

INHALTSVERZEICHNIS:

Kapitel	Seite
1. Allgemeine Beschreibung	2
2. Blockschaltbild CBM 8032	3
3. Aufbau des Adreßbereichs	4
4. Zero-Page und Stack-Register	5
5. IEEE-488 Bus	6
6. User Port	9
7. Tastaturabfrage	10
8. Bildschirmsteuerung	11
9. RAM-Speicher Organisation	12
10. Power On Reset Routine	15
11. Testhilfsmittel	16
12. Video-Board	17
13. Logic-Beschreibung	27
14. ASCII- und Bildschirmcode	32
15. Diagnostic-Box-Beschreibung	33
16. Hinweise zur Fehlersuche	35
17. No Operation-Tester	37
18. Technische Daten des CBM 8032	39

1. ALLGEMEINE BESCHREIBUNG

Die Zentraleinheiten der Serie 40xx/80xx sind Mikrocomputer, basierend auf dem Mikroprozessor 6502 der Firma MOS-Technologie, einer Tochter von Commodore.

Die wichtigsten Komponenten des Systems sind Bildschirm, Tastatur und Logic-Board mit verschiedenen Schnittstellen zum Anschluß von Peripheriegeräten.

Die beiden ersten Ziffern der Typenbezeichnung geben die maximale Anzahl der Zeichen/Bildschirmzeile an (z.B. 4032 = 40 Zeichen/Zeile)

Die beiden letzten Ziffern geben den Ausbau des RAM-Speichers an (z.B. CBM 4032 = 32 kByte Ram).

Als Tastaturen für die 40er Serie stehen zu Verfügung:

- a) Graphic-Tastatur mit Großbuchstaben und Graphiczeichen
- b) Business-Tastatur nach ASCII-Norm

Als Tastaturen für die 80er Serie stehen zu Verfügung:

- a) Business-Tastatur nach ASCII-Norm
- b) " " " " DIN-Norm

Alle Maschinen besitzen das gleiche Logic-Board; die 8096 hat zusätzlich ein 64kB Ram-Board eingebaut.

Das Betriebssystem ist in 18kB ROM gespeichert und heißt "Basic 4.0".

Um die 96kB RAM der 8096 zu Verwalten steht ein ladbares Betriebssystem zu Verfügung (LOS-96; Loadable Operating System).

An Schnittstellen zum Anschluß von Peripheriegeräten besitzen die Maschinen:

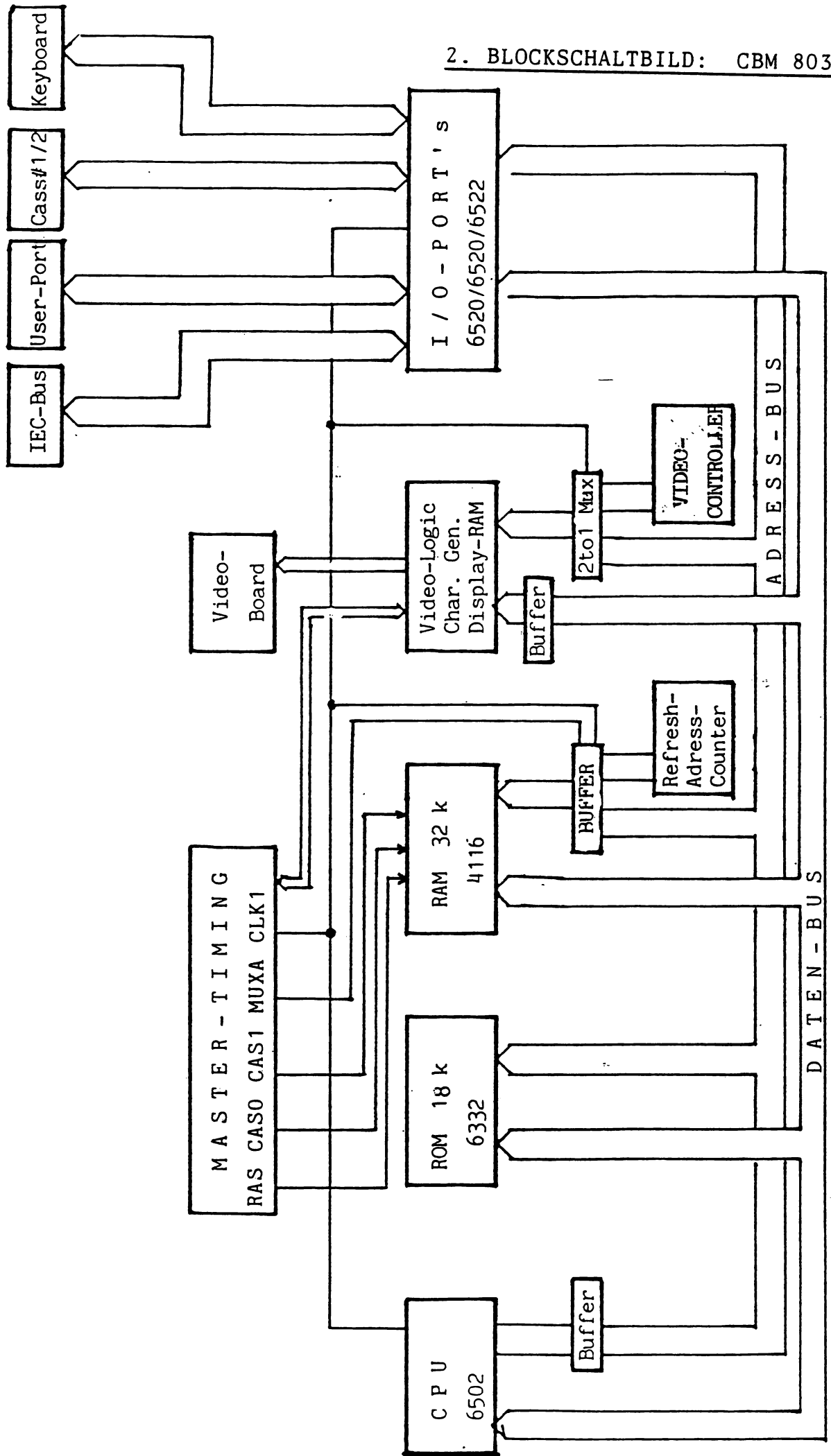
IEEE 488 Bus 8-Bit Parallelschnittstelle;
zum Anschluß der gesamten CBM-Peripherie;
Meßdatenerfassung

User Port frei programmierbare 8-Bit Parallel-
Schnittstelle; serielle Schnittstelle;
es stehen Video-Signale zum Anschluß
eines zweiten Monitors zu Verfügung.

2 Cassetten Ports: zwei serielle Schnittstellen zum Anschluß
von Datasetten

Memory
Expansion Port Steckerleiste, auf die der Systembus
(Adress-, Daten-, Steuerbus) herausgeführt
ist.

2. BLOCKSCHALTBILD: CBM 8032



3. AUFBAU DES ADRESSBEREICHES BEI CBM-COMPUTERN

Der Prozessor 6502 kann maximal 65535 Adressen (64kB) erzeugen. Von diesem Adressbereich werden max. 32kB für Ram-Arbeitsspeicher, 2kB für Bildschirm-Ram, 8kB für Speichererweiterung und 18kB für Basic-Interpreter, I/O-Steuerung und Betriebssystem benutzt.

Adressen (dezimal)	Inhalt	Adressen (hexadez)
65535	I-----I Operating System 4k ROM	FFFF
61440	I-----I I/O-Bausteine 2k PIA/VIA	F000
59392	I-----I I/O-Routinen 2k ROM	E800
57344	I-----I Basic-Interpreter 12k ROM	E000
45056	I-----I frei für Erweiterung 8k	B000
36864	I-----I	9000
34816	I-----I Bildschirm-RAM	8800
32768	I-----I	8000
	max. 32k	
	Basic-Programm- Speicherbereich	RAM
1024	I-----I Cassettenpuffer	0400
512	I Stack-Register	RAM --I 0200
256	I Zero-Page	--I 00FF
0	I-----I	0000

4. ZERO-PAGE UND STACK-REGISTER

Unter Zero-Page oder Page 0 (Page=Seite) versteht man den Adressbereich von 0 - 255.

Der Adressbereich von 0 - 65535 kann in 256 Seiten mit je 256 Adressen unterteilt werden.

Zur Adressierung einer dieser Seite werden zwei Bytes benötigt. Die einzige Ausnahme bildet die Zero-Page, bei der zur Adressierung nur ein Byte benötigt wird. Somit wird der Zugriff nahezu doppelt so schnell wie auf allen anderen Adressen.

Aus diesem Grund wird die Zero-Page vom Prozessor als Arbeitsspeicher benutzt, wo er Daten abspeichert, auf die er häufig zugreifen muß.

Das Stack-Register ist ein Speicherbereich in dem der Prozessor Rücksprungadressen ablegt wenn er in ein Unterprogramm springt oder ein Interrupt auftritt.

Der Stack ist als "last in first out"-Register organisiert, ist maximal 256 Bytes lang und liegt ab Adresse 512 abwärts.

Wichtig: Ist ein zur Zero-Page oder zum Stack gehöriges RAM defekt, wird der Computer beim Einschalten seine "Power On Reset"-Routine nicht ordnungsgemäß durchführen, was zur Folge hat, daß der Computer nicht "Ready" anzeigt sondern meistens der Bildschirm dunkel bleibt.

5. DER IEEE-488 BUS

Zum Anschluß von Peripheriegeräten (Drucker, Floppy, Meßgeräte) besitzen CBM-Computer einen IEEE-488 Bus. Ein Standard-Steckverbinder nach IEEE-488 Norm ist am Computer nicht vorhanden. Stattdessen wird ein Direktstecker mit 12 Positionen und 24 Kontakten verwendet. Kontaktabstand 3.96 mm. Codierschlitze befinden sich zwischen den Kontakten 2-3 und 9-10. Fan-out und Impedanzanpassung entsprechen IEEE-488.

Der IEEE-488 Bus ist, abgesehen von der Anschlußbelegung vergleichbar mit dem IEC-Bus.

IEEE : Institution of Electronics and Electrotechnical Engineers

IEC : International Electronic Commision

5.1 Beschreibung

Der IEEE-488 Bus ist ein bidirektionaler 8-Bit paralleler Datenbus. Zur Steuerung der Datenübertragung werden 5 Management-Signale und 3 Handshake-Signale benutzt.

Man unterscheidet drei verschiedene Geräte am Bus: Controller, Talker und Listener. In einem IEEE-Bus System darf es nur einen Controller geben (bei CBM die Zentraleinheit). Geräte, welche Daten senden heißen Talker, diejenigen, die Daten empfangen heißen Listener. Ein Gerät kann sowohl als Listener, als auch als Talker arbeiten.

5.3 Logische Zuordnung der Spannungspegel:

log 0 : $U > + 2.4 \text{ V}$ (high)

log 1 : $U < + 0.4 \text{ V}$ (low)

5.4 Geräteadressen:

Es können maximal 12 Geräte mit den Adressen 4 - 15 angeschlossen werden.

Die Adressen 0 - 3 sind vom Betriebssystem an folgende Geräte vergeben:

- 0 = Tastatur
- 1 = Cassette #1
- 2 = Cassette #2
- 3 = Bildschirm

Bei Commodore Druckern ist ab Werk die Adresse 4 eingestellt, bei Floppy Disk's die Adresse 8.

Den Geräten können aber auch beliebige andere Adressen zugeordnet werden.

5.4 Beschreibung der Bus-Signale

<u>Gruppe</u>	<u>Signal</u>	<u>Name</u>	<u>Funktionsbeschreibung</u>
Managment	ATN	Attention	Der Controller setzt dieses Signal auf low während er Befehle auf den Datenbus legt. Ist ATN high werden Daten übertragen.
"	EOI	End or Identify	Sobald das letzte Datenbyte übertragen wird, kann der Talker EOI auf low setzen
"	IFC	Interface clear	Wird beim Einschalten vom Controller kurzzeitig low und dient zum initialisieren der Peripheriegeräte
"	SRQ	Service Request	Bedienungsaufruf, wird vom Betriebssystem nicht bedient, kann aber per Programm abgefragt werden.
"	REN	Remote enable	wird bei CBM nicht benutzt und ist auf Masse gelegt
Handshake	DAV	Data valid	Wird vom Talker auf low gesetzt, wenn die Daten auf gültig sind.
"	NDAC	No Data accept	Wird vom Listener während dem Einlesen eines Bytes auf low gehalten. NRFD=high bedeutet das ein Datenbyte übernommen wurde.
"	NRFD	Not Ready for Data	NRFD=low sagt dem Talker, daß einer oder mehrere Listener nicht empfangsbereit sind.
Daten	DIO	DIO1-DIO8	8-Bit Datenbus

5.5 Der IEEE-Bus Status

Nach jeder IEEE-Bus Operation (input#;print#;get#) wird der aktuelle Status in die Variable ST übertragen.

Die Bedeutung der einzelnen Bits des Status, sowie ihr dezimaler Wert sind in folgender Tabelle aufgeführt:

<u>Bit</u>	<u>Dezimal</u>	<u>IEEE-Bus Status</u>	<u>Beschreibung</u>
0	1	Timeout beim Schreiben	wird gesetzt, wenn der Listener spätestens 64 ms nachdem der Talker DAV=low gesetzt hat, nicht mit NDAC=high geantwortet hat
1	2	Timeout beim Lesen	wird gesetzt, wenn der Talker spätestens 64 ms nachdem der Listener NRFD=high gesetzt hat, nicht mit DAV=low geantwortet hat
2	4	ohne Bedeutung	
3	8	"	
4	16	"	
5	32	"	
6	64	Dateiende (EOI)	wird gesetzt durch EOI=low
7	-128	DEVICE NOT PRESENT	wird gesetzt wenn NRFD <u>und</u> NDAC = high sind.

5.6 Der 64 ms Timeout

Um zu verhindern, daß der Rechner sich an einem langsamen Peripheriegerät "aufhängt", bricht das Betriebssystem nach 64 ms die Datenübertragung ab. Dieser "Timeout" ist abschaltbar:

POKE 1020,128 Timeout aus

POKE 1020,128 Timeout ein

Ist der Timeout ausgeschaltet, wartet der Computer solange bis das angesprochene Peripheriegerät geantwortet hat. Eine Unterbrechung dieser Warteschleife ist nicht über Programm, sondern nur durch Betätigung der Run/Stop-Taste möglich.

6. USER PORT

Der User Port ist ein frei programmierbare Schnittstelle, bestehend aus acht I/O-Leitungen, wobei jede wahlweise als Ein- oder Ausgang geschaltet werden kann.

Weiterhin stehen zwei Steuerleitungen zu Verfügung, die z.B. als Handshake-Leitungen benutzt werden können.

Mit Hilfe eines Schieberegisters ist auch serielle Datenübertragung möglich.

Zum Ansteuern eines zweiten Monitors sind die Video-Signale zum Stecker herausgeführt.

Der User Port wird durch den Baustein MCS 6522 (Versatile Interface Adapter; kurz VIA) realisiert. Der VIA liegt direkt am Systembus sodaß die Register des User Ports als normale Speicherstellen angesprochen werden. Dadurch können hohe Datenübertragungsraten erzeugt werden.

