

Inhalt

| | |
|---|----|
| Überblick | 1 |
| RAM-Verteilung | 3 |
| Slave-Zugriffe | 8 |
| Interne Operationen des Mikrorechners | 13 |
| Interface zum VDC | 20 |
| Prinzipien des CONTROL MODE | 23 |
| P10-Adressierung | 32 |
| Anschluß am Systembus | 36 |
| Adressierung des RAM | 42 |
| Steuerung der Slave-Zugriffe | 46 |
| Steuerung der CPU-Zugriffe | 52 |
| Ablaufsteuerung | 53 |
| Operationen im CONTROL MODE | 67 |
| Diagnostische Sonderzustände | 75 |
| Meßpunkte | 77 |
| Unbenutzte Bezeichnungen | 78 |
| Steckverbinder | 79 |
| Übersicht VDC-Interface | 80 |
| Übersicht über die PROM-Schaltgleichungen | 81 |

Überblick

Der VIDEO ADAPTER (VDAD) ist ein Mikrorechner, der am Systembus als Slave angeschlossen sowie mit dem VIDEO DISPLAY CONTROLLER (VDC) verbunden ist.

Ausstattungsmerkmale:

- 64 k RAM, über den Bus zugänglich und vom Mikrorechner bewerkbar.

In der bevorzugten Betriebsweise wird die Information zweifach redundant gespeichert (mit einem Paritätsbit pro Byte), um die Korrektur von "soft errors" zu ermöglichen.

Damit beträgt die nutzbare Speicherkapazität 32 k Bytes.

- 4 MHz CPU, 2 PIOs, 16 k ROM, 4 k RAM (Scratchpad) als elementare Mikrorechnerkonfiguration (ROM und Scratchpad sind nicht über den Systembus zugänglich)

- 2
- Anschluß für Diagnoseadapter
 - Spezielles Interface zum VDC
 - Zusätzliche Steuermittel und Vorkehrungen im Mikrorechner für schnelle Datenübertragung (CONTROL MODE)

Funktionen am Systembus:

- Les- und Schreibzugriffe, mit und ohne Codewandlung
- Interrupts
- maskierte Leszugriffe ("Suchzugriffe")
- selektives Rücksetzen

RAM- Aufteilung

Gen. Bild 1 ist der 64 k Byte - RAM
in zwei Bildbereiche zu 16 k Byte und
zwei Korrekturbereiche aufgeteilt.

Jeder Bildbereich kann bis zu 8 Bilder
zu je 2000 Bytes (25 Zeilen zu 80 Zeichen)
aufnehmen. Unter Steuerung der CPU
wird jeweils ein Bildinhalt zwecks Anzeige
zyklisch zum VDC transportiert. Bildinhalte
müssen stets am Anfang von einem der
jeweils 8 2048-Byte-Blöcke eines Bildbereichs
gespeichert werden.

Die Bezeichnungen "links" und "rechts" für die
Bildbereiche beziehen sich auf die physische
Position des jeweils zugeordneten Monitors.

Bildbereiche können neben Bildinhalten
auch beliebige anderweitige Information
aufnehmen.

Die höchstwertigen Positionen des zweiten
Bildbereichs sind für allgemeine Steuerzwecke
reserviert.

Zu jedem Bildbereich gibt es einen gleich
großen Korrekturbereich. Dieser nimmt die

zweiten Bytes auf, wenn die Betriebsart der redundanten Speicherung aktiviert ist.

Es sind aber auch unabhängige Zugriffe zu den Korrekturbereichen möglich.

Die allgemeine Struktur der RAM-Adresse ist in Bild 2 veranschaulicht.

