømforsyningen er sikret mod termisk overbelastning. Ved netudfald genererer strømforsyningen et 'power fail' signal mindst 10 m sek. inden udgangsspændingen bryder sammen. Ved fuld belastning (10A) vil et netudfald på mindre end ca. 30 m sek. ikke mærkes i udgangsspændingen. Strømforsyningen tændes og slukkes via SD (shut down) der afbryder DC-udgangene.

Centralcomputer:

Centralcomputeren SMP-E4-A3 er bygget op over en 8085 CPU. Kortet er strappet op til at rumme 16 k ROM/EPROM, begyndende på adr. 0000H. Derudover er der RAM-hukommelse fra 4000H til 47FFH.

Kommunikationen varetages af en MUART (Multifunktion Universal Asynkron Receiver Transmitter) idet port 2 anvendes til printeroutput, port 1 til LED-indikatorer samt input fra printeren og serie I/O anvendes til skærmterminalen.

CMOS-RAM:

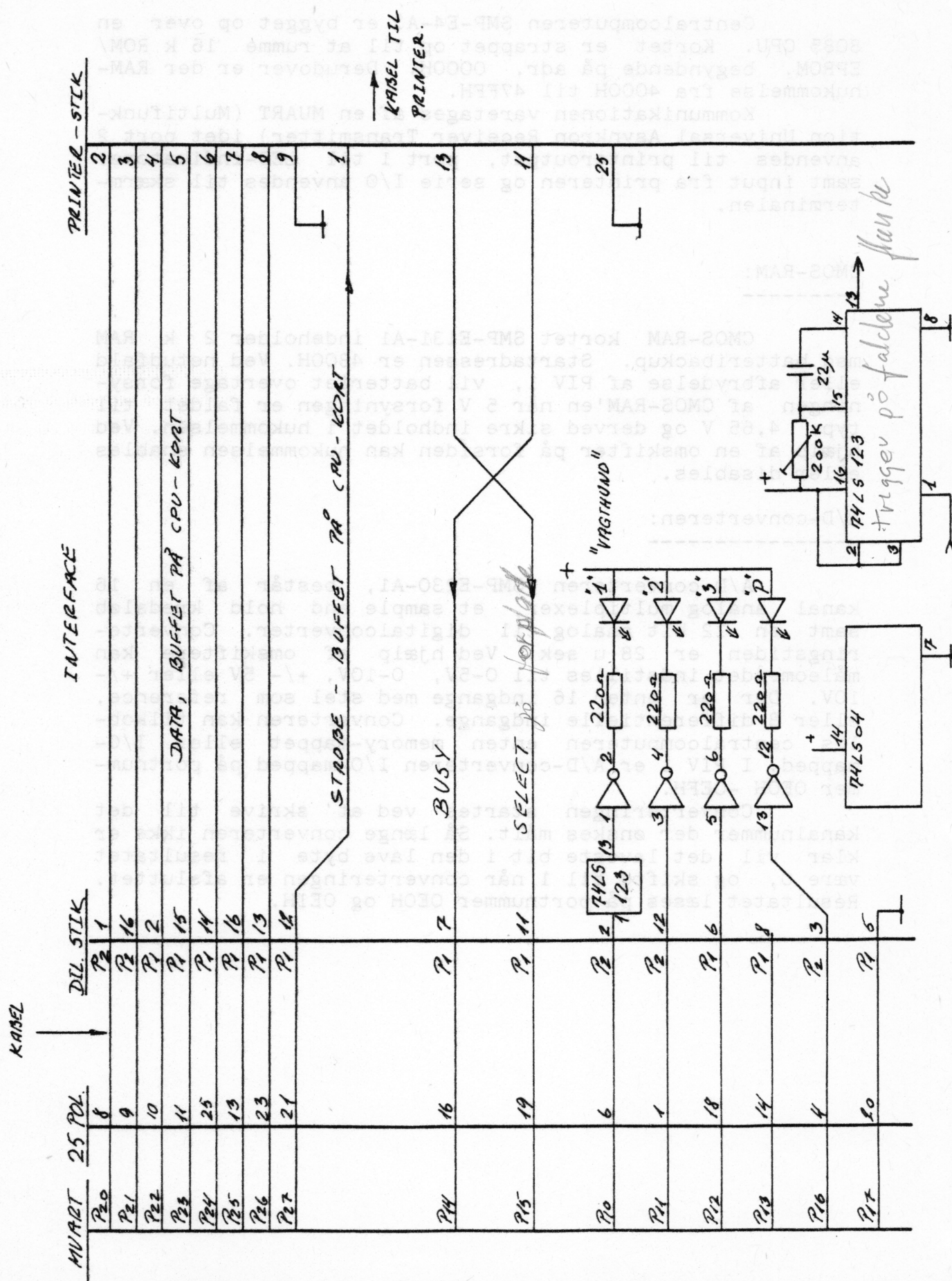
CMOS-RAM kortet SMP-E131-A1 indeholder 2 k RAM med batteribackup. Startadressen er 4800H. Ved netudfald eller afbrydelse af PIV 1, vil batteriet overtage forsyningen af CMOS-RAM'en når 5 V forsyningen er faldet til typ. 4,65 V og derved sikre indholdet i hukommelsen. Ved hjælp af en omskifter på forsiden kan hukommelsen enables eller disables.

A/D-converteren:

A/D-converteren SMP-E230-A1, består af en 16 kanal analog multiplexer, et sample and hold kredsløb samt en 12 bit analog til digitalconverter. Converteringstiden er 28 u sek. Ved hjælp af omskifttere kan måleområdet indstilles til 0-5V, 0-10V, +/- 5V eller +/- 10V. Der er enten 16 indgange med stel som reference, eller 8 differentielle indgange. Converteren kan tilkobles centralcomputeren enten memory-mappet eller I/O-mapped. I PIV 1 er A/D-converteren I/O-mapped på portnummer OE0H -OE1H.

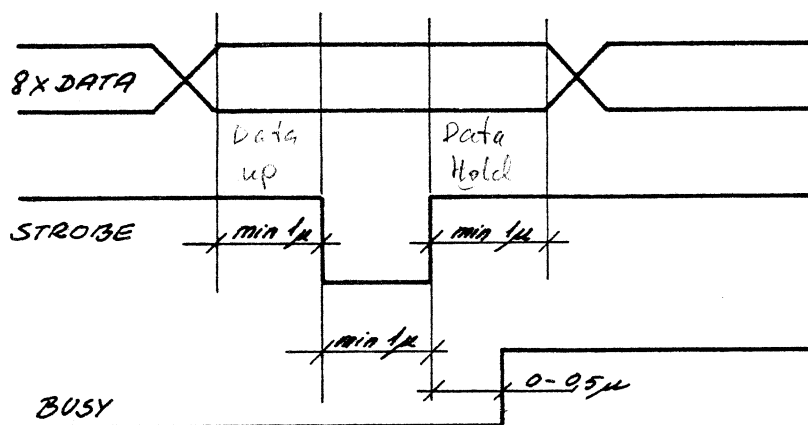
Converteringen startes ved at skrive til det kanalnummer der ønskes målt. Så længe converteren ikke er klar vil det laveste bit i den lave byte i resultatet være 0, og skifte til 1 når converteringen er afsluttet. Resultatet læses på portnummer OE0H og OE1H.

----- Interface til printer og LED-indikatorer -----



Printer interface:

Data til printeren overføres parallelt via et såkaldt Centronics snit. Signalerne omfatter 8 databit fra MUART'ens port 2 der alle er buffet på CPU-kortet ved hjælp af 2 stk SN7408 AND-gates. Mod printeren sendes en strobe-puls med hvilke den overførte databyte indlæses i printerens buffer, se pulsplan. Fra printeren modtager computeren to controlsignaler. SELECT indikerer med et high niveau at printeren er klar til at modtage data. Select følger LED-indikatoren på printerens (Microline 82A) forside. BUSY indikerer med et high niveau at printeren ikke kan modtage data, f. eks. fordi bufferen er fuld og den er igang med en formfeed.



Vagthund:

Vagthunden består af en LED-indikator og en gen-trigbar monostabil multivibrator der skal triggere af en faldende flanke fra MUART'ens port 1 bit 0. Pulstiden indstilles således at multivibratoren ikke når at time ud når PIV 1 er i orden. Vagthunden triggere fra UR-rutinen, og vil som følge deraf kun lyse når uret er startet. Fejl i udstyret vil indikeres ved at vagthunden slukker eller blinker.

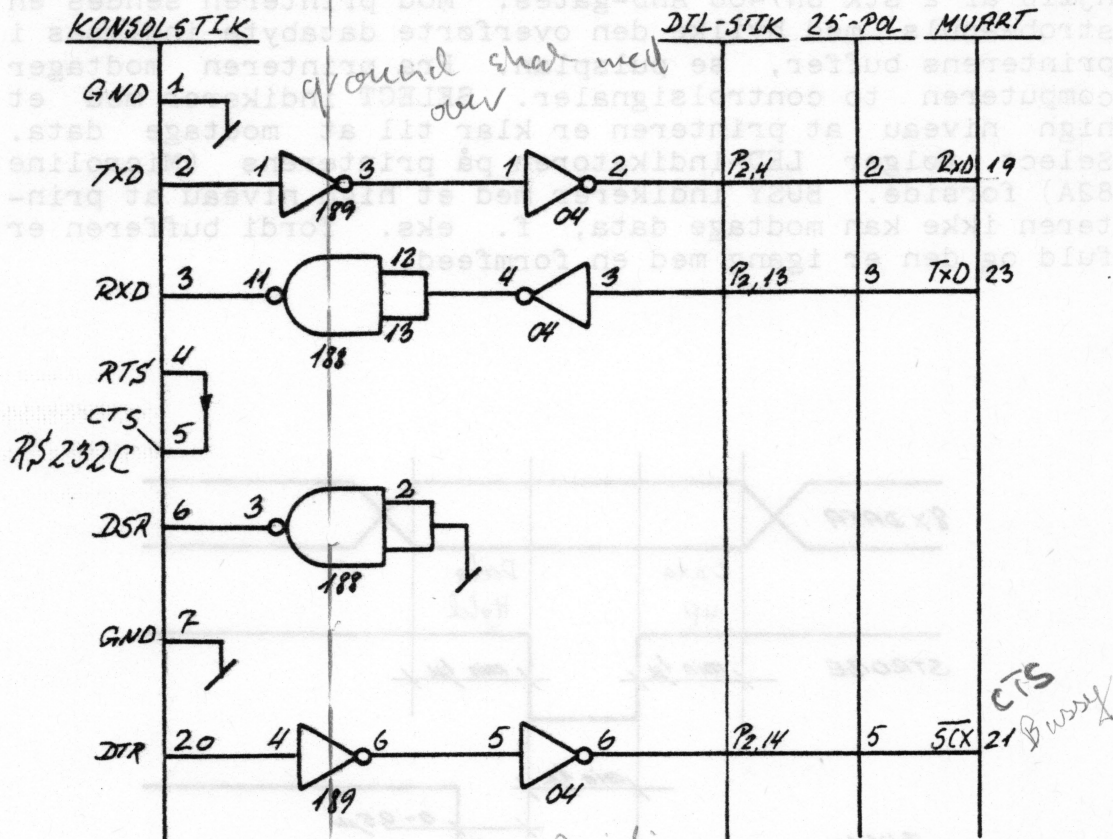
----- Serieinterface til skærmterminal -----

forbudt område for
logik probe brug

Logik probe kan anvendes

SCOPE

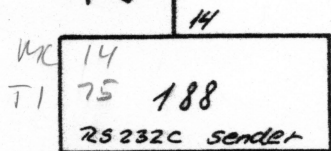
INTERFACE



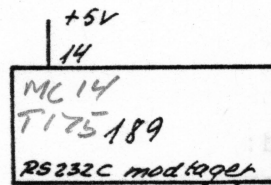
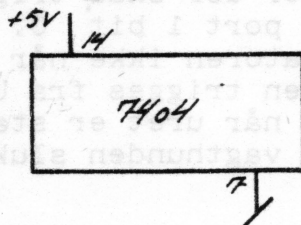
V24 standard
Negativ logik

TTL positiv
logik

+15V 3,3V Zener



+15V 3,3V Zener



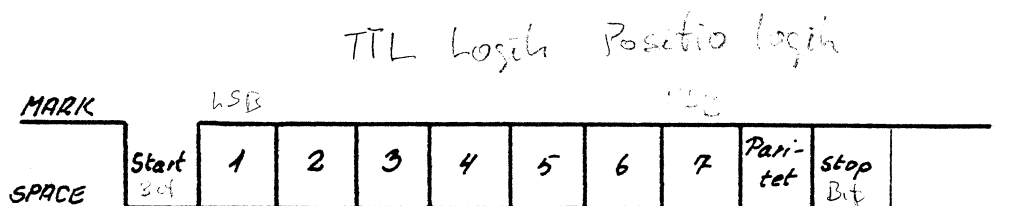
Skærmterminal interface:

Skærmterminalen sender og modtager data via en asynkron RS 232C serieinterface. En tilsvarende Europæisk standard findes i CCITT's V24 og V28. Oprindeligt er RS 232C beregnet til kommunikation mellem en terminal og et MODEM, hvilket navnene på signalerne hentyder til. Der er dog intet i vejen for at at sammenkoble andre typer udstyr over en RS 232C interface, blot kan der blive tale om at krydse visse signaler.

