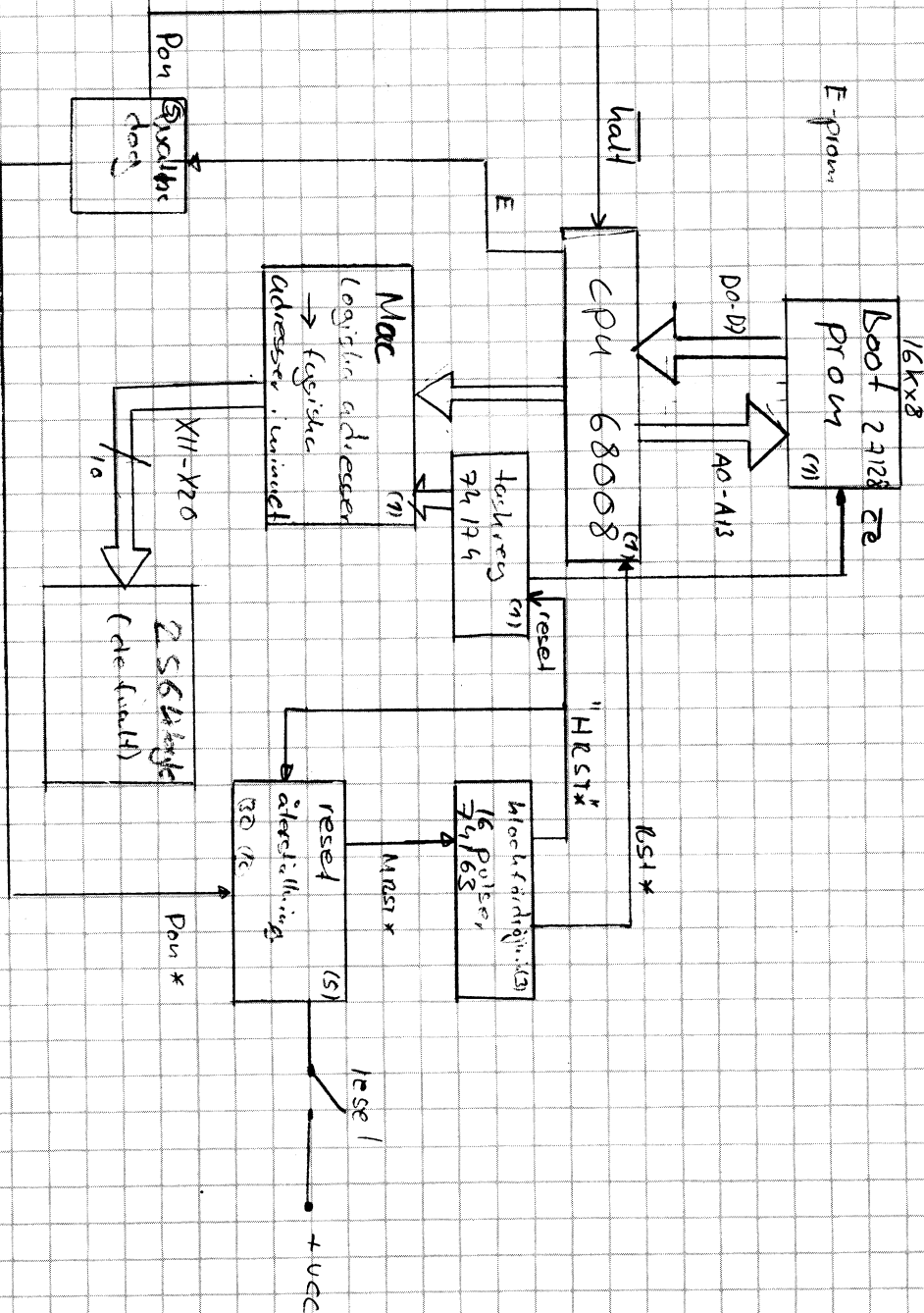


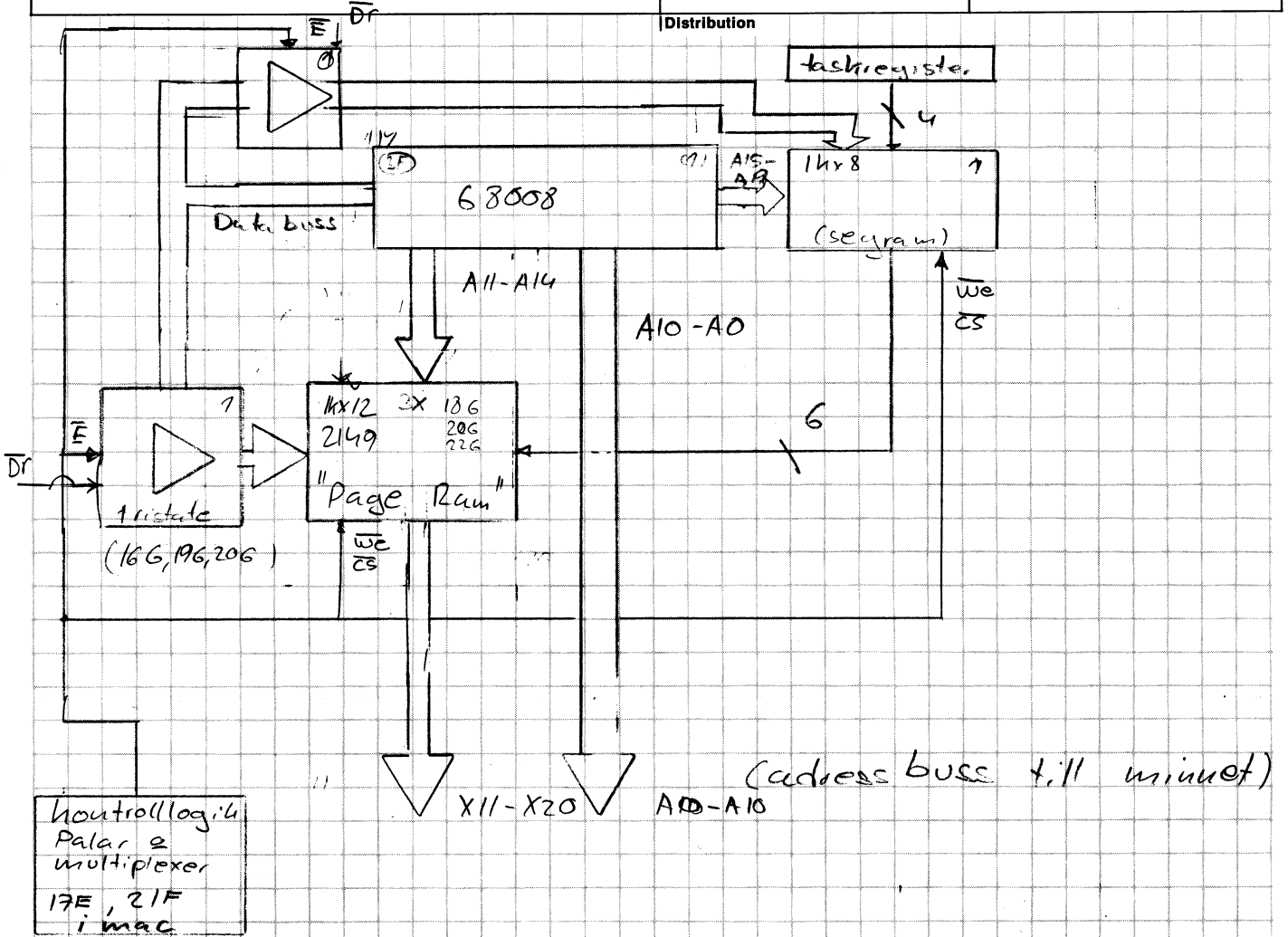
Från/From	Datum/Date 861114	Beteckning/Reference HH
Rubrik, ärende/Subject	Ert datum/Your date	Er beteckning/Your reference
Blockschema dator X25	Gäller fr o m/Effective date	Ersätter/Replaces

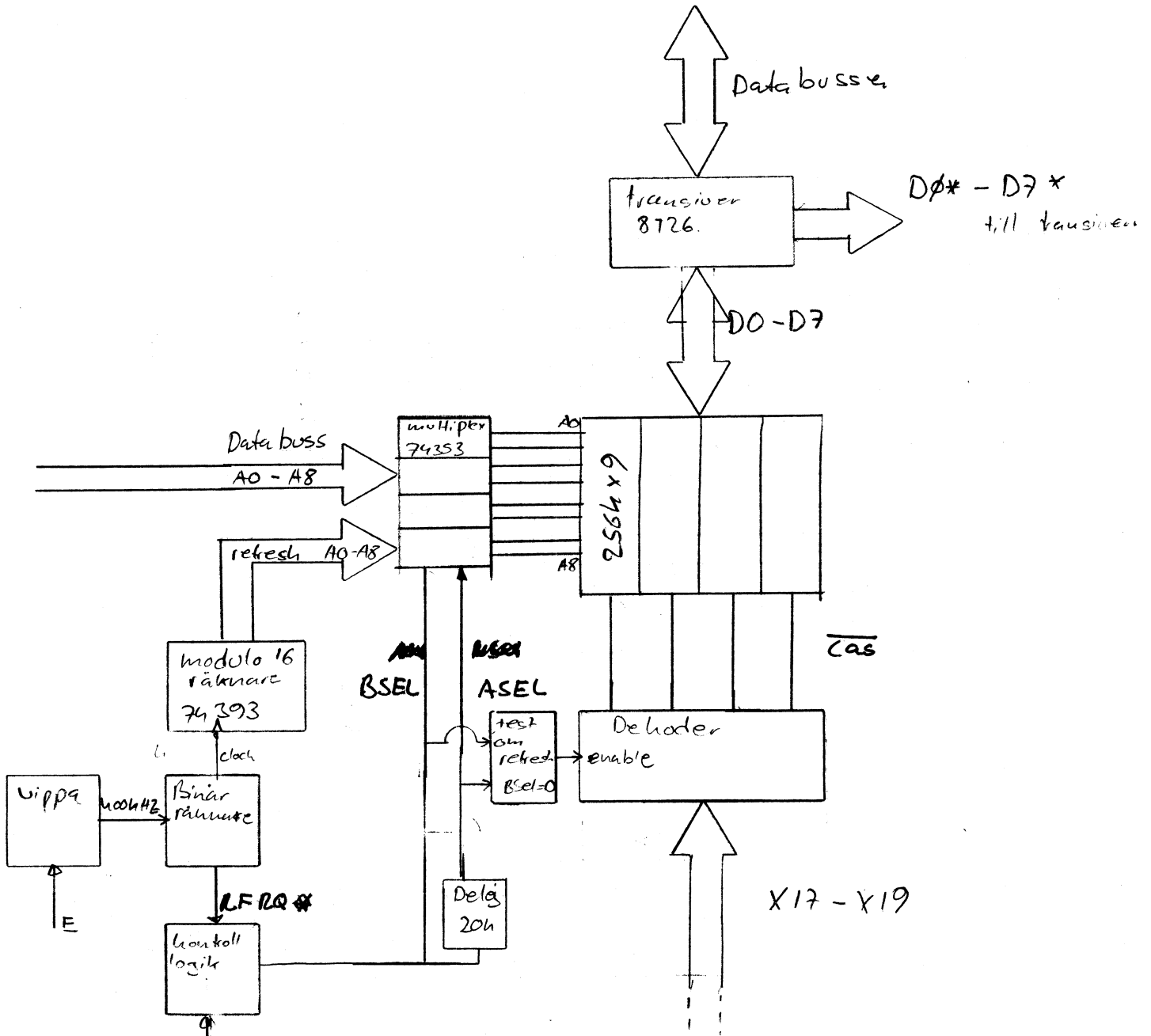
Distribution

— data
— adress



Från/From	Datum/Date	Beteckning/Reference HH
Rubrik, ärende/Subject X35 Mac	Ert datum/Your date	Er beteckning/Your reference
	Gäller fr o m/Effective date	Ersätter/Replaces

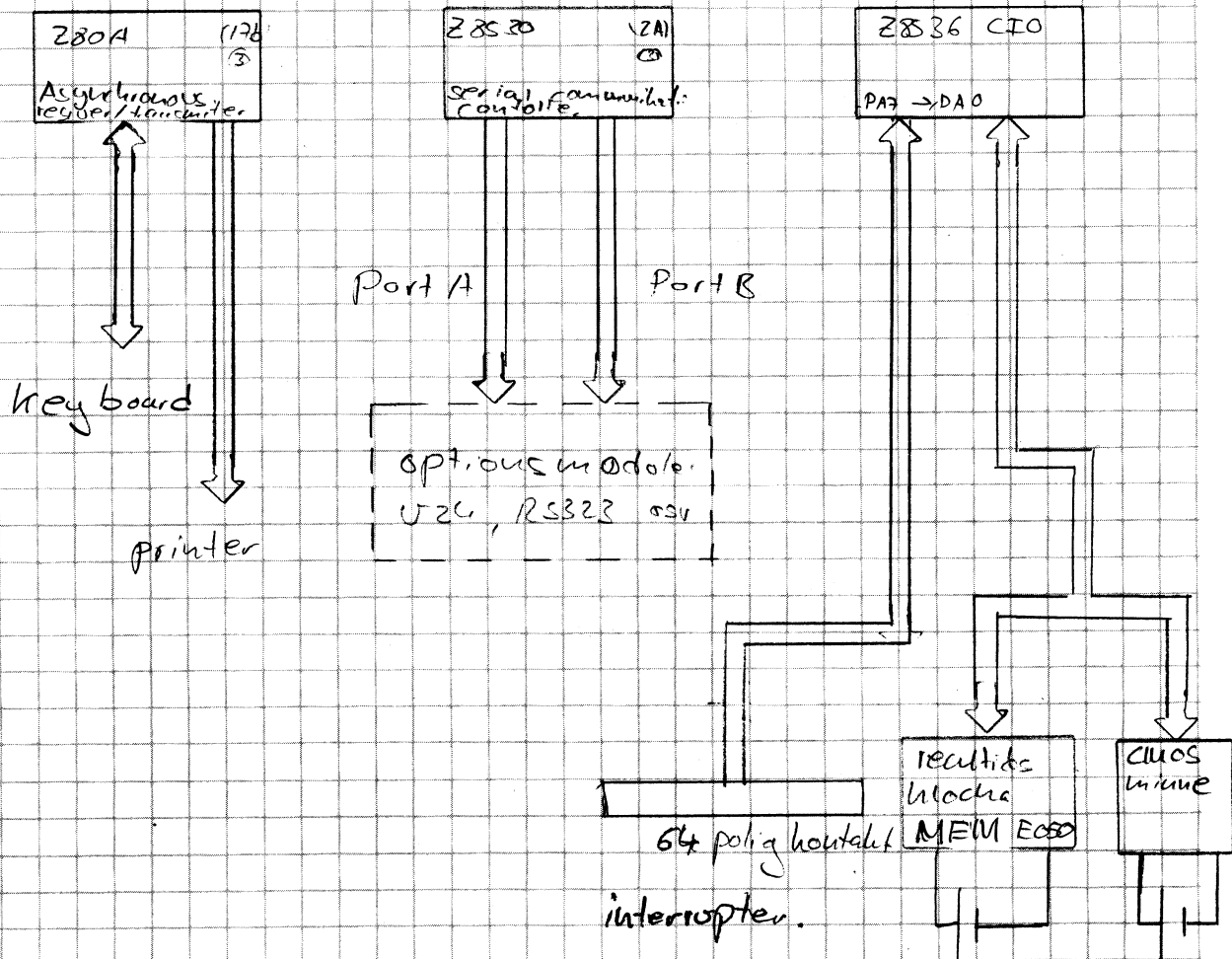




\bar{AS} = address stobe
angevner att datorn vill skriva eller läsa

Från/From	Datum/Date	Beteckning/Reference
Rubrik, ärende/Subject	Ert datum/Your date	Er beteckning/Your reference
X35 serial 1/0	Gäller fr o m/Effective date	Ersätter/Replaces

Distribution



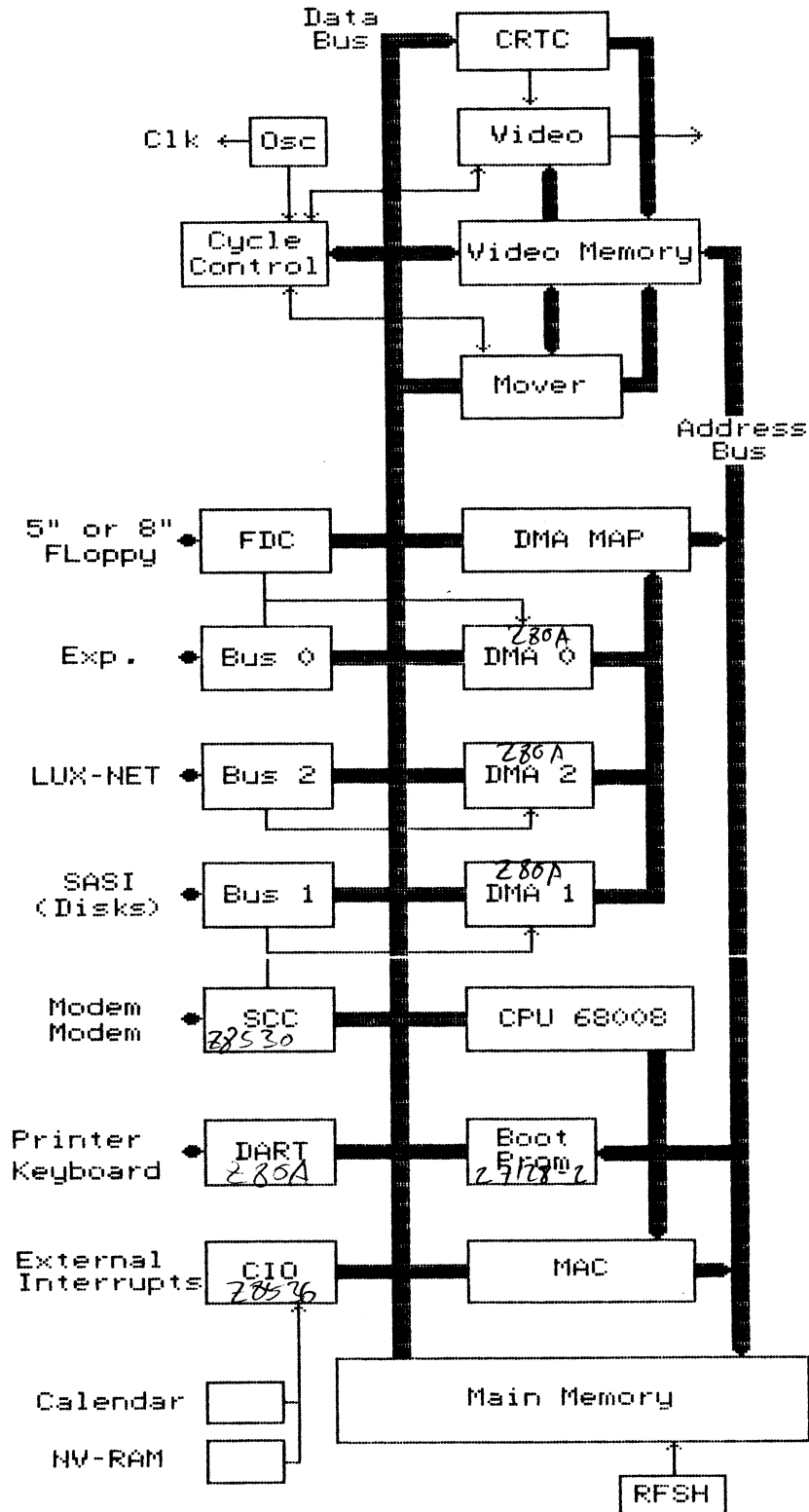


Från/From	Datum/Date	Beteckning/Reference
Rubrik, ärende/Subject	Ert datum/Your date	Er beteckning/Your reference
X-35 Bus block	Gäller fr o m/Effective date	Ersätter/Replaces
	Distribution	

2. Integrated Computer Hardware.

2.1 General Description

Two circuit boards contain all functions for computer and and graphic-generator. They are closely connected inside the X35 Basic-Unit.



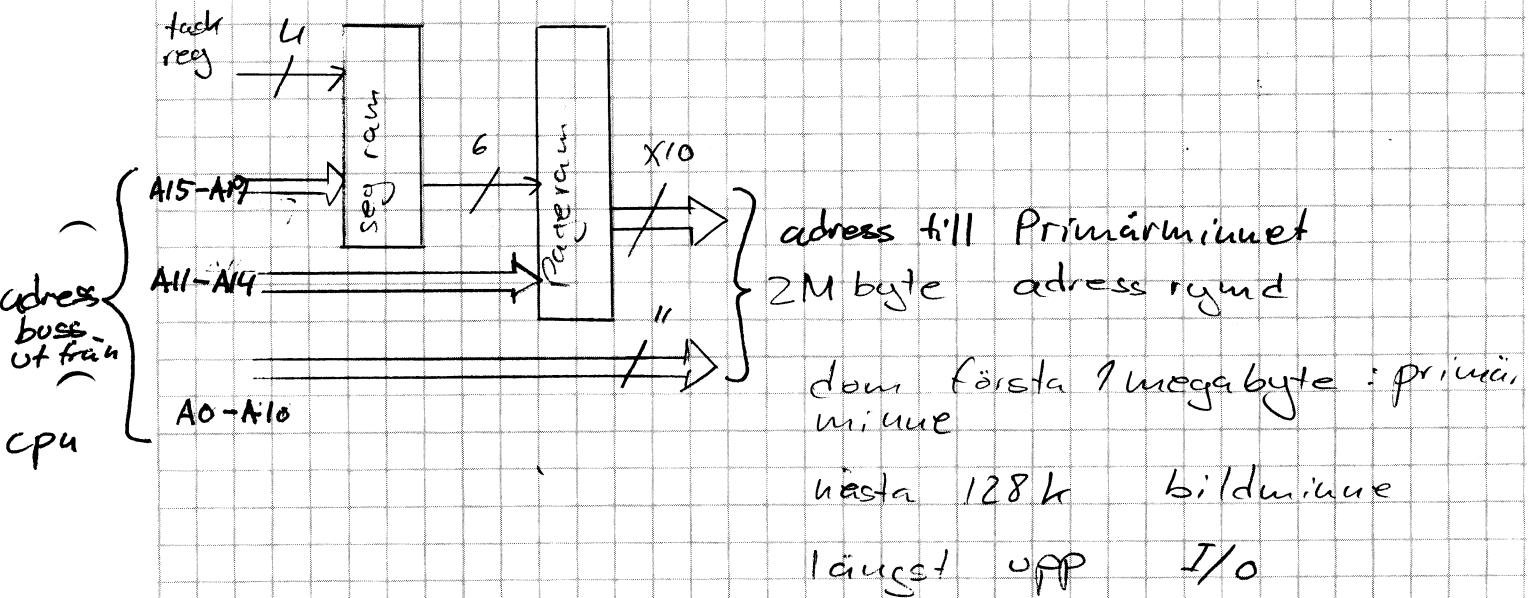
Pic 4. Block Diagram over the X35 CPU including the Graphics controller and video memory.

Från/From	Datum/Date 841114	Beteckning/Reference HH
Rubrik, ärende/Subject Mac X35	Ert datum/Your date	Er beteckning/Your reference
	Gäller fr o m/Effective date	Ersätter/Replaces

Distribution

memory Access controller Mac

- sköter minneskydd dvs man skall inte kunna skriva i gammal minnesutrymme
- omvandlar logiska adresser till fysiska adresser.
- mac kan programmeras att ge varje process en unik kod & data areor



(*task register*) talar om hur dom logiska adresserna från cpu skall bli omvandlade till fysiska adresser. 0-15 processer
 bit 6 talar om när Booten skall kopplas ih
 bit 7 talar om Magic (enable user data segment)
 taskregistret kan man bara skriva i (write-only memory)

Från/From	Datum/Date 841114	Beteckning/Reference HH
Rubrik, ärende/Subject X-35	Ert datum/Your date	Er beteckning/Your reference
	Gäller fr o m/Effective date	Ersätter/Replaces

Distribution

taskregistret nollställs med signalen

"HRST" som kommer från blad 3.

Alla utgångar blir låga när HRST* är låg, sker vid
buvudreset.

räkarna på blad 3 "74163" ger en

signal "HRST" som är ungefär 16 clockpulser låg.

Denna räkare triggas av (delvis)

signalen MRST som kommer från buvud

reseten, som ligger på blad 5 o

kommer från (3C). Den reseten nollställs

automatiskt av signalen HRST efter 16

clockpulser.

När "reset signal" nollställer taskregistret

görs "select enable" på "boot prommet", som börjas

ex. leveras, och sätta upp "macen" "transparent"

(*Mac*) Page rammet (17G) (22G) (20G)

adresser av Adressbuss A11-A14 samt

övriga 6 adress pinna från Seg Ram.

Minnes innehållerna i dessa läggs ut som

en fysisk adressbuss X11-X20 (till primärminnet)

Segment Rammet (17G, 15G) får sina adressbussar

utpekade av A15-A19 samt "taskregistret"

taskregistret pekar ut vilket process nummer som

gäller (dom 4 lägsta bitarna 0-3) (0-15 processer

Från/From	Datum/Date	Beteckning/Reference
Rubrik, ärende/Subject	Ert datum/Your date	Er beteckning/Your reference
X-35	Gäller fr o m/Effective date	Ersätter/Replaces

Distribution

segment rammet adressera
 Gst adressivejningar till Pagerammet.

När det skall skrivas i segment o Page rammet används det "tristate" på Databusen (D7-D0) som läser från databussen till segment rammet.

Tristaterna kan bara stå i en riktning
 (från data busen → pagerammet)

Dessa "tristate" o " \overline{we} ". till page och segment rammen styrs av

①: Page ram + : 2/1 multiplexern dess tristate till dessa selekt chip styrs av A_0 , samt för insigueler från Pal (17E) (\overline{we} , \overline{cs} på ram o tristate)

②: segment ram + tristate till dessa : direkt från palen. (17E) (\overline{we} , \overline{cs} på ram o tristate riktningar på tristaten kommer från R/W^*) epu

Under uppstartande: ① taskregistret nollställs
 ① Boot programmet hopplas in. Berouda på adresserna A_0 till Palen 1618 genereras \overline{cs} \overline{we} \overline{E} , \overline{cs}_{page} till segmentrammen samt 2/1s multiplexen 74158 gör chip selekt på pagerammet. A_1 A_2 A_19

Från/From	Datum/Date 7/116	Beteckning/Reference H ²
Rubrik, ärende/Subject X35 Memory	Ert datum/Your date	Er beteckning/Your reference
	Gäller fr o m/Effective date	Ersätter/Replaces

Distribution

Adresser i det dynamiska primärminnet delas upp i Page (sidor) som pekas ut av Pageramet X20-X11 som sätts upp av "Maceu" (minnes access kontroll.). Olika processnummer 0-15 har olika sidor att jobba på. Inom varje sida pekas adressen ut av adressbussen A10-A0 direkt.

Memory ritningen används 74137 (deloder) (27E) för att välja var minnesbetsarna sitter på kortet.

Storleken på primärminnet bestäms av bygglingen till NOR grunden 74260 (26G) samt till väntan på ritningen.

X18 & X19 adresserar dom två sista 256k

X20 används för I/O utgångarna se krets 22E memory X35

Minnet är uppbyggt av 256k x 1 bits kretsar

Totalt 9st sådana deliäs för att bidra

ett 256 x 9 bit minnes area (9 biten paritet)

Det sådana minnes areor ger 1M bits primär minne.

Write enable på Primärminnet

styras av multiplexern 74352 (26F)

Som selectas med signalen MEMW (7)

för att multiplexern skall väljs den signalen måste BSEL vara hög (7) samt 74175 (27F) triggas

MEMW } ⇒ read Primärminne
BSEL
ASEL

om BSEL = 0 görs ej Enable A på multiplexern

row address strobe \overline{RAS} läggs ut av multiplexern 74352 (26F)

den signalen fördröjs sedan i Digital delay line som sedan genererar \overline{CAS} genom 1-8 dekodern 74137 (27E) (selectar vilket minnesarea som skall

kopplas in

Adresserna till primärminnet kommer från

Adressbussen A0-A10 som går igenom multiplexern

74353 (27E-25E) multiplexarna styrs ut av

BSEL $\underline{\text{O}}$ ASEL

BSEL = 0 ⇒ refresh om ASEL = 1

BSEL = 1 $\underline{\text{O}}$ ASEL = 1 ⇒ minnesaccess CPU

Vår en refresh cykel skall göras håller vi

binär räkaren 74163 (23G) reda på

(den klockas med 100kHz) Den signalen delas med 3 och skickas till

modulo 16 räkaren 74393 (23F) samt

genereras en $RFRQ^*$ (refresh quest) av

sk-vippa 7474 (24G). Den signalen går till

74175 (4st flipflop) (24F) som testar om det

finns ledig tid för att refresh minnet.

Vilket åstadkommer att $BSEL=0$ \Rightarrow $ASel=1$

modulo 16 räkaren definiera de vilken adress

som skall refreshas. Dvs alla adresser

refreshas inte direkt. Den tar en val i taget.

signalen AS^* sänder datorn ut när korrekt

adressbuss ligger lågt på adressbussen.

$D0^* - D7^*$ läggs in på task-Registret

för att peka ut Process numret om signalen

WCC klockas in, genereras från per 17E.

NV-RAM Control (Non Volatile Parameter 3,8 Memory)

Är ett CMOS-minne som innehåller information om nuvarande "uppsättning" av hårdvara & mjukvara, samt styr NV-RAM + Calendar clock.

CMOS minnet har batteribackupp (samma som realtidsklockan).

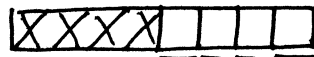
Minnet är av seriell typ & kopplat till Counter / Timer / Parallel I/O unit 28536 CIO

PC0 → PC4.

Ger innehåller NV-RAM och i friden vi skall ladda systemmjukvaran (floppy, winchester, Lux-Med etc)

Ligger på adress 1FF70FH Läger 0 i

4-biten



1FF70FH

Clock 0

Data 0

Calendar CS*

NV RAM chip select
0 Selected
1 Passiv

Minnet består av tvåets NMC 9306, som är ett seriellt EEPROM med seriel in och ut port

(* reelltids lås *)

för att kunna ha någon reelltids lås används en låskrets MEM F050-16 (ritning 4A) med batteri backup.

minnescellen 1FF70F H ger \overline{CS} på låskretsa.
Minnescellen 1FF70F H är NU-RAMET, beskrivs på nästa blad.

(* DMA kanaler 3st *)

Används för snabb IO-minnes överföring som disk & nätverk. Det finns 3 DMA som kan kopplas ihop 5 möjliga DMA-portar.

1. floppy disk controller

2. 3 externa bussar. interface

3. SCC, serial communications controller Z8530

DMA använder sig av fysiska adresser

(ges av macen).

DMA ADDR förs från DMA MAP ritning 7

ej helt klart.

Från/From	Datum/Date	Beteckning/Reference H/H
Rubrik, ärende/Subject X35 "watchdog"	Er datum/Your date	Er beteckning/Your reference
	Gäller fr o m/Effective date	Ersätter/Replaces

Distribution

watchdog kollar så att datorn inte
hängen upp sig. Genererar en
... hårdvaru reset om "hunden" inte blir
aktiviserad var annan sekund

Aktiveringen blir gjord genom att
2/läsa i ^{skriv i} cause register (80007H).

DVS det registret måste läsas ut
med jämna mellanrum (≤ 2sek)
(registret sköter NMI-interrupter)

indata till "hunden" kommer på signalen
cause* @ E. utsignal ligger på signalen
Pon @ Pon*

signalen cause* genereras från Pden
som styr program @ Pagenam. DVS
läs vi läses i en fysisk minnesadress genereras
även cause signalen.

signalen Pon gör halt på CPU

signalen Pon* gör set på JS-k-vippan (18b)
som ger en Resetsignal (Rst*) på CPU!
samt ger en låg reset på HRST* På task
register (kopier i Boot program) samt latches in
adress busser, code, paritetet på data busser

Från/From	Datum/Date	Beteckning/Reference H41
Rubrik, ärende/Subject	Ert datum/Your date	Er beteckning/Your reference
X35 - reset. ritning 3	Gäller fr o m/Effective date	Ersätter/Replaces

Distribution

Vid ett fel (paritets) eller datorn har hängt sig (watchdog gör reset)

läggs - adresse.nc, X16-X20 + code + paritets bit ut på

data bussen. Funktionen är att ett 8 ords bitregister 14E (3)

skiftar in X20-X16, code, paritets bit samt skiftar ut det på Data busse genom tri state 15E (3).

Cytdioden indikerar att det har skett ett paritetsfel. (ligger i burken).

Tristaterns öppnas vid varje signal cause (kommer från palarna i Pageinnet.)

signalen Pout skiftar in oavstämde bitar till tristaten, samt gör reset på kretsen.

Ovan används för att kunna läsa av vilket block felet har inträffat. (utpekas av X16-X20)

DVS Causeregistret är ett 8 bits register 74278 kret

Från/From	Datum/Date	Beteckning/Reference 74
Rubrik, ärende/Subject X35 serial I/O	Ert datum/Your date	Er beteckning/Your reference
	Gäller fr o m/Effective date	Ersätter/Replaces

Distribution

Det finns två möjligheter att ansluta tangentbordet till X35.

- ① Direkt till X35 enheten via Z80A Datt
- ② via display monitorn

Baudraten väljs från tangentbordet.

(Z80A Datt är en Asynchronous Receiver

Även printern ansluts till Z80A Datt kretsen. Standarden är U24 snitt. Baudrate ställs med CIO

till ingången Z80B Datt ansluts monitorn (grafikavdelningen)

Kommunikationen sker genom Z8530 ritning? (serial communication controller)

Det finns två seriella portar att kommunicera med. Man måste koppla på option modules (tex U24 (RS232), RS422 eller modem) för att kommunicera utåt

Klockfrekvens från grafikkortet X35 841116

På ritning 5 finns kretsarna som delar ner klockfrekvensen från grafik kortet

signalen som kommer från grafik kortet heter REFCLK* 8MHz, samt en klocksignal 2*CLK på 16MHz som delas ner till

4MHz och läggs ut på signal CLK.

Även denna signal delas med 2 till 2MHz

och läggs ut på signal PCLK

Dessa klocksignaler kan vi även få i inverterat pulståg.

2MHz läggs även ut på Buskorten som BPCLK* \approx BPCLK

X35-RUS

841919

H²

på ritning 4A har vi counter/timer
and parallel I/O-unit. Till I/O-ingångarna

PA₀-PA₇ hopplas interrupt signaler från

BUS ₀ →	interruptsignal I7	interrupt namn INT*	anm Buss 0 intr.
BUS ₀ →	I2	INTT2 *	Buss 0 extern
	I3	INTT3 *	
	I4	INTT4 *	
	I5	INTT5 *	
	I6	INT*	
BUS 1	I1	namnlös	Buss 2 inre
BUS 2	I0		Buss 1 inre

Till I/O-ingångarna PC₀ → PC₃ ä.
kalendern kopplad, krets MEM F050-16 med
batteri backup.

