

2nd Pixel FLI Distorter

The concept for this routine is to generate an animation of a logo wobbling in some sine waves. Instead of playing the animation normally, some sine values control which frame is displayed for each rasterline, similarly to e.g. rubbervector. I couldn't find a way to do this with hardware tricks, so the gfx had to be copied in software. To optimize it a bit, it skips a line if it's the same as the previous frame.

```

;-----
----
; 2nd Pixel FLI Distorter
;-----
----
; Coded by Cruzer/CML 2002-2004
; Released in You Know the Routine/Camelot
; Assembled with MXass
; To get the binaries: Run the demo part and save $1000-$bfff
;-----
----
; Disclaimer: I'm not very proud of the state of this code in terms of
; readability and stuff. The assembler is also kinda lame, so I cannot
; recommend it. I'm now using Kick Assembler, which I can highly recommend.
;-----
----
;0800-081f  Basic
;0820-09ff  Siner
;1000-1eff  Music
;2000-31ff  Main
;3000-3fff  Logo source / hiphop-font stuff
;4000-????  Generated wobble anims
;bf00-????  Wobbler
;c000-dfff  FLI-Bitmap
;e000-ffff  FLI-Screens

.la newlogo=0
;-----
----

.la width = 24  ;chars
.la height = 67 ;pixels
.la frames = 20 ;in logo animation

.la wp = $02
.la wpo = wp+height
.la hiphopzp = $a0

.la hiphopscreen = $0400
.la siner = $0820 ;*
;.la sine = $0c00 ;used by logo generator

