Quick exit from a timer interrupt

Written by Frantic.

I found the following trick when peeking through Mahoney's code for c64mp3 (source code available at Mahoney's homepage). After some more surfing I realized that this trick has been out there for a while (check this section of c64doc for example, written by John West and Marko Mäkelä), and that it is probably common knowledge among the elite coders...

Normally, when you exit from an interrupt, you first acknowledge the interrupt, and then execute the final rti instruction after that to return to the code that was executed before the interrupt was triggered. Acknowledging the interrupt involves reading a register, which typically takes 4 cycles:

```
    bit $dd0d    ;4 cycles - ack
    rti        ;quit interrupt routine
```

...and here comes the trick: If you set up a timer interrupt and make sure that $dd0c contains $40 (which is the byte value for the opcode rti), then you can simply quit the irq by performing a “jmp $dd0c”. Why would that be better? Well, the jmp only takes three cycles, instead of the four cycles consumed in the code above, and — here comes the trick – when the rti is executed it will implicitly “read” $dd0d (although the value of $dd0d does not end up in any of the cpu registers or anything like that), thus the interrupt will be acknowledged!

So...

```
    ;Interrupt init code
    […]
    lda #$40
    sta $dd0c    ;Put an "rti" instruction at $dd0c
    […]
```

...and:

```
    ;Interrupt code
    […]
    jmp $dd0c    ;3 cycles - quit interrupt
```

Of course, in either case the rti will have to be executed, taking 6 cycles, but the lda can be exchanged for a jmp, thus saving once cycle.

Why not? :)