Dataformat:

Data overføres som en serie af bit med det laveste bit først. Den normale ordlængde er 7 bit. Inden første databit overføres et startbit, som modtageren bruger til at synkronisere på, og efter databittene sendes et eller flere stopbit. Imellem databit og stopbit kan indføjes et paritetsbit til kontrol af dataoverførselens rigtighed.

Transmissionshastigheden udtrykkes i BAUD (udtales "bo") og er bit pr. sekund. De standardiserede transmissionshastigheder er 50, 75, 100, 110, 150, 200, 300, 600, 1200, 2400, 4800, 9600 og 19200 baud. I PIV 1 er baudraten 2400.



----- Signalnavne (V24)-----

TXD : Transmitted data fra terminal mod PIV 1.
 RXD : Received data fra PIV 1 til terminal.
 RTS : Request To Send. Signal fra terminal der med en space indikerer at terminalen er ~~i færd~~ med at sende en karakter. *er klar til at*
 CTS : Clear To Send. Signal fra et MODEM til terminalen der med en space indikerer at modemmet har forbindelse og er klar til at sende. Da PIV 1 ikke leverer et CTS signal lader man terminalen selv gøre det ved at sammenkoble RTS og CTS.

signalnavne fortsat:

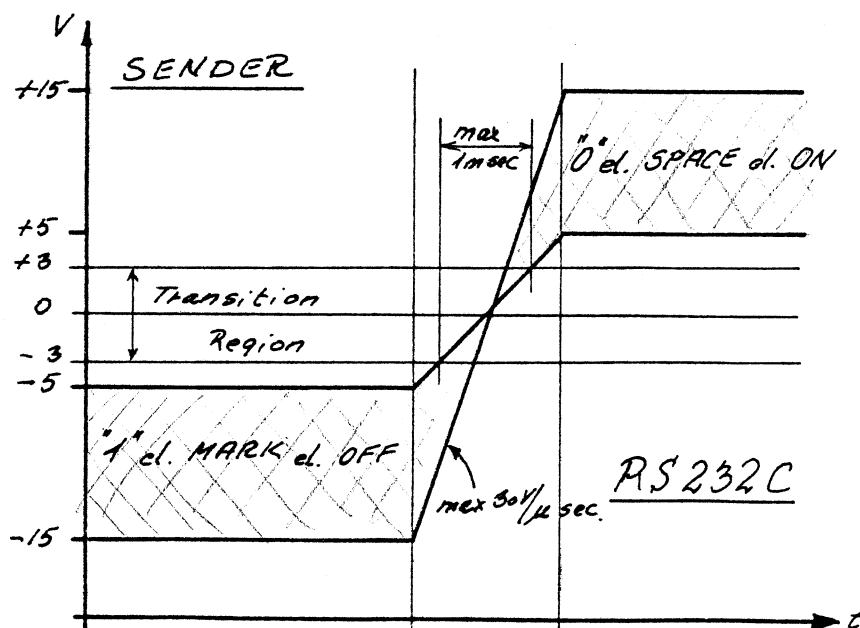
DSR : Data Set Ready. Signal til terminalen fra et modem der med et space fortæller at modemmet er forbundet til telefonlinien og er tændt. Da PIV 1 ikke leverer et sådant signal kobles DSR til en fast space leveret af en RS 232 senderkreds.

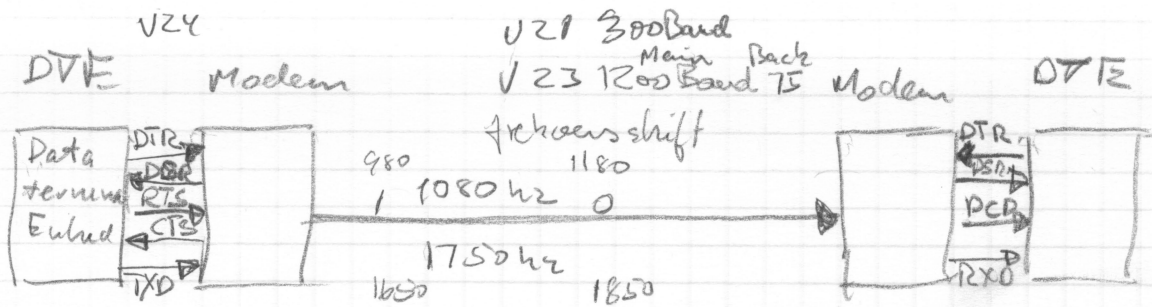
DTR : Data Terminal Ready. Signal fra terminalen der med et space indikerer at terminalen er tændt og i REMOTE-mode.

----- RS 232C signalniveauer (V28)-----

Sender:

Spændingerne i et RS 232C snit afviger fra "almindelig logik" ved at arbejde med to polariteter, således at det logiske niveau er knyttet til polariteten af spændingen, og ikke dens absolutte størrelse. For en RS 232C sender gælder at dens udgangsspænding skal ligge mellem 5 og 15V belastet med 3 - 7 kohm, være kortslutningssikker og ubelastet må udgangsspændingen ikke overskride 25V. Den positive polaritet er defineret som SPACE, ON og logisk 0, og tilsvarende er den negative polaritet tildelt betydningen MARK, OFF og logisk 1.





DTR Data Terminal Ready

DSR Data Set Ready *Power on of socket to send*

RTS Request to send

CTS Clear to send *→ RTS Ready for send
Bereitschaft klar*

DCD Data carrier detect

TXD

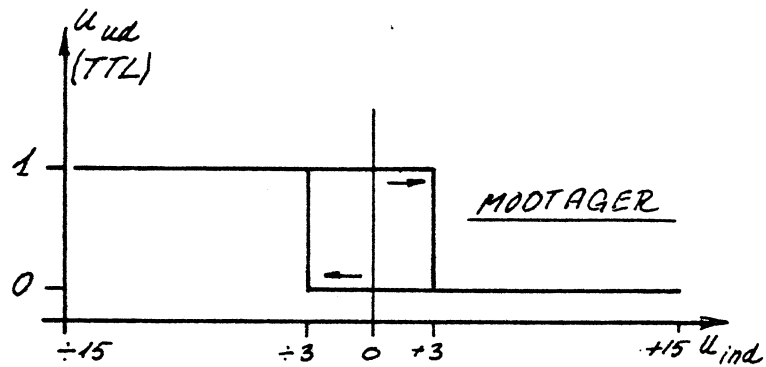
RXD

	Main	1200	Band
			0
V23	1300	1700	2100
	Back	75	Band
	390	420	450

Stelle für schauen da dem isthe Kfrescher
für CPU.

Modtager:

Modtageren er udstyret med hysteres (schmitt-trigger) på $\pm 3V$ således at støj på transmissionslinien ikke får nogen indflydelse.



----- PIV1 FIRMWARE -----

Firmware'n til PIV1 er anbragt i 3 EPROM'er af typen 2732. Firmware'n er opbygget ved sammensætning af to uafhængige software pakker, der er udviklet hver for sig. Den ene del er en monitor, der er udviklet af Siemens. Den anden del er styreprogrammet til PIV1, denne del består af følgende moduler:

GLOBAL: Indeholder tekstrækker, tekstpointere samt globale variable.

Module GLOBAL of FORLIS REL

Length of Program 1499

Length of Data area 315

Entry point(s):

AMOURS 0105"	AKANNR 012A"	AMENNR 0137"	AMINTS 0108"
ASCNDS 010B"	ATOLL 0132"	AVOLT 012D"	BELL 0007
BHOURS 0117"	BLANK 0020	BWINTS 011B"	BECNDS 0119"
BUFFER 0000"	CLKDTT 053F'	CLSCNT 0008'	CMDR1 00F0
CMDR2 00F1	CMDR3 00F2	CMDRP1 00F4	COLUMN 0121"
CR 0000	CUMOVY 0577'	EQ1 0088	EDTAB 0000
ERASEL 0531'	ESC 001B	ETX 0003	FALSE 0000
INPTR 0122"	INT0 0001	INT1 0002	INT2 0004
INT3 0008	INT4 0010	INT5 0020	INT6 0040
INT7 0080	INTADR 00F6	INTDIR 00F6	INTENR 00F6
INTMSK 00F6	ISCNTT 04E1'	KANALNR 0129"	LF 000A
LINE 0120"	MENUT 0509'	MODER 00F3	NXTPTR 0100"
OUTPTR 0124"	PORT1 00F8	PORT2 00F9	PAFINT 0553'
PRNLIN 055B'	PRNRYT 054B'	PRSTT 054F'	PSTART 0557'
PTABPT 011E"	PTBPTR 011D"	RCBR 00F7	SHOURS 010E"
SMINTS 0111"	SPECFR 00FF	SECNDS 0114"	STATR 00FF
STED 0126"	STOPUR 013A"	STPTR 0000"	TABPTR 011A"
TCLEAR 052F'	TCTRLV 0535'	TESCAP 05D1'	TIMER1 00FA
TIMER2 00FB	TIMER3 00FC	TIMER4 00FD	TIMER5 00FE
TINVRE 053B'	TMR 00F7	TOFKAN 057D'	TOKKAN 0585'
TRDY 0104"	TRUE 00FF	TSFRA 0595'	TSFRA2 0599'
TSOFFA 05A3'	TSOFT1 05AD'	TSTART 04F9'	TSTIL 059F'
TSTOP 04F5'	TTAB1 050D'	TTAB10 052B'	TTAB57 0527'
TTABZW 0523'	TVOLT1 05B7'	TVOLT2 05C3'	TXTPTR 0102"

AP8256: Indeholder restartlisten samt initiali-
sering af 8256.

Module AP8256 of FORLIB REL

Length of Program 106
Length of Data area 2

External reference(s):

ADPRS	004B'	AMOURS	004E'	CHARIN	0022	CLK	001B
CMDR1	0000	CMDR2	0000	CMDR3	0000	CMDRP1	0000
INISID	0063'	INTADR	0000	INTDIR	0000	INTENR	0000
INTMSK	0000	ISCNT7	0000	LSACV	0035	MODER	0000
NXTPTR	0048'	PORT1	0000	PORT2	0000	PANTCH	0032
RCBR	0000	SPECFR	0000	STATR	0000	STOPUR	003E'
STPTR	001C'	TABPTR	0045'	TCHOUT	002A	TEST	0025
TIM1	0017'	TIMER1	0000	TIMER2	0000	TIMER3	0000
TIMER4	0000	TIMER5	0000	TMBR	0000	TRDY	0000
VALB	0067'						

SERIO: Indeholder programmel der sørger for
kommunikation med skærmterminalen.

Module SERIO of FORLIB REL

Length of Program 1244
Length of Data area 2

Entry point(s):

ASCFL	0220'	CHARIN	000E'	CO	002E'	FETCHC	003B'
INISID	004E'	INITSK	04BA'	MOVOUR	006D'	STTXTD	00D4'
TCHOUT	00F1'	VALB	0101'				

External reference(s):

AKANNR	01D1'	AMENNR	0000	AVOLT	01D9'	BCDASC	022F'
BELL	0000	BLANK	0000	BUFFER	004F'	CMDR1	0000
CMDR2	0000	CMDR3	0000	CMDRP1	0000	COLUMN	0060'
CR	0000	CUMCVT	0080'	EKANAL	0153'	EDI	0000
EDTAB	0337'	ERASEL	01F9'	ESC	0000	ETX	0000
FALSE	0000	INPTR	0052'	INT0	0000	INT1	0000
INT2	0000	INT3	0000	INT4	0000	INT5	0000
INT6	0000	INT7	0000	INTADR	0000	INTDIR	0000
INTENR	0000	INTMSK	0000	ISCNTT	048B'	KANALNR	02CB'
KANPTR	049C'	LF	0000	LINE	0493'	MENUT	0105'
MODER	0000	NEWCLK	015C'	NXTPTR	00DC'	OUTPTR	0053'
PORT1	0000	PORT2	0000	RCBR	0000	SPECFR	0000
START	016B'	STATR	0000	STED	0074'	STOPUR	04C4'
TABPTR	04DE'	TCTRLV	0000	TESCAP	0162'	TIMER1	0000
TIMER2	0000	TIMER3	0000	TIMER4	0000	TIMER5	0000
TINVRE	0000	TMBR	0000	TCFKAN	020B'	TONKAN	0217'
TRUE	0000	TSFRA	0000	TSFRA2	02DE'	TSDFFA	02F8'
TSDFT1	02AD'	TSTART	04CD'	TSTIL	0326'	TSTOP	04D3'
TVOLT1	0361'	TVOLT2	0409'	TXTPTR	00E0'		

PARIO: Indeholder alt programmet der sørger for kommunikation med printeren.