```

```
.la d011      = $0c00 ;*
.la d018      = $0c50 ;*
.la sine2     = $0e00
.la main      = $21b0
.la logo      = $3000 ;note! overwritten by code+data now!
.la hiphopfont = $3200
.la hiphopcharset = $3800 ;*
.la anim      = $4000
.la wobbler   = $bf00
.la bitmap    = $c000
.la screens   = $e000

.ba main
.la border=$d0ff

;-----
----
    jmp start

x    .by 0
y    .by 0

tmp .wo 0

wobbling .by 0

;-----
----
start

    sei
    lda #$00
    sta $d020
    sta $d021

    jsr init

    lda #$00
    jsr $1000

    cli
    sei

    lda #<main1
    sta $0314
    lda #>main1
    sta $0315
    asl $d019
    lda #$7b
    sta $dc0d
```

```
    lda #$81
    sta $d01a
    lda #$1b
    sta $d011
    lda #$a0
    sta $d012
    cli

    ldx #0
    ldy #0
-
    nop
    nop
    inx
    bne -
    iny
    bne -

-loop
    jsr hiphopclear

    lda wobbling
    bne +skip

    lda msg
    asl
    tax
    lda hiphopscreenoffsets,x
    cmp #$ff
    bne +
    jsr hiphopscreensetup3
    jmp +skip
+
    sta hiphopscreenoffset
    lda hiphopscreenoffsets+1,x
    sta hiphopscreenoffset+1
    jsr hiphopscreensetup
+skip
    lda #1
    sta cleared

    jsr tag

    lda #1
    sta tagged

-wait
    lda hiphopstop
    bne +load
    lda tagged
    bne -wait
    jmp -loop
```

```
+load
```

```
    lda #1  
    sta stopirq
```

```
-   lda irqstopped  
    beq -
```

```
loadnext
```

```
    sei  
    lda #$0a  
    sta $d021  
    lda #$00  
    sta $d01a  
    sta $d015  
    lda #$14  
    sta $d018  
    lda #$37  
    sta $01  
    lda zp2e  
    sta $2e  
    lda zp2f  
    sta $2f
```

```
    ldx #0
```

```
-   lda $0a00,x  
    sta $c000,x  
    lda $0b00,x  
    sta $c100,x  
    inx  
    bne -
```

```
    jmp $c1f0
```

```
zp2e .by 0  
zp2f .by 0  
tagged .by 0  
cleared .by 0  
hiphopstop .by 0
```

```
stopirq .by 0  
irqstopped .by 0
```

```
;------  
----
```

```
main1 ;just text...
```

```
    asl $d019
```

```
    inc border
    jsr $1003

    lda doubledraw
    bne +notimetowaste
    inc border
    lda #$fc
-   cmp $d012
    bne -

    lda #$20
-   cmp $d012
    bne -

+notimetowaste
    lda #$97
    sta $dd00
    lda #$1e
    sta $d018
    lda #$c8
    sta $d016
    lda hiphopcolor
    sta $d021
    lda d011+$0f
    and #$1f
    sta $d011

    inc border
    jsr hiphopcontrol

    jsr spacecontrol

    lda #$09
    sta border

    lda wobbling
    beq ri

    lda #<main2
    sta $0314
    lda #>main2
    sta $0315
    lda #$53
    sta $d012
    asl $d019
ri
    pla
    tay
    pla
    tax
```

```
pla  
rti
```

```
main2 ;wobbling+text...
```

```
asl $d019
```

```
inc border  
jsr fli show
```

```
inc border  
jsr $1003
```

```
lda cnt  
ora cnt+1  
bne +wob  
lda #1  
sta border  
jsr updatesiner  
jmp +skip
```

```
+wob
```

```
inc border  
jsr wobbleit
```

```
+skip
```

```
jsr hiphopcontrol
```

```
lda #$00  
sta border
```

```
jsr effectcontrol
```

```
jsr spacecontrol  
jsr checkrastertime
```

```
lda #$09  
sta border
```

```
jmp ri
```

```
fadeoutirq
```

```
asl $d019
```

```
    lda stopirq
    bne +stop

    ldx cnt
    lda fadeacolz,x
    sta $d020
    lda cnt
    cmp #9
    beq +
    inc cnt
    jmp +j
+
    jmp loadnext
    lda #1
    sta hiphopstop
+j
    jmp ri

+stop
    lda #$00
    sta $d01a
    lda #1
    sta irqstopped
    jmp ri

fadeacolz .by $0,$b,$c,$f,$1,$1,$f,$c,$b,$0

;-----
----
spacecontrol

    lda $dc01
    cmp #$ef
    beq +space
    rts

+space
    lda #<fadeoutirq
    sta $0314
    lda #>fadeoutirq
    sta $0315
    lda #$00
    sta $d012
    sta cnt
    asl $d019
    sta $d418
    sta $d011

    rts
```

```
;-----  
----  
checkrastertime  
  
    ;if (rastertime==too little)  
    ;then use more, so hiphop-msg don't get tagged too fast...  
-   lda $d012  
    sec  
    sbc #$aa  
    bcs -  
-   lda $d011  
    and #$80  
    bne -  
  
    rts  
  
;-----  
----  
effectcontrol  
  
    lda cnt  
    clc  
    adc #1  
    sta cnt  
    lda cnt+1  
    adc #0  
    sta cnt+1  
  
    lda cnt+0  
    cmp #0  
    bne +  
    lda cnt+1  
    cmp #2  
    bne +  
    jsr startfadeout  
+  
  
    jsr effectfadein  
    jsr effectfadeout  
  
    rts  
  
cnt .wo 0  
  
neweffect  
  
    ldx effect  
    lda sineadds,x
```



```

    sta sineadd
    txa
    asl
    tax
    lda sinespreads,x
    sta sinespread
    lda sinespreads+1,x
    sta sinespread+1

    lda #0
    sta cnt
    sta cnt+1

    inc effect
    lda effect
    cmp #4
    bne +
    lda #0
    sta effect
+

    lda #1
    sta first

    jsr startfadein

    rts

sinespreads
    ;.wo $0100,$0080,$0140,$00c0
    .wo $0100,$0080,$0140,$fe80

sineadds
    ;.by $01,$ff,$01,$01
    .by $01,$ff,$01,$01

;-----
----
startfadeout

    lda #0
    sta fadeoutpnt
    rts

effectfadeout

    lda fadeoutpnt
    cmp #20
    bne +
    rts
+