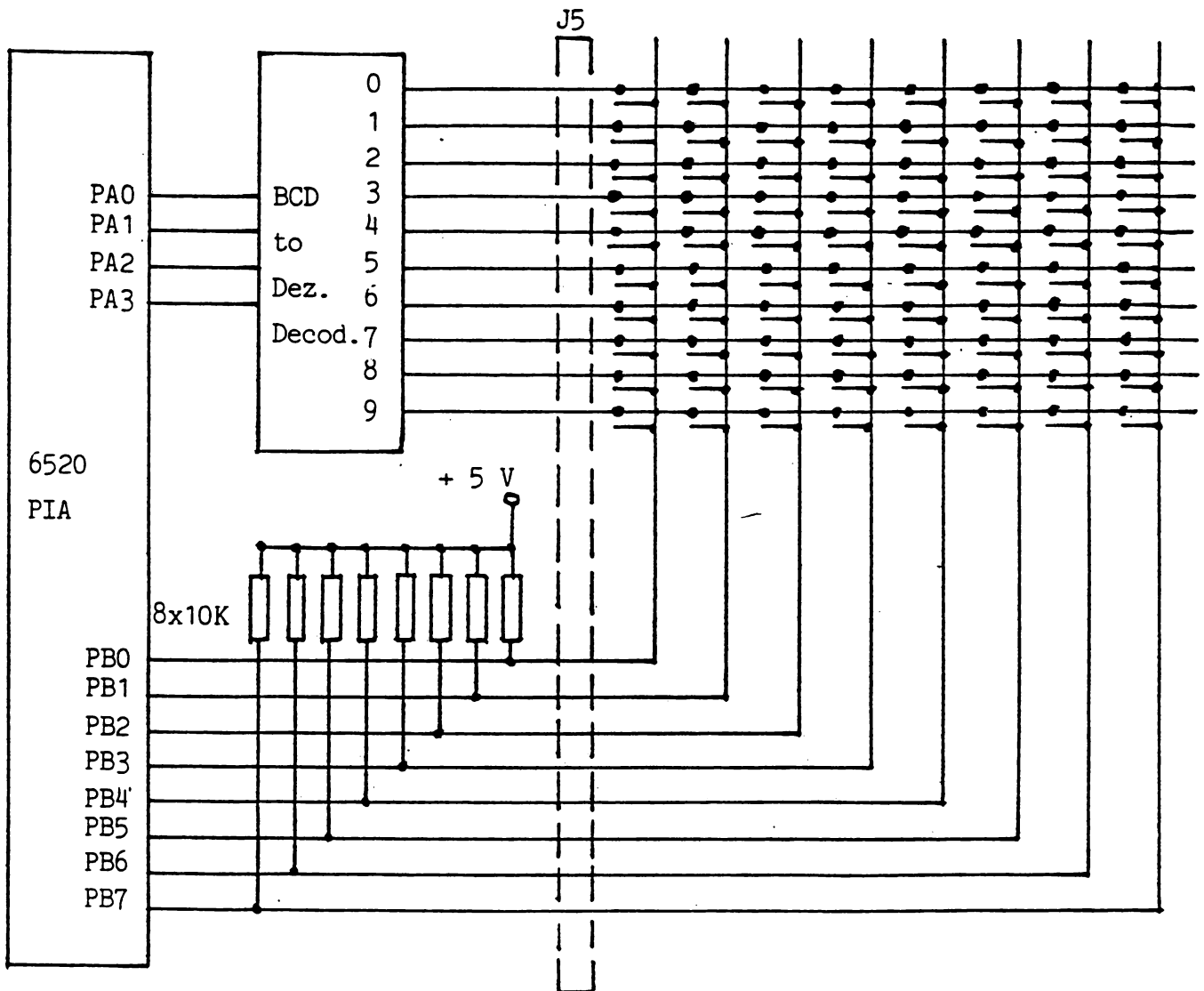
Adressen des 6522 im CBM :

Dezimal	Register/Funktion
59456	Output Register für I/O-Port B (ORB)
57	" " " " " A (ORA)mit Handshake
58	Datenrichtungsregister " B (DDRB)
59	" " " A (DDRA)
60	Lies LSB des Timers 1 und schreibe es als LSB ins Latch des Timers 1
61	Lies MSB des Timers 1, schreibe es als MSB ins Latch und beginne zu zählen
62	Hole LSB vom Latch des Timers_1
63	Hole MSB vom Latch des Timers 1
64	Lies LSB des Timers 2 , setze IFR Schreibe es als LSB ins Latch des Timers 2 ohne Reset
65	Lies MSB des Timers 2 Reset IFR beim Schreiben ins MSB des Latch
66	Seriellles I/O-Schieberegister
67	Hilfskontrollregister (ACR)
68	Peripheriekontrollregister (PCR)
69	Interrupt-Flag-Register (IFR)
70	Interrupt-Enable-Register (IER)
71	Output Register für I/O-Port A ohne Handshake

7. TASTATURABFRAGE

Die Tastatur wird vom Prozessor 50 mal pro Sekunde abgefragt (50HZ IRQ). Der Prozessor zählt die Ausgänge PA0 - PA3 binär von 0000 nach 1001. Je nachdem welcher binäre Wert am Eingang des Decoders anliegt, wird der entsprechende digitale Ausgang (0-9) auf low gelegt. Wird nun eine Taste gedrückt, wird dieser low-Pegel auf einen der Eingänge PB0-PB7 gelegt und der Prozessor ermittelt welche Taste gedrückt ist.

Wichtig: Um im Fehlerfall den Decoder zu überprüfen, ist es notwendig den Tastaturstecker abzuziehen und den Diagnostic-Stecker aufzustecken.



8. BILDSCHIRMANSTEUERUNG (40xx)

Der Bildschirm besteht aus 25 Zeilen à 40 Zeichen, d.h. es ist ein Speicher von 25 x 40 = 1000 Stellen notwendig (Display RAM) um den ganzen Bildschirm zu beschreiben. Der Adressbereich des Bildschirms liegt zwischen 32768 und 33767.

Beispiel:

POKE 32768,48 es wird eine 0 in die obere linke Ecke geschrieben

POKE 33767,48 es wird eine 0 in die untere rechte Ecke geschrieben

POKE 32769,48 es wird eine 0 neben die erste geschrieben

Die Zeichen werden auf dem Bildschirm in einer Matrix von 8 Spalten und 8 Zeilen dargestellt:

	1	-----*--*--*--*--*-----	
	2	-----*-----*-----	
	3	-----*-----*-----	
	4	-----*--*--*--*--*-----	Darstellung des Zeichens "P"
ZEILEN	5	-----*-----*-----	auf dem Bildschirm
	6	-----*-----*-----	
	7	-----*-----*-----	
	8	-----*-----*-----	
		I I I I I I I I	
		1 2 3 4 5 6 7 8	
		SPALTEN	

Der im Display RAM enthaltene Daten-Code eines Zeichens bildet die erste Adresse ab der die Matrixdaten im Charakter Generator abgespeichert sind.

Genauer: Unter dieser Adresse liegen 8-Bit Daten für die oberste Matrixzeile; unter der nächsthöheren Adresse die Daten für die nächste Matrixzeile usw.

9. RAM-SPEICHER ORGANISATION

9.1 RAM Typ

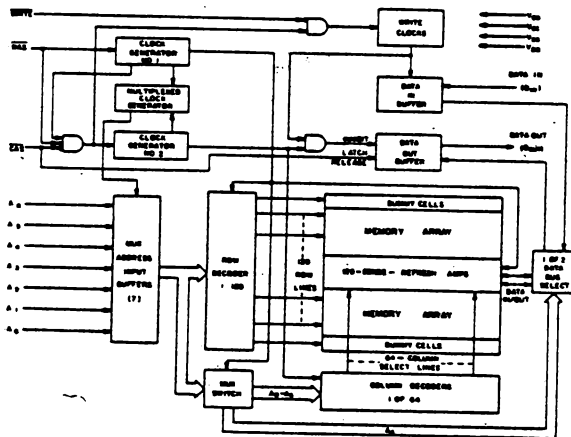
In allen neueren Geräten werden als RAM-Arbeitsspeicher dynamische RAM's des Typ's 4116 verwendet.

Technische Daten: Typ 4116-4

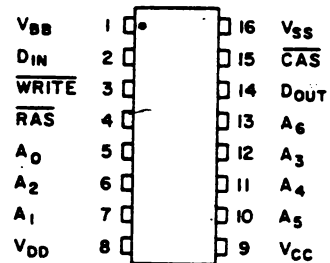
Access Time tRAC = 250 ns
 Cycle Time tRC = 410 ns
 Verlustleistung activ= 462 mW
 standby=20 mW

Gehäuse 16 Pin nach Industrie Standard
 Toleranz der Versorgungsspannung +/- 10 %
 128 Refresh-Cycles(2ms Refresh-Interval)
 Multiplexed Address Inputs
 TTL-kompatible Eingänge

FUNCTIONAL DIAGRAM



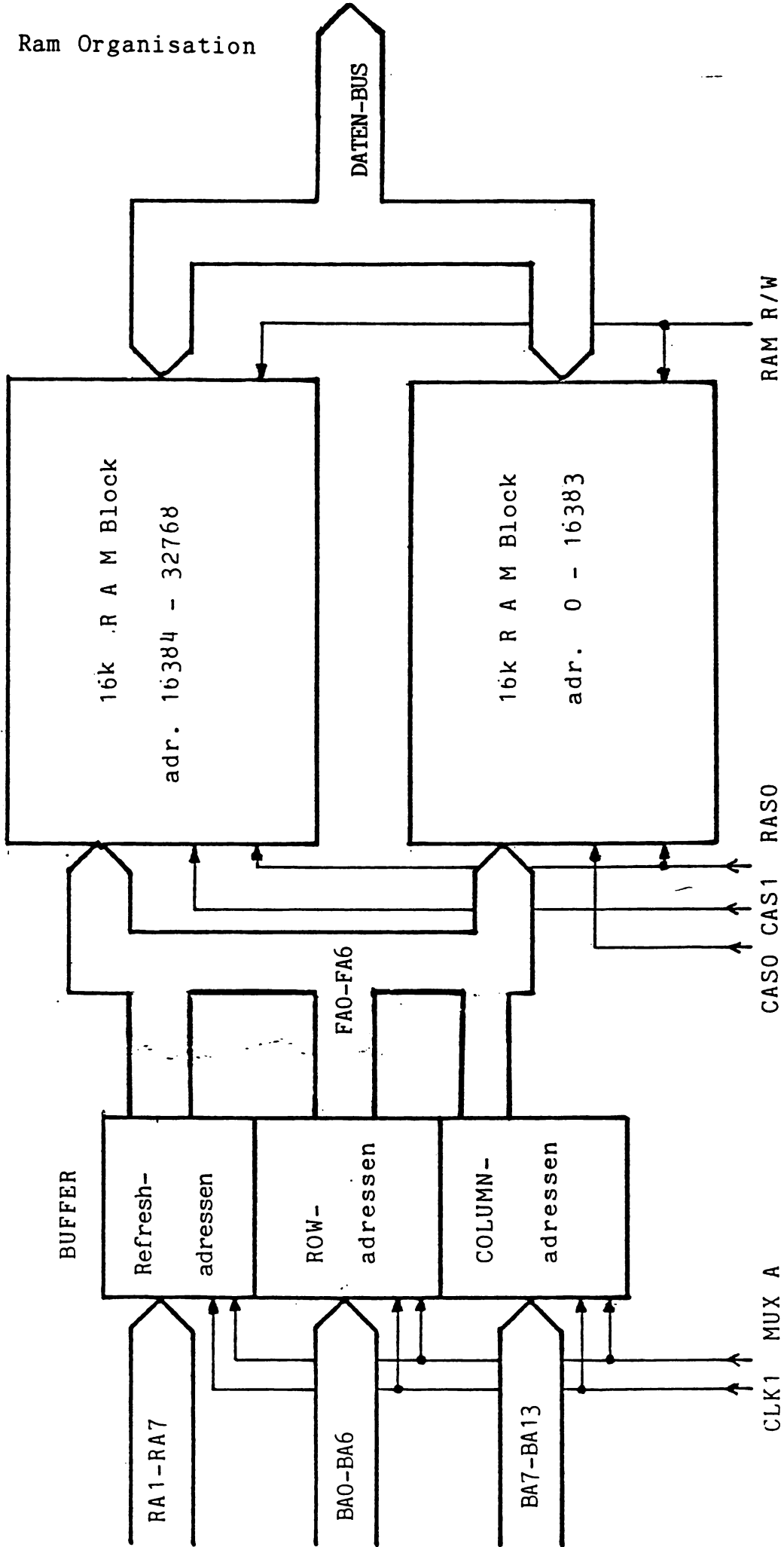
PIN CONNECTIONS



PIN FUNCTIONS

A ₀ -A ₆	Address Inputs	WRITE	Read/Write Input
CAS	Column Address Strobe	V _{BB}	Power (-5V)
DIN	Data In	V _{CC}	Power (+5V)
DOUT	Data Out	V _{DD}	Power (+12V)
RAS	Row Address Strobe	V _{SS}	Ground

9.2 Ram Organisation



9.3 Beschreibung des RAM-Arbeitsspeichers

Bei den 32k-Maschinen besteht der Arbeitsspeicher aus 16 RAM's des Typ's 4116, bei 16k-Maschinen aus 8 RAM's.

Da ein Datenbyte aus 8 Bits besteht und jedes RAM ein Datenbit besitzt, werden jeweils 8 RAM's zu einem Block zusammengefaßt. Um ein Byte auszulesen werden also genau 8 RAM's benötigt.

Die 8 Ram's in den Positionen UA5, UA7, UA9 usw. werden von den Adressen 0-16383 angesprochen, die RAM's in den Positionen UA4, UA6, UA8 usw. von den Adressen 16384 - 32768.

Mit den Signalen CAS 0 und RAS 0 wird der untere Adressbereich angesprochen und mit CAS 1 und RAS 0 der obere Bereich.

Der Adressbereich von 0-1024 wird vom Betriebssystem benutzt und steht dem Benutzer als Basic-Programmspeicher nicht zur Verfügung.

Die Ram's haben "Multiplexed Adress Inputs", d.h. die Adresseingänge A0-A6 werden zuerst zur Reihen Adressierung und dann zur Spaltenadressierung benutzt.

Die drei Buffer (74LS244) haben die Aufgabe, wahlweise die Refreshadressen oder die vom Prozessor erzeugten Adressen durchzuschalten. Die Auswahl wird durch die Signale MUX A und CLK 1 gesteuert:

MUX A	CLK1	Durchschalten von:
X	L	Refreshadressen
L	H	BA0 - BA6
H	H	BA7 - BA13

L = Low-Pegel

H = High-Pegel

X = Pegel ist ohne Bedeutung

10. POWER ON RESET ROUTINE

Diese Routine hat die Aufgabe das System nach dem Einschalten in eine definierte Ausgangsstellung zu bringen. Es werden verschiedene Werte in die Zero Page geschrieben, der Bildschirm wird gelöscht und der verfügbare RAM-Speicher wird berechnet.

10.1 Beschreibung

Nach dem Einschalten wird an Pin 40 für ca 100 ms ein low-Pegel angelegt (Reset-Signal). Dieses Signal setzt auch verschiedene Flip Flop's, Zähler und Register zurück.

Durch den Reset-Impuls wird der Prozessor veranlaßt sich von den Adressen 65532 und 65533 (F-ROM) einen Zeiger zu holen, der auf den Anfang der Power on Reset Routine zeigt (64790; F-ROM) und fängt nun an das Programm, welches ab dieser Adresse abgelegt ist abzuarbeiten:

Es werden die I/O-Bausteine initialisiert, die Glocke wird angeschlagen, der Bildschirmcontroller wird initialisiert und der Bildschirm wird gelöscht (E-ROM).

Nun wird die Diagnostic-Leitung (Pin 5 am User Port) abgefragt. Ist sie auf Masse gezogen springt der Computer in den Monitor (normalerweise nicht der Fall). Ist Diag=high kopiert er die CHARGET-Routine vom ROM in die Zero Page und berechnet den freien Speicher (D-ROM). Nun schreibt er die Meldung :

*** Commodore Basic 4.0 ***

xxxxx Bytes free

auf den Bildschirm (C-;B-ROM).

Zuletzt wird der Befehl NEW ausgeführt und "READY" auf den Bildschirm geschrieben.

Nun befindet sich der Computer in einer Warteschleife, in der er auf die Eingabe eines Zeichens wartet.

11. TESTHILFSMITTEL

Um die Funktionsfähigkeit der Computer zu testen und gegebenenfalls Fehler zu finden stehen folgende Hilfsmittel zu Verfügung:

Diagnostic Box

Testprogramm "RAM/ROM-Test"

No Operation Tester

Die Diagnostic Box ist das wirkungsvollste Testgerät für die Zentraleinheiten. Sie enthält ein ROM mit einem Testprogramm und wird mittels eines 40-poligen Steckers direkt auf den Prozessor gesteckt. Es werden getestet RAM, ROM, IEC-Bus, User Port, Video Logic, Tastatur und die zwei Cassetten Ports.

Die Diagnostic Box kann sowohl als Dauertest als auch zur gezielten Fehlersuche benutzt werden. Erfahrungsmäßig werden ca 80 % aller Fehler mit Hilfe der Diagnostic Box gefunden.
(nähere Beschreibung siehe Anhang)

Der "RAM/ROM-Test" ist auf Diskette erhältlich und eignet sich zum feststellen von defekten RAM's und ROM's. Dieser Test ist eher als Dauertest als zur Fehlersuche zu benutzen.

