MPU-laren

TIDSKRIFT FÖR PRIVATDATAKLUBBEN PD68



Nr 2 1988
ÅRGÅNG 11

INNEHÅLLSFÖRTECKNING

	Sida
Sekreteraren har ordet	1
Klubbens stadgar	2
Hayes modem kommandobeskrivning	5
Annonser	8
Beskrivning av danskdatorn (Per-Olof Ryeng)	10
Tangentbordsdator	54

Sekreteraren har ordet.

Hej alla tystlåtna medlemmar. Som ni märkt är klubbens aktiviteter mycket stillsamma sedan en tid. Detta kan med all sannolikhet skyllas på IBM-klonerna. Samtidigt kan man konstatera att pionjärtiden är över när det gäller att bygga datorer och annan elektronik.

Nåväl nu är sommaren definitivt slut. Det finns som vanligt ingen återvändo när det gäller årstider. Dessbättre finns det möjligheter när det gäller datorer. För att klubben inte skall ta slut som sommaren avser vi att inköpa en AT. På den sätter vi upp ett BBS modell Opus. På så sätt får vi tillgång till programbank och brevlåda.

Program för olika datorer kan lagras i AT:n som ASCII-filer. Detta resulterar då i att vi kan sända in eller hämta hem program såsom vi gjorde i begynnelsen.

UL

PD68:s STADGAR

1 KLUBBENS UPPGIFT

Klubbens ändamål är att ideellt verka för att sprida kunskap om datorer och datoranvändning i samhället.

2 MEDLEMSKAP

Medlemskap kan erhållas genom ansökan och erläggande av medlemsavgift.
Medlemskap upphör ej om avgift erläggs före mars månads utgång.
Hedersmedlemmar väljes av styrelsen.

3 STYRELSEN

Klubbens angelägenheter handhas av en styrelse som äger att handla å klubbens vägnar med de inskränkningar i dess befogenheter som förekomma i dessa stadgar.
Styrelsen består av fem ledamöter valda för en tid av två år.
Valet förrättas på ordinarie årsmöte. Första gången dessa stadgar är i kraft väljes tre ledamöter för två år och två ledamöter för ett år. För styrelsen väljs tre suppleanter.

4 STYRELSEVAL

Styrelsen utser inom sig ordförande, kassör, sekreterare, vice ordförande och vice sekreterare.

5 STYRELSEBESLUT

Vid styrelsens sammanträden skall protokoll föras över dess beslut. Styrelsen är beslutsför då mer än halva antalet ledamöter är närvarande. Såsom styrelsens beslut gäller den mening, varom de flesta röstande förena sig. Vid lika röstetal gäller den mening som biträds av ordföranden.

6 KLUBBENS TECKNARE

Styrelsen utser inom sig de två personer som i förening äger rätt att teckna klubbens ekonomiska transaktioner.

7 STYRELSENS HEMVIST

Styrelsen har sin hemvist i Stockholm.

8 KLUBBENS EKONOMI

Styrelsen tillser att betryggande försäkring finns för klubbens egendom och kassaförvaltning. Kontanta tillgångar placeras, i den mån de inte behövs för omedelbart förestående utbetalningar, på postgiro.

H

9 RÄKENSKAPSÅR

Kalenderåret är räkenskapsår. Styrelsen upprättar balansräkning, vinst- och förlusträkning samt överlämnar dessa handlingar jämte protokoll och räkenskaper till revisorerna senast den 31 januari påföljande år. Styrelsen skall föra förteckning över medlemmarna. Styrelsen skall vid ordinarie årsmöte avgiva berättelse för det gångna verksamhetsåret.

10 REVISION

För granskning av styrelsens förvaltning och räkenskaper utses på ordinarie årsmöte två revisorer och en revisorssuppleant. Mandatet gäller intill dess nästa ordinarie årsmöte hållits. Revisorerna skall senast tre veckor före ordinarie årsmöte till styrelsen överlämna revisionsberättelse.

11 ÅRSMÖTE

Ordinarie årsmöte hålles årligen senast under februari månad. Extra årsmöte hålles då styrelsen finner så lämpligt eller då minst en tredjedel av klubbens medlemmar skriftligen, med angivande av skäl, så påfordrar.

12 KALLELSE

Kallelse till ordinarie årsmöte skall ske senast två veckor och till extra möte senast en vecka före mötets avhållande.

13 ÅRSMÖTESFÖRHANDLINGAR

1 Val av ordförande på mötet.
2 Val av justeringsmän.
3 Fråga om kallelse skett i tid.
4 Anmälan om ej förhandsanmälda frågor samt beslut om dessa skall tas upp till behandling på mötet.
5 Styrelsens redovisningshandlingar.
6 Revisorernas berättelse.
7 Fastställande av balansräkning.
8 Fråga om ansvarsfrihet för avgående styrelsen.
9 Fråga om beslut i anledning av uppkommen vinst eller förlust enligt den fastställda balansräkningen.
10 Styrelsens förslag till inkomst- och utgiftsstat samt fastställande av årsavgift.
11 Val av:
   a) styrelseledamöter
   b) suppleanter
   c) revisorer
   d) revisorssuppleant
   e) valberedning
12 Inkomna motioner.
13 Övriga frågor som under mötet under punkt 4 ovan beslutat skall tas upp till behandling.
14 Avslutning

H

Intill dess ordförande på mötet blivit utsedd, förs ordet av den avgående styrelsens ordförande eller vice ordförande, eller vid förfall för båda dessa, den styrelsen därtill utser. Den avgående styrelsens sekreterare svarar för mötesprotokollet.

14 MEDLEMSÄRENDEN

Ärende som enskild medlem senast tre veckor före mötets avhållande anmält för behandling på mötet till styrelsen, skall styrelsen bereda och avge yttrande över före behandling på mötet. Ärende som ej förhandsanmälts skall tas upp till behandling på mötet där minst två tredjedelar av de närvarande röstberättigade medlemmarna så beslutar.

15 STADGEÄNDRING

Ändring av dessa stadgar skall, för att bli gällande, antagas av två på varandra följande klubbmöten varav minst ett ordinarie.

16 KLUBBENS UPPLÖSNING

Upplöses klubben skall, sedan verksamheten lagligen avvecklats, eventuellt överskott tillfalla en av styrelsen godkänd ideell verksamhet.

HAYES MODEM RS232
Hayes modems command structure.

////////////////////////// HAYES COMMAND STRUCTURE //////////////////////////

Overview of the commands:

DIALING
=======

D- Digits 0-9
    symbols @ and *
    commands: P (Pulse dialing)
              T (Touch Tone Dialing)
              , (Dialing pause, 2 seconds)
              ; (Return tot COMMAND MODE after dialing)
              R (Calling an originate only modem)

A/ Automatically reexecutes last command

ANSWERING
=========

A   Manually answers an incoming call

MODEM OPERATIONS
================

AT Beginning of a HAYES command
-C C0 Transmit carrier signal off
    C1 Transmit carrier signal on
-E E0 No command echo
    E1 Echo command
-F F0 Half duplex
    F1 Full duplex
-H H0 Modem on-hook
    H1 Modem off-hook
    H2 Special off-hook
-M M0 Modem speaker off
    M1 Speaker on until carrier
    M2 Speaker always on
-O Return to data mode
-Q Q0 Send modem responses
    Q1 Do not send modem responses
-Sr? Read value in register r
-Sr=n Set register r to value n
-V V0 single digit respons
    V1 Word responses
-X X0 Basic respons set
    X1 Extended respons set
    X2 Recognize no dial tone
    X3 Recognize busy signal
    X4 Recognize no dial tone and busy tone
-&G &G0 no guard tone
    &G1 550 Hz guard tone
    &G2 1800 Hz guard tone

REGISTER SETTING

DIALING

S6
Function: Wait for dial tone
Parameters: 2-255 seconds

S8
Function: Pause time for comma
Parameters 0-255 seconds

S11
Function: Duration and space of touche tone
Parameters: 50-255 milliseconds

ANSWERING

S0
Function: Number of rings before autoanswering
Parameters: 0-255 rings

S1
Function: Count number of incoming rings
Parameters: 0-255 rings

DIALING AND ANSWERING

S7
Function: Wait time for remote carrier after dialing or answering
Parameters: 1-255 seconds

S9
Function: Carrier detect respons time
Parameters: 1-255 (1/10 seconds)

S10
Function: Delay time between lose of carrier and hangup
Parameter: 1-255 (1/10 seconds)

CHARACTER DEFINITION

S2
Function: Value of escape character
Parameter: 0-127 (ASCII)

S3
Function: Value of carriage return character
Parameter: 0-127 (ASCII)

S4
Function: Value of line feed character
Parameter: 0-127 (ASCII)

S5
Function: Value of backspace character
Parameter: 0-32 or 127 (ASCII)

SPECIAL FUNCTIONS
==========

S12
Function: Guard time for escape characters
Parameters: 0-255 (1/50 seconds)

S16
Function: Put modem in self test mode
Parameters: 0-2

<<EOF>>

NEC 3,5"

Till salu.

2 st NEC floppy 80 spår.
Dubbelsidiga.

Med dessa drivar får du ut 720 kb på
en PC/XT/AT med ett nytt kontrollkort
som hanterar alla fyra diskformaten.

Pris: 1500:- för båda.

Ulf Holm
Tel: 08 - 713 98 74

ULF KNÖPPEL

Privatdataklubben PD 68
Box 1098

122 21 Enskede

Stockholm den 4 april 1988.

Hej på PD 68!

Kan en gammal medlem få en annons införd i tidningen - som radannons?

TILL SALU:

2 st SWTPC datorer varav 1 st MULTI USER + extra minnskort
2 st TEC-70 monitorer m tangentbord - grön text
1 st 8 tum dubbla skivminnen - enkelsidiga
1 st 8 tum dubbla skivminnen - dubbelsidiga, dubbel packning
1 st Winchester SWTPC felfri
1 st Winchester SWTPC defekt
1 st CENTRONIC printer
2 st modem för IBM skönskrivare
      manualer
      disketter 8 tum

Programvara som ingår: operativsystem FLEX + div. extra program
                        STYLUS ordbehandling
                        BERGS databas

Säljes komplett eller i delar. Fullständig uppsättning av manualer.

Ring eller skriv till Ulf Knöppel Studio
Armfeltsgatan 18
115 34 Stockholm

Tel. dagtid   08/ 664 00 17 eller 664 00 18
     kvällstid 08/ 664 00 18 eller 664 00 19
     telefax   08/ 664 00 29

INNEHÅLLSFÖRTECKNING
====================

1. SYSTEMBESKRIVNING

1.1 BUSSBESKRIVNING
1.2 KONTAKTBESKRIVNING
1.3 MINNESKARTA
1.3.1 MINNESADRESSERING > 64KB
1.3.2 RAMDISK
1.3.3 PROMDISK
1.4 AVVIKELSE FRÅN FLEX-ORGINAL
1.5 PRINTERSPOOLING

2. HÅRDVARUBESKRIVNING

2.1 PROCESSORKORT
2.1.1 DC4BUG
2.2 FLOPPY/CRT-KORT DC2
2.2.1 KONTROLLKARAKTÄRER OCH FUNKTIONSTANGENTER
2.2.2 TECKENPROM
2.3 EPROMKORT DC4
2.4 IO/EPROMBRÄNNARE DC5
2.5 RAM 1MB DC6
2.6 BAKPLAN
2.7 KRAFT
2.8 PC-TANGBORDSINTERFACE

3. SYSTEMBESKRIVNING MJUKVARA

3.1 PROMDISKKATALOG - STANDARD EPROM12

EPCUF - Enhanced Personal Computer Using Flex

1. SYSTEMBESKRIVNING

1.1 BUSSBESKRIVNING

EPCUF använder sig av en 6809E-processor. Det är en 8-bitars processor från MOTOROLA. 6809E har inte någon inbyggd oscillator för systemklockan. Systemklockan är en yttre oscillator som används både för processor och bildskärm. Den yttre oscillatorn gör det möjligt att synkronisera läs- och skrivoperationerna mot bildminnet och på så sätt undvika flimmer på skärmen.
EPCUF är en vidareutveckling av den traditionella "FLEX-datorn". Genom att lägga till funktioner som t.ex Ramdisk, Printerbuffert och Promdisk har systemet förbättrats avsevärt.
Processorbussen i ECPCUF är inte någon standardkonstruktion.