```

```
    lsr
    tax
    lda d011,x
    ora #$40
    sta d011,x
    lda d011+10,x
    ora #$40
    sta d011+10,x
    lda d011+20,x
    ora #$40
    sta d011+20,x
    lda d011+30,x
    ora #$40
    sta d011+30,x
    lda d011+40,x
    ora #$40
    sta d011+40,x
    lda d011+50,x
    ora #$40
    sta d011+50,x
    lda d011+60,x
    ora #$40
    sta d011+60,x
    lda d011+70,x
    ora #$40
    sta d011+70,x

    inc fadeoutpnt
    lda fadeoutpnt
    cmp #20
    bne +
    jsr neweffect
+

    rts

fadeoutpnt .by 20

startfadein

    lda #0
    sta fadeinpnt
    rts

effectfadein

    lda fadeinpnt
    cmp #20
```

```
    bne +
    rts
+
    sec
    sbc #10
    bcs +
    jmp +j
+
    tax
    lda d011,x
    and #$3f
    sta d011,x
    lda d011+10,x
    and #$3f
    sta d011+10,x
    lda d011+20,x
    and #$3f
    sta d011+20,x
    lda d011+30,x
    and #$3f
    sta d011+30,x
    lda d011+40,x
    and #$3f
    sta d011+40,x
    lda d011+50,x
    and #$3f
    sta d011+50,x
    lda d011+60,x
    and #$3f
    sta d011+60,x
    lda d011+70,x
    and #$3f
    sta d011+70,x

+j
    inc fadeinpnt

    rts

fadeinpnt .by 20

;.text "      "
;-----
----
flishow

    lda #$00
    sta $d021

    nop
    nop
```

```
lda #$94
sta $dd00

lda $d011
ora #$30
sta $d011

lda #$d8
sta $d016

;nop
;nop
;nop
nop
nop

ldy #$00
-loop
lda d011,y
sta $d011
lda d018,y
sta $d018
iny
cpy #$4c
bne -loop

inc border

lda d011+$0f
ora #$70
;and #$1f
sta $d011

lda #$97
sta $dd00
lda #$1e
sta $d018
lda #$c8
sta $d016
lda hiphopcolor
sta $d021

ldx #$1c
- dex
bne -
lda d011+$0f
and #$1f
sta $d011
```

```
lda #$00
sta $ffff
rts
```

```
;-----
-----
```

```
wobbleit
```

```
;copy wobblepointers to old wobblepointers
```

```
lda wp+0
sta wpo+0
lda wp+1
sta wpo+1
lda wp+2
sta wpo+2
lda wp+3
sta wpo+3
lda wp+4
sta wpo+4
lda wp+5
sta wpo+5
lda wp+6
sta wpo+6
lda wp+7
sta wpo+7
lda wp+8
sta wpo+8
lda wp+9
sta wpo+9
lda wp+10
sta wpo+10
lda wp+11
sta wpo+11
lda wp+12
sta wpo+12
lda wp+13
sta wpo+13
lda wp+14
sta wpo+14
lda wp+15
sta wpo+15
lda wp+16
sta wpo+16
lda wp+17
sta wpo+17
lda wp+18
sta wpo+18
lda wp+19
sta wpo+19
```

```
lda wp+20
sta wpo+20
lda wp+21
sta wpo+21
lda wp+22
sta wpo+22
lda wp+23
sta wpo+23
lda wp+24
sta wpo+24
lda wp+25
sta wpo+25
lda wp+26
sta wpo+26
lda wp+27
sta wpo+27
lda wp+28
sta wpo+28
lda wp+29
sta wpo+29
lda wp+30
sta wpo+30
lda wp+31
sta wpo+31
lda wp+32
sta wpo+32
lda wp+33
sta wpo+33
lda wp+34
sta wpo+34
lda wp+35
sta wpo+35
lda wp+36
sta wpo+36
lda wp+37
sta wpo+37
lda wp+38
sta wpo+38
lda wp+39
sta wpo+39
lda wp+40
sta wpo+40
lda wp+41
sta wpo+41
lda wp+42
sta wpo+42
lda wp+43
sta wpo+43
lda wp+44
sta wpo+44
lda wp+45
```

```
sta wpo+45
lda wp+46
sta wpo+46
lda wp+47
sta wpo+47
lda wp+48
sta wpo+48
lda wp+49
sta wpo+49
lda wp+50
sta wpo+50
lda wp+51
sta wpo+51
lda wp+52
sta wpo+52
lda wp+53
sta wpo+53
lda wp+54
sta wpo+54
lda wp+55
sta wpo+55
lda wp+56
sta wpo+56
lda wp+57
sta wpo+57
lda wp+58
sta wpo+58
lda wp+59
sta wpo+59
lda wp+60
sta wpo+60
lda wp+61
sta wpo+61
lda wp+62
sta wpo+62
lda wp+63
sta wpo+63
lda wp+64
sta wpo+64
lda wp+65
sta wpo+65
lda wp+66
sta wpo+66
```

```
inc border
```

```
;lda effect
;beq +showanim
```

```
+sine

    lda sinepnt
    tay
    lda first
    beq +
    jsr siner+$a0
    jmp +j
+   jsr siner
+j  lda #0
    sta first

    lda sinepnt
    clc
    adc sineadd
    sta sinepnt

    jmp +skip

first .by 0
effect .by 0
sinepnt .by 0
sineadd .by 0

+showanim

    ;show anim-frames...

    ldx #height
    lda +testfr
    beq +
    cmp #2
    beq +sk
    dex
+   lda +testfr+1
-   sta wp,x
    dex
    dex
    bpl -

+sk

    inc +testfr
    lda +testfr
    cmp #3 ;3rd frame
    bne +
    lda #0
```



```
    sta +testfr
    inc +testfr+1
    lda +testfr+1
    cmp #20
    bne +
    lda #0
    sta +testfr+1
```

```
+
```

```
    jmp +skip
```

```
+testfr .wo 0
```

```
+skip
```

```
    inc border
```

```
    lda #$34
    sta $01
```

```
    jsr wobbler
```

```
    lda #$36
    sta $01
```

```
    rts
```

```
+fr .by 0
```

```
;-----
----
;hiphop-font depacking...
;-----
----
```

```
hiphopinit
```

```
    lda $2e
    sta zp2e
    lda $2f
    sta zp2f
```

```
    jsr hiphopclear
    jsr hiphopscreensetup
```

```
    rts
```

