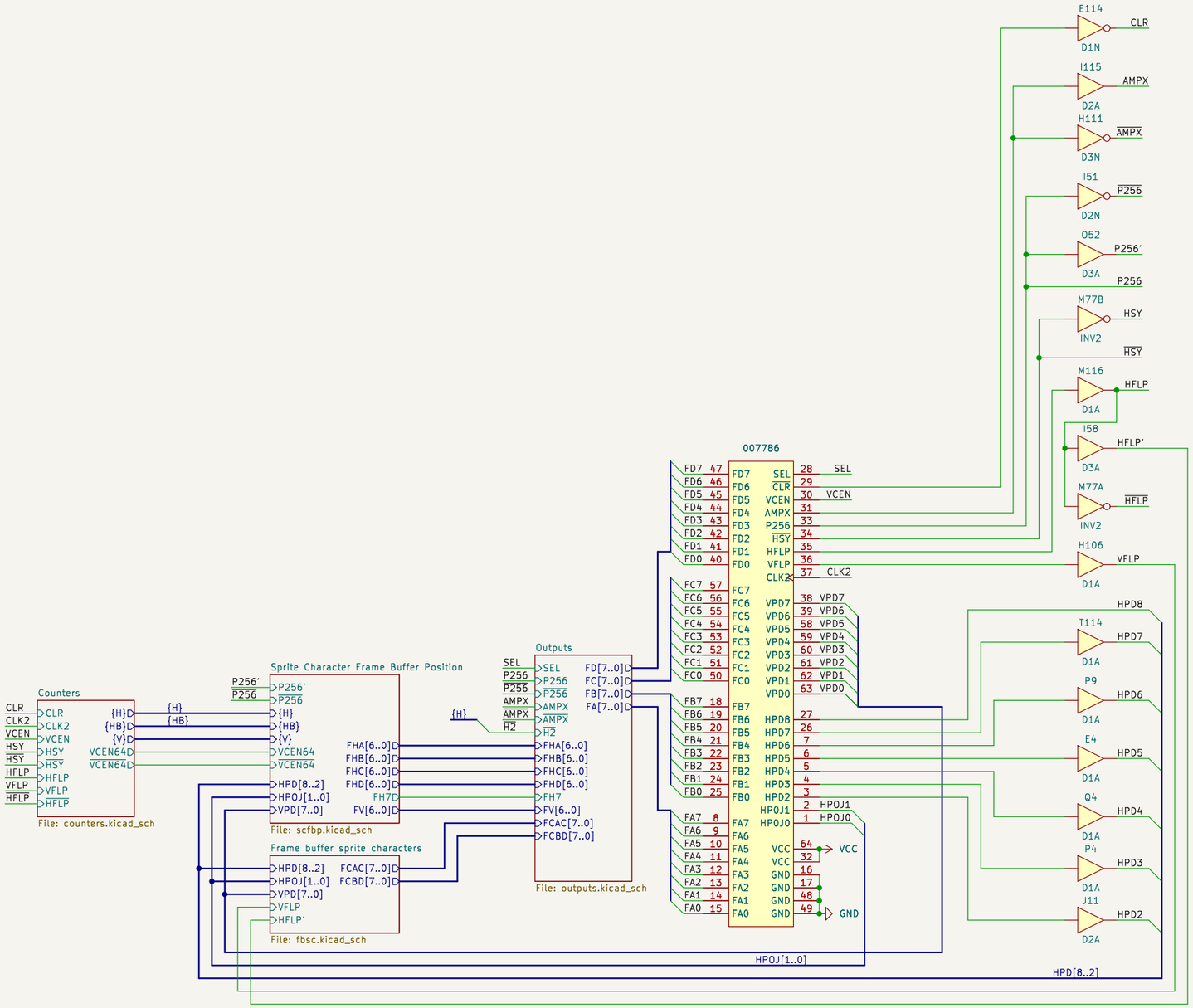


The 007786 is a sprite frame buffer controller.
It works in two modes:
– When SEL = 0, it generates addresses for the frame buffer.
– When SEL = 1, it parses sprite character data
It also handles refreshing of data in the DRAM memory.

