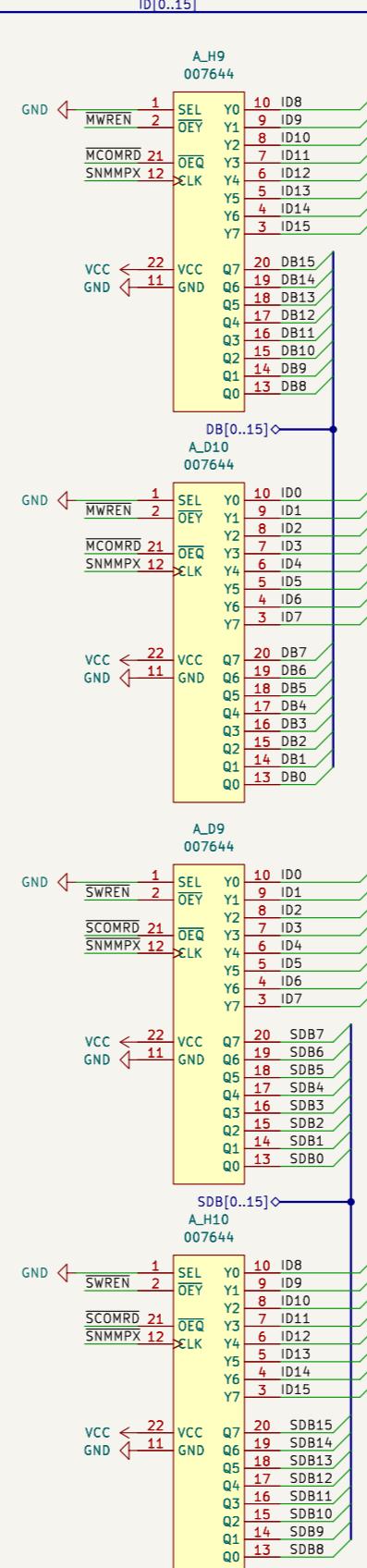
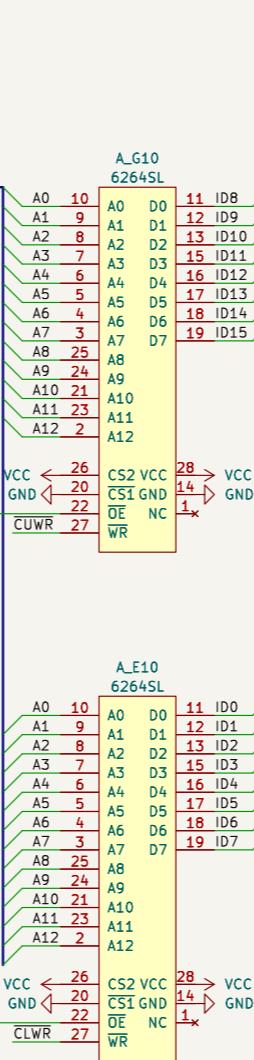
