Research on arcade systems

Konami's Ping Pong PCB Analysis

A Ping Pong arcade PCB was analyzed in order to convert the system to modern FPGA-based electronics. The full schematics were extracted, annotated and sensible signal names were attached. The function of the custom chips was derived from the surrounding discrete logic and measurements were taken to check assumptions on them. A small custom firmware was written to check other aspects of custom chips. The CPU bus contention was measured and modeled.

Acknowledgments

This work was sponsored by JOTEGO's patrons. The PCB was loaned by Twitter user @supercrow123. Special thanks to DJ Hardrich, Martin and atrac17 for their continuous support.

Board Overview

The board belong to the Konami series based on the 082, 083, 503 and other custom chips. Although it does not use all of them and the screen counters are exposed as discrete logic. Two DIP-28 footprints connect to two identical daughter boards with just three standard logic chips on them.

The graphics are only 2-bit per pixel and the color DAC uses 3 bits for red and green and only 2 bits for blue. Sound is generated by a single SN76489 chip. There is only one CPU, a Z80 and the CPU shares the RAM chip with the object table, which means that there will be bus contention when accessing it. All these features point to a low-budget project, particularly when compared to other such as Track & Field.

The PCB with the two 28-DIP daughter boards removed

CPU and Bus Contention

The critical signals are controlled in this way:

- Bus request: not exercised
- NMI: every 16 lines
- INT: once per frame, when entering V-blank
- WAIT: when the CPU tries to access the video memories but the hardware is reading them, the WAIT signal is asserted halting the CPU

The frequency is 3.07MHz, dttrrraddrived from a crystal clock at 18.43MHz, which is divided by three using two JK latches, and then by two by a D flip flop.

There is a watchdog and reset chip, the TA7900S, common to other Konami boards too. A jumper on the board allows to bypass the watchdog by closing it with solder.

Wait goes low if the GFX are driving the memories but the CPU also needs to access

CPU Bus Custom Chip

A custom chip at location D5 serves the following functions:

- External data bus line driver
- External bus selection line, derived from A15
- Arbiter between the CPU and the video hardware to divide access to RAM chips
- Write signal to bus devices

Graphic RAM is time-multiplexed at 1.5MHz between the CPU and the graphic chips.

System I/O

The board uses Konami's small edge connector (not JAMMA). There are three DIP switches, two of eight bits and a third one of 4 bits. However, the MSB of the third one is not connected to the data bus.

There are two test connectors to which the CPU can write. A couple of general signals (the GFX multiplexor select andV256) are also passed on through these connectors. The CPU cannot read from them.

Sound

A single SN76489 chip is used, unlike other systems of the same age that used several sound sources. There are no programmable filters or sound enable relays. The power amplifier used is the LA4460, common to other Konami games of the era. Data is written to the chip in two steps, first a byte is written to a latch and then a second write will make the SN chip read the latch.

Graphics

There are two graphic layers: the characters and the objects. There is no scroll layer. The objects are drawn on top the characters. The custom chip 503 produces a signal to distinguish blank object pixels. Based on this signal either the character pixel or the object pixel is chosen and passed to the video DAC.

There are three palette PROMs, one for characters, one for objects and a final one to drive the flash DAC. The bus connection from the object PROM output to the pixel multiplexer is reversed, probably by mistake: the MSB is connected to the LSB and so on.

The character layer is made of 2-bit pixels, which makes the reading circuitry simpler. Nonetheless, there are 6 bits to select different 4-color palettes so the amount of color in the screen doesn't seem as poor as one could expect from 2-bit pixels. There is no scroll and no global flip signal but the character can be flipped horizontally and vertically.

The object layer can draw a total of 24 objects, each object data is made of four bytes. The object table occupies the first 24x4 byte locations of the RAM. However, due to what seems to be another mistake, one of the bytes is kept during the time equivalent to 16 pixels stored in a latch. This means that the rendering hardware gets that byte from the previous object in the table and not the current one. The software works around this problem by writing that byte offset by one in memory, but the first object in the table cannot be recovered and is lost due to this problem.

The object table is read in the following order:

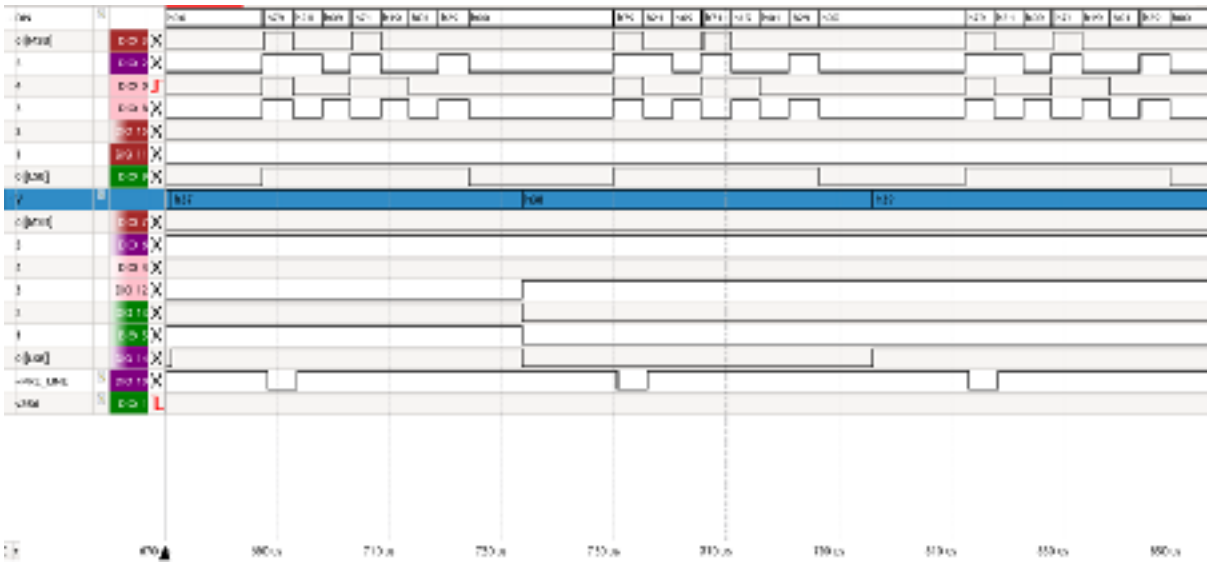
1. objects from 8 to 15
2. objects from 0 to 7
3. finally, from 16 to 23