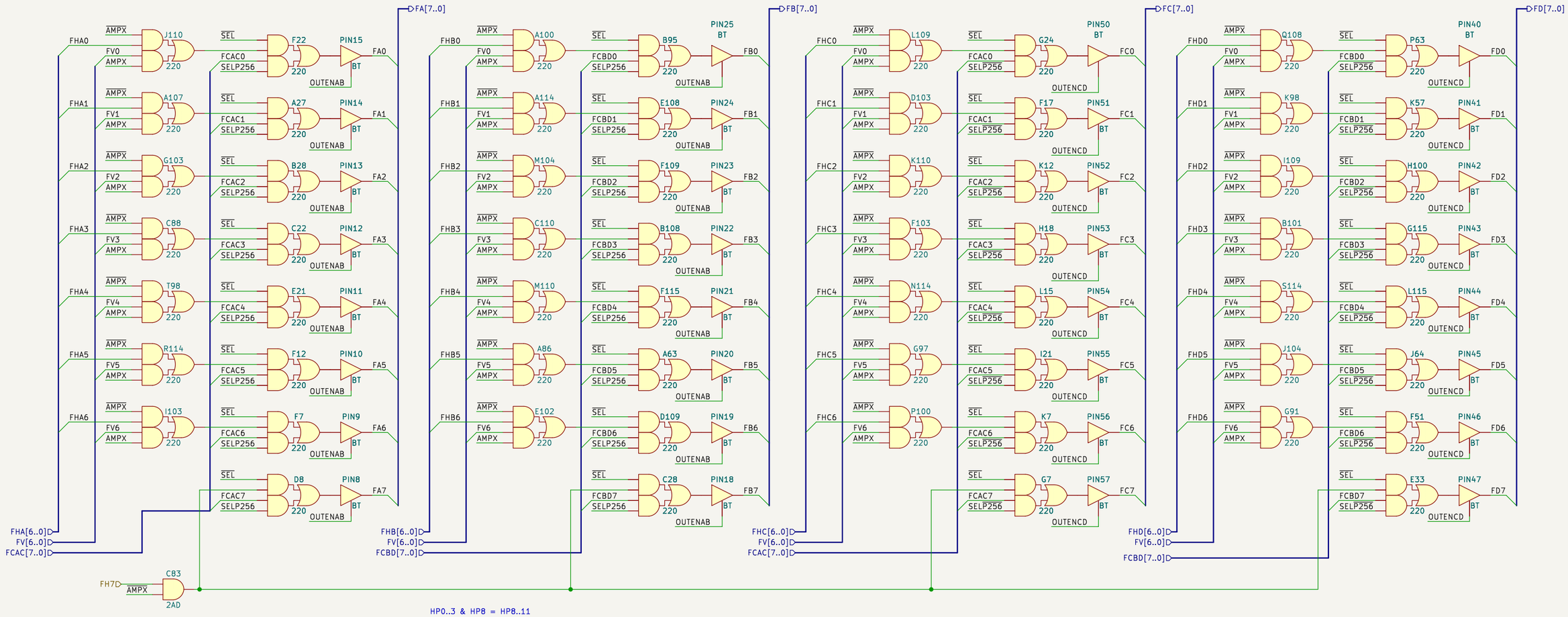
The 007786 uses an OKI 74V000 CMOS gate array
with 2400 unit cells.

Ulf Skutnabba, twitter: @skutis77

Sheet: /
File: 007786.kicad_sch

Title: *Konami 007786*

Size: A3	Date: 2024-01-28	**Rev:**
KiCad E.D.A. 8.0.8		Id: 1/5



Output buffer enable

Addressing mode, SEL = 0

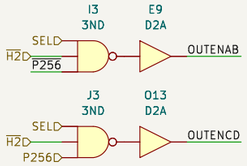
Outputs are enabled all the time when the 007786 is used for addressing sprite characters in the frame buffer, SEL = 0.

Sprite character mode, SEL = 1

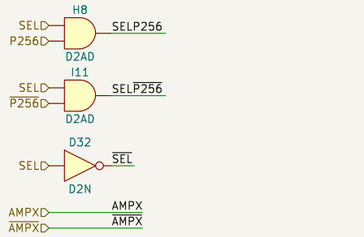
When used for writing sprite characters to the frame buffer it is enabled when H2 = 1 or for FA and FB outputs when P256 = 1. The corresponding for C and D outputs is when P256 = 0.

A and B output
OUTENAB =
~SEL | ~H2 | ~P256 =
~SEL | H2 | P256

OUTENCD =
~SEL | ~H2 | ~P256 =
~SEL | H2 | P256



SCM, P256 = 1, AB out enabled.



AMPX and AMPX selects between Row and Columns in the frame buffer DRAM.

Ulf Skutnabba, twitter: @skutis77

Sheet: /Outputs/
File: outputs.kicad_sch

Title: Konami 007786

Size: A3	Date: 2024-01-28	Rev:
KiCad E.D.A. 8.0.8		Id: 2/5

V1F is synchronized to $\overline{HSY}$ while the other vertical counters are synchronized to VCEN.

