C64 Kernal v2.1

Killing the Killer Signature

Introduction

There is a number of games (such as Gyruss, PAC-MAN, Poltergeist etc.), which make use of a fake cartridge signature CBM80 (Figure 1) as a copy protection. This redirects the reset and the NMI vector, so it is not possible to create a memory dump etc. to copy the game.

It is very annoying that the signature stays in DRAM even after switching off the C64. As a result, the computer does not boot properly. It might be required to switch off the C64 for a couple of minutes, before the signature disappears.

It was reported, that Poltergeist stayed intact after a short power cycle and started after switching on again. The C64 was haunted!

The Kernal v2.1 is a slightly modified original Commodore Kernal, which destroys this signature at the very beginning of the Reset routine, so the C64 boots properly, even after a game, that makes use of this copy protection mechanism.

Figure 1: Memory dump after playing Gyruss with a fake CBM80 cartridge signature

The Cartridge Signature

Most cartridges for the expansion port are mounted by the Kernal using a simple mechanism:

The EXROM signal of the expansion port is held low by the cartridge. As a result, the content of this cartridge appears in the $8000 - $9FFF address space in memory. Further on, there is a cartridge signature, which also contains the start address of the cartridge software, the new NMI vector (that is pointing at the routine, which is executed by the RESTORE key or other NMI source). There is also a cartridge signature, which is "CBM80".

Right after Reset or Power On, the Kernal is searching for the cartridge signature and in case it is found, the NMI vector is redirected and the CPU continues execution at the cartridge start address.

This is a disassembly of the reset routine, which the reset vector is pointing at.

```
FCE2   A2 FF      LDX #$FF
FCE4   78         SEI
FCE5   9A         TXS
FCE6   D8         CLD
FCE7   20 02 FD   JSR $FD02 ; checking the cartridge signature
FCEA   D0 03      BNE $FCEF
```

```
FCEC   6C 00 80   JMP ($8000) ; if found, jump tot h cartridge start
address
```
[…]

; compare the content of $8003 … to the signature

```
FD02   A2 05      LDX #$05
FD04   BD 0F FD   LDA $FD0F,X
FD07   DD 03 80   CMP $8003,X
FD0A   D0 03      BNE $FD0F
FD0C   CA         DEX
FD0D   D0 F5      BNE $FD04
FD0F   60         RTS

; cartridge signature „CBM80"

FD10   .BY $C3,$C2,$CD,$38,$30
```

Memory Persistence

The reason for the persistence in RAM is the internal structure of the DRAM. A DRAM cell is a simple structure which mainly consists of a MOSFET transistor and a capacitor. The capacitor holds the load that represents the state of the bit (which is of course "0" or "1").

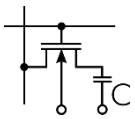

Figure 2: DRAM cell

To keep the information, it is periodically refreshed. In the C64, this job is done by the VIC-II chip. The cycle time of this refresh is much, much shorter, than the charge stays in the capacitor. So even after being powered off for a while, the charge might still be enough to hold the previous state of the bit.

It was reported by @CommodoreLad, that this persistence was longer with ASSY 250469 mainboards, especially those who had a Fujitsu type of DRAM.

The Code of Death

Since it takes quite some time to load a game, a small program was written, which pokes the fake cartridge signature into the RAM at address $8000…

```
10 DATA 00,127,09,128,195,194,205,56
20 DATA 48,32,121
30 FOR I=32768 TO 32778
40 READ V
50 POKE I,V
60 NEXT
```

This code will disable a proper boot of the C64 for a while. Even a power cycle will not destroy it. Don't panic, though, a power off of several minutes will help.

The "Unstoppable" EXROM Reset as a Cure

A so called EXROM reset is suggested to be the better reset. It is a temporary cure for the troubles caused by a fake cartridge signature in the RAM. Still, it does not erase the CBM80.

This can be done with a simple command:

```
POKE32772,0
```

which alters the CBM80 to @BM80, which is not recognized as a cartridge signature anymore.

How does the EXROM reset work?

Whether the RAM or the EPROM of the cartridge is read is determined by the PLA (a logic chip in the C64). A LOW level at the EXROM signal will select the cartridge EPROM, a HIGH signal selects the RAM. The CPU does not notice a difference here, it all happens automatically.

An EXROM reset hold the EXROM signal low for a short time after the RESET signal goes HIGH again. So instead of reading the fake CBM80 in RAM, the (random) content of a non-exiting cartridge is read. No CBM80 is found and the C64 boots normally. After EXROM is high again, the RAM is fully accessible again.

This is a good thing, but can conflict with some cartridges, that can switch the EXROM signal to HIGH to deactivate itself. In this case, the EXROM reset is not working perfectly.

The Kernal v2.1

The Kernal v2.1 is an original Commodore Kernal, which is slightly modified:

In an unused space of the Kernal, a very simple routine was inserted, which is setting the first byte of the fake signature CBM80 to $00. Then the execution is continued at the original reset routine.

```
E4B7   A9 00      LDA #$00
F4B9   8D 04 80   STA $8004
F4BC   4C E2 FC   JMP $FCE2
```

To start the execution at this routine, the RESET vector was modified:

```
FFFC   .WD $E4B7   ; RESET vector
```

Also, the start message "**** COMMODORE 64 BASIC V2 ****" at $E479 was modified to "*** COMMODORE 64 BASIC V2.1 ***"

Tests with different game that have proved to use this kind of copy protection have shown, that the Kernal v2.1 is preventing the annoying behavior as described before.