

20000 Byte mehr

20 KByte adressierbarer Speicher werden uns beim normalen Basic vom Computer vorenthalten. Dieser Speicherplatz ist jedoch nicht tot, er kann mit ein paar Tricks für alle möglichen Zwecke nutzbar gemacht werden.

Der C 64 besitzt zwei Arten von Speicher-Chips. Zum einen den RAM-Speicher (Random Access Memory), zum anderen den ROM-Speicher (Read Only Memory). Auf das RAM kann beliebig zugegriffen werden. Das heißt, man kann diesen Speicher beliebig beschreiben und lesen. Dieser Speicher dient zur Aufnahme der eigenen Programme. ROM heißt soviel wie Nur-Lese-Speicher. Er enthält ein festes Programm oder Daten (Zeichensatz etc.) und kann nur gelesen werden. Der ROM-Speicher behält seine Informationen auch nach dem Ausschalten des Computers. Beim Einschalten des Computers oder nach einem Reset verzweigt der Prozessor ins ROM (Betriebssystem) und initialisiert das gesamte System.

Der C 64 besitzt 64 KByte RAM und 20 KByte ROM. Im ROM sind Basic, Betriebssystem und Standardzeichensatz gespeichert (Bild 1). Für die Ein-/Ausgabevorrichtungen benötigt der C 64 nochmal 4 KByte Speicherkapazität. Ein 8-Bit-Computer mit 16-Bit-Adreß-Bus kann aber normalerweise nur 64 KByte (2 hoch 16 Bit) adressieren. Wo liegt aber der Unterschied zwischen C 64 und den anderen 8-Bit-Computern der 65xx-Reihe?

Das Geheimnis liegt im 6510-Prozessor. Der Chip verfügt im Gegensatz zum 6502 über ein Prozessorregister. Über diesen Port wird gesteuert, ob RAM, ROM oder Ein-/Ausgabe in bestimmten Speicherbereichen aktiv sind. Hierfür benötigt das Register (Speicherstelle 1) allerdings nur 3 Bit. Weitere 3 Bit steuern den Kassettenport. Die restlichen 2 Bit sind unbesetzt und werden einfach überlesen. Das Datenrichtungsregister für diesen Port liegt im Speicherplatz 0.

Die einzelnen Bits im Kontrollregister des 6510 sind wie folgt definiert:

Name	BIT	Funktion
LORAM	0	Basic-ROM/RAM (\$A000 bis \$BFFF)
HIRAM	1	Betriebssystem-RAM (\$E000 bis \$FFFF)
CHAREN	2	I/O-Zeichen-ROM (\$D000 bis \$DFFF)
	3	Schreibleitung
	4	Schalter
	5	Motorsteuerung (Bit 3 bis 5 betreffen nur Kassettenport)
	6	(unbesetzt)
	7	(unbesetzt)

Zu jedem Daten-Port gehört auch ein Datenrichtungsregister. Dieses liegt für den Prozessor-Port in Speicherstelle 1. Der Normalwert für das Datenrichtungsregister lautet 47 oder Binär 00101111, wobei 0 für Eingabe und 1 für Ausgabe

steht. In Standardmodus stehen LORAM, HIRAM, CHAREN, Schreibleitung und Motorsteuerung auf »Ausgabe«. Nur der Kassettenschalter steht auf »Eingabe«.

Die Steuerbits des Prozessorregisters erfüllen im allgemeinen folgende Funktion:

BIT 0 (LORAM):

Dieses Bit steuert die Belegung des Speicherbereichs von \$A000 bis \$BFFF. Ist Bit 0 gesetzt, so belegt der 8-KByte-Basic-ROM den Adreßbereich. Bringt man diese Bit auf Low, so verschwindet das Basic-ROM und wird durch 8 KByte RAM-Speicher ersetzt.

BIT 1 (HIRAM):

Ist dieses Bit gesetzt, hat der Prozessor im Adreßbereich \$E000 bis \$FFFF Zugriff auf das Betriebssystem (Kernel-ROM) des Computers. Bei Low-Pegel wird das Kernel durch ebenfalls 8 KByte ersetzt.

Bit 2 (CHAREN):

Dieses Bit ist dafür verantwortlich, ob sich das 4K-Byte-Zeichengenerator-ROM oder das I/O-RAM im Prozessor-Adreßbereich von \$D000 bis \$DFFF befindet. Wenn Bit 2 gesetzt ist, dann hat der Prozessor Zugriff auf folgende Ein-/Ausgabe-Register:

\$D000-\$D3FF : VIC
 \$D400-\$D7FF : SID
 \$D800-\$DBFF : Farb-RAM
 \$DC00-\$DCFF : CIA1
 \$DD00-\$DDFF : CIA2
 \$DE00-\$DEFF : freier I/O-Port #1
 \$DF00-\$DFFF : freier I/O-Port #2

Ist Bit 2 gelöscht, kann der Prozessor aus dem Zeichen-ROM lesen. In diesem Fall sind die Ein-/Ausgabe-Register gesperrt. Im Normalzustand ist dieses Bit gesetzt. Dieses Bit muß nur gelöscht werden, wenn man den Zeichensatz ins RAM kopieren will. Dabei darf aber das Programm nicht unterbrochen werden, da ja die CIA1 (Tastatur) abgeschaltet ist. Um Bit 2 von Basic aus auf Low zu setzen, muß vorher der Interrupt gesperrt werden, da die Interruptroutine statt des Timers im CIA den Charaktergenerator vorfindet und dadurch abstürzt.

