

Er weiß deshalb jederzeit, wann welche Transaktionen stattfinden können. Auch kann er die Steuerung des Bus durch den Prozessor unterbinden. Das ist immer dann nötig, wenn VIC auf Videoram, Charactergenerator oder Farbram zugreifen muß. Diese zyklischen Zugriffe müssen sein, damit kontinuierlich ein Bild auf dem Schirm erscheint. Um den Prozessor nicht bei seiner Arbeit zu stören, nutzt er geschickt die Taktstückchen aus, in denen der Prozessor den Bus nicht benötigt. Die Vereinbarung ist einfach folgende, daß immer wenn $\theta_{2=1}$ ist, der Prozessor den Bus belegt, und bei $\theta_{2=0}$ der VIC. Um den Bus auch physikalisch freizumachen, legt VIC den Pin AEC auf 0, worauf der Prozessor den Bus seinerseits in den hochohmigen Zustand (Tri-State) bringt. Auf diese Weise stört keiner den anderen.

3.3 Die Schnittstelle zum Ram

Zur Verdeutlichung des hier Gesagten dient das Blockschema auf der nächsten Seite.

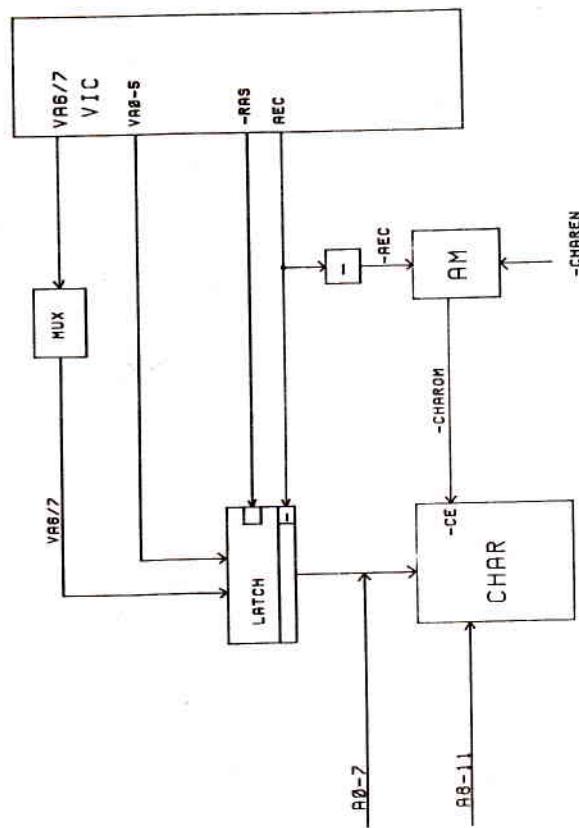
Der VIC kann von Hause aus nur 16K Ram adressieren. Deshalb muß er, und das ist wichtig, beim Verschieben von Videoram oder Zeichen generator, die beiden fehlenden Adressbits von außerhalb zu Hilfe nehmen. Diese Bits werden von PAG-1 der CIA 2 zur Verfügung gestellt. Verändert man diese Bits, so verschiebt sich der Videoram gleich in 16K-Schritten. Diese Bits stellen also die Adressbits 14 und 15 des Videorams dar. Unter der Voraussetzung, daß man über genügend freien Speicher verfügt, kann man mit diesen Bits mehrere Bildschirmsichten umschalten. Beachten Sie hierbei bitte, daß diese Bits L-aktiv sind, d.h. um die unterste 16K-Büte zu selektieren, müssen beide Bits in der CIA 2 auf 1 gesetzt sein. Für die obere Seite müssen beide 0 sein. Im Hi-Ram Modus könnten zum Beispiel die eine Bildschirmsseite dargestellt werden, während eine andere unsichtbar bereits aufbereitet wird.

In feineren Schritten von 1K kann man den Videoram mit Hilfe des VIC-Registers 24 verschieben. Hier bedeuten die Bits 4-7 die Ramadressen 10-13.

3.4 Die Charactergenerator-Schnittstelle

Das Bild auf der übernächsten Seite illustriert die folgende Erklärung.