Bussen består av 4 delar adressbuss, databuss, kontrollbuss och kraftbuss. Bussens utseende och konstruktion bestäms av 6809E-processorn. EPCUF har en utökad minnesadressering. Adressbussen har därför utökats med 4 adressledningar(A16 - A19).
Systemklockans placering på DC2-kortet och valet av 6809E ökar antalet signaler i bakplanet.

Adressbuss	A0 - A19	fyra extra bitar A16 - A19
Databuss	D0 - D7	
Kontrollbuss	R-/W	R/W från processor
	/R	/R konstruerad signal PALDC1
	E	E klocka till processor
	/E	inverterad E klocka
	Q	Q klocka
	S-OFF	buffrad PB4 (ej använd)
	/BELL	Ljudsignal från DC2-kort
	CRT Q	Q klocka från DC2-kort
	/NMI	NMI avbrott
	/IRQ	IRQ avbrott
	/FIRQ	FIRQ avbrott
	/RES	Resetsignal
	/ME	Enable för minne $0000-DFFF
	/IO	Enable för IO $E000-E0FF
	/CRT	Enable för bildminne $E800-EFFF
Kraftbuss	+5V	Matning +5V
	+12V	Matning +12V
	-12V	Matning -12V
	0V	GND

1. SYSTEMBESKRIVNING

BUSS FÖR DCX-KORT

		A		C	
	GND	+	01	+	GND
	/NMI	+	02	+	/ME $0000-$DFFF
	/IRQ	+	03	+	/IO $E000-$E0FF
	/FIRQ	+	04	+	
	/RES	+	05	+	
		+	06	+	
		+	07	+	E
		+	08	+	Q
$E800-$EFFF	/CRTRAM	+	09	+	CRT Q
	A19	+	10	+	/E
	A18	+	11	+	S-OFF
	A17	+	12	+	/BELL
	A16	+	13	+	
		+	14	+	D0
		+	15	+	D1
		+	16	+	D2
		+	17	+	D3
		+	18	+	D4
		+	19	+	D5
		+	20	+	D6
		+	21	+	D7
	R-/W	+	22	+	/R
	+12V	+	23	+	-12V
	A15	+	24	+	A0
	A14	+	25	+	A1
	A10	+	26	+	A2
	A13	+	27	+	A3
	A11	+	28	+	A4
	A9	+	29	+	A5
	A8	+	30	+	A6
	A12	+	31	+	A7
		+	32	+	+5V
		A		C	

1. SYSTEMBESKRIVNING

1.2 KONTAKTBESKRIVNING

Till EPCUF kan följande periferiutrustningar anslutas. Vissa enheter behövs för att systemet ska fungerar.

1. floppydisk
2. bildskärm (monitor med composite video)
3. tangentbord seriellt 1200 baud
4. Centronic interface för printer

DC1-CPU KORT

S2 - ACIA (MC 6850)

PB3	10	9	
BOOT	8	7	
5V	6	5	TXD
GND	4	3	RXD
/NMI	2	1	

S3 - PRINTER (CENTRONICS)

ACKNLG	20	19	PB6
D7	18	17	PB0
D6	16	15	GND
D5	14	13	GND
D4	12	11	GND
D3	10	9	GND
D2	8	7	GND
D1	6	5	GND
D0	4	3	GND
STROBE	2	1	GND

DC2-FLOOPY/CRT KORT

S3 - FLOOPY

READY	34	33	GND
SS	32	31	GND
READDAT	30	29	GND
W-PROT	28	27	GND
TR00	26	25	GND
WG	24	23	GND
WD	22	21	GND
STEP	20	19	GND
DIR	18	17	GND
MON	16	15	GND
DS3	14	13	GND
DS2	12	11	GND
DS1	10	9	GND
INDEX	8	7	GND
DS0	6	5	GND
TG43	4	3	GND
SPARE	2	1	GND

S2 - VIDEO

VIDEO	1
GND	2
GND	3
LIGHTP	4

1. SYSTEMBESKRIVNING

DC5-I/O KORT (PROMBRÄNNARE)

S4 - VIA

	CA1	26	25	PB6
		24	23	PB0
		22	21	GND
		20	19	GND
	PA7	18	17	GND
	PA6	16	15	GND
	PA5	14	13	GND
	PA4	12	11	GND
	PA3	10	9	GND
	PA2	8	7	GND
	PA1	6	5	GND
	PA0	4	3	GND
	CA2	2	1	GND

S2 - ACIA (6551)

+12V	10	9	DCD
RTS	8	7	GND
CTS	6	5	TXD
DSR	4	3	RXD
RXC	2	1	DTR

S3 - ACIA (6551)

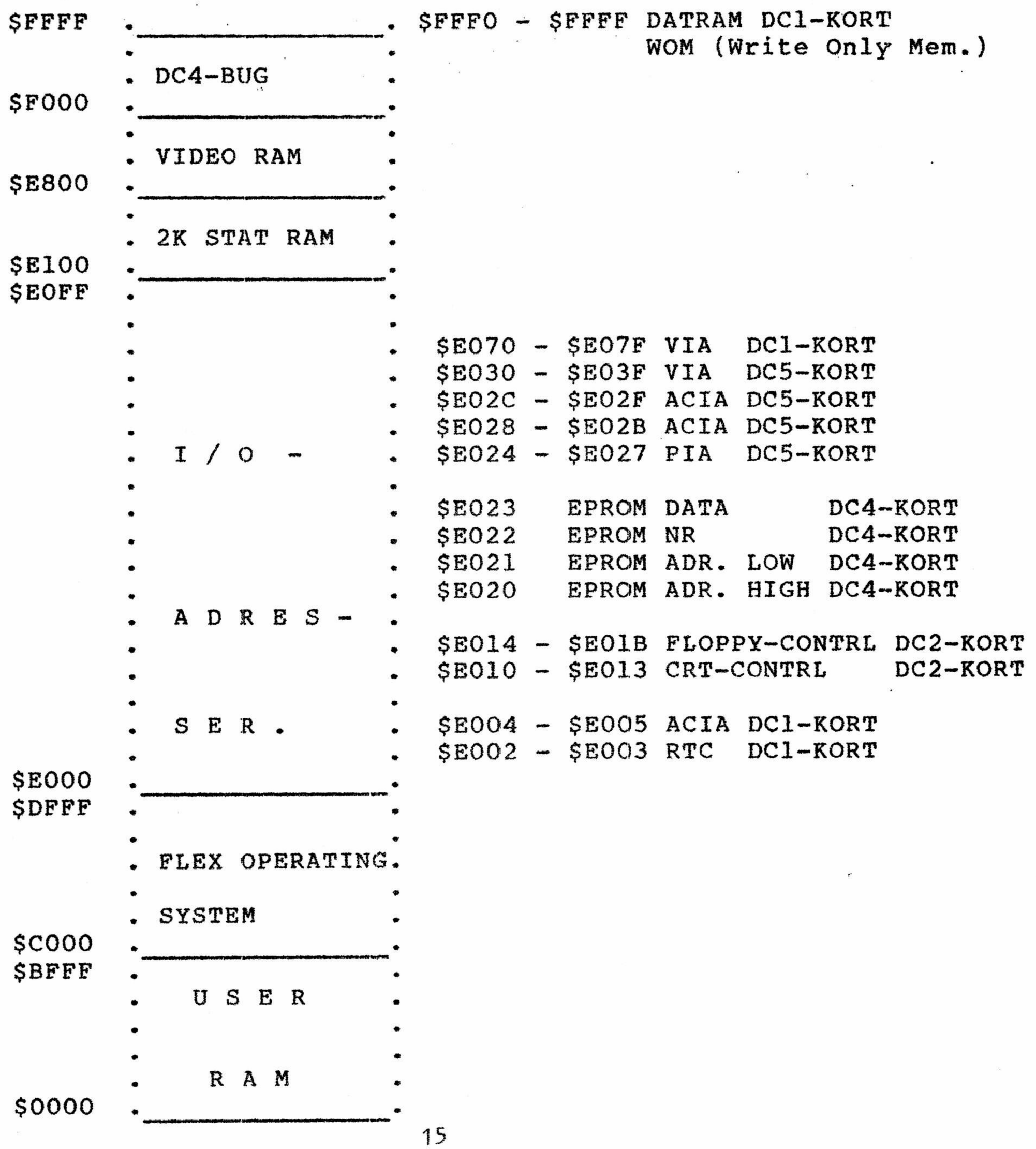
-12V	10	9	
CTS	8	7	GND
RTS	6	5	TXD
	4	3	RXD
	2	1	

Ansluts ett DC5-kort till bussen så ökas antalet tillgängliga interface. DC5-kortet är utrustat med 2 st serieportar och en parallellport.

1. SYSTEMBESKRIVNING

1.3 MINNESKARTA ÖVER DATORN

EPCUF:s 6809E-processor kan direktadressera 65536 minnespositioner. Denna adresseringsrymd delas av primärminne (RAM), I/O-organ samt systemprogram (ROM). I EPCUF är primärminnet utökat till 1MB. Det utökade minnet adresseras över ett DATRAM (Dynamic Address Translation RAM). Den minnesarea som styrs av DATRAM kan inte direktadresseras av processorn. Systemet utnyttjar denna minnesarea för printerbuffert, ramdisk och temporär datalagring.
Den ordinarie minnesdispositionen framgår av figuren nedan.



1. SYSTEMBESKRIVNING

1.3 MINNESKARTA

1.3.1 MINNESADRESSERING ÖVER 64KB

I punkt 1.3 visades minneskartan över systemet. 6809E-processorn kan adressera 65536 minnesceller. Adressen från processorn (logisk adress) översätts via DATRAM (Dynamic Address Translation RAM) till en fysisk adress mot det dynamiska minnet (1MB). Det innebär att systemet arbetar med 2st adressbussar en logisk och en fysisk adressbuss. Införandet av DATRAM (Dynamic Address Translation RAM) gör att processorn kan växla mellan olika minneskartor eller sidor. Den logiska adressen har området $0000 - $FFFF (64K) och den fysiska $00000 - $FFFFF (1MB). Det ska påpekas att den logiska adressen är begränsad till $DFFF (56KB). Det beror att I/O, 1K RAM, bildminne och DC4BUG delar utrymme på denna buss. Ovanstående begränsning får i praktiken ingen större nackdel. Det betyder den sida man kan "mappa" åt gången inte får vara större än 56K.Det blir dock fortfarande möjligt att nå hela minnet(1MB).
Den fysiska adressbussen byggs enligt figur nedan.

DATRAM (Dynamic Address Translation RAM)

1. SYSTEMBESKRIVNING

Den fysiska adressen består av 5st 4-bitars grupper (eng. nibbles). Adressledningarna A0 - A11 kommer direkt från processorn och passerar endast buffertar ut till bussen. Dessa adressledningar från processorn bildar ett 4K byte block. Resterande adressledningar (A12 - A19) kommer från ett DATRAM (Dynamic Address Translation RAM). DATRAM består fysiskt (på DC1-kort) av 2st bipolära RAM som kan lagra 16 nibble. Dessa båda minnen adresseras parallellt för man ska få en byte. Den fysiska adressen blir avhängig av informationen som finns lagrad i detta minne.