H16B is the same as H16 but shifted one half H8 period later (90 deg).

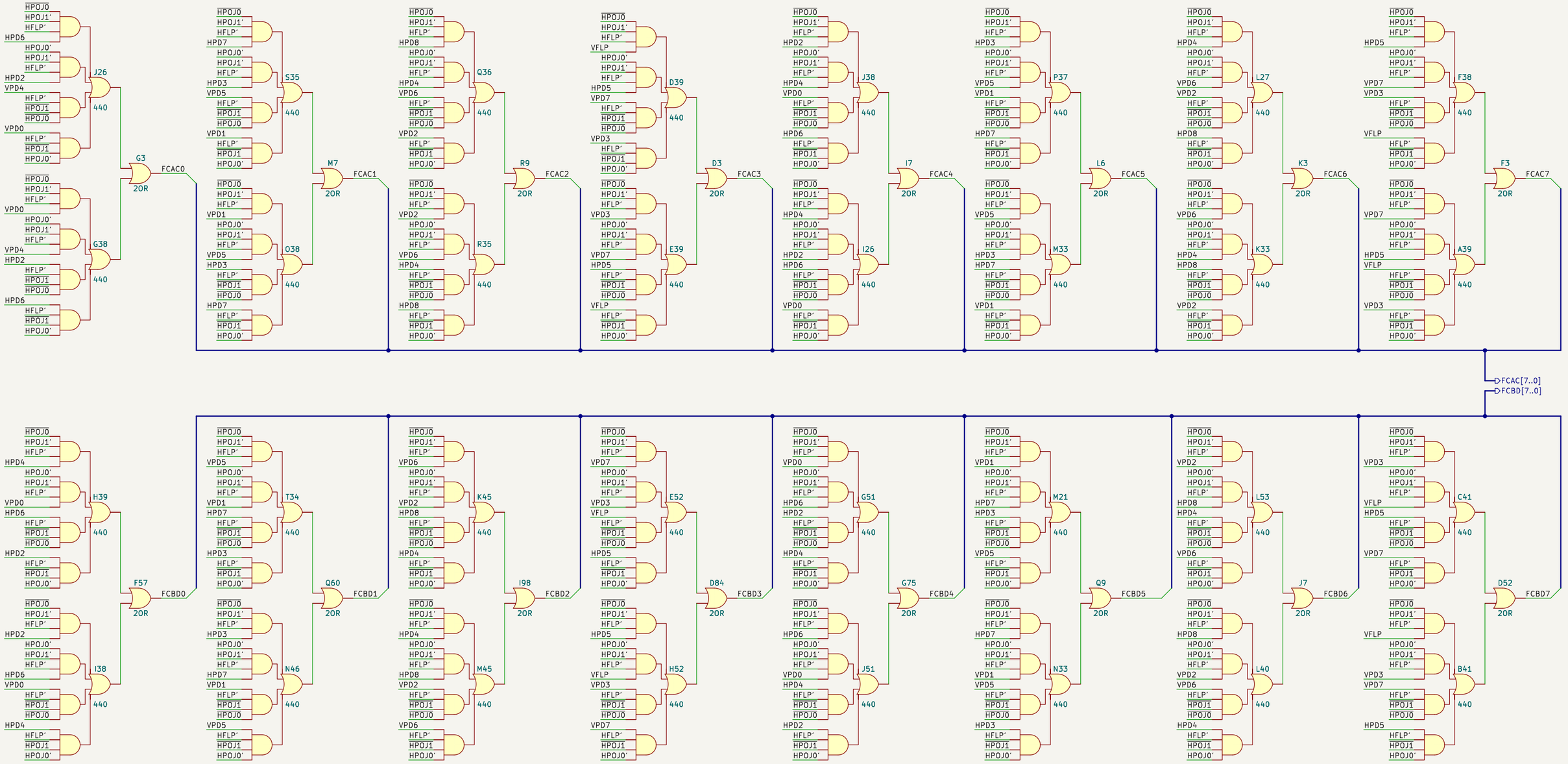
H8B = -H8

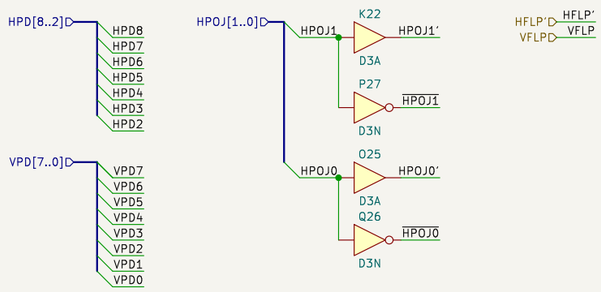
Ulf Skutnabba, twitter: @skutis77

Sheet: /Counters/
File: counters.kicad_sch

Title: Konami 007786

Size: A3	Date: 2024-01-28	**Rev:**
KiCad E.D.A. 8.0.8		Id: 3/5





Ulf Skutnabba, twitter: @skutis77

Sheet: /Frame buffer sprite characters/
File: fbsc.kicad_sch

Title: Konami 007786

Size: A3	Date: 2024-01-28	**Rev:**
KiCad E.D.A. 8.0.8		Id: 4/5

Horizontal Position

$\overline{VCEN64}$ active: Data is written to the frame buffer
VCEN64 active, four times pre frame: Data is refreshed in DRAM, every 4ms.

HP1 to HP4

HPD2 / $\overline{HP0}$ → S19 (EXR) → VCEN64, P256', P256', VCEN64, $\overline{H2}$, $\overline{P256}$, VCC' → R54 (330) → FHA0

HP4..7

HPD2 / $\overline{HP0}$ → O4 (2ND) + HPD3 → J16 (ENR) → VCEN64, P256', P256', VCEN64, H4, $\overline{P256}$, VCC' → A69 (330) → FHA1

HPD3 / HPD2 / $\overline{HP0}$ → T5 (3ND) + HPD4 → S13 (ENR) → VCEN64, P256', P256', VCEN64, H8, $\overline{P256}$, VCC' → Q75 (330) → FHA2

HPD4 / HPD3 / HPD2 / $\overline{HP0}$ → N9 (4ND) + HPD5 → G12 (ENR) → VCEN64, P256', P256', VCEN64, H16, $\overline{P256}$, VCC' → C74 (330) → FHA3

HPD4 / HPD5 / $\overline{HP0}$ / HPD2 / HPD3 → P20 (5ND) + HPD6 → P14 (ENR) → VCEN64, P256', P256', VCEN64, H32, $\overline{P256}$, VCC' → T87 (330) → FHA4