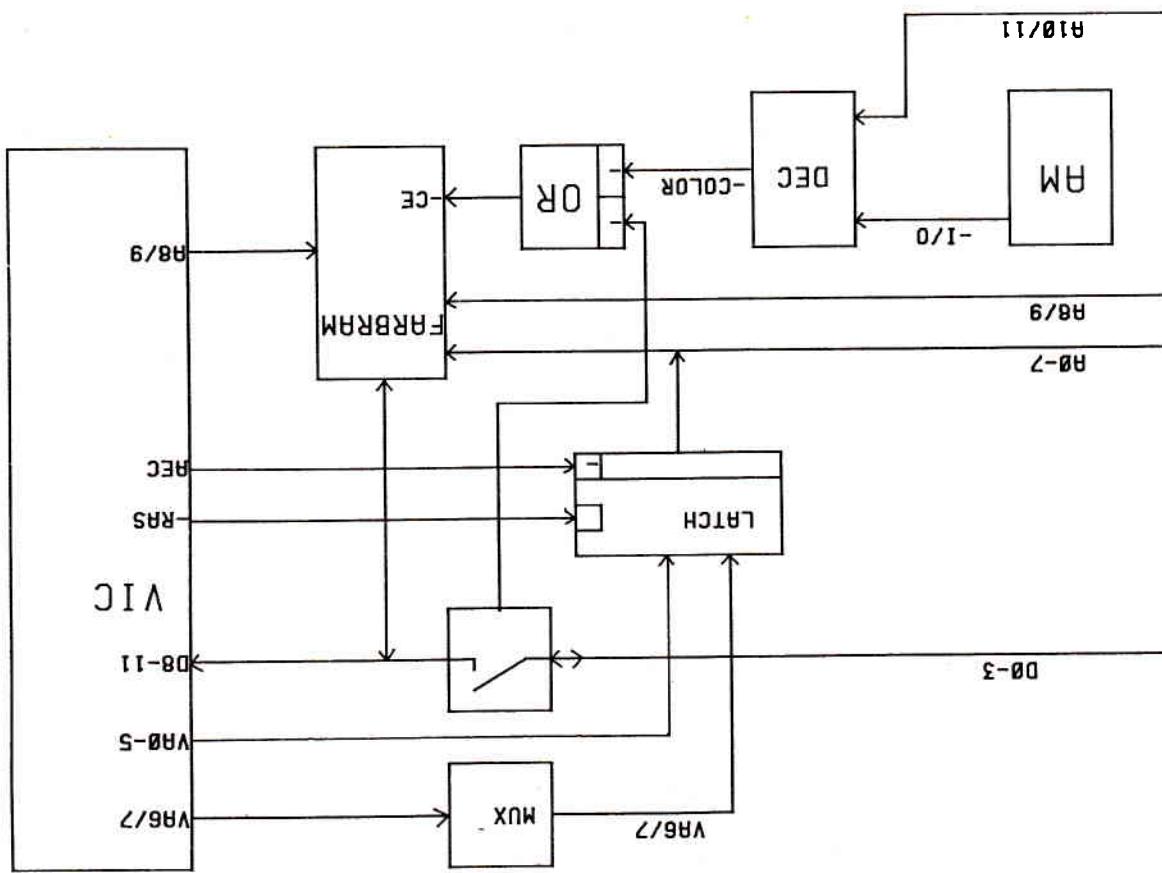
Ahnlich wie im vorigen Abschnitt ist der Ablauf auch hier. Die CIA-Bits verschieben gleichzeitig mit dem Videoram auch den Charactergenerator in ebensolchen Schritten.



Mit Hilfe der Bits 1-3 im REG 24 des VIC kann er in 2K-Schritten verschoben werden. Sie stellen die Adressbits 11-13 dar.
 Der im CBM 64 eingebaute Character-Rom besitzt dank des Addressmanagers einen Sonderstatus. Dieses wird vom VIC nur angesprochen, wenn sich seine relative Adresse zwischen \$1000-\$1FFF, bzw. \$9000-\$9FFF bewegt.
 Tatsächlich liegt der Generator im Prozessordressraum bei \$D0000.
 Da der Character-Rom rein physisch nicht im Zugriffsbereich aller Videoadressen liegt, müssen die fehlenden Bits in einem besonderen Puffer zwischengespeichert werden. Es sind diese die Adressbits 0-7. Die Charactergeneratorkerbasis im REG 24 spielt noch eine weitere Rolle. Sie legt nämlich nicht nur die Startadresse des Generators fest, sondern Bit 3 bestimmt im Einzelbit-Modus auch die Lage des Bildschirmspeichers. Hierbei bleiben die Bits 1 und 2 unberücksichtigt.