There are two custom chips for object drawing:

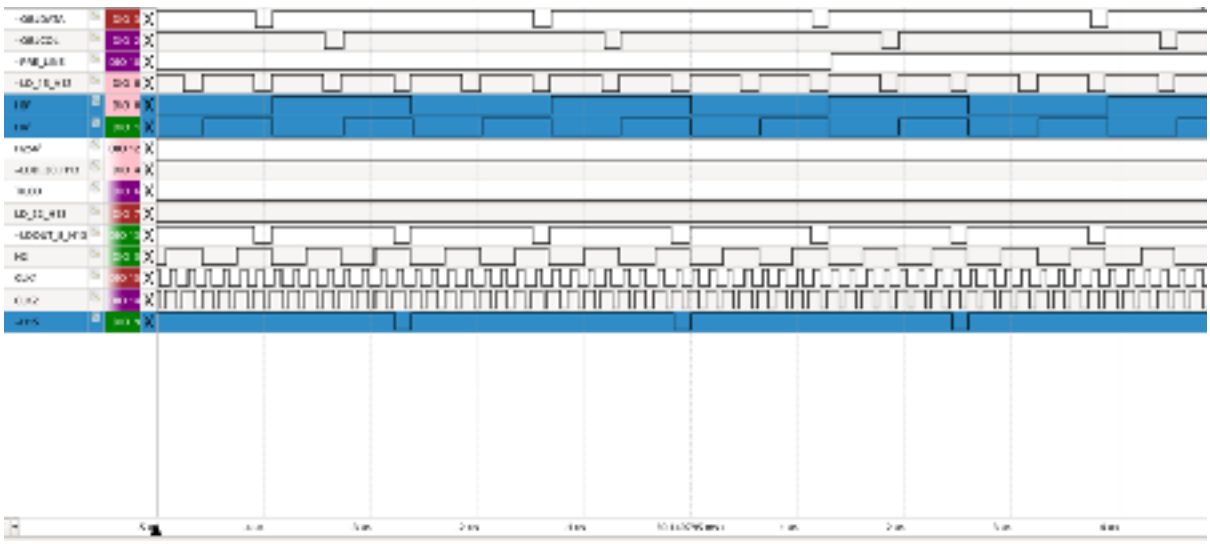
- Konami's 503 is used to identify when an object must be drawn

- Konami's 502 controls the double line buffer

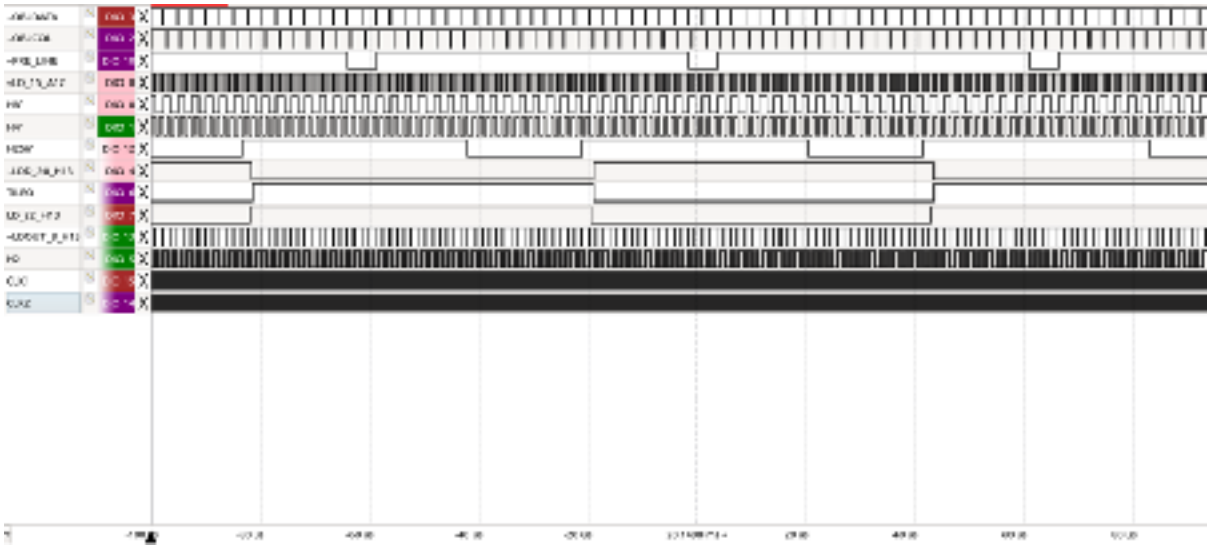
Objects drawn first will hide objects drawn later due to the way the 502 logic works. The 503 also stores the vertical and horizontal flip values and applies it to the lines driving the object graphics ROM.