HPD3 / HPD2 / $\overline{HP0}$ / HPD4 / HPD5 / HPD6 → R27 (6ND) + HPD7 → S49 (ENR) → VCEN64, P256', P256', VCEN64, H64, $\overline{P256}$, VCC' → T76 (330) → FHA5

$\overline{HP0}$ / HPD2 / HPD3 / HPD4 / HPD5 / HPD6 / HPD7 → T25 (7ND) + HPD8 → T48 (ENR) → VCEN64, P256', P256', VCEN64, H128, $\overline{P256}$, VCC' → N92 (330) → FHA6

FHA[6..0]

VPD0 → VCEN64, P256', P256', VCEN64, H256, $\overline{P256}$, VCC' → D74 (330) → FH7

HP2 to HP5

HPD2 / HPOJ1 → L10 (EXR) → VCEN64, P256', P256', VCEN64, $\overline{H2}$, $\overline{P256}$, VCC' → L66 (330) → FHB0

HPD2 / HPOJ1 → L3 (2ND) + HPD3 → H3 (ENR) → VCEN64, P256', P256', VCEN64, H4, $\overline{P256}$, VCC' → B71 (330) → FHB1

HPD3 / HPD2 / HPOJ1 → O8 (3ND) + HPD4 → Q14 (ENR) → VCEN64, P256', P256', VCEN64, H8, $\overline{P256}$, VCC' → Q74 (330) → FHB2

HPD4 / HPD3 / HPD2 / HPOJ1 → N14 (4ND) + HPD5 → H13 (ENR) → VCEN64, P256', P256', VCEN64, H16, $\overline{P256}$, VCC' → C65 (330) → FHB3

HPD4 / HPD5 / HPOJ1 / HPD2 / HPD3 → R20 (5ND) + HPD6 → T18 (ENR) → VCEN64, P256', P256', VCEN64, H32, $\overline{P256}$, VCC' → Q65 (330) → FHB4

HPD3 / HPD2 / HPOJ1 / HPD4 / HPD5 / HPD6 → N25 (6ND) + HPD7 → T11 (ENR) → VCEN64, P256', P256', VCEN64, H64, $\overline{P256}$, VCC' → S70 (330) → FHB5

HPOJ1 / HPD2 / HPD3 / HPD4 / HPD5 / HPD6 / HPD7 → S26 (7ND) + HPD8 → Q20 (ENR) → VCEN64, P256', P256', VCEN64, H128, $\overline{P256}$, VCC' → R83 (330) → FHB6

FHB[6..0]

HP3 to HP6

HPD2 / HP3 → O19 (EXR) → VCEN64, P256', P256', VCEN64, $\overline{H2}$, $\overline{P256}$, VCC' → O65 (330) → FHC0