Module PARIO of FORLIB REL

Length of Program 548
Length of Data area 5

Entry point(s):

PRNTCH 0011' START 01D4' STPRNT 0089'

External reference(s):

AKANNR	0198'	ASCFL	01A3'	ATOLL	01A0'	AVOLT	0181'
BCDASC	0192'	BELL	0000	BLANK	0000	CMDR1	0000
CMDR2	0000	CMDR3	0000	CMDRP1	0000	COLUMN	00E5'
CONV	0203'	CR	0000	EDI	0000	EDTAB	0000
ESC	0000	ETX	0000	FALSE	0000	FETCHC	0271'
GETDAT	0200'	HEXBCD	018E'	INPTR	0266'	INT0	0000
INT1	0000	INT2	0000	INT3	0000	INT4	0000
INT5	0000	INT6	0000	INT7	0000	INTADR	0000
INTDIR	0000	INTENR	0000	INTMSK	0000	KAN101	01E4'
KAN102	0000	KANALNR	0279'	KANPTR	025F'	LF	0000
LINE	00E2'	MODER	0000	MOVOUR	00EA'	OUTPTR	026A'
PORT1	0000	PORT2	0000	PRFINT	0081'	PRNLIN	013F'
PRNRYT	00D9'	PRSTT	0028'	PSTART	01AE'	PTABPT	01C2'
PTBPTR	00AA'	RCBR	0000	RESULT	0206'	SPECFR	0000
STATR	0000	STDTIM	00CC'	STOPUR	0000	STXTX0	00DF'
TABPTR	00DC'	TIMER1	0000	TIMER2	0000	TIMER3	0000
TIMER4	0000	TIMER5	0000	TMBR	0000	TRUE	0000
WCHD6	0000						

TIMER: Indeholder alt programmet der servicer timerne i MUART 8256.

Module TIMER of FORLIB REL

Length of Program 124

Length of Data area 2

Entry point(s):

STDTIM 0000' STTIM1 000D' TIM1 0022' WCHDB 0049'

External reference(s):

INPTR 0054' OUTPTR 0058' TRDY 0068'

UHR: Indeholder programmet der omregner time-
'rudløb til tid, samt start og initiali-
sering af uret.

Module UHR of FORLIB REL

Length of Program 545

Length of Data area 1

Entry point(s):

ACPRS 019B' ASCBCD 003C' BODASC 0049' CLK 005A'

NEWCLK 0148' SETCLK 0162'

External reference(s):

AHOURS 0211' AMINTS 01F2' ASCNDS 0208' BHOURS 01B8'

BMINTS 00C4' BSCNDS 00CD' BUFFER 0000' CLKOTT 0000'

CD 013E' COLUMN 0000' FETCHC 00EE' INISIO 01C5'

INPTR 0000' LINE 0000' MOVCLR 0000' OUTPTR 0000'

SHOURS 0214' SMINTS 0000' SSCNDS 0000' STOPUR 015B'

STTXTO 0203' TABPTR 0200' TOLCAR 014F' TTAB1 01C8'

TTABST 0105' TTABZW 01FD' WCHDB 016A'

ANALOG: Indeholder programmel der starter AD
converteringen, samt henter og beregner
den målte værdi og omsætter det til
ASCII.

Module ANALOG of FORLIB REL

Length of Program 479

Length of Data area 204

Entry point(s):

CONV	0195'	EKANAL	00BB'	GETDAT	004A'	HEXBOD	0098'
KAN101	0002"	KAN102	000C"	KANPTR	0000"	RESULT	00C4"

External reference(s):

ASCBCD	0102'	BCDASC	0151'	BELL	0000	BLANK	0000
CMDR1	0000	CMDR2	0000	CMDR3	0000	CMDR4	0000
CO	00FA'	COLUMN	00E8'	CR	0000	EOI	0000
EDTAB	004B'	ESC	0000	ETX	0000	FALSE	0000
FETCHC	00ED'	INPTR	0175'	INT0	0000	INT1	0000
INT2	0000	INT3	0000	INT4	0000	INT5	0000
INT6	0000	INT7	0000	INTADR	0000	INTDIR	0000
INTENR	0000	INTMSK	0000	LF	0000	LINE	0185'
MODER	0000	MODCUR	0000	OUTPTR	0181'	PORT1	0000
PORT2	0000	RCBR	0000	SPECFR	0000	STATR	0000
STTIM1	0169'	STTXD	0172'	TABPTR	016F'	TIMER1	0000
TIMER2	0000	TIMER3	0000	TIMER4	0000	TIMER5	0000
TMBR	0000	TRDY	015E'	TRUE	0000		

LSAOV: Indeholder program der anvendes i for-
bindelse med LSA øvelse.

Module LSAOV of FORLIB REL

Length of Program 27

Length of Data area 0

Entry point(s):

LSAOV 0000'

TESTPI: Indeholder program der tester ROM, RAM
samt kommunikationen med skærmtermina-
len.

Module TESTPI of FORLIB REL

Length of Program 336
Length of Data area 0

Entry point(s):

TEST 0000' TESTBUM 014E'

External reference(s):

CMDR1	0000	CMDR2	0000	CMDR3	0000	CMDRP1	0000
INTADR	0000	INTDIR	0000	INTENR	0000	INTMRK	0000
MODER	0000	PORT1	0000	PORT2	0000	RCBR	0000
SPECFR	0000	STATR	0000	STATR	0000	STXTD	0000
TABP7R	0000	TIMER1	0000	TIMER2	0000	TIMER3	0000
TIMER4	0000	TIMER5	0000	TMR	0000		

----- Måleøvelser -----

Dette afsnit indeholder forslag til måleopgaver i tilknytning til de i kurset behandlede instrumenter. Antallet af opgaver til de forskellige instrumenter er større end det antal det kan forventes at kursisterne kan gennemføre, hvorfor det er instruktørens opgave at udvalge de øvelser der skal udføres. Ligeledes er instruktøren frit stillet med hensyn til at fremstille supplerende opgaver.

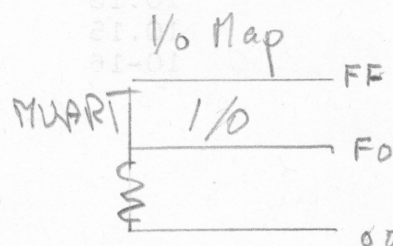
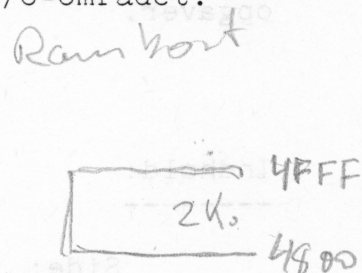
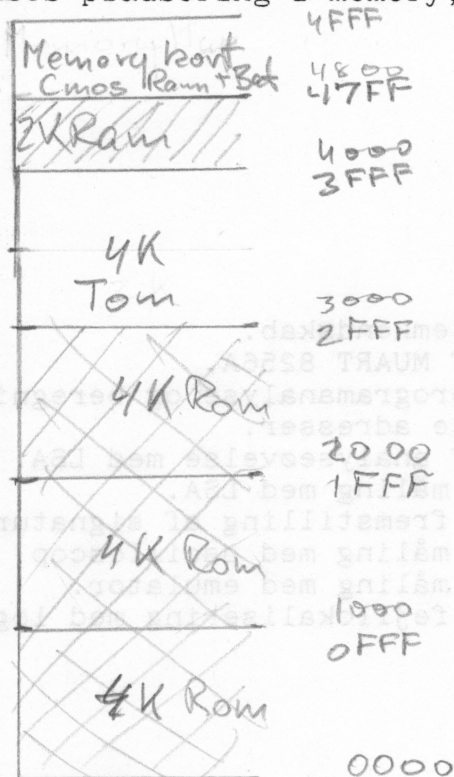
Indhold:

Side:	Emne:
10-2	PIV-1 systemkendskab.
10-3	Analyse af MUART 8256A.
10-5	Øvelse i programanalyse og beregning af absolute adresser.
10-9	Kontrol af analyseøvelse med LSA.
10-10	Øvelser i måling med LSA.
10-12	Opgaver i fremstilling af signaturer.
10.13	Opgaver i måling med oscilloscop.
10.15	Opgaver i måling med emulator.
10-16	Opgaver i fejllokalisering med logik-prober.

----- PIV-1, system- og kredsløbsanalyse -----

Opgave 1:

Foretag på grundlag af printkortene i PIV-1 samt den foreliggende dokumentation, en analyse der resulterer i et memory-map samt et I/O-map der viser de forskellige kredses pladsering i memory, henholdsvis I/O-området.



Opgave 2:

Find ud af hvor mange wait-states CPU-kortet og C-MOS RAM'en er indstillet til.

CPU-kort: |

C-MOS RAM: |

Opgave 3:

Hvor lang en ACCESS-tid må de EPROM'er have, der uden at ændre på de indstillede waite-states, kan køre i CPU-kortet.

Max access-tid:

575 n sek

----- Analyseøvelse af MUART 8256 -----

Øvelsens formål er at få kendskab til MUART'ens funktioner, og specifikt hvorledes den anvendes i PIV-1. Analysen skal foretages ved hjælp af soft-ware dokumentationen til PIV-1, i hvilke det kan læses hvilke bit-mønstre der overføres til de forskellige kontrolregistre i MUART'en, samt databogen for 8256. Det aktuelle program findes i modul AP8256.

Undersøg følgende:

-
- 1) Hvilken baud-rate indstilles der til?
 - 2) Arbejdes der med paritetsbit, og i bekræftende fald lige eller ulige paritet.
 - 3) Hvor mange stopbit udsendes der?
 - 4) Hvor mange databit udsendes der mellem start og stopbit?
 - 5) Hvilke ben er input og hvilke er sat som output i parallellportene?
 - 6) Hvilke clockfrekvens tæller timerne på?
 - 7) Er lysdioderne tændt efter initialisering af MUART'en?
 - 8) Hvad bruges de forskellige ind- og udgange på MUART'en til?
 - 9) Er der noget output fra INT-benet på MUART'en, og er der forbindelse mellem dette og CPU'ens INT-indgang?
 - 10) Kontroller at skærmterminalen er indstillet i overensstemmelse med MUART'en initialisering af serieporten (RS 232C).

Besvarelser:

Besvarelser fortsat:

----- Analysevejledning af MUART 8256 -----
Oversættelsen af programmet er at få kendskab til MUART'ens funktioner, og specielt hvorledes den anvendes i PIV-1. Analysen skal foretages ved hjælp af software dokumentation til PIV-1. I hvilke del kan læses hvilke bit-monstre der overføres til de forskellige kontrolregister i MUART'en, samt dataloggen for 8256. Det aktuelle program findes i modul AP8256.

----- Undersøg følgende: -----

- 1) Hvilken baud-rate indstilles der til?
- 2) Arbejdes der med paritetbit, og i hvilken retning? (altså lige eller ulige paritet).
- 3) Hvor mange stopbit udsendes der?
- 4) Hvor mange databit udsendes der mellem start og stopbit?
- 5) Hvilke den er input og hvilke er sat som output i parallelporten?
- 6) Hvilke clockfrekvens tæller timerne på?
- 7) Er lysdiodeerne tændt efter initialisering af MUART'en?
- 8) Hvad bruges de forskellige ind- og udgange på MUART'en til?
- 9) Er der noget output fra INT-benet på MUART'en, og er der forbindelse mellem dette og CPU'ens INT-indgang?
- 10) Kontrollet af akkumulatoren er indstillet i overensstemmelse med MUART'en initialisering af parallelporten (RS 232C).

----- Besvarelser: -----

EPROM output data ^{via read} styres af output enable
Multiplexede systemer kan give problemer
hvis ram brydes en for hurtige idet de
sender data ud mens adresser udsendes
af CPU

8256

8 interrupt

2 input porte

5 timer

1 serial VART RS 232 C med programmer
bar Baud rate generator

9 kontrol registre

Software initialisering AP 8256

Linie 370 - 400

Spørgsmål 10-3

Lysdecoder 9-8

1 Baud rate 2400 Command
register 2

2 den kører uden paritet
Bit 7 og 8 00

3 Command register 11 bit 4.5
der anvendes 2 stop bit

4 der er 8 bit

5 Port 1 ^{4 og 5 er input resten} er output ^{input}
Port 2 FF Port 2 output

6 Hukker klokfrekvens
1/3 af klokfrekvens
2 ^{41 Co} 01 command register
an 1 MHz 1,024
1 kHz eller 16 kHz her 1 kHz
7 ikke givet klokke
23 @ slukket

8 Port 2 udgang til printer
UART console
int til interrupt styring af CPU
se side 9,8

9 14 gennems 2 gates LS 240
total Buffer line driver 3state output

10 on off on 2400 Baud

----- Øvelse i programanalyse -----

Nedenstående programstykke stammer fra modul SERIO, underprogram VALG og varetager programvalg som følge af indtastning af en soft-key. Efter en indtastning returneres med en ASCII-karakter i accumulatoren, denne omsættes til HEX og multipliceres med 2. Derefter hentes det valgte programs startadresse fra tabellen ind i D- og E-registret, hvis indhold slutteligt skubbes på stacken. Det endelige hop til det valgte program sker med RET (return)!!

```

;
;===== OMSÆT TIL HEX OG *2 =====
;
        SUI      'O'
        ADD      A
;
;===== HENT PROGRAM ADRESSE =====
;
***** LXI      H,JTABLE
        MVI      D,O
        MOV      E,A
        DAD      D
        MOV      E,M      ; HENT LOW ORDER ADR.
        INX      H
        MOV      D,M      ; HENT HIGH ORDER ADR.
;
;===== EKSEKVER PROGRAM =====
;
        PUSH     D
        ffffffff RET
;
;===== SPRINGTABEL =====
;
JTABEL:  DW      O
        DW      MENUAE
        DW      MENUBE
        DW      KANALE
        DW      SATURE
        DW      MAALTE
        DW      SLUTE
;
;***** PROGRAM SLUT *****

```

Opgave:

Foretag en analyse af programstykket mellem markeringerne (* - f), således at der redegøres for signalerne på A-bussen og AD-bussen på tidspunkterne for ALE, RD og WE signalerne. Anvend skemaet på næste side til løsningen. For objectkoden henvises til assemblerlisten for PIV-1.