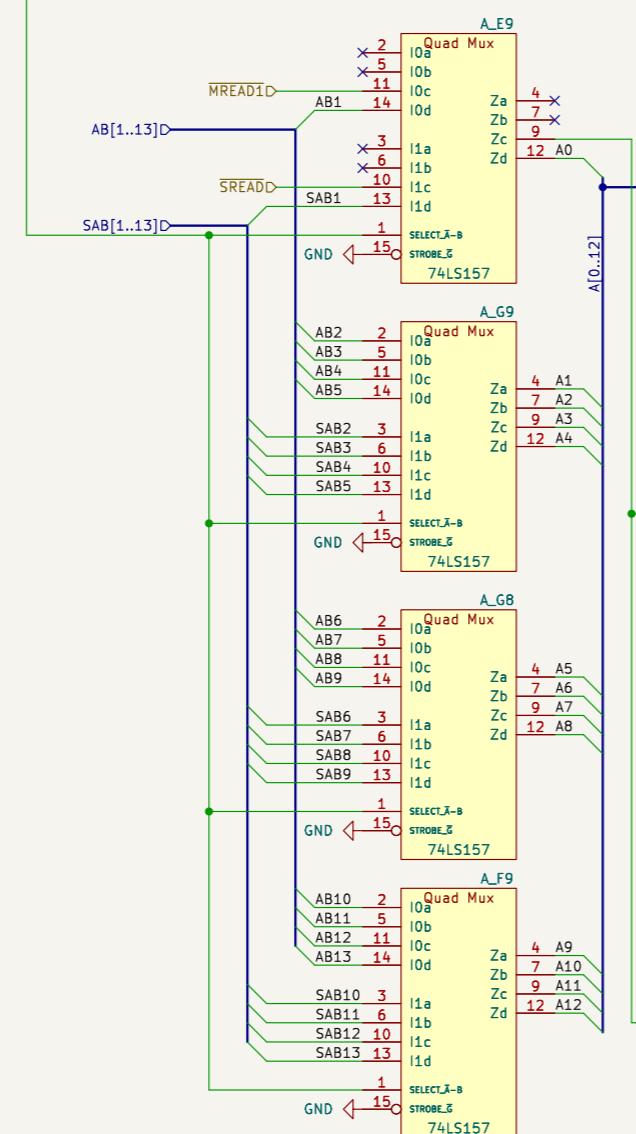
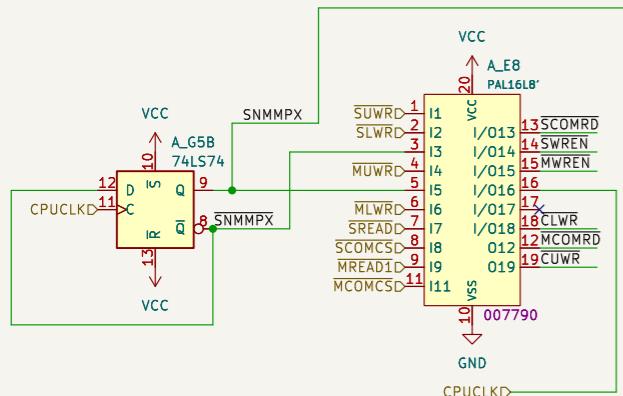


