



```

NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
// Add one extra nop for 65 cycle NTSC machines

// CYCLECOUNT: [64 -> 71]

WedgeIRQ:
// At this point the next Raster Compare IRQ has triggered and the
jitter is max 1 cycle.
// CYCLECOUNT: [7 -> 8] (7 cycles for the interrupt handler + [0 -> 1]
cycle Jitter for the NOP)

// Restore previous Stack Pointer (ignore the last Stack Manipulation by
the IRQ)
txs

// PAL-63 // NTSC-64 // NTSC-65
//-----//-----//-----
ldx #$08 // ldx #$08 // ldx #$09
dex // dex // dex
bne *-1 // bne *-1 // bne *-1
bit $00 // nop
// nop

// Check if $d012 is incremented and rectify with an additional cycle if
necessary
lda $d012
cmp $d012 // <- critical instruction (ZERO-Flag will indicate if Jitter
= 0 or 1)

// CYCLECOUNT: [61 -> 62] <- Will not work if this timing is wrong

// cmp $d012 is originally a 5 cycle instruction but due to pipelining
tech. the
// 5th cycle responsible for calculating the result is executed
simultaneously
// with the next OP fetch cycle (first cycle of beq *+2).

// Add one cycle if $d012 wasn't incremented (Jitter / ZERO-Flag = 0)
beq *+2

// Stable code

```

Don't forget to set up the next IRQ-Vector before exiting the IRQ.

If you want to make it more simple, you could store this as a pseudo or macro (STABILIZE) and just:

```
IRQ_Begin:
  pha
  txa
  pha
  tya
  pha
  :STABILIZE
  //..xxXX[Stable Code]XXxx..
  lda #<Next_IRQ
  sta $fffe
  lda #>Next_IRQ
  sta $ffff
  lda #$01
  sta $d019
  pla
  tay
  pla
  tax
  pla
  rti
```

TWW/Creators

From:

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

Permanent link:

[https://codebase64.org/doku.php?id=base:stable\\_raster\\_routine](https://codebase64.org/doku.php?id=base:stable_raster_routine)

Last update: **2015-04-17 04:34**