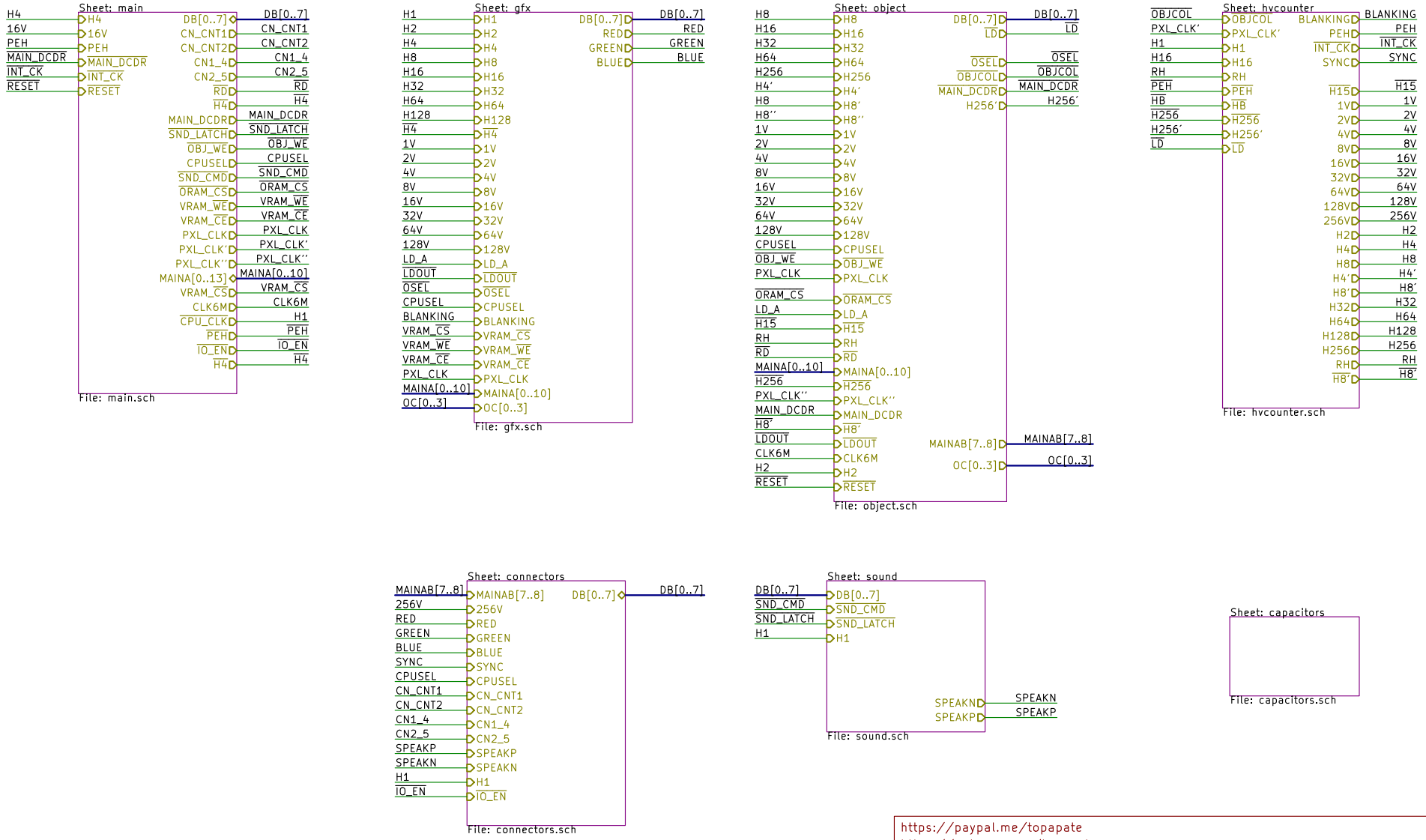
Custom chip 503 capture. ~PRE_LINE indicates an active object transfer



Clock signals controlling the object engine. ~H15 is the wrong one mentioned



Custom chip 502: double line buffer controller



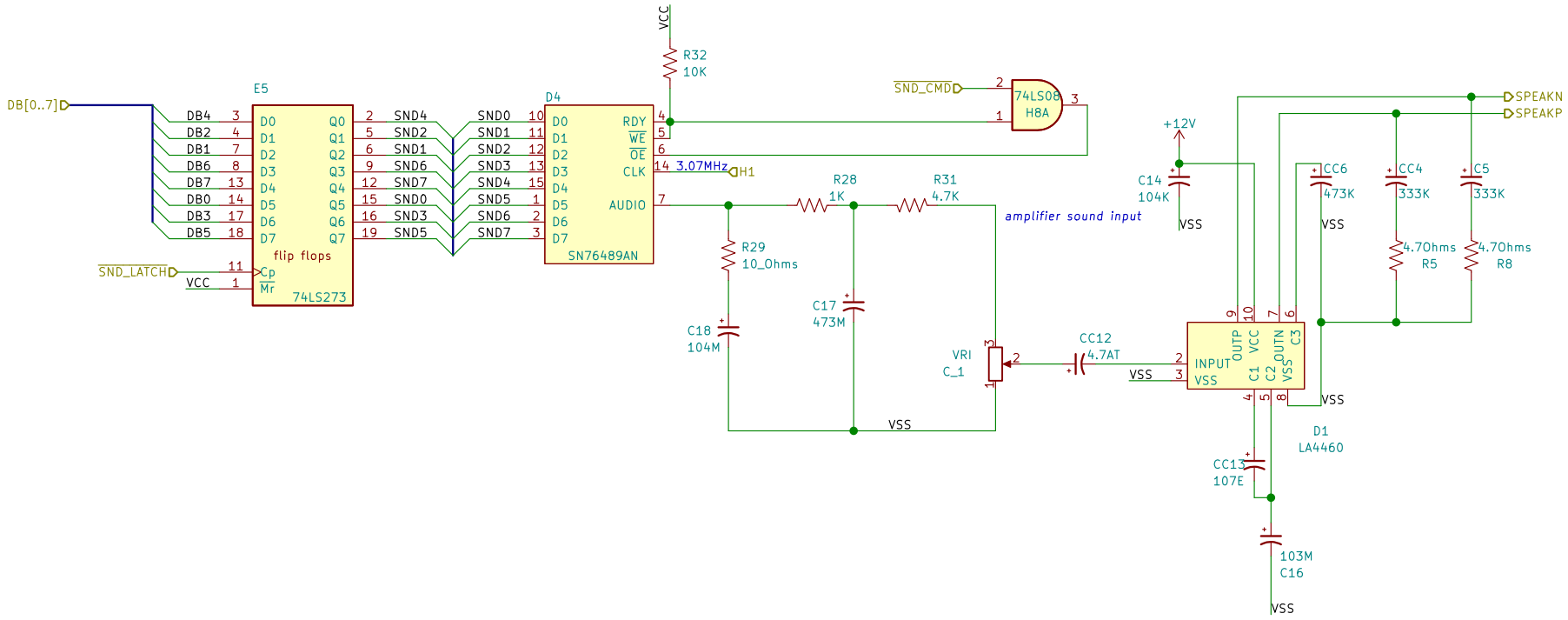
https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