DATRAM's 16 minnesceller adresseras med 4 adressledningar (A0 - A3). Dessa adressledningar är kopplade till utgången av en 4-bitars multiplexor.
Funktionen är följande. Normalt när processorn arbetar och inte skriver mot DATRAM så är DATRAM's adresstrådar kopplade mot processorns mest signifikanta adressledningar A12 - A15. Information lagrad i DATRAM kommer att avläsas och skickas ut mot systembussen i takt med processorns adressering.
A12 - A15 kan anta värden från $0 - $F hexadecimalt. Antag att den logiska adressen har värdet 0000 0000 0000 0000 binärt. Det innebär att DATRAM's adress 0 utpekas. Innehållet i adress 0 kommer att skickas ut på adressledningarna A12 - A19. Antag också att innehållet i adress 0 (DATRAM) är $01. Den fysiska adressen mot dynamiska minnet på systembussen får då värdet.

fysisk adress 0000 0001 0000 0000 0000 20-bitar

obs den logiska adressen har värdet

0000 0000 0000 0000 16-bitar

Den fysiska adressen kan generellt skrivas:

DD XXX

DD byte från DATRAM
XXX 4K block i dynamiska minnet

Byten DD från DATRAM kan anta 256 kombinationer. Det är möjligt att adressera 256st 4K block. 256 x 4KB blir 1MB.

16st minnespositioner i DATRAM kan utpeka 16st 4K block i systemminnet. I praktiken kan bara 14st block (56KB) utnyttjas.
DATRAM's innehåll har ovan beskrivits att vara på varandra följande dvs blocken tillsammans bildar en kontinuerlig area. Det är inget krav. Ordningen mellan blocken kan vara helt godtycklig. Ett 4K block kan bytas ut i den vanliga minneskartan genom att manipulera innehållet i DATRAM. Det är möjligt att t.ex temporärt spara undan ett 4K block och sedan ta tillbaka det. Växlingen mellan olika minnesareor blir också mycket snabb.

1. SYSTEMBESKRIVNING

Hur adresseras DATRAM.
Man måste kunna byta innehållet i DATRAM oberoende av den temporära minneskartans utseende. Dvs DATRAM får inte ha en adress som kan mappas bort. Detta har lösts genom att begränsa den logiska adressens värde. Signalen /ME som kopplar in minnet kommer från PALDC1. Minnet blir anropat inom området $0000 - $DFFF dvs 56K. Området över 56K är reserverat för systemets I/O och DC4BUG.

PAL-kretsen (PALDC1) avkodar I/O-adresser. Det finns en signal benämnd /EX som styr accessen mot DATRAM. När processorn skriver mot en adress i området $FFF0 - $FFFF så avkodas adressen i PALDC1. Ut från denna PAL kommer signalen /EX. /EX går till multiplexorns selectingång. Multiplexorn växlar läge och adressledningarna A0 - A3 når DATRAM istället för A12 - A15 samtidigt förses DATRAM med data från databussen. Det betyder att DATRAM kan programmeras med godtyckligt data.

1. SYSTEMBESKRIVNING

Det går endast att skriva mot DATRAM med ovan beskrivna konstruktion. Operativsystemet i EPCUF för därför en tabell som speglar innehållet i DATRAM. Tabellen har en fast placering i minnet ($DFDO - $DFDF). Förändras innehållet i DATRAM så ska tabellen updateras.

$DFDO motsvarar innehållet i DATRAM adress $FFFO, $DFD1..$FFF1 ..osv.

Vid uppstartning av systemet så måste DATRAM förses med vissa grundvärden, som framgår av nedanstående tabell.

DATRAM

innehåll adress		
	$FFFO	$00
	$FFF1	$01
	$FFF2	$02
	$FFF3	$03
	$FFF4	$04
	$FFF5	$05
	$FFF6	$06
	$FFF7	$07
	$FFF8	$08
	$FFF9	$09
	$FFFA	$0A
	$FFFB	$0B
	$FFFC	$0C
	$FFFD	$0D
	$FFFE	$0E
	$FFFF	$0F

Ovan uppräknade värden är alltså utgångsvärdena. Med dessa värden nås det normala adressområdet $0000 - $FFFF.

Ett exempel kan vara på sin plats.
Antag att innehållet i $FFFO ändras från $00 till $10. Antag också att processor skriver mot adressen $0100 i ett 4K block. Den totala adressen mot bussen ser nu ut så här:

$ 1 0 1 0 0

1. SYSTEMBESKRIVNING

1.3 MINNESKARTA

1.3.2 RAMDISK

Ramdisk är ett sekundärminne där man har emulerat (efterliknat) funktionen av en floppydisk i RAM. Det innebär att den måste formateras. Disken måste ha ett systemspår där FLEX kan finna SIR (System Information Record) och filkatalogen. Bootsektorn kan däremot utelämnas inte heller behöver disken innehålla någon formateringsbakgrund. RAMDISKEN är fysiskt placerad i det dynamiska minneskortet. Minnet är uppdelat i minnesbanker. Minnesområdet från 0 upptill 64KB användes för FLEX. Området därutöver delas mellan RAMDISK och PRINTERBUFFERT. Formatering av ramdisken görs med VDISK.CMD. Kommandot vdisk kommer att beskrivas mer ingående i SYSTEMBESKRIVNING - MJUKVARA.

1.3.3 PROMDISK

Det är den här funktion som verkligen lyfter konstruktionen och faktiskt förändrar FLEXsystemet till ett mycket attraktivt system. Operativsystemet FLEX gör på följande sätt när det tolkar ett kommando som användaren har skrivit efter prompten +++.

1. Datorn tolkar kommandot och hämtar detsamma från SYSTEMDRIVE placerar programmet i ram. Startar upp programmet.

2. Datorn fortsätter sedan att läsa in ev arbetsfil från WORKDRIVE

Alla dess läsningar mot floppydisken tar tid och det gör maskinen långsam även om processorn arbetar med en klockfrekvens på 2 MHz. Det är här som PROMDISKEN eller rättare sagt prommade kommandon kommer in i bilden. Operativsystemet FLEX är omskrivet (patchat) så att datorn i första hand försöker att finna efterfrågat kommandot i PROMDISKEN. Finns kommandot i prom så läser datorn in kommandot från detta.

1. SYSTEMBESKRIVNING

1.4 AVVIKELSE FRÅN TSC FLEX 09

EPCUF har del nya egenskaper som inte finns i orginalversionen av TSC FLEX. De flesta av oss som har varit med om att bygga en RT-dator eller dator av liknande konstruktion har väl sneglat åt datorer med finesser som printerbuffert, realtidsklocka, ramdisk och kanske kommandokatalog härbärgerad i ROM. Nu är det dags att sluta drömma och bara njuta av verkligheten.

FLEX kan i EPCUF bootas på två olika sätt. Det vanliga sättet är att läsa bootsektorn från skivan och sedan läsa ned hela FLEX i RAM. I EPCUF är det möjligt att läsa ned en prommad FLEX. Prom nr 13 på promkortet DC4 är helt reserverad för operativsystem FLEX. Datorn läser VIA:n på cpukortet (DC1) för att avgöra var ifrån bootning ska ske skivan eller promkortet.

Själva kärnan av FLEX (FLEX.COR) är på vissa ställen rejält omstuvad så här går det inte att känna igen sig längre. De nya funktioner som är införda har ställt nya krav på hård- och mjukvara. Nedan följer en uppräkning på de nya funktionerna och hur det har påverkat FLEX systemet.

Realtidsklocka
Läsning vid start av systemet. Det inte längre nödvändigt att svara med dagens datum innan systemet bootar. Rätt tid och datum läses från den batteriuppbackade klockan.

Printerspooler
Det utökade minnet användes också som printerbuffert. Utilities för att skriva mot printern är som förut.
När kommando P,LIST,TEST matas in till datorn sker följande.

1. P dirigerar till utskriften till printerbufferten (reserverad del av minnet).
2. LIST läser filen TEST.

Utmatning från minnet (bufferten) till printerporten sker sedan med jämna intervall m.h.a avbrott (interrupt). Avbrotten genereras av timern i VIA:n på CPU-kortet. I praktiken upplevs utmatning och fortsatt jobb mot datorn att ske samtidigt. Enda tillfället man märker att datorn utför två uppgifter är vid arbete mot floppydisken. Floppy har företräde framför printerutskriften.

1. SYSTEMBESKRIVNING

Kommandokatalog i PROM
Användarens kommando tolkas på annorlunda sätt i EPCUF. FLEX undersöker först om det finns ett DC4-kort anslutet i systemet. Om så är fallet försöker operativsystemet att i första hand hitta det efterfrågade kommandot promdiskens kommandokatalog. Finner FLEX inte kommandot så söker systemet vidare på SYSTEMDRIVE. Misslyckas sökningen även här så rapporteras fel i vanlig ordning.
DC4-kortet har plats för 14st 28pin promkapslar. Kortet kan bestyckas med olika typer av prom. Det är i första hand 2764, 27128, och 27256 som kommer ifråga. Bestyckas kortet med 13st 27256 och 1st 2764 fås minneskapacitet på 424KB. Den 14:e kåpan måste vara av typen 2764. Operativsystemet självt är placerat i detta minne.
Med 424KB som tillgängligt lagringsutrymme blir det möjligt att lägga FLEX utilities, assembler, editor mm i prom.

I/O
Inmatningsrutinerna mot FLEX är förändrade. Det går inte längre att ansluta en terminal. Terminalen är redan inbyggd i systemet. Datorn tar in data från tangentbordet via en ACIA placerad på CPU-kortet. Tecken som matas in i datorn passerar rutiner i DC4BUG och hamnar sedan hos FLEX. Till EPCUF ansluts ett tangentbord med seriellt snitt. Tangbordet ska lämna ascii-tecken med 1200 Baud överföringshastighet. Det kan i detta sammanhang nämnas att det går alldeles utmärkt att använda ett PC-tangentbord. Man måste dock sätta en processor mellan tangentbordet och danskdatorn som översätter PC-tangentbordets scankoder till ascii-tecken.

Utmatningsrutinerna mot bildskärm är förändrade. FLEX skriver inte längre mot en ACIA utan mot DC4BUG. I DC4BUG finns en driver för bildskärmen i systemet. MC6845 användes som bildprocessor. Programmet för bildskärmen emulerar en CT-82 terminal. CRT-kortet lämnar en composite videosignal. Till EPCUF ansluts en monitor med composite videoingång.

2. HÅRDVARUBESKRIVNING

2.1 PROCESSORKORT DC1

På cpukortet finns följande funktioner:

Processor 2MHz MC68B09E

Alla bussignaler är buffrade
(alla RESET är inte utan buffert. Det ställer till med vissa problem mer om detta i BUGGAR och ERFARENHETER av PDMEGA-datorn)

Utökat minnesområde till 1MB
För Virtuell disk (ramdisk) Printerbuffert
helt SWTPC kompatibel

4K systemmonitor
Bildskärm, Bootningsrutiner för FLEX

2K statiskt RAM på kortet

Buffrad seriell port för tangentbord
programmerbar överföringshastighet

Realtidsklocka med batteribackup
timer och alarm funktion

Parallelport VIA Centronics interface
16 bit timer (används för printerspooler)

Cpukortet avsett för operativsystem FLEX

Konstruktionen använder PAL-kretsar för adressavkodning och alstring av olika kontrollsignaler. Pal-kretsar har blivit mycket populära att konstruera med. Det ger konstruktören mölighet att hålla antalet kåpor ne och att också dölja konstruktionen.
Det är användning av PAL-kretsar som gör konstruktionen lite svår att genomskåda. Nedan följer en genomgång av PAL-kretsen på CPU-kortet.

PALDC1 används för att avkoda olika adressområde. PALDC1 är av typen 16L8 vilket betyder att den har 16st in- och 8st utgångar.
Signalerna A8 - A15, R/W, Q och E (11st) är anslutna till DC1:s ingångar.
Ut från PALDC1 kommer följande signaler:

/RE	enable för Eprom pin /OE	
/RO	enable för Eprom pin /CE	$F000 - $FFFF
/RA	enable för statiskt Ram	$E100 - $E7FF
/EX	strobe DATRAM	$FF00 - $FFFF
/IO	enable extern I/O	$E000 - $E0FF
/E8	enable videoram	$E800 - $EFFF
/ME	enable extended memory	$0000 - $DFFF

/IO och A4-A7 används till att avkoda adresserna till DC1-kortets I/O. Realtidsklocka, VIA, och ACIA.