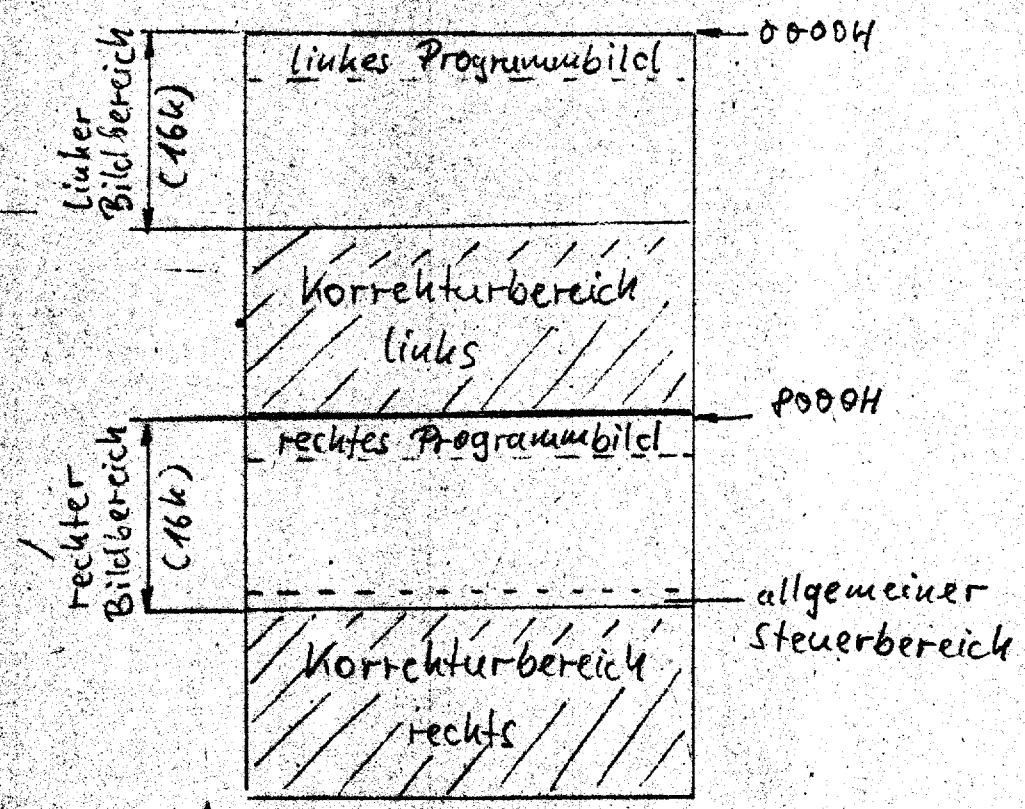
Bit 15 dient zur Unterscheidung zwischen dem "linken" und dem "rechten" Bildbereich.

Ist Bit 14 = 0, so wird zum jeweiligen Bildbereich zugegriffen. Ist die Betriebsart der redundanten Speicherung aktiviert, so wird bei jedem Schreibzugriff automatisch eine Kopie des Bytes in den jeweiligen Korrekturbereich eingeschrieben.

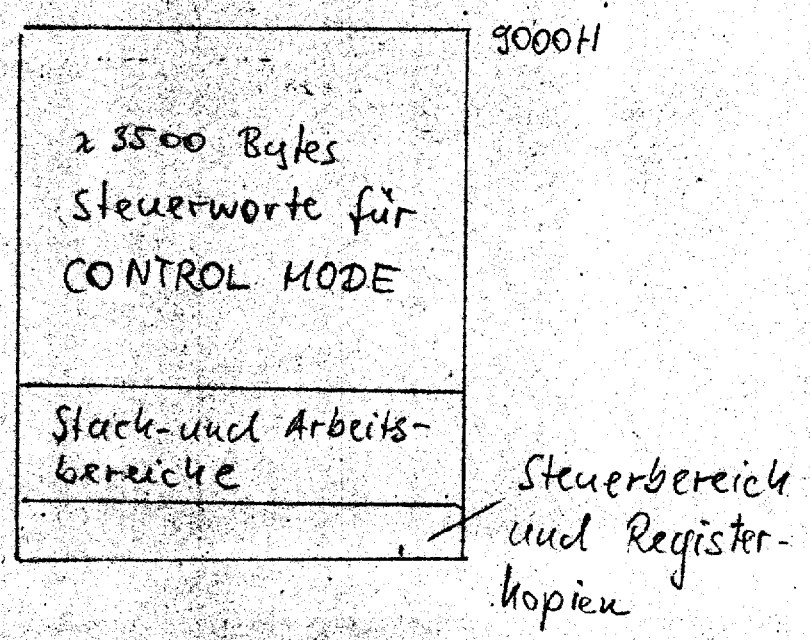
Bei jedem Lesezugriff findet eine Paritätsprüfung statt, und im Fehlerfall wird automatisch das jeweilige Byte aus dem Korrekturbereich geholt. Damit wird der Inhalt des Bildbereichs korrigiert.

Ist Bit 14 = 1, so wird zum jeweiligen Korrekturbereich zugegriffen (ohne doppeltes Speichern bzw. Fehlerkorrektur).

Aufteilung des RAM Bild 1



Aufteilung des Scratchpad Bild 3



Die Aufteilung des Scratchpad ist
aus Bild 3 ersichtlich. Die ersten ca. 3500
Bytes sind mit Steuerworten für die
Beschleunigung zeitkritischer Abläufe belegt
(CONTROL MODE). Daran schließen sich
Stack- und Arbeitsbereiche, Steuerbereiche
sowie Positionen für Registerkopien an.