Der "No Operation Tester" wird benutzt, um den Adressbus zu überprüfen. Er besteht aus einem 6502 Prozessor an welchem hardwaremäßig der Befehl "EA" verdrahtet wird.
(nähere Beschreibung siehe Anhang)

12. VIDEO BOARD

In den Zentraleinheiten der Serie 40xx/80xx wird zum Ansteuern der Bildröhre ein Video Board der Firma Hitachi verwendet: CDU-B80-CMD12"

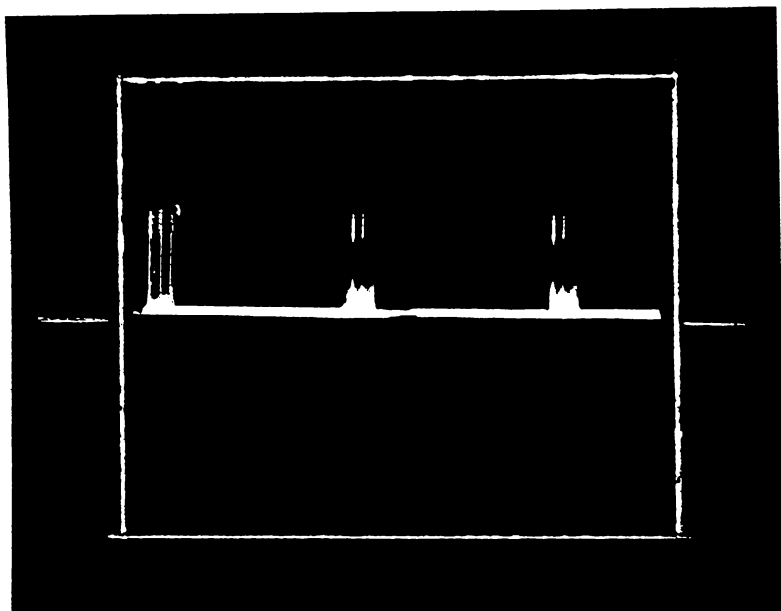
Die Schaltung kann man in vier Teile zerlegen:

Video Signal
Vertikale Ablenkung
Horizontale Ablenkung
Spannungsversorgung

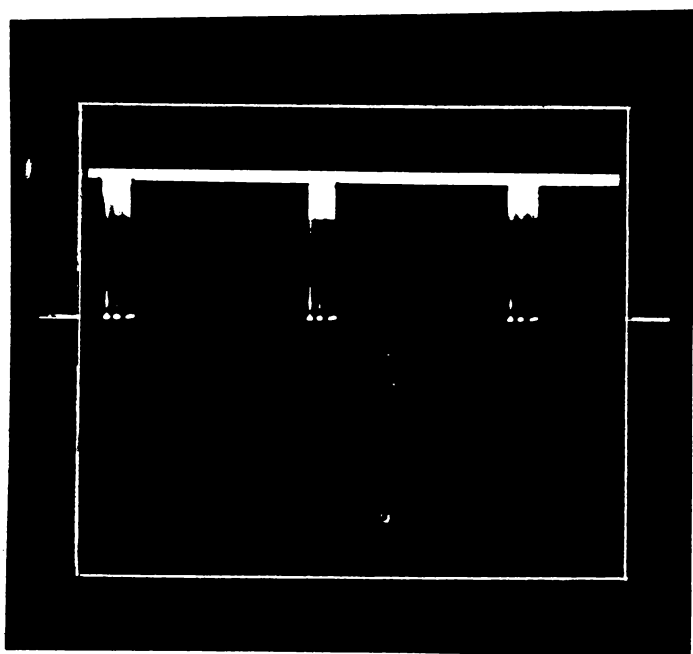
Die auf der Platine verwendeten Bauteile sind im Schaltplan mit dreistelligen Ziffern gekennzeichnet.
Fängt die Ziffer mit "2" an, so gehört das Bauteil zur Logic des Video Signals, "6" gehört zur vertikalen Ablenkung, "7" zur horizontalen Ablenkung und "9" zur Spannungsversorgung.

Potentiometer -----	Funktion -----
R 255	Helligkeit grob
R 261	Helligkeit fein
R 253	Scharfeinstellung (Focus)
R 622	Bildhöhe (Bildfang)
R 612	Linearität
Spule L723	Bildbreite

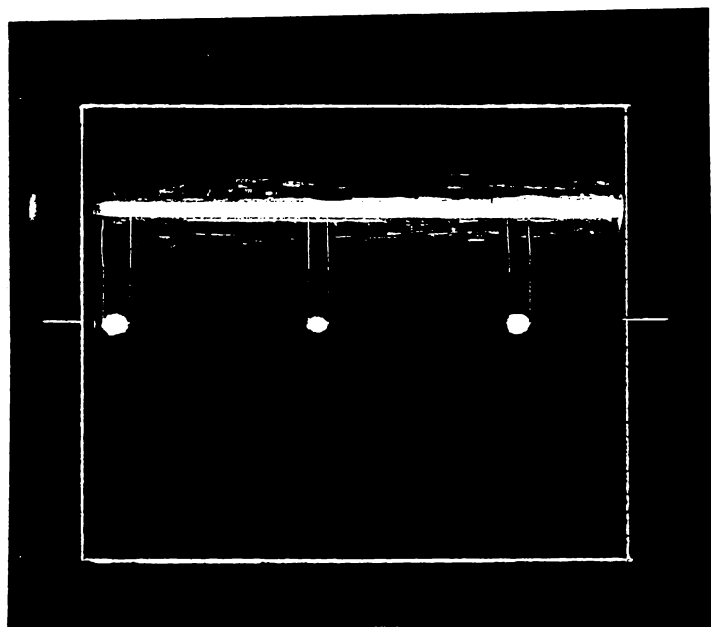
Im Schaltplan sind mit einem Kreis gekennzeichnet die Testpunkte 1 - 18 für welche auf den nachfolgenden Seiten die zugehörigen Oszillogramme abgebildet sind:



Video
Testpunkt 1
2 V/cm
5 ms

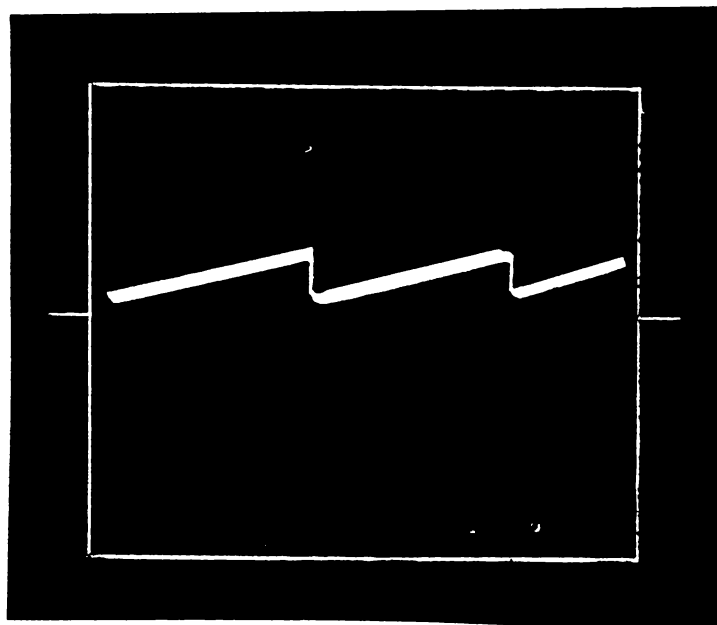


Testpunkt 2
10 V/cm
5 ms

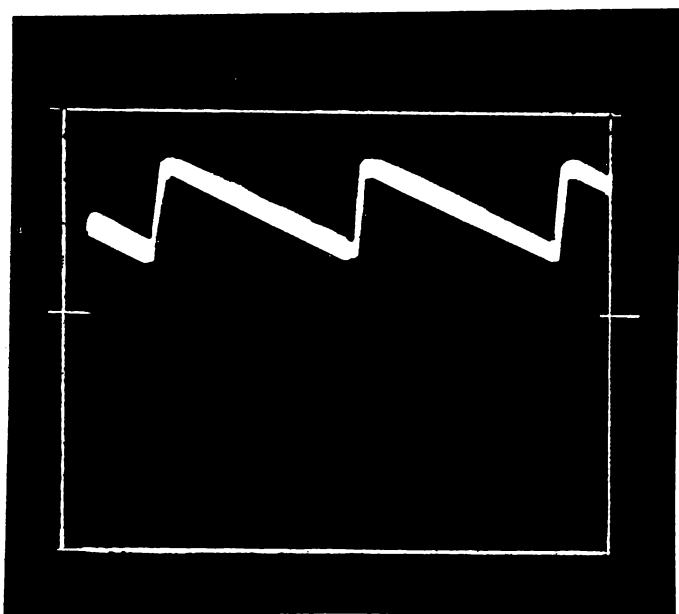


Vertikal Drive

Testpunkt 3
2 V/cm
5 ms

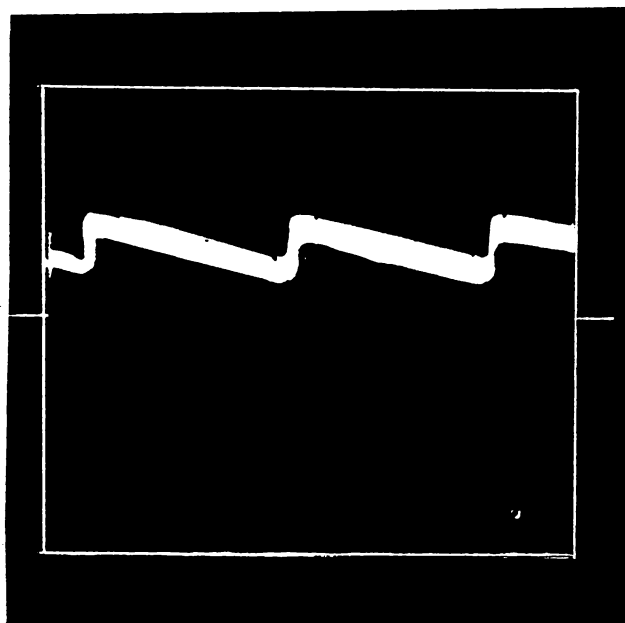


Testpunkt 4
5 V/cm
5 ms

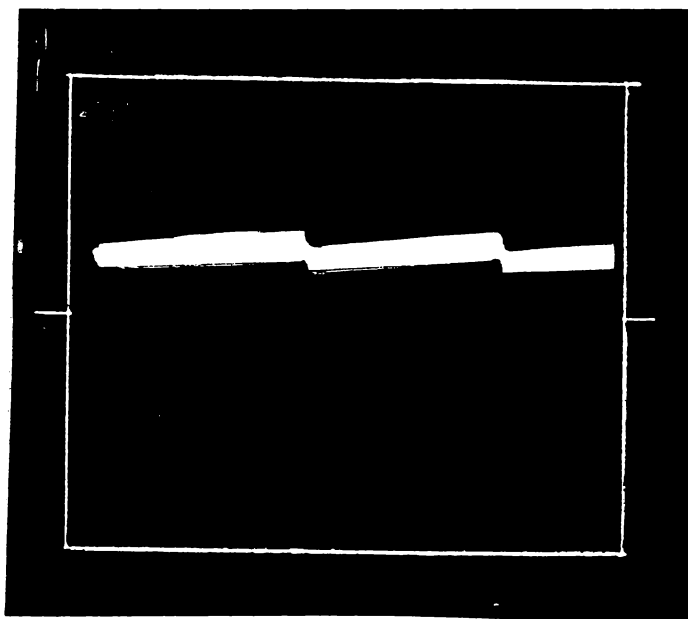


Vertikal Drive

Testpunkt 5
5 V/cm
5 ms

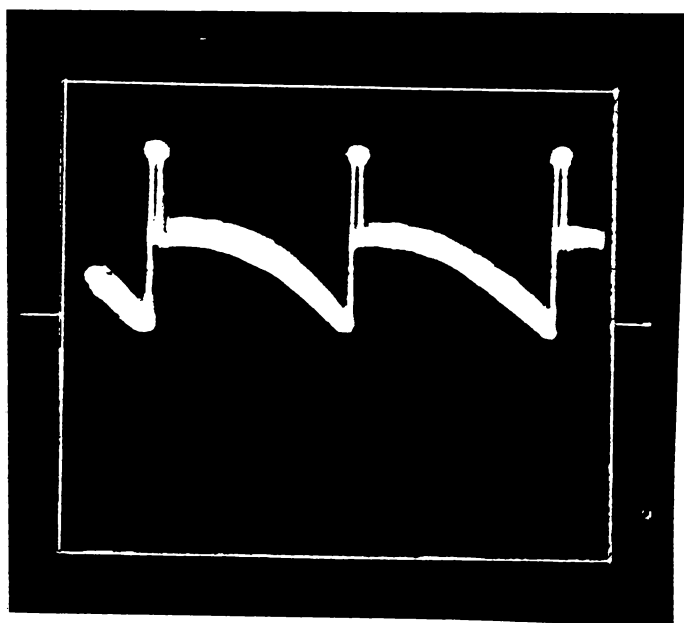


Testpunkt 6
5 V/cm
5 ms

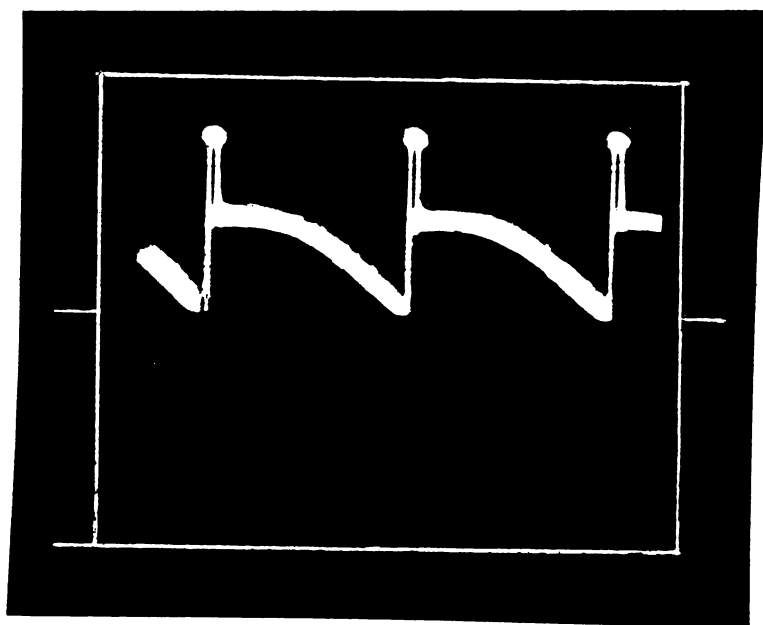


Vertikal Drive

Testpunkt 7
0.5 V/cm
5 ms

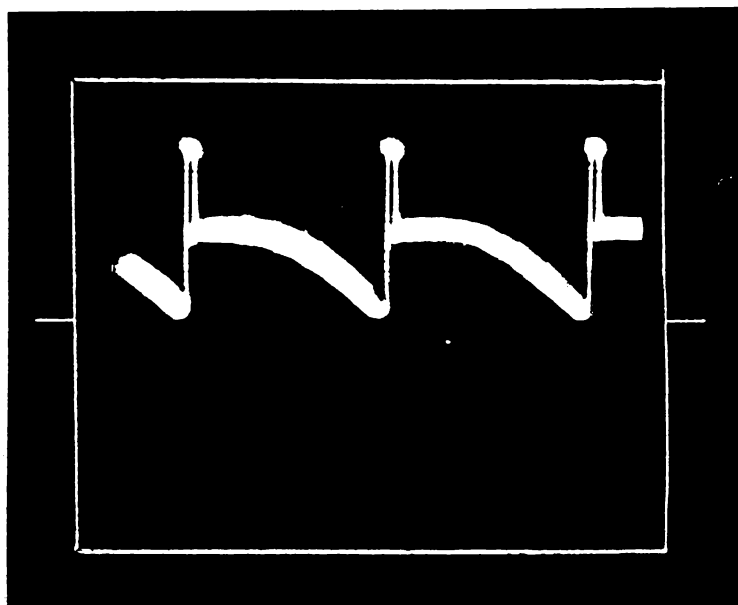


Testpunkt 8
5 V/cm
5 ms

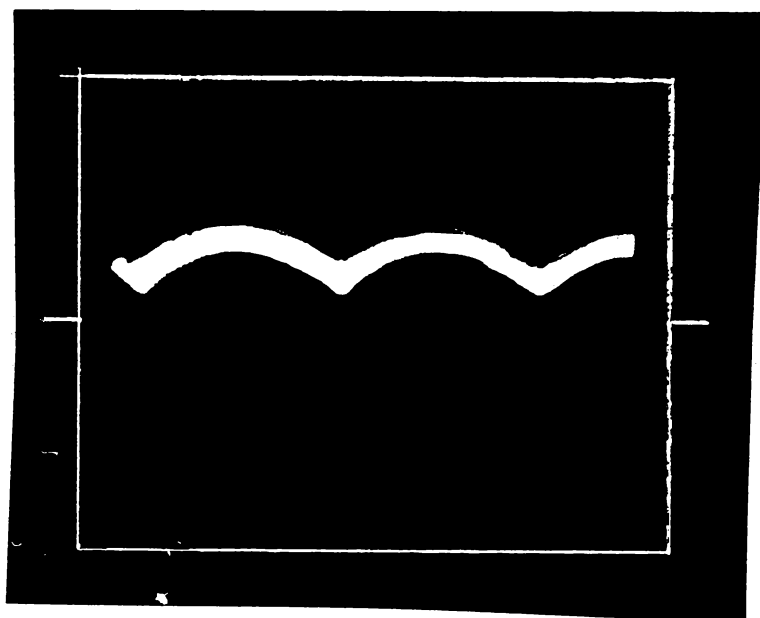


Vertikal Drive

Testpunkt 9
5 V/cm
5 ms

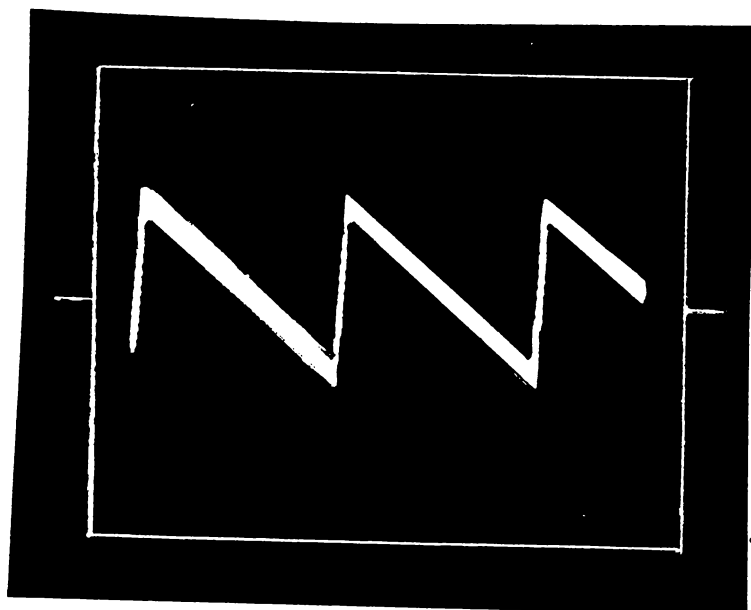


Testpunkt 10
5 V/cm
5 ms

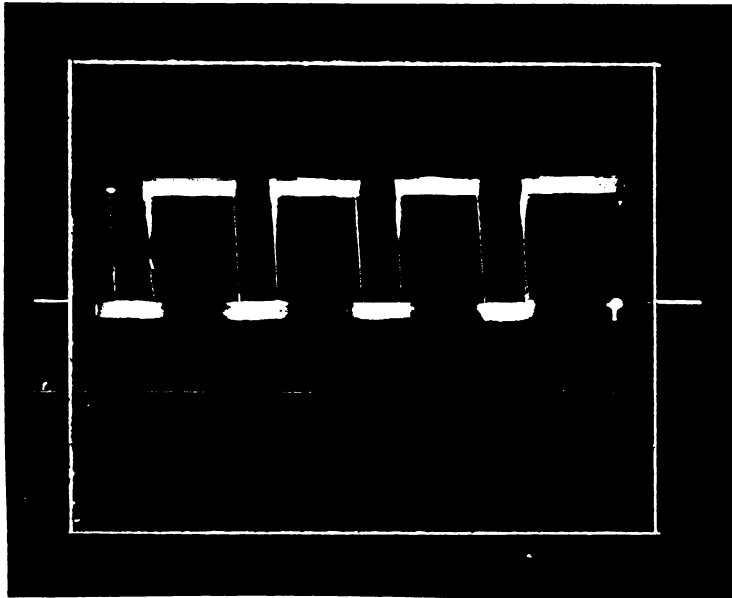


Vertikal Drive

Testpunkt 11
5 V/cm
5 ms

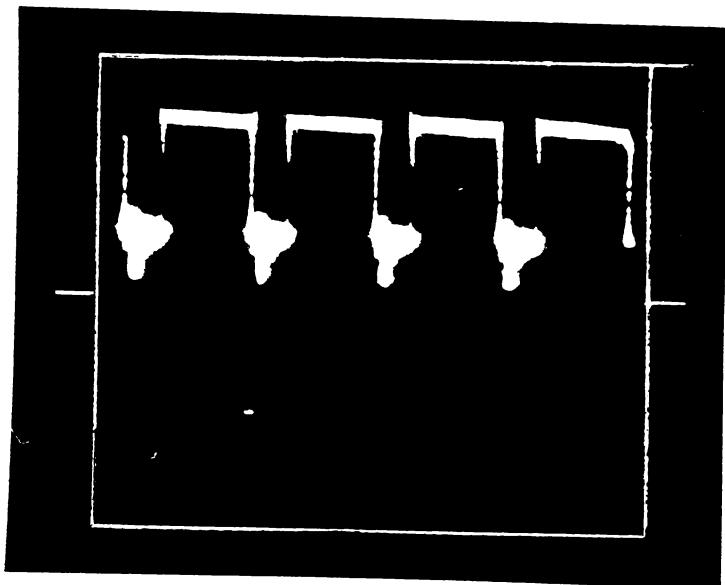


Testpunkt 12
0.5 V/cm
5 ms

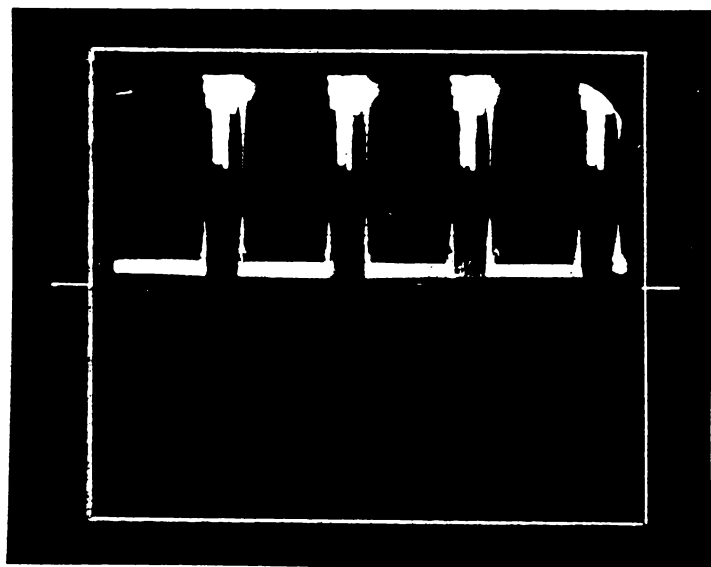


Horizontal Drive

Testpunkt 13
2 V/cm
20 ms

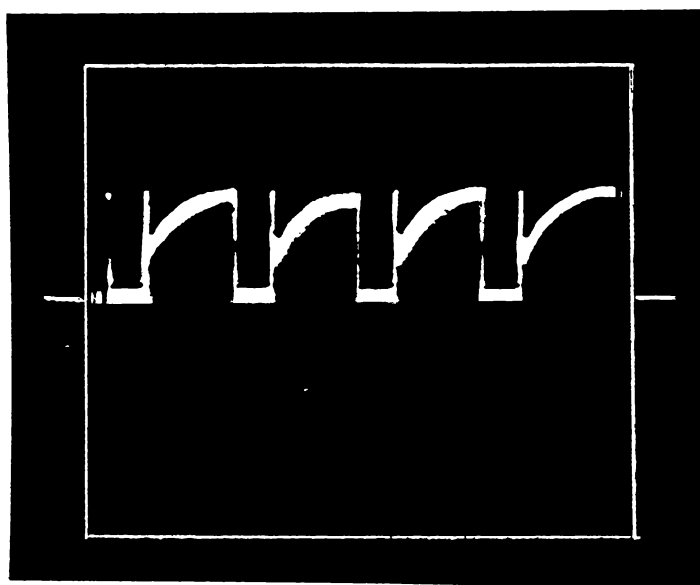


Testpunkt 14
0.2 V/cm
20 ms

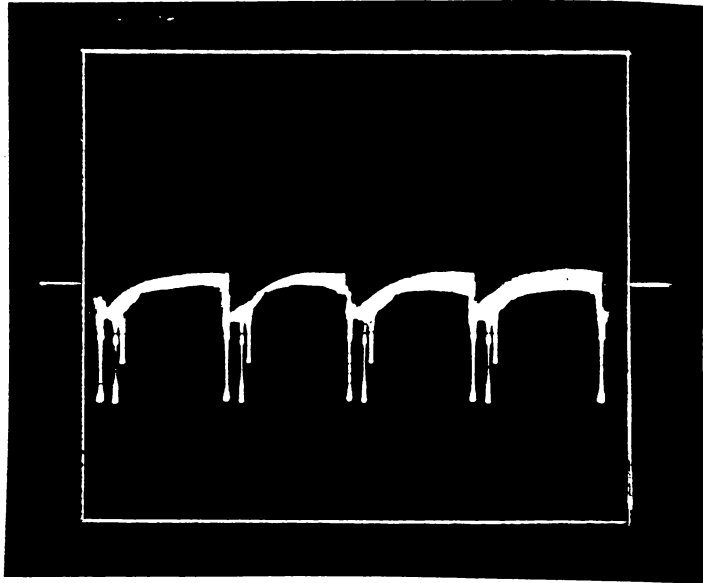


Horizontal Drive

Testpunkt 15
0.2 V/cm
20 ms

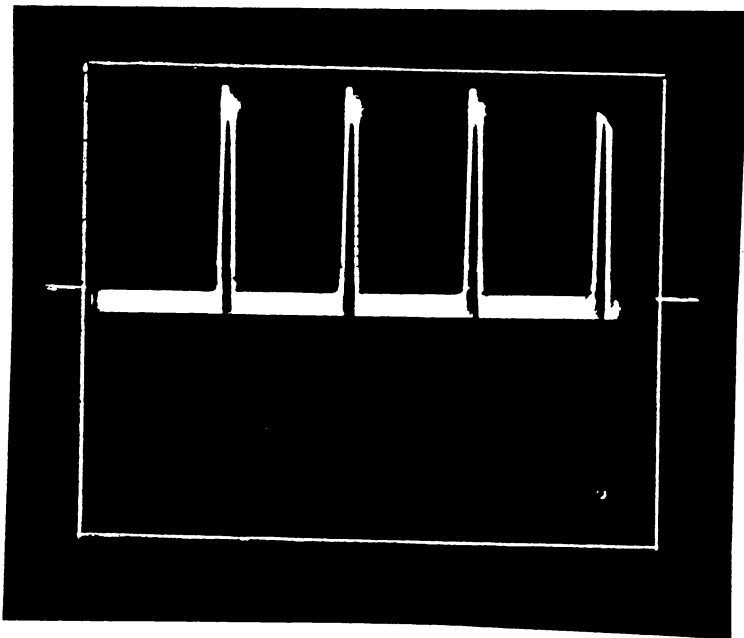


Testpunkt 16
10 V/cm
20 ms



Horizontal Drive

Testpunkt 17
5 V/cm
20 ms



Testpunkt 18
50 V/cm
20 ms

13. Logic-Beschreibung

Folgende Seiten beziehen sich auf die Logic-Blätter der Nummer 8032081, PCB-ASSY-UNIV. DYNAMIC CBM

Die Bezeichnungen der linken Spalte können sein, entweder Signale, Stecker, Brücken oder diverse Bauteile wie Chips, Widerstände usw.

Legende:

- U = integrierte Schaltung
- Q = Transistor
- R = Widerstand
- C = Kondensator
- CR = Diode
- J = Stecker, Brücke

Beispiel: UA3/6 bedeutet: Signal an Pin 6 des IC's in Position A3

Logic-Seite 1 von 11:
(uP, Buffer, Memory Expansion)

- UB14 Mikroprozessor 6502, wird mit 1 MHz getaktet,
 16 Adress-, 8 Datenleitungen, positive Logic
- UB14/34 Read/Write-Signal (high beim Lesen)
- UB14/4 Interrupt Request Eingang (wenn low, wird der Prozessor
 veranlaßt in ein Unterprogramm zu springen:
 50 Hz-Interrupt zur Tastaturabfrage
- UB14/37 Input-Takt 1MHz
- UB14/39 Output " "
- UB14/40 Reset Eingang (beim Einschalten des Systems für ca
 100 ms low
- UD16 Timer zur Erzeugung des Reset-Signals
- UD13/UD14 Adressbus-Buffer
- UE12 4 nach 16 Decoder, erzeugt aus den 4 höchsten
 Adressbits Select-Signale zum Anwählen von
 4k-Memory-Blöcken
- UB9/UB10 Datenbus-Buffer für Display-Rams und Memory Expansion
- UE14/8 NOR-Gatter, erzeugt das Signal x8xx das in Verbindung
 mit Select E zur Adressierung der I/O-Bausteine
 benutzt wird
- J M geschlossen: der Adressbereich der freien ROM-Sockel
 wird dem Mem.Exp. zugewiesen (9000-AFFF)

Logic-Seite 2 von 11:
(Teile des IEC-Bus)

UB16 Peripheral Interface Adapter (6520; PIA)
 programmierbarer I/O-Chip zur Ein/Ausgabe der
 IEC-Bus Daten

UC12,UB17,UA20 IEC-Bus Leistungstreiber MC 3446

J1 IEC-Bus Stecker

Logic-Seite 3 von 11:
(User Port, Tastaturabfrage, Cassetten Ports, IEC-Bus)

UB15 Versatile Interface Adapter (6522,VIA)
 stellt 8 I/O-Leitungen und zwei Steuerleitungen
 für die User Port Schnittstelle(PA0-PA7,CA1,CB2)
 weiterhin werden Signale für den IEC-Bus und Cassetten
 Ports erzeugt

UB15/39 mit dieser Leitung wird Text- oder Graphic-Mode
 ausgewählt.

UB12 PIA, über PA0-PA3 und PB0-PB7 wird die Tastaturabfrage
 gehandhabt, weiterhin werden Signale zum Ansteuern des
 IEC-Bus und der Cassetten Ports ausgegeben.

UB12/18 Vertikal Drive; 50 Hz-Signal das vom CRT-Controller
 abgegeben wird und den IRQ-Ausgang an Pin 37 des PIA
 aktiviert. Dieser Ausgang liegt am IRQ-Eingang des
 Prozessors und verursacht die zur Tastaturabfrage
 notwendigen Interrupts.

Logic-Seite 4 von 11
(ROM-Betriebssystem)

Das Betriebssystem besteht aus vier 4k-Rom's(Typ 2332) und einem 2k-
Rom (Typ 2316).Die ROM's liegen ungepuffert am Datenbus und werden
von den gepufferten Adressen BA0-BA11 adressiert. Jedes ROM wird
durch eine Select-Leitung angewählt (Pin 20, Chip Select).

UD11,UD12 freie ROM-Sockel für die Adressen
 9000-9FFF und A000-AFFF hex.

Logic-Seite 5 von 11
(RAM-Memory)

UA4-UA19 RAM's, Typ 4116, dynamisch, 16k x 1 Bit

UE8 Buffer für Refresh-Adressen

UE9 " " Reihenadressierung (RAS)

UE10 " " Spaltenadressierung(CAS)

CLK 1B low: Refreshadressen werden durchgeschaltet
 high: UE9 oder UE10

MUX A low: Reihenadressen werden durchgeschaltet(RAS)
 high: Spaltenadressen werden durchgeschaltet(CAS)

RAS Row Adress Strobe

CAS0 Column Adress Strobe für untere RAM-Reihe;
 Adressen 0-16k

CAS1 Column Adress Strobe für obere RAM-Reihe;
 Adressen 16-32k

Brücken falls 4116 Ram's verwendet werden sind folgende
 Brücken geschlossen: C,D,G

Logic-Seite 6 von 11:
(Master Timing)

Y1 16 MHz Quarz

UE3 8-Bit Shift-Register, erzeugt aus 16 MHz und 1 MHz
 acht phasenverschobene 1 MHz Takte, welche zur
 Erzeugung von Steuersignalen für die RAM-Memory
 gebraucht werden

UE2 Vier 2 nach 1 Multiplexer
 mit BR/W gesteuert wird je nachdem, ob geschrieben
 oder gelesen wird die benötigte Phase zur Erzeugung
 von RAS0, CAS0, CAS1 und MUX A durchgeschaltet

UD3 4-Bit Zähler
 erzeugt den 1MHz Takt und einen Nadelimpuls für
 die Video-Logic

UE6/UE7 Refresh Adress-Zähler
 wird mit CLK1 getaktet und erzeugt die Adressen
 zur Auffrischung des RAM-Inhalts

Brücken geschlossen für 32k: W,Z

Logic-Seite 7 von 11:
 (Display-RAM Adress Multiplexer)

Hat die Aufgabe entweder die vom Prozessor erzeugten Adressen (BA0-BA10) oder die vom CRT-Controller erzeugten Adressen(TA0-TA9) an die Display-RAM's zu legen. Die Umschaltung geschieht durch CLK1B

Logic-Seite 8 von 11
 (Dot-Generator und Even Display RAM's)

UC3 BCD nach Dezimal Decoder
 erzeugt Signale zum Auswählen der Odd oder Even Rams

BR/W	BA0	aktiver Ausgang
------	-----	-----------------

0	0	write even
0	1	write odd
1	0	read even
1	1	read odd

UC4/UC5 Even Display RAM's
 Typ 2114, 1k x 4 Bit, statisch
 Zeichenkode-Speicher für gerade Bildschirmadressen

UB3 8-Bit Latch Even
 Even Screen Data werden zwischengespeichert und bilden die Adressen A3 - A10 des Charakter-Generators

UA3 Charakter-Generator
 enthält die Daten zur Hell/Dunkelsteuerung des Elektronenstrahls