3.5 Die Schnittstelle zum Farb-Ram

Das Farbram ist, im Gegensatz zu Videoram und Zeichengenerator, nicht verschieblich. Die Adressierung kann, ebenso wie beim Character-Rom, von zwei Seiten erfolgen, nämlich vom Prozessorbus und vom VIC-Bus. Auch hier fehlen dem VIC zur Adressierung wieder die Bits 0-7. Sie werden auf die gleiche Weise gewonnen wie beim Character-Rom.
 Zum VIC hin hat der Farbram eine eigene Datenschmittstelle. Sie ist mittels eines Analogschalters vom Datenbus des Prozessors trennbar. Die Angelegenheit funktioniert denkbar einfach:
 Bei AEC=0 (also VIC steuert den Bus) liegt der Schalter in Richtung VIC. Bei AEC=1 in der anderen Richtung. Darüber hinaus wird der Schreib/Lesezugriff seitens des Prozessors auf den Farbram vom Address-Manager gesteuert. Das Blockschema auf der nächsten Seite soll diese Umstände noch anschaulicher machen.



1.3 Der Prozessor 6510 und seine Besonderheiten

Der CBM 64 hat als Prozessor die MPU 6510 (MPU=Micro Processing Unit). Dies ist ein 8-Bit-Prozessor, der im CBM 64 mit einer Taktfrequenz von ca. 980Khz betrieben wird.

Die MPU 6510 ist ein neues IC aus der Familie 65xx. Darin, daß der Chip über sechs I/O-Leistungen verfügt.

Das hat in Minimalkonfigurationen gegenüber dem 6502, der über keinerlei I/O-Leistungen verfügt, den Vorteil, daß man ohne zusätzliche Peripherie-ICs über einzeln programmierbare Leistungen verfügt, sei es zur Verbindung mit der Außenwelt (z.B. Tastatur), sei es für interne Steuerzwecke. Im CBM 64 unterstützen diese Leistungen zum einen den Kassettenbetrieb, zum anderen die Speicherverwaltung.

Der Befehlssatz ist der gleiche wie beim 6502. Interessant für den Programmierer dürfte noch sein, daß die I/O-Leistungen die Adressen 0 und 1 belegen, nämlich 0 als Datenrichtungsregister und 1 als Datenregister.

Der Stack kann sich im Bereich \$0100 bis \$01FF bewegen. Unter Stack versteht man einen Stackspeicher, auf dem, z.B. bei Sprung in Unterprogramme, die Prozessorregister abgelegt werden können, um sie beim Rücksprung ins Hauptprogramm wieder hervorzuholen und mit den ursprünglichen Inhalten weiterzuarbeiten.

Zu der nun folgenden Pin-Bellegung noch eine Anmerkung: Es ist uns nicht gelungen, ein dem tatsächlichen entsprechendes Datenblatt aufzutreiben. Wir haben zwar mehrere, doch auf jedem ist das Pinout anders dargestellt. Deshalb sind wir den empirischen Weg gegangen und haben kontrolliert, wo die einzelnen Leistungen dann tatsächlich hingehen. Hier nun das Ergebnis:

- 1 DIN Systemtakt Eingang! im CBM 64 ca. 980KHz.
 - 2 RDY Ready @-Prozessor hält beim nächsten Lesezyklus an, bis RDY=1. Von dieser Möglichkeit macht man z.B. beim Betrieb langsamer Speicher Gebrauch.
 - 3 -IRQ Interrupt Request! @-Prozessor holt sich die nächste Befehlsadresse von \$FFE und macht dort weiter. Dieser Umstand tritt nur ein, wenn der Interrupt erlaubt war (Bit 2 im Flag-Register=0).
 - 4 -NMI Non-Maskable Interrupt @-Prozessor holt sich die nächste Befehlsadresse von \$FFFA und macht dort weiter.
 - 5 AEC Address Enable Control! @-Prozessor bringt Daten- und Steuerbus in den hochohmigen Zustand (Tri-State). Der Bus kann nun von anderen Einheiten betrieben werden, z.B. ein zweiter Prozessor.
 - 6 VCC Betriebsspannung +5V
 - 7 -20 A0-A13 Adressbus
 - 21 GND
 - 22-23 A14-A15! Adressbus
 - 24-29 P5-P0! I/O-Pins
 - 30-37 D7-D0! Datenbus
 - 38 R-/W! @-Schreibzugriff, 1=Lesezugriff
- All die Zugriffe finden nur während $\overline{O2}=1$ statt.

- 39 #2OUT! Systemtakt Ausgang zur Versorgung anderer Bausteine.
- 40 RES Reset! @-Prozessor geht in den Ruhezustand. Beim Übergang von 0 nach 1 holt sich der Prozessor eine Adresse von \$FFFFC und beginnt dort mit dem Programm.

1.4 Speicherbellegungssätze

Aufgrund der mehrfachen Bereichsüberlappungen und der damit notwendigen Verwaltung des Adressraumes ergeben sich etliche mögliche Belagerungen.