Allgemeine Struktur der RAM-Adresse

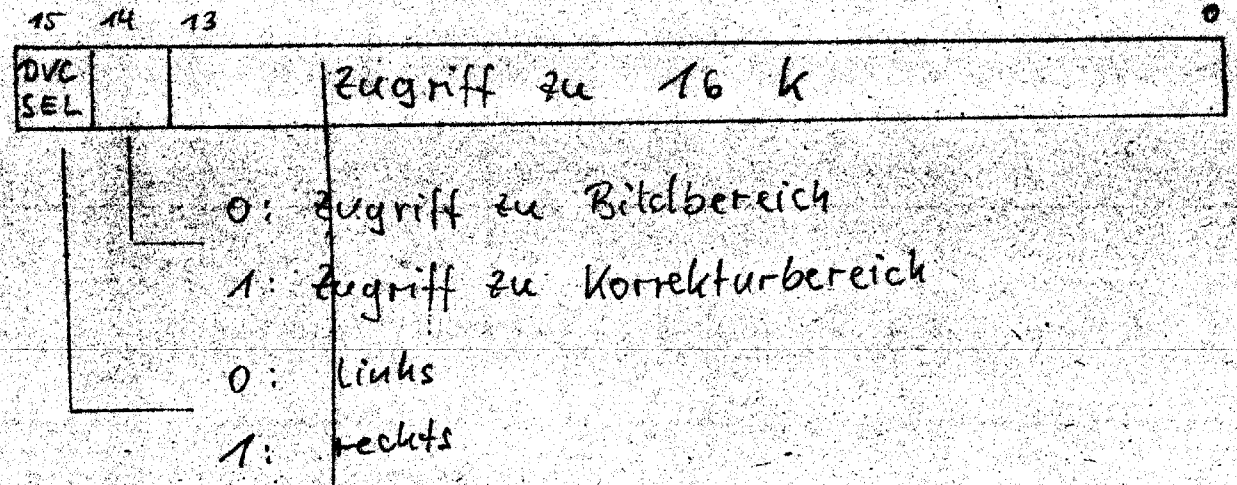


Bild 2

Slave-Zugriffe

Der VDAD belegt drei aufeinanderfolgende Slave-Adresspositionen. Bei der ersten Position müssen die Adressbits 17, 16 die Belegung 0, 0 haben.

Der VDAD erscheint am Systembus als RAM von 64 k Bytes mit einigen speziellen Zugriffsmöglichkeiten.

Sonderfunktionen:

- Empfangen von Interrupts
- Selektives Rücksetzen ist wirksam
- RELEASE kann signalisiert werden (wenn ein Interrupt nicht angenommen werden kann)

Wesentlich ist, daß bei den RAM-Zugriffen und Interrupts automatisch Codewandlungen ausgeführt werden.

Die darzustellende Bildinformation ist im RAM grundsätzlich im STM-Code gespeichert.

Bei Zugriffen mit einer Slave-Adresse der Form $x y 0 0$ (x, y = Adressenbits 15, 16)

erfolgt keine Codewandlung. Damit sind direkte Zugriffe zu den 64 k Bytes des RAM möglich (Interpretation der 16 niedrigwertigen Adressenbits gem. Bild 2). Interrupts mit einer solchen Slave-Adresse wirken auf PIO 2 Port A, wobei der Interruptvektor direkt (ohne Codewandlung) übertragen wird.

Bild 4 gibt einen Überblick über die Adressierung bei Slave-Zugriffen.

Es sind Zugriffe mit zwei weiteren Codes möglich:

1. EBCDIC - (IBM-) Code
2. ISO - Code

Weiterhin können Suchzugriffe ausgeführt werden. Dabei wird keine lesen ein charakteristisches Datenbyte auf den Bus gelegt, wenn der RAM-Inhalt dem jeweiligen Suchkriterium entspricht.

Tafel 1 gibt einen Überblick über Suchkriterien, charakteristische Datenbytes und Struktur der jeweiligen Adresse.

Bei IBM- bzw. ISO-Zugriffen sind nur die Bildbereiche zugänglich, nicht die Korrekturbereiche. Umfasst die nutzbare Adresse weniger als 14 Bit, so ist damit stets der Anfang des jeweiligen Bildbereichs adressierbar (die ersten 4 k Bytes bei IBM-Zugriffen bzw. die ersten 8 k Bytes bei ISO-Zugriffen).

Interrupts wirken auf jeweils einen der Ports von PIO 1, wobei der Interruptvektor stets im IBM-Code erwartet wird.

Ist bei IBM-Zugriffen Bit 14 = 0 und Bit 12 = 1, so werden Kleinbuchstaben automatisch in Großbuchstaben gewandelt.