UA3/6-8 RRA0-RRA2, Signale zur Adressierung der acht Zeilen der Zeichenmatrix

UA3/D0-D7 Zeilendaten der Zeichenmatrix

UA2 Parallel nach Serial Wandler zur Umwandlung der Zeilendaten

LOAD SR bestimmt wann ein neues Byte in den Wandler UA2 eingelesen wird

DISP EN inaktiv, so wird die Hellsteuerung des Elektronenstrahls verhindert (bei Rücklauf zum linken Rand)

NO ROW inaktiv, so wird die Hellsteuerung des Elektronenstrahls verhindert
 im Graphic-Mode immer high, im Text-Mode oszillierend

VIDEO enthält die Daten zur Hell/Dunkelsteuerung des Elektronenstrahls

Logic-Seite 9 von 11

Odd Display RAM's, Zeichenkode-Speicher für ungerade Bildschirm-adressen

Logic-Seite 10 von 11
(CRT-Controller)

- UB13 CRT-Controller (CRT=Cathode Ray Tube)
erzeugt vertikales und horizontales Ablenssignal
und Adressen zur Bildwiederholung
- UB13/3 Pen Strobe, kann zum Anschluß eines Lichtstifts
benutzt werden
- UB13/25 Chip Select, der CRT-Ctr. wird mit den Adressen
59520/21 angesprochen
- UB13/17 Charakter Option; zum Anwählen eines zweiten
Zeichensatzes
- UB13/16 Invert, normal high
low: die Hell/Dunkelsteuerung wird invertiert
- UB13/4-13 TAO-TA9
Display-Ram Adressen
- UB13/34-38 siehe Seite 7 von 11
- UB13/39 horizontaler Synchron-Impuls
- UB13/40 vertikaler " "

Logic-Seite 11 von 11
(Spannungsversorgung)

Spannungsregler -----	Ausgangsspannung -----	Meßpunkt -----
VR 1	-5V	CR1 Kathode
VR2	+12V	CR5 Anode
VR3	+5V	CR6 Anode
VR4	+5V	CR7 Anode
J10/J11	Stecker zur Versorgung von Zusatzplatinen	
J8	Zwischen Pin 1 und 5 liegen ca 17,8V AC vom Trafo	
	" " 8 " 9 "	ca 15,3V AC " "
	" " 2 " 3 (+/-)	ca 10,6V DC zum Elko

DEZIMAL	HEXA-DEZIMAL	ASCII	BILDSCHIRM	DEZIMAL	HEXA-DEZIMAL	ASCII	BILDSCHIRM	DEZIMAL	HEXA-DEZIMAL	ASCII	BILDSCHIRM	DEZIMAL	HEXA-DEZIMAL	ASCII	BILDSCHIRM
0	00	NULL	@	33	21	!	!	66	42	B	☐	99	63	c	☐
1	01	SOH	A	34	22	"	"	67	43	C	☐	100	64	d	☐
2	02	STX	B	35	23	#	#	68	44	D	☐	101	65	e	☐
3	03	ETX	C	36	24	\$	\$	69	45	E	☐	102	66	f	■
4	04	EOT	D	37	25	%	%	70	46	F	☐	103	67	g	☐
5	05	ENQ	E	38	26	&	&	71	47	G	☐	104	68	h	■
6	06	ACK	F	39	27	'	'	72	48	H	☐	105	69	i	▣
7	07	BL	G	40	28	((73	49	I	▣	106	6A	j	☐
8	08	BS	H	41	29))	74	4A	J	▣	107	6B	k	▣
9	09	HT	I	42	2A	*	*	75	4B	K	▣	108	6C	l	▣
10	0A	LF	J	43	2B	+	+	76	4C	L	☐	109	6D	m	▣
11	0B	VT	K	44	2C	,	,	77	4D	M	▣	110	6E	n	▣
12	0C	FF	L	45	2D	-	-	78	4E	N	▣	111	6F	o	☐
13	0D	CR	M	46	2E	.	.	79	4F	O	☐	112	70	p	▣
14	0E	LF	N	47	2F	/	/	80	50	P	☐	113	71	q	▣
15	0F	SI	O	48	30	0	0	81	51	Q	●	114	72	r	▣
16	10	DLE	P	49	31	1	1	82	52	R	☐	115	73	s	▣
17	11	DC ₁	Q	50	32	2	2	83	53	S	♥	116	74	t	☐
18	12	DC ₂	R	51	33	3	3	84	54	T	☐	117	75	u	☐
19	13	DC ₃	S	52	34	4	4	85	55	U	▣	118	76	v	☐
20	14	DC ₄	T	53	35	5	5	86	56	V	▣	119	77	w	☐
21	15	NAK	U	54	36	6	6	87	57	W	○	120	78	x	☐
22	16	SYC	V	55	37	7	7	88	58	X	♣	121	79	y	☐
23	17	ETB	W	56	38	8	8	89	59	Y	☐	122	7A	z	☐
24	18	CAN	X	57	39	9	9	90	5A	Z	◆	123	7B	{	☐
25	19	EM	Y	58	3A	:	:	91	5B	[▣	124	7C	:	▣
26	1A	SUB	Z	59	3B	;	;	92	5C	/	▣	125	7D	}	▣
27	1B	ESC	[60	3C	<	<	93	5D]	☐	126	7E	~	▣
28	1C	FS	/	61	3D	=	=	94	5E	↑	π	127	7F	DEL	▣
29	1D	GS]	62	3E	>	>	95	5F	←	▣				
30	1E	RS	↑	63	3F	?	?	96	60	,	☐				
31	1F	US	←	64	40	@	-	97	61	a	▣				
32	20	┌	┌	65	41	A	♠	98	62	b	▣				

ASCII & BILDSCHIRMCODE

Testbeschreibung

Nach Betätigung der Reset-Taste wird der CRT-Controller initialisiert und der CBM-Zeichensatz in den oberen Teil des Bildschirms geschrieben womit festgestellt wird ob es überhaupt möglich ist Daten auf den Bildschirm zu schreiben.

Nun wird Zero-Page und Stack getestet. Treten hier Fehler auf wird entweder ein RAM als defekt angezeigt oder eine Adressleitung.

Eine genaue Erkennung des Fehlers ist aber in manchen Fällen nicht möglich, sodaß zum Teil unrichtige Fehlermeldungen ausgegeben werden. Um den Fehler zu lokalisieren kann man die CAS-Leitungen an den RAM's vertauschen und den Test neu starten. Wird wieder eine Fehlermeldung ausgegeben, so liegt der Fehler wahrscheinlich nicht an den RAM', sondern an der Ansteuerung(Adressleitungen, Refresh, RAS, CAS, RAM R/W MUX A).

Ist Zero-Page und Stack in Ordnung wird ein Testprogramm in den RAM-Bereich geschrieben und die Aufforderung REMOVE CLIP erscheint auf dem Bildschirm.

Wird nun die Diagnostic-Box vom Prozessor entfernt, werden von dem in der Zero-Page befindlichen Testprogramm die weiteren Teile des Systems getestet. Die ausgegebenen Fehlermeldungen sind im allgemeinen zuverlässig.

CHKSM TEST: Um zu überprüfen, ob das Testprogramm richtig in die Zero-Page geladen wurde, wird eine Prüfsumme gebildet die 00 ergeben muß.

VIDEO RAM TEST: Es werden in jede Bildschirmadresse Daten geschrieben wieder gelesen und verglichen

32K RAM TEST: Es wird der gesamte RAM-Speicher ausser Zero-Page und Stack getestet

ROM TEST: Es werden Prüfsummen von jedem ROM gebildet (siehe Liste der Prüfsummen)

REFRESH TEST: Es werden Daten in die RAM's geschrieben, eine zeitlang gewartet und die Daten wieder ausgelesen. So wird überprüft, ob der Refresh in Ordnung ist.

KEYBOARD TEST: Es wird die Funktion des Decoders und des PIA überprüft

50 HZ IRQ TEST: Es wird getestet, ob von dem Signal Vert Drive über den PIA Interrupts erzeugt werden.

TMR1/2 TEST: Es werden die beiden TIMER des VIA getestet

CASS1/2 TEST Es werden die beiden Cassetten-Ports getestet
Beim Lesen werden Interrupts erzeugt (VIA, PIA)

IEEE Tests: Es werden Daten und Steuersignale getestet. Die IEEE-Buffer werden mitgetestet

HINWEISE ZUR FEHLERSUCHE

Voraussetzung ist Anschluß der Diagnostic-Box vor Einschalten des Gerätes.

Folgende Liste soll helfen defekte Bauteile zu finden, zum einen mit Hilfe der Fehlermeldungen der Diagnostic-Box. Andererseits gibt es öfters Fälle in denen die Diagnostic-Box keine oder falsche Fehlermeldungen auf den Bildschirm schreibt. Auch auf diese Fälle wird eingegangen.

In der linken Spalte steht der Zustand des Bildschirms nachdem das Gerät mit angeschlossener Diagnostic-Box eingeschaltet wurde, oder eine Fehlermeldung der Box.

In der mittleren Spalte steht der Name des Signals oder des Bausteins der defekt sein könnte.

In der rechten Spalte steht die Logic-Seite und die Chip-Position des zu messenden Signals.

Beispiel: 6/UA10,6

bedeutet: Logic-Seite 6; Chip UA10; Pin 6

Bildschirm	I Signal/Chip	I Meßpunkt
-----	I-----	I-----
dunkel	I Datenbus belieb. Bit	I - - -
"	I Buf.Dat.Bus/ "	I - - -
"	I Adressbus/ "	I - - -
"	I CAS0/1	I 6/UD5/11;8
"	I 2nach1Mux	I 6/UE2/15
"	I " /B RW	I 6/UE2/1
"	I " /B RW	I 6/UE2/12
"	I Load SR	I 8/UD1/12
"	I Hsync	I 10/UB13/39
"	I PIA/vert drive	I 3/UB12/18
"	I VIA/CA1	I 3/UB15/40
voller Zeichen	I CLK1B	I 6/UE4/8
"	I SEL8	I 8/UC3/12
"	I Read Even	I 8/UC3/7
"	I Read Odd	I 8/UC3/9
"	I Video Latch	I 8/UD1/13
falsche Zeichen	I Buf. Datenbus	I - - -
"	I ESD0-ESD7	I 8/UB4/UB5
"	I OSD0-OSD7	I 9/UB6/UB7
"	I LSD0-LSD7	I 8/UB3
flimmern und Zeichen in falscher Position	I BAO-BA7	I - - -
Zeichen unregelmäßig verteilt	I Refresh Adr. Buffer	I UE8/1
doppelte Zeichen;Worte	I TA0-TA9	I 10/UB13
Zeichen flimmern im oberen Bildsch. Ber.	I SA0-SA9	I 8/UC4
spaltenweise Zeichen	I write even	I 8/UC3/5
	I	I

Fortsetzung: Hinweise zur Fehlersuche

Bildschirm	I Signal/Chip	I Meßpunkt
Matrixzeilen fehlen	I CRT-Controller;RRA0-RRA2	I 10/UB13
Matrixspalten "	I Charakter Generator Data	I 8/UA3
horizontaler Strich	I VSYNC/CRT-CTR	I 10/UB13/40
Bild verzerrt	I DISP ENA/CRT-CTR	I 10/UB13/18
Fehlermeldungen:		
A7 Adress bad	I FA0-FA6/ADR.Buffer;Ram's	I 5/UE8-UE10
"	I BA0-BA7/Display Buffer	I 8/UB4;5;6;7
"	I MUXA	I 6/UE2/9
"	I RASO	I 6/UD1/10
"	I Ram Buffer enable	I 5/UE5/8;11
"	I SA0-SA9	I 8/UC4
UA19 Ram bad	I	I 6/UD4/8
UC6 Ram bad	I B R/W	I 6/UE2/1
UA15 Ram bad	I	I 6/UE2/12
UC6 Ram bad	I	I 8/UC3/6;14
Uxxx Ram bad	I xxx=Pos. des def. Ram's	I - - -
Ax Adress bad	I x=defektes Adressbit	I - - -
Keyboard bad	I PIA,BCD-DEZ.Decoder	I 3/UB12;UC11
wrong IRQ	I VERT DRIVE/PIA;VIA	I 3/UB12;15;16
no IRQ (cass test)	I PIA;VIA	I 3/UB12;15
IEEE DIO x bad	I PIA; IEC-Buffer	I 2/UB16
IEEE CTR bad	I PIA; VIA	I 3/UB12;15

No Operation-Tester

Den No-Operation-Tester - kurz NOP-Tester genannt - benötigt man um Kurzschlüsse oder Unterbrechungen auf dem Adressbus festzustellen.

Um den NOP-Tester herzustellen, wird ein μP 6502 oder μP 6504 modifiziert. Der NOP-Tester kann überall dort angewandt werden, wo ein 6502 oder 6504 im Einsatz ist.

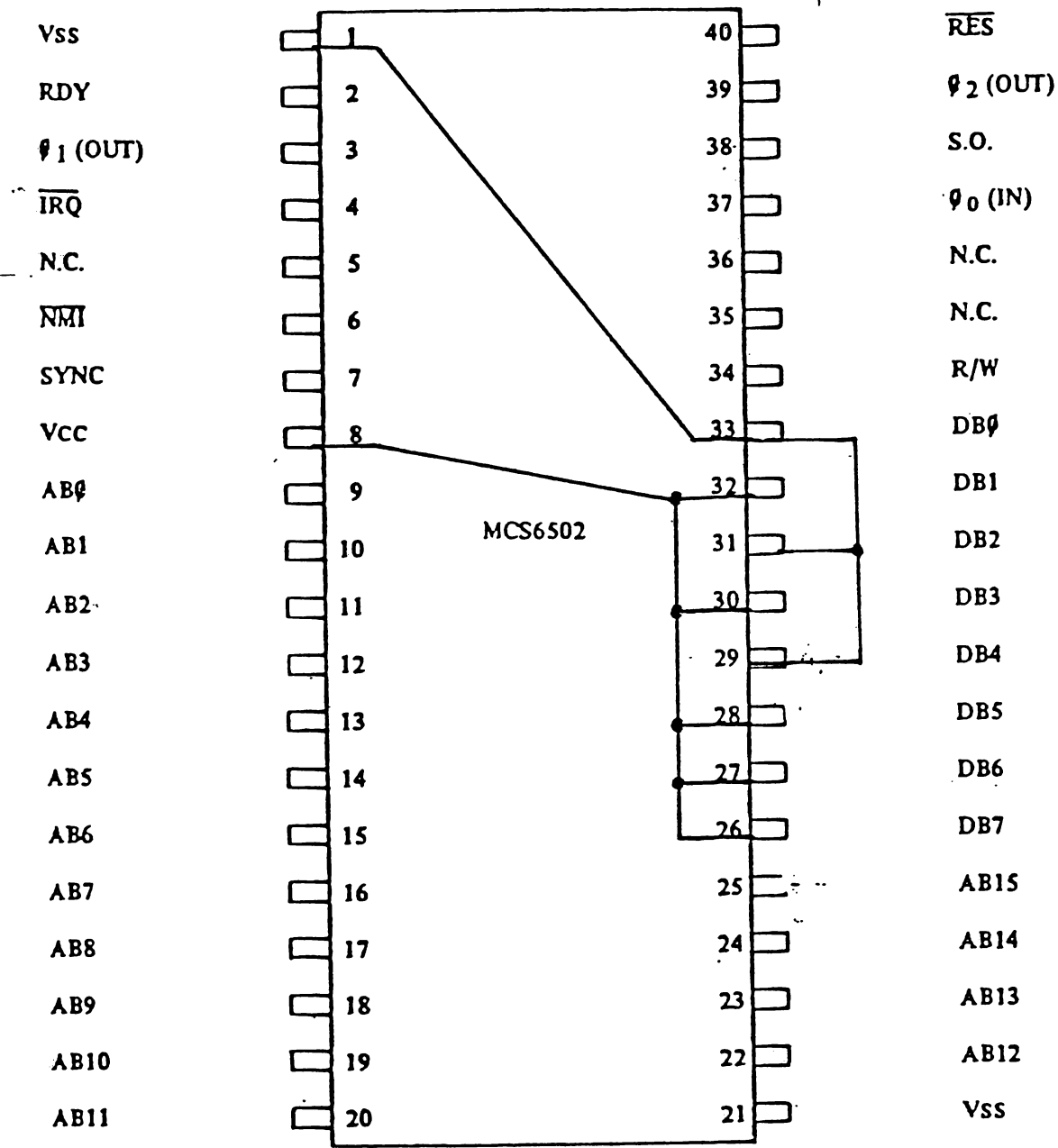
Funktionsweise

Durch die Verdrahtung liegen am Datenbus des Mikroprozessors konstant die Daten EA (Hex) an. Das ist der Betriebscode für den Befehl NOP (no operation). Er veranlaßt den Mikroprozessor, den Programmzähler um 1 hochzuzählen, keine Operationen auszuführen, den nächsten NOP-Befehl zu lesen usw. Der Mikroprozessor zählt so durch alle Adressen (6502 1 bis 65535). Auf den Adreßleitungen sind nun stabile Rechtecksignale. Man hat so auf dem Adreßbus einen Satz von genau vorhersagbaren Signalen, die gut mit dem Oszilloskop verfolgt werden können.