Beregning af absolutte adresser:

For at løse denne opgave er det nødvendigt at kende de absolutte adresser hvorpå programmet ligger. Disse kan ikke læses direkte i assemblerlisten, da denne kun giver den relative adresse inden for de enkelte moduler. Således starter alle modulerne på den relokerbare (flyttelige) adresse 0000'H.

De forskellige moduler får først deres endelige absolutte adresser når de er LINKET sammen til et sammenhængende program. Linkeren giver som et af sine output et LINK-MAP, der er en oversigt over hvilke absolutte adresser linkeren har "brugt" til de globale symboler. Et symbol vil kun være globalt hvis det er erklæret PUBLIC i et modul.

Hvis der netop findes et globalt symbol på det sted i programmet man søger den absolutte adresse for, kan denne findes direkte i LINK-MAPPEN. Hvis man ikke er så heldig (det er man næsten aldrig!) må man finde det nærmeste globale symbol, og derefter regne sig frem eller tilbage til det punkt man ønsker den absolutte adresse for.

Eksempel:

Den absolutte adresse på symbolet UDSKR i modul SERIO søges. Om UDSKR findes følgende oplysninger:

linienummer i assemblerliste:	1333
Relokerbar adresse	: 0199'H

Ved at sammenholde LINKER-MAPPEN og KRYDSREFERENCELISTEN for modul SERIO kan det konstateres, at det nærmeste symbol med en kendt absolut adresse er ASCFL:

linienummer i assemblerliste:	1637
Relokerbar adresse	: 025C'H
Absolut adresse	: 08F4H

Først beregnes forskellen mellem de to relokerbare adresser:

$$025C\ H - 0199\ H = 00C3\ H$$

Derefter beregnes UDSKR's absolutte adresse ved at trække den beregnede forskel fra ASCFL's absolutte adresse. Der skal trækkes fra fordi UDSKR ligger før ASCFL.:

$$08F4\ H - 00C3\ H = 0831\ H = UDSKR$$

Op start of logic analyser

ALE	foldende	flanke	Adresser
RD	stigende	flanke	DATA + Adr 8-15
WR	"	"	DATA
DATA 2000			
OP CODE	FETCH	OF	
OPERAND	Low	MR	
"	High	MR	
EXECUTE		MR	

Analyse af opgave 10-5

*/M

Data 4800 4A13 (531)
Program 0000 237E (9086)

ACPRS 1FBE	AHOURS 4905	AKANNR 492A
AMENNR 4937	AMINTS 4908	ASCBOD 1E5F
ASCFL 1865	ASCNDS 490B	ATOLL 4932
AVOLT 492D	BCDASC 1E6C	BELL 0007
BHOURS 4917	BLANK 0020	BMINTS 4918
BSCNDS 4919	BUFFER 4800	CHARIN 1653
CLK 1E7D	CLKOTT 153F	CLSCNT 100B
CMDR1 00F0	CMDR2 00F1	CMDR3 00F2
CMDRP1 00F4	CD 1673	COLUMN 4921
CONV 21D9	CR 000D	CUMDVT 1577
EKANAL 20FF	EOI 008B	EDTAB 0000
ERASEL 1531	ESC 001B	ETX 0003
FALSE 0000	FETCHC 1680	GETDAT 208E
HEXBOD 20DC	INISIO 1693	INITSK 1AFF
INPTR 4922	INT0 0001	INT1 0002
INT2 0004	INT3 000B	INT4 0010
INT5 0020	INT6 0040	INT7 0080
INTADR 00F6	INTDIR 00F6	INTENR 00F5
INTMSK 00F5	ISCNTT 14E1	KAN101 4949
KAN102 4953	KANALNR	4929 KANPTR 4947
LF 000A	LINE 4920	LSADV 2223
MENUT 1509	MODER 00F3	MOVOUR 16B2
NEWCLK 1F6B	NXTPTR 4900	OUTPTR 4924
PORT1 00F8	PORT2 00F9	PRFINT 1553
PRNLIN 155B	PRNRYT 154B	PRNTCH 1B32
PRSTT 154F	PSTART 1557	PTABPT 491E
PTBPTR 491C	RCBR 00F7	RESULT 4A0B
SETCLK 1F85	SHOURS 490E	SMINTS 4911
SPECFR 00FF	SSCND5 4914	START 1CF3
STATR 00FF	STDTIM 1DA7	STED 4926
STOPUR 493A	STPRNT 1BA8	STPTR 4800
STTIM1 1DB4	STTXTO 1709	TABPTR 491A
TCHOUT 1736	TCLCAR 152F	TCTRLV 1535
TESCAP 15D1	TEST 223E	TESTSUM 237C
TIM1 1DC9	TIMER1 00FA	TIMER2 00FB
TIMER3 00FC	TIMER4 00FD	TIMER5 00FE
TINVRE 153B	TMBR 00F7	TOFKAN 157D
TONKAN 1585	TRDY 4904	TRUE 00FF
TSFRA 1595	TSFRA2 1599	TSDFFA 15A3
TSDFTI 15AD	TSTART 14F9	TSTIL 159F
TSTOP 14F5	TTAB1 150D	TTABIO 152B
TTABST 1527	TTABZW 1523	TVOLT1 15B7
TVOLT2 15C3	TXTPTR 4902	VAL6 1746
WCHD6 1DF0		

33118 Bytes Free

*PIV/N:P

*/Y

*/E

Data 4800 4A13 (531)
Program 0000 237E (9086)

33118 Bytes Free

E0000 4A13 74A

Origin below loader memory, move anyway(Y or N)?N

	2 3 8 4
TEST PIV	2 2 3 E
LSA OV	2 2 2 3
ANALOG	2 0 4 4
UHR	1 E 2 3
TIMER	1 D A 7
PARIO	1 B 2 1
SERIO	1 6 4 5
AP 8256	1 5 D B
GLOBAL	1 0 0 0
MONITOR	

PROBLEM NO.		START ADDR		PROGRAM	
Joelse 10-5					

DESCRIPTION						DATE	SHEET	OF
LINE	ADDRS	DATA			SYMBOLIC INSTRUCTION OPCODE	OPERANDS	COMMENT	
		B0	B1	B2				
1	1763	21	6F	17	LXI H		J Table 176FH	
2	1766	16	00		MVI D		0 F D	
3	1768	5F			MOV EA		2 fre A - D	
4	1769	19			DAD D			
5	176A	5E			MOV EH		Heut low Adresse	
6	176B	23			INXH		Tal H op	
7	176C	56			MOV, M		Heut High Adresse	
8	176D	D5			PUSH D			
9	176E	C9			Ret			
10	176F	00					00	
11	1770	00					00	
12	1771	8A	17		178A		MENVAE	
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								

ASCII CHARACTER SET (7-BIT CODE)

MSD \ LSD		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	0000	NUL	DLE	SP	0	@	P	a	p
1	0001	SOH	DC1	!	1	A	Q	b	q
2	0010	STX	DC2	"	2	B	R	c	r
3	0011	ETX	DC3	#	3	C	S	d	s
4	0100	EOT	DC4	\$	4	D	T	e	t
5	0101	ENQ	NAK	%	5	E	U		u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(8	H	X	h	x
9	1001	HT	EM)	9	I	Y	i	y
A	1010	LF	SUB	.	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[k	{
C	1100	FF	FS	,	<	L	\	l	
D	1101	CR	GS	-	=	M]	m	}
E	1110	SO	RS	.	>	N	^	n	~
F	1111	SI	VS	/	?	O	_	o	DEL

NUL	— Null	DLE	— Data Link Escape
SOH	— Start of Heading	DC	— Device Control
STX	— Start of Text	NAK	— Negative Acknowledge
ETX	— End of Text	SYN	— Synchronous Idle
EOT	— End of Transmission	ETB	— End of Transmission Block
ENQ	— Enquiry	CAN	— Cancel
ACK	— Acknowledge	EM	— End of Medium
BEL	— Bell	SUB	— Substitute
BS	— Backspace	ESC	— Escape
HT	— Horizontal Tabulation	FS	— File Separator
LF	— Line Feed	GS	— Group Separator
VT	— Vertical Tabulation	RS	— Record Separator
FF	— Form Feed	US	— Unit Separator
CR	— Carriage Return	SP	— Space (Blank)
SO	— Shift Out	DEL	— Delete
SI	— Shift In		

Opgave 10 - 5

Sevco absolut adresse 1645
relativer adresse 0
relativer 011E
absolut adresse 1763

012A
011E

1645
012A

176F

1645
0145
178A

Stackpointer 4800

Stackpointer her 4800-2

----- Kontrol af analyseøvelse -----

Opgave:

Kontroller ved hjælp af logikanalysator PM3543 og PIV-1 at det programstykke der er analyseret i den foregående øvelse virkelig kører som du har fundet ud af.

Gennemførelse:

Forbind PM3543 til PIV-1, således at der både tilføres analysatoren RD- og WR-clock. Se nærmere om dette i afsnit 3. Foretag en indspilning og lav eventuelt en udprintning af listen, både parallelt og sequentielt formateret.

Sammenlign de opsamlede lister med analyseøvelsen, og undersøg og forklar eventuelle forskelle.

Notater:

----- Øvelser i måling med logkanalysator -----

Øvelsesprogram:

Program eksemplet fra side 3.11 er indeholdt i PIV-1's programhukommelse. I PIV-1's programdokumentation findes dette programstykke under symbolet "LASOV". Beregn, ved hjælp af oplysningerne fra LINKER-MAPPEN samt "LSAOV's" relative adresse, den absolutte adresse for øvelsesprogrammet.

Start af øvelsesprogrammet:

Øvelsesprogrammet startes via RST 6,5. For at få programmet kørt en gang, startes med at resette PIV-1 hvorved RST 6,5 enables, hvorefter vipekontakten "LSAØV" trykkes kortvarigt ned.

Opgave 1:

Afprøv herefter eksemplerne på de forskellige triggermodes fra side 3-12 til -20.

Opgave 2:

Foretag en opsamling af data på det tidspunkt hvor der bliver hentet en måleværdi fra A/D-converteren. Foretag nogle indspilninger med forskellige instillinger af indgangsspændingen til den aktive kanal i A/D-converteren. Det aktuelle sted i softwaren findes i MODUL ANALOG, underprogram GETDAT. PIV i 1 kanal voltmeter stilling

Opgave 3:

Foretag en dataopsamling der viser hvad der lægges ud på hukommelsespladsen HEXKAN ved indtastning af kanalnumret når PIV-1 er sat op til at være et etkanal voltmeter. Det aktuelle sted i softwaren findes i MODUL ANALOG, underprogram EKANAL.

Opgave 4:

Foretag en dataopsamling der udelukkende viser dataflowet til hukommelsespladsen RESULT på det tidspunkt hvor det endelige resultat af en måling er beregnet. Det aktuelle sted i softwaren findes i MODUL ANALOG, underprogram CONV og ADDITI.

* og fra A/D converter

Opgave 5:

Find ud af hvad der sker efter der er indtastet et antal space ved starten af monitoren i PIV-1. Monitoren i PIV-1 startes ved først at vippe knappen RST/MON opad, hvorved PIV-1 resettes og starter sit normale program som voltmeter. Derefter trykkes RST/MON ned, hvilket via RST 7,5 bringer PIV-1 til at køre monitorprogrammet, i starten af hvilket der skal indtastes nogle spaces. Disse bruges af monitoren til at måle BAUD-raten fra skærmterminalen.

Notater:

----- Signatur analyse -----

Opgave:

Optag et sæt signaturer for PIV-1, og fremstil den tilhørende dokumentation.

Omfang:

Signaturerne skal først og fremmest omfatte CPU-kortet, og derefter i den udstrækning det er muligt RAM-kortet og A/D-converteren i den nævnte rækkefølge.

Til at optage signaturer på CPU-kortet foreslås følgende testrutiner i HP's mikroprocessor exerciser:

- 1) Signaturer på ADO - AD7 og A8 - A15 optages med test 02 (ALE som clock).
- 2) Signaturer på chip-select logikken optages med test 01 (RD som clock).
- 3) Fællessignaturer på ROM'erne (en for hver) optages med test 54 (qualifyer = 0).
- 4) Signaturer for RAM-hukommelsens dataindhold optages med test 05. Under denne test disables CMOS RAM'en. Denne test vil give et sæt signaturer der er fælles for hele RAM'en på CPU-kortet. Denne test kan kombineres med bruges af en qualifyer, hvorved der kan optages et sæt signaturer for hver 1k RAM-gruppe.

Dokumentation:

Udform dokumentationen enten som en signaturliste eller noter signaturerne i et diagram over PIV-1.