Bei Interrupts wird diese Wandlung durch Setzen von PIO 2 Port B Bit 3 wirksam.

Adressierung bei Slave-Zugriffen

| | 19 | 18 | 17 | 16 | 15 | |
|----------|----|----|----|----|----|--|
| SLV ADRS | 0 | 0 | | | | Zugriff zu 64 k im SCM-Code |
| SLV ADRS | 0 | 1 | | | | Zugriff zu 32 k im IBM-Code (EBCDIC) ¹⁾ |
| SLV ADRS | 1 | 0 | | | | Zugriff zu 32 k im ISO-Code |

Wirkung bei Interrupts

- PIO 2 Port A (SCM)
- PIO 1 Port A (IBM)
- PIO 1 Port B (IBM)

1) Struktur der 16-Bit-Adresse bei IBM-Zugriffen

| | 15 | 14 | 13 | 12 | 11 | |
|---------|----|------|-------|------|----|------------------|
| DVC SEL | 0 | SEEK | LC/UC | CONV | | Zugriff zu 4 k** |
| DVC SEL | 1 | | | | | Zugriff zu 16 k* |

*: Bildbereich gem. DVC SEL

** : Anfang des Bildbereichs gem. DVC SEL

2) Struktur der 16-Bit-Adresse bei ISO-Zugriffen

| | 15 | 14 | 13 | 12 | | |
|---------|----|----|----|----|--|--------------------------------|
| DVC SEL | 0 | | | | | Zugriff zu 16 k* |
| DVC SEL | 1 | | | | | Zugriff zu 8 k** (Suchzugriff) |

0: UNPROTECTED
1: MODIFIED

| Suchkriterium | A-Reg. (CPIR) | Zugriff |
|----------------------------------|---------------|--|
| NUL-Zeichen | OOH | direkte Speicherzugriffe zu beliebigen Codes |
| nicht-NUL-Zeichen | POH | IBM-Zugriffe mit ADRS 14 = 0, ADRS 13 = 1 |
| FCC | ADH | IBM-Speicherzugriffe |
| ungeschütztes Feld (15/5 = 0) | AOH | ISO-Zugriffe mit ADRS 14 = 1, ADRS 13 = 0 |
| modifiziertes Feld (15/6 = 1) | COH | ISO-Zugriffe mit ADRS 14 = 1, ADRS 13 = 1 |

Übersicht "Suchzugriffe"

Tafel 1

Interne Operationen des Mikrorechners

Der eigentliche Mikrorechner umfasst die CPU, zwei PIOs, 16 k Bytes ROM, 4 k Bytes RAM (im folgenden als Scratchpad bezeichnet), ein Segmentregister sowie einen Diagnoseadapter - Anschluss. Weiterhin sind Zugriffe zum 64 k-RAM des VIDAD möglich.

Die eigentliche Mikrorechner - Anwendung arbeitet weitgehend unabhängig vom 64 k-RAM und seinen spezifischen Schaltmitteln (Vermittlung, Zyklussteuerung usw.). Völlig unabhängig sind Zugriffe zum Scratchpad, zu den Registern sowie ROM-Zugriffe, die keine M1-Zugriffe sind.

M1-Zugriffe zum ROM sind dann unabhängig, wenn FORCE EVEN PARITY (über PIO 2 Port B) aktiviert ist. In diesem Fall werden jedoch keine Refresh-Zugriffe zum 64 k-RAM ausgeführt (der Datenverlust kann dann lediglich durch explizite Zugriffe der CPU zum RAM gewährleistet werden).

Normalerweise werden die Refresh-Zyklen der CPU zum Auffrischen des 64k-RAM benutzt (FORCE EVEN PARITY muß dazu ausgeschaltet sein). Dies geschieht dadurch, daß für jeden M1-Zugriff über die Vermittlungsschaltung ein CPU-Zyklus angefordert wird. Wird dieser Zyklus ausgeführt, so werden Slave-Anforderungen solange unterdrückt, bis das RFSH-Signal der CPU wieder inaktiv wird (d. h. bis zu dessen LO-HI-Flanke). In diesem Intervall wird die Refreshadresse der CPU zum RAM geführt. Das MREQ-Signal des Refreshzyklus wird direkt als RAS-Impuls für den RAM verwendet. Das Schieberegister der Adreßsteuerung ist dabei nicht in Betrieb.

Bei M1-Zugriffen zum ROM wird der angeforderte CPU-Zyklus nicht bis zum Ende ausgeführt (von PA... END) sondern sofort nach dem Aktivieren wieder beendet. (Der CPU-Zyklus veranlaßt lediglich das Blockieren von Slave-Zugriffen für

das nachfolgende Refresh-Intervall.)
Wartezeiten treten bei solchen M1-Zu-
griffen nur auf, wenn gleichzeitig
Slave-Zugriffe angefordert werden.