Paritetsbit nullställning

$$X12 = 1$$

$$X11 = 1$$

$$A10 = 1$$

$$A9 = 1$$

$$A8 = 0$$

X35 buss ritning

biten PARTST skall sättas med data

GC om paritetsbiten skall skrivas felaktigt

- samt paritetstest påverkar ej programmet (bra vid test av paritetsminnes kretsarna)

- kretsen 3E skall få en signal från 146
har rätt adress är utlaggd

ADRESSEN ÄR 7FE00

till special control register.

84119 H-

Nad vi kommer åt genom anslutningsdonen

Z8530 : serial communication controller

Databussen D ϕ -D7 seriellt.

Kontakterna är dom två vertikala
RS232/V24 (sina kort)

Z80A Dart: Asynchronous Receiver/Transmitter

Databussen D ϕ -D7

Kanal A: Printeranslutningen (högra 25 poliga kontakt)
Printer

Kanal B: Display/console anslutning (vänstra 15 poliga kontakt)
Display/console

+ KEY BOARD (7 pole Din 180°)

Z8536: interrupt signalerna från
BUS ϕ i
BUS ϕ x
BUS 1
BUS 2

som tas genom dom "vertikala kontakterna"
vilka insignalerna är se på beskrivning X35-BUS

X35 - BUS

"Expansion slots"

BUS 2	BUS 0 external
BUS 1	BUS 0 inter.

BUS 0 inter. alla Data D0-D7 \Leftrightarrow Data busse

BUS 0 ext. alla Data D0-D7

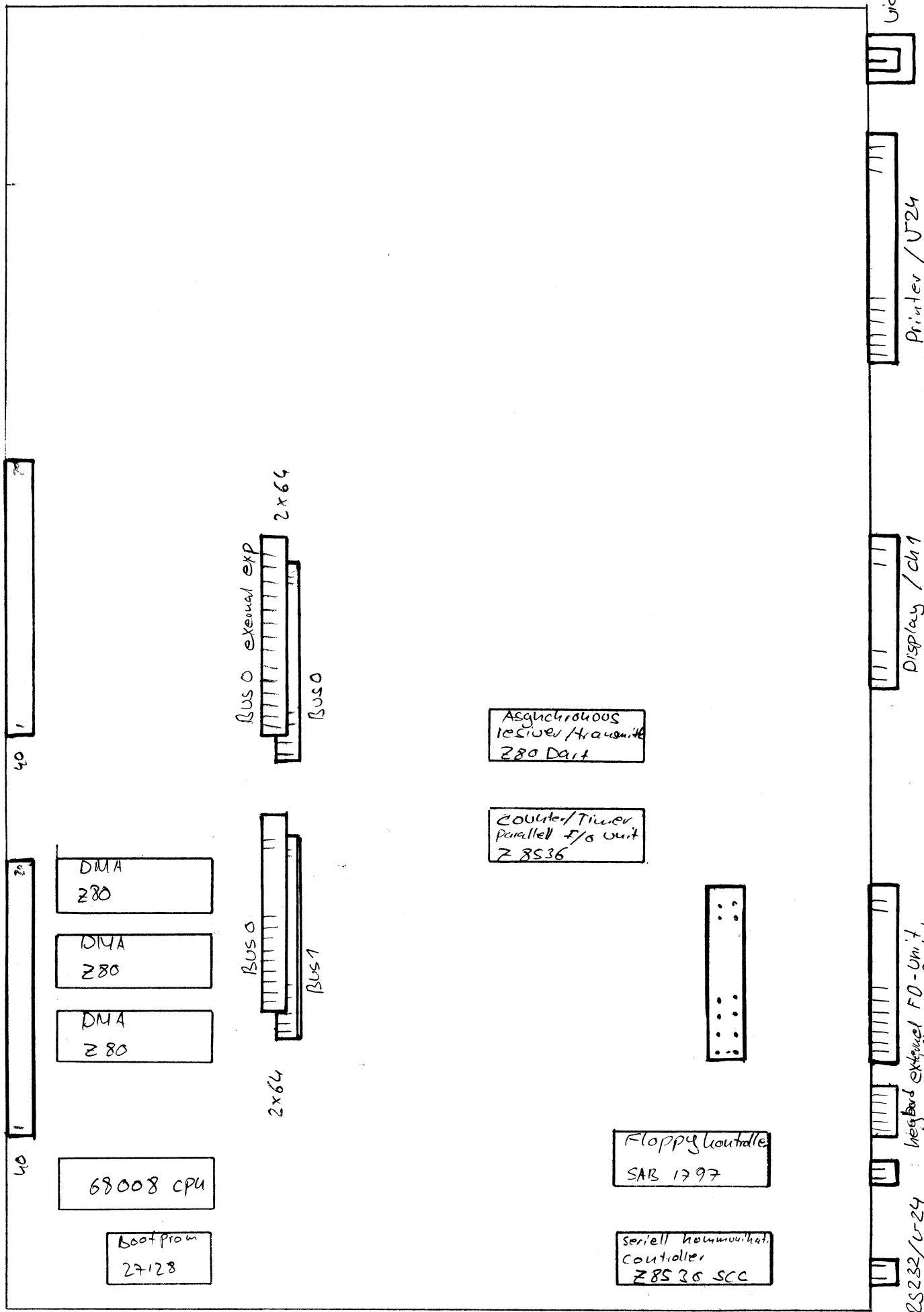
BUS 1 alla Data D0-D7

Bus addresser A0-A4

w/r*

huvudlaster +
ut/in-kontakter.

videokort)
↑

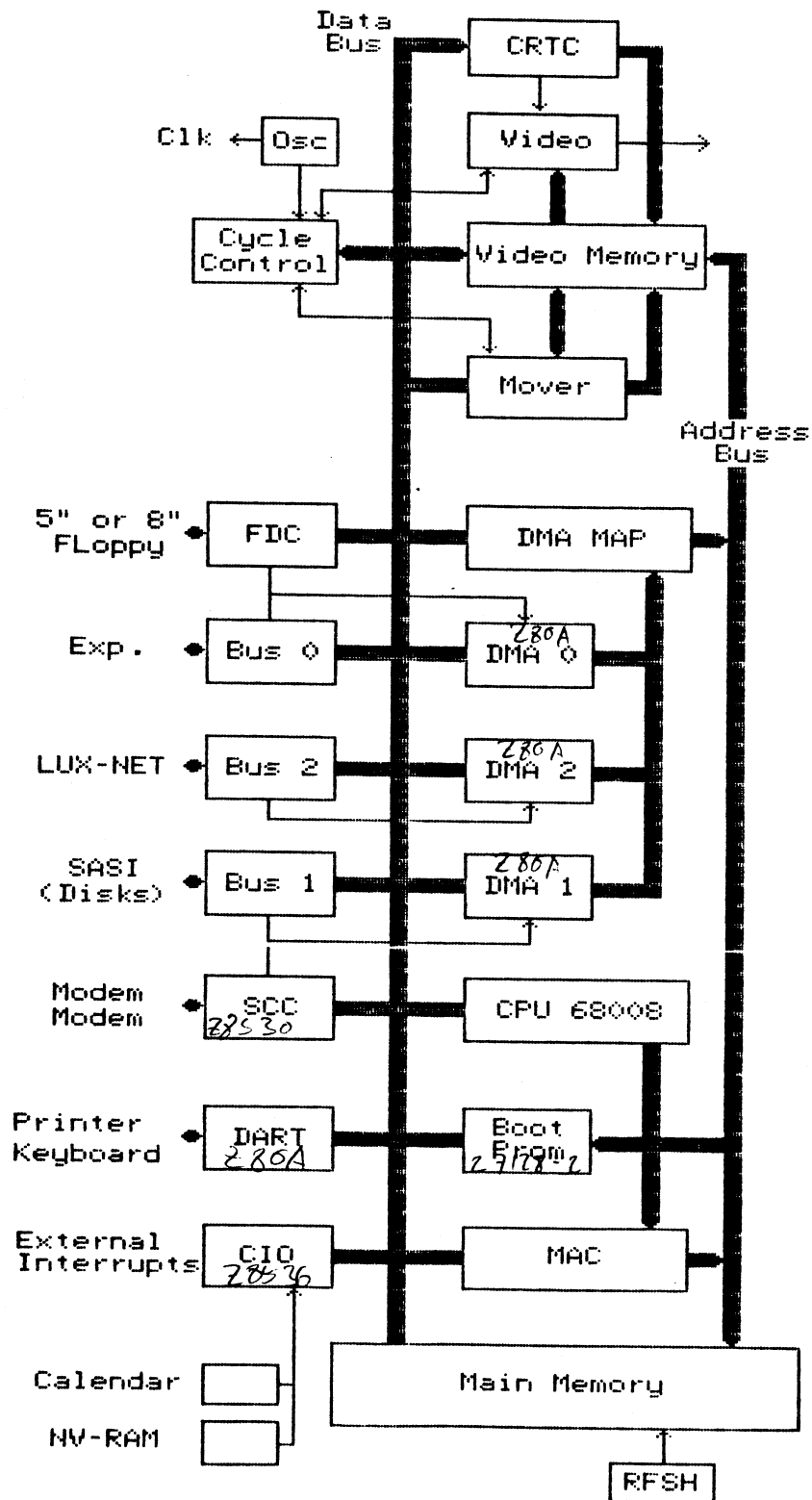


RS232/C-24 : keyboard external F0-unit,
floppy Disk

2. Integrated Computer Hardware.

2.1 General Description

Two circuit boards contain all functions for computer and and graphic-generator. They are closely connected inside the X35 Basic-Unit.



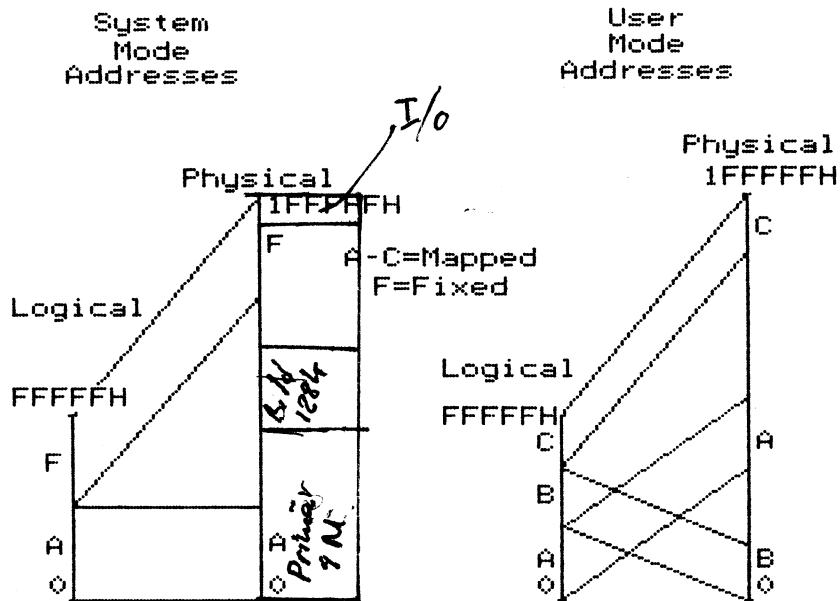
Pic 4. Block Diagram over the X35 CPU including the Graphics controller and video memory.

2.2 CPU

The microprocessor used is a Motorola 68008, which is a 32-bit processor with an 8-bit bus. The clock frequency is 8Mhz. The 68008 is software compatible with other processors in the 68000 family. The resources available to the user consist of the following:

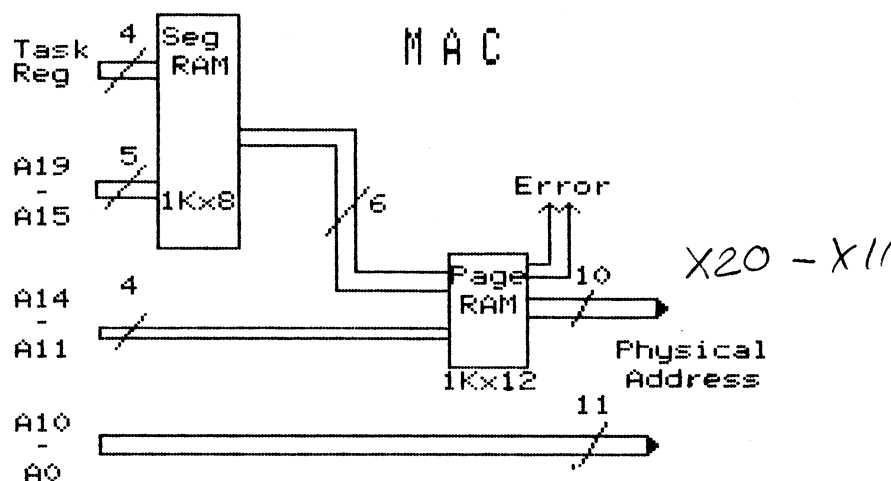
- * 17 32-bit data and address registers
- * 56 basic instruction types
- * Extensive Exception Processing
- * Memory mapped IO
- * 14 addressing modes
- * Complete code compatibility with the 68000

Addressing: There are two main addressing modes from hardware point of view: System (supervisor) mode and User mode. In the System mode the 1Mb address space is divided in two areas, RAM space from 0 to 512kb, and IO-space from 512k to 1Mb. The RAM is addressed through the MAC (so mapping to any physical memory areas can be made) but the IO-space is addressed directly (MAC itself and the task register)



2.3 Memory Access Controller (MAC).

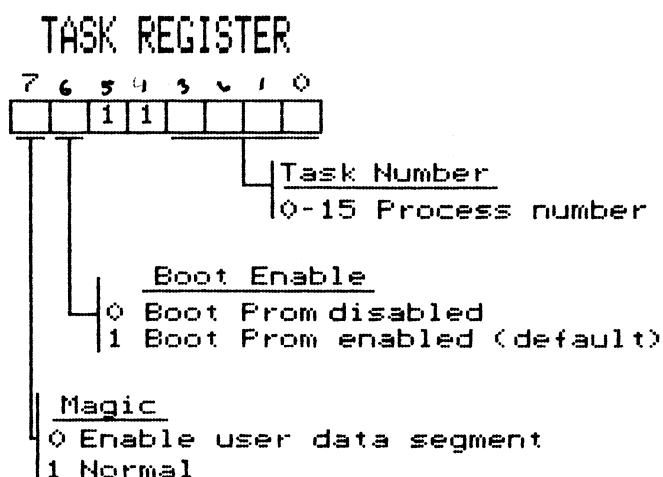
The MAC provides memory protection between user processes and between users and operating system. It translates logical addresses into physical memory addresses. The MAC can be programmed to give each process its own code and data areas. The smallest physical partitions of memory are 2k byte.



The MAC is only possible to set up in supervisor mode. As it is impossible to access the main memory without proper values in the MAC it is tested and set up immediately after reset or power up by the Boot-PROM-program. The Boot-PROM is addressed directly from CPU-address bus and does not rely on MAC-values.

2.3.1 Task-Register

To inform the MAC how the logical addresses from the CPU is to be translated to physical addresses there is a Task-Register. The Task Register is accessed in the logical address space and is a write-only register.



2.3.2 MAC Segment RAM

The Segment RAM is a 1k x 8bit memory. It uses four bits from the Task-Register and five bits from CPU address bus (A15 through A19) as input address. Six output ~~bits~~ ^{IC} bits are used to address the MAC Page RAM.

2.3.3 MAC Page RAM

A14) as input address. Ten output bits are used to form the high part of the physical address bus (X11 through X20). The low part (X0 - X10) comes from the CPU.

2.4 Bootstrap Loader

The Bootstrap loader is a 16k x 8 bit FROM. It performs all necessary cold-start functions such as hardware initialization and hardware checks. It is possible to disable and enable the PROM through setting a bit in the Task-Register. A functional description over the Bootstrap-loader is found in appendix A.

2.5 Random Access Memory

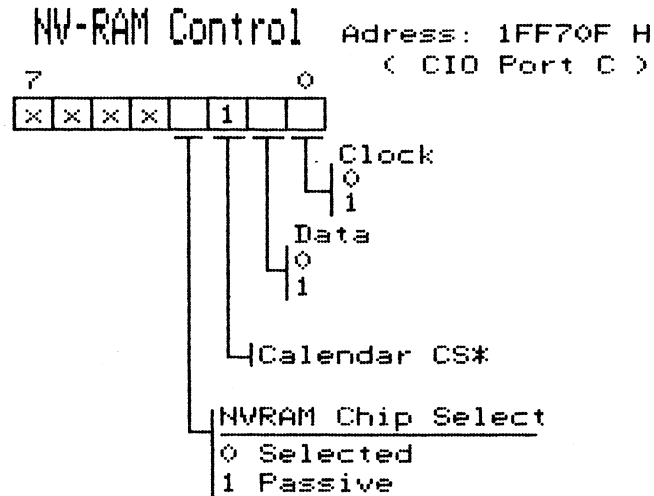
The main memory consists of 512kbyte dynamic RAM with parity check. It is possible to expand memory size to 1Mbyte ~~without~~ *without adding any new circuit board. by adding memory components to the main circuit board.*

2.6 Non Volatile Parameter Memory

The parameter memory is a battery-backed up CMOS memory which is used to hold information about the actual configuration of hardware and software. Example of information type in the NVRAM is where to load system software from on power up (floppy disk or whichester or LUX-NET...).

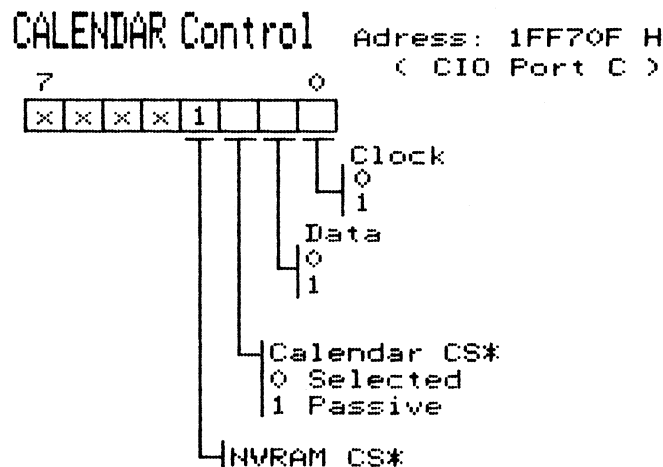
The NVRAM is accessed through the CIO (Counter-IO) module at address 1FF70FH with the following bit pattern:

(See Appendix A for NVRAM Content)



2.7 Calendar

The Calendar is a CMOS battery driven absolute time counter. It counts seconds, minutes, hours, date of the month, day of the week, month and year. Every 4th year, February has 29 days.



through the single data line. After the Chip select has been activated a serial 3-bit address word will first be accepted by the chip to select particular timing information. A fourth bit selects read/write mode, thus specifying whether the selected timing information should be written into the circuit or read from it. When chip select is high the data line has high impedance, neither accepting nor giving out data. For complete information about the calendar chip programming, see the data sheet of "Microelectronic Marin, MEM E050-16". As an example a complete data readout will

be done if the following sequence is given:

1,1,1,1,hour,min,date,month,year,day of week,sec

Each of the time fields are given in 8bit sequences and a complete transfer thus requires $4+7*8=60$ pulses on the clock line. There are also commands for readout of each time field separately.

2.8 Interrupt System

The 68008 processor has three priority levels of interrupts. In the X35 they are used in the following way:

```
00 normal
01 CIO interrupts. (Bus interface, floppy disk...)
10 DART, SCC
11 Power fail , NMI
```

The third level is the NMI (Non Maskable Interrupt). Lower level interrupts can be interrupted by higher level interrupts.

Each interrupt source supply a vector which is memory locations from which the processor fetches the address of a routine which will handle the interrupt. The CIO (Counter-ID)-component is programmed as a priority interrupt controller and supply the vectors from 64 through 102. Here is a table of the most important interrupts:

```
== Motorola defined vectors: ==
0,1 0000H Reset vector (8 bytes)
2   0008H Bus error
3   000CH Address Error
4   0010H Illegal instruction
5   0014H Zero divide
6   0018H CHK instruction
7   001CH TRAPV instruction
8   0020H Privilege Violation
9   0024H Trace

24  0060H Spurious interrupt

32-47 0080H TRAP instruction vectors

== User Interrupt Vectors ==
64   0100H Bus 2 interrupt
66   0108H Bus 1 interrupt
68   0110H Bus 0 xint*2
70   0118H Bus 0 xint*3
72   0120H Bus 0 xint*4
74   0128H Bus 0 xint*5
76   0130H Bus 0 ext.interrupt
78   0138H Bus 0 int.interrupt
80   0140H reserved
82   0148H Power fail/ok*
84   0150H reserved
86   0158H reserved
88   0160H MINT (mover interrupt)
90   0168H FREN* (Bus 0)
92   0170H FREN* (Bus 1)
94   0178H FINT (Floppy disk interrupt)
96   0180H Counter/Timer 3 (20ms)
98   0188H Counter/Timer 2
```

```

100 0190H Counter/Timer 1
102 019BH Counter/Timer error

```

```

== Dynamically allocated interrupts ==-

```

```

104 01A0H
106 01A8H
108 01B0H
110 01B8H
112 01C0H
114 01C8H
116 01D0H
118 01D8H
120 01E0H
122 01E8H
124 01F0H
   :   :
   :   :

```

2.9 DMA Channels

The DMA channels are used for fast IO-Memory transfers such as disk and network accesses. The X35 has 3 DMA channels that can serve 5 possible DMA-ports. These ports are floppydisk controller, 3 bus extension interfaces (4680 type option cards such as SASI-adapters for Winchester disks *and tape drives*).

The DMA channels operates with physical addresses wich means that the program has to calculate logic addresses into physical addresses. Program also have to set up the DMA-MAP registers:

DMA-controllers:

```

DMA 0      address:  1F F300H
DMA 1      address:  1F F400H
DMA 2      address:  1F F500H

```

DMA Map registers:

```

DMA-MAP 0  address  1F FD06H
DMA-MAP 1  address  1F FD04H
DMA-MAP 2  address  1F FD00H

```

Map registers: CPU Write Data		Output data	DMA-A15
-----		-----	-----
Databit	15	R/W*	0
	14	N A	0
	13	N A	0
	12	X20	0
	11	X19	0
	10	X18	0
	9	X17	0
	8	X16	0
	7	R/W*	1
	6	N A	1
	5	N A	1
	4	X20	1
	3	X19	1
	2	X16	1
	1	X15	1
	0	X16	1

During a DMA cycle, the physical bus X0-X20 get the low address (X0-X15) from the DMA-Controllers, and the high part of the address from the DMA-MAP registers as programmed before the DMA-transfer. The MAP-registers are programmed as 16-bit registers but the physical address generated depends on address bit A15 during a DMA-cycle as shown in the table above.

2.10 Counter IO and Interrupt controller

The Counter-IO (CIO) module consists of a Z8536 chip and handles interrupts, controls the NVRAM and Calendar interrupts. Below is a list of the usage of the different CIO-ports.

PA07	BUS 0	interrupt
PA16	BUS 0x	interrupt
PA25	BUS 0	XINT*5
PA34	BUS 0	XINT*4
PA43	BUS 0	XINT*3
PA52	BUS 0	XINT*2
PA61	BUS 1	interrupt
PA70	BUS 2	interrupt
PB07	FINT	(Floppy Interrupt)
PB16	PREN*	(Bus 0)
PB25	PREN*	(Bus 1)
PB34	MINT	(Mover interrupt)
PB43	-	reserved
PB52	-	" "
PB61	Power fail/ok*	
PB70	Printer baudrate out	
PC03	Parameter NVRAM CS	(NSC NMC9306)
PC12	Calendar Clock CS*	
PC21	Bidirectional Data	
PC30	Clock (Calendar and NVRAM)	

Z8536 CIO Port Addresses:

PA	1FF704H
PB	1FF702H
PC	1FF700H
Control port	1FF706H

For further details of the Z8536 operation see the Zilog Z8536 data sheet.

Cause Register.

The Cause register holds information of the cause of a NMI-interrupt. The possible NMI-sources are:

Bit 0	<i>rst BUT: on reset line pp ned try 41</i>
Bit 1	
Bit 2	<i>program error, starts till null on validity reset ritung 4</i>
Bit 3	<i>watch 13E</i>
Bit 4	
Bit 5	
Bit 6	
Bit 7	

Physical address: 80007H

The readout of the cause register also gives a "push" to the watch-dog timer. This means that this port has to be read periodically to keep processor running (see Watchdog).

Watchdog.

The watchdog is a "hang-up" detector that generates a hard reset to the processor if it is not "pushed" at every 2 seconds or less. The "push" is made by reading the cause register at physical address 80007H.

Special Control Register

2.11 Serial Interfaces

2.11.1 Keyboard Interface

The keyboard interface that is related to the video display unit uses one of the channels in the Z80-DART chip. There are two ways of connecting the keyboard: Directly to the X35 ~~Basic~~ UNIT or via the display monitor. In the latter case the keyboard lines in the monitor cable to the Basic Unit. The Keyboard uses TTL-signal levels. The baudrate clock is supplied from the keyboard.

2.11.2 Printer Interface.

The printer interface uses one of the two channels in the Z80-DART ~~chip~~-chip. The interface is a standard V24 interface with a programmable baudrate through the CIO (Counter IO) module.

2.11.3 ^{Communication} ~~Modem~~ Interfaces.

There are two serial general purpose ports in the X35. To make them complete data communication interfaces it is necessary to add option modules. Several types of interface types can be connected: V24 (RS232) , RS422 and even complete modems.

2.11.3.1 V24 Option module.

The V24 option module contain receivers and drivers for the V24-type of signal levels, and also the 25-pin D-connector. It is also possible to use split speed-operation, and for this purpose there is a clock-divider on the module.

2.11.3.2 RS422 Module.

The RS422 module contain receivers and drivers for the RS422 signal levels.

2.12 Bus Interfaces.

To be supplied.

2.13 Floppy disk interface.

The integrated floppydisk-controller employs a WD 1797 controller chip that is capable of controlling single and double density formats for 5 1/4" and 8" disk drives. The following IO-ports apply to the floppy disk controller:

LDAB /344L ,E.Paananen

04 01 25

Spec. X-35

DATORDEL

Mekanik

Datorlåda i aluminium.

Dimensioner: 470 x 450 x 180 mm

Vikt : 7kg

Funktionsuppbyggnad

Datorn består av två funktionsblock :

Dator och grafikgenerator

Dator:

- CPU - CIO - DART - SCC - DMA
- ROM/EPROM Bootstrap
- RAM (användarminne)
- Kommunikationsinterface (3st)
- Interfacebuss 4 st
- Flexskiveinterface

Grafikgenerator

- STATE MASKIN
- SNABB "MOVER"
- RAM

Centralenhet

Mikroprocessor 68008
 Ord längd 8 bitar / 32 bitar
 Klockfrekvens 8 MHz

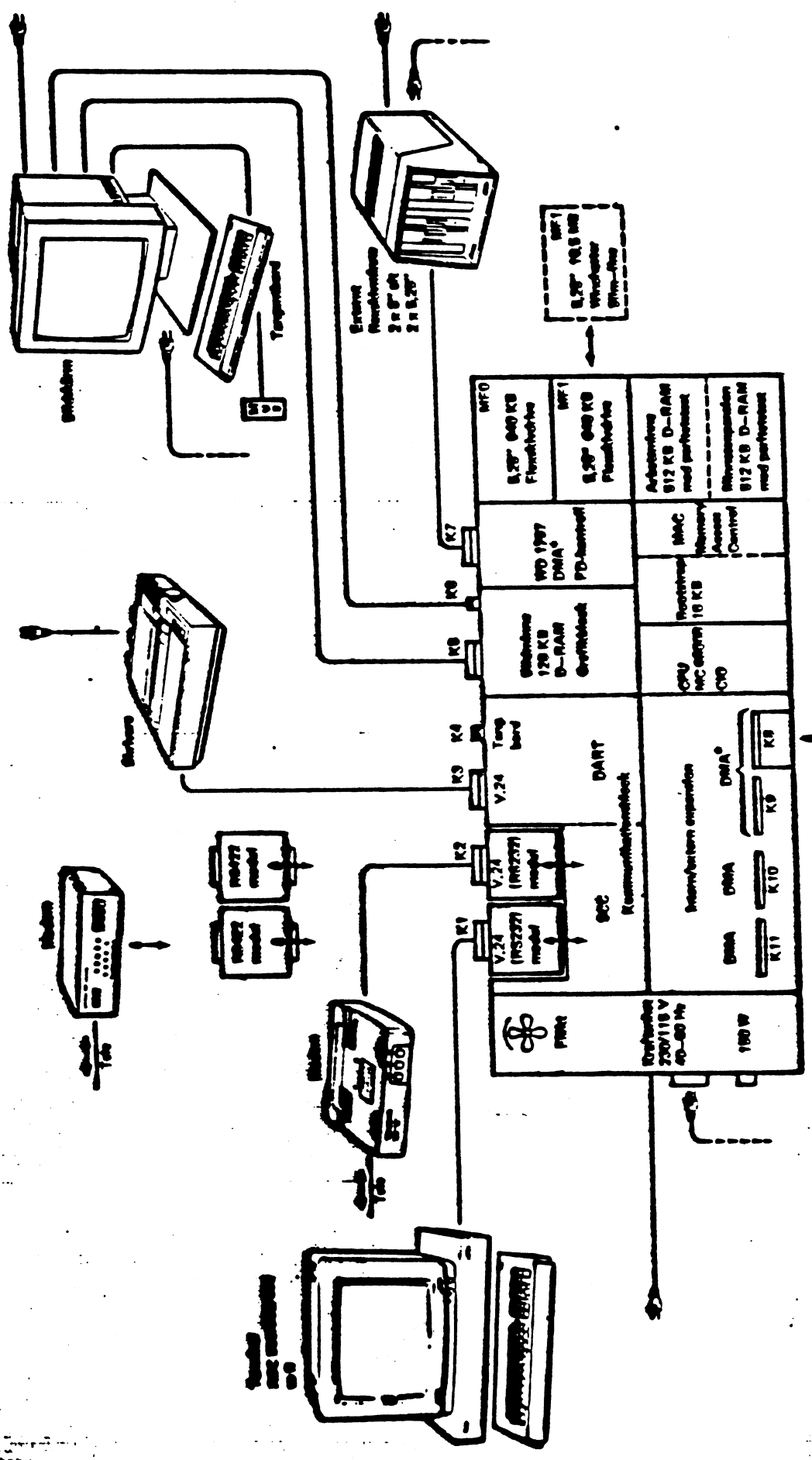
Minnen

ROM :

- 16 KB Bootstrap 27128

RAM :

- 512 Kbyte användarminne, utbyggbart till 1 Mbyte
 (4Krutaketter 256 Kbit & resten)



© 1992, 1993 Intel Corp. All rights reserved.

