

0. Versions used

Article written by AlexC. Feel free to add contents!

- Vice 1.22
- C64C Pal
- Action Replay MK VI

1. Detecting cartridges

Most cartridges (including RR if active) can be detected by writing to range DE00-DEFF and checking if value written there will be persistent. This allows to detect: Action Replay MK VI, Final Cartridge III and Retro Replay. This method will fail against Trilogic Expert.

```
start:  lda $de10
        ldx #$0a
        ldy #$00

compare:
        cmp $de10
        bne nocart

delay:  dey
        bne delay

        dex
        bne compare

cartfound:
        jsr print
        rts

nocart: rts
```

1.1 FCIII

FCIII can also be detected by analyzing system vectors starting from \$0302:

```
$0302 = $DE41
$0304 = $DF8D
$0306 = $DE49
$0308 = $DE73
$030a = $DE4F
$0330 = $DE21 (LOAD)
```

```
$0332 = $DE35 (SAVE)
```

Following code can be use to detect FCIII vectors:

```
lda #$de
cmp $0303
beq crtfound
rts
crtfound:
jsr $fce2
```

The following code will restart system into clean basic - FCIII extensions will not be available until freeze, however cartridge led will be lit.

Another methods to detect FCIII can be based on it's control register: \$DFFF.

```
Bit 0: number of bank to show at $8000 ($3)
Bit 1: unused
Bit 2: unused
Bit 4-5: 00 turn all 16kb of ROM
01 start freezer
10 enable first 8kb of ROM
11 disable FCIII ROM
Bit 6: unused
Bit 7: 1 (always show 16kb of bank 0)
```

FCIII has following banks:

```
0 BASIC, Monitor
1 Notepad, BASIC (Bar)
2 Desktop, Freezer/Print
3 Freezer
```

To jump into freezer you can use following code:

```
lda #$9f
sta $dfff
```

The above code works however only in Vice - on real C64 (at least on my original copy of FCIII this enters freezer and hangs system). [this needs further research]

Using #\$B3 value instead will give you interesting result on real C64 and will jam CPU if you are using Vice.

How to use FCIII control register for detection? Simple: you can read from it, however it will not return the value that has been written to it. Instead - in case of default system start - it will always return #\$FF. This leads us to following code:

```
10 for i = 1 to 10: print peek(57343):next
```

Both on real C64 and Vice in case of FCIII you will get ten \$FF results. This allows to detect cart even if

kill command has been used. If you disconnect the cartridge you will get different results: at least two 0 values from above basic test (other values than 0 and FF are possible too!). Check it out yourself!

\$DFFF read value can differ if any writes has been done to it besides normal system start (using BASIC option from System menu).[this needs further research]

Another detection method is based on the fact that 512 bytes of FCIII ROM can't be turned off - this code is always there at \$DE00-DFFF. At \$DE01 you will find following code:

```
DE01 8D FF DF STA $DFFF
DE04 60 RTS
```

So to detect FCIII you can check if those bytes are there. If not - FCIII is not connected to the system.

1.2 Action Replay

Original Action Replay on real C64 will crash system if it's control register at \$DE00 is being read. Check the following code:

```
start: ldy #$0a
       lda $de00
       dey
       bne start
       rts
```

Actually one read is enough to crash system in case of all AR and it's clones I have.

Ever tough INC \$D020 can be dangerous for AR? I must be kidding right? Than check the following code on real C64:

```
$9000 nop
$9001 inc $d020
$9004 jmp $9000
```

Now try to freeze it few time. You will quickly find out that usually either the PC value is incorrect or after restarting you will hit BRK and enter monitor again.

As stated above AR has control resister at \$DE00. Here is a list of possible values you can write to it:

```
$00 enable bank 0
$06 disable cart and I/O area for it
$08 enable bank 1
$0A disable cart
$10 enable bank 2
$18 enable bank 3
$20 enable RAM at $8000 - $9FFF write to C64 memory underneath is enabled
$23 enable RAM at $8000 - $9FFF write to C64 memory underneath is disabled
```

Writing \$23 to \$DE00 will result in jump to the freezer if PC is above \$0FFF.

Here is the meaning of bits:

```
Bit 0: Game low (=1)
Bit 1: Exrom high (=1)
Bit 2: disable cart (=1) – turns off $de00 register
Bit 3: Rom bank selector low
Bit 4: Rom bank selector high
Bit 5: enable ram at $8000 and I/O
Bit 6: resets freeze mode
Bit 7: unused
```

1.3 Retro Replay

Detection – unless ZAP command has been used – is quite easy: use the code from section 1.

You can use following routine to jump into main menu (works with AR too):

```
sei
lda #$00
sta $de00
jmp $fce2
```

You can disable it (works with AR too) by following code:

```
sei
lda #$14
sta $de00
jsr $e453
cli
rts
```

Please note that this will not affect freeze button.

RR has in fact 2 control registers (from official RR manual): \$DE00 and \$DE01. There is theoretically possibility of disabling freeze button due to bit 2 of DE01. Bit 2 has NoFreeze name and can disable freeze button if set to 1 but can be written only once. RR 3.8 set it to 0 (enables freeze) at \$817F.

Consult RR manual for further Information: http://rr.c64.org/rr_manual.html#appb

1.3.2 Freeze vs breakpoint

If you ever wondered about differences between freezepoints and breakpoint here it is: breakpoint will work only if vector at \$FFFE and \$0316 has not been changed (it uses BRK). Freezepoint is using however JSR \$DFD3. [this needs further explanation]

From:
<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:
https://codebase64.org/doku.php?id=base:cartridge_detection

Last update: **2015-04-17 04:30**

