```
;--------------------------------------------------------------------------
RasterRow = $2d ; May not be a line with sprites or a badline
_rx = $03
_ry = $04
_pageflip = $05
;--------------------------------------------------------------------------
* = $0801
;--------------------------------------------------------------------------
;- SYS $0810
;--------------------------------------------------------------------------
    .byte $0c, $08, $00, $00, $9e, $20, $32, $30
    .byte $36, $34, $00, $00, $00, $00, $00
;--------------------------------------------------------------------------
* = $0810
;--------------------------------------------------------------------------
    sei
;--------------------------------------------------------------------------
; Turn all sprites off
;--------------------------------------------------------------------------
    lda #$00
    sta $d021
    sta $d020
    sta $d015
;--------------------------------------------------------------------------
; Wait for VBLANC
;--------------------------------------------------------------------------
    bit $d011
    bpl * - 3
    bit $d011
    bmi * - 3
;--------------------------------------------------------------------------
; Perform halfvariance polling
;--------------------------------------------------------------------------
.page
    ldx $d012
    inx
    cpx $d012
    bne * - 3
    ldy #$0a
    dey
    bne * - 1
    inx
    cpx $d012
    nop
    beq Time_1
    nop
    bit $24
Time_1  ldy #$09
    dey
    bne * - 1
    nop
```

```
    nop
    inx
    cpx $d012
    nop
    beq Time_2
    bit $24
Time_2  ldy #$0a
    dey
    bne * - 1
    inx
    cpx $d012
    bne Time_3
Time_3  nop
    nop
    nop
    nop
    nop
;-------------------------------------------------------------------------
; Raster is stable here.
; Launch a continous timer which a count of 63.
;-------------------------------------------------------------------------
    lda #$3e
    sta $dc04
    sty $dc05
    lda #$11
    sta $dc0e
.endp
;-------------------------------------------------------------------------
; Initialize everything needed to be inited here
;-------------------------------------------------------------------------
    ; Set multicolor-mode and turn on the bitmap, in the process set
    ; the high end of the raster to 0, point the screenmemory to
    ; $0000 - $3fff, allocated positions are:
    ; Bitmap @ $4000
    ; Screen 1 = $6000
    ; Screen 2 = $6400
    ; Screen 3 = $6800 (For doublebuffering)
    ; Screen 4 = $6c00 (For doublebuffering)
    lda #$3b
    sta $d011
    lda $d016
    ora #%00010000
    sta $d016
    lda $dd00
    and #%11111100
    ora #%00000010
    sta $dd00
    ldx #0
    stx _pageflip
    lda #%10100101
FillScreen
```

```
    sta $4000, x
    sta $4100, x
    sta $4200, x
    sta $4300, x
    sta $4400, x
    sta $4500, x
    sta $4600, x
    sta $4700, x
    sta $4800, x
    sta $4900, x
    sta $4a00, x
    sta $4b00, x
    sta $4c00, x
    sta $4d00, x
    sta $4e00, x
    sta $4f00, x
    sta $5000, x
    sta $5100, x
    sta $5200, x
    sta $5300, x
    sta $5400, x
    sta $5500, x
    sta $5600, x
    sta $5700, x
    sta $5800, x
    sta $5900, x
    sta $5a00, x
    sta $5b00, x
    sta $5c00, x
    sta $5d00, x
    sta $5e00, x
    sta $5f00, x
    inx
    bne FillScreen
SetColorloop
    lda #$9e
    sta $6000, x
    lda #$2f
    sta $6400, x
    lda #$57
    sta $6100, x
    lda #$8a
    sta $6500, x
    lda #$c1
    sta $6200, x
    lda #$bd
    sta $6600, x
    lda #$10
    sta $6300, x
    lda #$f6
    sta $6700, x
```

```
    lda #$69
    sta $6000 + $800, x
    lda #$2b
    sta $6400 + $800, x
    lda #$58
    sta $6100 + $800, x
    lda #$83
    sta $6500 + $800, x
    lda #$f1
    sta $6200 + $800, x
    lda #$be
    sta $6600 + $800, x
    lda #$10
    sta $6300 + $800, x
    lda #$a6
    sta $6700 + $800, x
    inx
    bne SetColorloop
;--------------------------------------------------------------------------
; Setup irq-code
;--------------------------------------------------------------------------
    lda #$7f
    sta $dc0d
    sta $dd0d
    lda #1
    sta $d01a
    sta $d019
;--------------------------------------------------------------------------
; Setup interrupt position
;--------------------------------------------------------------------------
    lda #RasterRow
    sta $d012
;--------------------------------------------------------------------------
    lda #$35
    sta $01
    lda #<irq
    sta $fffe
    lda #>irq
    sta $ffff
    lda $dc0d
    lda $dd0d
    asl $d019
;--------------------------------------------------------------------------
; Setup the nmi-irq
;--------------------------------------------------------------------------
    lda #<nmiirq
    sta $fffa
    lda #>nmiirq
    sta $fffb
;--------------------------------------------------------------------------
; Setup the timer adjusted to work every eight row, (63 * 8) - 1 cycles
```

```
;--------------------------------------------------------------------------
    lda #$f7