84 01 25

In- och utmatning
-----**Kommunikationskanal A :**

- asynkron överföring
- programmerbar överföringshastighet 50 - 9600 baud
- avsedd för skrivare

Kommunikationskanal B och C :

- synkron / asynkron sändning / mottagning
- programmerbar överföringshastighet 50 - 9600 baud asynkront
50 - 512K baud synkront
- NRZ - NRZI inbyggt enl. produktspec. SCC
- avsedd för kommunikation, skrivare m.m.
- drivkretsar och kontakt till ena SCC-kanalen på modulkort
Byte av modulkort möjliggör växling mellan V24/RS232 och RS422

Systembuss :

- 4 st. kortplatser för expansion.
En av dessa utrustad för anslutning av yttre expansionsenhet.
Alla kortplatser har DMA.
Det finns 3 DMA kanaler för ändamålet

Flexskiveinterface :

- för såväl 5.25" som 8" flexskivenhet

Kalender :

- fast inbyggd CMOS-kalender med klocka
- kalendern drivs med accumulator
- parameter minne 256 bit (16 x 16)

Kontaktödon
-----**Bildskärm :**

- 15-polig D-sub (hylsödon)
- 1 st BNC

Kommunikationsmodul A , B och C :

- 25-polig D-sub hylsödon Kanal A
- 25-polig D-sub stiftödon Kanal B , C

Systembuss :

- 4st 64-poliga Europakontakter med olika avbrottnivåer

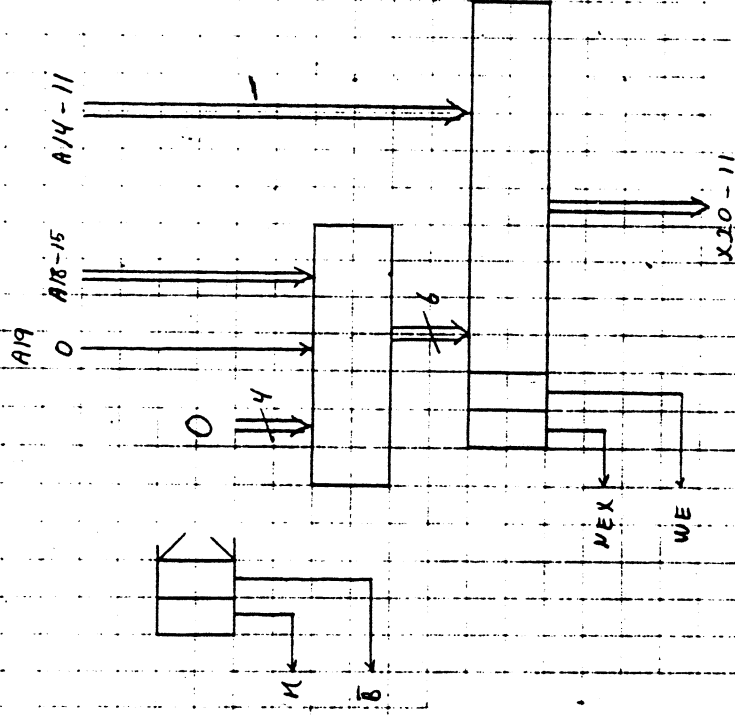
Tangentbord :

- 7 polig DIN

Flexskiveinterface

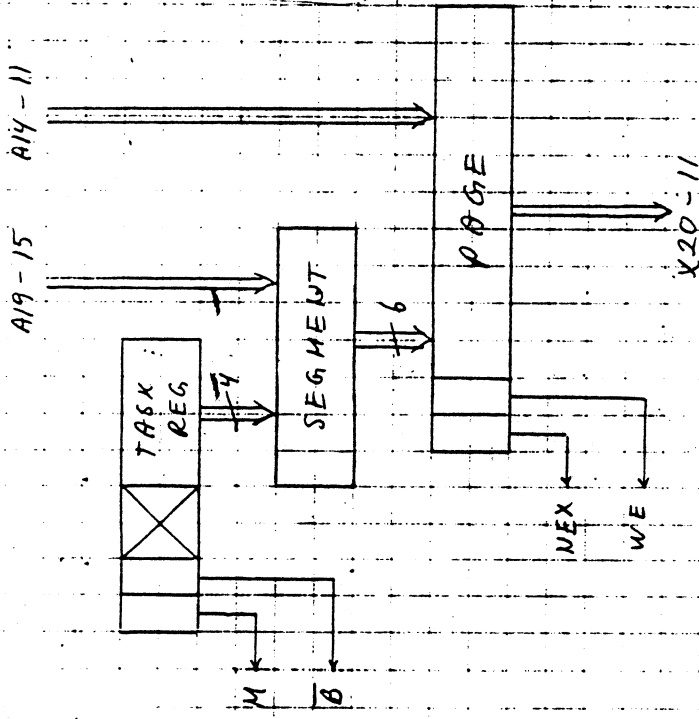
- 24 och 50 polig för interna drivenheter

Buss	DMI
- Adress	32 bitar
- Data	32 bitar
- Bandbredd	10 MHz
- Överföring	max 13.3 Mbyte/s
Primärminne	512 Kbyte/kort
- Datalagring	18 bitar (byteparitet)
- Minneskapslar	64 Kbit
- Acesstid	200 ns
Massminne	
- Flexskiveenhet	1 Mbyte 8"
- Winchester-skiveenhet	20/60 Mbyte 5 1/4"
- Anpassning	SCSI/SASI
Säkerhetskopiering	
- Flexskiva	se ovan
- Magnetbandstreamer	45/60 Mbyte (för DC 450/600 XL resp.)
Terminal- och skrivar-anslutningar	V24/RS232C
- Hastighet	19200 baud
Extern kommunikation D-NET	
- Hastighet	512 Kbit/s
- Avstånd	1000 m
Synkron överföring	V24/RS232C
- Hastighet	9600 baud
<u>Kraftaggregat</u>	
- Spänning	190-260 V
- Frekvens	47-400 Hz
- Effekt för grundsystem	175 W
- Elsäkerhetsstandard	SEMKO
<u>Fysiska data</u>	
- Vikt	40 kg
- Dimensioner (b x h x d)	220 x 645 x 600
- Temperaturintervall (drift)	10° - 35° C
- Temperaturintervall (transport)	-40° - 60° C
- Luftfuktighet	20 - 80 %
- Ljudnivå	44 dB (A)



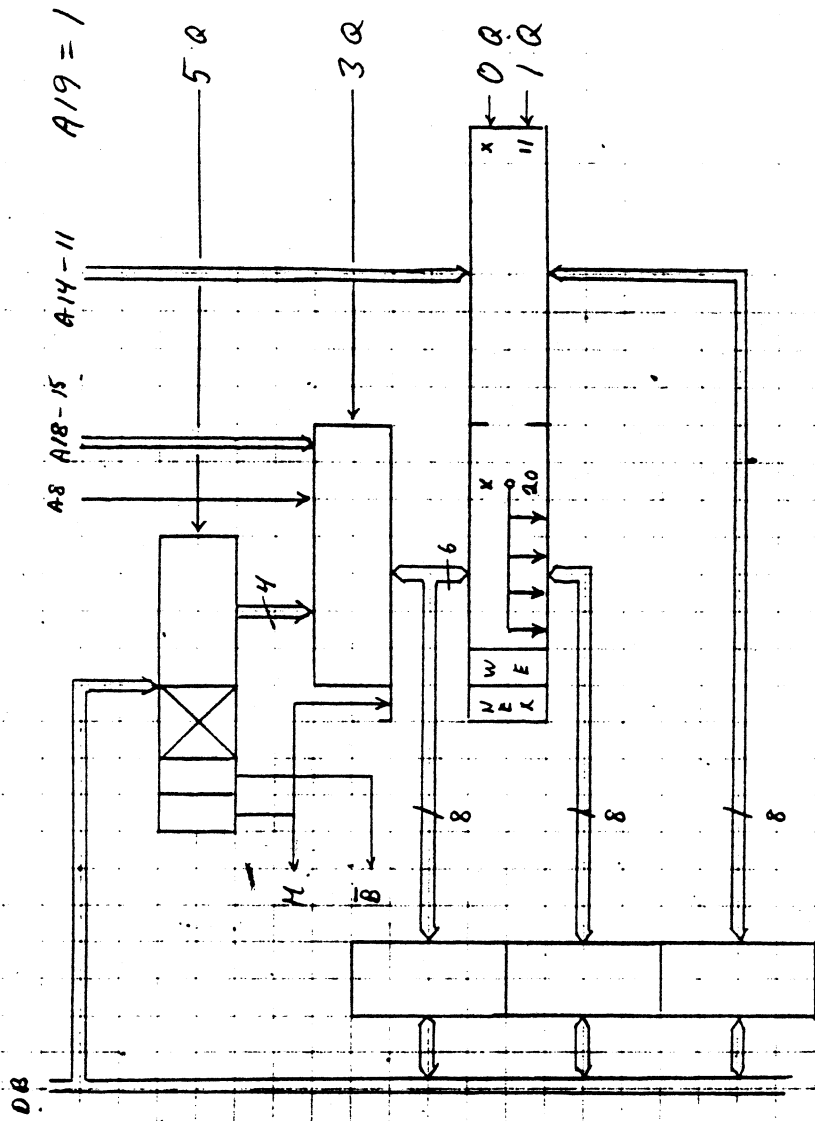
System mode
and

Magic user prog ref.



Normal user mode
and

Magic user data ref.



System mode MAC accesses

Oscillator

64 MHz neddelas till

32, 16, 8, 4, 2 MHz

- Reset (buffered)
disablar alla sign utom
WCLK1
Refclk
2 * clk
clocks disabled
- Enable Clocks släpper fram klockorna
efter att state ramarna laddats
- WCLK1 nondelay shift load reject
clock
- shift load pulse shift reg load
pulse
- double CCLK lessens the load

CRT-controller and videocircuits

- \overline{CCLK} character clock
each char means 2 words
- \overline{IORQ} Phases CPU communication
- R/\overline{W} Determines com direction
- $CRT\#$ selects the controller port
- $A\#$ register select ?
- $\overline{Endisp.}$ starts 6845 counters
- \overline{Reset} resets counters
- $Disp\ en$ blank retrace blanking
- $Disp\ mem\ adr$ display address

extremely sensible area

- fix info from sh reg digital video
- fix polarity into polarity
- Blank flag enables/disables display
- CCLK clocks sync

- Video out
- Hsync
- Vsync

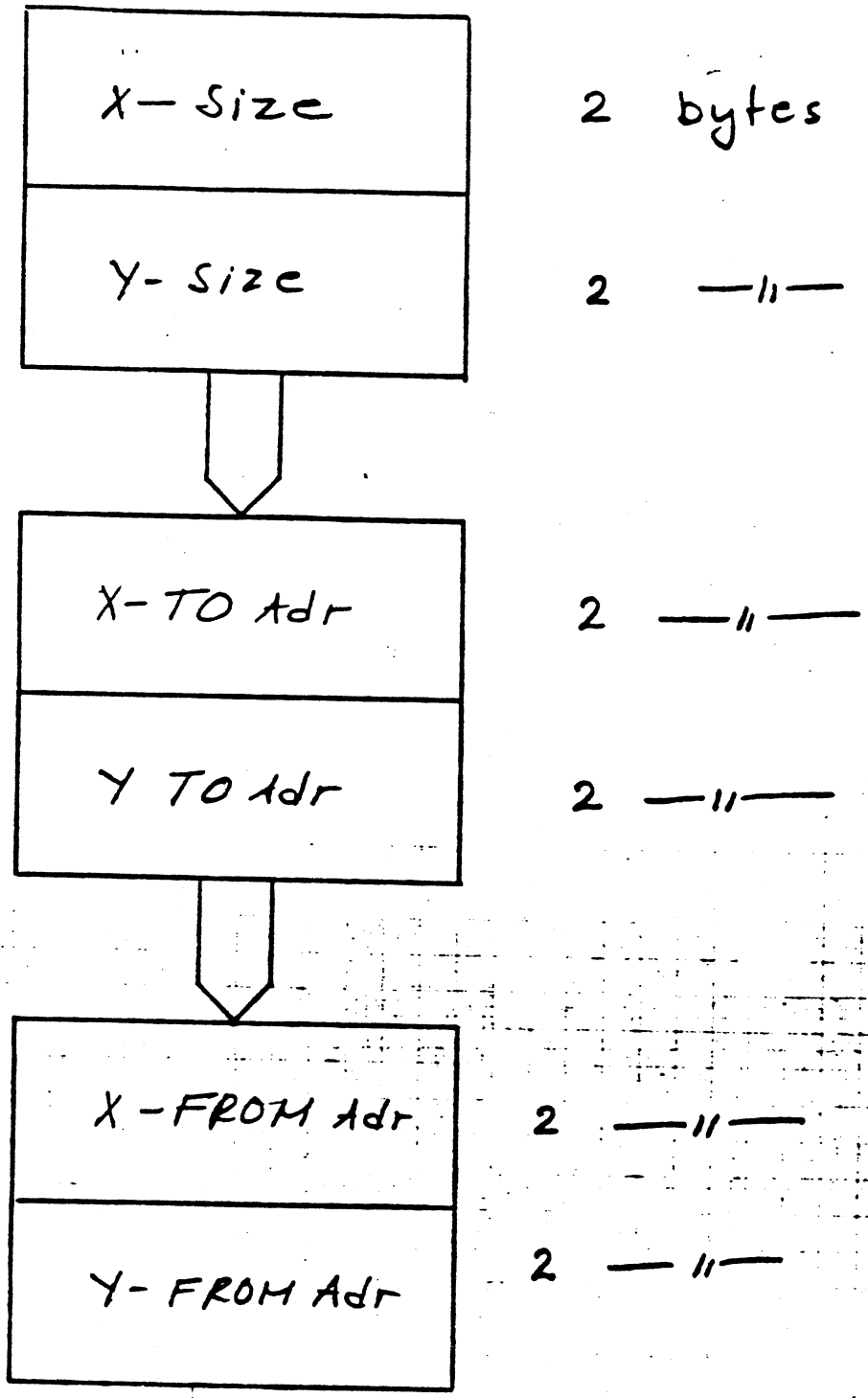
analog video
horizontal sync
vertical sync

Cycle Control

- \overline{CCLK} , \overline{WCLK} ,
 \overline{PIXTB} , \overline{PIXIS} state ram addresses
- $\overline{Move\ State\ OE}$ selects Move State Ram: S
- \overline{WRML} Write Move Low byte strobe
- \overline{WRDL} Write display " - " - " -
- $\overline{Disp\ State\ OE}$ selects disp. state Ram: S
row blanking
- Gate out ch reg load
- $\overline{Pix\ clk/2}$, $\overline{Pix\ clk}$ clocks for pipelining
- \overline{RAS} Row ADR strobe
- $\overline{DTACK\ clk}$ clock Data Acknowledge
- \overline{MAM} Mem ADR mux
- \overline{CAS} column adr strobe
- $\overline{Disp\ reg\ clk}$ Pipelined disp words
- $\overline{Data\ clk}$ clocks read data regs
- \overline{DMWE} Disp Mem Write En
- $\overline{Move\ cycle\ clock}$ clocks in move cycle strobe
- $\overline{Load/shift\ left}$ controls shift regs.
- $\overline{Blank\ test}$ clocks display blanking
- $\overline{Disp\ cyc\ pre\ latch}$ makes sure the disp adr
propagates in time
- $\overline{Disp\ cyc\ select}$ For mem addr priority control

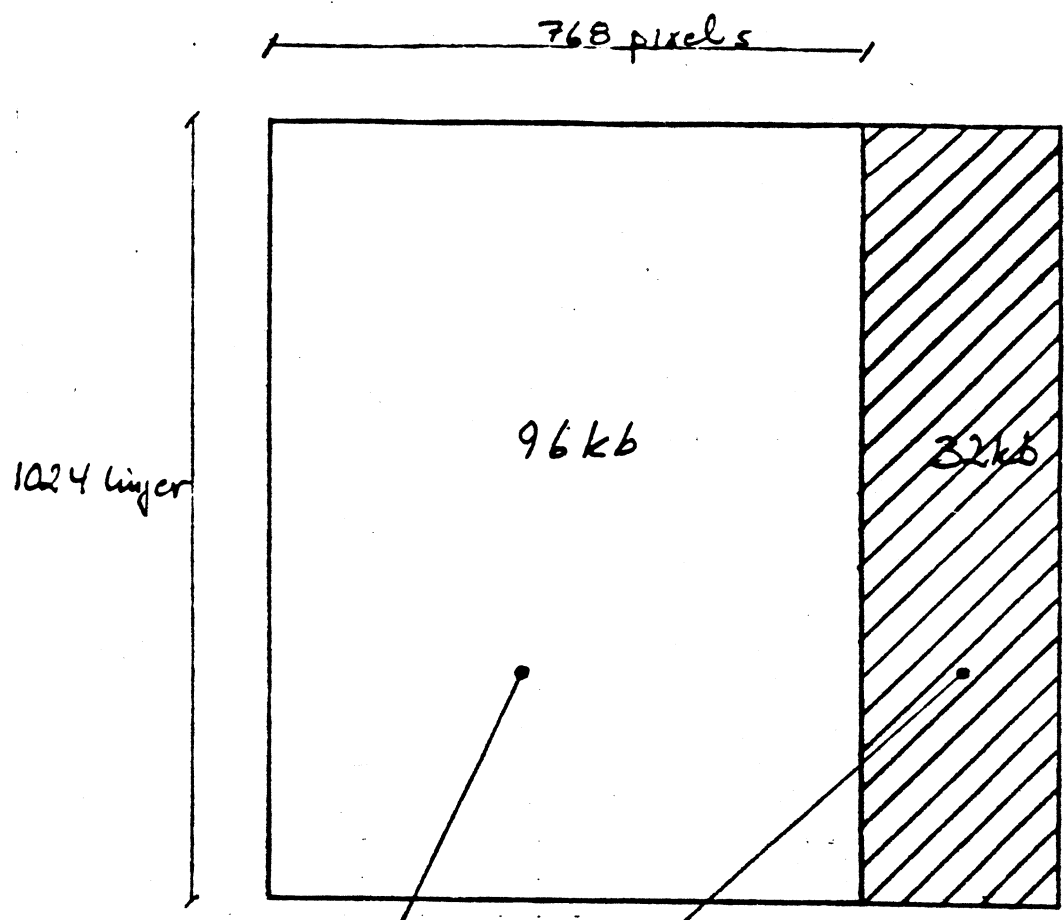
Blockmove

Command sequence



total 12 bytes

Upplösning

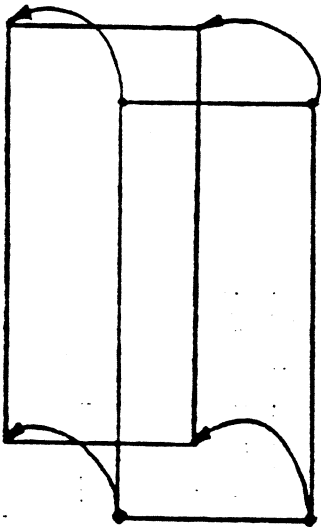
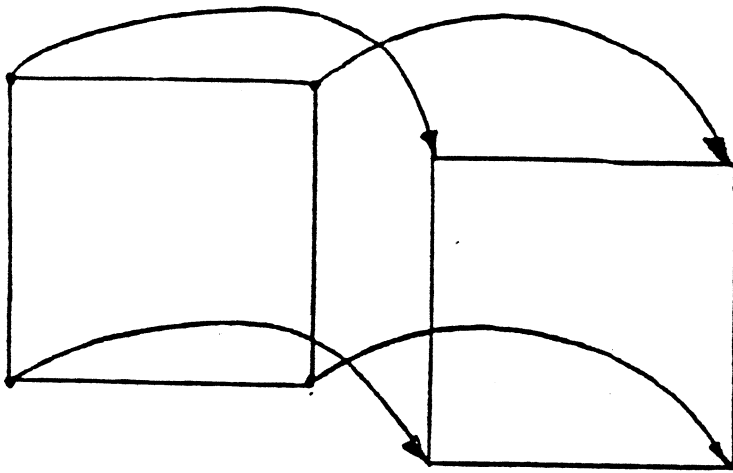


Del av minnet som inte syns på skärmen
I denna del ligger karaktärgenerators
och ev. grafiska symboler.

Del av minnet som visas på skärmen.

Block move

(14)

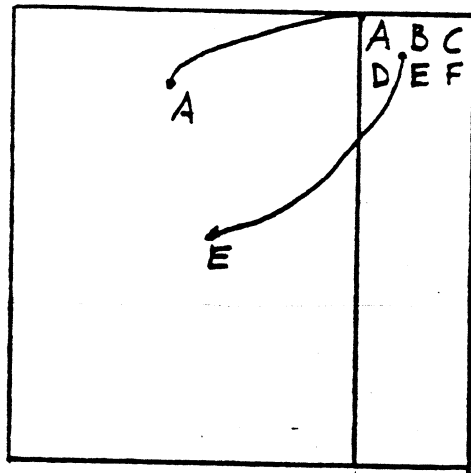


- Move kan göras på rektangulära ytor.
- Move kan göras i valfri riktning.
- Adresserna kan räknas uppåt eller nedåt
⇒ man kan överlappa block utan att skriva sänder data som skall läsas

Blockmove

(10)

- De rektangulära blockens storlek, source och destination bestäms med pixelupplösning i såväl x-, som y-led.
- När man arbetar i karaktärsmode ligger karaktärsgeneratorn lagrad i den dolda delen av grafikminnet och kopieras in i den synliga delen på beställning.



- Efter varje move automatiskt räknas ut till nästa karaktärs position på raden.
- Blocksize anges en gång // karaktärs generator

TO ADR anges en gång // rad

FROM ADR anges en gång // karaktär (synl)

Detta ger att man för varje tecken inom en rad endast behöver leverera FROM ADR = 4 bytes och TO ADR = 4 bytes för varje CR.

Graphic Commands

Line

Circle

Fill.
(Rectangular)

Paint
(Polygon)

Graphic Commands

- De grafiska kommandona betecknas helt i mjukvaran.
- Minnesoperationerna är

REPLACE

SET

RESET

Mjukvara

Blockindelning

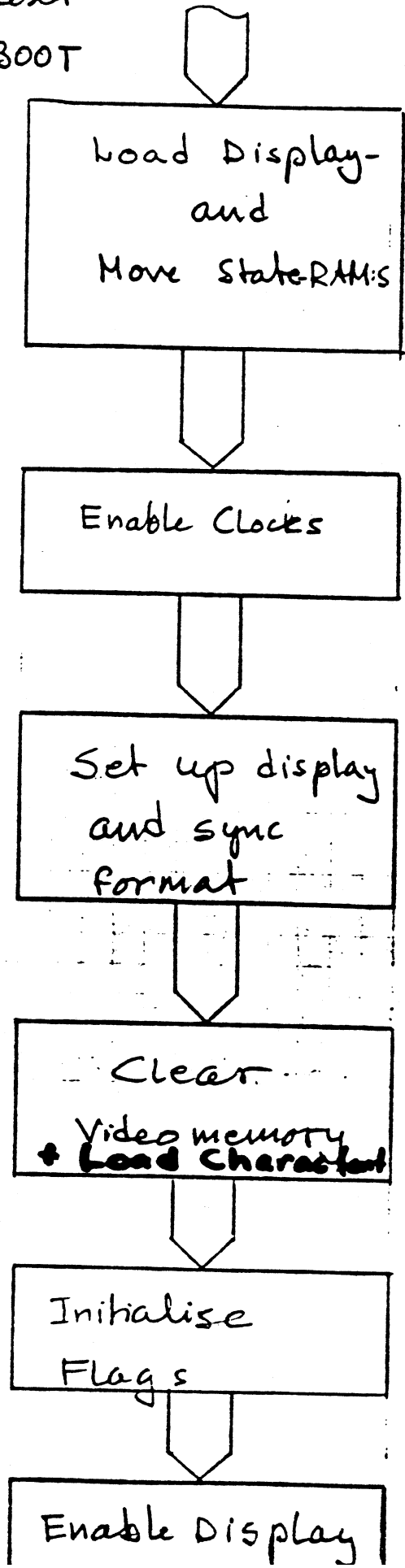
Initialisation

Blockmove

Graphic
Commands

Initialisation

From Reset
and BOOT



X35 - Grafiken

①

• Hårdvara

Blockindelning

- Osc (Oscillator och klocksignaler)
- CRTC (CRT-controller)
- Cycle Control (Logik som styr timing)
- Video
- V.Memory (Video Memory)
- Mover

• Mjukvara

Blockindelning

- Initialisation
- Blockmove
- Graphic Commands

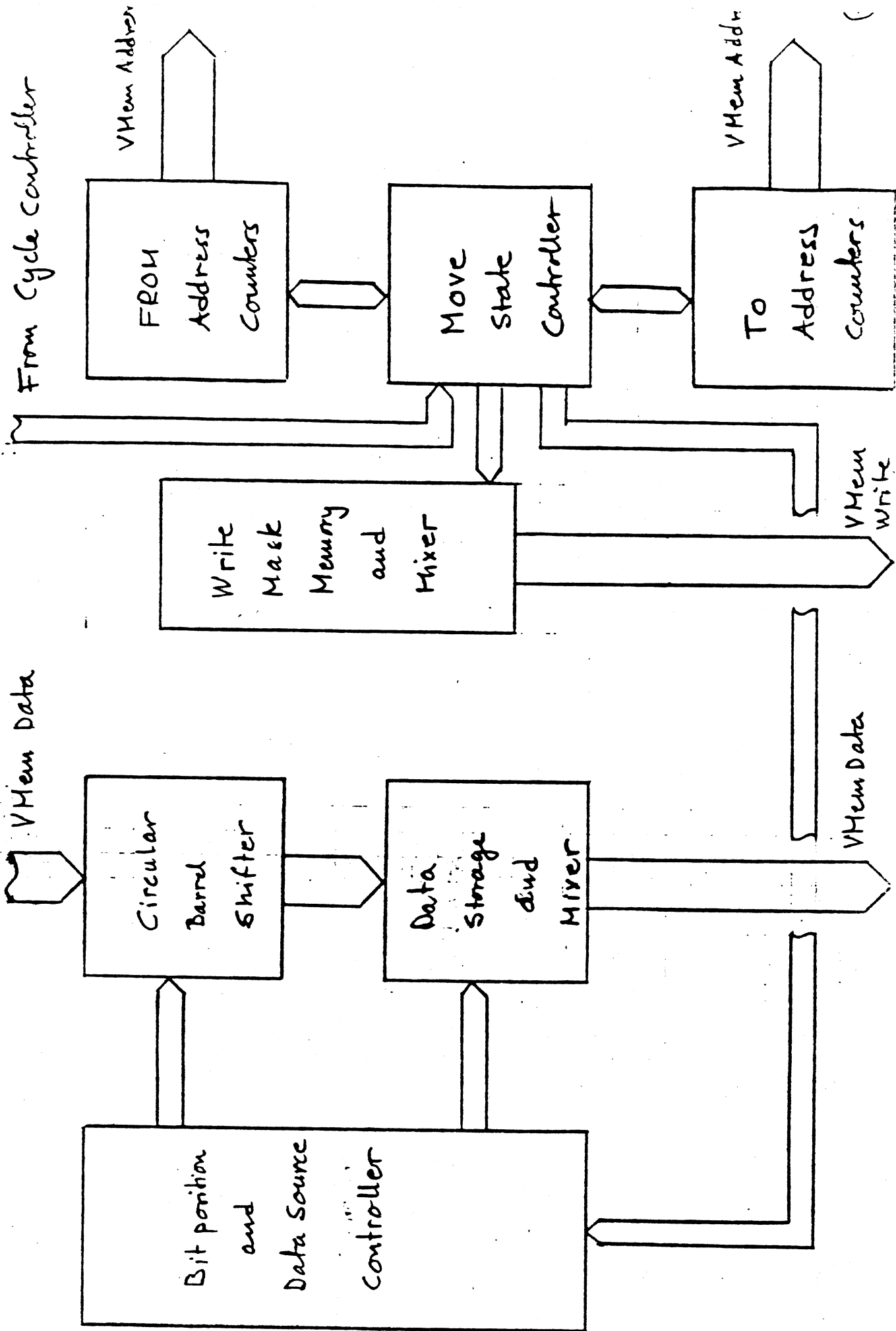
Blockmove

(16)

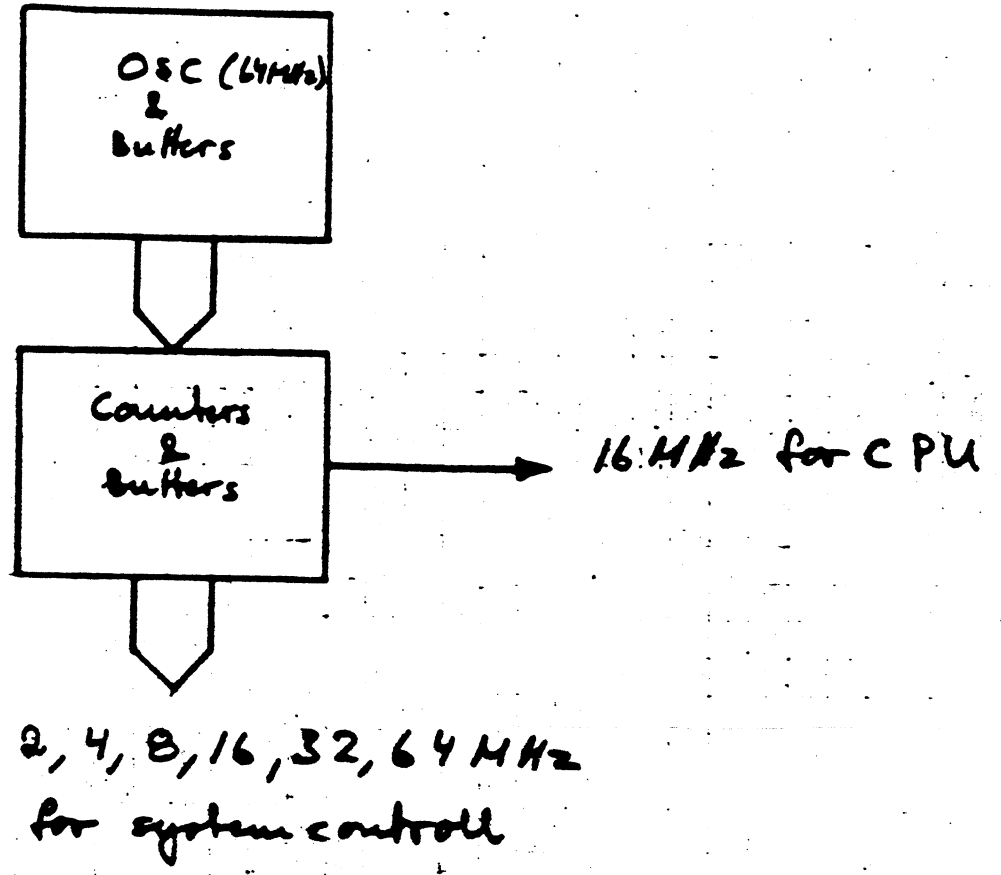
- Minnesoperationen vid move är REPLACE. Detta innebär att destinationdata helt ersätts av source-data inom blockets ramar.
- Blockmove kan endast göras i grafikminnet alltså inte mellan grafik- och CPU-minnet.

- Grafikminnet är av treparts typ. Detta innebär att minnet accessas från
 - Displayenheten
 - Blockmovern
eller
 - CPU:nCyklerna är nämnda i prioritetsordning.
- CPU-accessen är av konventionell typ.
- Display-accessen utnyttjar nibblemode, d.v.s. vill säga snabb utläsning av 4 konsekutiva ord, för att skära ned displayaccessertid.
- Blockmovraccesserna är av Read-Modify-Write typ men med Read-och Write-data på skilda adresser.