hiphopscreensetup3

```
    ;lda #<244+280
    lda #<533
    sta hiphopscreenoffset
    lda #>533
    sta hiphopscreenoffset+1
    jsr hiphopscreensetup
```

```
    rts
```

hiphopscreensetup

```
    ldx #$00
    txa
-
    sta hiphopscreen+$0000,x
    sta hiphopscreen+$0100,x
    sta hiphopscreen+$0200,x
    sta hiphopscreen+$0300,x
    inx
    bne -
```

```
.la hhs=$f0
```

hiphopscreensetup2

```
    lda #<hiphopscreen
    sta hhs
    lda #>hiphopscreen
    sta hhs+1
    lda hhs
    clc
    adc hiphopscreenoffset
    sta hhs
    lda hhs+1
    adc hiphopscreenoffset+1
    sta hhs+1
    lda #0
    sta +tmp
```

```
    ldy #0 ;x
```

```
-xloop
```

```
    ldx #0 ;y
-yloop
    lda +tmp
    sta (hhs),y
    inc +tmp

    lda hhs
    clc
    adc #40
    sta hhs
    lda hhs+1
    adc #0
    sta hhs+1

    inx
    cpx #8
    bne -yloop
    lda hhs
    sec
    sbc #<320
    sta hhs
    lda hhs+1
    sbc #>320
    sta hhs+1
    iny
    cpy #32
    bne -xloop

    rts
```

```
hiphopscreenoffset .wo 0
hiphopscreenoffsets .wo 0,568,284,88,557,0,284,$ffff,564,564
+tmp .by 0
```

hiphopclear

```
    ldx #$00
-loop
    lda #$ff
    sta hiphopcharset+$0000,x
    sta hiphopcharset+$0100,x
    sta hiphopcharset+$0200,x
    sta hiphopcharset+$0300,x
```

```
    sta hiphopcharset+$0400,x
    sta hiphopcharset+$0500,x
    sta hiphopcharset+$0600,x
    sta hiphopcharset+$0700,x
    inx
    bne -loop
```

```
;    lda #%00111000
;    sta hiphopcharset+3
;    sta hiphopcharset+6
;    lda #%01111100
;    sta hiphopcharset+4
;    sta hiphopcharset+5
```

```
    rts
```

hiphopcontrol

```
    lda cleared
    bne +
    lda #0
    sta hiphopcolor
    rts
```

```
+
    ldx hiphopcolor+1
    lda hiphopcolors+15,x
    sta hiphopcolor
    lda tagged
    bne +
    rts
```

```
+
    dec hiphopcnt
    beq +
    jmp +fade
```

```
+
    lda #0
    sta tagged
    sta cleared
    sta hiphopcolor
    lda hiphopcnt+1
    sta hiphopcnt
    lda hiphopcolor+1
    clc
    adc #$10
    cmp #$60
    bne +
    lda #$00
```

```

+   sta hiphopcolor+1
    rts

+fade
    lda hiphopcncnt
    sec
    sbc #$10
    bcc +
    rts

+
    lda hiphopcncnt
    ora hiphopcolor+1
    tax
    lda hiphopcolors,x
    sta hiphopcolor
    rts

hiphocncnt .by $10,$68,$58

hiphopcolor .by 2,$20
hiphopcolors
    .by $0,$0,$9,$b,$4,$a,$f,$7,$1,$1,$1,$1,$7,$f,$a,$4
    .by $0,$0,$0,$6,$b,$e,$f,$7,$1,$1,$1,$1,$7,$f,$e,$e
    .by $0,$0,$9,$2,$4,$a,$f,$7,$1,$1,$1,$1,$7,$f,$a,$a
    .by $0,$0,$0,$6,$b,$e,$3,$d,$1,$1,$1,$1,$d,$3,$3,$3
    .by $0,$0,$9,$b,$5,$3,$d,$d,$1,$1,$1,$1,$1,$d,$d,$5
    .by $0,$0,$9,$2,$8,$a,$f,$7,$1,$1,$1,$1,$1,$7,$7,$7

tag
    lda #0
    sta +mp
    sta +x
    sta +y
    sta +line
    sta doubledraw
    lda #1
    sta oneline

-loop

    ldx +mp
ml   lda msg1,x
    cmp #$f0
    bne +
    lda #1
    sta wobbling
    jmp +skip
+   cmp #$f1
    bne +