Notater: *test 03 ram test, der kan tælle ok pladsen*

Signature Analyse

start

stop

clock

stel

veldefinerede og veldokumenteret

veldefineret ~~for~~ ALE RD

ved at tage +5V signature har
man kontrolere at start stop og
clock er rigtigt forbundet
signature af DV vil altid give 0000

vel control af computeren skal den
kunne sættes i free run eller
CPU sættes i søvne som ^{der gives} fig 2-4 ^{nop}
og kontrol af start og stop til A15

test 2, 1, 54, 5

A 0	0000
A 1	5555
A 2	CCCC
A 3	7F7E
A 4	5H21
A 5	0AFA
A 6	UPFH
A 7	52F8

Signatur

A0 - A15

test 01 09 02

A 8	HC89
A 9	2H70
A 10	HPP0
A 11	1293
A 12	HAP7
A 13	3C96
A 14	3827
A 15	755D

DB 0	F820
DB 1	49A0
DB 2	FP5F
DB 3	450V
DB 4	3645
DB 5	079H
DB 6	32FH
DB 7	HP18

optaget under Chip select
RAM DATA tilfølgeligt

Hold 1 Signatur på CPU

Hold 2 1 Signatur på Cmos Ram Kort
+ placeringss tegning + positions nr

BEN 40

P189

3072

Ram viser 4K med Cmos

2K uden Cmos

3K med 1 Cmos

4K efter udskiftning

----- Måling med oscilloscop på mikrocomputer -----

Målingerne gennemføres på PIV-1 ved hjælp af Philips logic scope PM3543.

Opgave 1:

Kontroller at de høje adressebit (A8 - A15) er stabile med hensyn til spænding, samt timet korrekt i forhold til RD, pin 32 på 8085.

Se datablad for 8085A angående timing af en read-cycle med waite-states.

Niveauerne skal kontrolleres både på CPU'en og i bagplanet. Begge steder skal spændingerne overholde kravene til TTL-logik.

Fremstil selv en målerapport, der foruden kurver med tid og spænding indeholder oplysninger om måleinstrument, målepunkter, samt hvorledes oscilloscopet blev trigget. Ligeledes oplyses om målingerne er sket med PIV-1's normale program, eller der er anvendt et specielt testprogram der i givet fald gives i rapporten.

Opgave 2:

Kontroller adresseudsendelsen på de lave adressebit (AD0 - AD7), m.h.t. niveauer og timing i forhold til ALE'en.

Se datablad for 8085A angående timing af en read-cycle med waite-states.

Niveauer og tider skal kontrolleres både på den interne bus med direkte forbindelse til CPU'en, samt i bagplanet.

Øvelse 3:

Kontroller tider og niveauer af data i en read-cycle, hvor der læses data fra en programhukommelse.

Den kritisk størrelse er data-setup tiden, det vil sige den tid data fra hukommelsen har været stabile inden den stigende flanke på RD.

Se datablad for 8085A angående timing af en read-cycle.

Niveauer og timing skal måles på CPU'ens ADO - AD7 ben i forhold til RD på CPU'en.

Gennemfør først øvelsen ved at trigge oscilloscopet externt fra PIV-1, idet i finder et triggersignal der giver et stabilt billede af en read-cycle i den nederste programhukommelse. Find eventuelt ud af hvor i programmet i er ved at "læse" adressen bit for bit og holde dette sammen med PIV-1's programdokumentation.

Gennemfør derefter den samme måling, men benyt ordtrigning af oscilloscopet.

Øvelse 4:

Kontroller niveauer og timing af data i en write-cycle, hvor CPU'en skriver data til RAM-hukommelsen på CPU-kortet.

De kritiske tider i en write-cycle er overlappet mellem CS og WR, der kaldes write-tiden, samt den tid der er stabile data inden afslutningen af write-tiden. Sidstnævnte tid benævnes data-setup tiden, og er for de anvendte RAM-kredse (Am 9124 BPC) 200 n sek. Write-tiden er opgivet til minimum 250 n sek.

I lighed med øvelse 4 gennemføres målingen først med extern trigning af oscilloscopet, og derefter med ordtrigning.

Øvelse 5:

Måling af A/D-converterens converteringtid. Se dokumentation for A/D-converter. Converteringen startes ved at CPU'en skriver til ADC'ens portnummer (ADC'en er I/O-mapped), idet den høje nippel indeholder basisadressen og den lave nippel kanalnummeret. De overførte data er uden betydning. Dette giver en "startpuls" på pin 25T på ADC-modulet. Når converteringen er færdig indikeres dette ved en puls på pin 27B der får flip-floppen J4 til at gå i stilling 1. Dette kan undersøges ved at føle på det mindst betydende bit i den lave byte fra ADC'en.

Mål tiden mellem startpulsen på pin 25T, og stoppulsen på J4, pin 15. Undersøg om konverteringstiden er afhængig af DC-spænding der skal konverteres.

spænding
Har ingen indflydelse på konverteringstiden
det varer ca 5 µs før EOC går high og
ca 32 µs før den går low

1,5 n/sec Adresse

10 = 13 - 1

12 n/sec

tca 160 n/sec

hold off. for at p0 startet
cget test program

320 n/sec 600 n/sec TAC 320 n/sec

2

TAL

TLL

TLA

$$0,5 \times \frac{3,5}{1,75} \times 10 \mu$$

175 n/sec

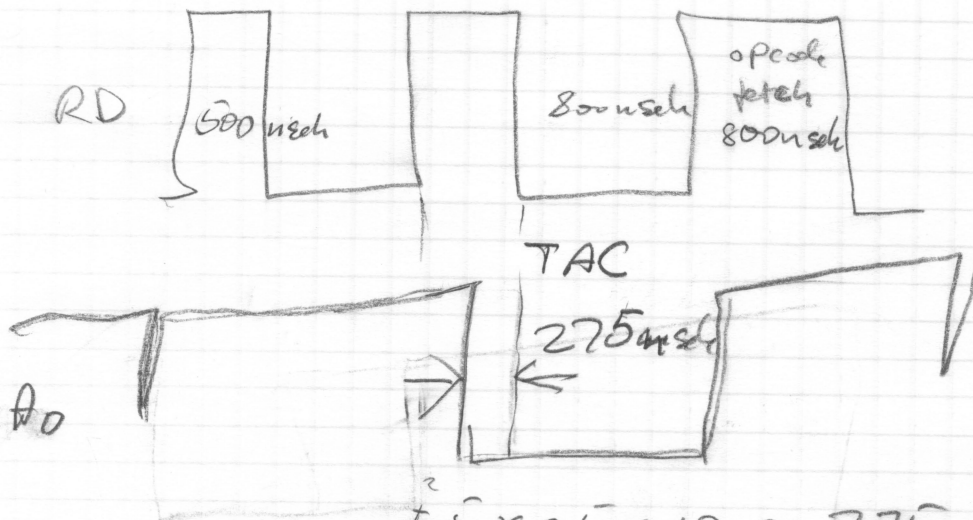
Push bndc of ALE

$$2,5 \times 0,5 \times 10$$

120 450 n/sec

120 n/sec

TALL



$$\frac{5,5}{27,5} \times 0,5 \times 10 = 275 \text{ nsec p0 Bagpover}$$

300 nsec p0 CPU

frigivning på chip selekt

frigivning på adresse

+ac +ca adresser på hukom-
melser mellem og read på CPU
for at afsløre om der er timings
problemer

Multiplexning af Dynamisk Ram
sker det ved udskeftning til hurtigere
ramme hvide at Ram når at udsende
data for den sidste adresse lednings
har modtaget de sidste adresser

Hvis man ikke kan se overgang mellem
data og adresse bus kan man pille
up or down med en belastnings
modstand på ca 1k for at se

chip select			p0
0000	1110	0001	0011
0	E	1	3
0001	0110	1000	0100
1	3	4	4

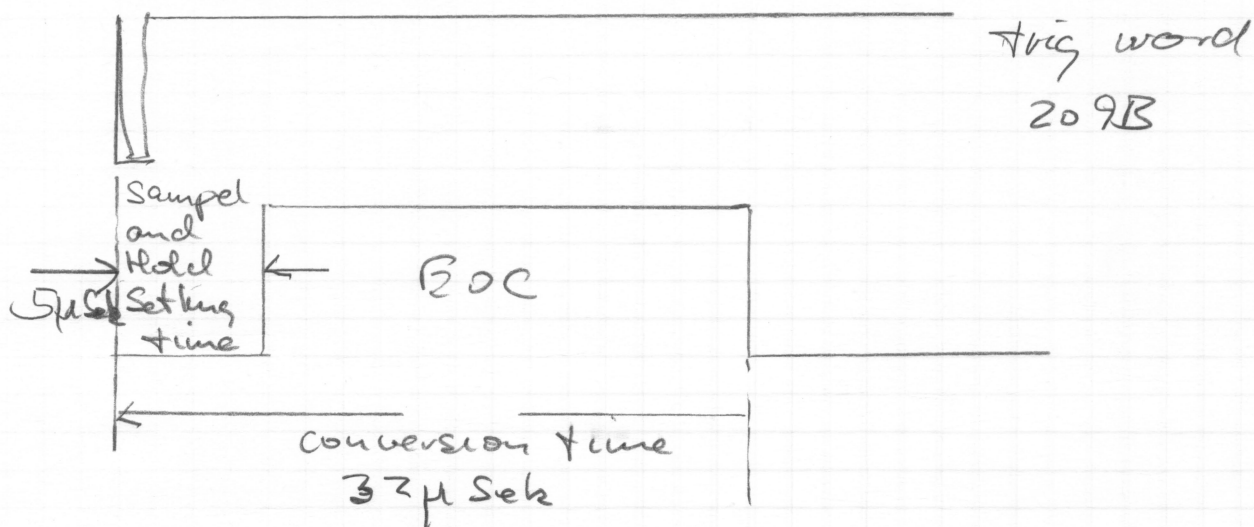
Monitor

Ram				F		servo
0100	1001	0010	0011	4922	1681	
4	9	2	3		1645	servo start
0001	0110	1000	0001			
1	6	8	1			
		1011	1100			
		B	C			
		D	3			

OP code

INPTR 49 22 low byte 22

stroke



converterings tid uafhængig af tilførselsspenning

SMP-E230-A1

Siemens

SMP 12,79

----- Måling med In-Circuit Emulator -----

- 1) Sluk for uSA'en og PIV-1. Fjern CPU'en fra PIV-1 og indsæt emulatorproben. Pas på at få den vendt rigtigt, pin 1 til pin 1.
- 2) Tænd for PIV-1.
- 3) Tænd for uSA'en og tryk på RESTART.
- 4) Start PIV-1 programmet ved at trykke "RUN" på uSA'en.
- 5) Mål clock-out frekvensen i PIV-1's bagplan.
- 6) Mål A/D-converteringstiden.
Restart adresse på modul GETDAT (adr. 004A')
Start på CS af A/D-converter.
Stop på klarmeling fra A/D-converter.
- 7) Prøvekør demo-testprogrammet der er installeret i PROM-fatningen nederst til højre på uSA'en. Gå frem efter den til programmet hørende manual.

----- Øvelser i fejlfinding med logikprober -----

----- B&O BEOCORD 9000 computer modul -----

1)

Der er en kortslutning mellem +5v Vcc og stel, find fejlstedet/fejlkomponenten.
(Elektrolytkondensator C55 er fjernet og erstattet med 2 stk wrapspyd).

2)

Der er en kortslutning mellem +14v Vcc og stel, find fejlstedet/fejlkomponenten.
(Elektrolytkondensator C57 er fjernet og erstattet med 2 stk wrapspyd).

--- BRUEL & KJÆR båndoptager type 7400, print ZD 0209 ---

1)

Det er konstateret at IOW, V32 pin 27 er kortsluttet til stel, men hvor??

2)

Det er konstateret at ADRESSEBUS 1 V38 pin 26 tilsyneladende er fast high. CPU'en er skiftet, men uden virkning. Der må altså være en kortslutning til 5v Vcc et eller endet sted, men hvor??

3)

Det ser ud til at DATABUS 0, V32 pin 13 konstant ligger på low, hvor er det galt?

4)

Det ser ud til at der er noget der trækker V13 pin 8 low. Hvad er der galt?. Printet kan evt. forsynes med 5v.

8085 Ready 2716
Run System 12prev normalt
FREQ = 0000 Hz jet
data probe p⁰ clock BVI Bagplon
Ben 44 CPU connect 3,073MHz
ALE 57,81 kHz

004A 208F

INITIAL OPTS 01880

Addr 208E Time 127 sek
time 3179 μ sek

4000 4FFF RA

000 0FFF ROM ?

F000 F7FF diagnose
Prom

Here ROM like er fyllet op standarder
Rom test ok for AO til AIS

20AA adresse for Break
point for control of HL

3210
1100

B02F HL
1101|0000|0010
59 D-

2218
3,437V
1,75 V

-- BRUEL & KJÆR måleforstærker type 2636 print ZD 0241 --

1)

Det er konstateret at DBUS3/G26b konstant ligger high. Lokaliser fejlstedet!

2)

Det er konstateret at INT/G19a konstant er low. lokaliser fejlstedet!

-- BRUEL & KJÆR måleforstærker type 2636 print ZD 0240 --

1)

Det er konstateret at PROG/D16a er holdt fast på low. Kontroller rigtigheden af dette med pulser og probe. Lokaliser fejlen!