    sta $dd04
    lda #$01
    sta $dd05
    cli
;--------------------------------------------------------------------------
; Here is the maincode executed
;--------------------------------------------------------------------------
MainCode
    lda _pageflip
    eor #$ff
    sta _pageflip
    jmp MainCode
;--------------------------------------------------------------------------
irq
;--------------------------------------------------------------------------
; Save the registers
;--------------------------------------------------------------------------
    pha
    php
    stx _rx
    sty _ry
;--------------------------------------------------------------------------
; Stabilize the raster
;--------------------------------------------------------------------------
    lda $dc04
    and #7
    sta Time_4 + 1
    lda #7
    sec
Time_4  sbc #4
    sta Time_5 + 1
.page
Time_5  bpl * + 2
    cmp #$c9
    cmp #$c9
    cmp #$c9
    cmp #$24
    nop
.endp
;--------------------------------------------------------------------------
; Stable raster here, start the nmi-irq if the raster is on the beginning of
the screen
;--------------------------------------------------------------------------
.page
    lda $d012
    cmp #RasterRow + 8
    bcs StopNMI
    lda _pageflip    ;3     3
    bmi _page_2    ;2 3     5  6
```

```
    lda #$81    ;2     7
    sta _p_2 + 1    ;4    11
    lda #$91    ;2    13
    sta _p_1 + 1    ;4    17
    nop         ;2    19
    jmp start_nmi    ;3    22
_page_2 lda #$a1    ;  2        8
    sta _p_2 + 1    ;  4       12
    lda #$b1    ;  2       14
    sta _p_1 + 1    ;  4       18
    nop         ;  2       20
    nop         ;  2       22
start_nmi
    bit $ea        ;3 3    25 25
.endp
    lda #$11
    sta $dd0e
    lda #$81
    sta $dd0d
    lda $dd0d
;-------------------------------------------------------------------------
; Set raster-row for ending the NMI here
;-------------------------------------------------------------------------
    lda #$f8
    sta $d012
;-------------------------------------------------------------------------
; Restore all registers
;-------------------------------------------------------------------------
    plp
    pla
    ldx _rx
    ldy _ry
;-------------------------------------------------------------------------
; Return from irq here
;-------------------------------------------------------------------------
    asl $d019
    rti
;-------------------------------------------------------------------------
; Stop the NMI-irq, play music and stuff
;-------------------------------------------------------------------------
StopNMI
    lda _p_2 + 1    ; Restore $d018 for the first row
    sta $d018
    lda #RasterRow
    sta $d012
    lda #$00
    sta $dd0e
;-------------------------------------------------------------------------
; Restore all registers
;-------------------------------------------------------------------------
    plp
```

```
    pla
    ldx _rx
    ldy _ry
;--------------------------------------------------------------------------
; Return from irq here
;--------------------------------------------------------------------------
    asl $d019
    rti
nmiirq
;--------------------------------------------------------------------------
; Stabilize the raster
;--------------------------------------------------------------------------
    sta reg_a + 1
    lda $dd04
    eor #$e7
    sta time_6 + 1
.page
time_6  bpl *
    cmp #$c9
    cmp #$c9
    cmp #$24
    nop
.endp
;--------------------------------------------------------------------------
; Stable raster here, create badline for fli and restore to next badline
;--------------------------------------------------------------------------
_p_1    lda #$91
    sta $d018
    lda #$3f
    sta $d011
_p_2    lda #$81
    sta $d018
    lda #$3b
    sta $d011
;--------------------------------------------------------------------------
; Trigger new NMI, restore A and return from IRQ here
;--------------------------------------------------------------------------
    lda $dd0d
reg_a   lda #0
    rti

;   .byte $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9
;   .byte $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9
;   .byte $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9
;   .byte $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9
;   .byte $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9
;   .byte $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9, $c9
;   bit $ea
```

Pitfalls

by Bitbreaker/Oxyron* (the code example is not mine, it was here before, i just added some rant)

Fix Me! isn't it a bad idea to do the following, as the asl clobbers the carry:

```
plp
pla
asl $d019
rti
```

So better pull P and A after asl $d019. However the processor status is anyway pushed to the stack on an interrupt and restored by the rti.

Also the NMIIRQ does not work with both CIA types, one might use two different NMI-handlers: In case of using a new CIA you better waste an additional cycle by:

```
sta $06
nop
lda $dd04
eor #$e7
...
lda $06
rti
```