```

```
    lda #1
    sta doubledraw
    jmp +skip
+   cmp #$21 ;"! "
    bne +
    lda #$1b
+   cmp #$2e ;"."
    bne +
    lda #$1c
+   cmp #$23 ;"
    bne +
    lda #$1d
+   cmp #$27 ;"'"
    bne +
    lda #$1e
+   cmp #$2d ;"- "
    bne +
    lda #$1f
+   cmp #$bd ;"½"
    bne +
    ldx #$20
    jmp +skip
+   cmp #$3c ;EOL
    beq +eol

    and #$1f
    sta +chr
    beq +space

    ldx +line
    bne +
    ;ldx #0
    stx oneline
+
    sta +chr
    ldx +x
    ldy +y
    jsr drawchar
    lda doubledraw
    beq +
    lda +chr
    ldx +x
    inx
    ldy +y
    iny
    jsr drawchar
+
+space
    ldx +chr
+skip
```

```
    lda +x
    clc
    adc +widths,x
    sta +x
    inc +mp
    ;lda +mp
    ;cmp #20
    ;bne +
    jmp -loop

+eol

    inc +mp

    ldx oneline
    lda +y
    clc
    adc linedifs,x
    sta +y
    lda #0
    sta +x

+
    inc +line
    lda +line
    cmp #2
    ;lda +mp
    ;cmp #40
    beq +
    jmp -loop

+
    inc msg
    lda msg
    cmp msg+1
    bne +

    lda #0
    sta msg

    lda #<msg1
    sta ml+1
    lda #>msg1
    sta ml+2
    ldx #17
-   lda msg99,x
    sta msg9,x
    dex
    bpl -

+

    lda msg
```

```
    asl
    tax
    lda messages,x
    sta ml+1
    lda messages+1,x
    sta ml+2

;   lda ml+1
;   clc
;   adc #40
;   sta ml+1
;   lda ml+2
;   adc #0
;   sta ml+2

    ldx oneline
    lda hiphopcnt+1,x
    sta hiphopcnt

    rts

+mp .by 0
+chr .by 0
+x .by 0
+y .by 0
+line .by 0
oneline .by 0
doubledraw .by 0
linedifs .by 24,12
msg .by 0,30

messages
    .wo msg1,msg2,msg3,msg4,msg5,msg6,msg7
    .wo msg8,msg9,msg10,msg11,msg11b,msg12,msg12b
    .wo msg13,msg14,msg15,msg16,msg17,msg18,msg19
    .wo msg20,msg21,msg22,msg23,msg24,msg25,msg26,msg27,msg28

msg1
    .te "<"
    .te "you know the deal<"
msg2
    .te "<"
    .te "you know the name<"
msg3
    .te "<"
    .te "you know the game<"
msg4
    .te "    you know the<"
    .te "name of the game<"
msg5
    .te "        you know<"
```



```
.te " the procedure<"
msg6
.te "<"
.te " and now ...<"
msg7
.by $f1
.te " you know<"
.te " the routine<"
msg8
.te "<"
.te " a demo by ...<"
msg9
.by $f0
.te "<"
.te " <"

msg10
.te "<"
.te " at it again1/2...<"
msg11
.te "commanding<"
.te "1/2 the commodore<"
msg11b
.te " and flexing the<"
.te " eight bit muscle<"

msg12
.te " with diz brand1/2-<"
.te "spanking new1/2...<"
msg12b
.te "<"
.te "kamel produktion1/2!<"

; .te "1/2 in a brand new<"
; .te "1/2kamel produktion<"
msg13
.te " now we hopefully<"
.te " 1/2 get back the ...<"
msg14
.te " street<"
.te " credibility !<"
msg15
.te "creditz for<"
.te " 1/2da whole thang1/2...<"
msg16
.te " code1/2'n'1/2stuff<"
.te "1/2 by cruiser ...<"
msg17
.te "1/2 and synthetic<"
.te "electronic sound...<"
```

```
msg18
    .te "mostly recycled<"
    .te "½oldskewl tunes½...<"
msg19
    .te "  by drax½...jch½<"
    .te "...fanta and jeff<"
msg20
    .te "<"
    .te "  pleez note ...<"
msg21
    .te "  diz demo<"
    .te "    only runs...<"
msg22
    .te "<"
    .te "½  on emulators½!<"
msg23
    .te "<"
    .te "  ½  bah½!<"
msg24
    .te "<"
    .te "    okiez ...<"
msg25
    .te "<"
    .te "  ½ space time½!<"
msg26
    .te "<"
    .te "know wutta mean½...<"
msg27
msg28
    .te "<"
    .te "<"

msg99
    .by $f1
    .te "<"
    .te "  camelot½!<"

+widths
    .by 12,21,15,16 ; abc
    .by 17,12,18,15 ;defg
    .by 12,14,20,17 ;hijk
    .by 17,17,12,12 ;lmno
    .by 14,13,17,22 ;pqrs
    .by 17,13,12,18 ;tuvw
    .by 15,20,22, 9 ;xyz!
    .by 7,14, 6,15 ;. "' -
    .by 7          ;½
```