2)

Det er konstateret at DBUS1/D2a og DBUS2/D1b tilsyneladende følges ad. Puls mellem dem og find ud af om de er kortsluttede. I bekræftende fald lokaliseres fejlen.

1b = Luftkarns Væser Ålborg

Lars = — m —

Jens-Ulrik = Viborg teknisk skole
Carsten = Københavns Brandvæsen
Erik = telefon
Niels = Post
Gunnar = Anker Data Ampliden
Finn = Brüel og Kjær
Bend = Grønlunds Televæsen
Harry = Viborg Teknisk skole
Bent = Olivette

CPU = 8085

Centronics

MUART =

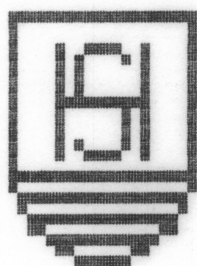
Serial RS232 Paralell + timer
interupt til CPU

Vi stoppes med rest startes på 4 på
menu

Computer til interface

Hebern interface til Hørslev

Kabel " til Printer



HANDVÆRKERSKOLEN
SØNDERBORG TEKNISKE SKOLE
SKOVVEJ 26
6400 SØNDERBORG

FEJLRAPPORT

Kanalnr	Reference	Målt værdi	Tidspunkt
01	3.75	3.08	11 : 11 : 55

Strapninger p3 CPU Kort
 oplysninger fra Manual ang memory
 og IO map

A 1+3 1 ventecyklus

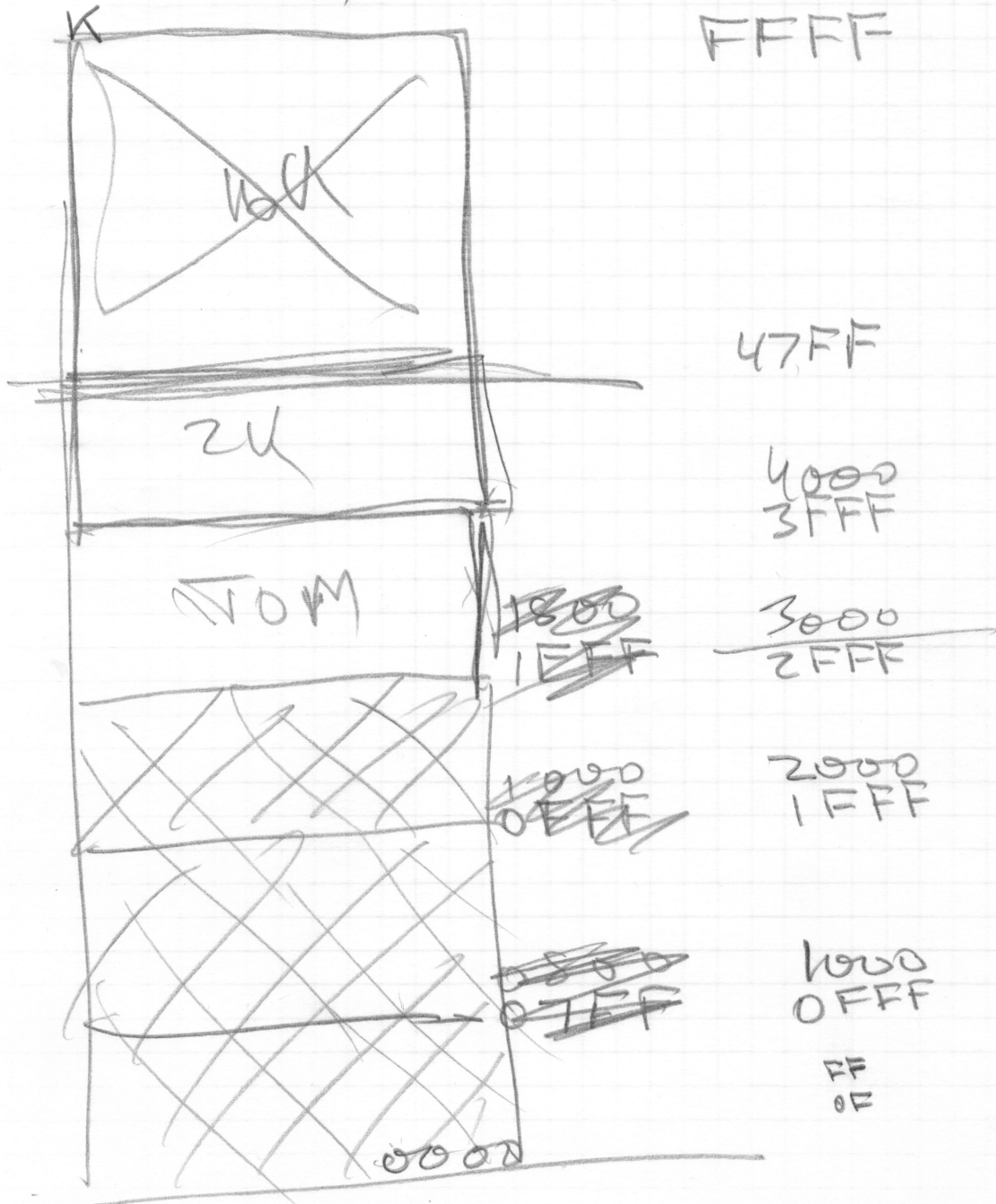
B Tom

D 2+3 1 ventecyklus for Lager + 1/0

E 2+3, 4+5, 6+7, 8+9 4 kilobyte
 adresse dekoder

H 1+2, 4+5 Fatning 2 og 3

F 1+2, 4+5 2732 (4Kx8)



Alle tekststränge avsluttes med 03
Alle tekstlabeler " " 0000

P Seg Data Segment Variablene & buffere

DS Define space

L80/P: XXXX / D: XXXX Global, AP8256,
CSEG DSEG

XXXX I Public

XXXX * Extern

Defineres med dette linie nummer

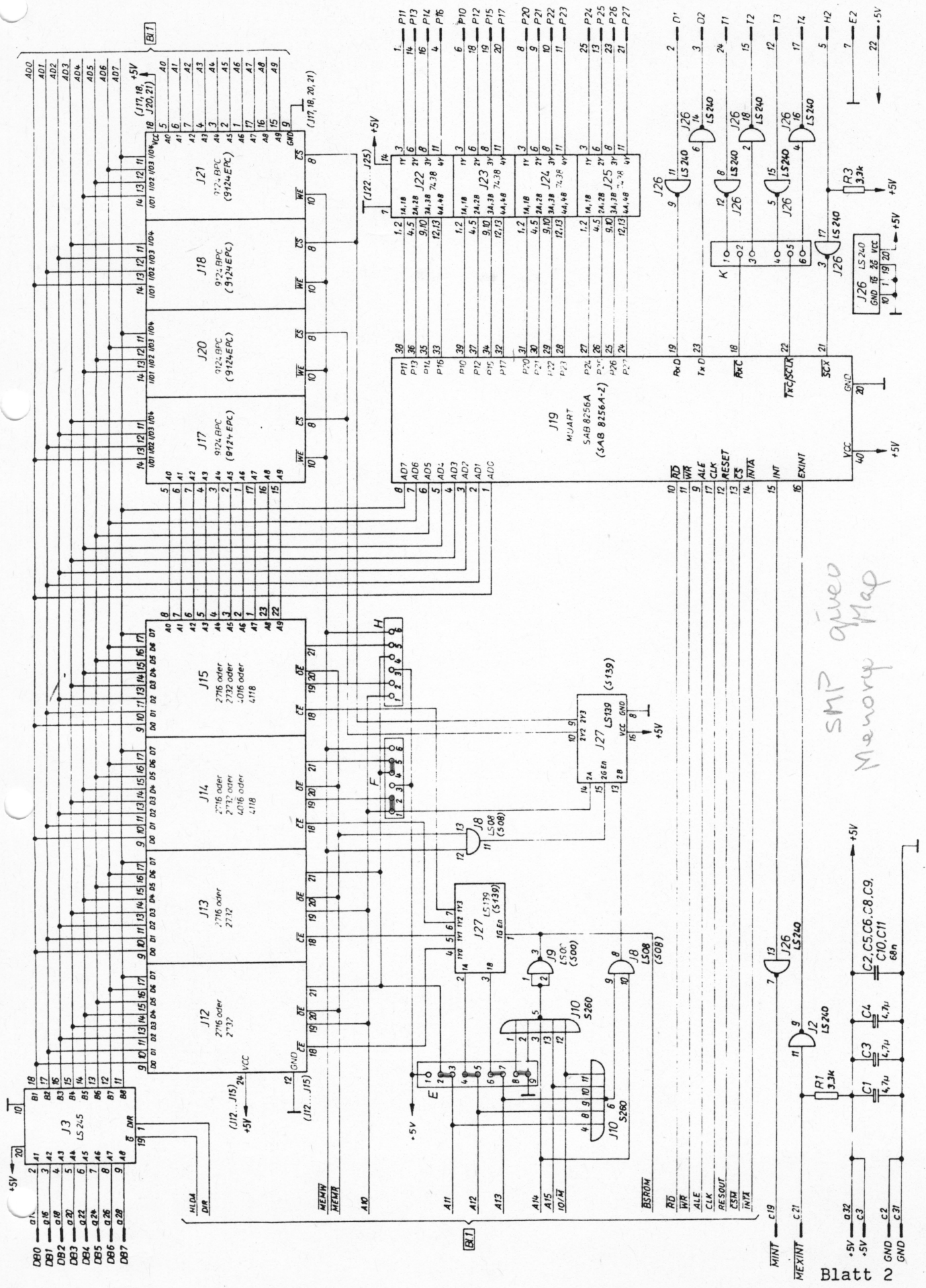
i krydsreference listen der står foran navnet