HPD2 / HP3 → S4 (2ND) + HPD3 → M11 (ENR) → VCEN64, P256', P256', VCEN64, H4, $\overline{P256}$, VCC' → F73 (330) → FHC1

HPD3 / HPD2 / HP3 → S8 (3ND) + HPD4 → R14 (ENR) → VCEN64, P256', P256', VCEN64, H8, $\overline{P256}$, VCC' → P80 (330) → FHC2

HPD4 / HPD3 / HPD2 / HP3 → M3 (4ND) + HPD5 → I16 (ENR) → VCEN64, P256', P256', VCEN64, H16, $\overline{P256}$, VPD4, H32BF → H65 (330) → FHC3

HPD4 / HPD5 / HP3 / HPD2 / HPD3 → H24 (5ND) + HPD6 → L21 (ENR) → VCEN64, P256', P256', VCEN64, H32, $\overline{P256}$, VCC' → S80 (330) → FHC4

HPD3 / HPD2 / HP3 / HPD4 / HPD5 / HPD6 → G30 (6ND) + HPD7 → N19 (ENR) → VCEN64, P256', P256', VCEN64, H64, $\overline{P256}$, VCC' → T65 (330) → FHC5

HP3 / HPD2 / HPD3 / HPD4 / HPD5 / HPD6 / HPD7 → H31 (7ND) + HPD8 → M16 (ENR) → VCEN64, P256', P256', VCEN64, H128, $\overline{P256}$, VCC' → P70 (330) → FHC6

FHC[6..0]

HP4 to HP7

HPD2 → VCEN64, P256', P256', VCEN64, $\overline{H2}$, $\overline{P256}$, VCC' → T55 (330) → FHD0

HPD3 → VCEN64, P256', P256', VCEN64, H4, $\overline{P256}$, VCC' → M68 (330) → FHD1

HPD4 → VCEN64, P256', P256', VCEN64, H8, $\overline{P256}$, VCC' → R73 (330) → FHD2

HPD5 → VCEN64, P256', P256', VCEN64, H16, $\overline{P256}$, VCC' → B80 (330) → FHD3

HPD6 → VCEN64, P256', P256', VCEN64, H32, $\overline{P256}$, VCC' → S90 (330) → FHD4

HPD7 → VCEN64, P256', P256', VCEN64, H64, $\overline{P256}$, VCC' → R63 (330) → FHD5

HPD8 → VCEN64, P256', P256', VCEN64, H128, $\overline{P256}$, VCC' → Q84 (330) → FHD6

FHD[6..0]

Vertical Position

P256' → R100 (220) → FV0

P256' / VPD2 / V4F / $\overline{P256}$ → K92 (220) → FV1

P256' / VPD3 / V8F / $\overline{P256}$ → H94 (220) → FV2

P256' / VPD4 / V16F / $\overline{P256}$ → I70 (220) → FV3

P256' / VPD5 / V32F / $\overline{P256}$ → M92 (220) → FV4

P256' / VPD6 / V64F / $\overline{P256}$ → K17 (220) → FV5

P256' / VPD7 / V128F / $\overline{P256}$ → D68 (220) → FV6

FV[6..0]

P256' active: Write addressing
$\overline{P256}$ active: Read addressing

HPOJ[1..0] → HPOJ1, HPOJ0

HPOJ1 / HPOJ0 → R4 (2OR) → $\overline{HP0}$ — HP1 to HP3

HPOJ1 / HPOJ0 → N4 (2AD) → HP3

HPD[8..2] → HPD2, HPD3, HPD4, HPD5, HPD6, HPD7, HPD8

VPD[7..0] → VPD0, VPD1, VPD2, VPD3, VPD4, VPD5, VPD6, VPD7

{H} → H2, $\overline{H2}$, H4, $\overline{H4}$, H4F, H8, H16, $\overline{H16}$, H32, H64, H128, H256

{HB} → H8BF, H16BF, H32BF, H64BF, H128BF, H256BF

{V} → V1F, V2F, V4F, V8F, V16F, V32F, V64F, V128F

VCC → K62 (HFX) → VCC'

P256'D → P256'
$\overline{P256}$D → $\overline{P256}$
VCEN64D → $\overline{VCEN64}$
$\overline{VCEN64}$D → VCEN64

Ulf Skutnabba, twitter: @skutis77

Sheet: /Sprite Character Frame Buffer Position/
File: scfbp.kicad_sch

Title: Konami 007786

Size: A3	Date: 2024-01-28	**Rev:**
KiCad E.D.A. 8.0.8		Id: 5/5