drawchar

```
    ;x=x y=y a=char

    stx +x
    sty +y

;   cmp #0 ;space
;   bne +
;   rts
;+
    sec
    sbc #1
    sta +chr

;fetch start-adr...

    ;adr for first char...
    lda #$80
    sta +adr+1
    lda #>hiphopfont+1
    sta +adr+2

    ;if (!first char) look it up...
    lda +chr
    beq +
    sec
    sbc #1
    asl
    tax
    lda hiphopfont,x
    sta +adr+1
    lda hiphopfont+1,x
    clc
    adc #>hiphopfont
    sta +adr+2
+

;fetch bytes pr. line...

    lda +chr
    asl
    tax
    lda mul10,x
    sta +l+1
    lda mul10+1,x
    clc
    adc #>hiphopfont
    sta +l+2
    ldy #0
```

```
    ldx #$40
-loop
+l  lda hiphopfont,x
    sta +tmp
    and #%11000000
    lsr
    lsr
    lsr
    lsr
    lsr
    lsr
    sta +bpl+0,y
    lda +tmp
    and #%00110000
    lsr
    lsr
    lsr
    lsr
    sta +bpl+1,y
    lda +tmp
    and #%00001100
    lsr
    lsr
    sta +bpl+2,y
    lda +tmp
    and #%00000011
    sta +bpl+3,y
    inx
    iny
    iny
    iny
    iny
    cpy #40
    bne -loop

;fetch lines'n'draw'em...

    lda #0
    sta +line
    sta +fetchpnt

-yloop

    lda +x
    sta +xx

-xloop

    ldy +line
    lda +bpl,y
```

```
    beq +nextline
    ldx +fetchpnt
+adr   lda hiphopfont+384,x
    tax
    lsr
    lsr
    lsr
    lsr
    sta +black
    txa
    and #$0f
    sta +white
    beq +zerowidth

    lda +y
    clc
    adc +line
    tay
    lda +xx
    clc
    adc +black
    tax
    lda +white

    jmp drawline
backfromdrawline

+zerowidth

    lda +black
    clc
    adc +white
    adc +xx
    sta +xx
    ldx +line
    dec +bpl,x

    inc +fetchpnt

    jmp -xloop

+nextline

    inc +line
    lda +line
    cmp #40
    bne -yloop
    rts

+x   .by 0
+xx  .by 0
```



```
    adc +w
    sta +cx+1

-loop
    lda +pixels,x
    beq +nextrow
    and ($a0),y
    sta ($a0),y

-back
    inx
+cx cpx #$ff
    bne -loop
    jmp +done

+nextrow

    lda #%11111110
    and ($a0),y
    sta ($a0),y

    lda $a0
    clc
    adc #$40
    sta $a0
    lda $a1
    adc #0
    and #$07
    ora #>hiphopcharset
    sta $a1
    jmp -back

+done

    jmp backfromdrawline

+w .by 0

+pixels
    .by %01111111
    .by %10111111
    .by %11011111
    .by %11101111
    .by %11110111
    .by %11111011
    .by %11111101
    .by 0
    .by %01111111
    .by %10111111
```

```
.by %11011111
.by %11101111
.by %11110111
.by %11111011
.by %11111101
.by 0
.by %01111111
.by %10111111
.by %11011111
.by %11101111
.by %11110111
.by %11111011
```

```
;-----
----
```