Sheet: /
File: pingpong.sch

GX555

Title: Ping Pong

Size: A4	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 1/11

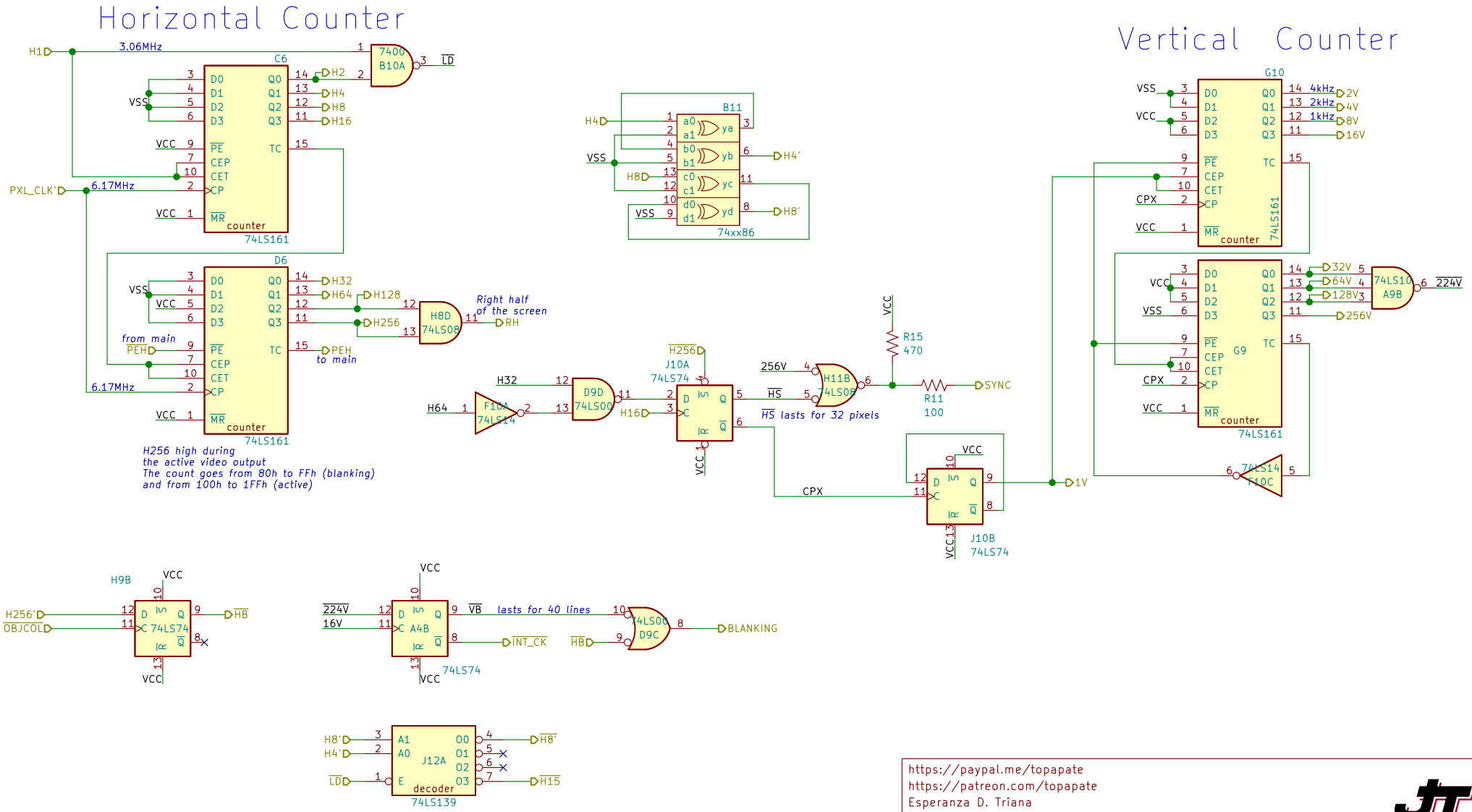


https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

Sheet: /sound/
File: sound.sch

Title: Sound generator

Size: A4	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 2/11

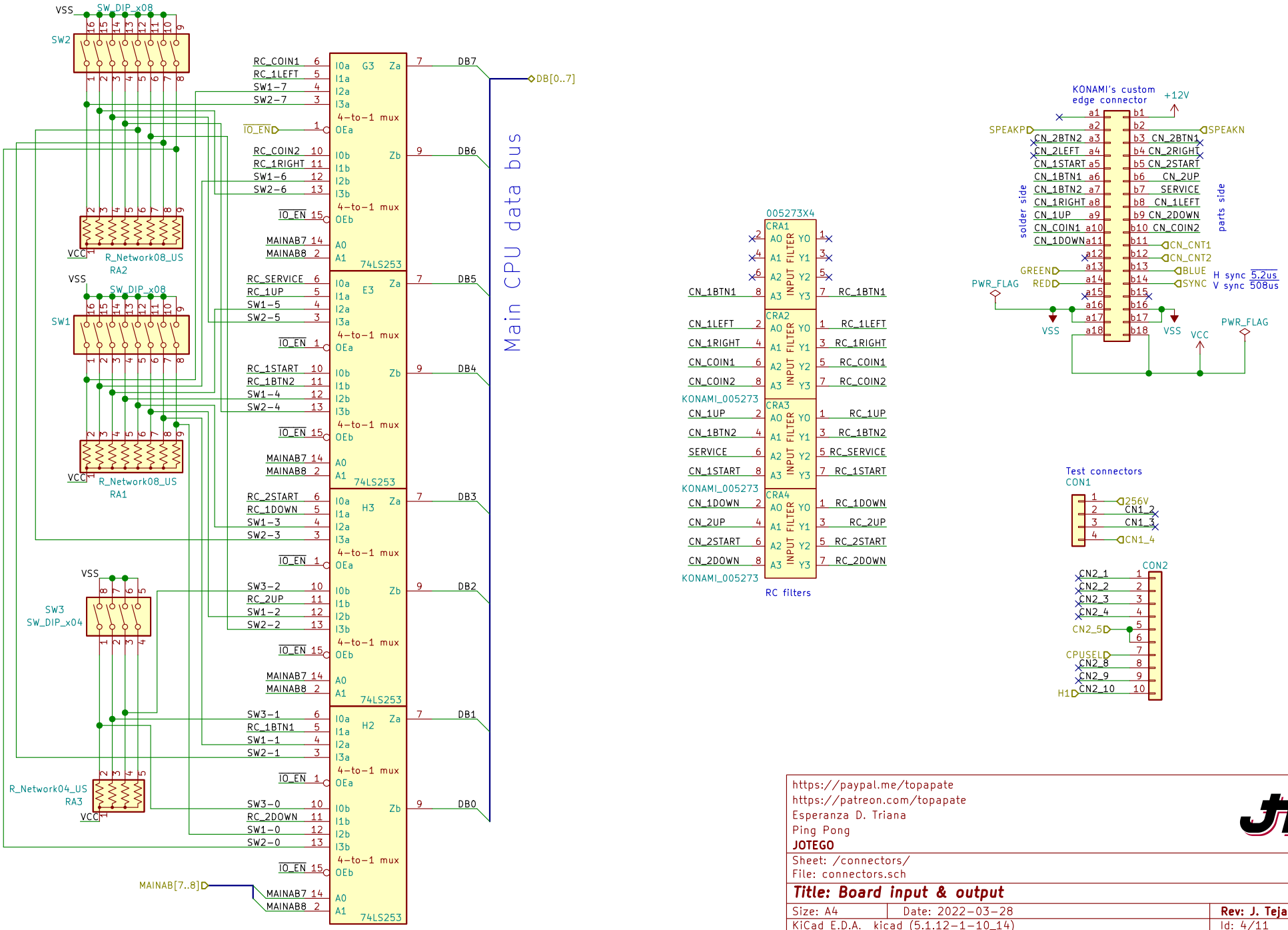


https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

Sheet: /hvcounter/
File: hvcounter.sch