RTC	adress	$E002 - $E003
VIA	adress	$E070 - $E07F
ACIA	adress	$E008 - $E009

2. HÅRDVARUBESKRIVNING

KOMPONENTLISTA DC1-kort

IC1 - IC4	74LS245	oktal buffert
IC5 - IC6	74S189	bipolärt RAM 16x4
IC7	74LS157	multiplexor 4-bit
IC8	2732	4K eprom för DC4BUG
IC9	MC68B09E	CPU 2MHz extern klocka
IC10	6116	2K statiskt RAM
IC11	74LS30	8-ing. NAND
IC12	PALDC1-B	Palkrets MMI 16L8
IC13 - IC14	74LS138	avkodare 1 av 8
IC15	MC68B50	ACIA 2MHz
IC16	6522A	VIA 2MHz
IC17	M3000	Realtidsklocka
IC18 - IC19	74LS245	oktal buffert
R1 - R3	9x4.7K	resistansnät
R4 - R6	4.7K	motstånd 4.7K 1/4W
R7	680 ohm	motstånd 680 ohm 1/4W * se anm.
R8 - R9	4.7K	motstånd 4.7K 1/4W
R10 - R11	4.7K	motstånd 4.7K 1/4W * se anm.
R12	10K	motstånd 10K 1/4W
R13	10K	motstånd 10K 1/4W
C1	10uF	tantalkond 10uF 10V
C2	10uF	tantalkond 10uF 10V
C3	5-40pF	trimkond. 5-40pF
C4 - C5	22pF	keramisk kond. 22pF * se anm.
CX	100nF	kond 100nF
XTAL1	8 MHz	kristall 8 MHz * se anm.
XTAL2	32768 Hz	kristall 32768 Hz
BAT	2736	litiumbatteri
J1	bygel	* se anm.
J2	bygel	strapp FIRQ från RTC
S1	64-pol	europadon 64-pol
S2	10-pol	2-radig stiftlist
S3	20-pol	2-radig stiftlist

Anmärkning:
Vid användning av MC68B09E utelämnas följande komponenter
R10 - R11
C4
XTAL1
samt ersätt C5 med en trådbygel

Vid använding av MC68B09 utelämnas följande:
J1
R7

2. HÅRDVARUBESKRIVNING

2.1 PROCESSORKORT DC1

2.1.1 DC4BUG SYSTEMMONITOR

Systemmonitorn i EPCUF skiljer sig ifrån Debuggers som SWTPC CBUG2.3. Den har inte som de andra kommandon för att ändra minneceller, dumpa minne, boota från floppy ..osv. EPCUF har en modifierad FLEX. Dessa förändringar tar större plats i minnet än den plats som är reserverad för FLEX. Rutiner återfinns i DC4BUG. Nedan följer en uppställning av DC4BUG innehåller.

rutiner för att emulera en CT-82 terminal
screensaver
rutiner för att läsa och ställa realtidsklockan
rutiner för sökning och laddning av filer från PROMDISKEN
bootstrap för floppy
drivrutiner för RAMDISK
initiering av I/O på CPU-kort
initiering av DATRAM CPU-kort
avbrottsrutiner

Efter poweron eller reset startar alltid EPCUF med att boota FLEX från skiva eller prom. Var ifrån bootning sker har datorn avgjort genom att läsa PB2 (VIA:n) på CPU-kortet. Om man håller någon tangent nedtryckt under själva bootningen stannar DC4BUG upp och skriver <C> T or R. Operatören kan då välja mellan T(testa minnet) och R(gå in i programmet REDIT.CMD). Ovanstående sker endast om datorn har ett DC4-kort på bussen och att EPROM nr 12 innehållande ett program SYSMON.CMD.

Det finns några rutiner i DC4BUG som det kan vara av intresse att känna till när man ska skriva egna program för systemet och man inte vill använda FLEX:s rutiner.

MONITOR	$F884	Varmstart av DC4BUG
INCH	$F96A	Läs tecken från ACIA(DC1) utan att eka till skärmen tecknet läggs i acc.A utan att maska bit7
INCHE	$F9A1	Läs tecken från ACIA eka till skärmen tecknet läggs i acc.A med maskad paritetsbit
INCHEK	$F991	Testa om det finns något emottaget tecken Z-flaggan = 0 om det finns ett tecken i ACIA:n
OUTCH	$F022	Sänd tecken till bildskärmen teckenet läggs i acc.A
PDATA	$F6FD	Sänd sträng utpekad av X-reg mot bildskärmen strängen avslutas med EOT(04)
PCRLF	$F6F4	Sänd <CR><LF> mot bildskärmen
PSTRNG	$F6F0	Sänd först <CR><LF> därefter sträng utpekad av X-reg strängen avslutas med EOT(04)

2. HÅRDVARUBESKRIVNING

2.2 FLOPPY/CRT-KORT DC2

Funktionerna på FLOPPY/CRT-kort DC2 kan sammanfattas enligt nedan.

2MHz MC68B45 CRT kontroller används för att generera en bild
med 24 rader och 80 tecken/rad
Bildminnet 2KB ligger direkt i processorns minneskarta
$E800 - $EFFF (MEMORY MAPPED RAM)

Rutinerna för bildskärmen finns i DC4BUG
Emulerar det mesta av SWTPC CT-82 terminalen utom grafiken

Lämnar composite videosignal ut

FLOPPY:

Floppy Disk Controller WD 2797

Klarar diskar med
5" eller 8"
enkel eller dubbel packning
diskar med enkel eller dubbel sida

Upp till 4 diskar kan anslutas

Buffrade signaler mot bussen och floppydiskarna.

Systemklocka 16MHz nerdelad till 2MHz

Adressavkodningen har lösts med en PAL-krets PALDC2.
Signalerna /BU1, /CE och /WC används till CRTC:n

/BU1	enable för bussdrivare
/CE	enable för CRTC
/WC	strobe skrivning mot bildminne

Övriga signaler använder FDC

/BU2	enable bussdrivare
/RD	strobe för att läsa DRQ och IRQ från FDC
/WE	strobe för skrivning mot FDC
/WD	strobe för driveregister
/FLP	enable FDC WD2797

2. HÅRDVARUBESKRIVNING

KOMPONENTLISTA FLOPPY/CRT-KORT DC2

IC1	PALDC2	Palkrets MMI 10L8
IC2 - IC4	74LS245	oktal buffert
IC5	7404	hex inverter
IC6	74LS163	synkron räknare
IC7	74LS86	exclusive OR grind
IC8	WD 2797	Floppy-controller
IC9	MC 68B45	CRT Controller
IC10 - IC12	74LS157	multiplexor 4-bit
IC13	74LS74	data vippa
IC14	9602	dubbel monovippa
IC15	74LS175	fyra D-vippor
IC16	6116	statiskt 2K RAM
IC17	74LS175	fyra D-vippor
IC18	2003	ULN2003 relädrivare
IC19	74LS138	avkodare 1 av 8
IC20	74LS06	hex inverter open collector
IC21	74LS574	oktal latch
IC22	2732	Eprom 2732 teckengenerator
IC23	74LS245	oktal buffert
IC24	74LS07	hex inverter open collector
IC25	74LS166	Skiftregister 8-bit
R1 - R2	470 ohm	motstånd 470 ohm 1/4W
R3	4.7K	motstånd 4.7K 1/4W
R4	1K	motstånd 1K 1/4W
R5	47K	motstånd 47K 1/4W
R6	4.7K	motstånd 4.7K 1/4W
R7	2.2K	motstånd 2.2K 1/4W
R8	4.7K	motstånd 4.7K 1/4W
R9	150 ohm	motstånd 150 ohm 1/4W
R10	1K	motstånd 1K 1/4W
R11	330 ohm	motstånd 330 ohm 1/4W
R12	82 ohm	motstsånd 82 ohm 1/4W
R13	100 ohm	motstånd 100 ohm 1/4W
R14	50K	trimpot. 50K 10-varv.
R15	10K	trimpot. 10K 10-varv.
R16	9x150 ohm	resistansnät 9x150 ohm
R17 - R18	4.7K	motstånd 4.7K 1/4W
C1	22pF	keramisk kond. 22pF
C2	5-40pF	trimkond. 5-40pF
C3	220nF	polyesterkond. 220nF
C4	47uF	tantalkond 47uF 10V
C5	10uF	tantalkond. 10uF 10V
CX	100nF	polyesterkond. 100nF
D1	1N4148	diod 1N4148
D2	3.6V	zenerdiod 3.6V 1N748
T1 - T2	BC307B	transistor BC307B
S4	34-pol	kontakt 34-pol 2-radig
S5	4-pol	kontakt 4-pol 1-radig

2. HÅRDVARUBESKRIVNING

2.2 FLOPPY/CRT-KORT

2.2.1 CRT KONTROLLKARAKTÄRER OCH FUNKTIONSTANGENTER

2. HÅRDVARUBESKRIVNING

2.2.2 CRT TECKENPROM

2. HÅRDVARUBESKRIVNING

2.3 EPROMKORT DC4

Epromkortet har en kapacitet på 424KB. Epromkortet utgör en egen minnes area dold för datorn bakom några portar. Konstruktionen upptar endast 4 adresser av datorns adressutrymme. Koden i eprommarna når man genom en indirekt adressering.

Inne på kortet finns tre stycken register bestående av IC3 - IC5 (74LS574). IC4 pekar ut vilken promkapsel som man läser från. IC3 och IC5 pekar ut adressen inom resp. minneskapsel. IC5 står för den lägre minnesadressen och IC3 för den högre.

IC3 har adress $E020 den lägre adressbyten
IC5 har adress $E021 den högre adressbyten
IC4 har adress $E022 pekar ut eprom på kortet
IC2 har adress $E023 databuffert mot bussen

Ett exempel kanske kan göra det litet begripligare. Antag att vi vill läsa bytes på adress $0100 - $0200 från eprom nr 7. Vi går då tillväga på följande sätt.

a) välj eprom på kortet
   skicka en byte med värdet $07 till register(IC4) på adress
   $E022

b) sätt upp adress $0100 mot prommet
   Skicka talet $00 till IC5 som ligger på adress $E021
   och $01 till IC3 på adress $E022.

c) Läs byte och placera i RAM
   Byten från eprom nr 7 adress $0100 fås genom att läsa adress
   $E023.

Ska vi läsa 256 byte från promminnet måste ovanstående procedur upprepas 256 ggr. Det sker lättast i en loop där man ökar adressutpekning med ett steg för varje varv i loop:en.

2. HÅRDVARUBESKRIVNING

Ovanstående förklarar principen för hur man kan läsa kod från epromkortet. Programmen som ligger i prommen är lagrade i ett speciellt format som är nytt i FLEX-sammanhang. De första sex byten i varje program har en mycket viktig funktion.

```
BYTE 0 - 1    LADDADRESS postion var i RAM koden läggs
BYTE 2 - 3    SLUTADRESS längden på programmet
BYTE 4 - 5    STARTADRESS för programmet
```

När ett program eller en utility läggs till promdisken så måste dess kod packas om. Denna packning sköter prombränningsprogrammet om.

I prom nr 12 finns en filkatalog över promdisken. Det är viktigt att nämna att det går endast att lagra filer med extension .CMD på promkortet. Det går alltså inte att lagra filer som ERRORS.SYS, PRINT.SYS, STARTUP.TXT ...osv. Filkatalogens format avviker också från katalogen på flexskivorna.

Filkatalogen i eprom nr 12 börjar på adress $7800. Den innehåller programnamn, eprom nr, adress till prommet se nedan.

```
        ORG $7800          START ANVÄNDAR TABELL

        FCC 'DEBUG',0      FILNAMN DEBUG avslutas med NULL
        FCB $11            EPROM NR 11
        FDB $1420          ADRESS i eprom nr 11
        END
```

Ovanstående är ett exempel på hur man bygger upp sin filkatalog. I avsnittet om epromprogrammering för DC4-kortet ges en mera detaljerad beskrivning.