Nachfolgend eine Tabelle der Signale am μP 6504:

<u>μP - P/N</u>	<u>Funktion</u>	<u>μP-Signal</u>
9	AB 0	2 μs
10	AB 1	4 μs
11	AB 2	8 μs
12	AB 3	16 μs
13	AB 4	32 μs
14	AB 5	64 μs
15	AB 6	0,128 ms
16	AB 7	0,256 ms
17	AB 8	0,512 ms
18	AB 9	1,024 ms
19	AB10	2,048 ms
20	AB11	4,096 ms
22	AB12	8,192 ms
23	AB13	16,384 ms
24	AB14	32,768 ms
25	AB15	65,636 ms
4	IRQ	High
6	NMI	High
7	SYNC	2 μs
34	R/W	High
39	PHI2	1 μs
40	RES	High

Verdrahtung des 6502 zum NOP-TESTER



N.C. = NO CONNECTION

Achtung!!!

Pin 26 bis 33 kürzen, damit keine Verbindung zum Sockel entsteht.

1. TECHNISCHE DATEN DES CBM 8032

Höhe : 36 cm
 Breite : 42 cm
 Tiefe : 47 cm
 Gewicht : 22,7 kg
 Spannung : 220 V +/- 10 % / 50 Hz
 Verbrauch : 250 W

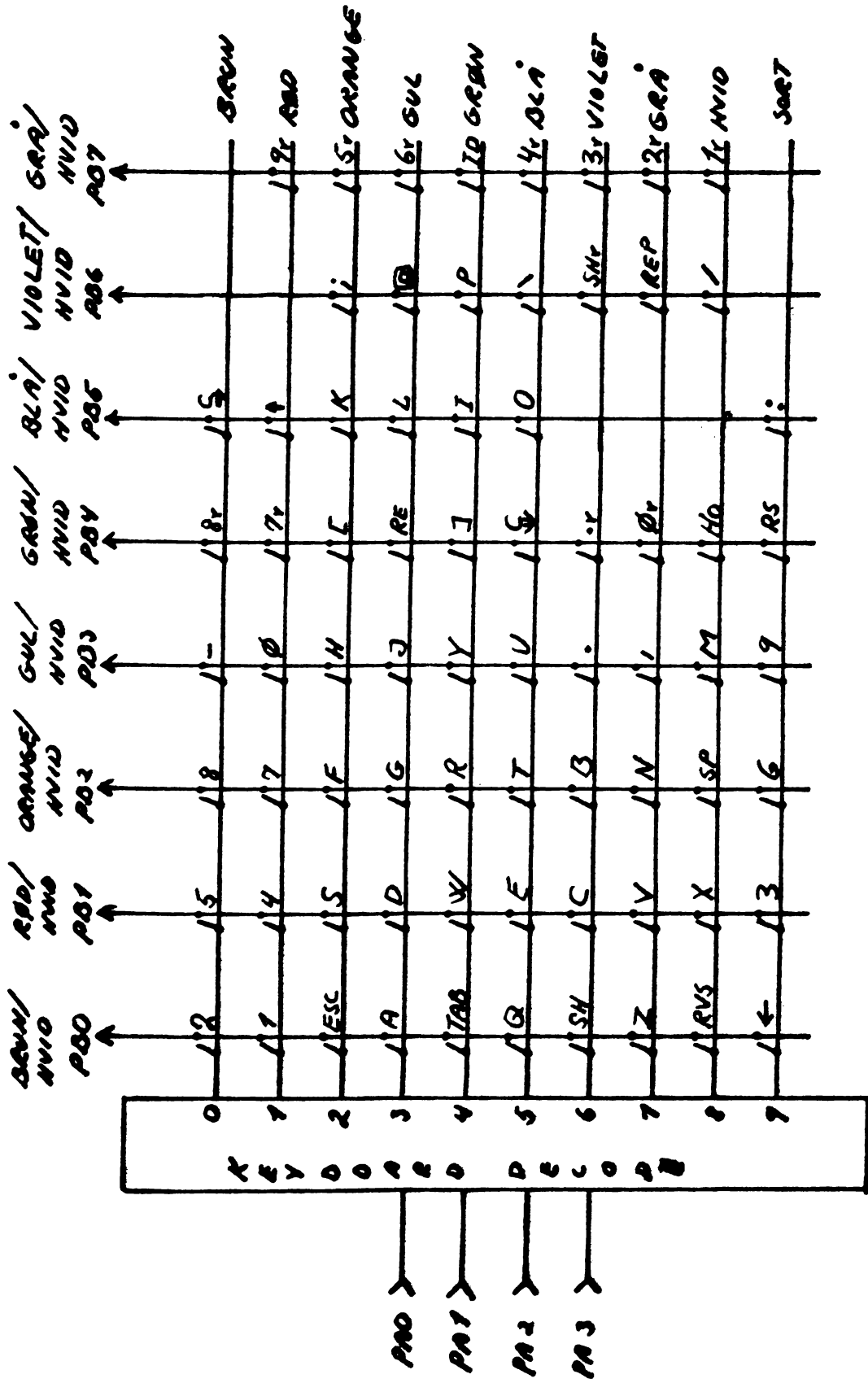
Bildschirm

Diagonale : 31 cm
 Zeilen : 25
 Spalten : 80
 Zeichenauflösung : 8 * 8 Punktmatrix vorprogrammiert
 Darstellungsart : grün auf dunklem Grund
 (programmierbar dunkel auf grünem Grund)
 Sonderzeichen : 64 grafische Zeichen

Programmierung

Sprache : Commodore-BASIC festgespeichert
 in 18 kB ROM
 Anzahl Befehle : 91
 Klartextmeldungen : 26
 Kapazität (RAM) : 32 kB
 Kassette : Rekorder für Standard Musikkassetten
 Aufzeichnungsformat : Einspur-Frequenzmodulation nach
 eigenem Standard
 Anschlüsse : IEC-Bus
 8-bit Parallel mit 2 Handshake-
 Leitungen
 1. Kassettenrekorder
 2. Kassettenrekorder
 Interner Bus
 Prozessor : 6502
 Taktfrequenz : 1 MHz

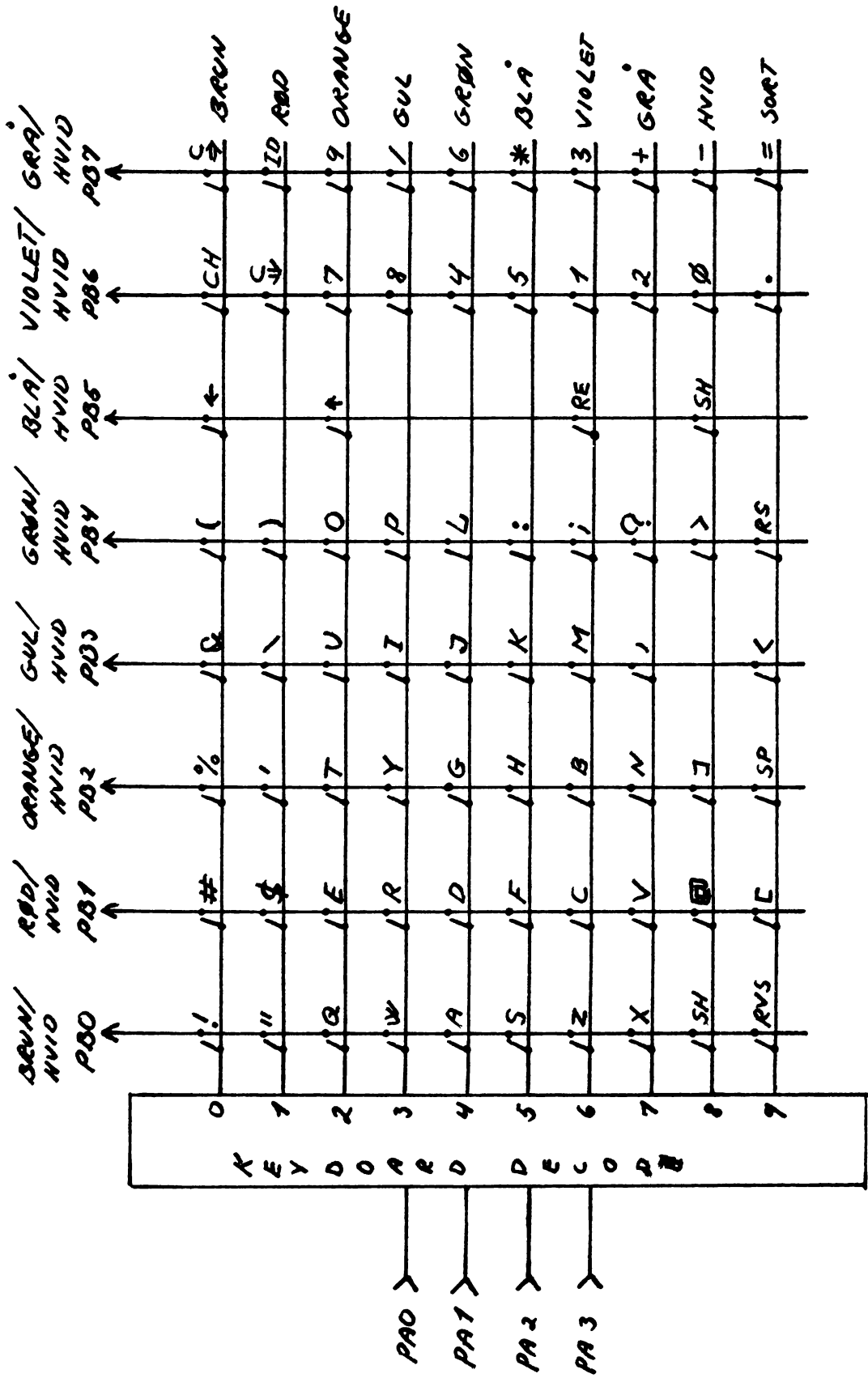
CBM KEYBOARD MATRIX B TASTATUR



6520 DATA REGISTER ADDRESS PA = 59408
 PB = 59410

r = TAL KEYBOARD

CBM KEYBOARD MATRIX N TASTATUR



6520 DATA REGISTER ADDRESSER PA = 59408
PB = 59410

CBM 8296

SUPPLEMENT to the 8032 Manual		Seite
1.	Introduction	1
2.	Memory system	1
2.1	Memory Mapping	1
2.2	Control Register	3
2.3	Modification of memory system	4
2.4	User Jumper	5
3.	Video-RAM	6
4.	Hardware expansion via J4, J9	7
4.1	Memory expansion bus J4, J9	7
4.2	Power supply for hardware expansion	7
5.	Programming the expansion memory bank	8
5.1	I/O Handling	8
5.2	Interrupt Processing	9
5.3	Interrupt Handler Program	9
Annex		
A	Memory map	12
B.1	Connectors 8296	13
B.2	Connectors, modification as compared with 8032	17
C	Software addendum	18

1. Introduction

The CBM 8296 is an expanded version of the CBM 8032, and is fully software-compatible with the CBM 8096. Where the hardware is concerned, there are some amendments and additional options, since the CBM 8296 has a total of 128 k RAM. The plug arrangements are also slightly different.

2. Memory system

The CBM 8296 has two RAM banks, each with 64 k bytes, designated here the main memory and the expansion memory.

The main memory is activated on switch-on, and the computer reacts like a standard CBM 8032.

2.1. Memory Mapping

Expansion memory can be mapped into main memory addresses \$8000 through \$FFFF. See Figure 2-1. Only two of the 16k expansion blocks can reside in main memory at one time. This provides an additional 32k bytes of memory to the user.

Selection of the expansion blocks is by bits 2 and 3 of the expansion memory control register. Each 16k block has a 16k alternate that can be selected by bits 2 and 3 of the control register.

Main memory addresses \$8000 through \$BFFF can only be mapped by expansion blocks 0 or 1. Main memory addresses \$C000 through \$FFFF can only be mapped by expansion blocks 2 or 3.

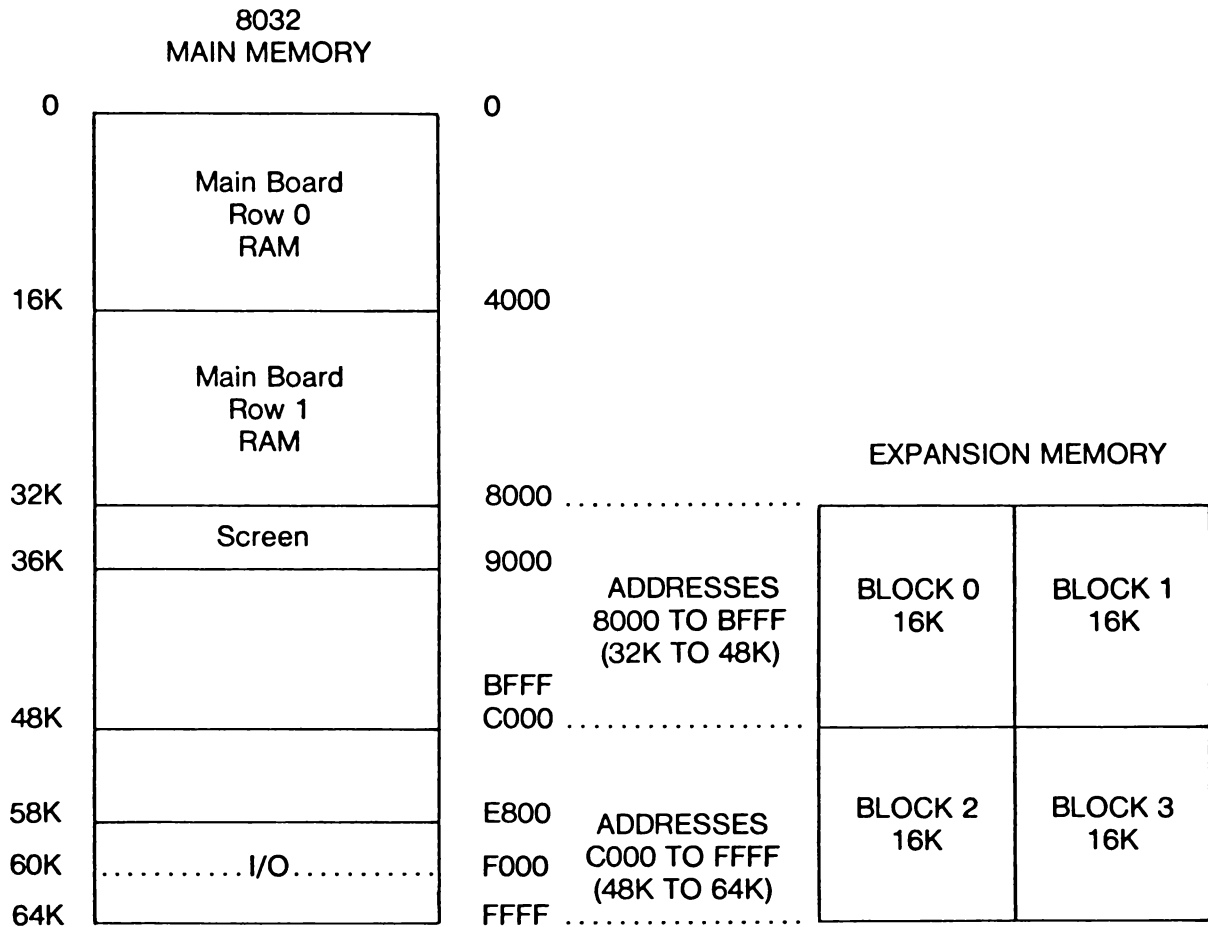


Figure 2-1. Expansion Memory Mapping

2.2. Control Register

Control of the expansion memory is through a memory control register located at address \$FFF0. The memory control register provides selection of 16k-byte blocks, write protection, enabling the expansion memory, I/O peek through and screen peek through. Because the memory control register is write only, a copy of the register should be kept in the lower 32k of main memory.

Figure 2-2 depicts the functions of the memory control register. The paragraphs following it describe these functions in detail.