Title: Horizontal and vertical counters

Size: A4	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 3/11

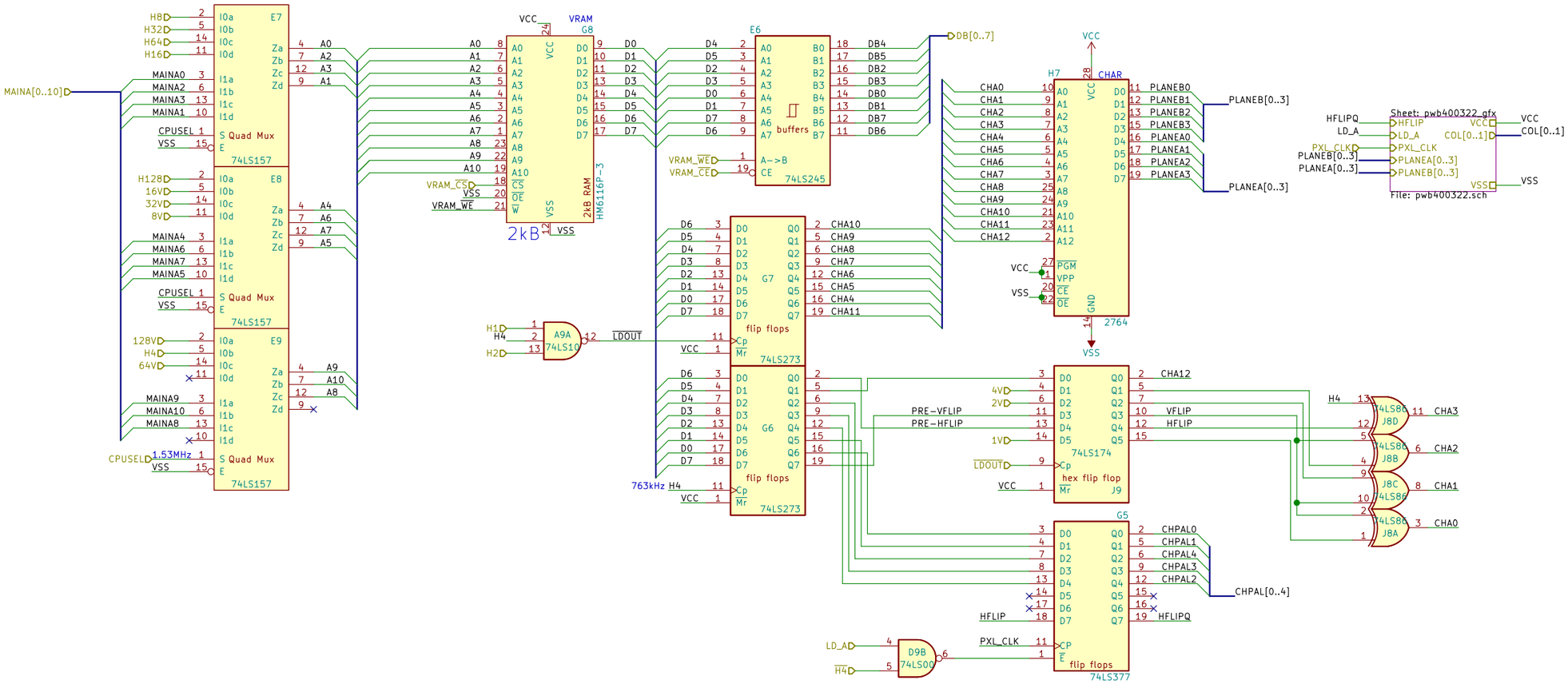


https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

Sheet: /connectors/
File: connectors.sch

Title: Board input & output

Size: A4	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 4/11

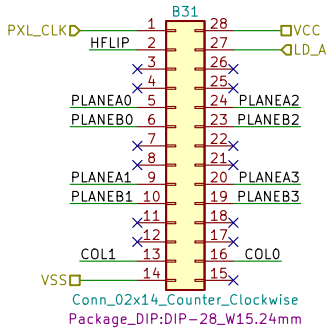


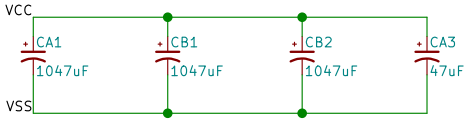
https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