2. HÅRDVARUBESKRIVNING

De sista byten i varje promkapsel talar om för systemet storlek och typ av eprom samt länk till nästa eprom på kortet. Det kan hända att ett program startar i en promkapsel och fortsätter i en annan. Systemet klarar självt övergången mellan olika prommar och promtyper.
De olika typer av eprom som får förekomma på DC4-kortet är 2764, 27128 och 27256. Det finns vissa möjligheter att använda 27512 men då krävs ändring av DC4:s hårdvara.
Utseendet på de sista fyra byten är beroende på vilken typ av eprom som används. Bränns prommarna för DC4-kortet med programmet DCBURN.CMD så räknar programmet själv ut alla koder och programmerar dessa i minnet.
Nedan följer en uppställning på hur koderna ser ut.

2764:

```
BYTE $0000 - $1FFB              programkod
BYTE $1FFC - $1FFD  $1FFC       eprom storlek-4
BYTE $1FFE          $E0         eprom typ 2764
BYTE $1FFF          $05         länk till nästa eprom t.ex §5
```

27128:

```
BYTE $0000 - $3FFB              programkod
BYTE $3FFC - $3FFD  $3FFC       eprom storlek-4
BYTE $3FFE          $C0         eprom typ 27128
BYTE $3FFF          $03         länk till nästa eprom t.ex §3
```

27256:

```
BYTE $0000 - $7FFB              programkod
BYTE $7FFC - $7FFD  $7FFC       eprom storlek-4
BYTE $7FFE          $80         eprom typ 27256
BYTE $7FFF          $01         länk till nästa eprom t.ex §1
```

27512:

```
BYTE $0000 - $FFFB              programkod
BYTE $FFFC - $FFFD  $FFFC       eprom storlek-4
BYTE $FFFE          $00         eprom typ 27512
BTYE $FFFF          $0A         länk till nästa eprom t.ex §10
```

2. HÅRDVARUBESKRIVNING

KOMPONENTLISTA ÖVER EPROMKORT DC4

IC1	PALDC4	MMI 10L8
IC2	74LS245	oktal buffert
IC3 - IC5	74LS574	oktal latch
IC6	74LS154	avkodare 1 av 16
IC7 - IC20		EPROM 2764, 27128, 27256

OBS !!! IC7 måste av typen 2764

CX	100nF	polyesterkond. 100nF
S1	64-pol	europadon 64-pol

2. HÅRDVARUBESKRIVNING

2.4 IO/EPROMBRÄNNARE

Funktioner på I/O-kortet.

2st serieinterface ACIA 6551 med mjukvarustyrd överföringshastighet

1st parallelinterface VIA 6522 (Centronic)

1st programmeringssockel för eprom

1st spänningsaggregat 12.5V och 21V för programmering av eprom

Palkretsen används i adressavkodningen för kortet. Ut från Pal:en kommer 4st signaler /BU, /VIA, /PIA, /ACIA1 och /ACIA2.
IC8 (TL497) utgör huvuddelen i den programmerbara spänningskällan för eprombrännaren. Eprombrännaren klarar att bränna eprom med Vpp 12.5V och 21V. Reläer används för att växla funktionerna på epromsockelns anslutningstift så att val av epromtyp kan styras helt av programmet DCBURN.CMD

Parallellporten (VIA) på kortet delas mellan att dels vara adress mot eprommet dels vara centronic-interface.

Seriekanalerna är litet olika utrustade. Det ena interface:t är tänkt att kopplas ihop med ett modem och har därför alla nödvända signaler utdragna till anslutningenskontakten. Seriekanal nr 2 däremot är tänkt att jobba mot en t.ex en printer med seriell kommunikation. Interface 2 har därför endast TXD, RXD, RTS, CTS samt GND utdragna till kontakten. Det finns dock lediga funktioner i V24-buffertarna varför det är möjligt att göra interface 2 identiskt med interface 1.

2. HÅRDVARUBESKRIVNING

KOMPONENTLISTA ÖVER IO-EPROMBRÄNNARE DC5

IC1	PALDC5	MMI 10L8
IC2	74LS245	oktal buffert
IC3 - IC5	C42	diprelä 5V med 1-pol växling
IC6	2003	ULN2003 relädrivare
IC7	C42	diprelä 5V med 1-pol växling
IC8	TL497	DC-omvandlarkrets
IC9	7404	hex inverter
IC10	MC68B21	PIA
IC11	6551A	ACIA
IC12	C42	diprelä 5V med 1-pol växling
IC13	6551A	ACIA
IC14	6522A	VIA
IC15	1488	RS232A buffert sändare
IC16	1489	RS232A buffert mottagare
IC17	1488	RS232A buffert sändare
IC18	1489	RS232A buffert mottagare
IC19	74LS245	oktal buffert
IC20	74LS245	oktal buffert
C1	100uF	el.lyt.kond. 100uF 25V
C2	100pF	keramisk kond. 100pF
C3	100pF	keramisk kond. 100pF
C4	10uF	el.lyt.kond 10uF 10V
C5	150pF	keramisk kond. 150pF
CX	100nF	polyesterkond. 100nF
R1	2.2K	motstånd 2.2K 1/4W
R2	4.3K	motstånd 4.3K 1/4W
R3	10K	motstånd 10K 1/4W
R4	1 ohm	motstånd 1 ohm 1/4W
R5 - R6	470 ohm	motstånd 470 ohm 1/4W
H1	100uH	drossel 100uH
XTAL	1.8432MHz	kristall 1.8432 MHz
REG	7806	regulator 6V
K1	64-pol	europadon 64-pol
K2 - K3	10-pol	2-rad. stiftdon
K4	20-pol	2-rad. stiftdon

2. HÅRDVARUBESKRIVNING

2.5 1MB DYNAMISKT MINNESKORT DC6

Minneskortet använder dynamiska minnen. Dynamiska minnen brukar normalt kräva en hel del kretsar för att klara av refreshen. Det kommit en ny typ av minnen vilka utnyttjar en teknik som kallas "CAS BEFORE RAS REFRESH". Dynamiska minnen har två strobsignaler RAS och CAS. RAS står för Row Address Strobe och CAS för Column Address Strobe. När man skriver och läser mot minnet kommer signalerna i bestämd turordning. Funktionen är som följer.

Antag att man ska läsa innehållet i en cell

a) skicka adresserna A0 - A7 till minnet

b) stroba RAS A0 - A7 läggs i register internt i minneskåpan

c) skicka adresserna A8 - A15 till minnet

d) stroba CAS
   nu aktiveras minnet utgång och man kan läsa innehållet i adresserad minnescell.

Ovanstående är bara ett starkt förenklat resonemang på hur dynamiska minnen arbetar. Dynamiska minnen måste ha refresh för att bibehålla informationen i sina celler. Det betyder att inom en viss tidsrymd måste alla celler i minnet ha blivit adresserade. Själva refreshlogiken finns internt i varje minneskåpa. Att göra refresh på alla minnesceller tar tid och det skulle i värsta fall innebär det kunde inkräkta på datorns kapacitet för andra uppgifter. DC6-kortet har minnen med CAS before RAS. Det innebär att on man ändrar på turordningen mellan RAS och CAS så kommer minnet uppfatta det som begäran om refresh och utföra den. Pal-kretsen PALDC6+ har här en central funktion. Kretsen användes dels som adressavkodare för utpekning av minnesbank dels som styrning för adressmultiplexorn.

2. HÅRDVARUBESKRIVNING

KOMPONENTLISTA MINNESKORT DC6

IC1 - IC2	74LS157	multiplexor 4-bit	
IC3	74LS245	oktal buffert	
IC4	PALDC6+	MMI 16L8	
IC5	74LS14	hex inverter schmitt trigger	
IC7 - IC38	HM50256-15	dynamiskt minne	* se anm.
IC39	74LS14	hex inverter	* se anm.
R	27 ohm	motstånd 27 ohm 1/4W	
CX	100nF	kond. 100nF	
S1	64-pol	europadon 64-pol	

anmärkning

Endast minnen med CAS before RAS är lämpliga

Det har tillkommit en krets (74LS14)för att avhjälpa ett fel i konstruktionen.