Da mit die CPU zu den 64 k des
RAM zugreifen kann, werden die
höchstwertigen Positionen der RAM-
Adresse von einem ladbaren Segment-
register geliefert. Bei Zugriffen zu
diesem Register wird eine Kopie im
Scratchpad mitgeführt, so dass z. B.
Abfrage- und Modifikationsbefehle auf
den Registerinhalt angewendet werden
können.

Die CPU kann kurze oder lange Segmen-
te im RAM adressieren.

Kurze Segmente sind 2 k Bytes lang.
Die CPU liefert die niederen 11 Adressen-
bits (Bereich A000H ... AFFFH).

Die höheren fünf
Bits der 16-Bit-RAM-Adresse kommen
aus dem Segmentregister (Positionen 4-8).

Lange Segmente sind 16 k Bytes lang.
Die CPU liefert die niederen 14

bits (Bereich 000H...FFFFH). Die beiden
höchstwertigen Adressbits werden von
den Positionen 4, 3 des Segmentregisters
bereitgestellt.

Details der Adressierung sind in den
Bildern 5-7 dargestellt.

Adressierung bei CPU-Zugriffen

| | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|--|
| 15 | 14 | 13 | 12 | 11 | 10 | 3 | 2 | 1 | 0 | |
| 0 | 0 | | | | | | | | | Zugriff zu 16 k ROM |
| 0 | 1 | | | | | | | | | Zugriff zu 16 k ROM, CONTROL MODE |
| 1 | 0 | 0 | 0 | | | | | | | Registerzugriffe |
| | | | | | | | | | | DUAL CLR SEG- ADL FREQ MENT |
| 1 | 0 | 0 | 1 | | | | | | | Zugriff zu 4 k Scratchpad-RAM |
| 1 | 0 | 1 | X | X | | | | | | RAM-Zugriff (kurzes Segment) |
| 1 | 1 | | | | | | | | | RAM-Zugriff (langes Segment) |
| 0 | | | | | | | | | | DUAL CLR SEL MENT |
| | | | | | | | | | | RAM-Zugriff bei DATA TO RAM (CONTROL MODE) |

- 0 ... 3FFFFH
- 4000H ... 7FFFFH
- 8000H ... 8FFFFH
- 9000H ... 9FFFFH
- A000H ... BFFFFH
- C000H ... FFFFFH

Bild 5

Bildung der RAM-Adresse bei CPU-Zugriffen

| | | | | | | |
|-------------|---|---|---|---|---|---|
| SEGMENT REG | | | | | RAM-Zugriff (kurzes Segment = 2 k) | * |
| 4 | 3 | 2 | 1 | 0 | | |
| SEG. REG | | | | | RAM-Zugriff (langes Segment = 16 k) | |
| 4 | 3 | | | | | |
| DVC SEL | 0 | | | | RAM-Zugriff bei DATA TO RAM im CONTROL MODE | |

Auswahl des anzuzeigenden Bildes

* auch bei RAM TO BUFFER im CONTROL MODE (SEGMENT REG 3 bewirkt Geräteauswahl)

Bild 6

Register - Zugriffe

| empfohlene Adresse | Wirkung |
|--------------------|---|
| PFFGH | Lesen ¹⁾ bzw. Schreiben des Segmentregisters |
| PFFAH | Schreiben veranlaßt Löschen ²⁾ des Feldattributregisters |
| PFFEH | lesen DIAG IN bzw. Schreiben DIAG OUT |