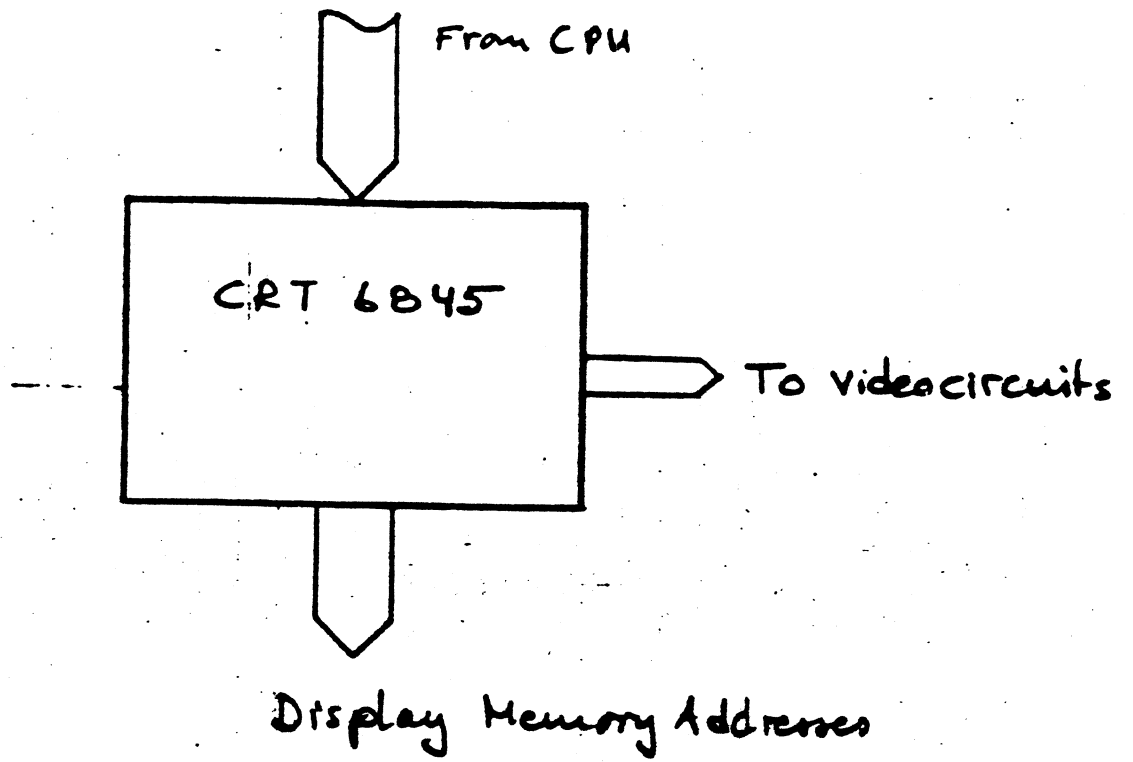
Mover



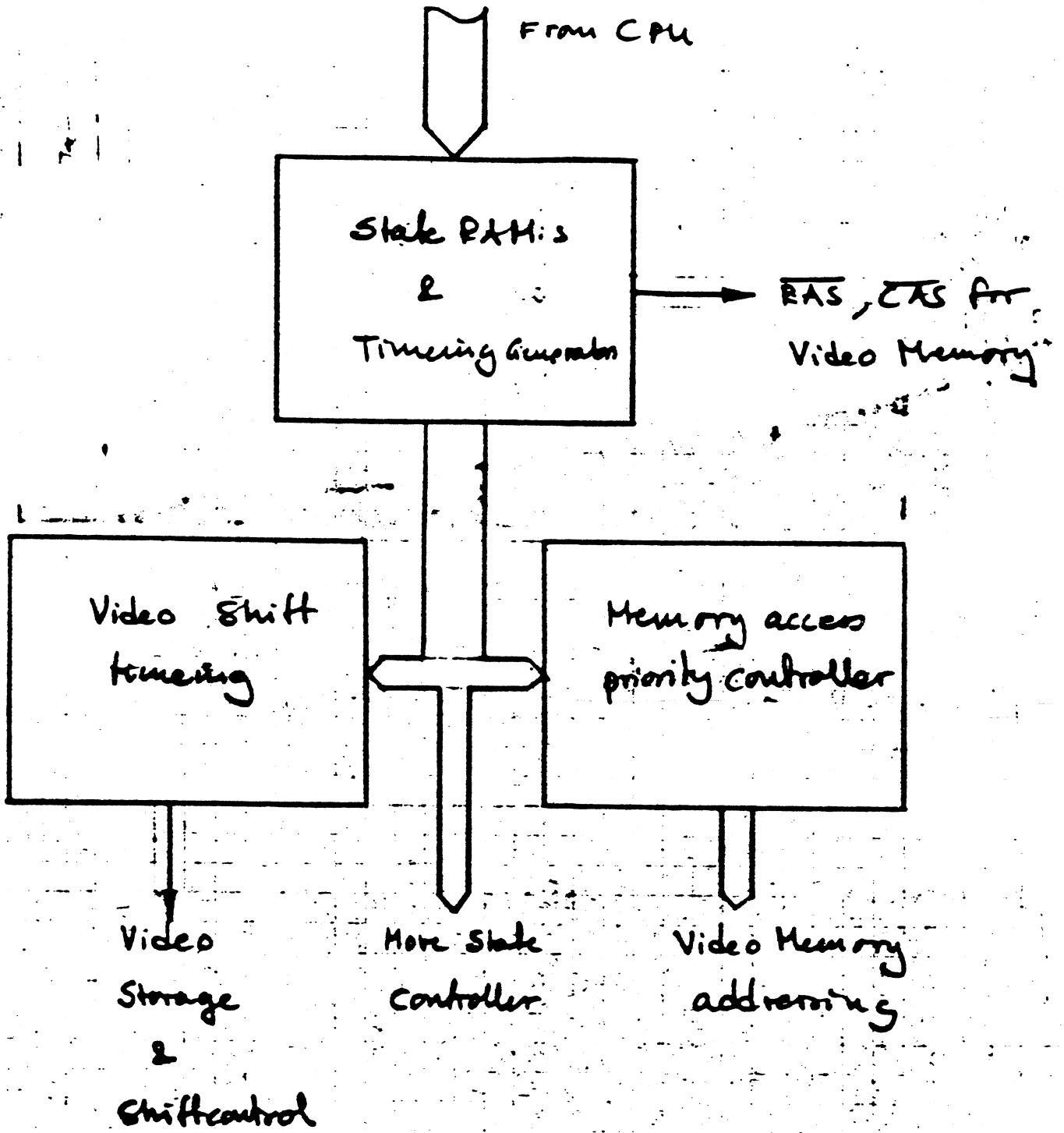
OSC



CRTG

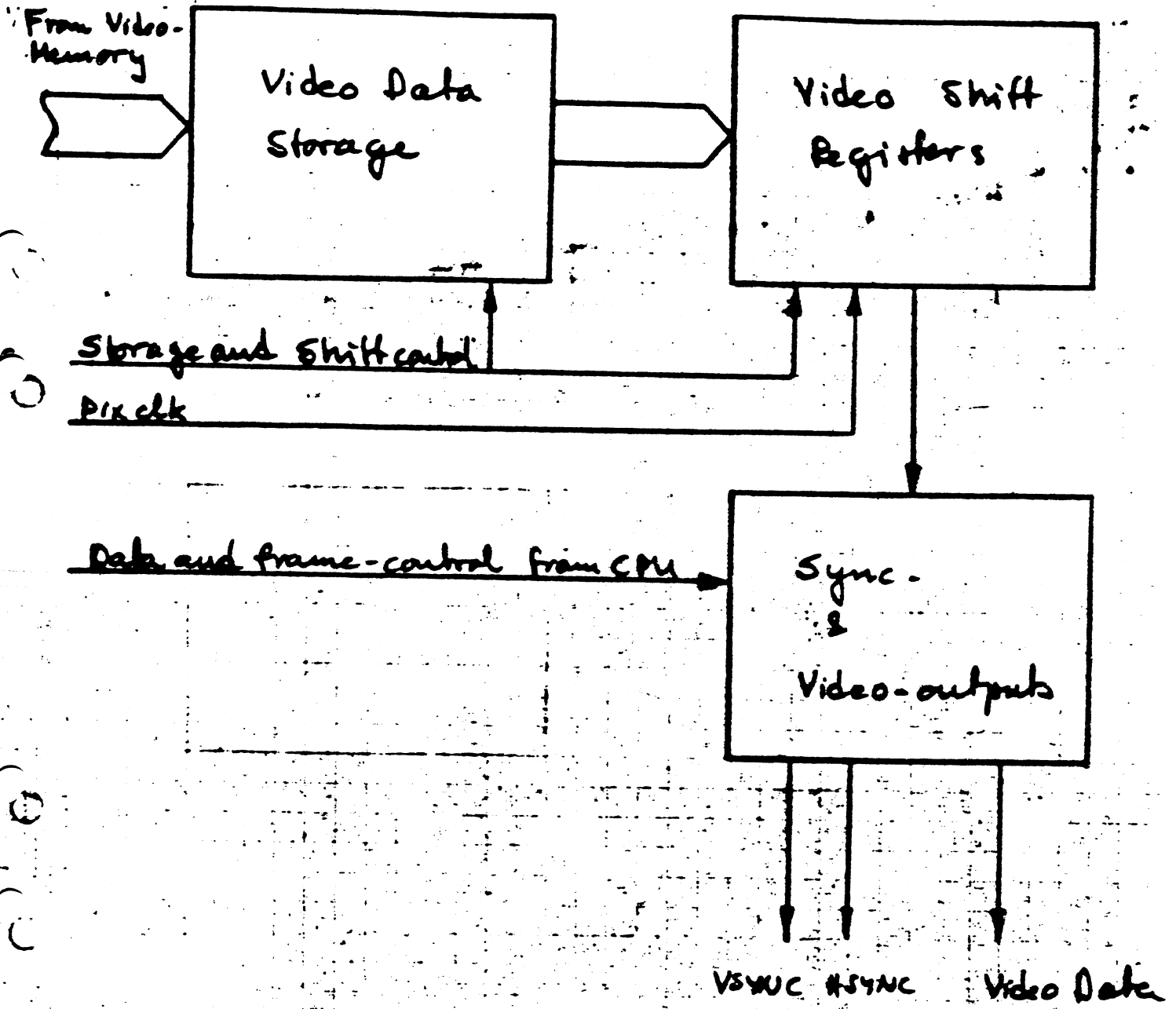


Cycle Control

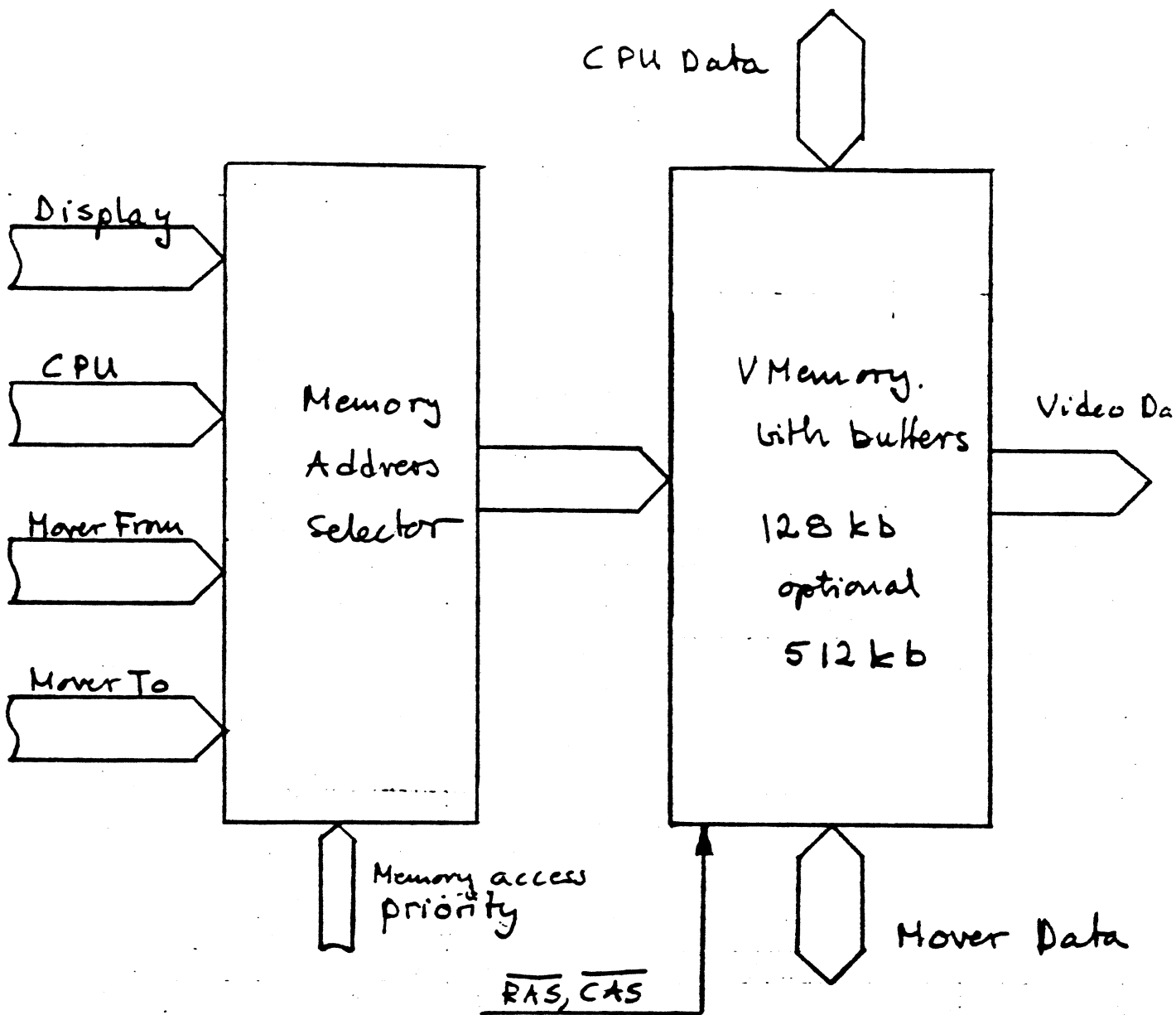


Video

5



Video Memory



state runs for fast access.
and flexible tuning.

⑥ • Gate out sh reg load helps get the right frame

⑤ state ram adr selector.

③ Memory access priority controller

- 1 Display access
- 2 Move access
- 3 CPU access

- AMM Active Mode Mask
- Disp cyc select Disp cycle select
- RD/WR Mask read or write move address
- CPU adr OE CPU address
- AMM Move address
- Disp Cyc prelate display address

Video

- D0p reg clk clocks the data words
- D0IP-N⁻ D0p data in
- Frame polarite sets frame polarity
- Pix clk video speed clk
- Load / shift left shift reg mode control

sensible area

Mover

(13)

Bit position and Data Source Controller

- $\overline{U/DX}$ up/down count on xcounters
 - $TSP3-\phi$ to start pixel
 - $FSP3-\phi$ from start pixel
- } 16 bits ord.
- \overline{CMC} column mode count (one for each full x-count)

• Hold 1W cycle

Hold 1 word cycle in beg of every line to get enough data

• $SH3-\phi$

shift count

• Anded Marks

mark in middle of word.

(21)

• $DR33-\phi\phi$

Direct/Register for the 16 bits in move word.

9

Circular barrel shifter

- MDI 15- ϕ Moves data in
- SH 3- ϕ Number of shifts
- MDO 15- ϕ Moves data out

show pict.

10

Data storage and mixer

- Complement move flag
- DR 33- ϕ select direct or reg data
- MDD 15- ϕ Move data Direct
- MDR 15- ϕ Move data Register
- PD 15- ϕ PM DATA OUT

13

Write mask memory

- Move Mem Wr Write strobe
- AMM Active mode mask
- WPMST- ϕ Write mask select
Init middle final

- U/D Y

- PS8-Ø

pixel size (16 bit)

- WRM15-Ø

write mark.

(16, 12)

From adr counters

3 st 169 incremental bara register

2 st 169 word counters

174-374 from regs

2 st 169 from counters

367 244 incremental buffers

174, 374 size regs

283 to + size

521 end comparator

to adr counters

PS> förutom att to reg ej incrementeras
och from ej behöver laddas an

8

- Rd/ \overline{wr} master decides the move cycle
- AMM clk synchronous with \overline{LD} clocks the AMM reg
- hold signals
- avoid bus crash buffers

14

- \overline{LD} FROM \overline{Y} CNT strobe \overline{LD} starts the move
- \overline{CMC} column mode count
- \overline{RMC} Row mode count
- **AMM** Active mod mark vibting

CPU interface

⑩ Replace, Set & Reset mode control

- Replace/set Reset mode flag
- Data strobe from CPU HB, LB
- CPU WRMask strobe HB, LB from I/O write mask port.

expl. about diff modes

- Data port strobe HB, LB
- write port strobe HB, LB

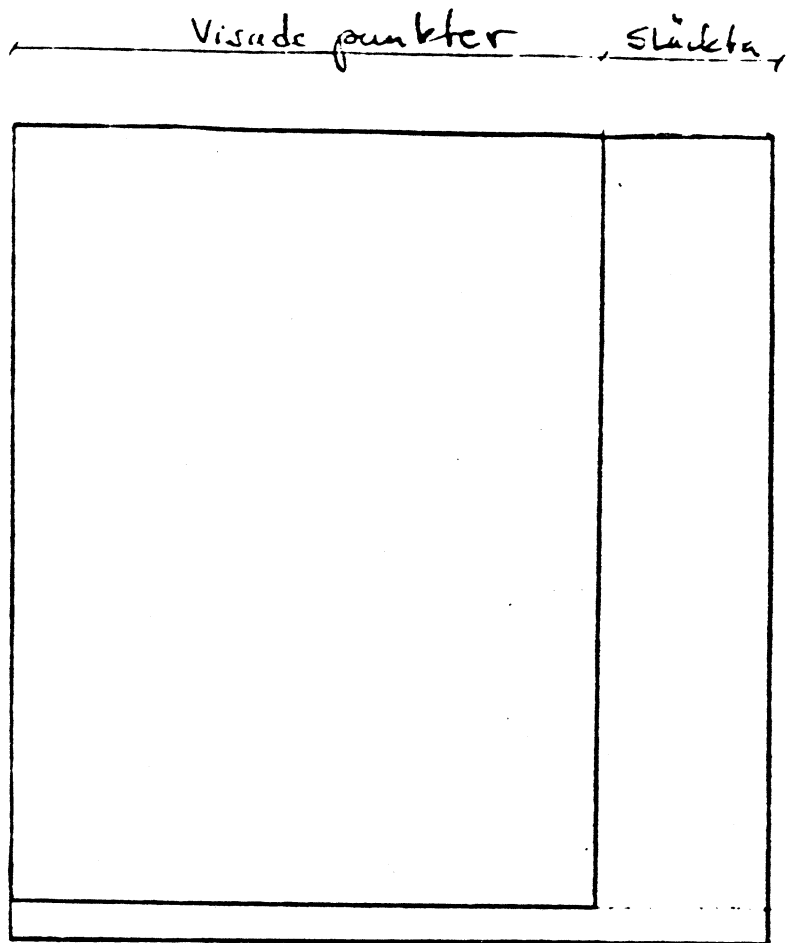
Replace — data fill data port
WR → WR port

Set — 1:0r fill data port
data → WR port

reset — 0:0r fill data port
data → WR port

Blockmoverns Prestanda

(10)



- Movern flyttar 24 M pixels/s under visnings tid och 32 M pixels/s under strålåtergå tid.
- Visningstiden är 74% och släcktiden 26% av totala tiden.
- Detta ger att movern hårdvaramässigt kan flytta 26.1 M pixels/s eller flytta skärmens innehåll 33.2 ggr/s.

① Mem write and \overline{DTACK} signals

- \overline{DS} (bust) data strobe
- $A\emptyset$
- $X20-12$
- \overline{DAS} delayed ddr strobe
- $\overline{NO DAS}$ not delayed
- $\overline{Stactck}$ clock to sync CPU \Rightarrow memory
- Wr / \overline{Rd}
- $A\emptyset$ byte \Leftrightarrow word
- $\overline{Data\ strobe\ from\ CPU}$ 4B, 1B
- $\overline{GMEM\ DATA\ OE}$ oe for read data
- \overline{DTACK} pick data acken.

② I/O errors \Rightarrow flags.

Video Memory

③

- \overline{CCLK} clocks dirp adr
- Dirp Mem ADR 15-2 (M. bble mode \rightarrow 1- \emptyset)
only 128 kbytes addressing

• CPU ADR 8-1 (16 bit word CPU-bytes)

• Move TO 7- \emptyset Mover to adr

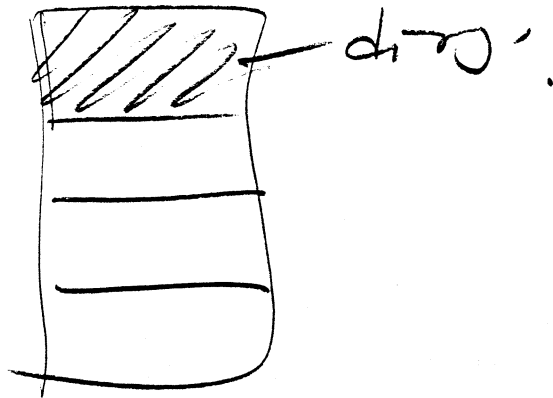
• Move From 7-0

• CPU 18-9

• Move TO 17-8

• Move From 17-8

} 512 kbytes adr.



• MAM memory adr mus

• MADDR 8/17 — 8/9

OBS: the adr order

spes A \emptyset and A1 Mibble adrs

①

- ~~RAS~~
- ~~CAS~~

Row Adr stroke
col. Adr stroke

- Disprop clk

same delay as CAS
CAS is clocking out
the nibble data

- A17- \emptyset

disp mem adr.

- AMM

Active mode mask
disables CPU data when moving

- DPS HB, LB

Data Post stroke high byte
low byte

- WPS HB, LB

write Post stroke . . .

Performs

set/Reset or Replace CPU
modification of memory

- D7-D \emptyset

CPU Data

- P00/P5 \emptyset

PAL DATA out

- WR CPU TO mem HB, LB

write stroke from
CPU enables CPU
write mask

- WR M15- \emptyset

Mixer write mask

- Data Mask

write is pixel selectable
Determines when From mem
with means read of mem dot
clocks data into reg.

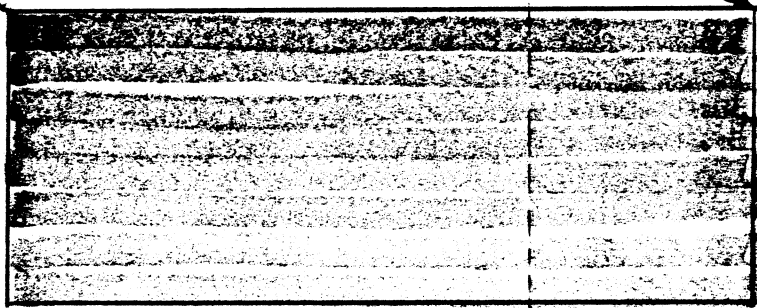
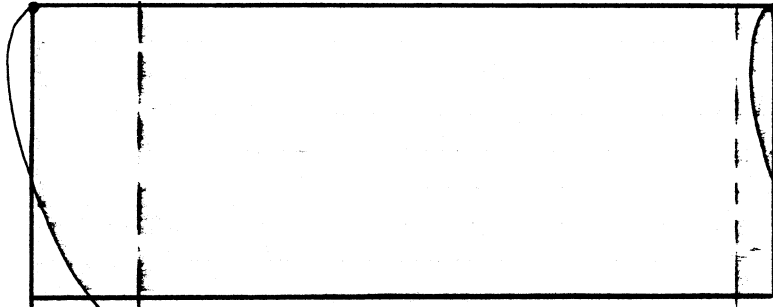
- Data clk

- DDF 15-0 Display data in
- MAI 15-8 Mover data in
- CPU & MAI 7-0 CPU and Mover data
- CPU date 15-8 High byte

Shift = 3



3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2

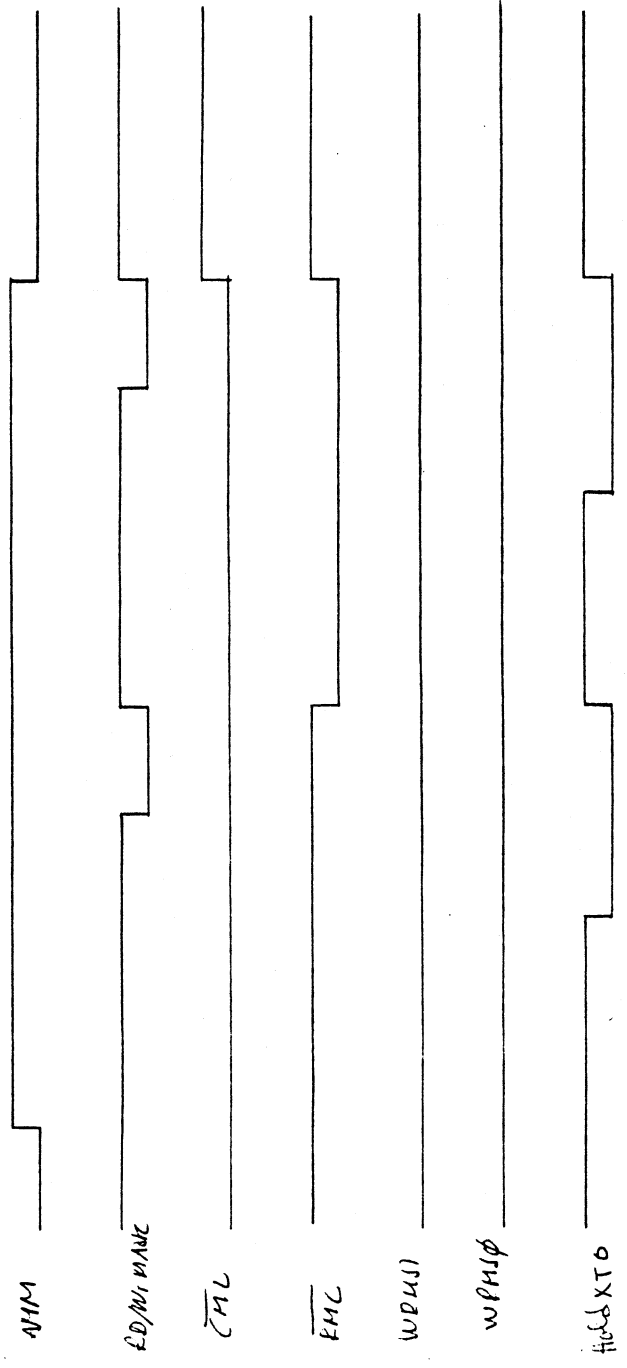


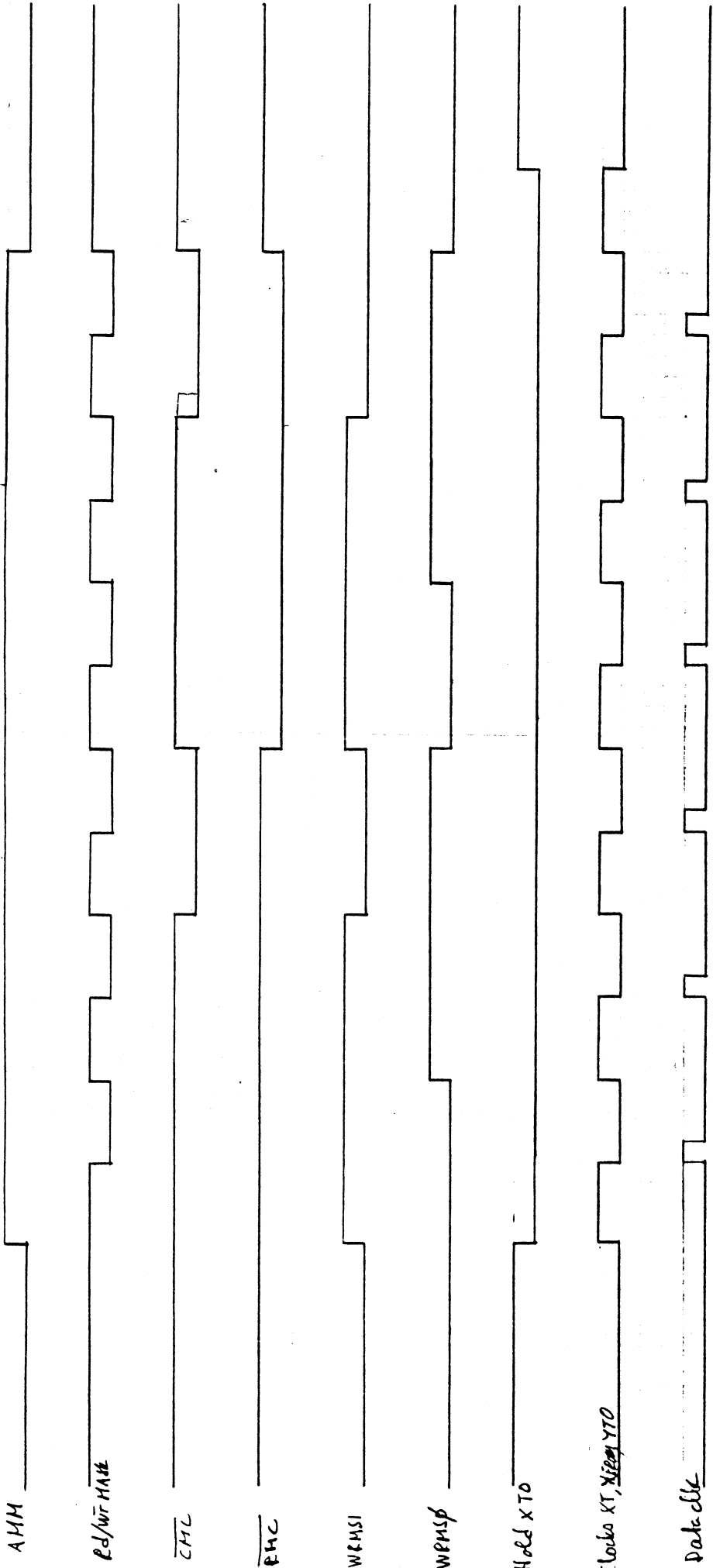
Physical word limits

)))

)))

Timing for counters holds 8 and 10



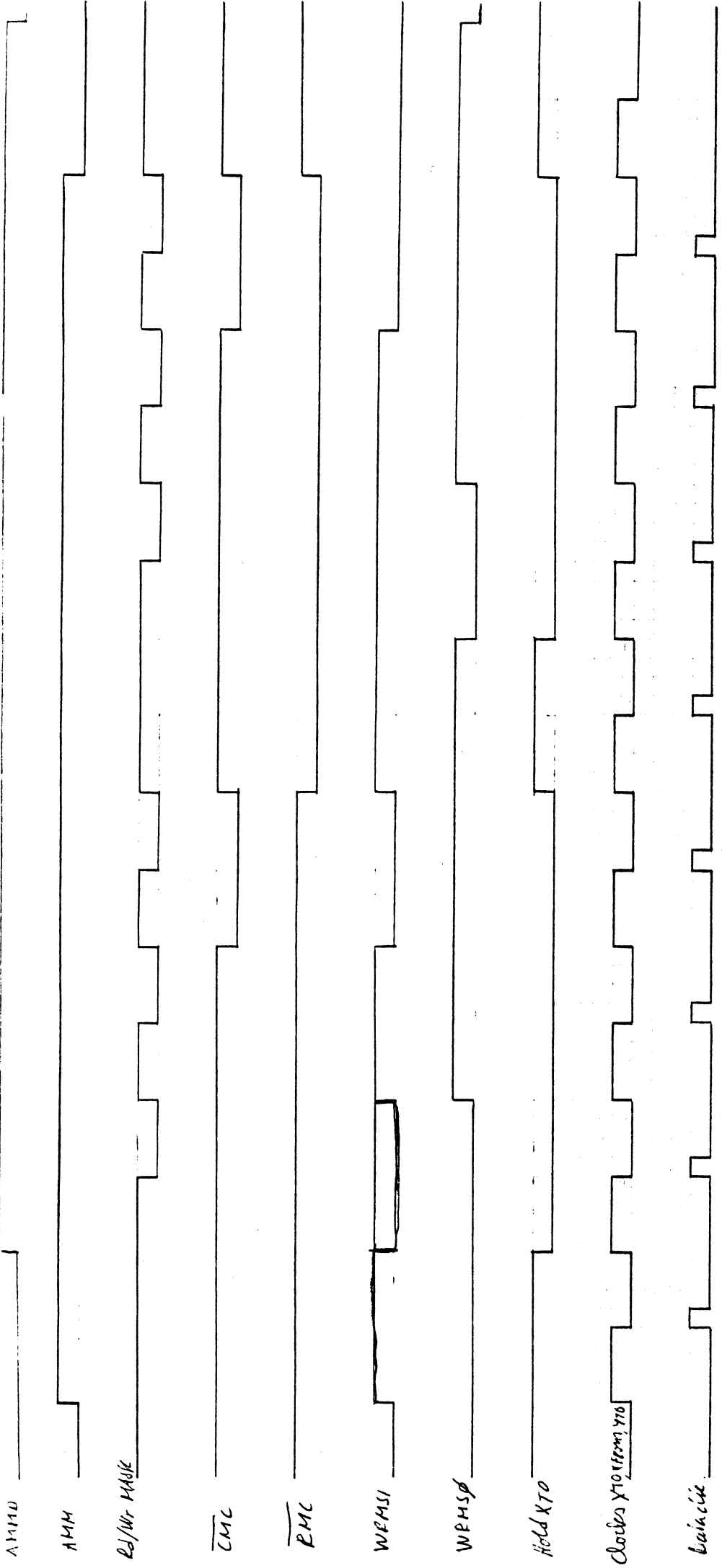


Timing for counters (No hold)

))))))))))