Konami 007790

$MCOMRD = \neg(MREAD1 \& \neg MCOMCS) = MREAD1 \mid MCOMCS$   
 $SCOMRD = \neg(SREAD \& \neg SCOMCS)$   
 $SWREN = \neg(SNMMPPX \& SREAD \& \neg SCOMCS)$   
 $MWREN = \neg(SNMMPPX \& MREAD1 \& \neg MCOMCS)$   
 $CLWR = \neg(CLWR \& \neg SNMMPPX \& SREAD \& \neg SCOMCS \& \neg CPUCLK \mid \neg SNMMPPX \& MREAD1 \& \neg MCOMCS \& \neg CPUCLK)$   
 $CUWR = \neg(CUWR \& \neg SNMMPPX \& SREAD \& \neg SCOMCS \& \neg CPUCLK \mid \neg MUWR \& \neg SNMMPPX \& MREAD1 \& \neg MCOMCS \& \neg CPUCLK)$

A All AND logic can be converted to OR form. But since AND logic and inverters are used in the PAL16LB device it is kept in that form. The equations follow verilog syntax. The active low signals are not inverted, the bar is there only to show activeness.



Ulf Skutnabba, twitter: @skutis77  
**Konami GX785/GX870/GX808/GX903**

Sheet: /Shared Resources/  
File: shared\_resources.kicad\_sch

**Title: Twin 16 – Rev B**

Size: A3 Date: 2024-09-08  