2. HÅRDVARUBESKRIVNING

2.6 BAKPLAN

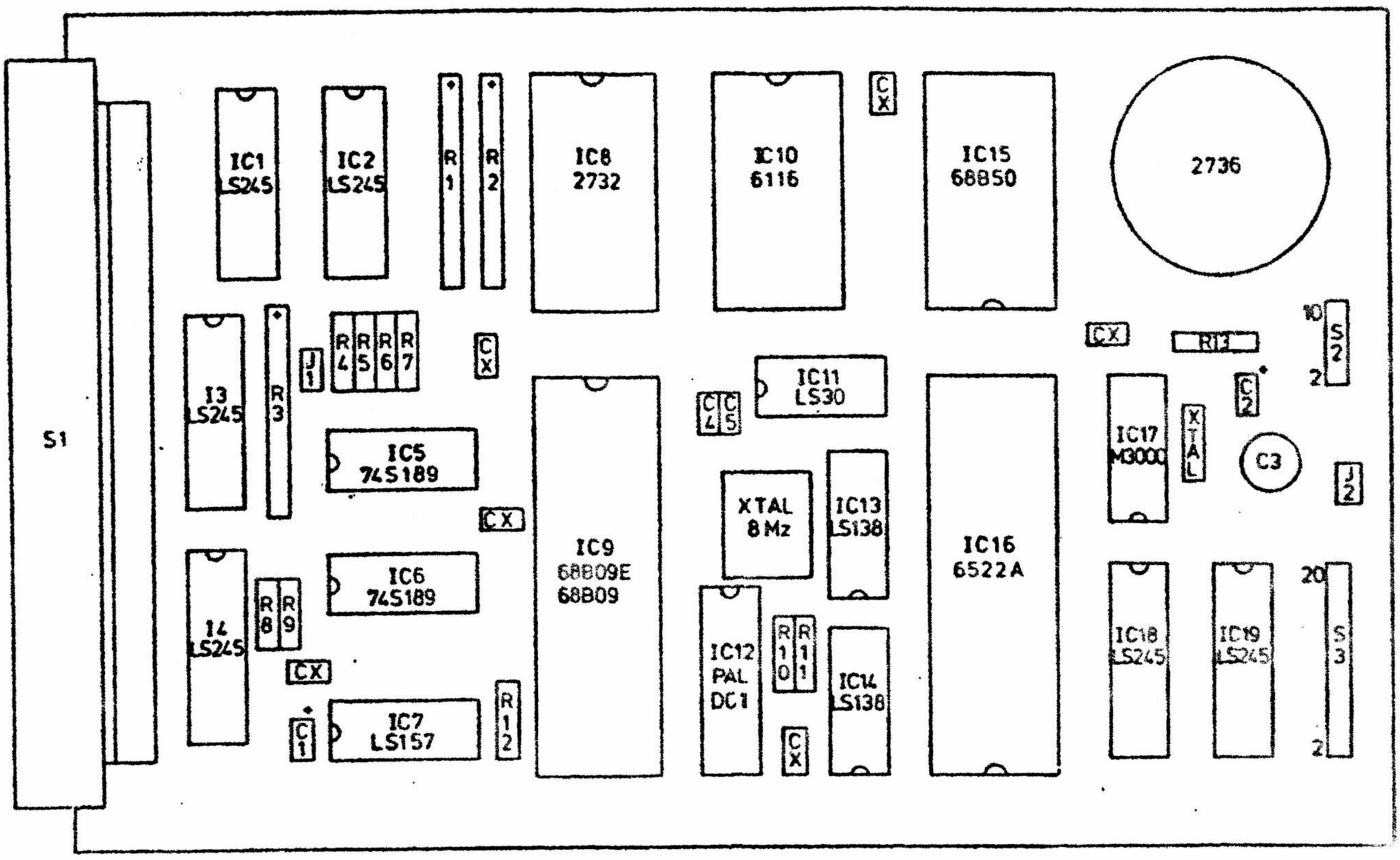
2. HÅRDVARUBESKRIVNING

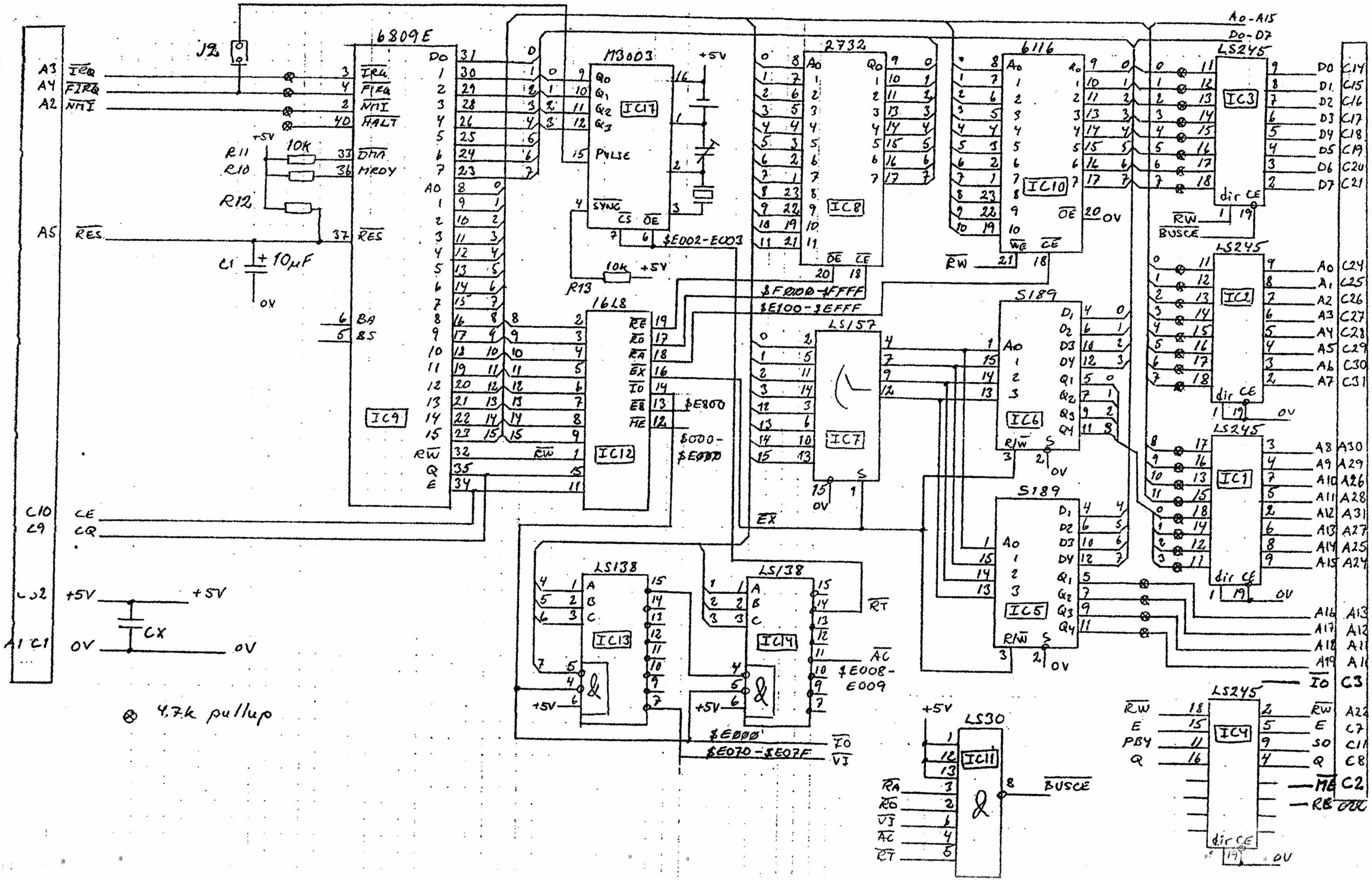
2.7 KRAFT

2. HÅRDVARUBESKRIVNING

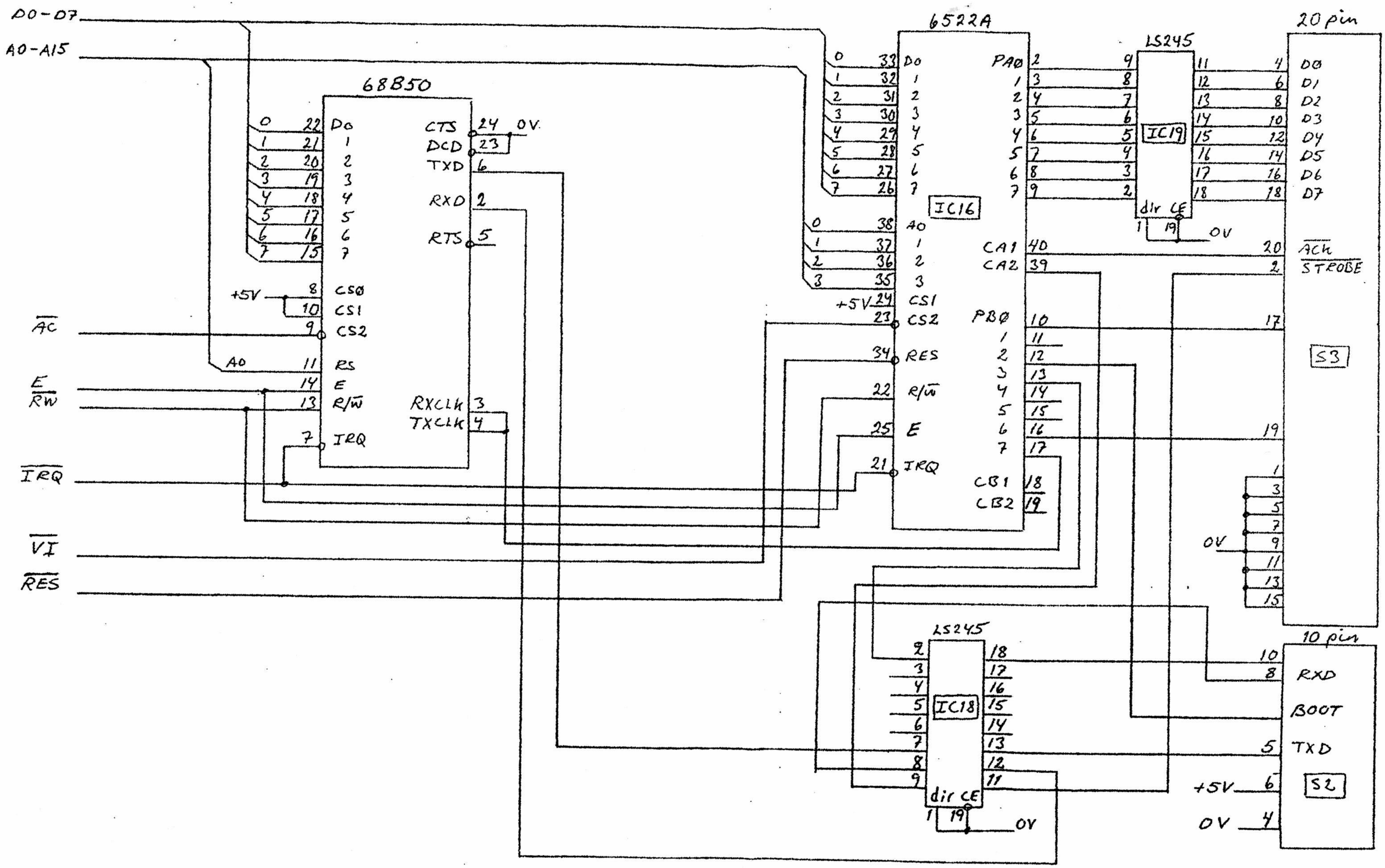
2.8 PC-TANGENTBORDSINTERFACE

***** DC1 CPU BOARD *****



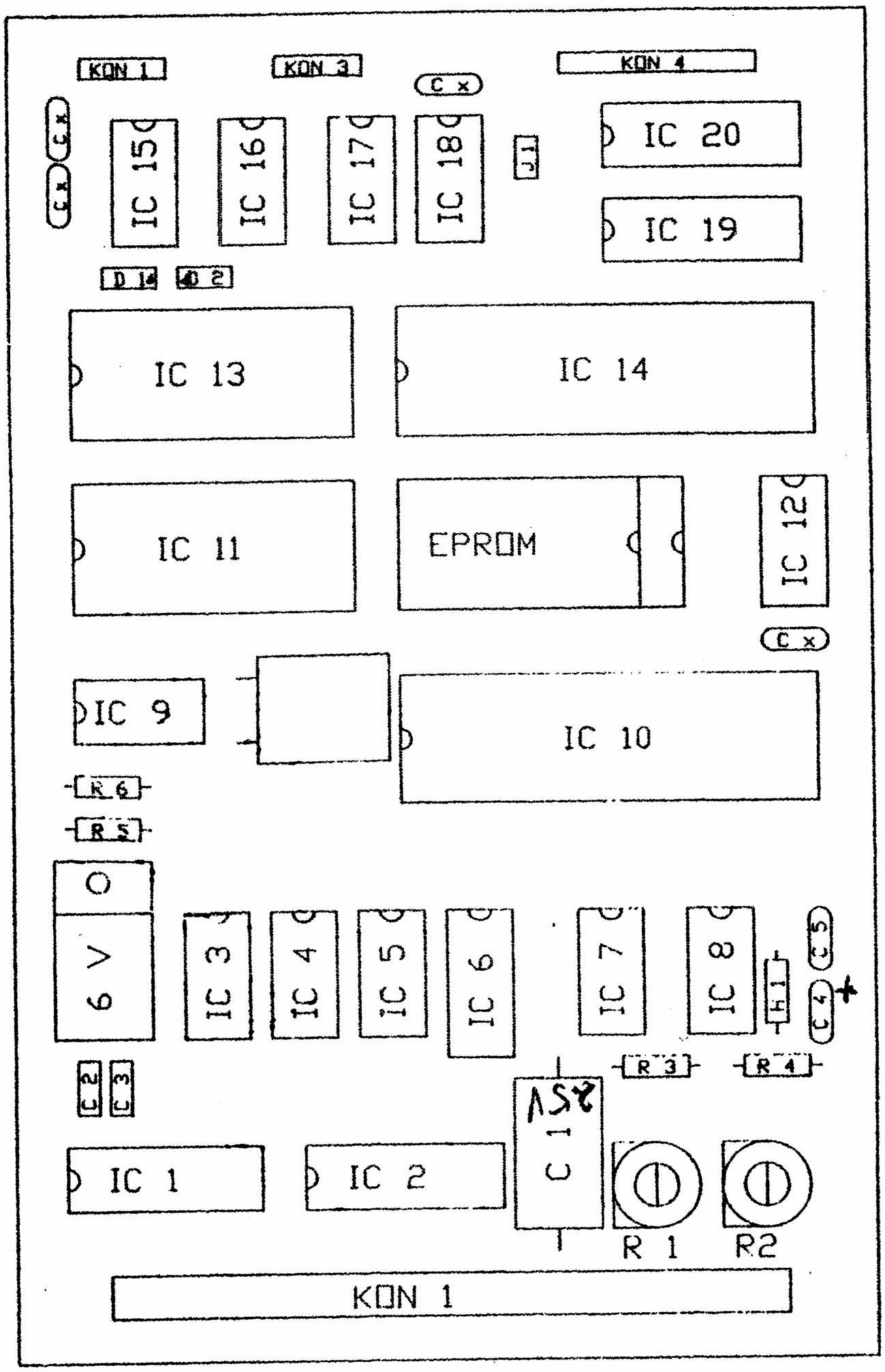


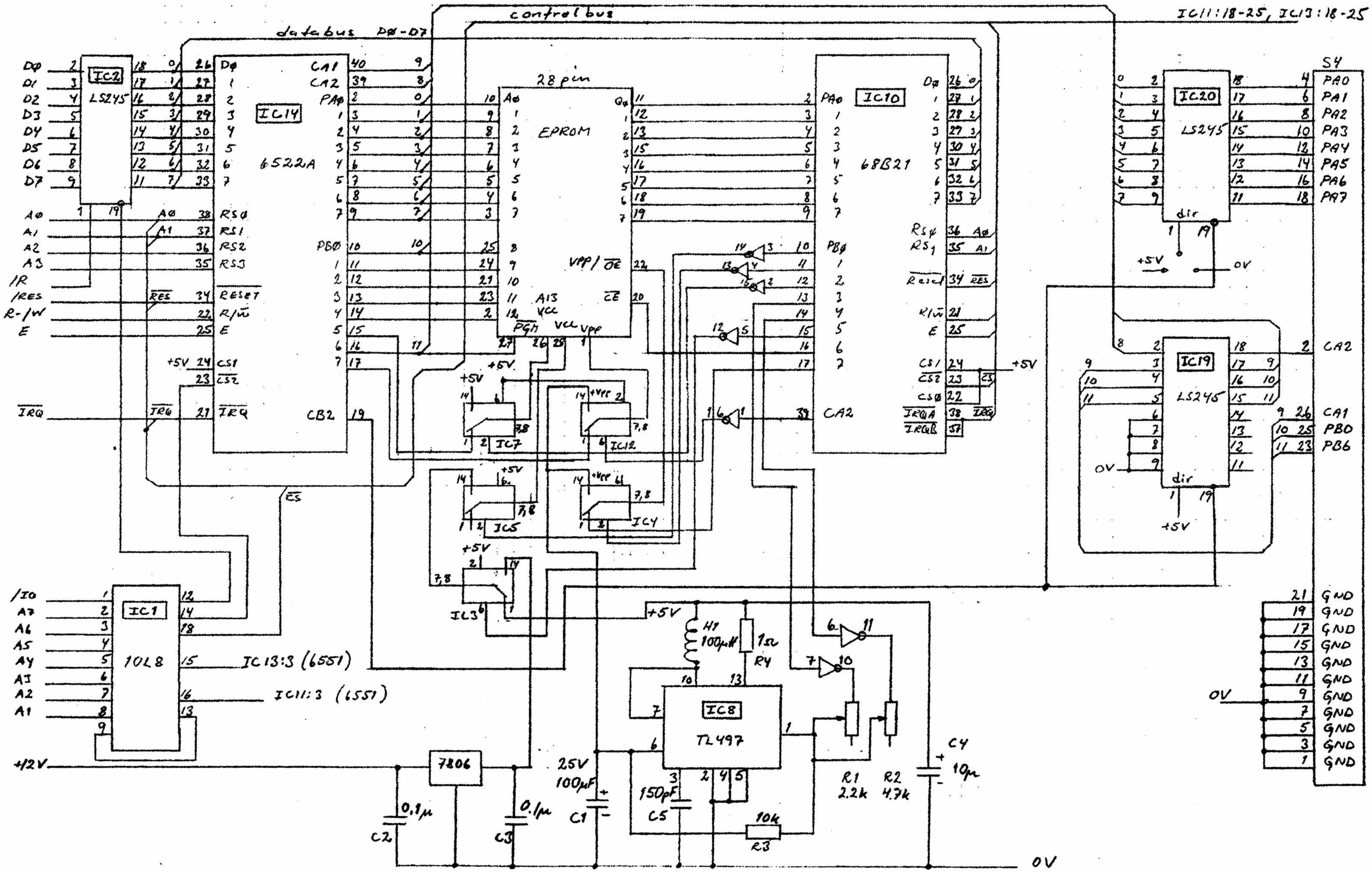
6809E
M3003
2732
6116
LS245
16L8
LS157
S189
LS138
LS30
IC1
IC2
IC3
IC4
IC5
IC6
IC7
IC8
IC9
IC10
IC11
IC12
IC13
IC14
IC17
J2
R11
R10
R12
R13
10K
C1
10µF
CX
+5V
0V
$E002-E003
$F000-$FFFF
$E100-$EFFF
$E800
$000-$E000
$E008-E009
$E000
$E070-$E07F
BUSCE
A0-A15
D0-D7
4.7k pullup



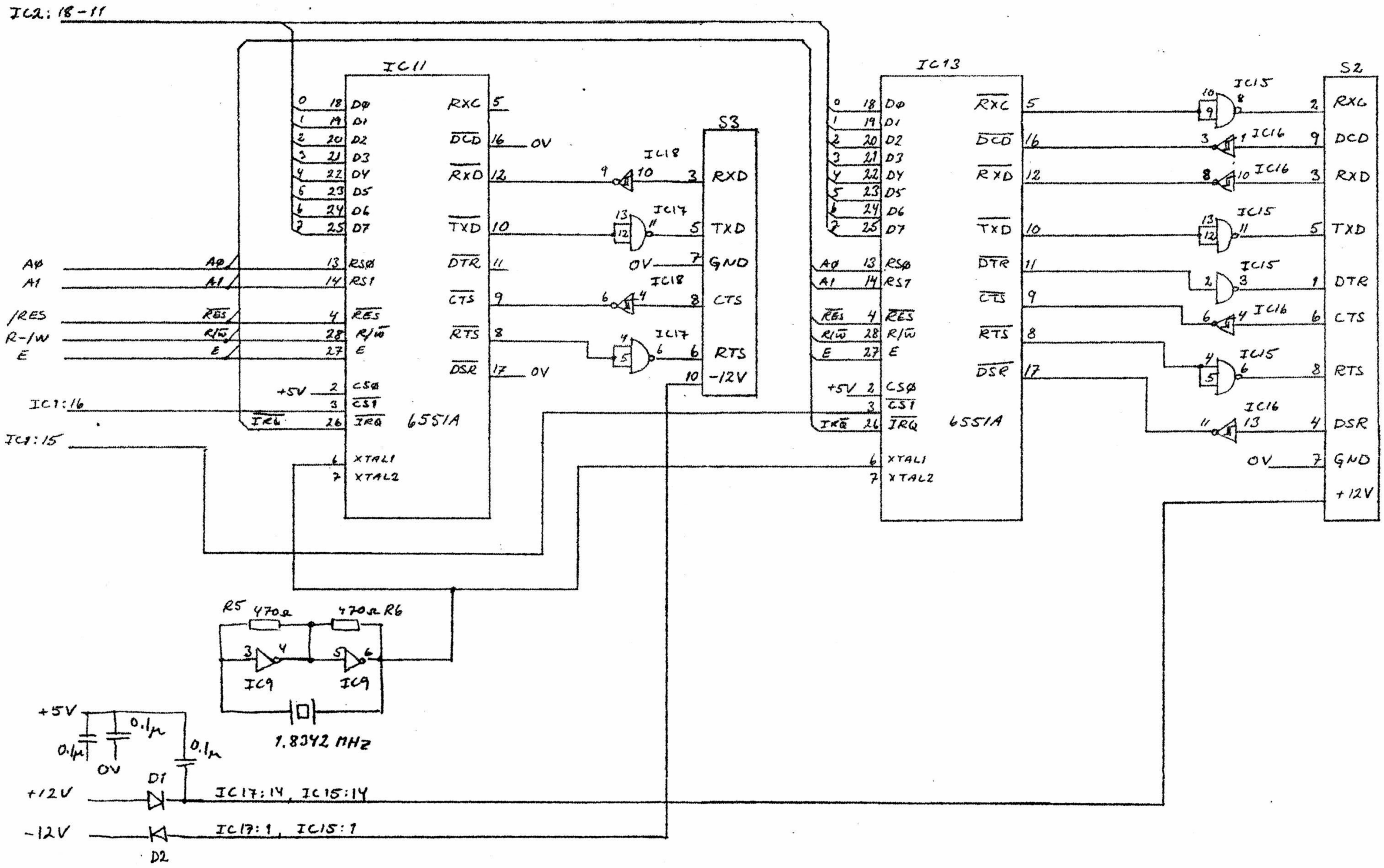
D0-D7
A0-A15
68B50
CTS
DCD
TXD
RXD
RTS
RXCLK
TXCLK
CS0
CS1
CS2
RS
E
R/W
IRQ
+5V
0V
AC
E
RW
IRQ
VI
RES
6522A
IC16
PA0
CA1
CA2
PB0
CB1
CB2
RES
LS245
IC19
dir
CE
20 pin
D0
D1
D2
D3
D4
D5
D6
D7
ACK
STROBE
S3
LS245
IC18
10 pin
RXD
BOOT
TXD
S2

IO BOARD DC5


KON 1
KON 3
KON 4
C x
C x
C x
IC 15
IC 16
IC 17
IC 18
J1
IC 20
IC 19
D 14
D 2
IC 13
IC 14
IC 11
EPROM
IC 12
C x
IC 9
IC 10
R 6
R 5
D
6 V
IC 3
IC 4
IC 5
IC 6
IC 7
IC 8
H1
C5
C4
C2
C3
R 3
R 4
IC 1
IC 2
C 1
R 1
R2
KON 1



IC11:18-25, IC13:18-25
control bus
databus D0-D7
IC2
LS245
IC14
6522A
28 pin
EPROM
IC10
68B21
IC20
LS245
IC19
LS245
S4
IC1
10L8
IC13:3 (6551)
IC11:3 (6551)
IC7
IC12
IC5
IC4
IC3
IC8
TL497
7806
+12V
0V
+5V


IC2: 18-11
IC11
IC13
S3
S2
IC15
IC16
IC17
IC18
6551A
RXC
DCD
RXD
TXD
DTR
CTS
RTS
DSR
GND
+12V
-12V
+5V
0V
A0
A1
/RES
R-/W
E
IC1:16
IC1:15
XTAL1
XTAL2
R5 470Ω
R6 470Ω
IC9
1.8342 MHz
0.1μ
D1
D2
IC17:14, IC15:14
IC17:1, IC15:1

SYSTEMBESKRIVNING - MJUKVARA

PROMMADE KOMMANDON