makeanim

```
;cut'n'pasted in seperate file!
rts
```

```
;-----
----
```

init

```
lda #$36
sta $01
```

```
jsr vicinit
jsr fillbitmap
```

```
ldx #$00
- lda sine2,x
sta sine2+$100,x
inx
bne -
```

```
lda #newlogo
beq +
jsr makeanim
```

```
+ jsr hiphopinit
```

```
jsr makewobbler
jsr makesiner
jsr neweffect
```



```
    ldx #0
    lda #$ff
-
    sta wp,x
    sta wpo,x
    inx
    cpx #height
    bne -
    ldx #$4f
-
    lda d011,x
    ora #$40
    sta d011,x
    dex
    bpl -
    rts

;-----
----
vicinit

    lda #$97
    sta $dd00
    lda #$c8
    sta $d016
    lda #$00
    sta $d011
    lda #$00
    sta $d015
    sta $d020
    sta $d021

    ldx #0
    lda #$00
-
    sta $d800,x
    sta $d900,x
    sta $da00,x
    sta $db00,x
    inx
    bne -

    rts

;-----
----
fillbitmap

;fill bitmap with 2nd pixel pattern...

    lda #$34
    sta $01
```

```
    lda #<bitmap
    sta $fe
    lda #>bitmap
    sta $ff

    ldy #$00
-loop
    tya
    and #$01
    tax
    lda +bytes,x
    sta ($fe),y
    iny
    bne -loop

    inc $ff
    lda $ff
    and #$1f
    cmp #$18
    bne -loop

;clear fli-bug area...
    lda #<bitmap+$10
    sta $fe
    lda #>bitmap+$01
    sta $ff

    ldx #17
-loop1
    ldy #$00
    tya
-loop2
    sta ($fe),y
    iny
    cpy #8*12
    bne -loop2
    lda $fe
    clc
    adc #$40
    sta $fe
    lda $ff
    adc #$01
    sta $ff
    dex
    bpl -loop1

    lda #$36
    sta $01
```

```
;remove lower pixels...
lda #<bitmap+$20
sta $fe
lda #>bitmap+$0f
sta $ff

ldy #$00
tya
-loop2
;sta ($fe),y
iny
iny
iny
iny
iny
iny
iny
iny
iny
bne -loop2

lda #$36
sta $01

;fill screens with black...

lda #<screens
sta $fe
lda #>screens
sta $ff
-loop
lda #$00
sta ($fe),y
iny
bne -loop
inc $ff
lda $ff
sec
sbc #>screens
cmp #$20
bne -loop
rts

+bytes
.by %00010010
.by %01001000
```

```
;-----  
----  
makewobbler  
  
    sei  
    lda #$34  
    sta $01  
  
    lda #<wobbler  
    sta $fe  
    lda #>wobbler  
    sta $ff  
  
-yloop  
  
    lda +y  
    asl  
    tax  
    lda +ststarts,x  
    sec  
    sbc #16  
    sta +st+1  
    lda +ststarts+1,x  
    sbc #0  
    sta +st+2  
  
    lda $fe  
    clc  
    adc #$99  
    sta +jw+1  
    lda $ff  
    adc #$00  
    sta +jw+2  
  
    ldy #0  
-  
    lda wobblersrc1,y  
    sta ($fe),y  
    iny  
    cpy #9  
    bne -  
    jsr yaddfe  
  
    inc +lx+1  
    inc +cx+1  
  
    lda #0
```

```
    sta +x

-xloop

    ldy #0
-
    lda wobblersrc2,y
    sta ($fe),y
    iny
    cpy #6
    bne -
    jsr yaddfe
    lda +la+1
    clc
    adc #20
    sta +la+1
    lda +la+2
    adc #0
    sta +la+2

    lda +st+1
    clc
    adc #1
    sta +st+1
    lda +st+2
    adc #0
    sta +st+2

    inc +x
    lda +x
    cmp #width
    bne -xloop

    inc +y
    lda +y
    cmp #9
    beq +addjmp1
    cmp #36
    beq +addjmp2
    cmp #40
    beq +addjmp3