Timing for counter roll

Timing for counter roll

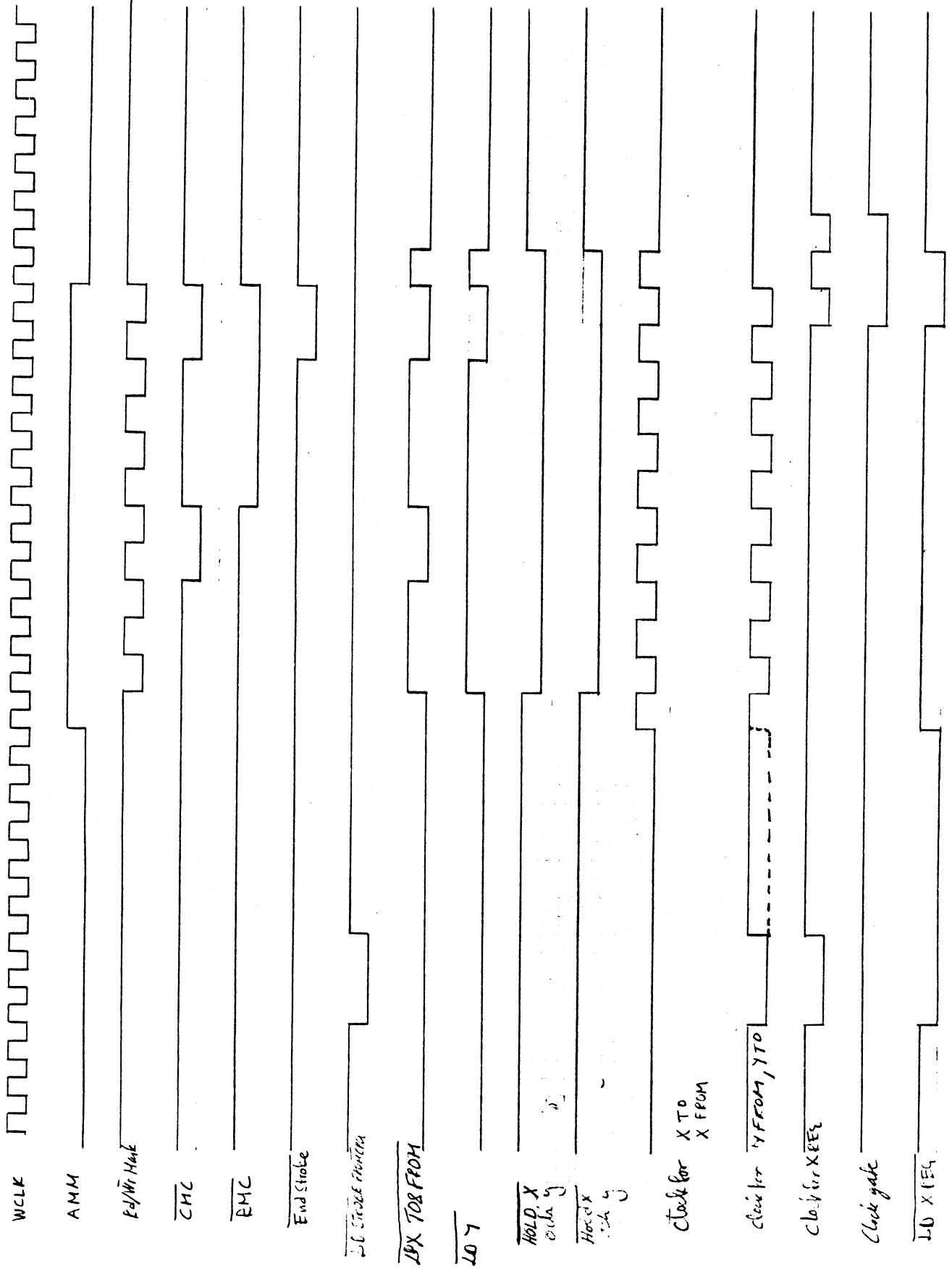


Portrait

Level and Clock timing for H01

8401/16

line 0, line 1

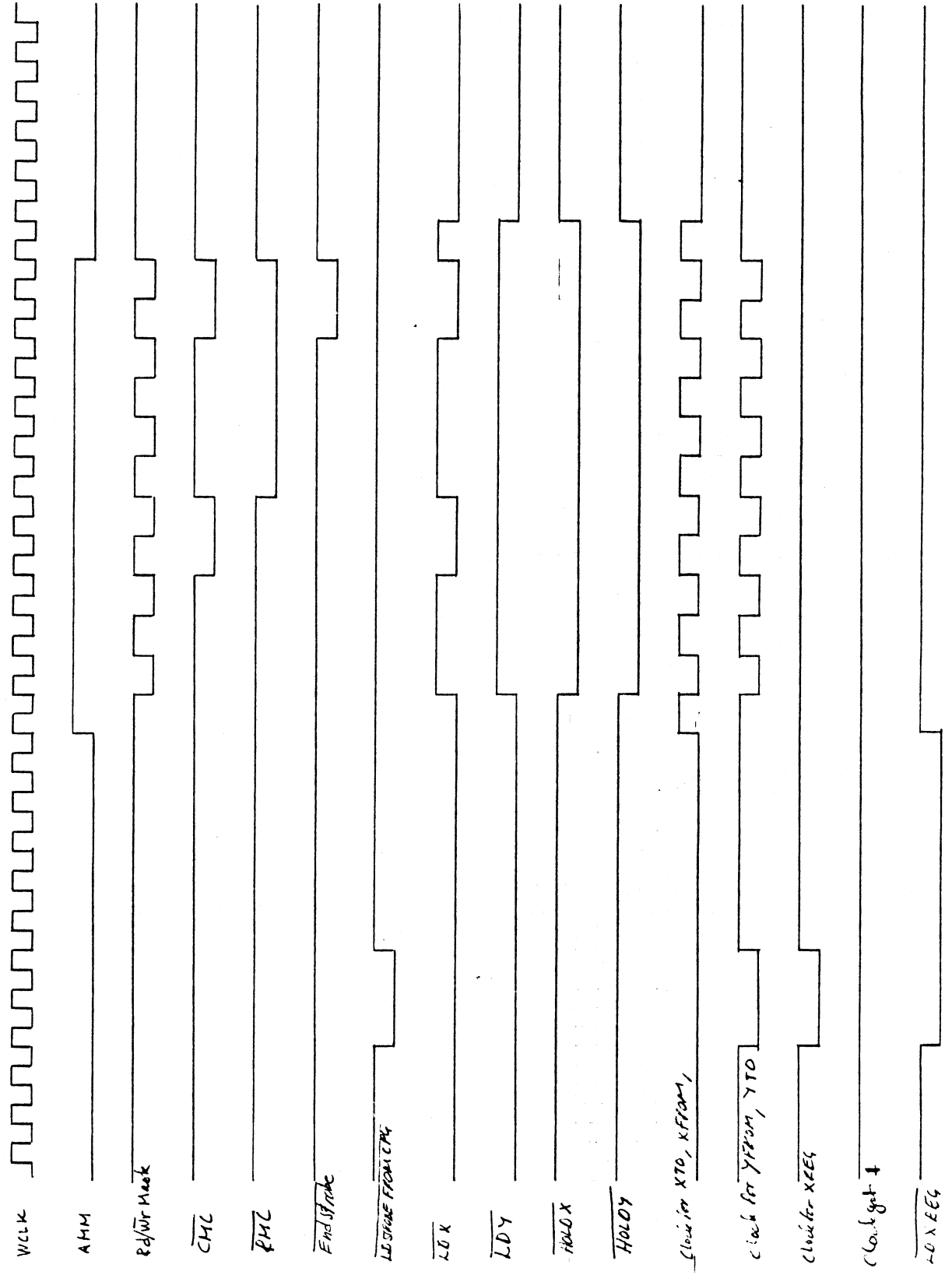


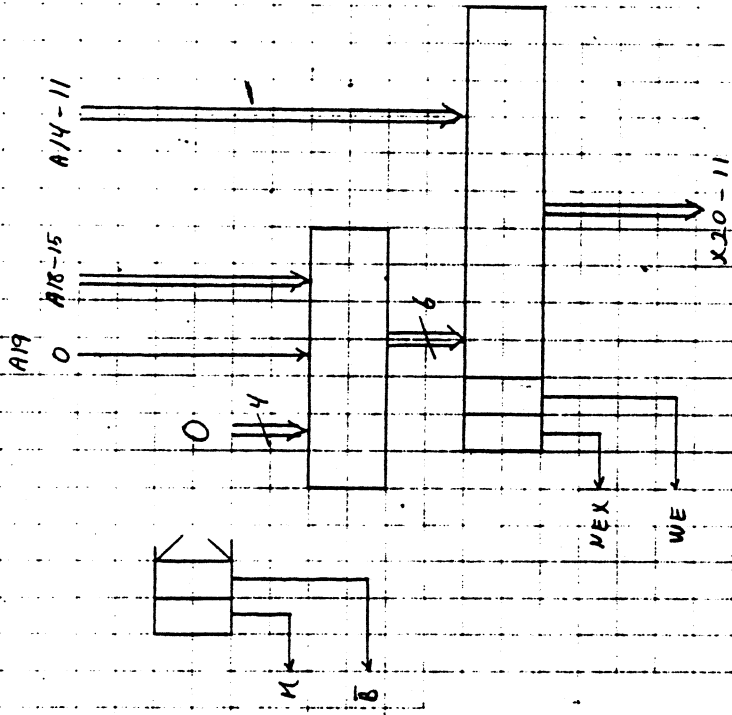
840116

Layout and Clock Timing for M01

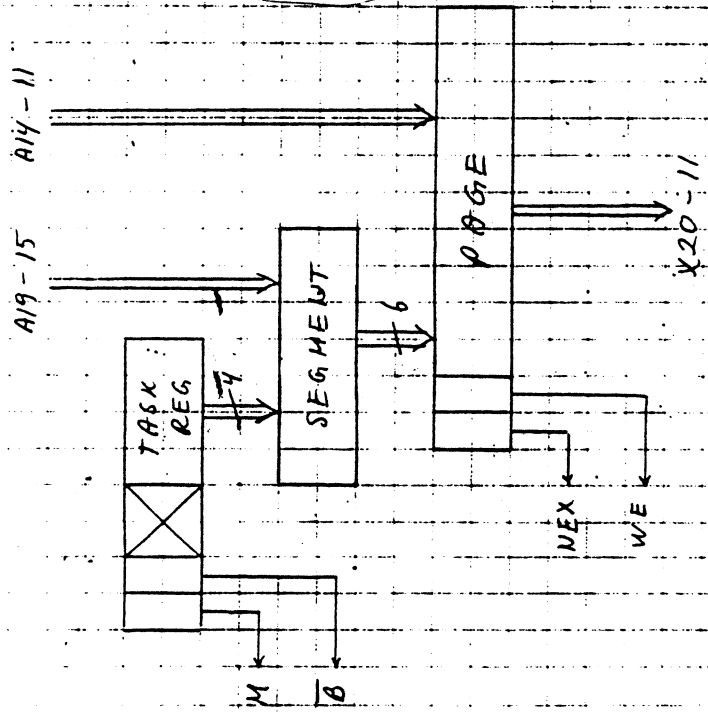
Handcraft

line ϕ line 1

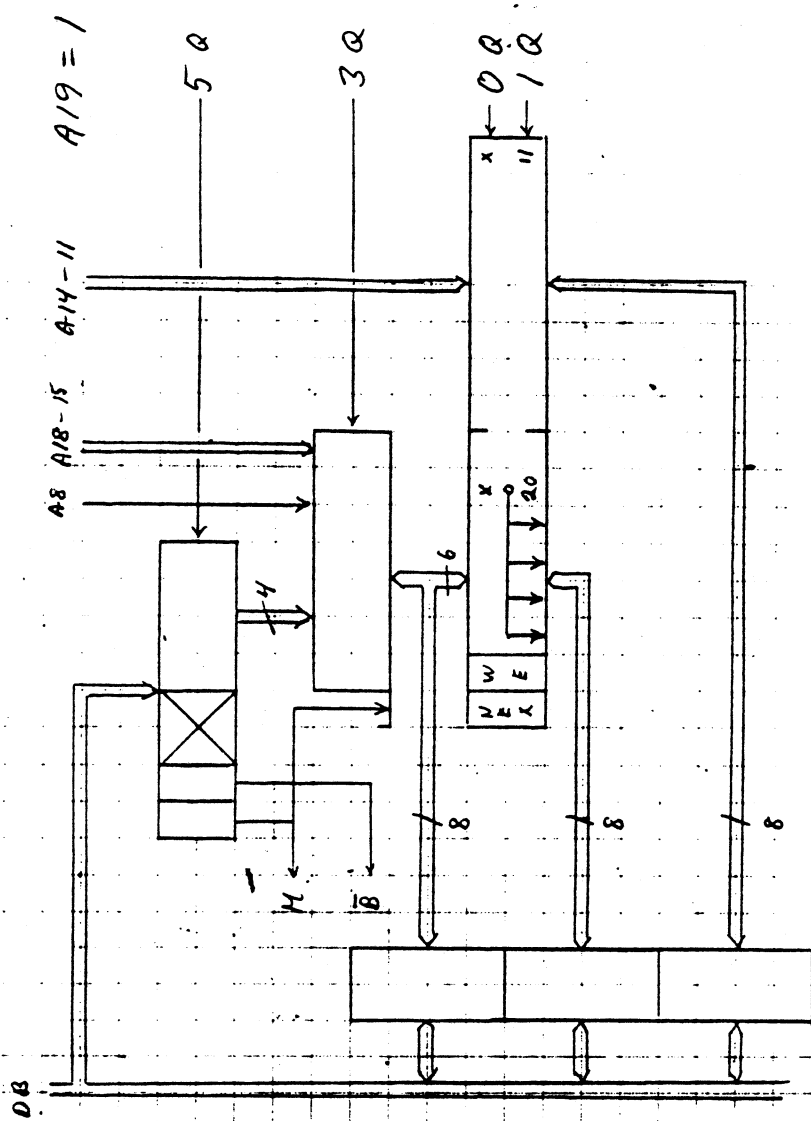




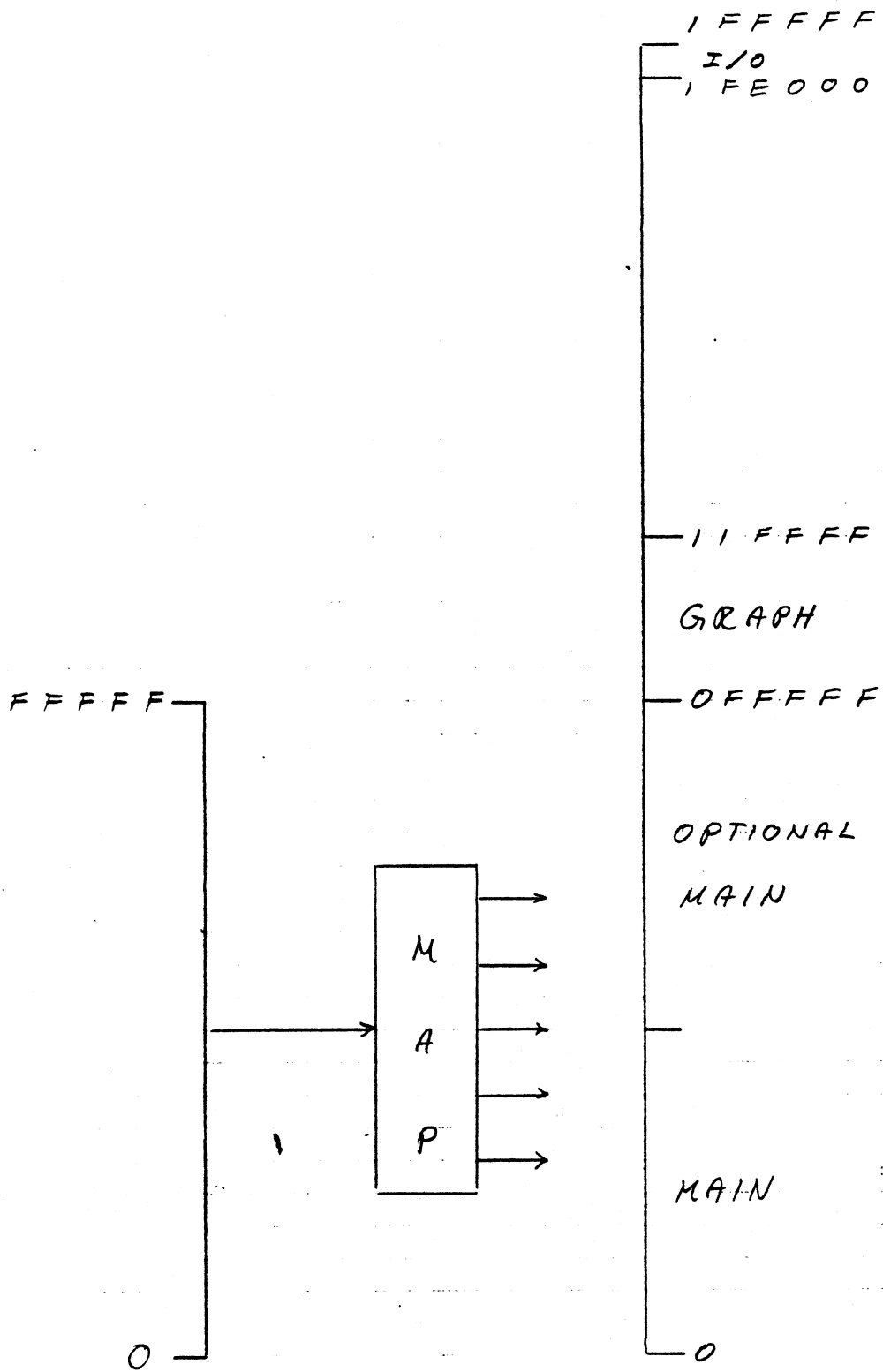
System mode
and
Magic user prog ref.



Normal user mode
and
Magic user data ref.



System mode MAC accesses

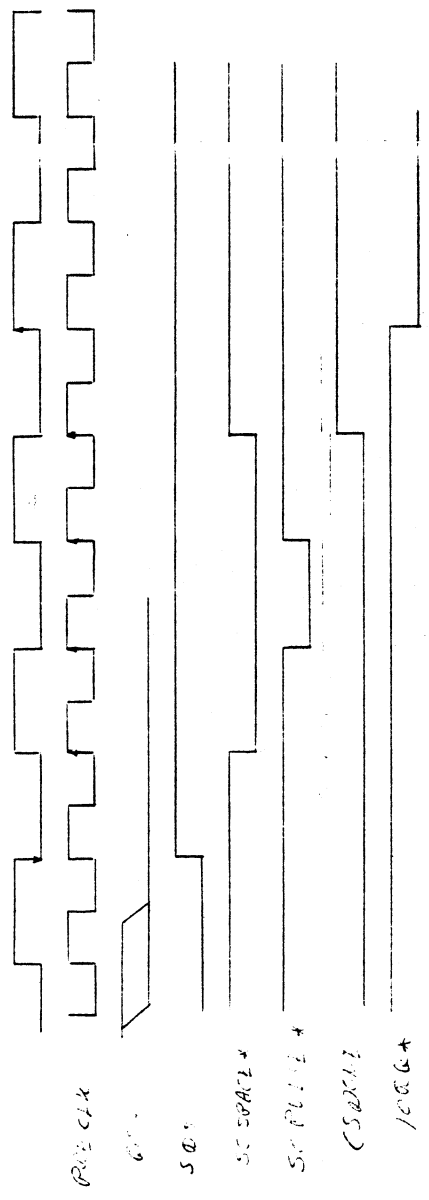


Logical
address space
of 68008

Physical
address space
of X35

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100.

PRECLK = input of CLK. Buffer

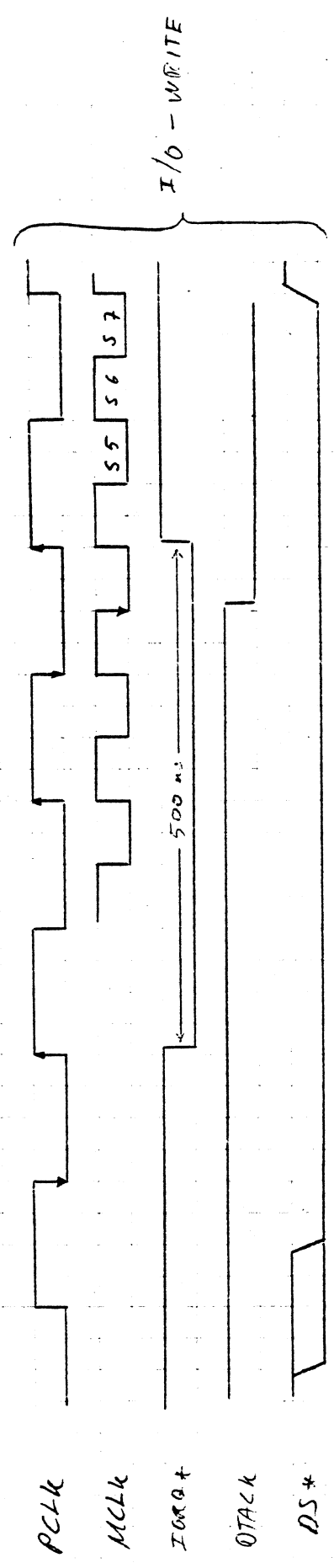
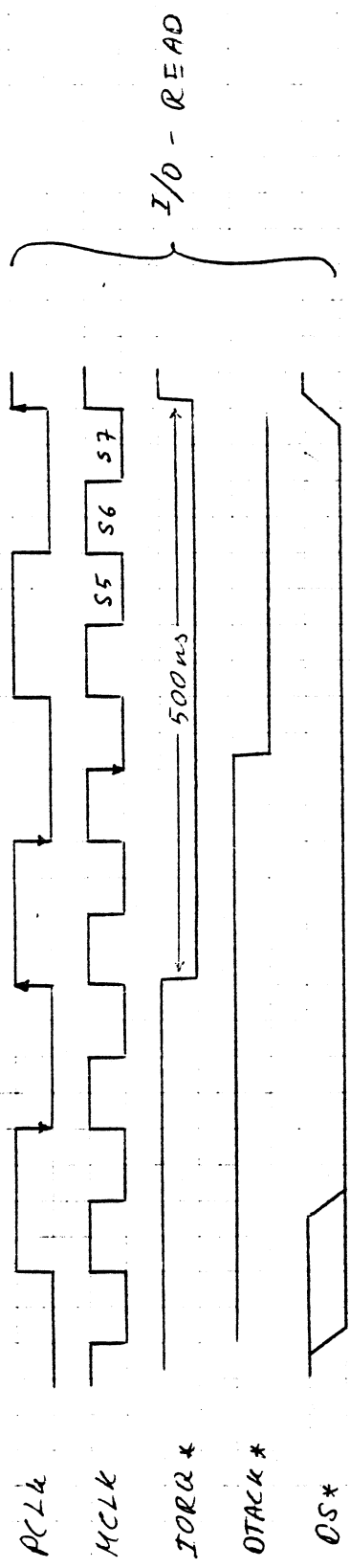


```

/CSBANK = SDS . IOL
SDS . IOL
SDS . IOL
SDS . IOL
CNTL . /CSBANK
/SCSPACE = SDS . IOL . IOL . IOL . /CSBANK . CNTL
/SCPUSET = SCPUSET . /SCSPACE . CNTL
/CNTL = /SCPUSET
  
```

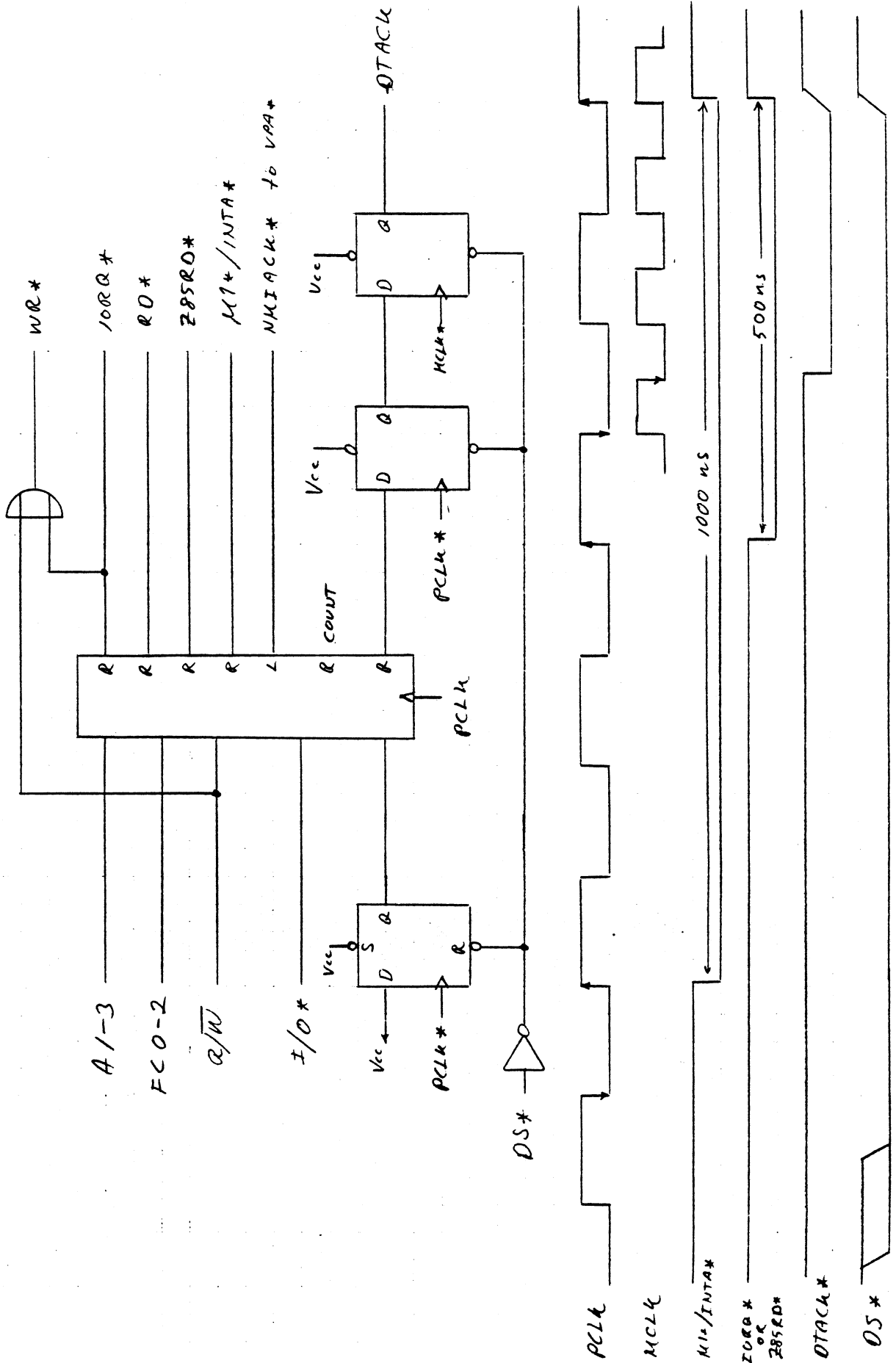
```

/OCG = SDS . /INTL + /EOL = BUSO . /IOL . /X12 . CSBANK . INTL +
SDS . /IOL . X12 /EOL . /IOL . /X12 . CSBANK . INTL +
SDS . /IOL . CSBANK /EOL . /IOL . /X12 . CSBANK . INTL +
/OCG = /OCG . /OCG + /OCG . /OCG
  
```



Z85RD* är alltid lågt vid RD* utan vid INTACK CYCLE

74ALS1032



• DD5 15-0

Display data in

• MAI 15-8

Move data in

• CPU & MAI 7-0

CPU and Move data in

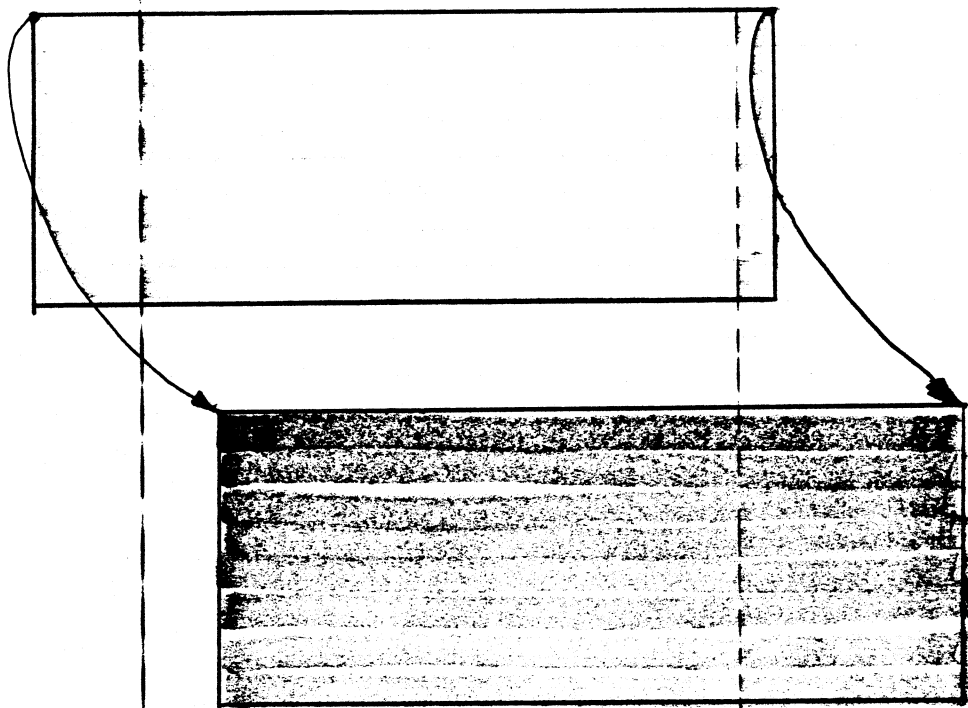
• CPU date 15-8

high byte

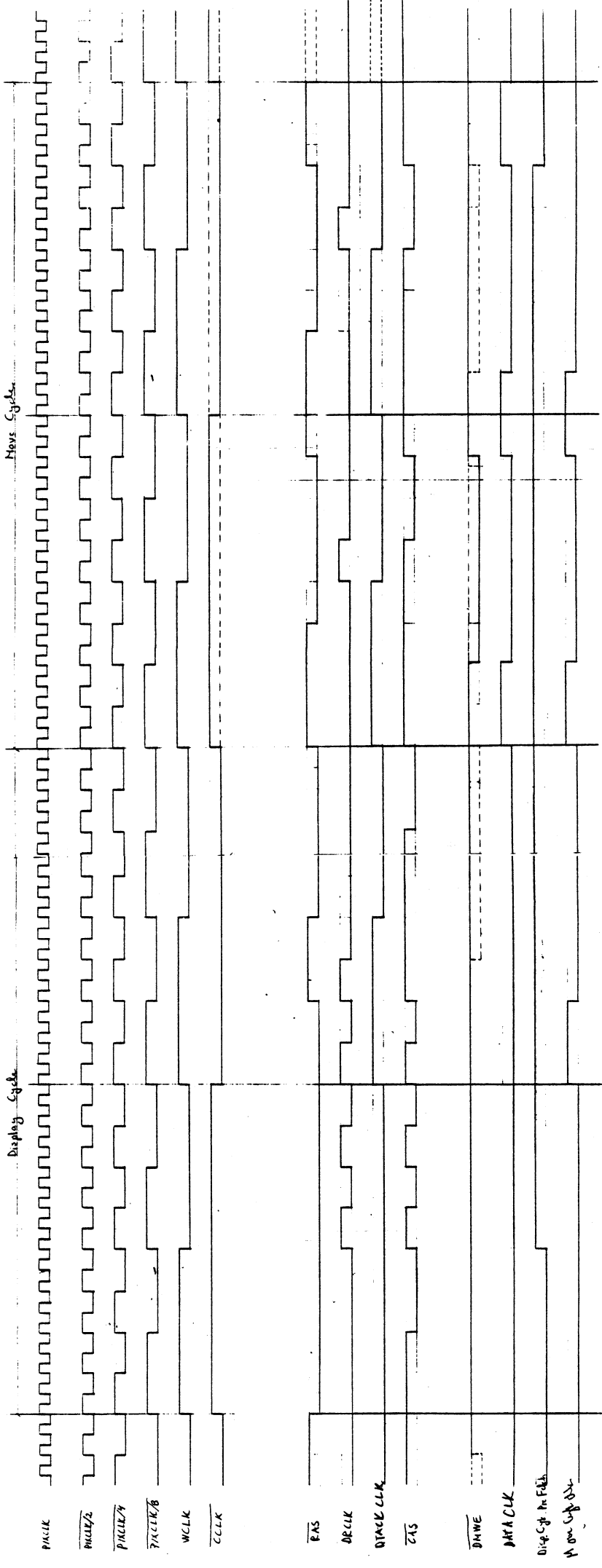
shift = 3



3 4 5 6 7 8 9 10 11 12 13 14 15 0 1 2



Physical word limits

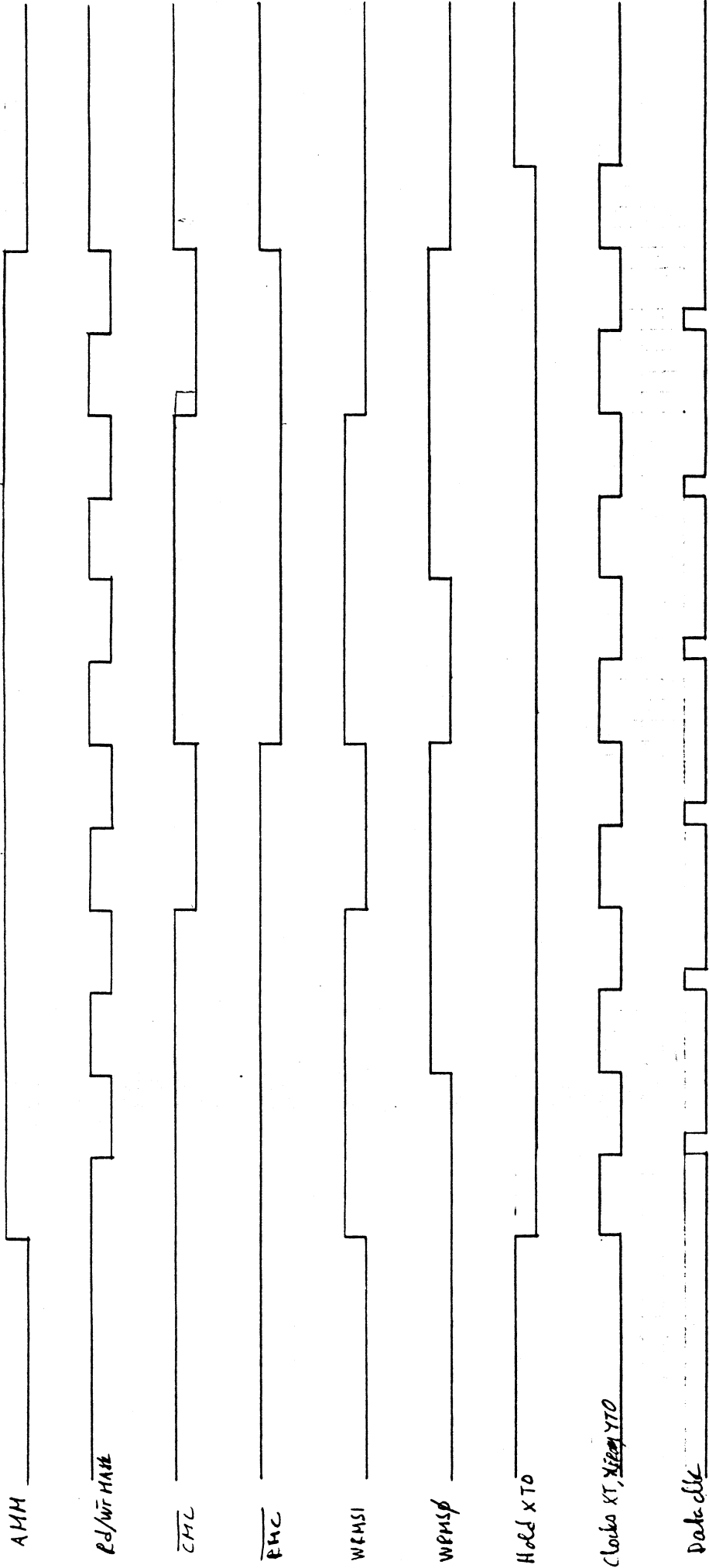


)))))

Timing for counters holds & ends

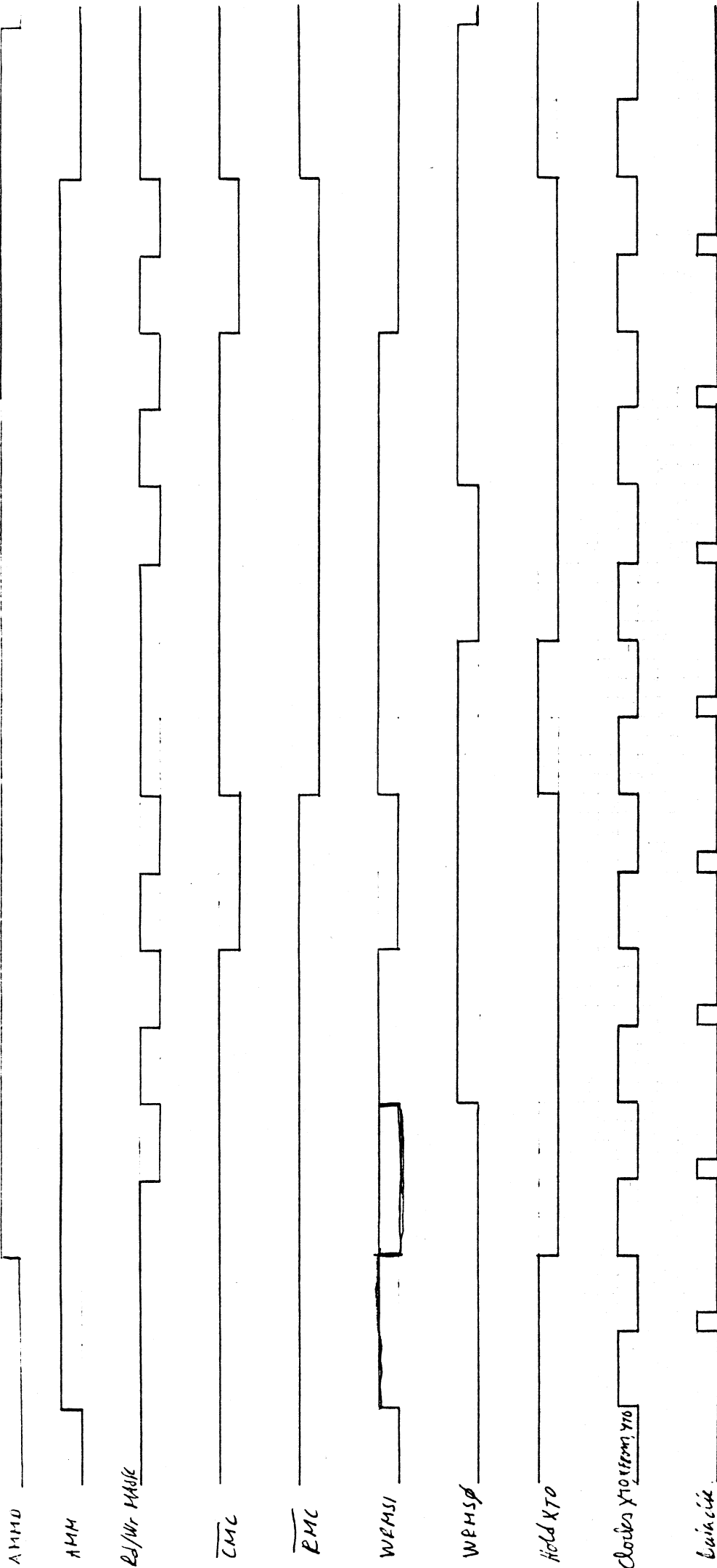


)))
Training for counters (no hold)