Till EPCUF hör ett FLEX-system som är speciellt skrivet för datorn. Till datorn hör också program som är speciellt framtagna. Grundrepertoaren av program och utilities kan utan ändringar användas i EPCUF. Dessa kommandon kommer bara kort att beröras emedan det går att läsa om dem i FLEX manualen. Det krävs också att varje person som önskar erhålla grundmjukvaran för danskdatorn innehar en egen FLEX-licens.
Nedan följer en uppräkning av de kommandon som finns i standardutförandet av eprom 12.

APPEND.CMD	BCOPY.CMD	BUILD.CMD	CAT.CMD
COPY.CMD	DATE.CMD	DELETE.CMD	DIR.CMD
EPCAT.CMD	EXEC.CMD	EXER.CMD	I.CMD
JUMP.CMD	LINK.CMD	LIST.CMD	MAP.CMD
NEWDISK.CMD	O.CMD	P.CMD	PROT.CMD
PRTSET.CMD	REDIT.CMD	RENAME.CMD	RTCSET.CMD
SAVE.CMD	SAVELOW.CMD	STAT.CMD	TIME.CMD
TTYSET.CMD	VDHELP.CMD	VDISK.CMD	VERIFY.CMD
VERSION.CMD	XOUT.CMD		

Det finns kommandon som inte är FLEX standard och dessa ska vi beskriva mer ingående.

Följande kommandon är speciellt utvecklade för danskdatorn

BCOPY.CMD
DBASIC.CMD
EPCAT.CMD
PRTSET.CMD
RTCSET.CMD
STAT.CMD
VDISK.CMD

BCOPY

BCOPY är ett kopieringsprogram avsett att kunna köras när man har DBASIC:en laddad. Normalt överlappar basic och t.ex COPY varandra och det blir minneskonfikt som gör att datorn låser sig.
BCOPY har samma syntax som COPY. För att kunna använda BCOPY krävs det att man har reserverat utrymme med option R i VDISK.CMD.

Beskrivning av kommando, den syntaxen för BCOPY är:

a) +BCOPY ÄoptionÅ,<filespec>,<filespec>
b) +BCOPY ÄoptionÅ,<filespec>,<drive>
c) +BCOPY ÄoptionÅ,<drive>,<drive>Ä,<match list>Å

+ används för att in ifrån BASIC kunna ge ett kommando ut till FLEX och för att sedan återvända.

Optioner som finns för kommandot kan man få fram genom att endast skriva +BCOPY på kommandoraden.