KiCad E.D.A. 8.0.5

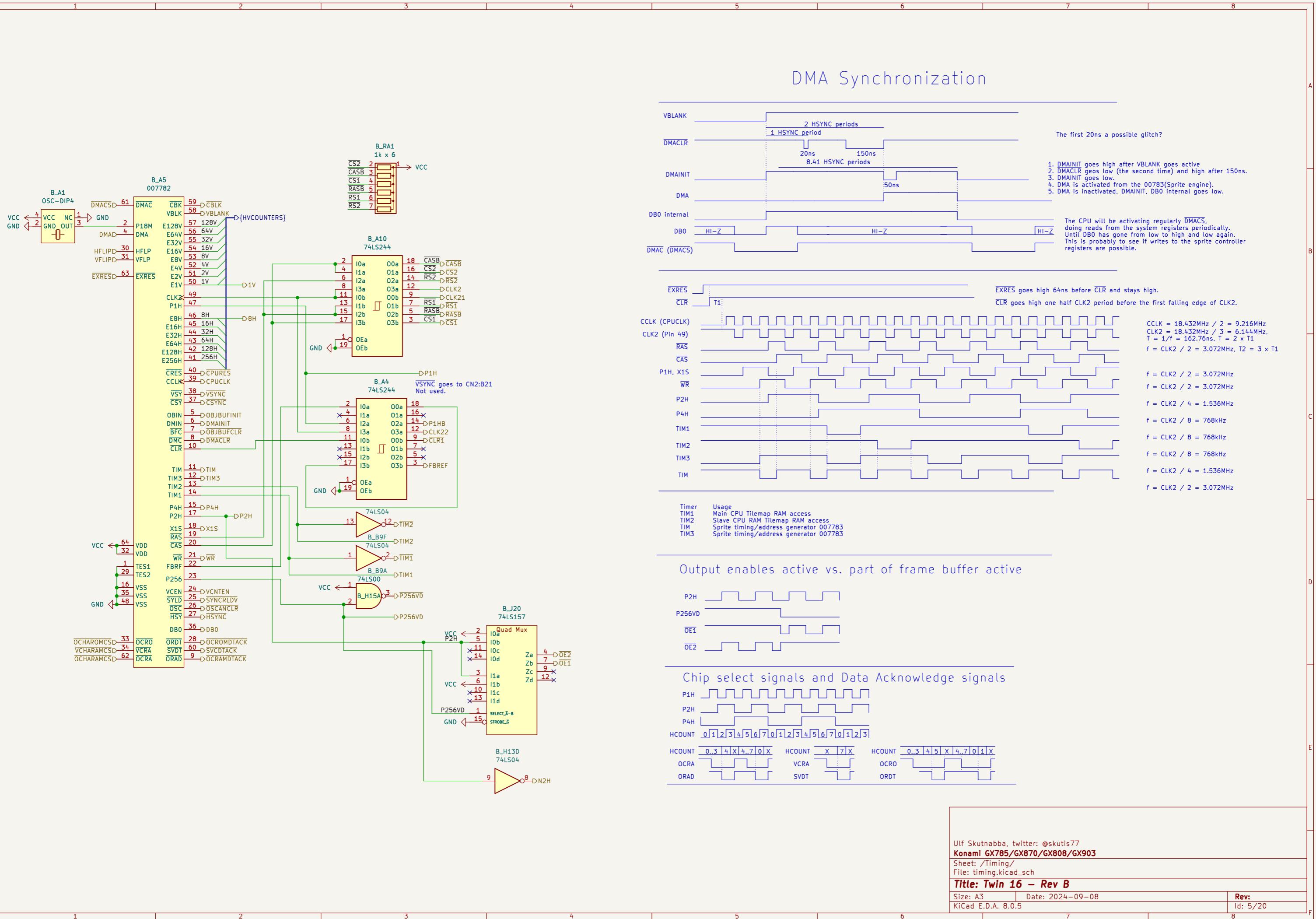
Rev: Id: 4/20

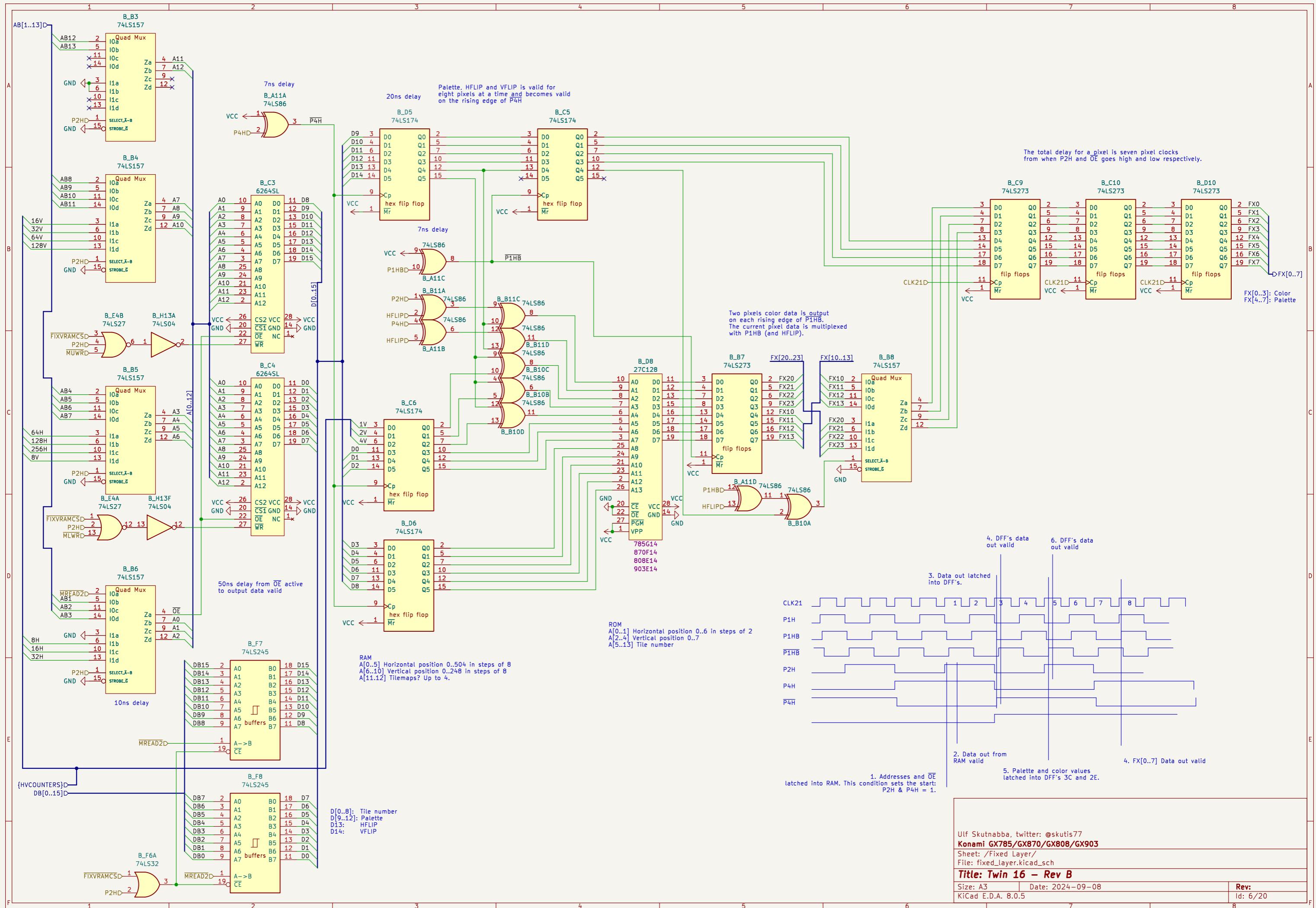
This is from jammarcade.net

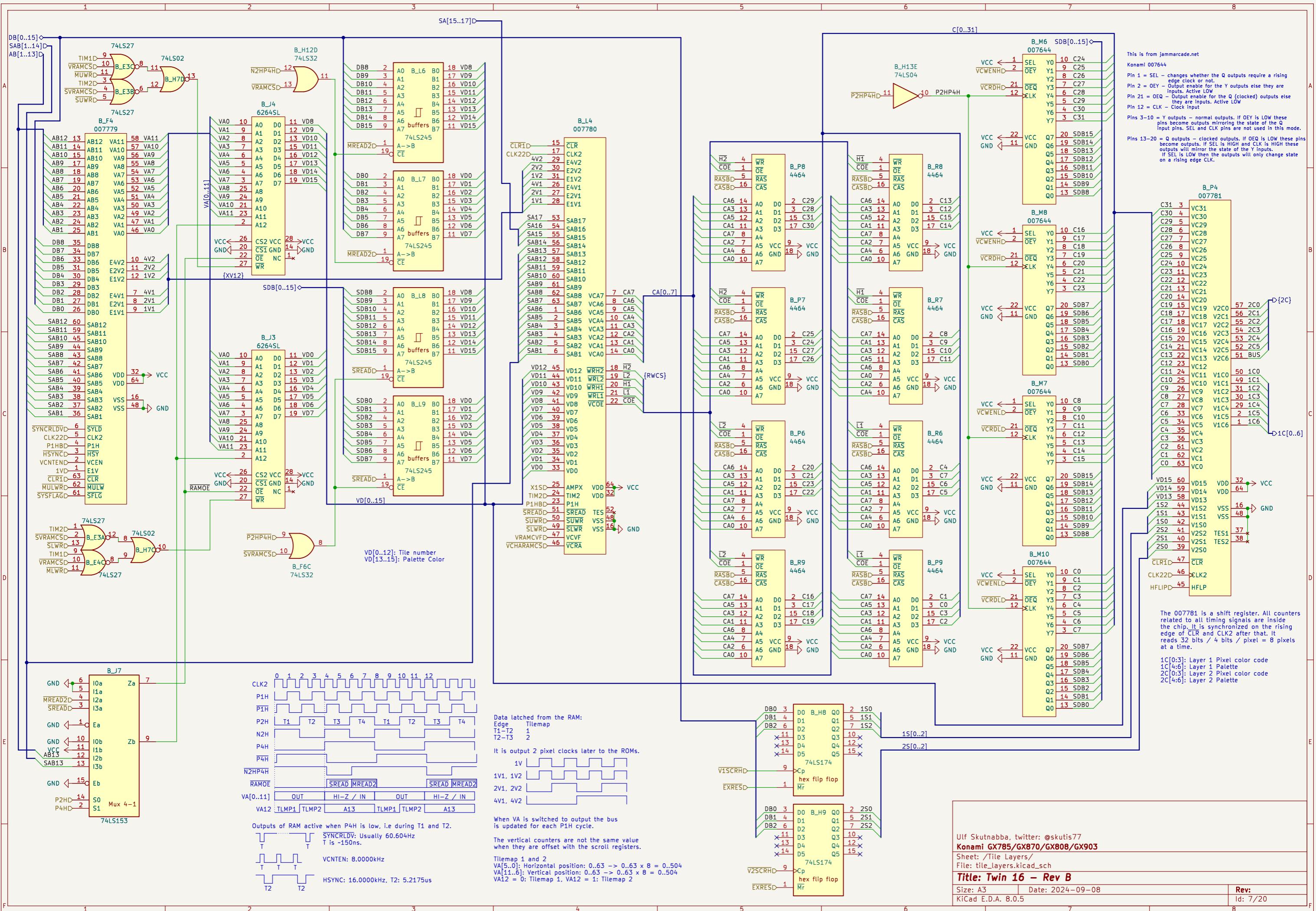
Pin 1 = SEL – changes whether the Q outputs require a rising edge clock or not.  
Pin 2 = OEQ – Output enable for the Y outputs else they are inputs. Active LOW  
Pin 21 = OEQ – Output enable for the Q (clocked) outputs else they are inputs. Active LOW  
Pin 12 = CLK – Clock input