1) aus Kopie im Scratchpad

2) Datenbyte kann beliebig sein

Belegung des Segmentregisters

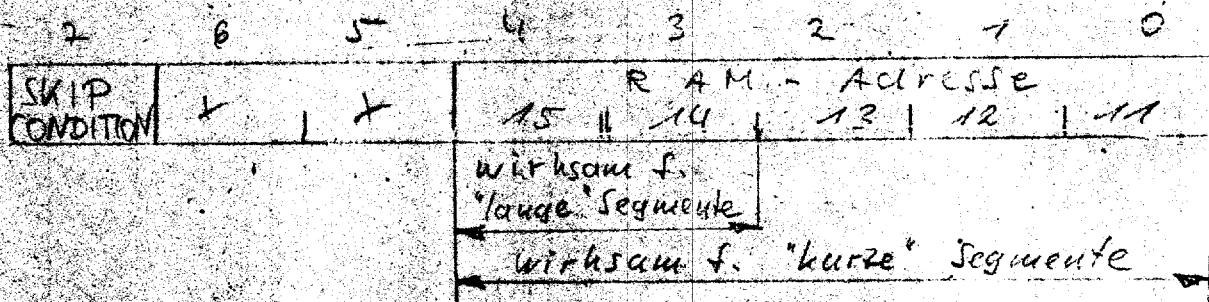


Bild 7

Interface zum VDC

Das Interface umfaßt einen Schreibdatenweg und einen LeseDatenweg aus jeweils 18 Leitungen. Jeder Datenweg besteht aus Flachbandkabel. Beide Datenwege sind physikalisch so ausgelegt, daß eine Kurzschlußprüfung am VIDAD möglich ist, indem beide Steckverbinder über ein Flachbandkabel miteinander verbunden werden.

Der Schreibdatenweg umfaßt 17 Informationsleitungen und eine Strobeleitung, der LeseDatenweg 17 Informationsleitungen und eine Besetzleitung.

Die 17. Informationsleitung des LeseDatenweges führt üblicherweise das Bildrücklaufsignal (FLYBACK) und ist an Port B der PIO 2 angeschlossen. Zugriffsmöglichkeiten bestehen zu

- dem Pufferpeicher
- dem Zeichengenerator
- dem Sync-RAM
- dem Adressenzähler
- verschiedenen diagnostischen Schaltmitteln.

Jeder Zugriff wird durch einen Strobeeimpuls (über den Schreibdatenweg) ausgelöst. Ist der VDC momentan besetzt, d. h. nicht in der Lage, die Information zu akzeptieren bzw. zu liefern, so signalisiert er auf die Vorderflanke von VDC STROBE über den Lesedatenweg VDC BUSY. Dies kann einen Interrupt im Mikrorechner auslösen. VDC BUSY schaltet nach der Rückflanke von STROBE ab, wenn ein Zugriff ausgeführt wird, der den SYNC RAM nicht betrifft. Dieser Besetztzustand kann nur bei Zugriffen zum SYNC-RAM auftreten.

Die höchstwertige Informationsleitung des Schreibdatenweges kennzeichnet einen Schreibzugriff zum Adressenzähler des VDC (die weiteren 16 Bits werden in den Adressenzähler geladen). Aussonstern wird die Art des Zugriffes (Lesen bzw. Schreiben) sowie die gewünschte Einrichtung des VDC durch die höchstwertigen sechs der Schreib- Informationsleitungen bestimmt (Bit 17 = 0, Bits 16 ... 11).

Die Belegung der weiteren 16 Les-
Informationsleitungen wird bei
jedem VDC-Zugriff in ein Leseregister
übernommen. Die CPU kann
zu diesem Leseregister sowohl in
Zusammenhang mit Lesezugriffen zum
VDC als auch unabhängig davon
zugreifen.

Prinzipien des CONTROL MODE

In diesem Modus veranlassen Befehle, die von der CPU ausgeführt werden, zusätzliche Steuerungswirkungen, insbesondere zur Steuerung des VDC-Interfaces (zur Illustration s. die Bilder 8-10).

Der CONTROL MODE ist bei Befehlszugriffen im Bereich von 4000H...7FFFH wirksam. Die eigentlichen Befehle werden dabei aus dem ROM gelesen. Parallel dazu wird der Scratchpad adressiert. Dessen Inhalt wird in ein CONTROL-Register übernommen und veranlaßt spezifische Steuerungswirkungen.

Die wesentlichen Steuerungsfunktionen sind:

a.) Abholen VDC-Keeregister.

Der betreffende Befehl ist ein LD HL, um; LD BC, um; LD DE, um.

Dem Operationscode entspricht die Scratchpad-Belegung 00H (keine Steuerungswirkung). Dem folgenden Byte (u) korrespondiert 04H im Scratchpad (GATE LO) und dem weiteren Byte (u) korrespondiert 08H (GATE HI).

Diese Steuerbits veranlassen, daß aus dem
des ROM der jeweilige Teil des
VDC-Register auf dem Datenbus
der CPU gelegt wird.

Damit steht nach Ausführung des
Befehls die VDC-Teilinformation in
dem betreffenden Doppelregister zur
Verfügung.

B.) Datentransport CPU → RAM (DATA TO
RAM). Die RAM-Adresse muß in
einem Doppelregister stehen, das
Datenbyte in A. Der betreffende
Befehl ist dann z. B. LD (HL), A.