Das Zeichen-ROM kann wie folgt eingeschaltet werden:

```
10 POKE 56334,PEEK(56334) AND 254: REM INTERRUPT
   AUS
20 POKE 1,PEEK(1) AND 251: REM ZEICHEN-ROM EIN
30 Programm.....(Kopie des Zeichensatzes etc.)
1000 POKE 5634,PEEK(56334) OR 1: REM INTERRUPT EIN
```

Ausgeschaltet wird die Sache mit folgendem Programm:

```
10 POKE 56334,PEEK(56334) AND 254: REM INTERRUPT
   AUS
20 POKE 1,PEEK(1) OR 4 : REM I/O-RAM EIN
30 POKE 5634,PEEK(56334) OR 1: REM INTERRUPT EIN
```

Sie werden sich nun aber fragen, wie der Computer trotz Tastaturabfrage Zeichen am Bildschirm ausgeben kann. Dies geschieht mit Hilfe des sogenannten ROM-IMAGE. Weil der VIC-II-Chip gleichzeitig nur auf 16 KByte zugreifen kann, muß der Zeichensatz immer in der jeweiligen Bank des VIC-II-Chips liegen. Im Standardmodus greift der VIC auf Bank 0 zu (\$0000 bis \$3FFF). Das System ist so entwickelt, daß VIC-II davon ausgeht, daß sich der Zeichensatz im »zugriffsberechtigten« Speicherbereich befindet. Arbeitet der VIC-II in Bank 0, nimmt er an, der Zeichensatz befindet sich ab \$1000, obwohl dieser in Wirklichkeit in \$D000 liegt. Der Speicherbereich \$1000 bis \$1FFF ist aber trotzdem für den Anwender als RAM-Bereich vorhanden. Gleiches gilt für die drei anderen möglichen Banks des VIC-II:

Bank	VIC-II-Chip	ROM-IMAGE
0	\$0000-\$3FFF	\$1000-\$1FFF
1	\$4000-\$7FFF	\$5000-\$5FFF
2	\$8000-\$BFFF	\$9000-\$9FFF
3	\$C000-\$FFFF	\$D000-\$DFFF

Wenn man nun Bit 1 und 2 löscht, müßte man theoretisch auf 64 KByte RAM Zugriff haben. Das probieren wir gleich mal aus (vorher eventuelle Programme sichern!):

POKE 1,PEEK(1) AND 252

Jetzt stehen die 64 KByte RAM voll zur Verfügung, nur ist der Computer jetzt »scheintot«. Er reagiert weder auf Tastendruck noch auf gut zureden. Hier hilft nur noch ein Reset oder Ausschalten.

Was passiert bei PEEK und POKE?

Befindet sich der C 64 im Standardmodus (wie nach dem Einschalten), dann sind Basic-ROM und Kernel-ROM aktiv. Wie wir wissen, kann man den ROM-Speicher nicht beschreiben (Read Only Memory). Was passiert nun, wenn man versucht, in diesen Speicherbereich einen Wert zu POKE? Der Befehl wird nicht einfach ignoriert, wie man das so vom VC 20 oder den anderen CBM-Computern kennt. Der Prozessor schiebt den Wert einfach in das darunterliegende, »versteckte« RAM. Versucht man mit PEEK (X) das Byte wieder zu lesen, stellt man fest, daß der Prozessor auf das ROM zugreift.

Auf diese Weise läßt sich einfach das Basic- und Kernel-ROM ins RAM kopieren. Setzt man anschließend Bit 1 des Prozessorports auf Low, verändert sich scheinbar nichts. Der Prozessor greift jetzt auf das RAM zu und kann beliebig verändert werden.