Dreh- und Angepunkt des gesamten Komplexes ist ein IC, welches im Folgenden kurz Adress-Manager (AM) genannt wird. Beim AM handelt es sich um ein IC von der Sorte FPLA (Field Programmable Logic Array).

Derartige Bausteine enthalten eine Vielzahl potentieller logischer Verknüpfungen, die vom Anwender in der gewünschten Kombination programmiert werden können.

Im Falle des CBM 64 ist dies ein IC mit 16 Eingangsleistungen und 8 Ausgangsleistungen.

Durch die Programmierung wurde jeder Eingangskombination (eine aus 65536 möglichen) eine Ausgangskombination (eine aus 256 möglichen) zugeordnet.

Da dieses IC ein wesentlicher Bestandteil des CBM 64 ist, sollen hier die Ein- und Ausgangssignale erklärt werden mit Zweck (Ausgänge) und Herkunft (Eingänge).

Zunächst die Eingänge:

- CAS Signal vom VIC zur Steuerung der Ramadressen.
- DRAM Signal vom Prozessorport Bit 0
- HIRAM wie oben, jedoch Bit 1
- CHAREN wie oben, jedoch Bit 2. Soll das Auslesen des ZeichenGenerators ermöglichen. Versuchen Sie das nie in Basic, da die vom Timer angestoßene Interruptroutine statt des erwarteten Timers dort nun den Charactergenerator und Timer in der CIA belegen. Dieselben Adressen und deshalb falsch verweigt, was zu einem 'Absturz' des Betriebssystems führt.
- VA14 kommt von CIA 2 Port A Bit 0! stellt einen Teil der Basisadresse von Zeichengenerator und Videoram dar.
- A13-A15 Adressbussignale dienen hier zur Dekodierung des I/O-Bereiches.
- BA Signal vom VIC; wirkt auf die RDY-Leitung des Prozessors
- AEC @-Prozessor steuert den Bus, 1=VIC steuert den Bus. Das Signal ist durch Invertierung von AEC des VIC gewonnen.
- R/-W Steuersignal des Prozessors
- EXRD Eingangssignal vom Expansion-Slot
- GAME wie oben
- VA12-13 Adressbus des VIC

Nun die Ausgänge:

- CASRAM 0-Ram übernimmt das höherwertige Byte der Speicheradresse
- BASIC 0-Basic-Rom ist selektiert
- INTERNAL 0-Betriebssystem-Rom ist selektiert
- CHARROM 0-Auswahl des Charactergenerators
- GR/-W 0-Schreibzugriff auf das Farbrom
- I/O 0-I/O-Dekoder selektiert
- ROML Signal zum Expansion-Slot
- ROMH wie oben

Auf der nächsten Seite finden Sie eine (fast) vollständige Liste der Zuordnungen von Eingangskombinationen zu Ausgängen.

Zu dem 'fast' eine kleine Anmerkung:
 Es wäre uns sicher ein Leichtes gewesen, dieses Buch mit allen der zu den 65536 möglichen Eingangskombinationen gehörigen Ausgangskombinationen zu füllen. Bei ICs dieser Gattung gibt es jedoch eine große Menge redundanter Eingangsbits, d.h. Bits, bei denen sich in einer bestimmten Konstellation nichts am Ausgang ändert. Solche Bits sind in der Tabelle durch ein X gekennzeichnet. Solche Un diese Bits aber herauszufinden, haben wir ein Programm entwickelt, welches über vier Wochen ohne Unterbrechung lief und immer noch nicht ganz fertig war. Deshalb mögen am Ende der Tabelle vielleicht einige Kombinationen fehlen. Das sind ein paar der Ausgangskombination \$FE und alle der Kombination \$FF. Die letztere ist ohnehin nicht von großer Bedeutung, da ja, wie Sie sehen, alle Ausgänge 0-aktiv sein müssen, um etwas zu bewirken.

Wenn Sie sich nicht durch diese Tabelle durchkämpfen wollen, finden Sie auf den nächsten Seiten bildlich in geraffter Form die wichtigsten Auswirkungen des AM auf die Speicherzuordnung.

*** PAL CBM 64 ***

- C -	- H H L	- V A I O	- E	- V V G X R	- C -	- C K - C
- -	- -	- -	- -	- A A A A A A A A	- R R - R A R A S	- G H E B A
- -	- -	- -	- -	- M M M M M M M M	- O O I / R N S R	- M M / - O A I A
- -	- -	- -	- -	- N M M M M M M M M	- H L Q W M L C M	-
- -	- -	- -	- -	- 3 2 E M W C A 2 3 4 5 4 N M M S	-	-