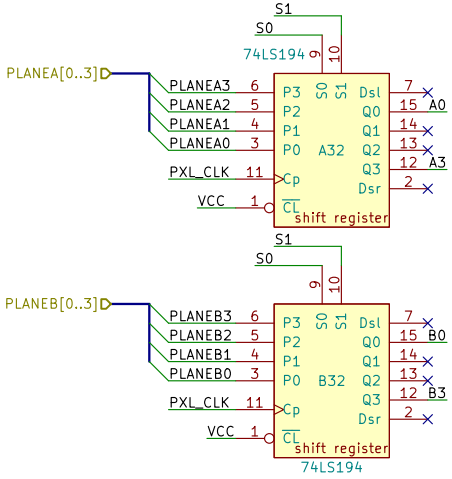
Sheet: /gfx/
File: gfx.sch

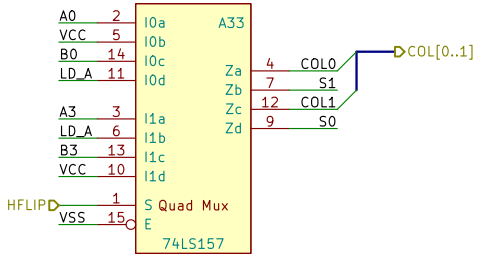
Title: Character generator

Size: A3	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 5/11









This is a small board soldered to a DIP-28 footprint. It appears twice in the design

PWB-400322

https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

Sheet: /gfx/pwb400322_gfx/
File: pwb400322.sch

Title: Pixel shift register

Size: A4	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 6/11

MAIN PROGRAM ROM

A5 Z80CPU
A6 27128
A7 27128
C4 74LS174
D5 KONAMI_UK
A_2 74LS368
A2 74LS107
C3 74LS138
B4 74xx74
C2 74xx04
74LS74
74LS139
74LS32
74LS00
74LS08
74LS07
74LS14
F100
JP1 Jumper
TR1 2SC2320
TR2 2SD400
TR3 2SD400
TR4 2SC2320
DD1 10DI
DD2 10DI
D8A 74LS32
CA2 47uF
CC1 475T
CC3 106E_16V
C54 102C
R35 4.7k
R34 4.7k
R9 47K
R10 1.2K
R12 120
R7 10K
R4 6.8K
R2 150
R3 4.7K
R6 30K
R1 100K
R25 51K
R17 100K
R16 120
R30 4.7K
18432.00 KHz
3.07MHz
6.14MHz
18.43MHz
1.53MHz
3.06MHz

https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

Sheet: /main/
File: main.sch

Title: CPU & address decoders

Size: A3	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 7/11

OBJRAM

MAINA[0..10]

192kHz H16
781kHz H4
385kHz H8
96kHz H32

74LS157 (S Quad Mux) — MAINA2, MAINA0, MAINA1, MAINA3 — CPUSEL, VSS

16.1kHz H256
47.6kHz H64
16.1kHz RH

74LS157 (S Quad Mux) — MAINA6, MAINA4, MAINA5, MAINA7 — CPUSEL, VSS

VSS

74LS157 (S Quad Mux) — MAINA10, MAINA9, RD, MAINA8 — CPUSEL, VSS

1.54MHz

74LS157 (S Quad Mux)

VCC

A8 HM6116P-3 (2kB RAM, OBJECT TABLE)

AA0 AA1 AA2 AA3 AA4 AA5 AA6 AA7 AA8 AA9 AA10 OEB OBJ_WE VSS

24 objects read
in order from
the first 24x4 bytes

DA0 DA1 DA2 DA3 DA4 DA5 DA6 DA7

A10 74LS245 (buffers, A->B)

OBJ_WE1 ORAM_CS

DB[0..7]

DB0 DB1 DB2 DB3 DB4 DB5 DB6 DB7

DA0 DA1 DA2 DA3 DA4 DA5 DA6 DA7

H15 VCC

E11 74LS273 (flip flops)

ATTR0 ATTR1 ATTR2 ATTR3 ATTR4 ATTR5 ATTR6 ATTR7

PRE_LINE HFLIP' PXL_CLK OBJCOL

74LS377 (flip flops) G11

LINECS HFLIP' PAL4 PAL0 PAL1 PAL2 PAL3

COL[0..1]

J11 TBP24S10N (PROM, Color PROM)

COL0 COL1 PAL0 PAL1 PAL2 PAL3 PAL4 VSS

IC[0..3]

IC3 IC2 IC1 IC0

E10

ATTR6 ATTR7 MAINA8 MAINA7 H8' VSS

DN6 (hflip) DN7 (vflip) MAINAB8 MAINAB7

MAINAB[7..8]

74LS367

DA7 DA5 DA3 DA1 DA0 DA2 DA4 DA6

A11 74LS374 (3-state flip flops)

H8' H8''

DN7 DN6 DN5 DN4 DN3 DN2 DN1 DN0

A12 V/HFLIP XOR ed

128V 64V 32V 16V 8V 4V 2V 1V

OBJV8 OBJH8 OBJH4 OBJV4 OBJV2 OBJV1 OBJDATA OBJCOL HFLIP LINE_CS H8 (input) H4 LD

17383kHz H8'
16766kHz H4'