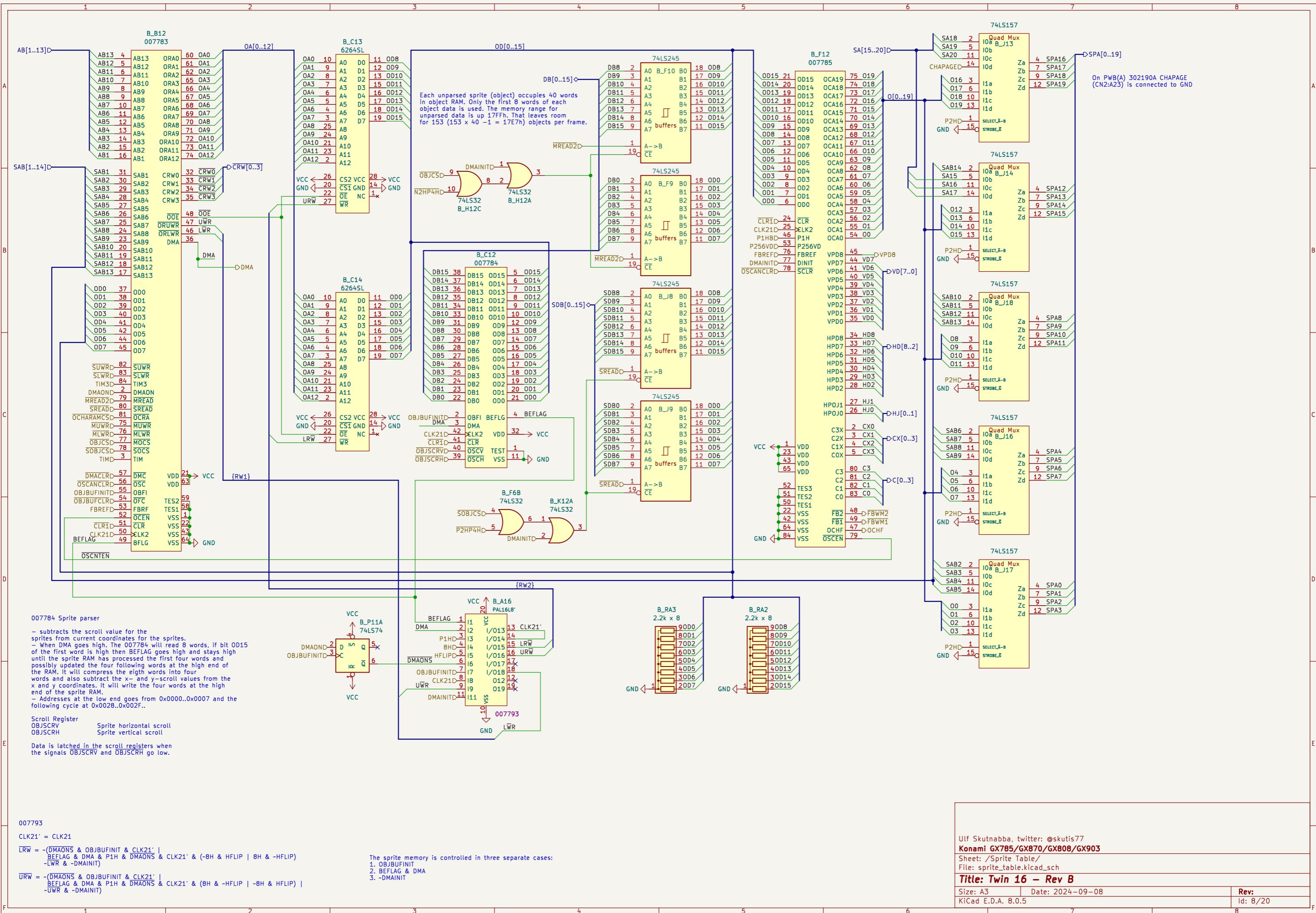
Pins 3–10 = Y outputs – normal outputs. If OEQ is LOW these pins become outputs  
mirroring the state of the Q input pins. SEL and CLK pins are not used in this mode.

Pins 13–20 = Q outputs – clocked outputs. If OEQ is LOW these pins become outputs.  
If SEL is HIGH and CLK is HIGH these outputs will mirror the state of the Y inputs.  
If SEL is LOW then the outputs will only change state on a rising edge CLK.