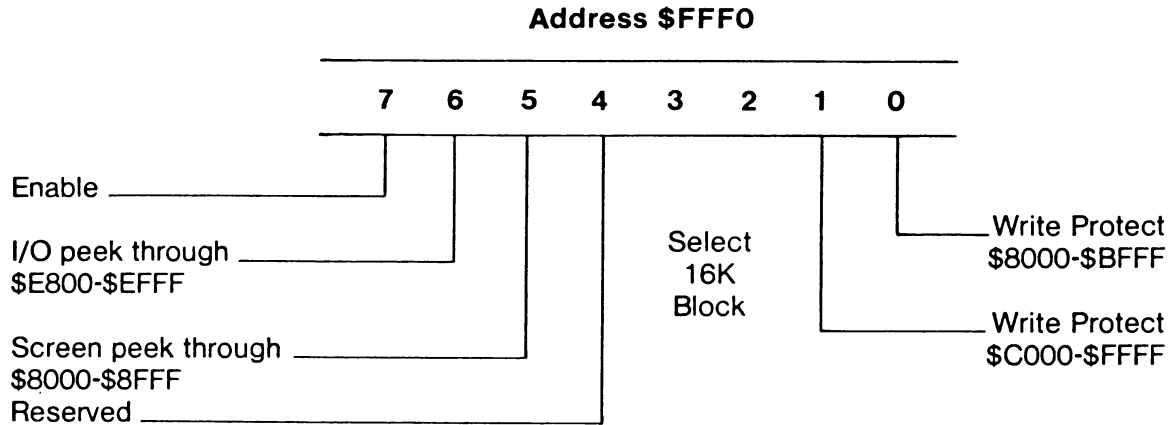


Figure 2-2.
Expansion Memory
Control Register

0	0	2 and 0
0	1	2 and 1
1	0	3 and 0
1	1	3 and 1

Control Register Bit 7 - When equal to 1, enables the expansion memory. When bit 7 equal to 0, expansion memory is disabled. Bit 7 defaults to 0 on power up.

Control Register Bit 6 - When equal to 1, I/O peek through is enabled.

Control Register Bit 5 - When equal to 1, screen peek through is enabled.

Control Register Bit 4 - Reserved.

Control Register Bit 3 - When equal to 1, block 3 (16k-byte) is selected. When equal to 0, block 2 (16k-byte) is selected.

Control Register Bit 2 - When equal to 1, block 1 (16k-byte) is selected. When equal to 0, block 0 (16k-byte) is selected.

Control Register Bit 1 - When equal to 1, addresses \$C000 through \$FFFF on the *Expansion Memory Board only* are write protected. I/O is not write protected if I/O peek through is enabled. When equal to 0, the addresses are not write protected.

Control Register Bit 0 - When equal to 1, addresses \$8000 through \$BFFF on the *Expansion Memory Board only* are write protected. The screen is not write protected if screen peek through is enabled. When equal to 0, the addresses are not write protected.

2.3. Modifications to the main memory system

By contrast with the CBM 8032, each address in the 64 k main memory is covered by RAM. A write command into any address (apart from I/O) causes the corresponding RAM address to be written at, and a read command reads from RAM between \$0000 and \$8FFF; above \$9000 from the empty ROM sockets, from the ROM or from the I/O devices.

RAM	\$0000 - \$FFFF
Empty socket	\$9000 - \$AFFF
ROM	\$B000 - \$FFFF, except \$E8XX
I/O	\$E800 - \$E8FF

The memory system for CPU read access can be modified by means of the signals RAM SEL 9, RAM SEL A, and RAM ON (J4, Pins 12, 13, 14) in accordance with the following table:

Control Register \$FFFO		main memory \$8000-\$FFFF									
CR 7	CR 6	RAM ON	RAM SEL 9	RAM SEL A	\$E800 -\$E8FF	\$F000 -\$FFFF	\$E000 -\$E7FF \$E900 -\$EFFF	\$B000 -\$BFFF	\$A000 -\$AFFF	\$9000 -\$9FFF	\$8000 -\$87FF
0	X	1	1	1	I/O	Kernal	Editor	BASIC	EPROM	EPROM	SCREEN
0	X	1	1	0	I/O	Kernal	Editor	BASIC	RAM	EPROM	SCREEN
0	X	1	0	1	I/O	Kernal	Editor	BASIC	EPROM	RAM	SCREEN
0	X	1	0	0	I/O	Kernal	Editor	BASIC	RAM	RAM	SCREEN
0	X	0	1	1	I/O	Kernal	Editor	RAM	PAM	RAM	SCREEN
0	X	0	0	1	I/O	Kernal	RAM	RAM	RAM	RAM	SCREEN
0	1	0	X	0	I/O	RAM	RAM	RAM	RAM	RAM	SCREEN
0	0	0	X	0	RAM	RAM	RAM	RAM	RAM	RAM	SCREEN

2.4. User Jumper

These three signals can be set either fixed or programmable via user jumpers. Bits 0, 1 and 2 are used for control under program control.

Jumper	Description of function
--------	-------------------------

JU1	$\overline{\text{RAM SEL A}} = \text{LOW}$
JU2	$\overline{\text{RAM SEL 9}} = \text{LOW}$
JU3	$\overline{\text{RAM SEL A}}$ to Port A0
JU4	$\overline{\text{RAM SEL 9}}$ to Port A1
JU5	$\overline{\text{RAM ON}}$ to Port A2

To use the user jumpers JU3 to JU5, Pins 0, 1 and 2 of the user port must be programmed to output.

DR = 59459 (\$E843)
DA = 59471 (\$E84F)

POKE DR, PEEK (DR) OR 7

The desired combination of bits (0 7) can then be poked into the port register DA.

N.B. When using JU3, JU4 and JU5, make sure that the user port is not written at accidentally. This can happen for instance in a number of text processing programs, which emulate a Centronics interface via the user port.

The signal $\overline{\text{RAM ON}}$ must be high at the start, as it is not otherwise possible to complete the POWER ON routine in the ROM.

3. Video RAM

As previously, the video RAM is at \$8000 - \$87CF. With the CBM 8296 however, the CRT controller has access to the full range of addresses from \$8000 - \$9FFF. This produces 3 further screen pages, which can be used for program menus, help lists and the like. The four screen pages can be written at any time. The operating system uses the first page only.

Page turning on display is achieved by altering the VIDEO RAM start address in the CRT controller (reg. 12):

```
POKE 59520, 12
POKE 59521, x
```

wherein x=0 for page 1 (\$8000-\$87CF, switch-on figure)
x=4 for page 2 (\$8800-\$8FCF)
x=8 for page 3 (\$9000-\$97CF)
x=12 for page 4 (\$9800-\$9FCF)

Activating the switchover text-modus and graphic-modus brings the screen page 1 back each time to display.

To use pages 3 and 4, the IC socket UE10 must be free, or the RAM SEL 9 signal set at LOW. Otherwise the ROM in the UE10 transmits data at the same time.

For CPU read access to screen pages 3 and 4, RAM SEL 9 must again be set at LOW (see 2.).

Other parameters of the screen controller should be modified by comparison with the standard operating conditions (text or graphic) only with extreme caution. The dynamic RAM memories used require a cyclical refresh, which is produced by the CRT controller. To be sure of the refresh, the CRT controller must give at least 256 symbols in an interval of 4 ms.

4. Hardware expansions via J4, J9

4.1. Memory expansion bus J4, J9

Hardware expansions can be linked via the memory expansion bus (J4, J9).

The data buffer for the expansion bus is, as with the CBM 8032, always switched in as an output. If the CPU is to read from the expansion bus, close user jumper JU6 for the range \$A000 to \$AFFF, and user jumper JU7 for the range \$9000 to \$9FFF. The ROM sockets UE10 or UE9 must of course than remain free. The expansion bus cannot be read in the range \$8000 to \$8FFF.

A further way of gaining access to the expansion bus is provided by the line SEL EXP (J4, Pin 15). Low-level on this line switches the data bus buffer of the expansion bus to input on a read command, independently of the address on the CPU. Care should therefore be taken to avoid bus conflicts when using SEL EXP. Make sure that only one data source (RAM, ROM I/O or expansion bus) is activated.

The control line SEL EXP must be switched with a collector gate. Jumpers JU6 and JU7 may also be closed.

If the RAM SEL 9 or RAM SEL A line is activated, the RAM is switched in to this address range, and the expansion bus is switched out.

4.2. Power supply for hardware expansions

Connectors J10 and J11 are available for this purpose. As previously, ground and +9V DC unregulated are available at J11 (0.8 A max.). J10 is modified as compared with the CBM 8032, and now connected to 8V AC (1 A max.) and ground. Not more than 1A may be taken out via J10 and J11.

5. Programming the expansion memory bank

The user may choose to write his own programs to operate the expansion memory. A sample program is included in this chapter.

I/O Handling

I/O in the CBM 8032 consists of the following five devices:

1. A 6520 PIA at \$E810
2. A 6520 PIA at \$E820
3. A 6522 PIA at \$E840
4. A CRT controller at \$E880
5. Screen memory at \$8000 through \$87FF

These five I/O devices may be accessed in two ways. The first way is to simply switch out the expansion memory and restore main memory. This may be already accomplished by the memory manager software when a CBM I/O subroutine is called.

The second way (necessary when a RAM-loaded application program accesses I/O) uses the I/O peek through feature. Bit 6 of the control register enables I/O peek through.

NOTE

When using the I/O peek through, ensure that the currently running subroutine does not reside over the top of the peek through.

Accessing screen memory is accomplished in the same way as accessing the other I/O devices. When accessed, screen memory is seen as 25 lines of 80 columns. The data is stored row-wise as sequential bytes.

The CRT display circuitry cannot directly display out of the expansion RAM. A suggested memory manager function is to page whole screen-fulls of data out of the expansion RAM.

5.2 Interrupt Processing

The 6502 microprocessor is designed for a simple system architecture in which the lower 32k of the address space is RAM and the upper 32k is ROM. This allows the microprocessor to fetch the starting address of the first instruction out of ROM upon reset. The result is that three hardware vectors are stored in addresses \$FFFA - \$FFFF.

The memory manager must accomplish the following functions

1. Ensure that there is a valid address at each ROM address in the two 16k expansion blocks that are active. The only exception is if interrupts are disabled by a SEI instruction executed before bit 7 is set to a 1.
2. To avoid being interrupted when changing a vector, execute a SEI.
3. The ROM interrupt vectors in the CBM point to routines in ROM which are not accessible when the expansion memory is selected. For that reason, the memory manager should a) set the vectors to point at a routine that switches to main memory mode, b) call the interrupt service routine, and c) restore the expansion memory mode.

5.3 Interrupt Handler Program

The following sample program (Refer to Table 5-1) handles interrupts and passes control to the ROM routines for processing. The global variable MEMMAP contains the current contents of the control register. The user program must first call the INIT routine to load the 6502 interrupt vectors into the expansion RAM. After that, interrupts are pre-processed by the subroutines NMI and IRQ. Interrupts are exited by RTIP

Table 5-1. Interrupt Handler Program

```
; INITIALIZE INTERRUPT PROCESS
;
; ASSUME MEMMAP SETUP
;
INIT   SEI
      LDA MEMMAP
      STA $FFFO
;
; INIT USER IRQ VECTOR
;
      LDA #<IRQ
      STA $FFFE
      LDA #>IRQ
      STA $FFFF
;
; INIT USER NMI VECTOR
;
      LDA #<NMI
      STA $FFFA
      LDA #>NMI STA $FFFB
;
      CLI
;
      RTS
;
; PROCESS IRQ
;
IRQ    STA TMPA                ; PRESERVE .A
;
      PLA
      PHA
      STA TMPPS
;
      LDA #%00000000
      STA $FFFO
;
; PUSH RETURN FROM INTERRUPT ADDRESS
;
      LDA #>RTIP
      PHA
      LDA #<RTIP
      PHA
;
      LDA TMPPS                ; PUSH DUMMY STATUS
      PHA
;
      LDA TMPA                ; RESTORE .A
;
; GO TO ROM IRQ SERVICE
;
      JMP ($FFFE)
```

Table 5-1. Interrupt Handler Program (Continued)

```
; PROCESS NMI
;
; PROCESS NMI
;
NMI   STA TMPA                               ; PRESERVE .A
      ;
      PLA
      PHA
      STA TMPPS
;
;
      LDA #%00000000
      STA $FFFO
;
; PUSH RETURN FROM INTERRUPT ADDRESS
;
      LDA #>RTIP
      PHA
      LDA #<RTIP
      PHA

      LDA TMPPS
      PHA                                PUSH DUMMY STATUS

      LDA TMPA
;
; GO TO ROM IRQ SERVICE
;
      JMP ($FFFA)
;
; RETURN FROM INTERRUPT PROCESS
;
RTIP  PHA
;
; MAP BACK TO ORIGINAL RAM
;
      LDA MEMMAP
      STA $FFFO
;
; RESTORE OLD .A
;
      PLA
;
; BACK TO USER
;
      RTI
      .END
      .LIB MONITOR
```

A Memory Map

DEFAULT		MEMORY-MAP FOR CBM 8296										64K RAM UB1-UB8		for diagnostic		64K RAM UA1-UA8				
Address	Description	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	5 3/4K RAM	FREE	FREE	I/O	RAM	FREE	FREE	2 x 32 K zusätzl. RAM	
\$F000-\$FFFF	4K ROM KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL	KERNAL									
\$E900-\$EFFF	1 3/4K ROM EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR									
\$E800-\$E8FF	1/4K I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O	I/O									
\$E000-\$E7FF	2K ROM EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR	EDITOR									
\$8000-\$DFFF	12K ROM BASIC	BASIC	BASIC	BASIC	BASIC	BASIC	BASIC	BASIC	BASIC	BASIC	BASIC									
\$A000-\$AFFF	4K ROM USER	ROM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM									
\$9000-\$9FFF	4K ROM USER	RAM	ROM	RAM	RAM	RAM	RAM	RAM	RAM	RAM	RAM									
\$8800-\$8FFF	2K RAM USER	2K RAM	2K RAM	2K RAM	2K RAM	2K RAM	2K RAM	2K RAM	2K RAM	2K RAM	2K RAM									
\$8000-\$87FF	2K RAM SCREEN	2K SCREEN	2K SCREEN	2K SCREEN	2K SCREEN	2K SCREEN	2K SCREEN	2K SCREEN	2K SCREEN	2K SCREEN	2K SCREEN									
\$0000-\$7FFF	32K RAM	32K RAM	32K RAM	32K RAM	32K RAM	32K RAM	32K RAM	32K RAM	32K RAM	32K RAM	32K RAM									
		RAMON = 1 RAMSELA = 1 RAMSEL9 = 1 CR6 = X	RAMON = 1 RAMSELA = 1 RAMSEL9 = 0 CR6 = X	RAMON = 1 RAMSELA = 0 RAMSEL9 = 1 CR6 = X	RAMON = 1 RAMSELA = 0 RAMSEL9 = 1 CR6 = X	RAMON = 1 RAMSELA = 0 RAMSEL9 = 1 CR6 = X	RAMON = 1 RAMSELA = 0 RAMSEL9 = 1 CR6 = X	RAMON = 1 RAMSELA = 0 RAMSEL9 = 1 CR6 = X	RAMON = 1 RAMSELA = 0 RAMSEL9 = 1 CR6 = X	RAMON = 1 RAMSELA = 0 RAMSEL9 = 1 CR6 = X	RAMON = 1 RAMSELA = 0 RAMSEL9 = 1 CR6 = X	RAMON = 0 RAMSELA = 0 RAMSEL9 = X CR6 = 1	RAMON = 0 RAMSELA = 0 RAMSEL9 = X CR6 = 1	RAMON = 0 RAMSELA = 0 RAMSEL9 = X CR6 = 1	RAMON = 0 RAMSELA = 0 RAMSEL9 = X CR6 = 1	RAMON = 0 RAMSELA = 0 RAMSEL9 = X CR6 = 1	RAMON = 0 RAMSELA = 0 RAMSEL9 = X CR6 = 1	RAMON = 0 RAMSELA = 0 RAMSEL9 = X CR6 = 1	RAMON = 0 RAMSELA = 0 RAMSEL9 = X CR6 = 1	RAMON = 0 RAMSELA = 0 RAMSEL9 = X CR6 = 1
		NO ROM = 1										NO ROM = 0		NO ROM = X						
		CR7 = 0										CR7 = 1		CR7 = 1						

IEEE

B.1

connectors 8296

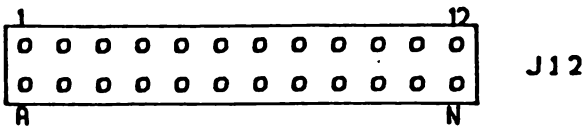
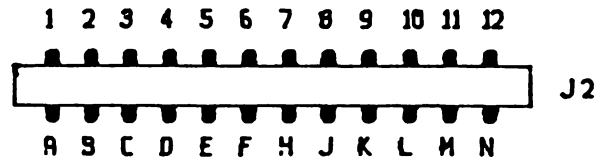
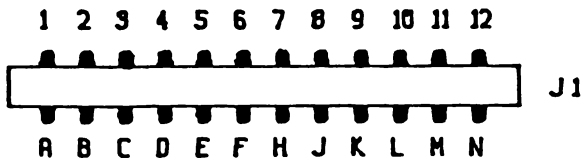
USER - PORT

Arrangement of pins plug J1/J12

PIN	SIGNAL	PIN	SIGNAL
1	DIO 1	A	DIO 5
2	DIO 2	B	DIO 6
3	DIO 3	C	DIO 7
4	DIO 4	D	DIO 8
5	EOL	E	REN
6	DAV	F	GND
7	NRFD	H	↓
8	NDAC	J	↓
9	IFC	K	↓
10	SRQ	L	↓
11	ATN	M	↓
12	CHASSIS GND	N	GND