KONAMI_503

OBJ5 OBJ4 OBJ3 OBJ2 OBJ1 OBJ0 OBJDATA OBJCOL HFLIP PRE_LINE

C12 2764 (8kB)

OBJ0 OBJ1 OBJ2 OBJ3 OBJ4 OBJ5 OBJ6 OBJ7 OBJ8 OBJ9 OBJ10 OBJ11 OBJ12

PGM VPP CE OE

OBJD0 OBJD1 OBJD2 OBJD3 OBJD4 OBJD5 OBJD6 OBJD7

OBJD[0..7]

C11 27128 (16kB) ONLY FOOTPRINT

OBJ0 OBJ1 OBJ2 OBJ3 OBJ4 OBJ5 OBJ6 OBJ7 OBJ8 OBJ9 OBJ10 OBJ11 OBJ12 OBJ13

OBJD0 OBJD1 OBJD2 OBJD3 OBJD4 OBJD5 OBJD6 OBJD7

DA6 DA2 DA3 DA4 DA5 DA0 DA1 DA7

C13 74LS273 (flip flops)

OBJ12 OBJ8 OBJ9 OBJ10 OBJ11 OBJ6 OBJ7 OBJ13

OBJDATA

H9A 74LS74

ATTR5 OE_OF OE_M

Sheet: pwb400322
File: pwb400322.sch

OBJD[4..7] OBJD[0..3] PXL_CLK HFLIP' LD_A

PLANEA[0..3] PLANEB[0..3] PXL_CLK HFLIP LD_A

VCC COL[0..1] VSS

Sheet: linebuffer
File: linebuffer.sch

PXL_CLK'' DA[0..7] LINECS H2 LD OBJCOL MAIN_DCDR H8' LDOUT H256 CLK6M IC[0..3] RESET

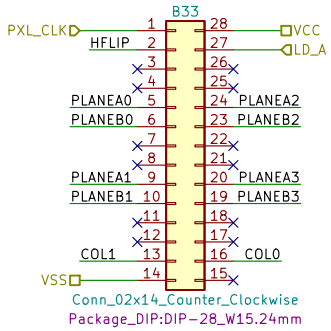
OC[0..3] OSEL H256 H8'' LD_A H256' MAIN_DCDR

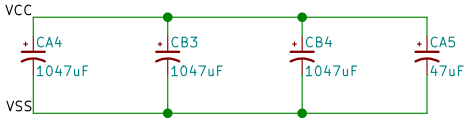
https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

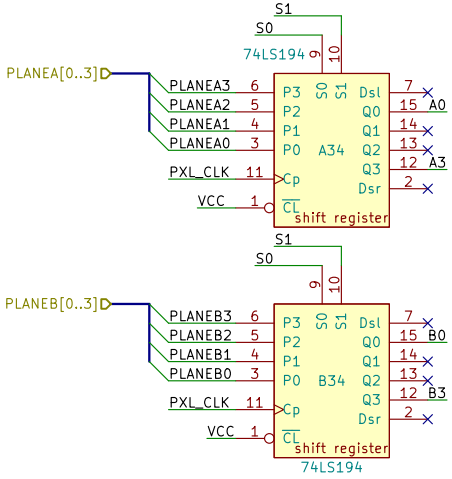
Sheet: /object/
File: object.sch

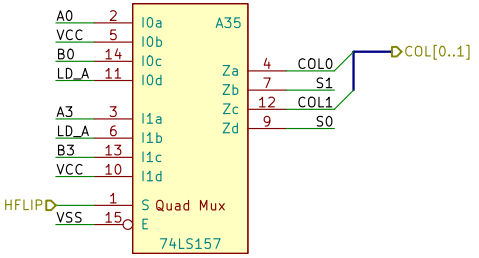
Title: Object generator

Size: A3	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 8/11









This is a small board soldered to a DIP-28 footprint. It appears twice in the design