Timing for counter

not

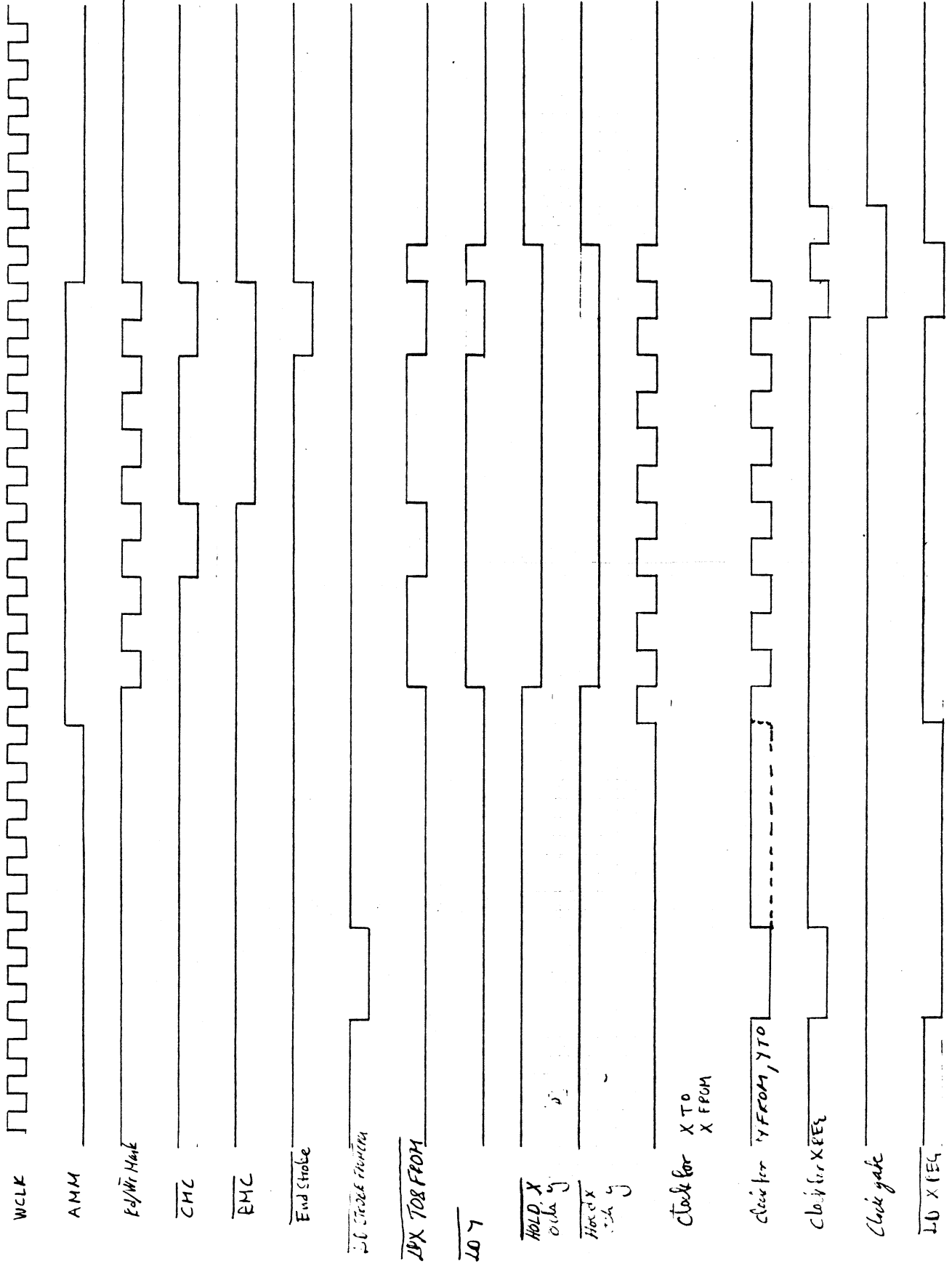


Portrait

Level and Clock timing for M07

840116

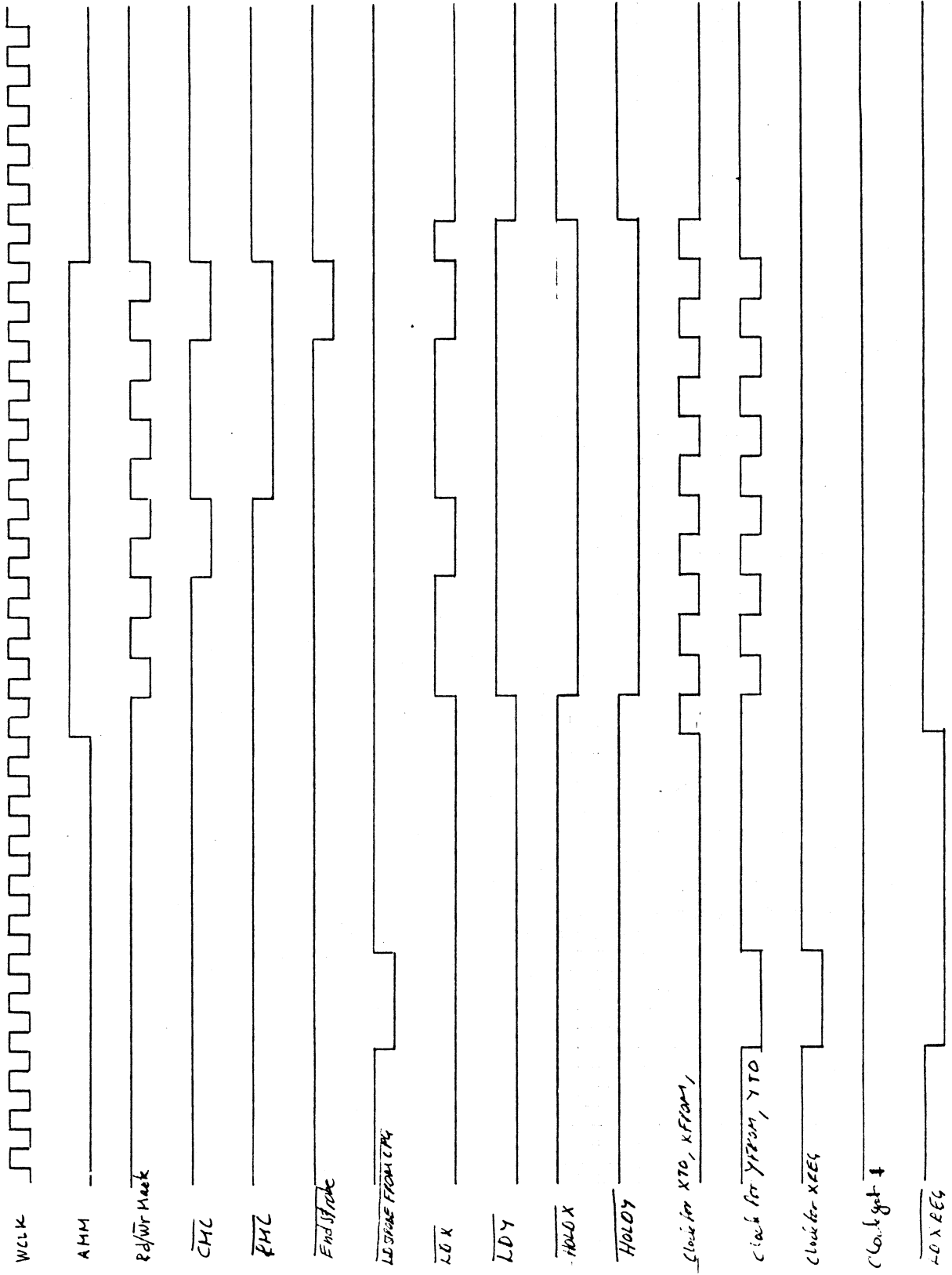
line 1

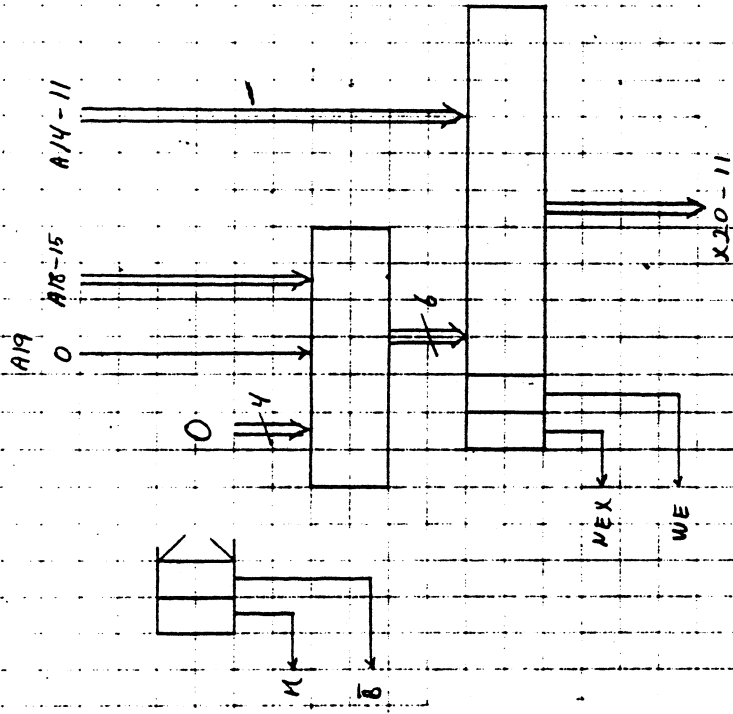


840116

Handwritten Level and Clock timing for M01

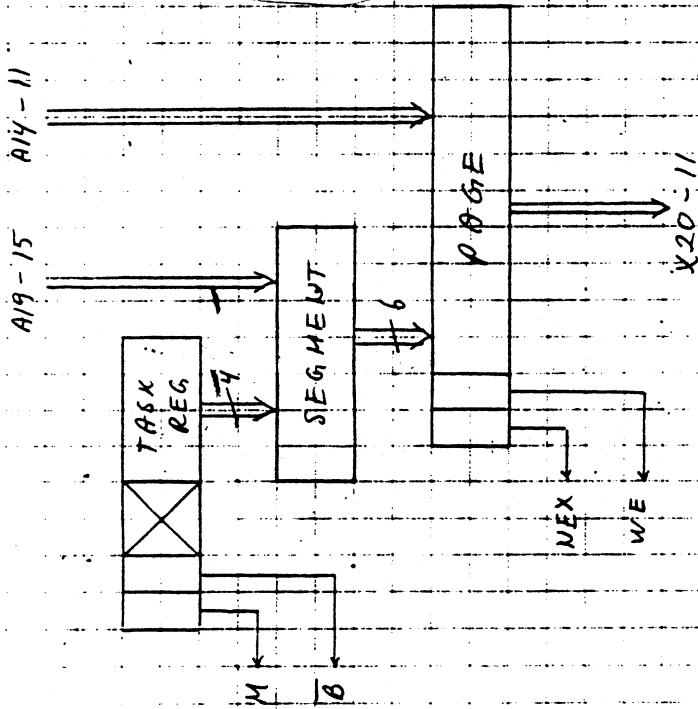
line ϕ / line 1





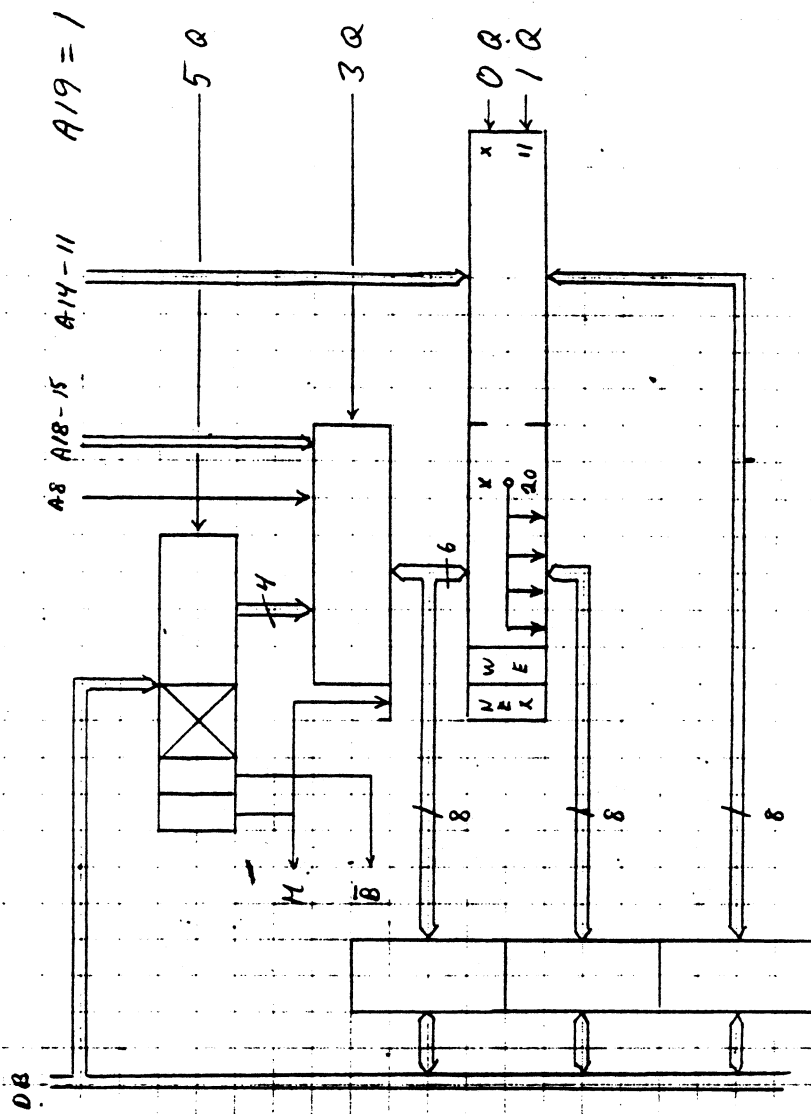
System mode
and

Magic user prog ref.

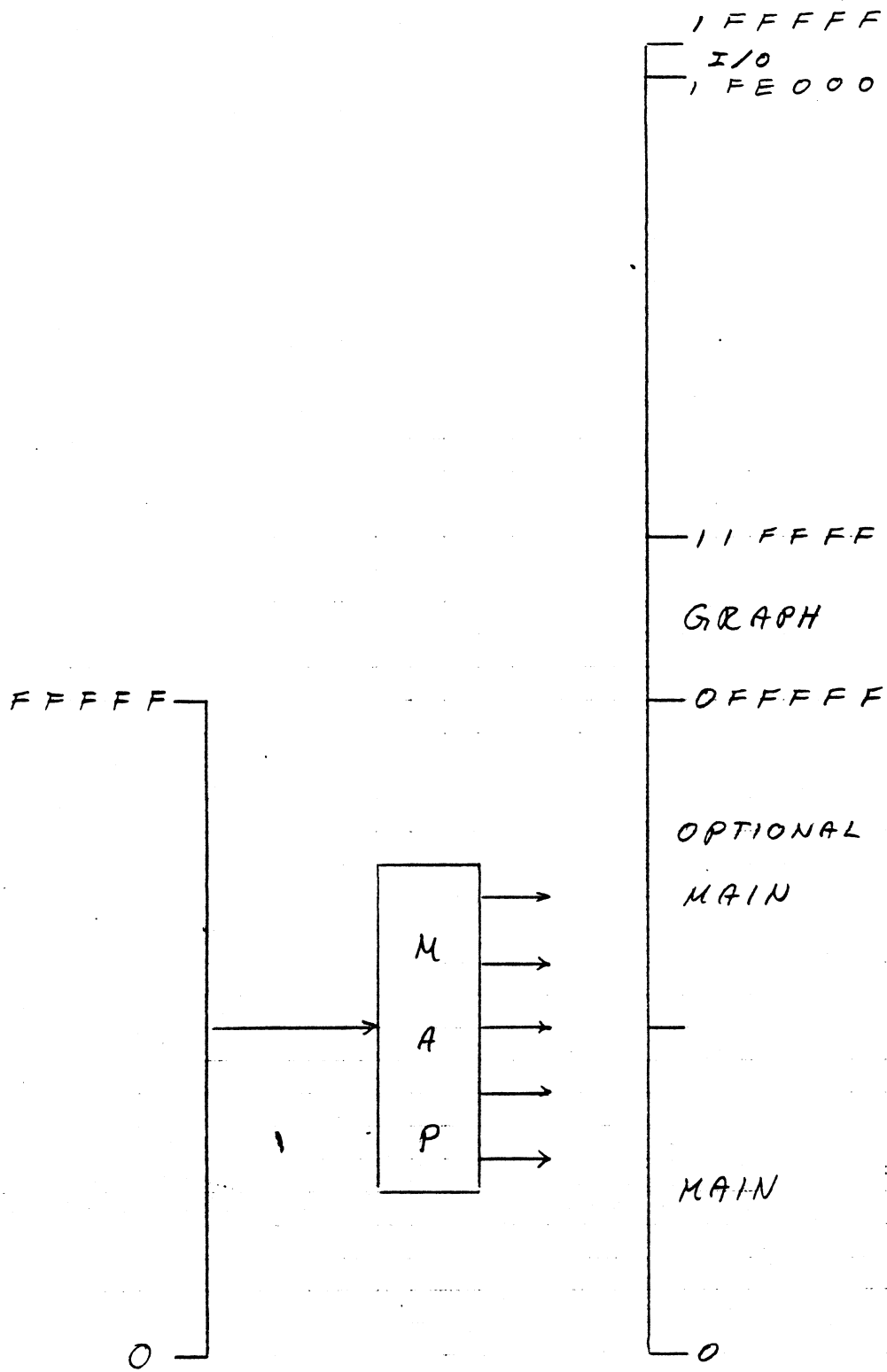


Normal user mode
and

Magic user data ref.



System mode MAC accesses

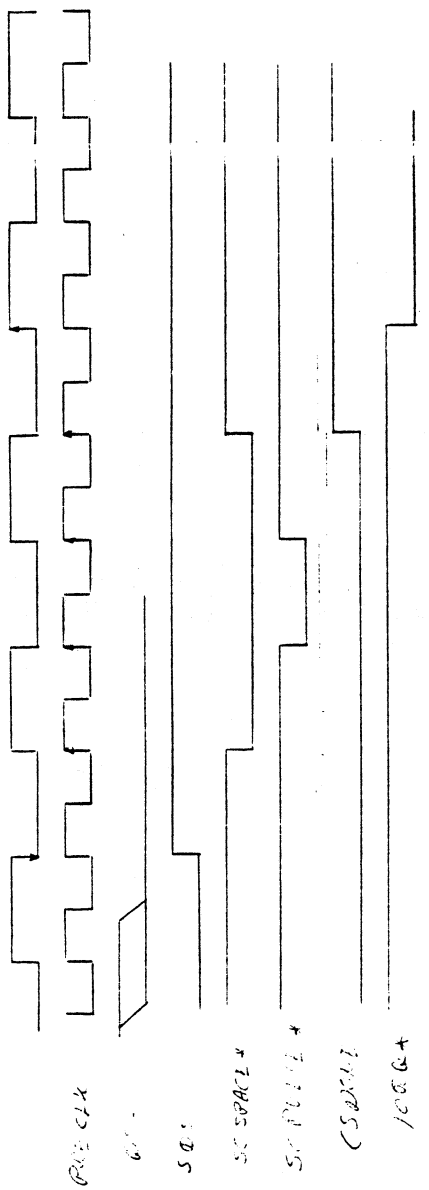


Logical
address space
of 68008

Physical
address space
of X35

CLK = 100 MHz
 CPU = 100 MHz
 BUS = 100 MHz

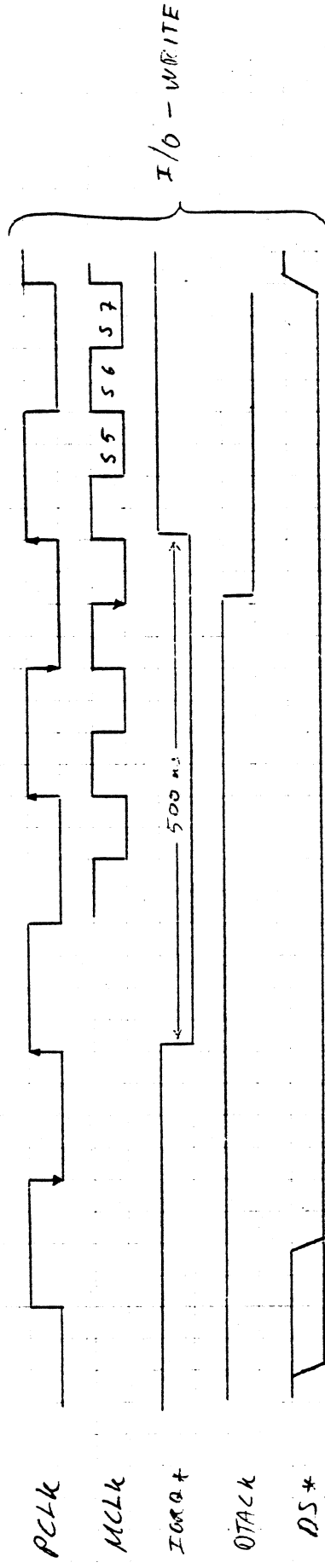
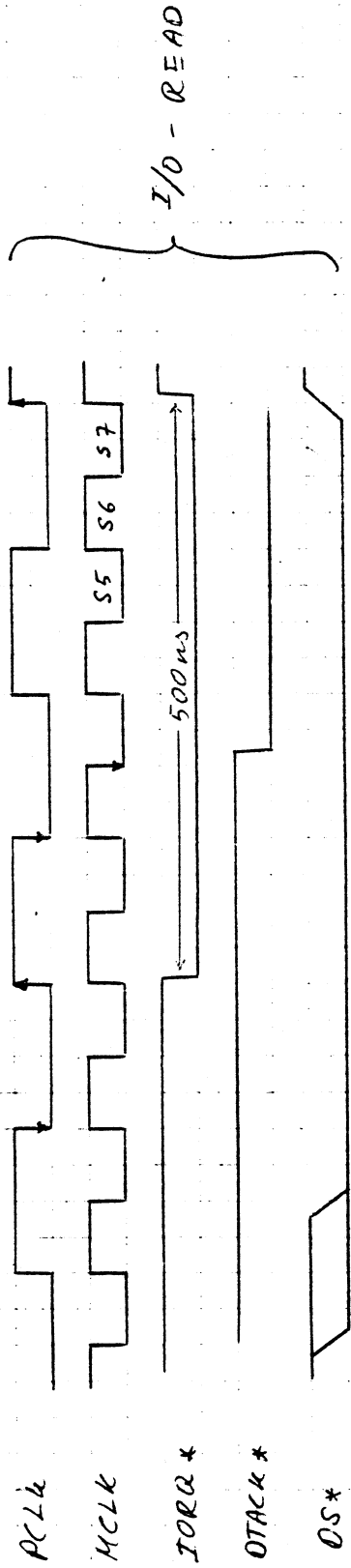
ONE CLK = 10 ns



$$\begin{aligned}
 \text{CPU SPACE} &= \text{SDS} \cdot \text{IO0} / \text{CPU} \cdot \text{INTL} \cdot \text{CPU} \cdot \text{SPACE} \cdot \text{CNTL} \\
 \text{CPU PULSE} &= \text{SCSPACE} / \text{SCSPACE} \cdot \text{CNTL} \\
 \text{CNTL} &= \text{TEST} \cdot \text{SPACE} / \text{CNTL}
 \end{aligned}$$

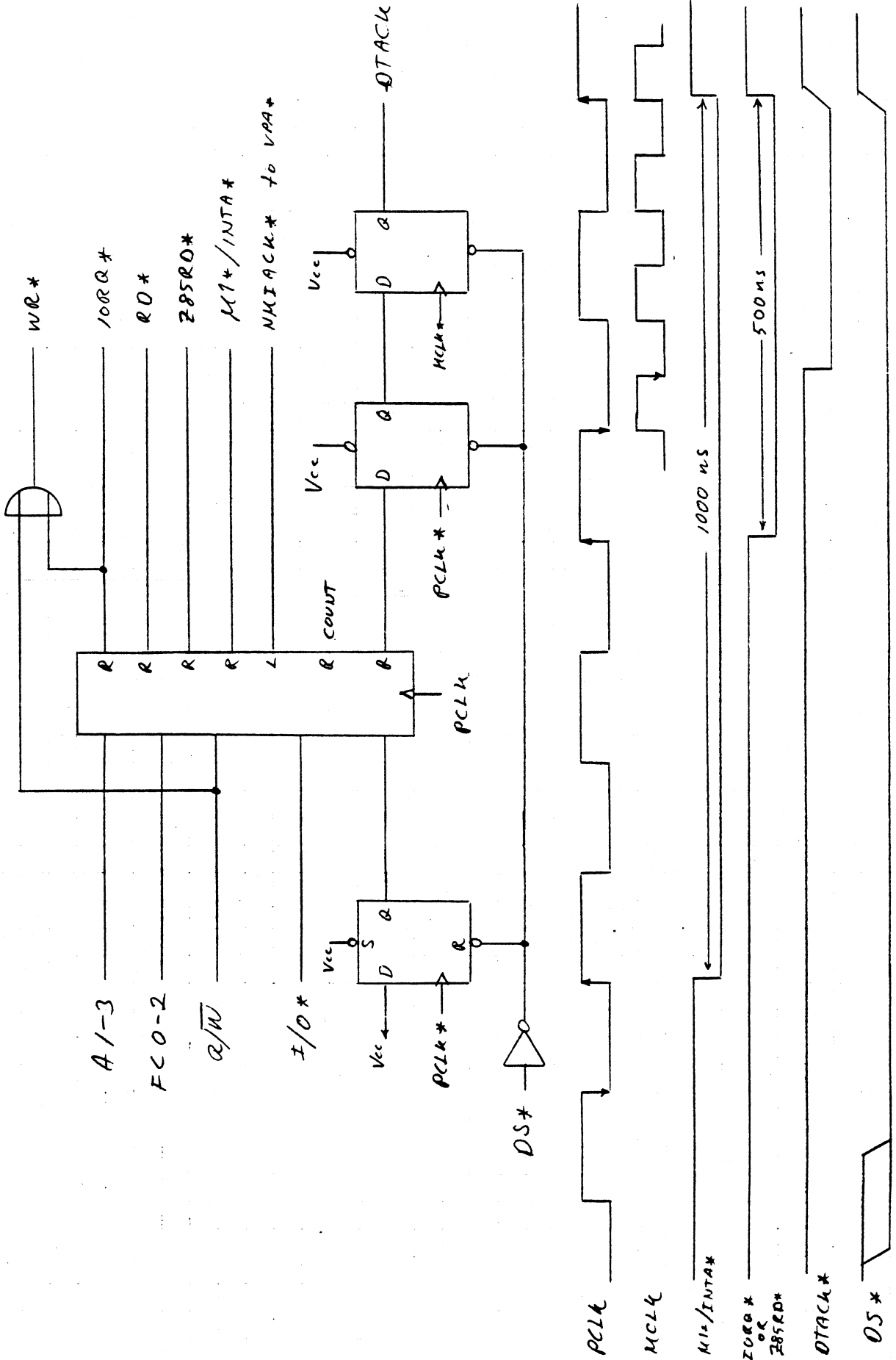
$$\begin{aligned}
 \text{CPU SPACE} &= \text{SDS} \cdot \text{IO0} \cdot \text{INTL} + \text{EOL} = \text{BUS0} \cdot \text{IO0} \cdot \text{X12} \cdot \text{CSDONE} \cdot \text{INTL} + \\
 &\quad \text{SDS} \cdot \text{IO0} \cdot \text{X12} \cdot \text{SPACE} \cdot \text{CNTL} \\
 \text{CPU PULSE} &= \text{SDS} \cdot \text{IO0} \cdot \text{CSDONE} \cdot \text{INTL} + \text{EOL} \\
 &\quad \text{SDS} \cdot \text{IO0} \cdot \text{X12} \cdot \text{SPACE} \cdot \text{CNTL}
 \end{aligned}$$

$$\begin{aligned}
 \text{CPU SPACE} &= \text{SDS} \cdot \text{IO0} \cdot \text{X12} \\
 \text{CPU PULSE} &= \text{SDS} \cdot \text{IO0} \cdot \text{X12}
 \end{aligned}$$



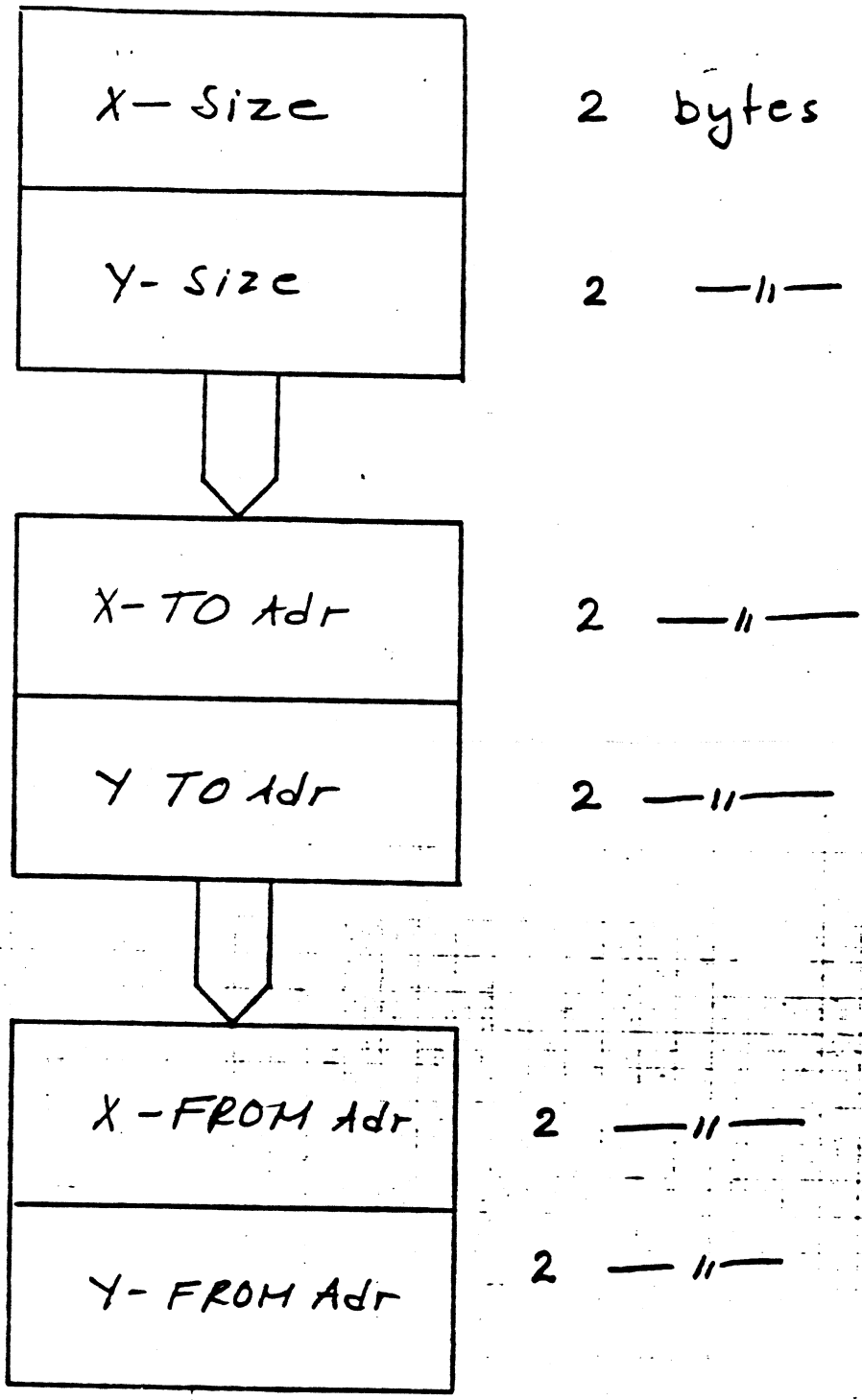
Z85RD* är alltid lågt RD* utan vid INTRACK CYCLE

74ALS1032



Blockmove

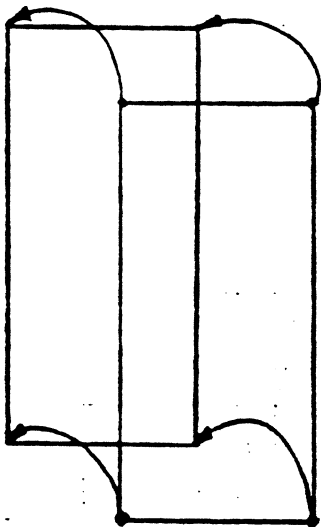
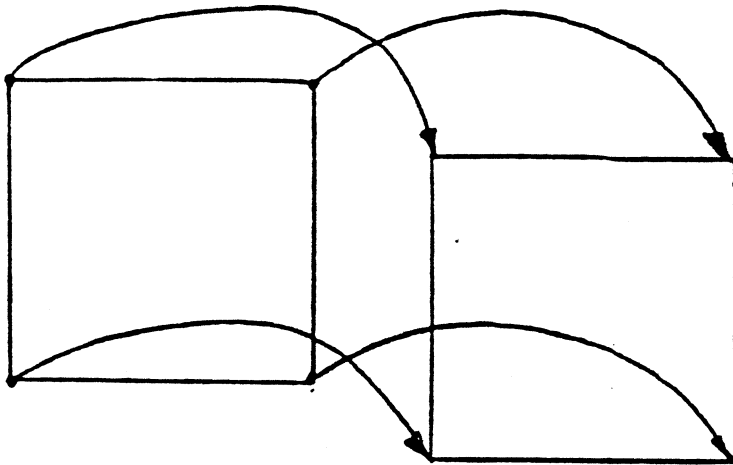
Command sequence



total 12 bytes

Block move

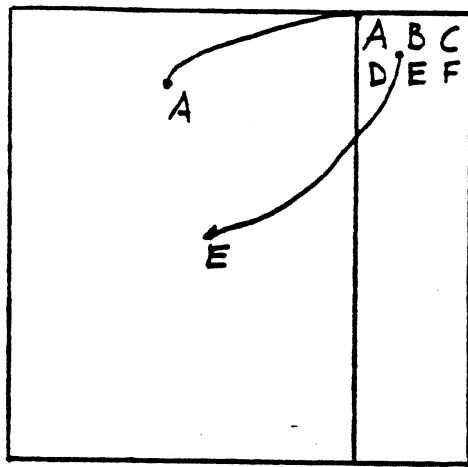
(14)



- Move kan göras på rektangulära ytor.
- Move kan göras i valfri riktning.
- Adresserna kan räknas uppåt eller nedåt
⇒ man kan överlappa block utan att skriva sänder data som skall läsas

Blockmove

- De rektangulära blockens storlek, source och destination bestäms med pixelupplösning i såväl x, som y-led.
- När man arbetar i karaktärsmode ligger karaktärsgeneratorn lagrad i den dolda delen av grafikminnet och kopieras in i den synliga delen på beställning.



- Efter varje move autoinkrementeras räkningar till nästa karaktärs position på raden.
- Blocksize anges en gång / karaktärs generator

TO Adr anges en gång / rad

FROM Adr anges en gång / karaktär (symt)

Detta ger att man för varje tecken inom en rad endast behöver leverera FROM Adr = 4 bytes och TO Adr = 4 bytes för varje CR.