    cmp #44
    beq +addjmp4
    cmp #48
    beq +addjmp5
    cmp #52
    beq +addjmp6
    cmp #56
    beq +addjmp7
    cmp #60
```

```
    beq +addjmp8
    cmp #64
    beq +addjmp9
    cmp #height
    beq +end
    jmp -yloop

+addjmp1
    ldx #0
    jmp +addjmp
+addjmp2
    ldx #3
    jmp +addjmp
+addjmp3
    ldx #6
    jmp +addjmp
+addjmp4
    ldx #9
    jmp +addjmp
+addjmp5
    ldx #12
    jmp +addjmp
+addjmp6
    ldx #15
    jmp +addjmp
+addjmp7
    ldx #18
    jmp +addjmp
+addjmp8
    ldx #21
    jmp +addjmp
+addjmp9
    ldx #24
    jmp +addjmp

+addjmp
    ldy #0
-
    lda +jmpsrc,x
    sta ($fe),y
    inx
    iny
    cpy #3
    bne -

    dex
    dex
    lda +jmpsrc,x
    sta $fe
    lda +jmpsrc+1,x
```

```
    sta $ff

    jmp -yloop

+end

    ldy #0
    lda #$60
    sta ($fe),y

    lda #$36
    sta $01

    rts

+x .by 0
+y .by 0

wobblersrc1
+lx ldx wp
+cx cpx wpo
    bne +
+jw jmp wobbler+$99
+

wobblersrc2

+la lda anim,x
+st sta $e0b8
+jmpsrc
    jmp $d040
    jmp $e202
    jmp $e602
    jmp $ea02
    jmp $ee02
    jmp $f202
    jmp $f602
    jmp $fa02
    jmp $fe02

+ststarts
    .wo $e0b8,$e4b8,$e8b8,$ecb8,$f0b8,$f4b8,$f8b8,$fcb8
    .wo $e0e0,$e4e0,$e8e0,$ece0,$f0e0,$f4e0,$f8e0,$fce0
    .wo $e108,$e508,$e908,$ed08,$f108,$f508,$f908,$fd08
    .wo $e130,$e530,$e930,$ed30,$f130,$f530,$f930,$fd30
    .wo $e158,$e558,$e958,$ed58,$f158,$f558,$f958,$fd58
    .wo $e180,$e580,$e980,$ed80,$f180,$f580,$f980,$fd80
    .wo $e1a8,$e5a8,$e9a8,$eda8,$f1a8,$f5a8,$f9a8,$fda8
    .wo $e1d0,$e5d0,$e9d0,$edd0,$f1d0,$f5d0,$f9d0,$fdd0
```

```
.wo $elf8,$e5f8,$e9f8,$edf8,$f1f8,$f5f8,$f9f8,$fdf8
;-----
----
makesiner

    lda #<siner
    sta $fe
    lda #>siner
    sta $ff

    ldx #height
-loop

    ldy #0
-
    lda sinersrc,y
    sta ($fe),y
    iny
    cpy #5
    bne -

    jsr yaddfe

    txa
    and #1
    ;beq +
    lda +a+1
    clc
    adc #$02
    sta +a+1
+

    inc +s+1
    dex
    bne -loop

    ldy #0
    lda #60
    sta ($fe),y

    jsr updatesiner

    rts

sinersrc

+a  lda sine2,y
+s  sta wp
```



```
updatesiner
```

```
    lda #<siner
    sta $fe
    lda #>siner
    sta $ff
```

```
    lda #0
    sta tmp
    sta tmp+1
```

```
    ldx #height
    ldy #1
```

```
-loop
```

```
    lda tmp+1
    sta ($fe),y
```

```
    lda tmp
    clc
    adc sinespread
    sta tmp
    lda tmp+1
    adc sinespread+1
    sta tmp+1
```

```
    tya
    clc
    adc #5
    tay
    lda $ff
    adc #0
    sta $ff
    dex
    bpl -loop
```

```
    rts
```

```
sinespread .wo 1
```

```
;-----  
----
```

```
yaddfe
```

```
    tya
    clc
    adc $fe
    sta $fe
    lda #0
```

```
adc $ff
sta $ff
rts
;-----
----
```

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

Permanent link:
https://codebase64.org/doku.php?id=base:2nd_pixel_fli_distorter

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