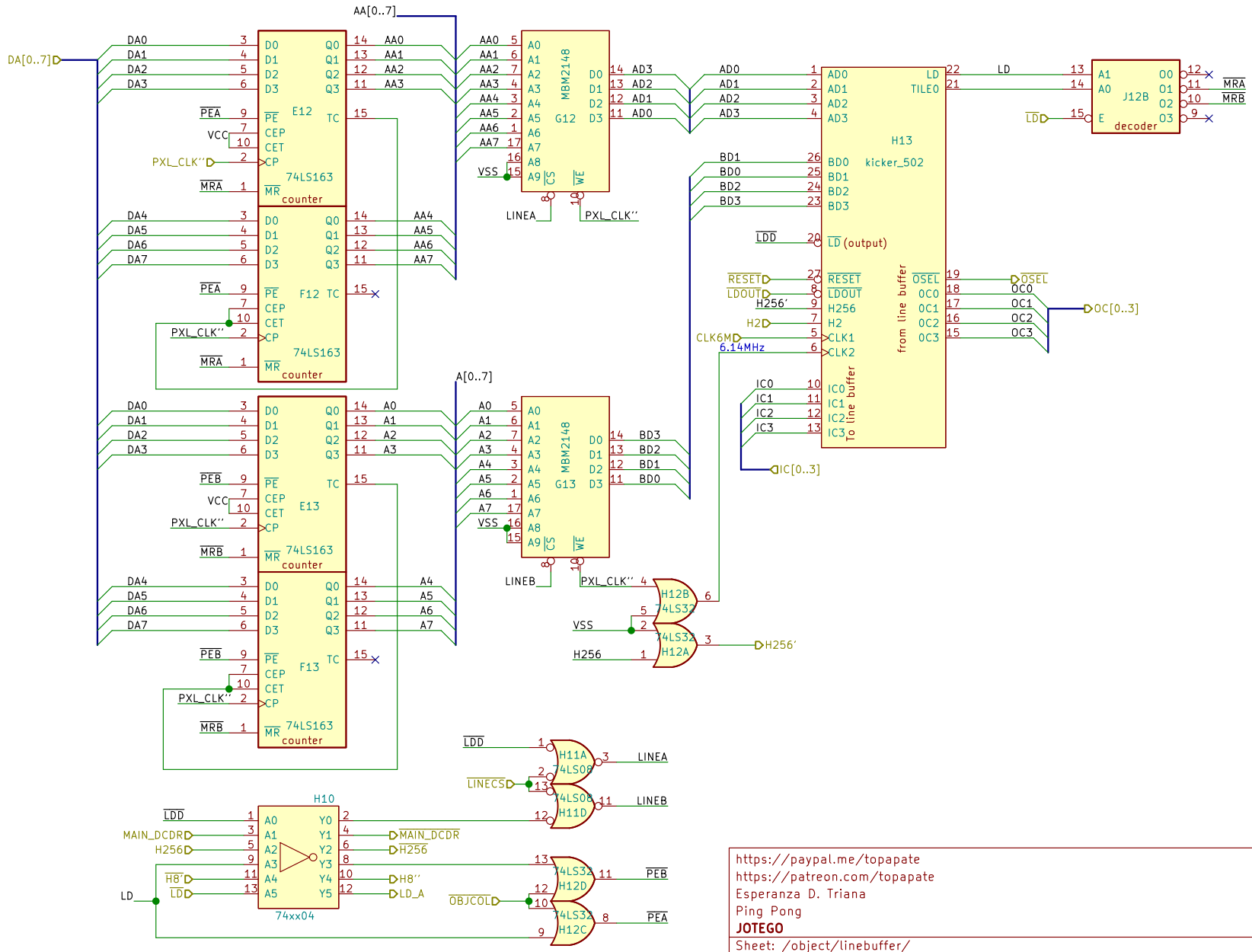
PWB-400322

https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

Sheet: /object/pwb400322/
File: pwb400322.sch

Title: Pixel shift register

Size: A4	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 9/11

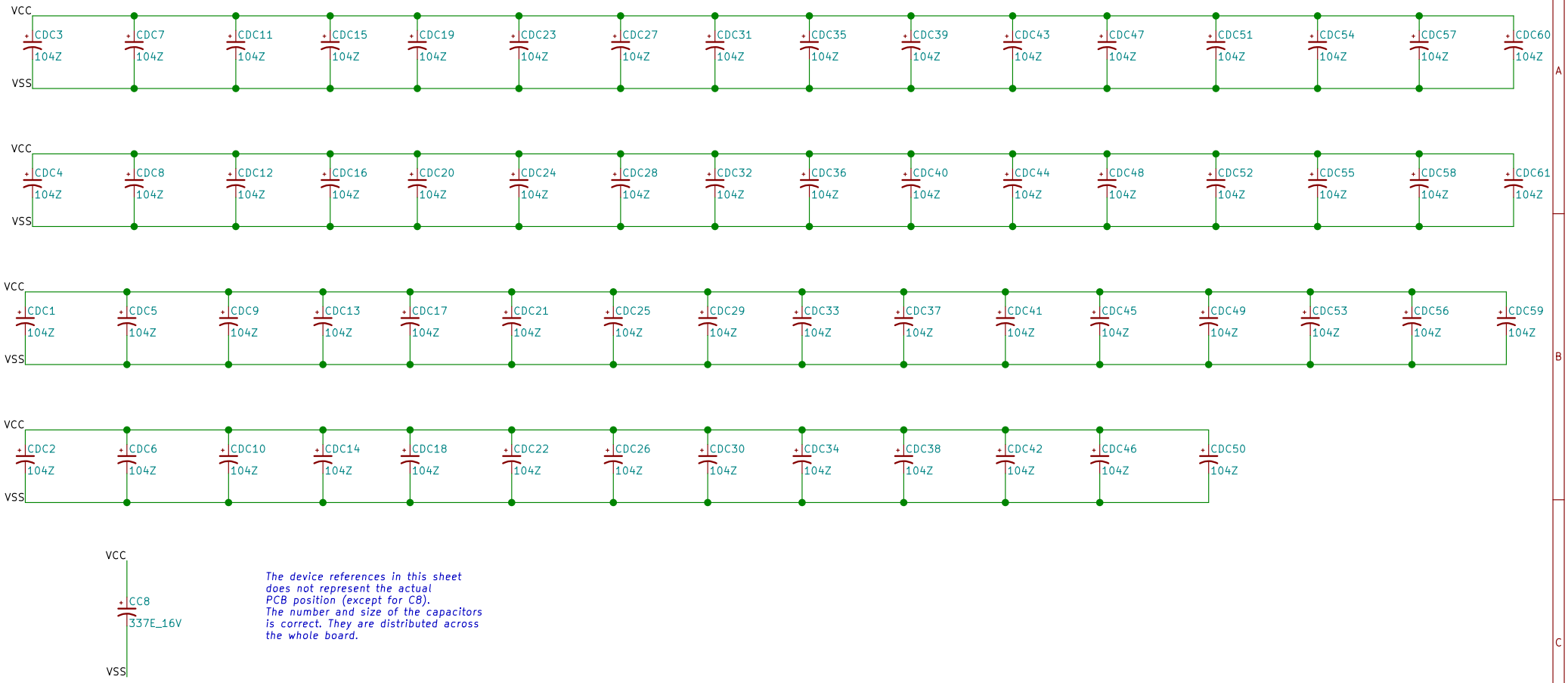


https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

Sheet: /object/linebuffer/
File: linebuffer.sch

Title: Doube line buffer

Size: A4	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 10/11



The device references in this sheet does not represent the actual PCB position (except for C8). The number and size of the capacitors is correct. They are distributed across the whole board.

https://paypal.me/topapate
https://patreon.com/topapate
Esperanza D. Triana
Ping Pong
JOTEGO

Sheet: /capacitors/
File: capacitors.sch

Title: Decoupling capacitors

Size: A4	Date: 2022-03-28	**Rev: J. Tejada**
KiCad E.D.A. kicad (5.1.12-1-10_14)		Id: 11/11