Graphic Commands

Line

Circle

Fill
(Rectangular)

Paint
(Polygon)

Graphic Commands

- De grafiska kommandona beräknas helt i mjukvaran.
- Minnesoperationerna är

REPLACE

SET

RESET

Mjukvara
Blockindelning

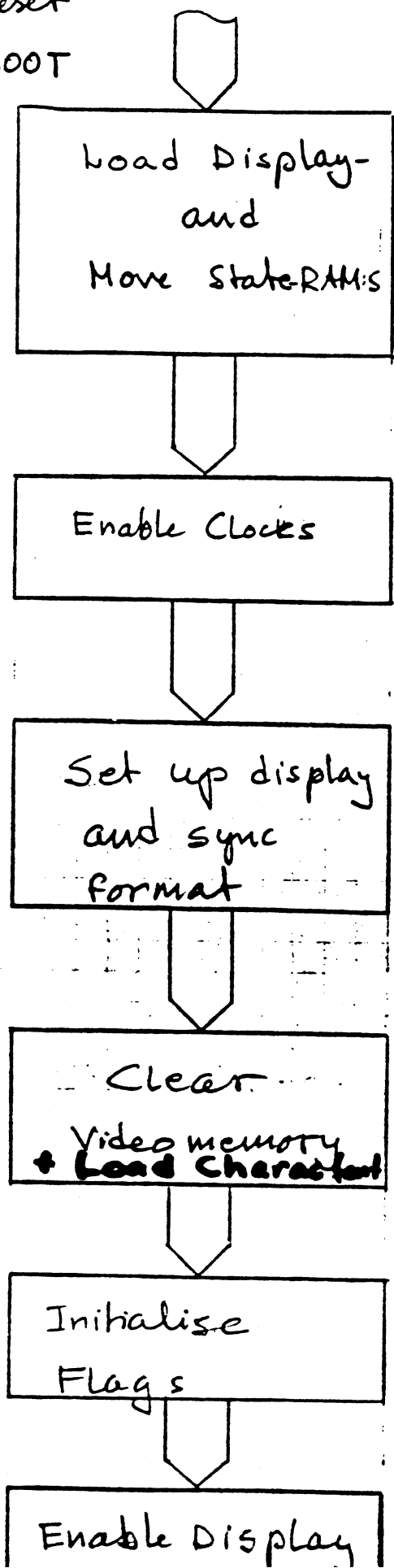
Initialisation

Blockmove

Graphic
Commands

Initialisation

From Reset
and BOOT



X35 - Grafiken

①

• Hårdvara

Blockindelning

- Osc (Oscillator och klocksignaler)
- CRTC (CRT-controller)
- Cycle Control (Logik som styr timing)
- Video
- VMemory (Video Memory)
- Mover

• Mjukvara

Blockindelning

- Initialisation
- Blockmove
- Graphic Commands

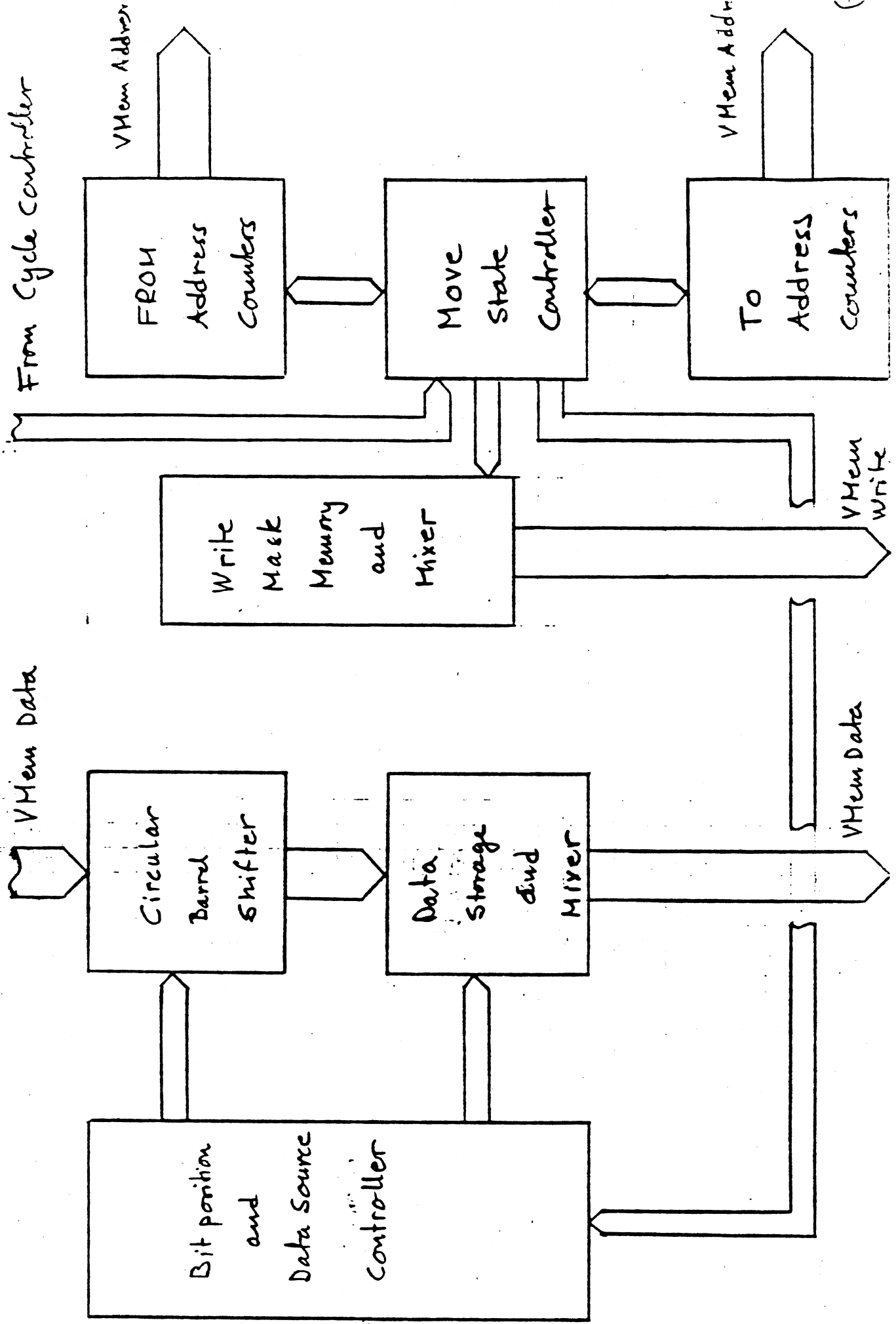
Blockmove

(16)

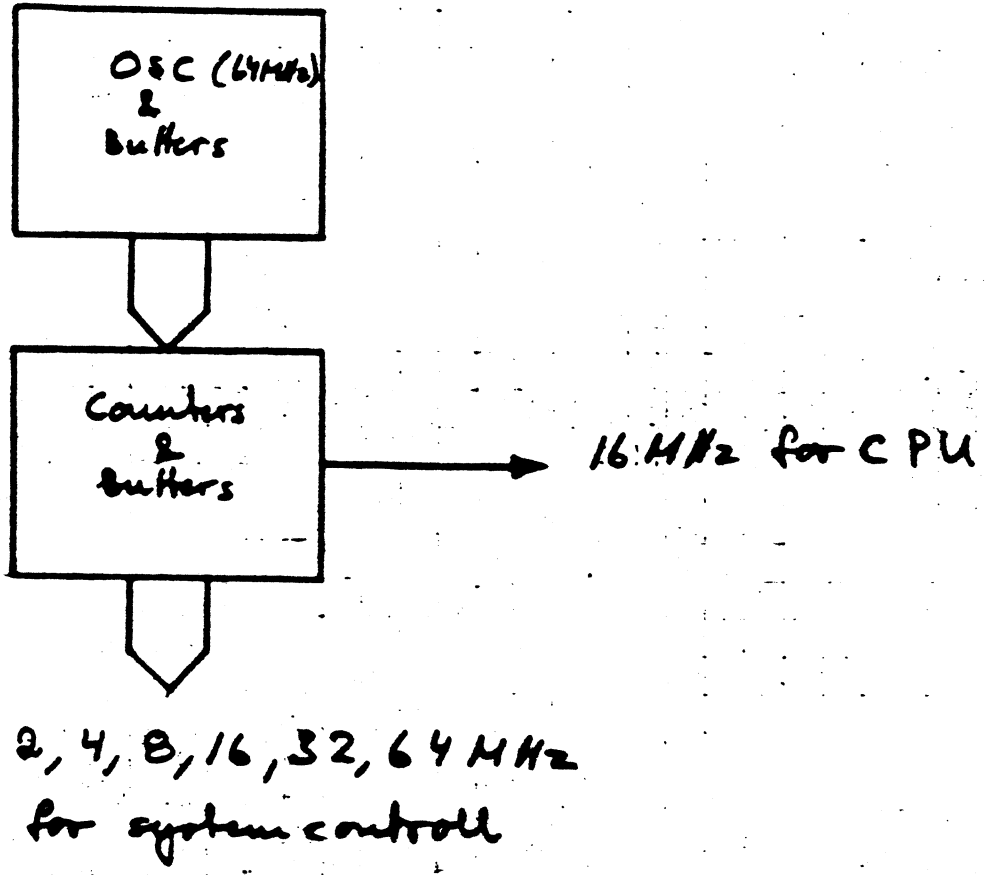
- Minnesoperationen vid move är REPLACE
Detta innebär att destinationdata helt
ersätts av source-data inom blockets rama
- Blockmove kan endast göras i grafik-
minnet alltså inte mellan grafik-
och CPU-minnet.

- Grafikminnet är av treports typ.
Detta innebär att minnet accessas från
 - Displayenheten
 - Blockmovern
eller
 - CPU:nCyklerna är nämnda i prioritetsordning
- CPU-accessen är av konventionell typ.
- Display-accessen utnyttjar nibblemode, d.v.s. vill säga snabb utläsning av 4 konsekutiva ord, för att skära ned displayaccesser tid.
- Blockmovraccesserna är av Read-Modify-Write typ men med Read-och Write-data på skilda adresser.

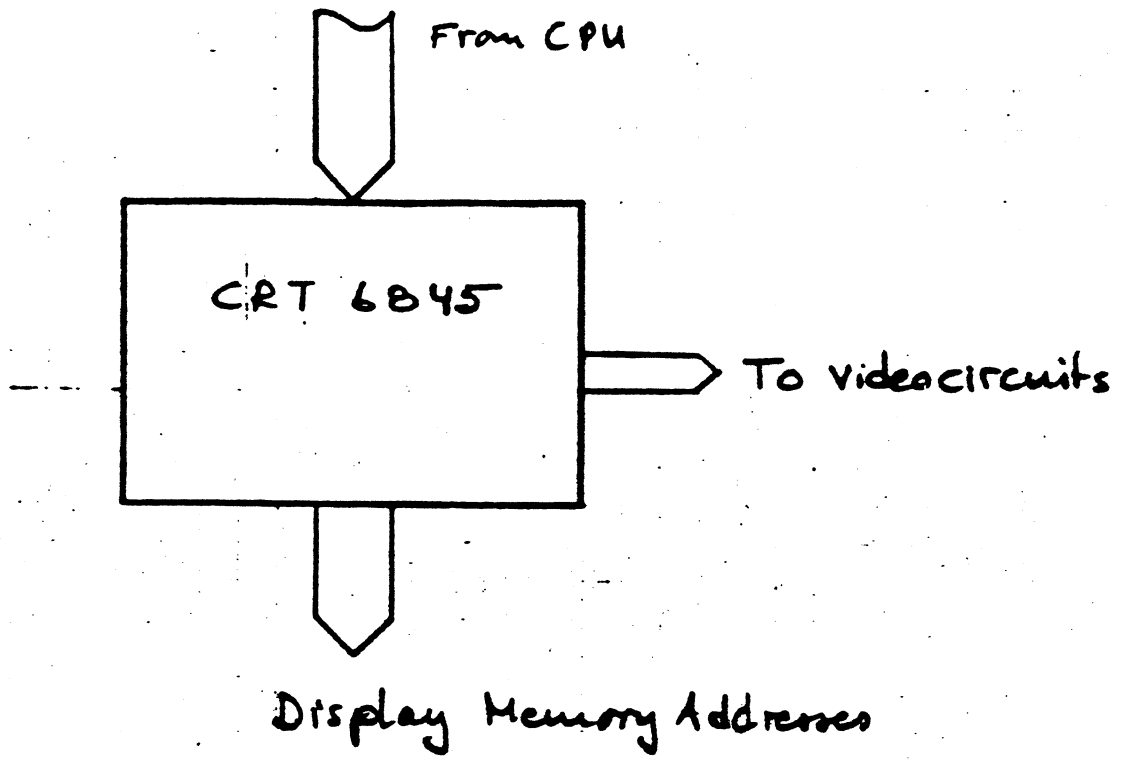
Mover



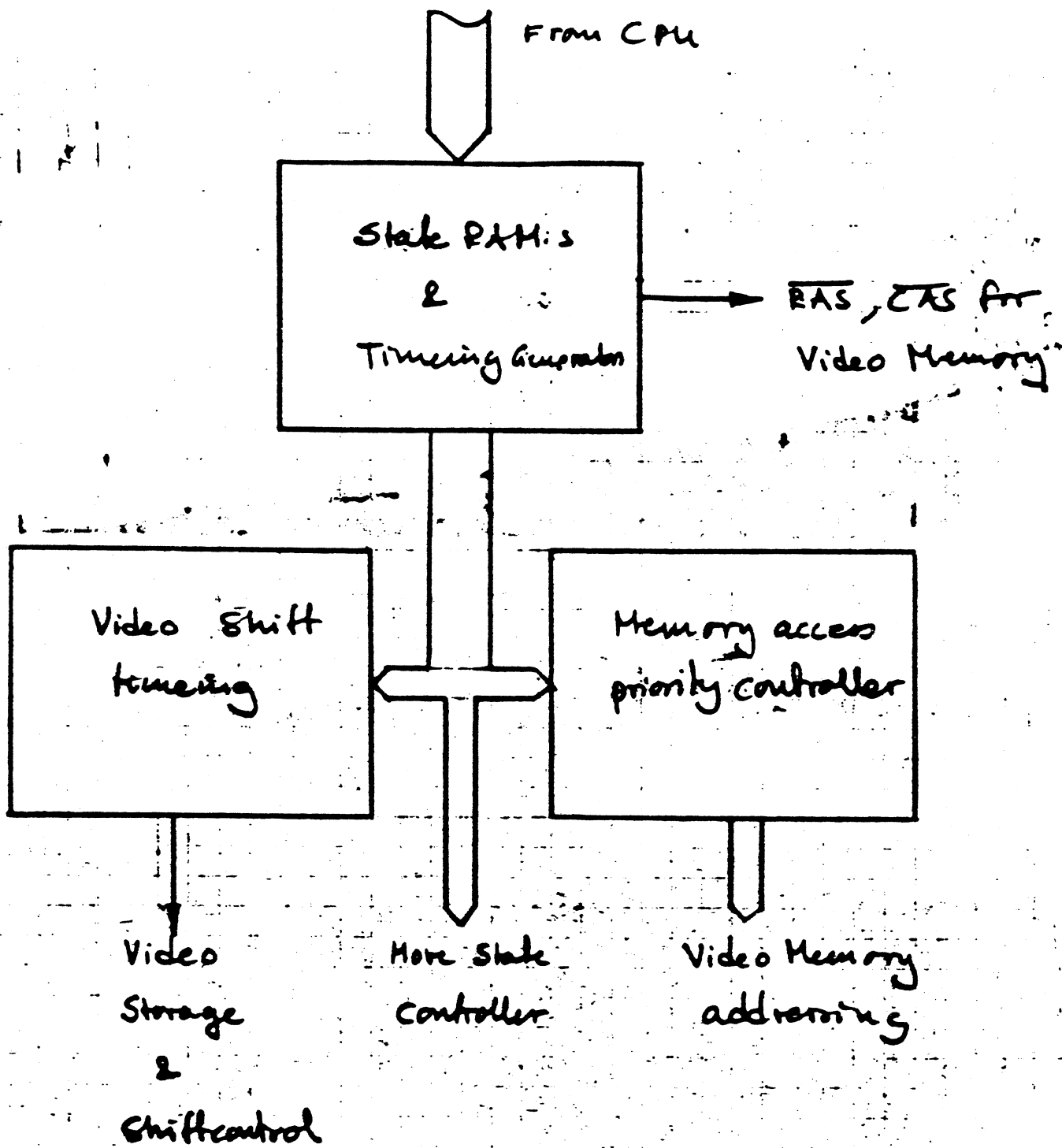
OSC



CRTG

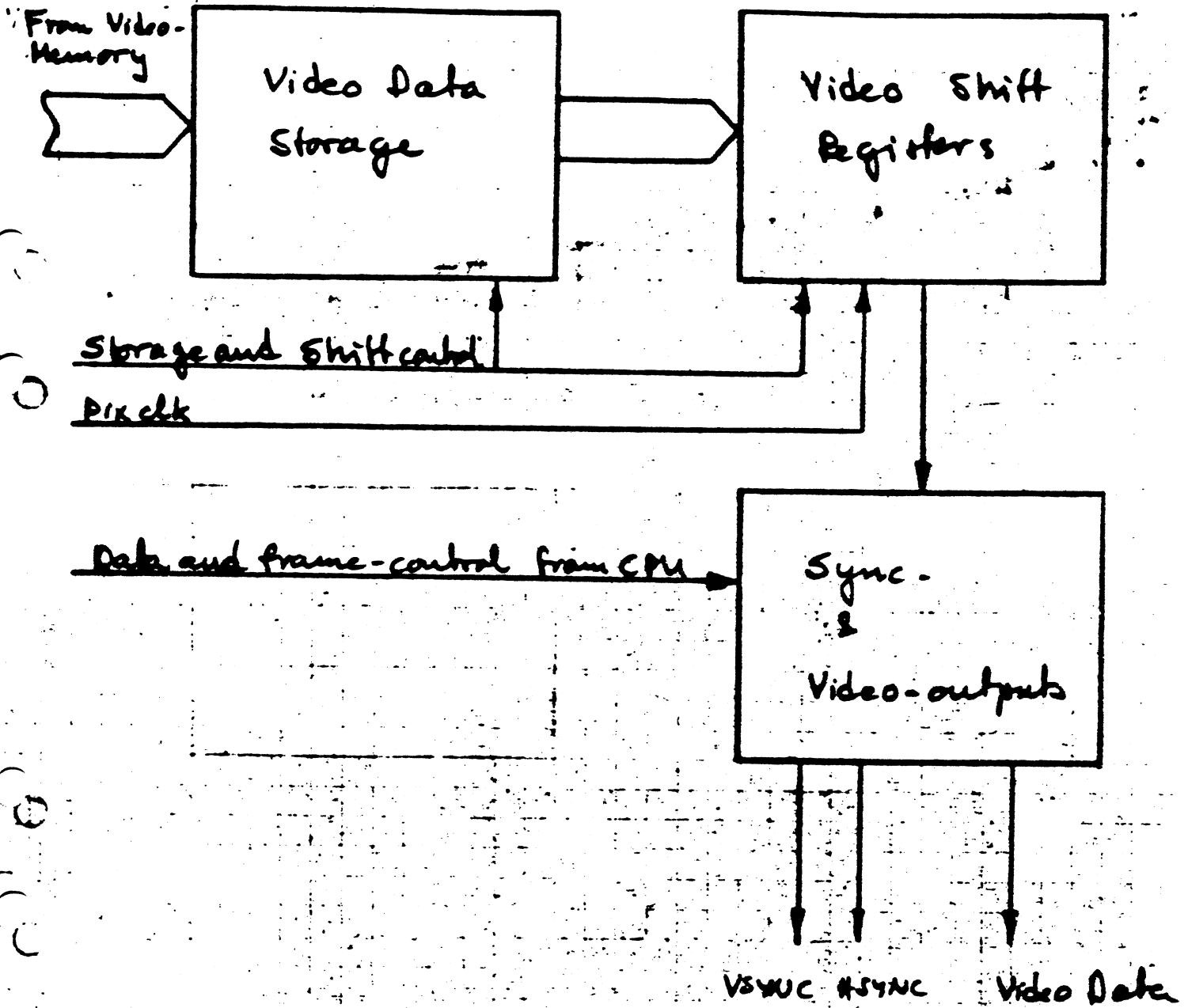


Cycle Control

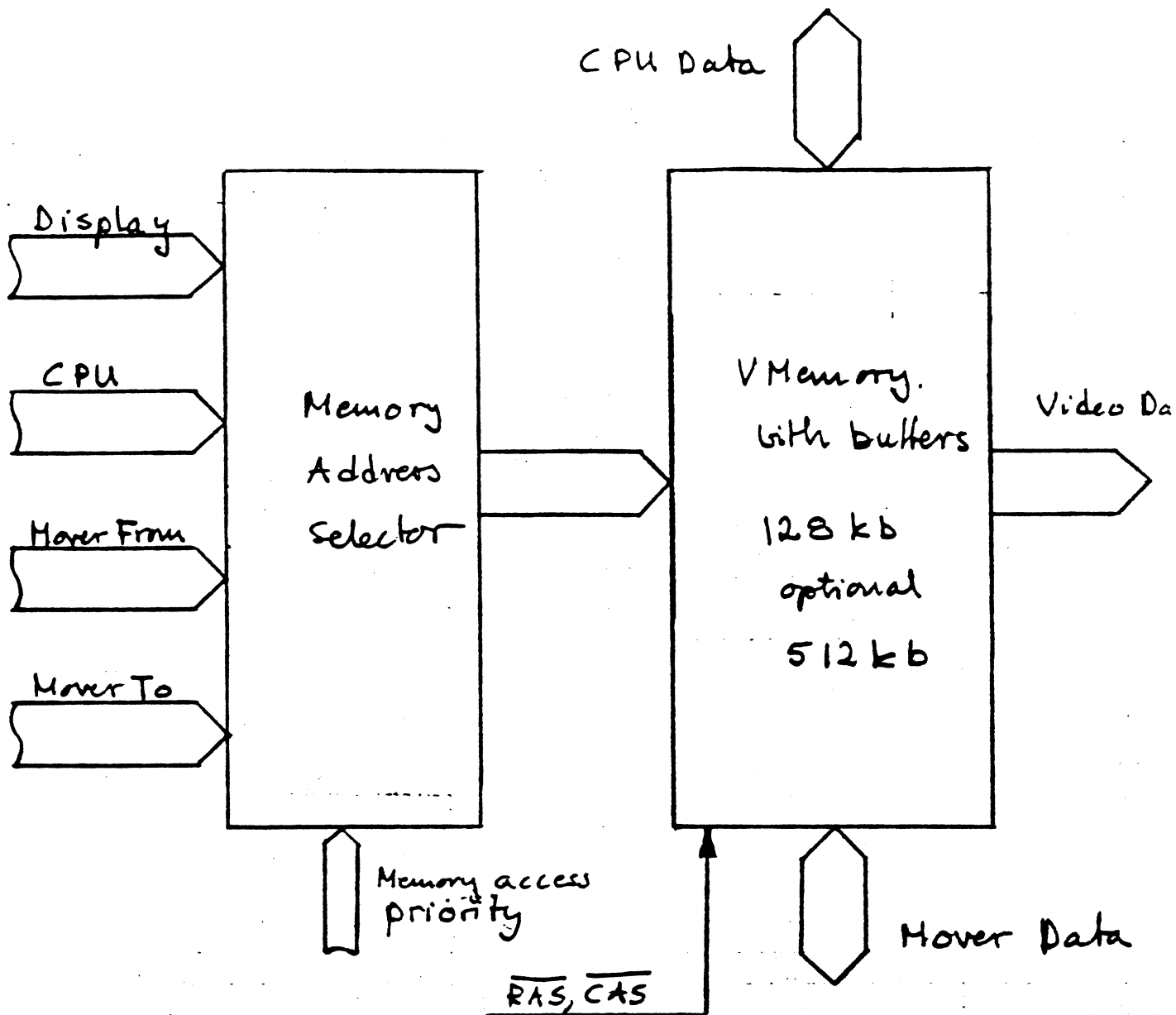


Video

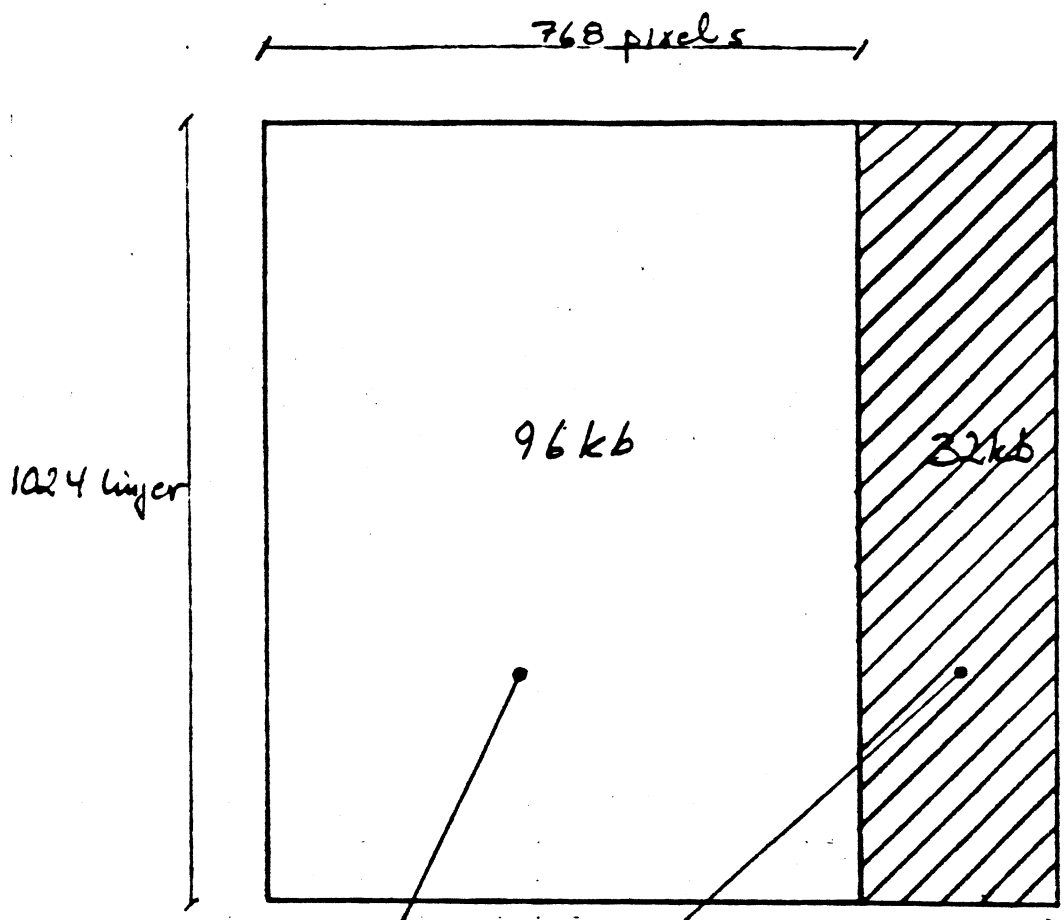
5



Video Memory



Upplösning



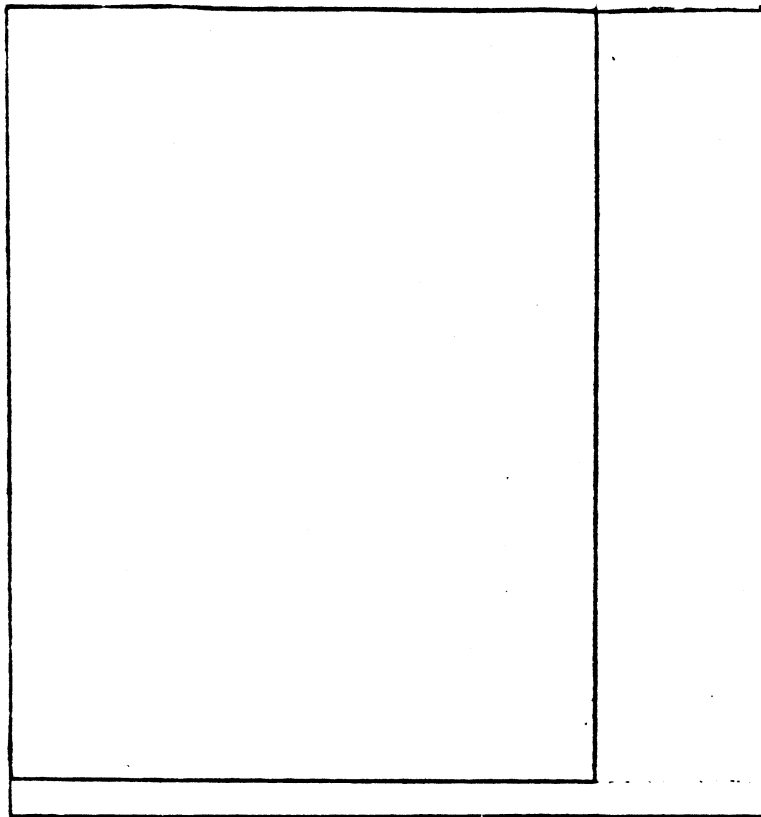
Del av minnet som inte syns på skärmen
I denna del ligger karaktärsgenerator
och ev. grafiska symboler.

Del av minnet som visas på skärmen.

Block movers Prestanda

(10)

Visade punkter Släckta

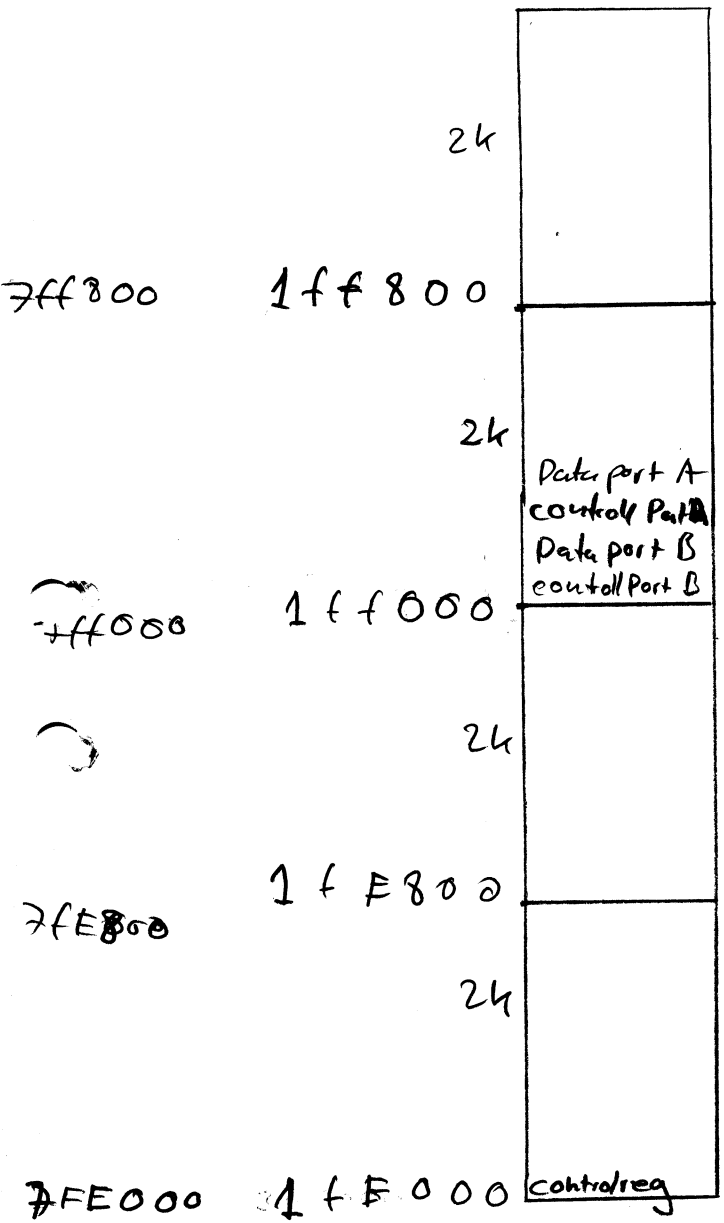
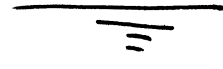


- Movern flyttar 24 M pixels/s under visnings tid och 32 M pixels/s under strålåtergå tid.
- Visnings tiden är 74% och släcktiden 26% av totala tiden.
- Detta ger att movern hårdvarumässigt kan flytta 26.1 M pixels/s eller flytta skärmens innehåll 33.2 ggr/s.

Logisk

fysisk (etter MACEN)

(* I/O minnes area *)



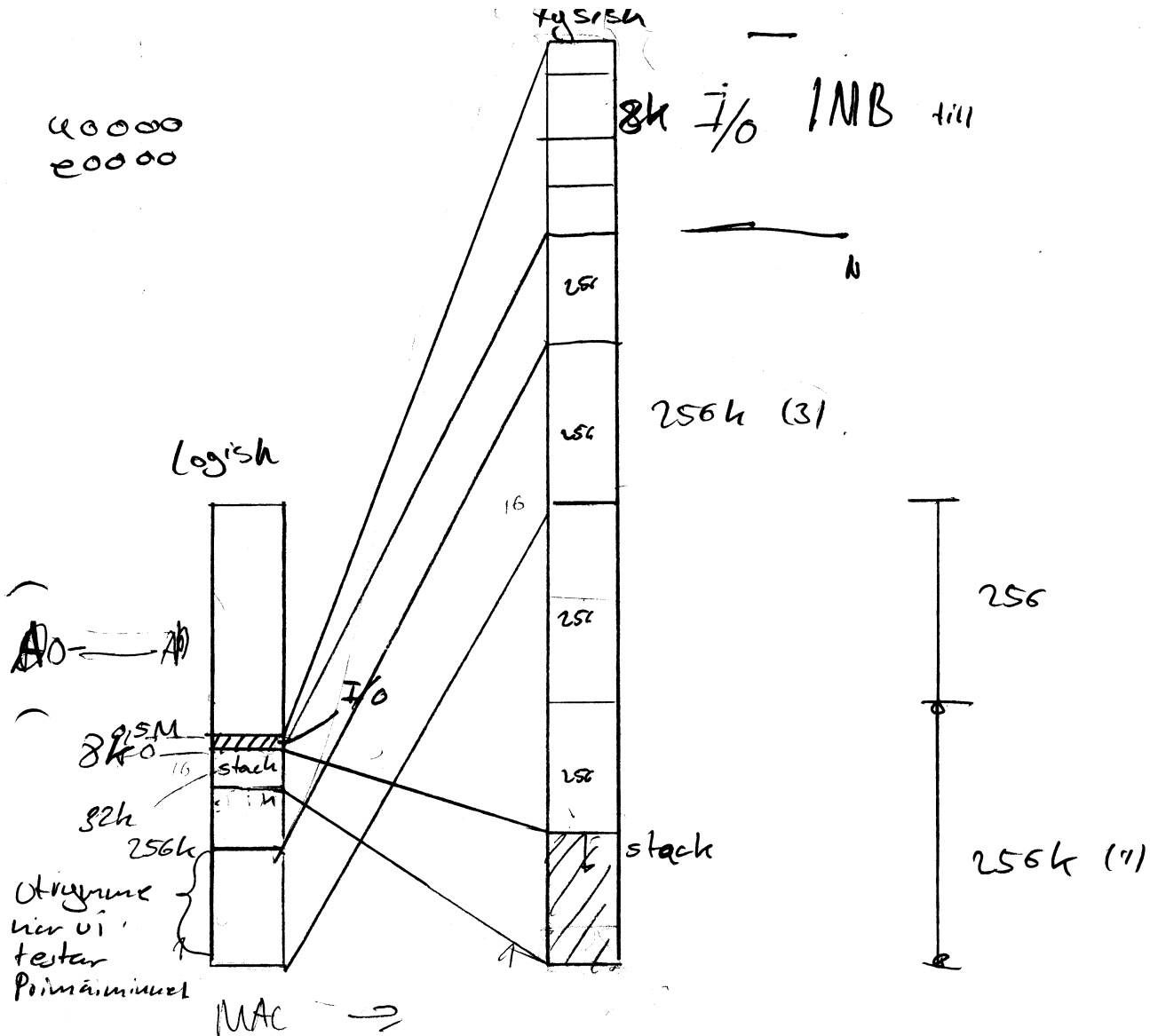
7F206
7F204
7F202
7F200

{ A1 o A2
Styr vad
som skall göras

instruering i Maccen :

| A | | D |
|---------------|---|------|
| FE000 (7E000) | → | 43FC |
| FE800 (7E800) | → | 43FD |
| FF000 | → | 43FE |
| FF800 | → | 43FF |

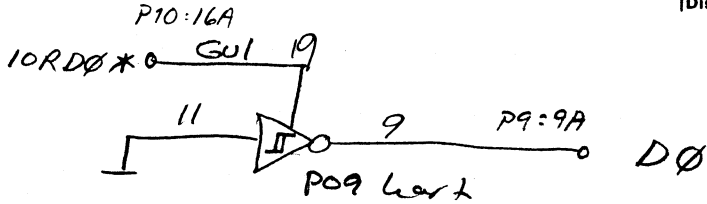
Macces



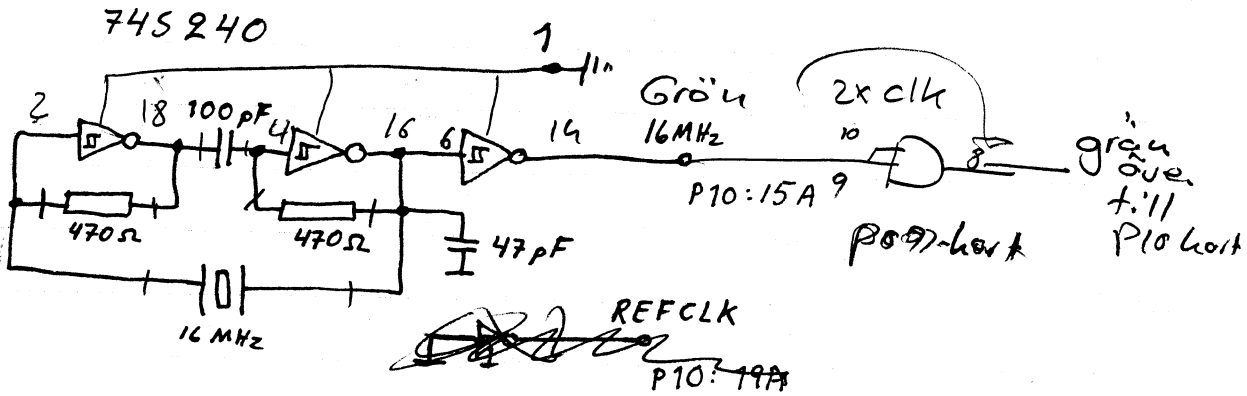
| Stacken | Logish adress | fysisk |
|---------|-------------------------------|---------------|
| 32k | 60000 → 68000 | 00000 → 08000 |
| | (skrivs in med e0000 → e8000) | |

| I/O | 7EFFF | ⇒ | 1FE xxx | Adress | 1 |
|-----|------------------|---|--------------------|--------|--------|
| | 7EFFF | ⇒ | 1FF xxx | 18000 | EROMER |
| | 7F7FF | ⇒ | 200 xxx | 188000 | 3 |
| | 7FFFF | ⇒ | 201 xxx | 190000 | 4 |
| | | | | 198000 | trunk |

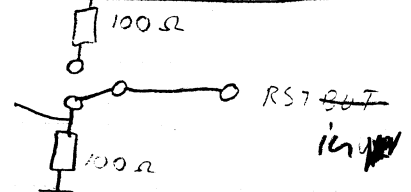
| | | |
|---|-------------------------------|------------------------------|
| Från/From
<i>RJ 232L</i> | Datum/Date
<i>84-11-27</i> | Beteckning/Reference |
| Rubrik, ärende/Subject
<i>55 11148-01 Hjälpkort X-35</i> | Ert datum/Your date | Er beteckning/Your reference |
| | Gäller fr o m/Effective date | Ersätter/Replaces |
| Distribution | | |



+5 P9:13A



+5V t:11 +5 P10:19A



55 11148-01 Hjälpkort X-35 (mk)

leset



€ 344L

123 Dual retriggerable monostable
multivibrators with clear

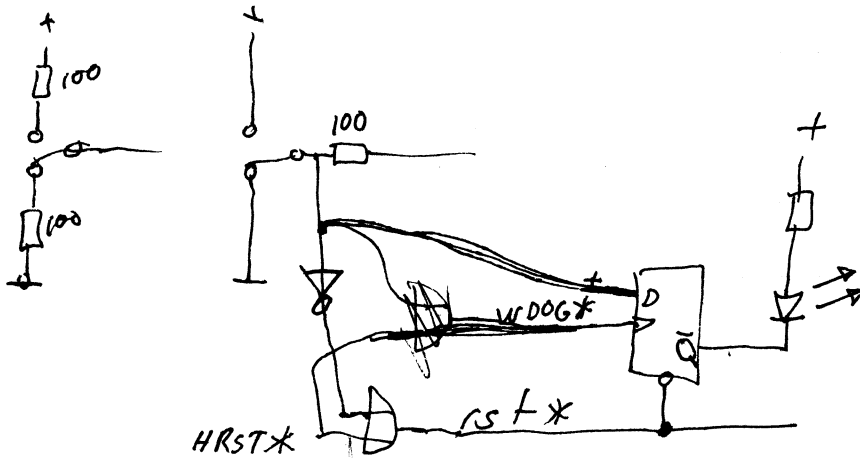
276 Quad $\bar{J}-\bar{K}$ - flip-flop

5-34

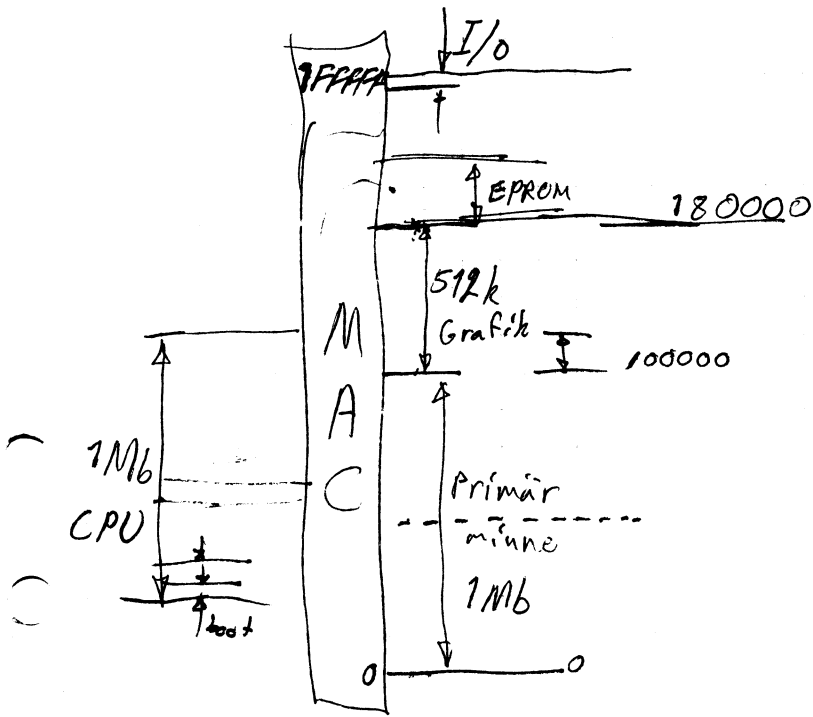
5-32

5-23

5-34



Dit x26 ger den højre
 av ryndem.