Några exempel

+BCOPY 0.START.TXT,2	Kopiera filen START.TXT från drive 0 till drive 2
+BCOPY P,1,2	Fråga fil för fil vilken som ska kopieras från drive 1 till drive 2
+BCOPY 0,1 CA	Kopiera alla filer från drive 0 till drive 2 som börjar med bokstäverna CA.

SYSTEMBESKRIVNING - MJUKVARA

EPCAT

EPCAT är ett kommando som används för att se vilka filer som finns i PROMDISKEN (EpromCatalog).

Beskrivning

Syntaxen för kommandot är:

```
  EPCAT
```

Har man satt upp TTY-parametrarna så kan man få utskriften att stoppa efter en skärm. Utskriften kan t.ex se ut som nedan

```
DIRECTORY OF DC-4 EPROM BOARD

NAME              EPROM §  ADDRESS   BEGIN  END   TRANSFER  SIZE

APPEND.CMD          12       1D50     C100   C313  C100       532
BCOPY.CMD           12       6589     C100   C102  FFFF         3
BUILD.CMD           12       1F6A     C100   C1B9  C100       186
COPY.CMD            12       24F1     0100   1284  0100      4485
DATE.CMD            12       367C     C100   C201  C100       258

osv ...
```

Förklaring på överskriften

NAME — Namnet på det program som finns i prom

EPROM § — Avser numret på det eprom i vilket programmet ligger

ADRESS — Adressen till programmet i t.ex eprom 12

BEGIN — Startadressen till vilken programmet laddas i RAM

END — Slutet på programmet

TRANSFER — Exekveringsadress för programmet
Observera det kan i bland stå FFFF på denna plats. Det betyder att programmet består av ett antal programsegment. FFFF är ett märke som talar om för programmet som laddar koden att koden fortsätter.

SIZE — Anger programmets storlek i byte. I det fall att TRANFERS adressen är FFFF får SIZE ett felaktigt värde.

PRTSET

PRTSET är ett kommando liknande TTYSET men som istället sätter parametrar för printern ansluten till datorn.

Syntaxen för kommando PRTSET är:

PRTSETÄ,<parameter list>Å

Där parameter-lista är en lista på ord med 2 bokstäver. Varje ord ska följas av likhetstecken (=) och sedan det värde man önskar ge. Varje parameter kan åtskiljas med komma eller ett mellanslag. Om inga parametrar anges visas nuvarande värden på de aktuella parametrarna. Värden anges med decimaltal.

Några exempel

```
>>>PRTSET
>>>PRTSET,DP=65,EJ=8,LM=5
```

Det första exemplet visar nuvarande värden på parametrarna.
I nästa rad sätts sidlängden (DP) till 65 rader, antalet rader som matas fram vid sidbyte (om man inte använder Formfeed) slutligen sätts vänster marginal (LM) till 5 positioner.

Parametrarna som kan programmeras är:

DP=hh DePth count
Parametern talar om hur många rader som skrivs per sida. Sätts DP=0 så antages ett utgångsvärde.

EJ=hh EJect count
Den här parametern beskriver hur många tommma rader som ska matas fram i slutet av varje sida. Om EJ=0 matas inga extra rader fram.

LM LeftMargin
Parametern används för att flytta fram startpositionen för vänsterkanten. Användbart för att slippa få uskrifter som skriver i hålen i de fall man har färdighålat papper.

SYSTEMBESKRIVNING - MJUKVARA

RTCSET

RTCSET är ett kommando om används för att ställa tid och kalender i klockan på cpu-kortet

Syntaxen för kommando RTCSET är:

```
RTCSET
```

Programmet startar och man får besvara frågor för att ställa tiden och datum.

Exempel

```
>>>RTCSET

WEEK NR   .: 53        ange vecko nr t.ex 53
DAY       .: 01        ange veckodag Måndag=1 Tisdag=2 osv...
YYMMDD    .: 880104    ange ÅR MÅNAD DAG t.ex 880104
HHMMSS    .: 192505    ange tiden TIMMAR MINUTER SEKUNDER i och att
                       ental sekunder ställs klockan och programmet
                       avslutas.
>>>
```

OBS ! dag anges med 2 siffror där 01 motsvarar Måndag
och 07 Söndag
<CR> används ej !!

STAT

STAT är ett kommando som när det exekveras talar om status för hela systemet. Status beskrivs för alla diskar, printerbuffert, parametrarna för printern, memend, terminalen, dispostionen av minnesbankerna och en timer i systemet.

Syntaxen för kommando STAT är:

```
STAT
```

Programmet startar:

1. diskarnas status testas och finns det flexskivor i läses namnet på volymerna liksom återstående fria sektorer.
2. Kvarvarande utrymme i printerbufferten och hur många filer den innehåller presenteras.
3. aktuella parametrar för printer och terminal visas.
4. rapport om hur minnet är disponerat.

Det finns tre kommandon som STAT kan tolka:
<F> Ta bort första filen i bufferten.
<C> Ta bort alla filer i bufferten.
<E> återgå till FLEX.

Här ett exempel på hur det kan se ut på skärmen.

```
05-01-88  * DC SYSTEM STATUS * 13:31:54

DRIVE STATUS:                          PRINTER STATUS:

DRIVE §0 DCPUSYS §2                    SIZE = 128 FILE = 1 LEFT = 127
162 SECTORS FREE                       PRINTER DP=0 EJ=8 LM=2
0=D0  SYSTEM


                                       SYSTEM STATUS:
DRIVE §1 PD 68K §1
1669 SECTORS FREE                      MEMEND $BFFF
1=D1                                   TERMINAL DP=0 WD=0 EJ=0 ESC=$20 PS=Y
                                       BANK SELECT MEMORY    = $00 - $0D
DRIVE §2  VIRTUAL-.MEM §0              BANK SELECT PRINTER   = $DF - $FE
3138 SECTORS FREE                      BANK SELECT VDISK     = $10 - $DE
2=V0  WORK                             RESERVED FREE BANKS   = NONE

VDISK 3296 SECTORS                     SYS. TIMER = 00:00:14

F = CLEAR FIRST FILE IN PRINTER BUFFER

C = CLEAR PRINTER BUFFER                              E = EXIT

<C>_
```

OBS STAT godkänner bara stora bokstäver.

VDISK

VDISK är kommandot för att formattera den virtuella disken. Vidare kan man definera WORK DRIVE och SYSTEM DRIVE varför kommandot ersätter ASN.

Syntaxen för kommando VDISK är:

VDISKÄ,<option list>Å

Skrivs endast VDISK följt av <CR> så får man hälpfunktion över VDISK. VDISK åtföljt av ett mellanslag och <CR> listar nuvarande värden på VDISK:s parametrar. Optioner som kan ges är:

F Formattera den virtuella disken med default printerbuffert 12K

D Sätt defaultvärden på diskarna 0=D0 1=D1 2=V0 S=0 W=1

L Skriv inte ut status på skärmen

Pxx Sätt printerbufferten till xxKB storlek.

Y Besvara frågan jakande om disken ska formatteras.

R Reservera 4KB RAM i toppen av minnet

C Ta bort reservation av minne som option R gjorde

S Sätt alla diskar som SYSTEM DRIVE

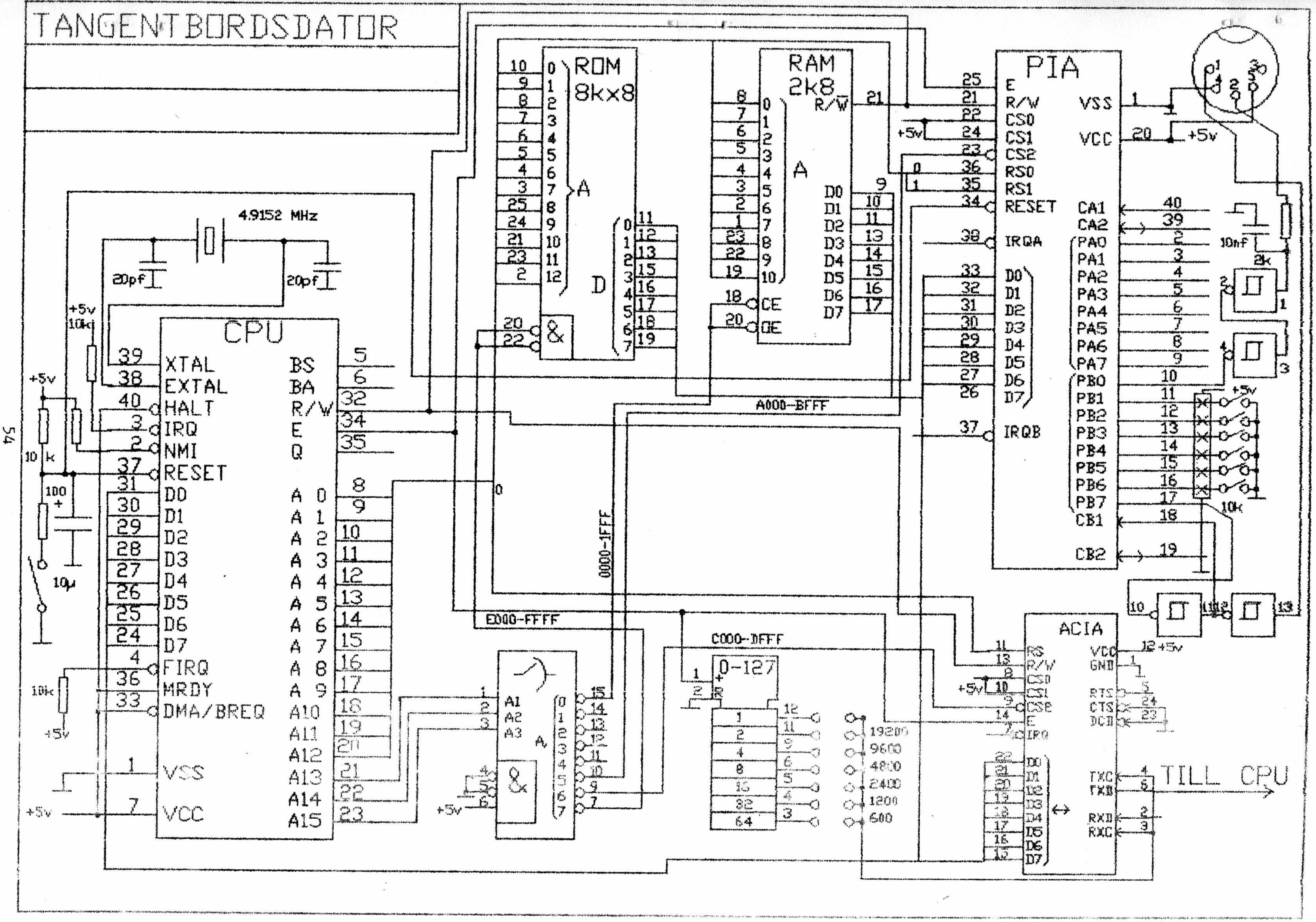
W Sätt alla diskar som WORK DRIVE

2W Sätt disk 2 till WORK DRIVE

1S Sätt disk 1 till SYSTEM DRIVE

Exempel:

```
>>>VDISK,0=V0,1=D0,2=D1  Sätt virtuella disken till disk 0
                         Sätt disk 0 till disk 1
                         Sätt disk 1 till disk 2

>>>VDISK <CR>            VDISK följt av mellanslag och vagnretur
                         ger aktuella värden

DC-VIRTUAL DISK §2

VDISK 3296 SECTORS

128K PRINTER BUFFER

0=D0  S                  disk 0 är SYSTEM DRIVE
1=D1  W                  disk 1 är WORK DRIVE
2=V0
```


TANGENTBORDSDATOR
ROM
8kx8
RAM
2k8
PIA
CPU
ACIA
4.9152 MHz
20pf
20pf
+5v
10k
10 k
100
10µ
10nf
2k
A000-BFFF
0000-1FFF
E000-FFFF
C000-DFFF
0-127
19200
9600
4800
2400
1200
600
TILL CPU