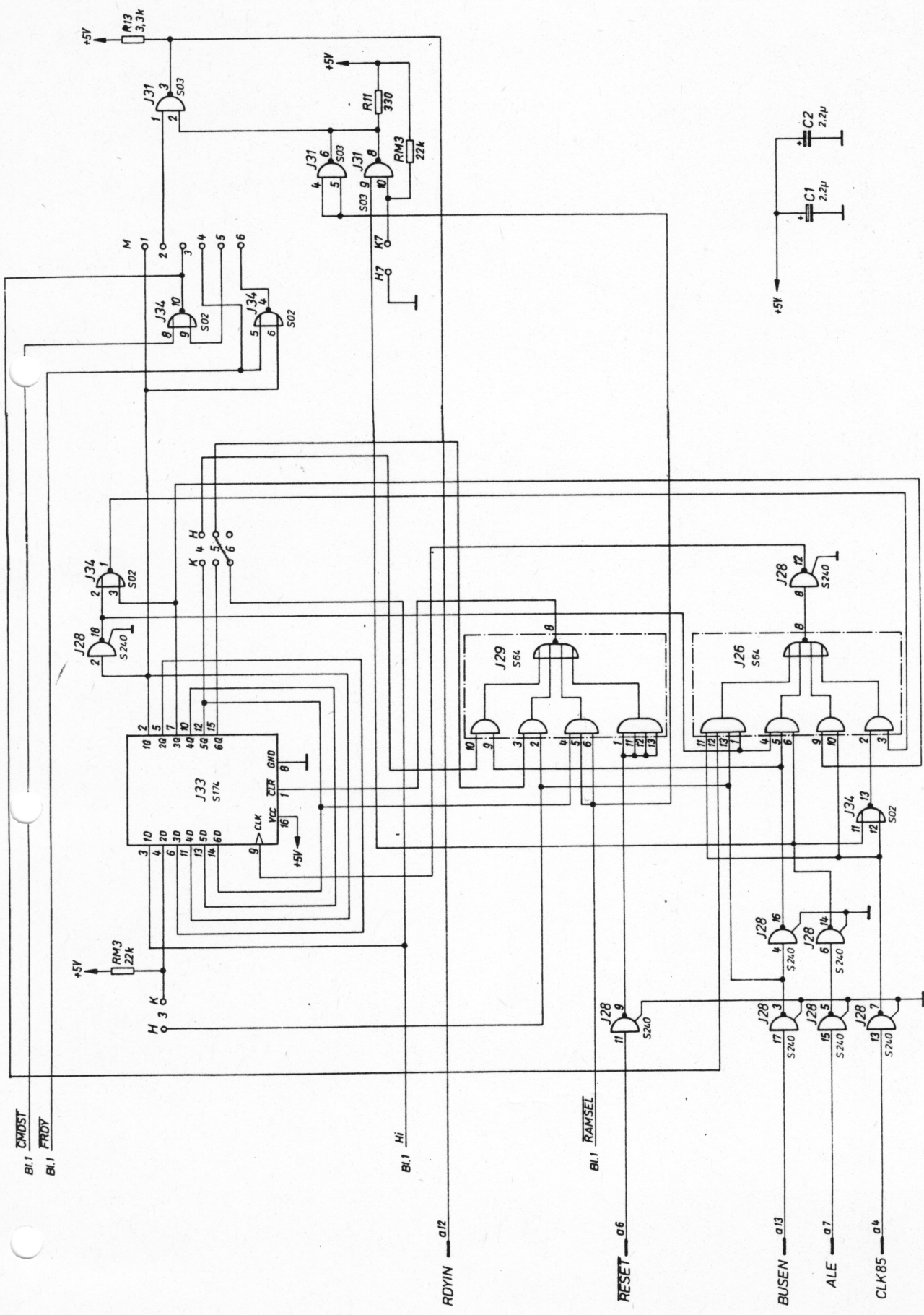
EXTERN 2 STPTR 1DC9

Opstart Test reset eller Restart 0

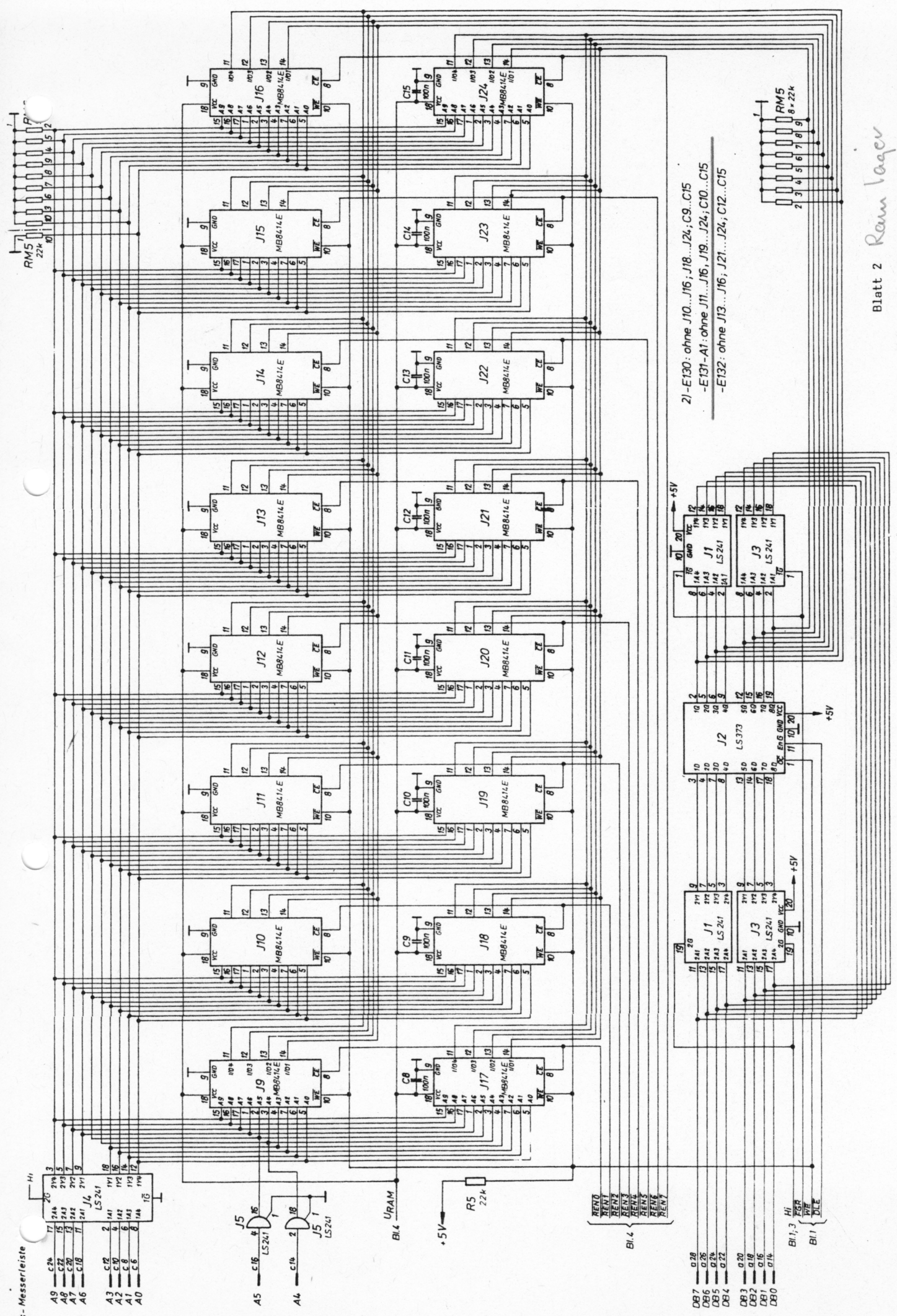
Linie 232

Linies lower symbol listet med alle
publike definerede adresser som ab-
solut adresser

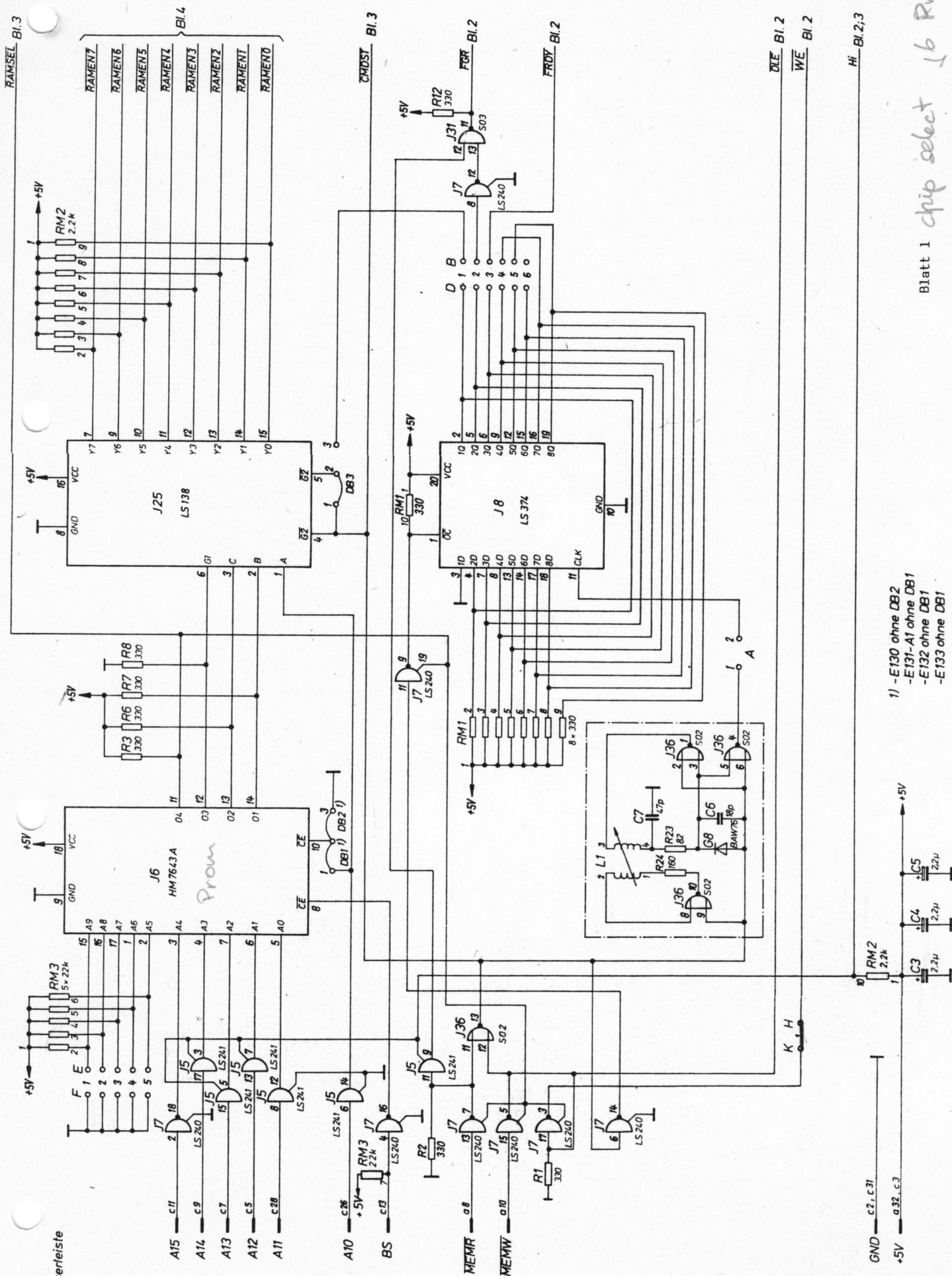




Blatt 3 wait state generator
for simhar CPU tel
Ram ev klar



21-E130: ohne J10...J16, J18...J24, C9...C15
 -E131-A1: ohne J11...J16, J19...J24, C10...C15
 -E132: ohne J13...J16, J21...J24, C12...C15



- 1) -E130 ohne DB2
- E131-A1 ohne DB1
- E132 ohne DB1
- E133 ohne DB1

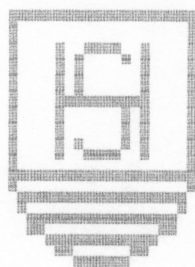
Blatt 1 chip select 16 Brown

Fejt II

Ved reset havde vi 07 RCL
 som data da vi ved at det skal
 være et JUMP undersøgte vi om
 der var flere PROM der blev selekteret
 samtidig Ben 18 på J13 10 konstant
 på nul Hvilket vil sige at J13
 konstant er CHIP Selected så J13
 vil udsende data samtidig med J12
 JMP DB 15

Fejt II

Ingen måling på data skærm
 ingen puls for End of conversion
 på DB0 Ved undersøgelse med logik
 probe fik vi puls på EOC på DAC
 men ingen puls på indgang af J3
 ben 14 undersøgelse af flip flop J4
 ben 14 og 15 som er udgange viste at
 begge udgange var 0 undersøgelse med
 ohm meter viste at ben 15 var kort
 sluttet til stel



HÅNDVÆRKERSKOLEN
SØNDERBORG TEKNISKE SKOLE
SKOVVEJ 26
6400 SØNDERBORG

FEJLRAPPORT

Kanalnr	Reference	Målt værdi	Tidspunkt
03	2.00	3.67	12 : 49 : 22
05	1.00	2.35	12 : 49 : 22
11	0.??	0.82	12 : 49 : 22
03	2.00	3.68	12 : 49 : 22
05	1.00	2.35	12 : 49 : 27
11	0.??	0.83	12 : 49 : 27
03	2.00	3.68	12 : 49 : 27
05	1.00	2.36	12 : 49 : 27
11	0.??	0.82	12 : 49 : 27
03	2.00	3.68	12 : 49 : 27
05	1.00	2.35	12 : 49 : 27
11	0.??	0.82	12 : 49 : 32
01	2.55	1.74	12 : 49 : 32
03	2.00	3.62	12 : 49 : 32
05	1.00	2.35	12 : 49 : 32
11	0.??	0.82	12 : 49 : 32
01	2.55	1.73	12 : 49 : 32
03	2.00	3.62	12 : 49 : 32
05	1.00	2.36	12 : 49 : 37
11	0.??	0.82	12 : 49 : 37
01	2.55	1.02	12 : 49 : 37
03	2.00	3.61	12 : 49 : 37
05	1.00	2.36	12 : 49 : 37
11	0.??	0.82	12 : 49 : 37
01	2.55	1.63	12 : 49 : 37
03	2.00	3.62	12 : 49 : 42
05	1.00	2.35	12 : 49 : 42
11	0.??	0.82	12 : 49 : 42
01	2.55	1.59	12 : 49 : 42
03	2.00	3.62	12 : 49 : 42
05	1.00	2.35	12 : 49 : 42

DATA TRANSFER GROUP

ARITHMETIC AND LOGICAL GROUP

Move	Move (cont)	Move Immediate
MOV A,A 7F	MOV E,A 5F	MVI A, byte 3E
A,B 78	E,B 58	B, byte 06
A,C 79	E,C 59	C, byte 0E
A,D 7A	E,D 5A	D, byte 16
A,E 7B	E,E 5B	E, byte 1E
A,H 7C	E,H 5C	H, byte 26
A,L 7D	E,L 5D	L, byte 2E
A,M 7E	E,M 5E	M, byte 36
MOV B,A 47	MOV H,A 67	
B,B 40	H,B 60	
B,C 41	H,C 61	
B,D 42	H,D 62	
B,E 43	H,E 63	
B,H 44	H,H 64	
B,L 45	H,L 65	
B,M 46	H,M 66	
MOV C,A 4F	MOV L,A 6F	
C,B 48	L,B 68	
C,C 49	L,C 69	
C,D 4A	L,D 6A	
C,E 4B	L,E 6B	
C,H 4C	L,H 6C	
C,L 4D	L,L 6D	
C,M 4E	L,M 6E	
MOV D,A 57	MOV M,A 77	
D,B 50	M,B 70	
D,C 51	M,C 71	
D,D 52	M,D 72	
D,E 53	M,E 73	
D,H 54	M,H 74	
D,L 55	M,L 75	
D,M 56		

byte = constant, or logical/arithmetic expression that evaluates to an 8-bit data quantity. (Second byte of 2-byte instructions).

dble = constant, or logical/arithmetic expression that evaluates to a 16-bit data quantity. (Second and Third bytes of 3-byte instructions).

adr = 16-bit address (Second and Third bytes of 3-byte instructions).

* = all flags (C, Z, S, P, AC) affected.

** = all flags except CARRY affected; (exception: INX and DCX affect no flags).

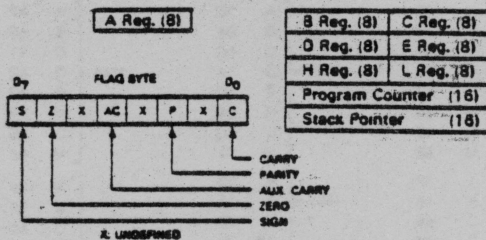
† = only CARRY affected.