Korrespondierend mit dessen Operations-
code steht im Scratchpad 40H bzw.
48H. Während des M1-Zyklus gibt
es keine Steuerwirkungen. Hingegen
wird während des folgenden Daten-
transports (Schreibzugriff mit Adresse
gem. (HL)) die Adresse zum RAM
durchgesteuert und ein Schreibzyklus
angeführt. Enthält der Scratchpad 48H,
so wird ein NMI ausgelöst, wenn die
Adressenbits 7-10 sämtlich = 1 sind
(NMI IF WRAP). Damit läßt sich eine

programmtechnische Reaktion für den Fall voraussetzen, daß die Adresse in einem Bildbereich den Wert 1920 (780H) erreicht hat.

Hinweis: Diese Sonderfunktion wirkt nur dann korrekt, wenn das Programm in dem sie benutzt wird, die aktuelle Adresse stets im Adreßlöt (von Null an).

c.) Leszugriffe zum VDC.

Die Funktion entspricht a.). Dem Befehl korrespondieren im Scratchpad die Bytes 80H... BBH. Zusätzlich werden bei den Operandenzugriffen Leseoperationen im VDC ausgeführt.

d.) Schreibzugriffe zu Einrichtungen des VDC.

Für die Daten ist dabei stets ein Doppelregister in der CPU vorzusehen. Zum Schreiben ist z.B. ein Befehl LD (HL), A zu verwenden. Das Datenbyte ist dabei bedeutungslos, es wird die 16-bit-Adresse als Information zum VDC übertragen. Die Funktion wird im Scratchpad durch 80H... BBH

codiert. Bei Schreibzugriffen vom
ADRS CTR wird die höchstwertige
Leitung des VDC-Schreibdatensusses
aktiviert. Auf den verbleibenden 16
Leitungen wird die vollständige
16-bit-Adresse der CPU zum
ADRS CTR übertragen.

Ansonsten werden die niederen 14 bit
der CPU-Adresse sowie spezifische
Steuerinformationen (Schreib/Lese-Aus-
wahl, DESTINATION CODE u. ADRS CONTROL
aus dem CONTROL-Register) übertragen.

e.) Datentransport RAM \rightarrow BUFFER (in
VDC).

Der Datentransport erfolgt während
der M1-Zyklen der betreffenden
Befehle. Dabei wird mit den
niederen 14 bit der Befehlsadresse
parallel ein Lesezugriff zum RAM
geführt. Die höchstwertigen fünf
Bits der RAM-Adresse werden
vom Sequenzregister geliefert (Zugriff
zu einem kurzen Segment). Zusätzlich
wird die Feldattribut-Information
(bitierend, immer aus) des betreffenden
Zeichens im...

register bereitgestellt. Dieses Register wird bei derartigen Zugriffen stets dann geladen, wenn aus dem RAM ein Feldsteuerzeichen (Bit 7 = 1) gelesen wird. Bei Übertragung der Feldsteuerzeichen selbst wird das Attribut "BLANK" eingefügt.

Die betreffenden Befehle sind im allgemeinen NO-OPs. Korrespondierend dazu sind im Scratchpad Bytes der Form CxH bereitzustellen.

Ist $x = 2H$, so wird die Information in den VDC-Puffer übernommen, und der Adressenzähler wird um Eins erhöht. Das UNDERLINE-Attribut wird dabei mit "0" geladen.

(Hinweis: UNDERLINE ist durch Zugriffe gemäß d.) mit ADPS CONTROL = 0, 1 zu aktivieren).

Ist $x = 3H$, so wird die Adresse weitergezählt, aber kein Schreiben ausgeführt.

Mit Bit 2 = 1 im Scratchpad läßt sich erreichen, daß der Datenbus der CPU ständig auf 0 gehalten wird,

so daß der folgende Befehl als
NOP gelassen wird. Man braucht
somit nicht eine lineare Folge
von z.B. 1920 NOPs im ROM
vorzusehen, sondern lediglich an
der eisten und an der 1920. Position.
Der dazwischen liegende ROM-Bereich
kann beliebig benutzt werden (für
Funktionen, die den CONTROL MODE
nicht benötigen). Es ist nur der
Scratchpad entsprechend zu laden
(im Beispiel 1919 mal mit C6H, an
der letzten Position mit C2H).

Mit Bit 3 = 1 im Scratchpad wird
das ~~Anterkennen~~ des Schreibens im VDC
(durch Bit 0 = 1, SUIP WRITE) nur dann
wirksam, wenn Bit 7 des Segmentregisters
= 1 ist (CONDITIONED SUIP)

Hinweis: ADRS CONTROL = 1, 1 ist
vorrangweise für die aktuelle
Cursorposition vorgesehen. Dazu
ist im geladenen Scratchpad in
dem entsprechenden Byte lediglich
Bit 0 einzuschalten. Die Cursorposition
wird dabei übersprungen. Sie kann
durch einen spezifischen Zugriff gemäß
1) nacheinander geladen werden.