Arrangement of pins plug J2

PIN	SIGNAL	PIN	SIGNAL
1	GND	A	GND
2	VIDEO	B	CA1
3	SRQ IN	C	PA0
4	EOT	D	↓ 1
5	DIAG	E	↓ 2
6	*2CASSREAD	F	↓ 3
7	CASSWRITE	H	↓ 4
8	*1CASSREAD	J	↓ 5
9	VERT DRIVE	K	↓ 6
10	HORZ DRIVE	L	↓ 7
11	GRAPHIC	M	CB2
12	GND	N	GND



CASSETTE 1

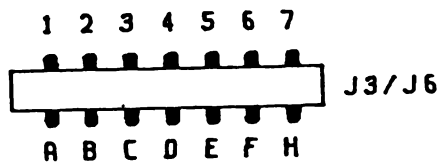
Arrangement of pins plug J3

PIN	PIN	SIGNAL
1	A	GND
2	B	+5V
3	C	CASS MOTOR
4	D	CASS READ *1
5	E	CASS WRITE
6	F	CASS SWITCH*1

CASSETTE 2

Arrangement of pins plug J6

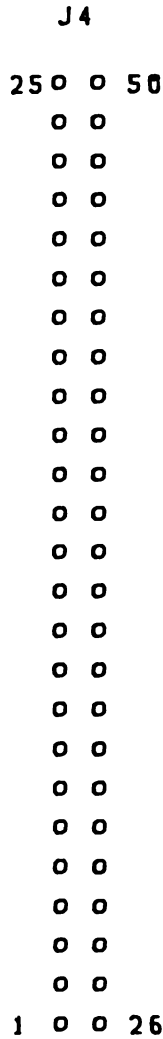
PIN	PIN	SIGNAL
1	A	GND
2	B	+5V
3	C	CASS MOTOR
4	D	CASS READ *2
5	E	CASS WRITE
6	F	CASS SWITCH*2



MEMORY - EXPANSION

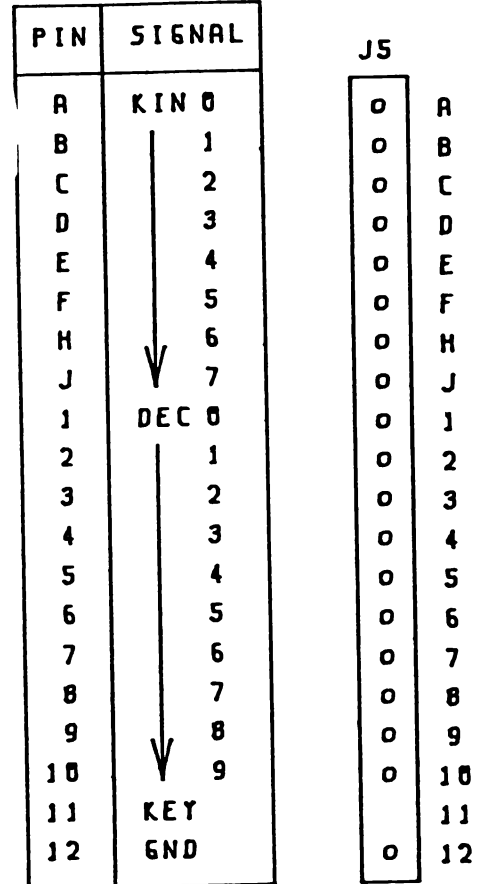
Arrangement of pins plug J4

PIN	SIGNAL	PIN	SIGNAL
1	GND	26	GND
2	BDO	27	↓
3	1	28	↓
4	2	29	↓
5	3	30	↓
6	4	31	↓
7	5	32	↓
8	6	33	↓
9	7	34	↓
10	N.C.	35	↓
11	N.C.	36	↓
12	RAMSEL9	37	↓
13	RAMSELA	38	↓
14	RAMON	39	↓
15	SELEXP	40	↓
16	N.C.	41	↓
17	CS9	42	↓
18	CSA	43	↓
19	CSE	44	↓
20	NOROM	45	↓
21	PENSTRB.	46	↓
22	RESET	47	↓
23	READY	48	↓
24	NMT	49	↓
25	GND	50	GND



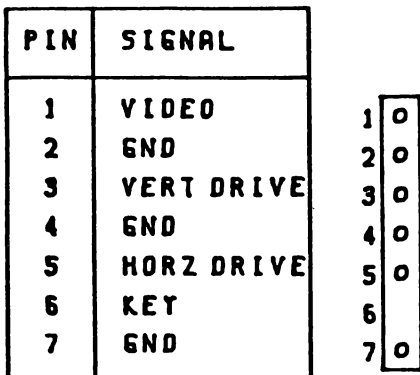
KEYBOARD

Arrangement of pins plug J5



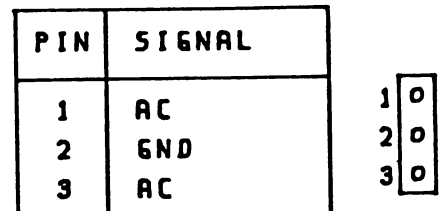
VIDEO

Arrangement of pins plug J7



POWER IN

Arrangement of pins plug J8



MEMORY - EXPANSION

Arrangement of pins plug **J9**

PIN	SIGNAL	PIN	SIGNAL	
1	GND	26	GND	J9
2	BA 0	27	↓	25 0 0 50
3	1	28		0 0
4	2	29		0 0
5	3	30		0 0
6	4	31		0 0
7	5	32		0 0
8	6	33		0 0
9	7	34		0 0
10	8	35		0 0
11	9	36		0 0
12	10	37		0 0
13	11	38		0 0
14	12	39		0 0
15	13	40		0 0
16	14	41		0 0
17	15	42		0 0
18	SYNC	43		0 0
19	TRQ	44		0 0
20	DIAG	45		0 0
21	PHI2	46		0 0
22	B R/W	47		0 0
23	B R/W	48		0 0
24	N.C.	49		0 0
25	GND	50		GND

POWER EXP. AC

Arrangement of pins plug **J10**

PIN	SIGNAL	
1	AC 8V	1 0
2	AC 8V	2 0
3	GND	3 0
4	KEY	4 0
5	GND	5 0
6	AC 8V	6 0
7	AC 8V	7 0

POWER EXP. DC

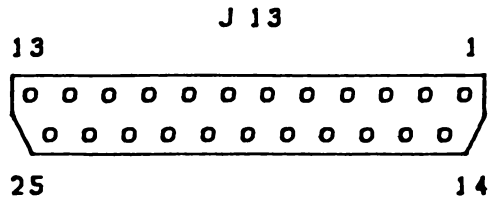
Arrangement of pins plug **J11**

PIN	SIGNAL	
1	+9V UNREG.	1 0
2	KEY	2 0
3	KEY	3 0
4	+9V UNREG.	4 0
5	GND	5 0
6	+9V UNREG.	6 0
7	GND	7 0

KEYBOARD

Arrangement of pins plug J13

PIN	SIGNAL
1	KIN 0
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	DEC 0
10	1
11	2
12	3
13	4
14	5
15	6
16	7
17	8
18	9
19	N. C.
20	GND
21	N. C.
22	N. C.
23	N. C.
24	N. C.
25	N. C.



RESET

Arrangement of pins plug J14

PIN	SIGNAL
1	RESET
2	GND



B.2 Connectors, modifications as compared with 8032

C B M 8 2 9 6

Changes in arrangement of pins

J 4 Memory expansion

Pin	Signal	
	old	new
12	<u>SEL 4</u>	<u>RAM SEL 9</u>
13	<u>SEL 5</u>	<u>RAM SEL A</u>
14	<u>SEL 6</u>	<u>RAM ON</u>
15	<u>SEL 7</u>	<u>SEL EXP</u>
16	<u>SEL 8</u>	N.C.
...		
19	<u>SEL B</u>	<u>CS E</u>

J 8 Power supply

8032		8296	
Pin	Signal	Pin	Signal
1	8V AC	1	AC
2	+9V UNREG	2	GND
3	GND	3	AC
4	8V AC		
5	8V AC		
6	GND		
7	KEY		
8	14V AC		
9	14V AC		

J 10 Power supply for hardware expansion

Pin	Signal	
	old	new
1	-9V UNREG	AC 1
2	-9V UNREG	AC 1
3	KEY	GND
4	+16V UNREG	KEY
5	+16V UNREG	GND
6	GND	AC 2
7	GND	AC 2

C Software addendum

TABLE OF CONTENTS

64k System Disk.....	19
Add-on-load.....	19
BASIC 2.0.....	20
BASIC 4.0/40.....	20
BASIC 4.0/80.....	21
Add-on-mon.....	21
Expanded BASIC.....	24

LIST OF ILLUSTRATIONS

f i g u r e

C-1 Modifications to BASIC 2.0.....	20
C-2 Modifications to BASIC 4.0/40....	21

64k SYSTEM DISK

By using the 64k Expansion Memory Bank you have given your system a soft load capability and also expanded program work space. We have provided a disk with several programs to demonstrate these functions.

```
ADD-ON-LOAD
ADD-ON-LOAD.SRC
ADD-ON-MON
ADD-ON-MON.SRC
BASIC2.0
BASIC4.0/40
BASIC4.0/80
EXPANDED-BASIC
EXP-BASIC.SRC
EXPANDED-DEMO
STTEST
TEST1
TEST2
TEST3
TEST4
```

For your convenience we have recorded these programs in 8050 format on one diskette surface and in 4040 format on the flip side. Use the corresponding side for your system and make a back-up copy before use.

ADD-ON-LOAD

This program is a special loader which loads one of three special versions of BASIC, which we have provided on the disk, and jumps to the proper entry to execute them. These versions of BASIC reprogram the CRT controller of your 8032 to become a 40-column display. With this environment, all old 40-column programs in BASIC 2.0 and 4.0 will run on your 8032.

To load and execute a version of BASIC, type the following sequence of commands:

```
DLOAD"ADD-ON-LOAD" <return>
RUN <return>
```

Your CRT should display the following message:

SYSTEM NAME?

The cursor will be blinking for you to enter one of the following system names followed by <return>:

BASIC2.0
BASIC4.0/40
BASIC4.0/80

In a few moments your 8032 will print the logon message from the program you selected.

BASIC 2.0

This is the 40-column BASIC found in all the 3016 and 3032 business keyboard machines. It was produced by dumping the ROMs and adding the following patch to re-initialize the CRT controller:

```
D71A EA EA
D745 A2 04
D747 BD 5E D7
D74A BC 59 D7
D74D 8C 80 E8
D750 80 81 E8
D753 CA
D754 10 F1
D756 4C DE E1
D759 01 02 04 07 09
D75E 14 20 28 21 07
FC05 20 45 D7
```

figure C-1

BASIC 4.0/40

This is the 40-column BASIC found in all the 4016 and 4032 business keyboard machines. It was produced by dumping the ROMs and adding the following patch to re-initialize the CRT controller:

FD1B	20	50	FD		
FD50	A2	04			
FD5F	BD	76	FD		
FD62	BC	71	FD		
FD65	8C	80	ED		
FD68	8D	81	ED		
FD6B	CA				
FD6C	10	F1			
FD6E	4C	00	E0		
FD71	01	02	04	07	09
FD76	14	20	28	21	07

figure C-2

BASIC 4.0/80

This version of BASIC is that supplied in the 8032 60hz versions. If you have a 50hz machine you probably do not need this version anyway because it is already in ROM.

Expansion Memory Monitor Program

The expansion memory monitor source program is provided on the diskette supplied with this manual. A listing of the program is provided as an addition to this manual. To run the monitor program:

1. Load the program into the lower 32K of RAM by typing:

```
dload "add-on-mon" <return>
```
2. Type:

```
run <return>
```

The following paragraphs describe the commands used in the expansion monitor program. The commands allow the programmer to examine or alter expansion memory, examine or alter 6502 registers, execute programs, load or save disk files and load the expansion control register.

Format: : Address [list of bytes]

Purpose: Alter bytes in memory.

Remarks: This command is automatically printed onto the CRT display preceding the address and data after execution of the display memory (M) command. To alter memory, use the screen editor to change the displayed bytes and press the <RETURN> key. The bytes are altered in the addresses specified by the expansion control register.

Format: ;PC,IRQ,SR,AC,SR,YR,SP

Purpose: Loads list into 6502 registers.

Remarks: The list of data following this command is loaded into the 6502 hardware registers when a G command is given. This command is automatically printed on the screen preceding the current list of data when an R command is executed. The list can be edited and re-entered in the same manner as the alter memory command. See the R command for contents of the list.

R

Format: R

Purpose: Displays the 6502 registers.

Remarks: This command displays the contents of a list loaded into the 6502 hardware registers when execution is transferred from the monitor. A sample display follows:

R <RETURN>

PC IRQ SR AC XR YR SP

;0400 E262 01 00 FF FF FE

The abbreviations are defined as follows:

PC = program counter
IRQ = interrupt vector
SR = status register
AC = accumulator
XR = X-index register
YR = Y-index register
SP = stack pointer

M

Format: M Address [Address]

Purpose: Displays bytes from memory

Remarks: Bytes are displayed from the addresses specified by the expansion control register. If one address is specified, 8 bytes are read and displayed on the screen, starting at that address. For more than one address, a range of bytes is displayed, but always the next even multiple of 8 bytes from the first. The STOP key stops the listing.

M 0400 <RETURN>

: 0400 00 00 00 AA AA AA AA

G

Format: G [Address]

Purpose: Start execution.

Remarks: If an address is not specified, the monitor dispatches to the location contained in the PC of the register display. If an address is specified, execution dispatches to that address. If a BRK (00) has been inserted in the user code, execution will return to the monitor and a register display given with the message "BREAK". On dispatch, the registers are loaded with the contents of the register display.

L

Format: L "name", Device

Purpose: Load memory

Remarks: Device number must be 4 or greater for CBM disks. The starting load address is implicit in the program load file. The STOP key will break a program LOAD.

S

Format: S "name", Device, Address, Address

Purpose: Save memory

Remarks: A file name must be specified in quotation marks followed by a device number, a starting save address and an ending save address. The STOP key will break a memory save.

@

Format: @ [disk command]

Purpose: Displays disk status buffer

Remarks: The command immediately followed by <RETURN> will read the disk status buffer and print its contents on the screen. The device is set at 8 and the command channel is 15.

@ <RETURN>

@@,ok,@@,@@

If a string follows the @, then that string is transmitted to device 8 channel 15 as a command.

@ INITIALIZE @

name

Format: name

Purpose: Load and execute file from disk

Remarks: When a command cannot be matched to the list of known commands, an attempt is successful, the monitor jumps to execute it.

*

Format: * byte

Purpose: Load expansion memory control register (mapper).

Remarks: Puts the byte value following the command into the expansion control register. When a byte value is not specified, a zero is stored. This value restores the machine to ROM operation.

EXPANDED-BASIC

This program is a pseud-cache memory system for use with the 8032 and the 64k add-on memory board only. It is loaded into high memory, (\$7600-\$7BE0):

DLOAD"EXPANDED-BASIC"

It is activated by:

SYS 30720

This routine works nearly the same as the DOS-support program

provided with some Commodore disk systems. It moves the top of memory for BASIC down to \$7800 and resets all the variables. Its commands are implemented by using an escape character sequence detected by a tap into the CHRGET routine in BASIC's zero page.

The EXPANDED-BASIC program is completely protected from normal BASIC programs--however it is vulnerable to POKE and machine code programs.

Because the 8032 ROM code is not modified, it is not possible to add space directly to a BASIC program. The EXPANDED-BASIC program allows a user to store or "cache" programs and data in the expansion RAM for ultra high-speed access in overlaying programs.

The following commands are available:

RECALL

Format: !R,0:"filename",s(u)(p),[device]

Purpose: Cache a file from disk.

Remarks: The file can be USR, SEQ, or PRG. The device number defaults to 8. The ST variable is set in the same manner as a file OPEN for read.

LOAD

Format: !L,"program name"

Purpose: Move data from add-on to BASIC text area.

Remarks: This command clears the current program and loads the named program. Will not work from a program--only direct.

OVERLAY

Format: !O,"program name"

Purpose: Move data from add-on to BASIC text area.

Remarks: This command overlays the current program. Variables and data are preserved.

EXECUTE

Format: !E,"program name"

Purpose: Move data from add-on to BASIC text area.

Remarks: This command clears the variables and data and loads in a new program. It is callable by a program.

QUIT

Format: !Q

Purpose: Turn off the expanded Basic functions.

Remarks: None

The programs and files are placed in the expansion memory in a contiguous manner. If a wrap-around occurs the data will not fit in the first 32k bank, it will continue over into the second bank. There are a maximum of 10 file control blocks so there is a maximum of 10 files that can be stored.

The expanded-BASIC program maintains the file name, start location, end location, and data pointer. The system does not know the difference between file types, so all operations can be performed on all types of files.

The system has the following restrictions: When a command is used in program mode it must be preceded by a colon. This is necessary to insure proper operation within an IF THEN.



This was brought to you

from the archives of

<http://retro-commadore.eu>