Add*	Increment**	Logical*
ADD A 87	INR A 3C	ANA A A7
B 80	B 04	B A0
C 81	C 0C	C A1
D 82	D 14	D A2
E 83	E 1C	E A3
H 84	H 24	H A4
L 85	L 2C	L A5
M 86	M 34	M A6
ADC A 9F	INX B 03	A AF
B 98	D 13	B A8
C 99	H 23	C A9
D 9A	SP 33	D AA
E 9B		E AB
H 9C		H AC
L 9D		L AD
M 9E		M AE
Subtract*	Decrement**	
SUB A 97	DCR A 3D	
B 90	B 05	
C 91	C 0D	
D 92	D 15	
E 93	E 1D	
H 94	H 25	
L 95	L 2D	
M 96	M 3D	
SBB A 9F	DCX B 0B	
B 98	D 1B	
C 99	H 2B	
D 9A	SP 3B	
E 9B		
H 9C		
L 9D		
M 9E		
	Specials	
	DAA* 27	
	CMA 2F	
	STC† 37	
	CMC† 3F	
	Arith & Logical Immediate	
	ADI byte C6	
	ACI byte CE	
	SUI byte D6	
	SBI byte DE	
	ANI byte E5	
	XRI byte EE	
	ORI byte F6	
	CPI byte FE	

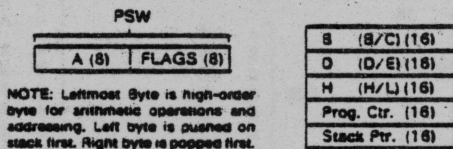
00 NOP	28 DCX H	56 MOV D,M
01 LXI B,dble	2C INR L	57 MOV D,A
02 STAX B	2D DCR L	58 MOV E,B
03 INX B	2E MVI L,byte	59 MOV E,C
04 INR B	2F CMA	5A MOV E,D
05 DCR B	30 SIM*	5B MOV E,E
06 MVI B,byte	31 LXI SP,dble	5C MOV E,H
07 RLC	32 STA adr	5D MOV E,L
08 ...	33 INX SP	5E MOV E,M
09 DAD B	34 INR M	5F MOV E,A
0A LDAX B	35 DCR M	60 MOV H,B
0B DCX B	36 MVI M,byte	61 MOV H,C
0C INR C	37 STC	62 MOV H,D
0D DCR C	38 ...	63 MOV H,E
0E MVI C,byte	39 DAD SP	64 MOV H,H
0F RRC	3A LDA adr	65 MOV H,L
10 ...	3B DCX SP	66 MOV H,M
11 LXI D,dble	3C INR A	67 MOV H,A
12 STAX D	3D DCR A	68 MOV L,B
13 INX D	3E MVI A,byte	69 MOV L,C
14 INR D	3F CMC	6A MOV L,D
15 DCR D	40 MOV B,B	6B MOV L,E
16 MVI D,byte	41 MOV B,C	6C MOV L,H
17 RAL	42 MOV B,D	6D MOV L,L
18 ...	43 MOV B,E	6E MOV L,M
19 DAD D	44 MOV B,H	6F MOV L,A
1A LDAX D	45 MOV B,L	70 MOV M,B
1B DCX D	46 MOV B,M	71 MOV M,C
1C INR E	47 MOV B,A	72 MOV M,D
1D DCR E	48 MOV C,B	73 MOV M,E
1E MVI E,byte	49 MOV C,C	74 MOV M,H
1F RAR	4A MOV C,D	75 MOV M,L
20 RIM*	4B MOV C,E	76 HLT
21 LXI H,dble	4C MOV C,H	77 MOV M,A
22 SHLD adr	4D MOV C,L	78 MOV A,B
23 INX H	4E MOV C,M	79 MOV A,C
24 INR H	4F MOV C,A	7A MOV A,D
25 DCR H	50 MOV C,B	7B MOV A,E
26 MVI H,byte	51 MOV C,C	7C MOV A,H
27 DAA	52 MOV D,D	7D MOV A,L
28 ...	53 MOV D,E	7E MOV A,M
29 DAD H	54 MOV D,H	7F MOV A,A
2A LHLD adr	55 MOV D,L	80 ADD B

81 ADD C	AC XRA H	07 RST 2
82 ADD D	AD XRA L	08 RC
83 ADD E	AE XRA M	09 ...
84 ADD H	AF XRA A	0A JC adr
85 ADD L	80 ORA B	0B IN byte
86 ADD M	81 ORA C	0C CC adr
87 ADD A	82 ORA D	0D ...
88 ADC B	83 ORA E	0E SBI byte
89 ADC C	84 ORA H	0F RST 3
8A ADC D	85 ORA L	08 RPO
8B ADC E	86 ORA M	E1 POP H
8C ADC H	87 ORA A	E2 JPO adr
8D ADC L	88 CMP B	E3 XTHL
8E ADC M	89 CMP C	E4 CPO adr
8F ADC A	8A CMP D	E5 PUSH H
90 SUB B	8B CMP E	E6 ANI byte
91 SUB C	8C CMP H	E7 RST 4
92 SUB D	8D CMP L	E8 RPE
93 SUB E	8E CMP M	E9 PCHL
94 SUB H	8F CMP A	EA JPE adr
95 SUB L	C0 RNZ	EB XCHG
96 SUB M	C1 POP B	EC CPE adr
97 SUB A	C2 JNZ adr	ED ...
98 SBB B	C3 JMP adr	EE XRI byte
99 SBB C	C4 CNZ adr	EF RST 5
9A SBB D	C5 PUSH B	F0 RP
9B SBB E	C6 ADI byte	F1 POP PSW
9C SBB H	C7 RST 0	F2 JP adr
9D SBB L	C8 RZ	F3 DI
9E SBB M	C9 RET	F4 CP adr
9F SBB A	CA JZ adr	F5 PUSH PSW
A0 ANA B	C8 ...	F6 ORI byte
A1 ANA C	CC CZ adr	F7 RST 6
A2 ANA D	CD CALL adr	F8 RM
A3 ANA E	CE ACI byte	F9 SPHL
A4 ANA H	CF RST 1	FA JM adr
A5 ANA L	D0 RNC	FB EI
A6 ANA M	D1 POP D	FC CM adr
A7 ANA A	D2 JNC adr	FD ...
A8 XRA B	D3 OUT byte	FE CPI byte
A9 XRA C	D4 CNC adr	FF RST 7
AA XRA D	D5 PUSH D	
AB XRA E	D6 SUI byte	

INTERNAL REGISTER ORGANIZATION



REGISTER-PAIR ORGANIZATION



REGISTER PAIR AND STACK OPERATIONS

	PSW (A/F)	Register Pair				SP	PC	Function
		B (B/C)	D (D/E)	H (H/L)				
INX		03	13	23	33			Increment Register Pair
DCX		0B	1B	2B	3B			Decrement Register Pair
LDAX		0A	1A	7E(1)				Load A Indirect (Reg. Pair holds Adrs)
STAX		02	12	77(2)				Store A Indirect (Reg. Pair holds Adrs)
LHLD				2A				Load H/L Direct (Bytes 2 and 3 hold Adrs)
SHLD				22				Store H/L Direct (Bytes 2 and 3 hold Adrs)
LXI		01	11	21	31	C3(3)		Load Reg. Pair Immediate (Bytes 2 and 3 hold immediate data)
PCHL						E9		Load PC with H/L (Branch to Adrs in H/L)
XCHG								Exchange Reg. Pairs D/E and H/L
DAD		09	19	29	39			Add Reg. Pair to H/L
PUSH	F5	C5	D5	E5				Push Reg. Pair on Stack
POP	F1	C1	D1	E1				Pop Reg. Pair off Stack
XTHL				E3				Exchange H/L with Top of Stack
SPHL					F9			Load SP with H/L

Notes: 1. This is MOV A,M. 2. This is MOV M,A. 3. This is JMP.

Flag Condition	Jump	Call	Return
Zero=True	JZ CA	CZ CC	RZ C8
Zero=False	JNZ C2	CNZ C4	RNZ C0
Carry=True	JC DA	CC DC	RC D8
Carry=False	JNC O2	CNC O4	RNC D0
Sign=Positive	JP F2	CP F4	RP F0
Sign=Negative	JM FA	CM FC	RM F8
Parity=Even	JPE EA	CPE EC	RPE E8
Parity=Odd	JPO E2	CPO E4	RPO E0
Unconditional	JMP C3	CALL CD	RET C9

ACCUMULATOR OPERATIONS

	Code	Function
XRA A	AF	Clear A and Clear Carry
ORA A	87	Clear Carry
CMC	3F	Complement Carry
CMA	2F	Complement Accumulator
STC	37	Set Carry
RLC	07	Rotate Left
RRC	0F	Rotate Right
RAL	17	Rotate Left Thru Carry
RAR	1F	Rotate Right Thru Carry
DAA	27	Decimal Adjust Accum.

BRANCH CONTROL GROUP

Jump	
JMP adr	C3
JNZ adr	C2
JZ adr	CA
JNC adr	O2
JC adr	DA
JPO adr	E2
JPE adr	EA
JP adr	F2
JM adr	FA
PCHL	E9
Call	
CALL adr	CD
CNZ adr	C4
CZ adr	CC
CNC adr	D4
CC adr	DC
CPO adr	E4
CPE adr	EC
CP adr	F4
CM adr	FC
Return	
RET	C9
RNZ	C0
RZ	C8
RNC	D0
RC	D8
RPO	E0
RPE	E8
RP	F0
RM	F8

I/O AND MACHINE CONTROL

Stack Ops	
PUSH	B C5 D D5 H E5 PSW F5
POP	B C1 D D1 H E1 PSW* F1
XTHL	E3
SPHL	F9
Input/Output	
OUT byte	D3
IN byte	D8
Control	
DI	F3
EI	F8
NOP	00
HLT	76
New Instructions (8085 Only)	
RIM	20
SIM	30

ASSEMBLER REFERENCE (Cont.)

Pseudo Instruction	
Generate:	
ORG	
END	
EQU	
SET	
DS	
DB	
DW	
Macros	
MACRO	
ENDM	
LOCAL	
REPT	
IRP	
IRPC	
EXITM	
Relocations:	
ASEG	NAME
DSEG	STKLN
CSEG	STACK
PUBLIC	MEMORY
EXTRN	
Conditional Assembly:	
IF	
ELSE	
ENDIF	

Constant Definition

0BDH	Hex
1AH	
105D	Decimal
105	
720	Octal
72Q	
11011B	Binary
00110B	
'TEST'	ASCII
'A'B'	

ASSEMBLER REFERENCE

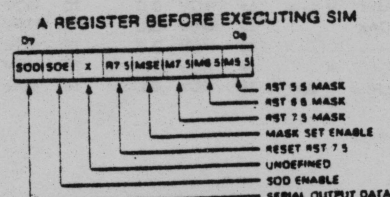
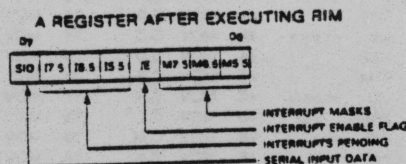
Operators	
()	
NUL	
LOW, HIGH	
MOD, SHL, SHR	
+	
-	
NOT	
AND	
OR, XOR	

RESTART TABLE

Name	Code	Restart Address
RST 0	C7	0000 ₁₆
RST 1	CF	0008 ₁₆
RST 2	07	0010 ₁₆
RST 3	0F	0018 ₁₆
RST 4	E7	0020 ₁₆
TRAP	Hardware*	0024 ₁₆
RST 5	EF	0028 ₁₆
RST 5.5	Hardware*	002C ₁₆
RST 6	F7	0030 ₁₆
RST 6.5	Hardware*	0034 ₁₆
RST 7	FF	0038 ₁₆
RST 7.5	Hardware*	003C ₁₆

*NOTE: The hardware functions refer to the on-chip interrupt feature of the 8085 only

USE OF THE A REGISTER BY RIM AND SIM INSTRUCTIONS (8085 ONLY)



ved reset af ingen indskrift på
Konsol ved af test program
vi test led 2 og 3 til at lyse
ingen fejl Hvilket indikerede
fejl i indskriftede Minimum og den
viste ledet i Ram Memory
den viste ikke vores Ram kort for
4800 til 4FFF

Hvilket tyder på at der er noget galt
med adresseringen af ram kort
Med logisk probe indtastet RZ/O
ingen aktivitet test indgange på 125
ingen aktivitet på udgang af prom
Indgange på prom tristatet
Ber 19 på 15 var kort sluttet til del
og der var udgangen 241 242 243
og 244 tristatet

Fejl 4

Efter reset skrives til consol og sker
fyldes op med ... ASCII karakter E 7

$$\begin{array}{r} 1705 \\ 1645 \end{array} \div \text{eris}$$

C6

ved test med signalanalyzer gav
2 K Ram (Vi har 3 K for di vi har
fejlt med de øverste 1K) dette viser
at vores Ram på REN 0 ikke bliver selectet

J 9 og J 17 blev ikke selectet,
afbrydelse af forbindelse mellem \overline{CE} ben
8 på J 9 og J 17 har ikke forbindelse til
ben 7 J 30

Bøjle til +5V manglende

test 03 03980