Scratchpad-Belegung im CONTROL MODE, Bild 8

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---|---------------|------------------|---|------------------|--------------|--------------|---|-----|
| 0 | 0 | 0 | 0 | GATE HI | GATE LO | 0 | 0 | a.) |
| 0 | DATA TO RAM | 0 | 0 | NMI IF WRAP | 0 | 0 | 0 | b.) |
| 1 | 0 | DESTINATION CODE | | GATE HI | GATE LO | 0 | 0 | c.) |
| 1 | 0 | DESTINATION CODE | | 0 | 0 | ADRS CONTROL | | d.) |
| 1 | RAM TO BUFFER | 0 | 0 | CONDITIONED SKIP | INSERT NO-OP | ADRS CONTROL | | e.) |

DESTINATION CODE

| | | |
|---|---|----------|
| 0 | 0 | BUFFER |
| 0 | 1 | CHAR GEN |
| 1 | 0 | SYNC RAM |
| 1 | 1 | ADRS CTR |

ADRS CONTROL

| | | |
|---|---|--------------------------|
| 0 | 0 | do nothing |
| 0 | 1 | "UNDERLINE" ATTRIBUTE* |
| 1 | 0 | COUNT ADRS |
| 1 | 1 | COUNT ADRS & SKIP WRITE* |

* wirksam nur bei d.) & e.) (Pufferzugriffe)

Formate

- Zugriff zu Lesegerät
- Datentransp. (PU → ...)
- Lesen VDC
- Schreiben CPU-Adresse nach VDC
- Datentransp. RAM → BUFFER

Schreibinformation für VDC

Bild 9

| | | | | | | | | | | | | |
|----|------------------|------------------|----|----|--------------|---|---|---|------------------------------|------------------------|---|--|
| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 0 | |
| 1 | ADRS REG CONTENT | | | | | | | | | | | |
| 0 | WRITE TO VDC | 0 | 0 | 0 | 1 | FIELD ATTRIBUTES BLANK INTENS BLINK INVERT | | | | 7-bit - CHARACTER CODE | | |
| 0 | WRITE TO VDC | 0 | 1 | | | | | | 9-bit - CHARACTER SHAPE INFO | | | |
| 0 | WRITE TO VDC | 1 | 0 | | | x | | | 10-bit - SYNC RAM CONTENT | | | |
| | | DESTINATION CODE | | | ADRS CONTROL | | | | | | | |

DESTINATION CODE

- 0 0 BUFFER
- 0 1 CHAR GEN
- 1 0 SYNC RAM
- 1 1 ADRS CTR

ADRS CONTROL

- 0 0 DO NOTHING
- 0 1 DO NOTHING; UNDERLINE FIELD ATTRIBUTE (IF WRITE TO BUFFER)
- 1 0 INCREMENT ADRS
- 1 1 INCREMENT ADRS & SKIP WRITE (IF WRITE TO BUFFER)

Lese-Information vom VDC

| | | | | | | | | | | | | | | | | | | | | | |
|-----------|----------------|----|----|----|----|----|---|---|--------|----------------------|-------------------|-----|-----|-------|-------|----------|-----------------------------|-----|-----|-----|-----|
| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | ADDR CONTROL DECTIMING CODE | | | | |
| ✓FLY-BACK | DIAG SLOW-DOWN | x | x | x | x | x | x | x | INVERT | 7-bit-CHARACTER CODE | | | | | | | 00 | | | | |
| | | | | | | | | | | 7 | CHAR GEN CONTENT | | | | | | 0 | 01 | | | |
| | | | | | | | | | | 7 | SYNC RAM CONTENT | | | | | | 0 | 10 | | | |
| | | | | | | | | | | VIDEO ENABLED | BUFFER UNDER/WRIT | | | INTEN | BLINK | CHAR GEN | SYNC RAM | | 11 | | |
| | | | | | | | | | | (10) | | (9) | (8) | (7) | (6) | (5) | (4) | (3) | (2) | (1) | (0) |

SCH-7CC

| | | | | | | | |
|---|----------|-----------|---|-------|--------|--------|-------|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | MODIFIED | PROTECTED | x | BLANK | INTENS | INVERT | BLINK |

IBM-Attributzeichen

| | | | | | | | |
|---|---|-----------|---------|----------------|----------|-----|---|
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| x | x | PROTECTED | NUMERIC | DISP LK/INTENS | RESERVED | MDT | |

- 0 03 DISPLAY
- 0 0 INTENS
- 1 1 BLANK

Bild 10