Ust 3x27256

mov w a0@, D6

EXOR D5, D6

OR D6, D3

↳ testa d3 om ≠ 0
 Pagefel ←
 maska av senaste
 ut

| | | |
|------------------------|------------------------------|------------------------------|
| Från/From | Datum/Date | Beteckning/Reference |
| Rubrik, ärende/Subject | Ert datum/Your date | Er beteckning/Your reference |
| | Gäller fr o m/Effective date | Ersätter/Replaces |

Distribution

1FFFFF

0 - 127

128 - 255

256 - 383

384 - 512

FFFFF

100000

Bildminne

Adress 19 hög

80000

MAC:en själv

00000

20000

60000

728 entry

80000

20000

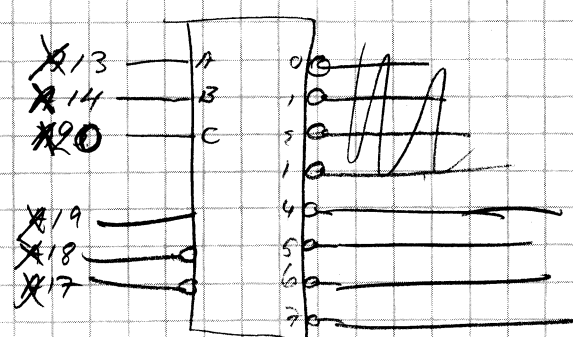
20000

90000

EPROMET

8 2 1
1 0 0 8

9



| | | |
|------------------------|------------------------------|------------------------------|
| Från/From | Datum/Date | Beteckning/Reference |
| Rubrik, ärende/Subject | Ert datum/Your date | Er beteckning/Your reference |
| Mac Page transport | Gäller fr o m/Effective date | Ersätter/Replaces |

Distribution

minnes
red

X 19

X 18

0

0

1

2564

0

1

2

2564

1

0

3

2564

1

1

4

2564

Walter opp program med

0-127

= 7/00

128-255

= 1/80

256 → 883

⇒ 1/00

384 - 511

| | | |
|-------------------------------------|------------------------------|------------------------------|
| Från/From | Datum/Date
8/12/19 | Beteckning/Reference |
| Rubrik, ärende/Subject
Primärium | Ert datum/Your date | Er beteckning/Your reference |
| | Gäller fr o m/Effective date | Ersätter/Replaces |

Distribution

För att komma åt primäriumet
måste A17 - A19 (uågon) vara hög
... låt vi ha "BOOTE"-biten satt
/40

DVS vi måste sätta upp masken
med A17, A18, A19 uågon hög av den

Pageran

segran

| | | | | |
|--------|-------|--------|----|------|
| 90 000 | 0 92A | A0003 | 0 | 2564 |
| 90 800 | 1 | A3003 | 1 | |
| 91 000 | 2 | B0003 | 2 | |
| 91 800 | 3 | B8003 | 3 | |
| 92 000 | 4 | C0003 | 4 | |
| 92 800 | 5 | C8003 | 5 | |
| 93 000 | 6 | D0003 | 6 | |
| 93 800 | 7 | D8003 | 7 | |
| 94 000 | 8 | E0003 | 8 | |
| 94 800 | 9 | E8003 | 9 | |
| 95 000 | 10 | F0003 | 10 | |
| 95 800 | | F8003 | 11 | |
| 96 000 | | 100003 | 12 | |
| 96 800 | | | | |
| 97 000 | | | | |
| 97 800 | | | | |
| 98 000 | | | | |
| 98 800 | | | | |
| 99 000 | | | | |
| 99 800 | | | | |
| 9A 000 | | | | |
| 9A 800 | | | | |

800000 → 00000000

000000

A19 o A20 ej inkopplade

1100000

100000

| | | |
|------------------------|------------------------------|------------------------------|
| Från/From | Datum/Date | Beteckning/Reference |
| Rubrik, ärende/Subject | Ert datum/Your date | Er beteckning/Your reference |
| | Gäller fr o m/Effective date | Ersätter/Replaces |

Mac uppsättning

uppsättning av mac för att testa
primär minnet

macen måste sättas upp för adresser

segment

00003

A19 = 1

A17 = 1

upp till pal



f8000

16C8 som
indikerar att

(100003 testas på)

⇒ 256 instruktioner ?

Rotprogrammet
inte skall vara
inladdat.

Pageammet innehåller var i primärminnet
vi ligger i D05 vi flyttar om
området

/20000 → 17ffff (*logisk)

→ adress i primärminnet.

indata d5 anger var vi skall
börja läsa i primärminnet

I segmentet skriver vi in 16 segment

nummer: (A15 - A18)

I pageammet skrivs sedan i antal ~~st~~ Page

256 st A11 - A18, Det ä. adresserna till

primärminnets ~~XXXX~~ X11 → X20

| | | |
|------------------------|------------------------------|------------------------------|
| Från/From | Datum/Date | Beteckning/Reference |
| Rubrik, ärende/Subject | Ert datum/Your date | Er beteckning/Your reference |
| Mecboppsättning | Gäller fr o m/Effective date | Ersätter/Replaces |
| | Distribution | |

Var i primärminnet Vi skall starta beskrivs av d3 som sedan kallas upp Page fram.

start adress 000000

anges med (wds = / 0000)

Physiska



07ffff

som är 256k minne

080000

(wds = / 0000)



1ffffff

256k totalt
1Mb

När vi adresserar ovan måste vi starta på adress

020000

(Befrågat adresserar inte det av låga adresser.)

| | | |
|---|------------------------------|------------------------------|
| Från/From
25L H | Datum/Date
841218 | Beteckning/Reference |
| Rubrik, ärende/Subject
Extravänne borker | Ert datum/Your date | Er beteckning/Your reference |
| | Gäller fr o m/Effective date | Ersätter/Replaces |

Distribution

7444

| D | C | B | A | Eprom | Hex | Utgång |
|-----|-----|-----|-----|-------|--------|--------|
| A20 | A19 | A18 | A17 | | | |
| 1 | 1 | 0 | 0 | 1 | 180000 | 5 |
| 1 | 1 | 0 | 1 | 2 | 1A0000 | 6 |
| 1 | 1 | 1 | 0 | 3 | 1C0000 | 8 |
| 1 | 1 | 1 | 1 | 4 | 1E0000 | 7 |

Epromen adress rynder.

| | | | |
|----|-------------------|---|-------------------|
| 1. | 180000 | → | 187fff |
| | 188000 | | 18ffff |
| 2. | 1A0000 | → | 187fff |
| | 190000 | | 197fff |
| 3. | 1E0000 | → | 1C7fff |
| | 188000 | | 19ffff |
| 4. | 1E0000 | → | 1E7fff |

OBS 32k minneskaplar

- Uppställningar : ① taskregistret nollställt
- ② 5 högsta bitarna i segmentram 1
0 bit 6 " "
- ③ ADDRESSBUSSEN A15-A18 styr segmentram
- ④ 6 högsta bitarna i pageram
- ⑤ ADDRESSBUSSEN A17-A14 styr pageram

| d0 | d1 | d2 | A0E | | segmentram
adress a0 | pageram
adress a0 |
|------|------|------|-----|---|-------------------------|----------------------|
| 00FF | 40FF | | 8 | 3 | | |
| | | 00FF | | | | |
| | 40FF | 000F | | | | |
| | 40FF | 000F | 8 | 3 | 000F | 40FF |
| | 40FF | | 8 | 1 | | |
| | 40FE | | 8 | 5 | | |
| 00FE | | 00FE | | | | |
| | | 000F | 8 | 3 | 000F | |
| | | | 8 | 1 | | |
| | | | 8 | 5 | | 40FE |
| | 40FD | | | | | |
| 00FD | | | | | | |
| | | 00FD | | | | |
| | | 000F | | | | |
| | | | 8 | 3 | 000F | |
| | | | 8 | 1 | | |
| | | | 8 | 5 | | 40FD |
| | 40F | | | | | |

(*Test program*)

Testprogrammet för "MACCEN" läggs in i ROOT startprogrammet.

Ut testerna i checkread page grupp rån
(segmentramens test) a i Checkread page rån
(pagesegment test)

- I segment rammen testas vilken bytet (4bit)
(hög låg) som felet ligger i och på så sätt pekar ut vilken krets av pagerammet som är fel
(176 högt byte, 15 lågt byte)

Alla pageramena testas X20-X11
(X20-X10 i 186, X18-X15 i 226, X14-X11 i 206)

om något fel uppstår fäns någon lysdiöd som indikerar att fel har uppstått.

| Lysdiöd | | krets | |
|---------|---|-------|--------------|
| 1 | 0 | 156 | } segmentram |
| 2 | 1 | 176 | |
| 4 | 2 | 186 | |
| 3 | 3 | 206 | } page rån |
| 1 | 4 | 226 | |
| 2 | 5 | | |
| 4 | 6 | | |
| 5 | 7 | | |

OM "MACCEN" korrekt.

När testerna är gjord skrivs ett ut på 7-segment displayen. och felet uttas på lysdiöden.

ORs om fel upptäcks i segment rammen avbryts testerna. annars gör page rån test

(*test av segment*)

D0 → D3 anger process nummer.
adressa fastregistret. \$80005-

segment A15 - A18, A8 + fastregistret
A19 ligger hög när (processnumret)
vi skriver i "maccen" annars
är det den som gäller istället för A8
segmentet på 16x4 (11 bitar: A₀-A₁₀)

vi har 5st adress bitar som skall gå igenom

80003 A₈
 0

↓
F8003

80103 A₈
 1

↓
F8103

i dessa minnes utrymme skall det skrivas
en vandrande "1"

① (adressbuss) och Data buss kallas

② minnescellerna kallas också

ordningar byte 80003 → 80103 →
88003, 88103, 90003, 90103
98003, 98103, A8003, A8103
....., F8003

avkodning till segment ram.

| \overline{CS} | A19 | A2 | A1 | A0 |
|-----------------|-----|----|----|----|
| 0 | 1 | 0 | 1 | 1 |

\overline{WE}

write 0 1 0 1 1 + R/W går låg

read 1 1 0 1 1 + R/W går hög

avkodning till pageram

| \overline{CS} | A19 | A2 | A1 | A0 |
|-----------------|-----|----|----|----|
| 0 | 1 | 0 | 0 | X |

\overline{WE}

write 206-226 0 1 0 0 0 + R/W signal låg

write 186 0 1 0 0 1 + R/W signal låg

read 206-226 1 1 0 0 0 + R/W signal hög

186 1 1 0 0 1 + R/W signal hög

Ben A1 skiljer mellan 1 : segment ram
0 : pageram

A19 selectar page e segment ram

841122

autodriving task register

signal WC ben 19 Pal 16C8 17E

clock 9 A19 A8 A1 A0

0 1 1 0 1

① i lastregistret lägges 01000600

Epiour enable

adress perlin räknes när a0 två steg
(word)

Då perlin vi i segment rammet med
adressperlin a0

du skrivs OF in

Den räkna när är adressa två
steg (word)

och skrivs in 40FF i pagerammet.

Sedan räknes adressen a0 när 7FC

DuS totalt 800

Det dras ett fän d1 (40FE)

d0 delumenteras med 1 och testas

om inte nåt görs en ny instruksion

tills vi har skrivit in 40FF → 4000

i 255 minnes utrymme

⚡ i segment rammet skrivs in FF-00
225 → 000

vid uppstartande görs "macer"
transparent mot databussen.

ADRESS BYMD "MACCEN"

Segment Ram 800003 (xbytex)

Page Ram 186 800001 write

(20-22)6 800000 write

(186 800001 read)
 (20-22)6 800000 read

taskregistret 19E 800005 (xbytex)

Segment adresseras A18 - A15 + taskregistret
 (+ A8 + A19)

Page adresseras A11 - A14 + seg 156

På adresserna A13 - A11 7F8 som räknas
 ut

$$\begin{array}{r}
 FF \\
 - 800 \quad A \\
 \hline
 7F800 \\
 80004 \\
 \hline
 FF804
 \end{array}$$

*test av Pagerant) kollställning

vi använder serjantens adress 8000?
som pagerans adress utöver.

Skriver 64 st olika adresser där i
när vi testar pagerarna.

har vi testa pagerarna

ändra vi All-År för varje i "skrift" i
page rummet

$$\begin{array}{r}
 80000 \\
 800 \\
 \hline
 80800 \\
 800 \\
 \hline
 81600
 \end{array}$$

↓
87800 sista adress 1

800
88000 testa per

↓ andra pagerarna

6st sista ä. 3f
testa på 800 40

Sjuseg: ~~1~~: segmentrammet fyllt
av ullor

om Lysdiöd: fel i test två

A: alla åtta tända har det skett
ett adressfel (hoppriering) till fel
minnesutrymme: Judikens med 8st
tända lysdiödar

B: om det är biffel i segmentrammet
pekas den troliga kretsen ut

Diöd 1: segmentram 1

Diöd 2: segmentram 2.

Det kan även vara fel på
kretsarna som ligger på databasen
in till segvramkretsarna.
om både av tända skall man nog
undersöka kretsarna

: databuss till segran +
bittest DATA test
sjuseg 2 Här skrivs det in ca
vandrande offset i alla adresser
(Process 0 - Process 15),
minimerellena testas var och en
samt att det inte är något fel
på databussen.

Testen utför genom att låta
en vandrande offset gå genom alla
minimerstrukturer.

Under testen indikeras det vilken data
buss som testas. Kan vara hjälp
vid eventuellt kortslutning i bussen.

Segmentramet görs även transparent här

~~ben lysdiöd fel i test 4~~

~~fill 3
seg~~

~~skall inte ske i fall
det inte är någon
buss error~~

fel i trim om bitfel i segran 156
skrivs det ut 00000001 på dörden
fel i segran 176 ger mönstret
00000010. felet kan även ligga i
transiverkretsarna 74245 som är buffrar mot
databussen.

Sjuseg 3. adressbuss till segran +
kopieringsstest (överstyrning)
ADDRESS TEST

Här testas adressbussen.

Det görs genom att skriva på
en adress i PROCESS NOLL
 sedan läses det på övriga adresser
i process noll att det inte har
blivit något skrivet där.

Det testas även på skrivna adresser
i övriga processer. Innehåller att

TASKREGISTRET fungerar

om lysdrödar: Det läggs ut |||||
om adressfel har inträffat.

om lysdrödar: fel i test 3

Det har blivit något biffel i
seggranen. Innehåller det på
första eller andra lysdröde
motsvara segran 3 eller segran 2?

Troliga fel är biffel i segran, men
det kan även vara fel på
74245 kretsarna som ligger på
databussen (transceiver)

se även test 3:

till
②

hw
-/33

sjuseg 5 Här har skrivboken i

programmet testas

Det testas med en användare efter

Under testen indikeras det i vilken
date buss som testas

Det testas för alla segutbuss 08st
plus pageadresser

om lysdioden lyser indikeras det
var felet ligger

| | | | | |
|---------------------------------------|---|---------------|-----|---|
| 000000 ¹ 0 ¹ 00 | : | fel i program | 206 | |
| 00000010 | - | fel i program | 226 | |
| 00000100 | - | fel i program | 286 | 0 |

transivern mellan Pageramerna slut

och i lottas 196, 216

samt 21F multiplexern 000

enable signaler
från

sjuseg 6.. har testat att adressbuss
till pageramen fungerar, testen

lägga in ett ord på en adress och
testar så att det inte har blivit
hoppierat över på någon annan plats
i pageramenet; Alla adresser från programmet
går igenom (ble st), Pageramenet
har 15 adresser.

~~Lagsdioder fel i test 6~~

~~Det skrivs ut 133 00110011
på dioderna när det har blivit
Bitfel i pageramenet~~

om det blir fel skrivs det
ute 01111111 på lagsdioderna

Program struktur

Diod

: laddar ut databuss på diod

Sjuseg

: räknar upp sjuseg

Diod seg

: räknar upp sjuseg + diodlatch

Delay

: fördröjningsrutin

Test

: testar dioder + sjuseg

seg res

: nollställa segram

seg DA fel

seg data

: testar bitmascelle

seg AD fel

seg add

: testar adressering a ~~test~~ adress

~~seg ram~~

: gör ~~seggram transparent~~

SEG RAM

: testar segram

~~Page ram~~

~~Page AD fel~~

Page res : gör segram transparent.

Page DA fel

Page AD fel

Page ADD

Page Data

PAGE RAM

: testar pageram

Seg tram

Kontakter mellan cpu-kort och video-kort (X35)

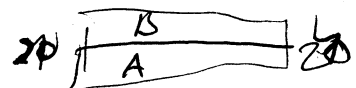
P10:

| | | A | | B | |
|-----------|-------|---|----|---|-------------|
| GND | | o | 1 | o | GND |
| NODAS* | <--- | o | 2 | o | ----> X15 |
| X16 | <--- | o | 3 | o | ----> X17 |
| X18 | <--- | o | 4 | o | ----> X14 |
| X13 | <--- | o | 5 | o | ----> X12 |
| X11 | <--- | o | 6 | o | ----> X19 |
| X20 | <--- | o | 7 | o | GND |
| VSYNC | ----> | o | 8 | o | GND |
| HSYNC | ----> | o | 9 | o | GND |
| SCREENPOS | ----> | o | 10 | o | GND |
| MINT | ----> | o | 11 | o | GND |
| IOWR2* | <--- | o | 12 | o | GND |
| IOWR1* | <--- | o | 13 | o | GND |
| IOWR0* | <--- | o | 14 | o | GND |
| 2*CLK | ----> | o | 15 | o | GND |
| IORD0* | <--- | o | 16 | o | GND |
| IORD* | <--- | o | 17 | o | GND |
| W/R* | <--- | o | 18 | o | GND |
| REFCLK* | ----> | o | 19 | o | ----> PRST* |
| BDS* | <--- | o | 20 | o | GND |
| | | A | | B | |

ändras till
enbart bit 0

P9:

| | | A | | B | |
|--------|-------|---|----|---|---------|
| GND | | o | 1 | o | GND |
| B8 | <--+ | o | 2 | o | ---> B9 |
| B10 | <--+ | o | 3 | o | ---> B1 |
| B3 | <--+ | o | 4 | o | ---> B5 |
| B6 | <--+ | o | 5 | o | ---> B7 |
| B4 | <--+ | o | 6 | o | ---> B2 |
| B0 | <--- | o | 7 | o | GND |
| DAS* | <--- | o | 8 | o | GND |
| D0 | <--> | o | 9 | o | GND |
| D2 | <--> | o | 10 | o | <--> D1 |
| D4 | <--> | o | 11 | o | <--> D3 |
| D6 | <--> | o | 12 | o | <--> D5 |
| +5V | | o | 13 | o | <--> D7 |
| PACK* | ----> | o | 14 | o | GND |
| CRT* | <--- | o | 15 | o | GND |
| RSTIN* | <--- | o | 16 | o | GND |
| GND | | o | 17 | o | GND |
| GND | | o | 18 | o | GND |
| GND | | o | 19 | o | GND |
| GND | | o | 20 | o | GND |
| | | A | | B | |



sett stiftsida
fel på
krets kortet

1983-11-01/6L

DMA.

DMA 0 for Floppy and Bus 0 address: 1F F300 H

DMA 1 for Bus 1 address: 1F F400 H

DMA 2 for Bus 2 address: 1F F500 H

DMA map registers:

| | | |
|--------|------|-------------|
| bit 15 | R/W* | DMA-A15 = 0 |
| bit 14 | N A | - " - |
| bit 13 | N A | - " - |
| bit 12 | X 20 | - " - |
| bit 11 | X 19 | - " - |
| bit 10 | X 18 | - " - |
| bit 9 | X 17 | - " - |
| bit 8 | X 16 | - " - |

| | | |
|-------|------|-------------|
| bit 7 | R/W* | DMA-A15 = 1 |
| bit 6 | N A | - " - |
| bit 5 | N A | - " - |
| bit 4 | X 20 | - " - |
| bit 3 | X 19 | - " - |
| bit 2 | X 18 | - " - |
| bit 1 | X 17 | - " - |
| bit 0 | X 16 | - " - |

DMA map register 0 address = 1F FD06 H

DMA map register 1 address = 1F FD04 H

DMA map register 2 address = 1F FD00 H

```

38 0
39 END;
40
41 /*
42 *   Dma table to use for reads from floppy.
43 *   Note that zero has a special meaning (insert parameters or end) !
44 *   so avoid zeroes !
45 */
46 unsigned char dmaifpr[] = BEGIN
47   0xc3,0xc3,0xc3,0xc3,0xc3,      /* Reset */
48   0x79,                          /* Transfer */
49   0,                              /* address and count gets inserted here */
50   0x54,0x0d,                    /* WR1 */
51   0x68,0x0d,                    /* WR2 */
52   0x80,                          /* WR3 */
53   0xcd,0x06,0x01,              /* WR4 */
54   0x92,                          /* WR5 */
55   0xcf,0x87,                    /* LOAD and GO */
56   0

```

FO
06
01
carby abdu.v lcs

FOOB
FOOB

```

57 END;
58
59 /*
60 *   Dma table to use for writes to floppy.
61 *   Note that zero has a special meaning (insert parameters or end) !
62 *   so avoid zeroes !
63 */
64 unsigned char dmaifpw[] = BEGIN
65   0xc3,0xc3,0xc3,0xc3,0xc3,      /* Reset */
66   0x79,                          /* Transfer */
67   0,                              /* address and count gets inserted here */
68   0x54,0x0d,                    /* WR1 */
69   0x68,0x0d,                    /* WR2 */
70   0x80,                          /* WR3 */
71   0xcd,0x06,0x01,              /* WR4 */
72   0x92,                          /* WR5 */
73   0xcf,0x05,                    /* LOAD and swap direction */
74   0xcf,0x87,                    /* LOAD and GO */
75   0

```

FLOPPY: 1F F000 Base address = Command/Status Register
1F F002 Track Register
1F F004 Sector Register
1F F006 Data Register
=====

These are the internal addresses of the ABC1600 function blocks.

=====
SPCR: 1F FE00 Address (Special Control Register)

DMAMAP2: 1F FD00 Address

DMAMAP1: 1F FD04 Address

DMAMAF0: 1F FD06 Address

FLCR: 1F FB00 Address (Floppy TTL Control Register)

CIO: 1F F700 Base address = Data Port C

1F F702 Data Port B

1F F704 Data Port A

1F F706 Control Port

SCC: 1F F600 Base address = Control Port B

1F F602 Data Port B

1F F604 Control Port A

1F F606 Data Port A

DMA2: 1F F500 Address (Bus 2)

DMA1: 1F F400 Address (Bus 1)

DMA0: 1F F300 Address (Floppy & Bus 0)

DART: 1F F200 Base address = Control Port B

1F F202 Data Port B

1F F204 Control Port A

1F F206 Data Port A

CRT: 1F F100 Base address

1983-11-01/8L

Z_8536_CIO_connections.

PA7 BUS 0i interrupt
PA6 BUS 0x interrupt
PA5 BUS 0 XINT*5
PA4 BUS 0 XINT*4
PA3 BUS 0 XINT*3
PA2 BUS 0 XINT*2
PA1 BUS 1 interrupt
PA0 BUS 2 interrupt

PB7 FINT (Floppy interrupt)
PB6 PREN* (BUS 0)
PB5 PREN* (BUS 1)
PB4 MINT (Mover interrupt)
PB3 Reserved
PB2 Reserved
PB1 Power fail/ok*
PB0 PRinter Baud Rate out

PC3 Parameter NVRAM CS (NSC NMC9306)
PC2 Calender Clock CS* (MEM E050-16)
PC1 Bidirectional data (program open drain)
PC0 Clock

Z 8536 CIO Register Addresses:

Data Port C 1F F700 H
Data Port B 1F F702 H
Data Port A 1F F704 H
Control Port 1F F706 H

1983-11-01/6L

FLOPPY_ITL_CONTROL_REGISTER. Address = 1F F800 H

15 N A
 14 N A
 13 Low Current
 12 Post Compensation
 11 Motor On
 10 SEL 3
 9 SEL 2
 8 SEL 1

 7 Precompensation select P2(9229)
 6 Precompensation select P1(9229)
 5 Precompensation select P0(9229)
 4 Head Load input (9229) program 1 always
 3 MINI input (9229) 0 = 8" 1 = 5"
 2 Head Load Timing input (1797)
 1 Density 0 = double 1 = single
 0 Master Reset input (1797) 0 = reset

FLOPPY_CONTROLLER (1797) REGISTER ADDRESSES.

Data Register 1F F006 H
 Sector Register 1F F004 H
 Track Register 1F F002 H
 Command/status reg. 1F F000 H

Z80A-DMA_setup_when_floppy.

| | |
|-------------------------|--|
| WR0 = 79H,1a,ha,1c,1c | Transfer mode. |
| WR1 = 54H,0DH | A address + count. |
| WR2 = 68H,0DH | A is memory and variable cycle.
Late end MRQ, early end RD, WR. |
| WR3 = 80H | B is I/O and variable cycle.
Late end IDRO, early end RD, WR. |
| WR4 = CDH,06H,10H | No stop on match
Burst mode. |
| WR5 = 92H | B address = 1006H. |
| START = CFH,87H | RDY low, CE/WAIT multiplexed. |
| START = CFH,05H,CFH,87H | if floppy to memory.
if memory to floppy. |

1983-11-01/6L

DMA.

DMA 0 for Floppy and Bus 0 address: 1F F300 H
DMA 1 for Bus 1 address: 1F F400 H
DMA 2 for Bus 2 address: 1F F500 H

DMA map registers:

bit 15 R/W* DMA-A15 = 0
bit 14 N A - " -
bit 13 N A - " -
bit 12 X 20 - " -
bit 11 X 19 - " -
bit 10 X 18 - " -
bit 9 X 17 - " -
bit 8 X 16 - " -

bit 7 R/W* DMA-A15 = 1
bit 6 N A - " -
bit 5 N A - " -
bit 4 X 20 - " -
bit 3 X 19 - " -
bit 2 X 18 - " -
bit 1 X 17 - " -
bit 0 X 16 - " -

DMA map register 0 address = 1F FD06 H
DMA map register 1 address = 1F FD04 H
DMA map register 2 address = 1F FD00 H

1983-02-02/6L

SPECIAL CONTROL REGISTER. Address = 1F FE00 H

| Byte data | Result |
|-----------|---|
| 00 H | (r) CS7 = 0 |
| 01 H | (r) Code = 0 |
| 02 H | (r) Driving Transmit Clock to SCC channel B |
| 03 H | (r) Driving Transmit Clock to SCC channel A |
| 04 H | (r) Generate and test Parity |
| 05 H | (r) Disable DMA:s |
| 06 H | (r) Disconnect DMA 1 from SCC |
| 07 H | (r) Disconnect DMA 0 from floppy |
| 08 H | CS7 = 1 |
| 09 H | Code = 1 |
| 0A H | SCC channel B may drive Transmit Clock |
| 0B H | SCC channel A may drive Transmit Clock |
| 0C H | Generate erroneous Parity and do not test |
| 0D H | Enable DMA:s |
| 0E H | Connect DMA 1 to SCC |
| 0F H | Connect DMA 0 to floppy |

(r) = after reset state

1983-12-12/6L

INTERFACE_MOT_X36_GRAPH.

X20 - X11
B10 - B0
D7 - D0
CRT* (= 6845 CS*)
IORQ*
W/R*
MINT (Mover interrupt)
BDS* (Bufferd DS*)
DAS* (Delayed AS*)
NODAS*
PACK*
PRST* (Reset)
2*CLK (16 MHz)
REFCLK*
IOWR2 - IOWR0
IORD0

HSYNC
VSYNC
HL/P* (Monitor position)
RST/RUN*

Kontakt mellan cpu-kort och busskort (X35)

| | A | B | |
|-------------------|------|---------|--------------|
| Vcc | o 1 | o | Vcc |
| +12V | o 2 | o | +12V |
| -12V | o 3 | o | -12V |
| GND | o 4 | o | GND |
| I2 <---> | o 5 | o <---> | I1 |
| I0 <---> | o 6 | o <---> | I3 |
| I4 <---> | o 7 | o <---> | I5 |
| I6 <---> | o 8 | o <---> | I7 |
| D0 <---> | o 9 | o <---> | D1 |
| D2 <---> | o 10 | o <---> | D3 |
| D4 <---> | o 11 | o <---> | D5 |
| D6 <---> | o 12 | o <---> | D7 |
| X11 <---> | o 13 | o <---> | PREN* bus 2 |
| E0* <---> | o 14 | o <---> | E12* |
| CSP* <---> | o 15 | o <---> | CSA* |
| BUS0 <---> | o 16 | o <---> | DIR |
| BPCLK <---> | o 17 | o <---> | BPCLK* |
| PREN* bus 1 <---> | o 18 | o <---> | PREN* bus 0 |
| GND | o 19 | o | GND |
| A6 <---> | o 20 | o <---> | A7 |
| A5 <---> | o 21 | o <---> | A4 |
| A3 <---> | o 22 | o <---> | A2 |
| A1 <---> | o 23 | o <---> | A0 |
| A9 <---> | o 24 | o <---> | A10 |
| BRST* <---> | o 25 | o <---> | A8 |
| IORQ* <---> | o 26 | o | |
| | o 27 | o <---> | NMI* BUS 0x |
| TRRQ* bus 2 <---> | o 28 | o <---> | CS7 |
| TRRQ* bus 1 <---> | o 29 | o | |
| TRRQ* bus 0 <---> | o 30 | o <---> | EXP* bus 0 X |
| PRAC* bus 2 <---> | o 31 | o <---> | RSCB* |
| PRAC* bus 0 <---> | o 32 | o <---> | FRAC* bus 1 |
| TREN* bus 2 <---> | o 33 | o <---> | TREN* bus 0 |
| TREN* bus 1 <---> | o 34 | o <---> | RD* |
| GND | o 35 | o | GND |
| | A | B | |

These are the internal addresses of the ABC1600 function blocks.

=====
SPCR: 1F FE00 Address (Special Control Register)

DMAMAP2: 1F FD00 Address

DMAMAP1: 1F FD04 Address

DMAMAP0: 1F FD06 Address

FLCR: 1F FB00 Address (Floppy TTL Control Register)

CIO: 1F F700 Base address = Data Port C
1F F702 Data Port B
1F F704 Data Port A
1F F706 Control Port

SCC: 1F F600 Base address = Control Port B
1F F602 Data Port B
1F F604 Control Port A
1F F606 Data Port A

DMA2: 1F F500 Address (Bus 2)

DMA1: 1F F400 Address (Bus 1)

DMA0: 1F F300 Address (Floppy & Bus 0)

DART: 1F F200 Base address = Control Port B
1F F202 Data Port B
1F F204 Control Port A
1F F206 Data Port A

CRT: 1F F100 Base address

FLOPPY: 1F F000 Base address = Command/Status R
1F F002 Track Register
1F F004 Sector Register
1F F006 Data Register
=====