Im folgenden werden zwei Programme gezeigt, um das Betriebssystem ins RAM zu kopieren und umzuschalten:

```
10 FOR I=40960 TO 49151: POKE I,PEEK(I) : NEXT
20 FOR I=57344 TO 65535: POKE I,PEEK(I) : NEXT
30 POKE 1,53 : REM FÜR C 64
30 POKE 1,5 : REM FÜR SX 64
```

oder wie in Listing 1:

```
>, C000 A9 C0      LDA #$C0
>, C002 85 FB      STA $FB
>, C004 A9 A0      LDA #$A0
>, C006 85 FD      STA $FD
>, C008 20 1E C0   JSR $C01E
>, C00B A9 00      LDA #$00
>, C00D 85 FB      STA $FB
>, C00F A9 E0      LDA #$E0
>, C011 85 FD      STA $FD
>, C013 20 1E C0   JSR $C01E
>, C016 A9 35      LDA #$35
>, C018 85 01      STA $01
>, C01A 8D D6 FD   STA $FDD6
>, C01D 60         RTS
>, C01E A0 00      LDY #$00
>, C020 84 FC      STY $FC
>, C022 B1 FC      LDA ($FC),Y
>, C024 91 FC      STA ($FC),Y
>, C026 CB         INY
>, C027 C0 00      CPY #$00
>, C029 D0 F7      BNE $C022
>, C02B E6 FD      INC $FD
>, C02D A5 FD      LDA $FD
>, C02F C5 FB      CMP $FB
>, C031 D0 EB      BNE $C01E
>, C033 60         RTS
```

Listing 1. So kopiert man das ROM ins RAM

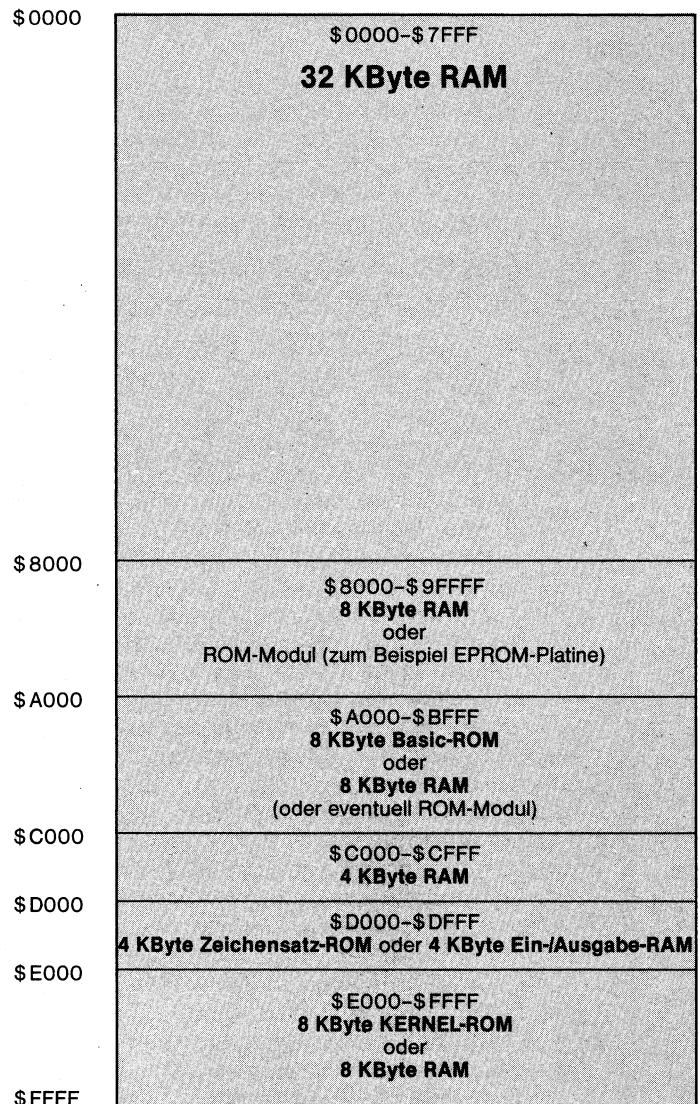


Bild 1. Die Speicheraufteilung im C 64

Hier stellen sich dem Anwender ungeahnte Möglichkeiten, auf die hier nicht im einzelnen eingegangen werden soll.

20 KByte frei - was tun damit?

Es gibt auch noch andere Möglichkeiten, den RAM-Bereich unter dem ROM zu nutzen.

Es ist durchaus möglich, zum Beispiel ein HiRes-Grafikbild in diesem RAM-Bereich zu erstellen. Durch Umschalten des VIC-II-Chips steht dem Anwender so sofort ein neues Grafikbild zur Verfügung, während das frühere Bild wieder editiert werden kann. Eine praktische Anwendung hierfür wäre zum Beispiel eine Dia-Show.

Die RAM-Bank unter dem ROM kann auch als sogenanntes RAM-Floppy verwendet werden. Es könnte also durchaus ein 20 KByte langes Maschinen-Programm ohne Schwierigkeiten im »versteckten« RAM untergebracht werden.

Es gibt noch sehr viele Möglichkeiten, den »versteckten« Speicher zu nutzen. Probieren Sie es doch selbst mal. Sie werden erstaunt sein, wo und was ihr C 64 alles speichern kann.

(C.Q. Spitzner/og)