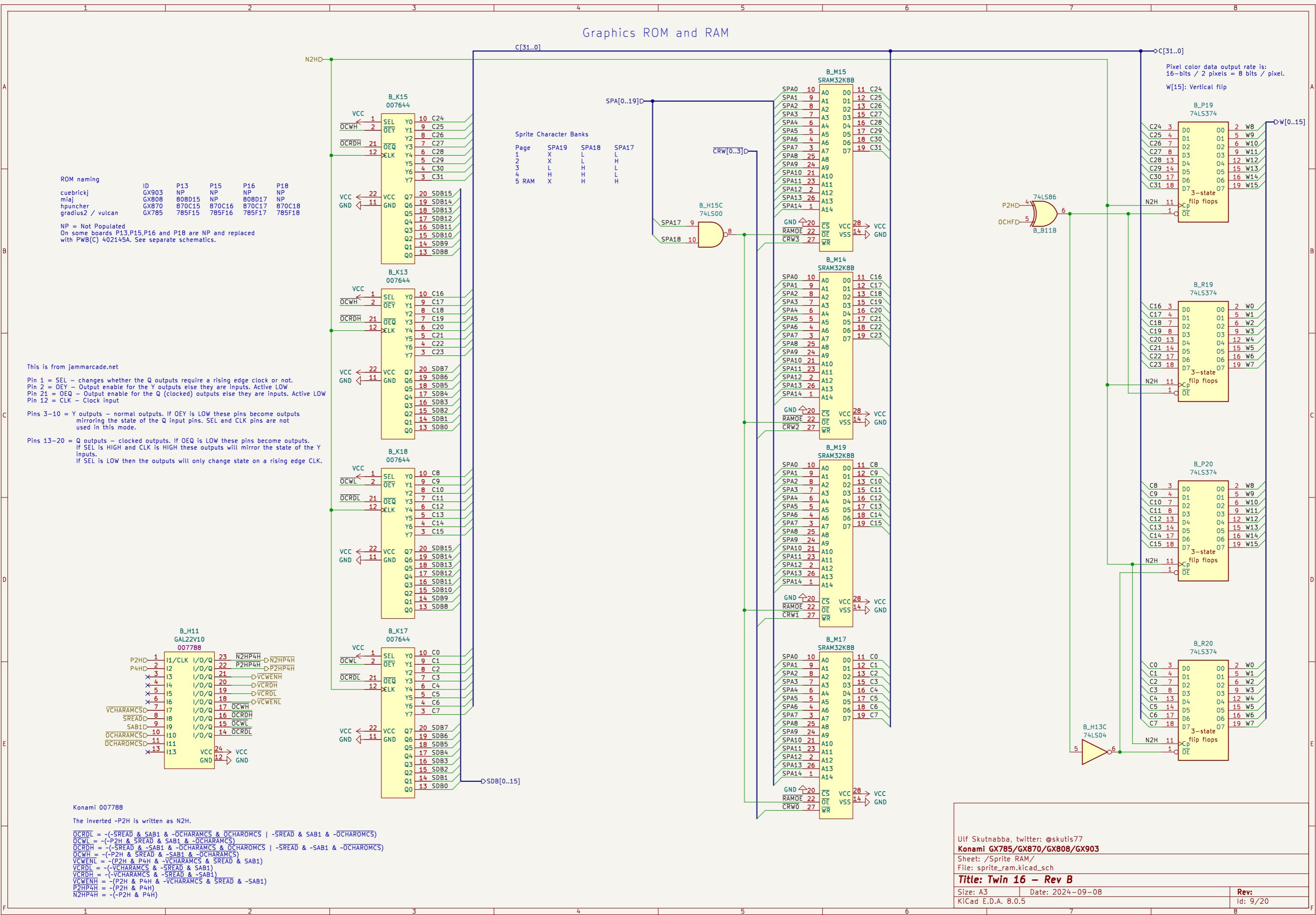


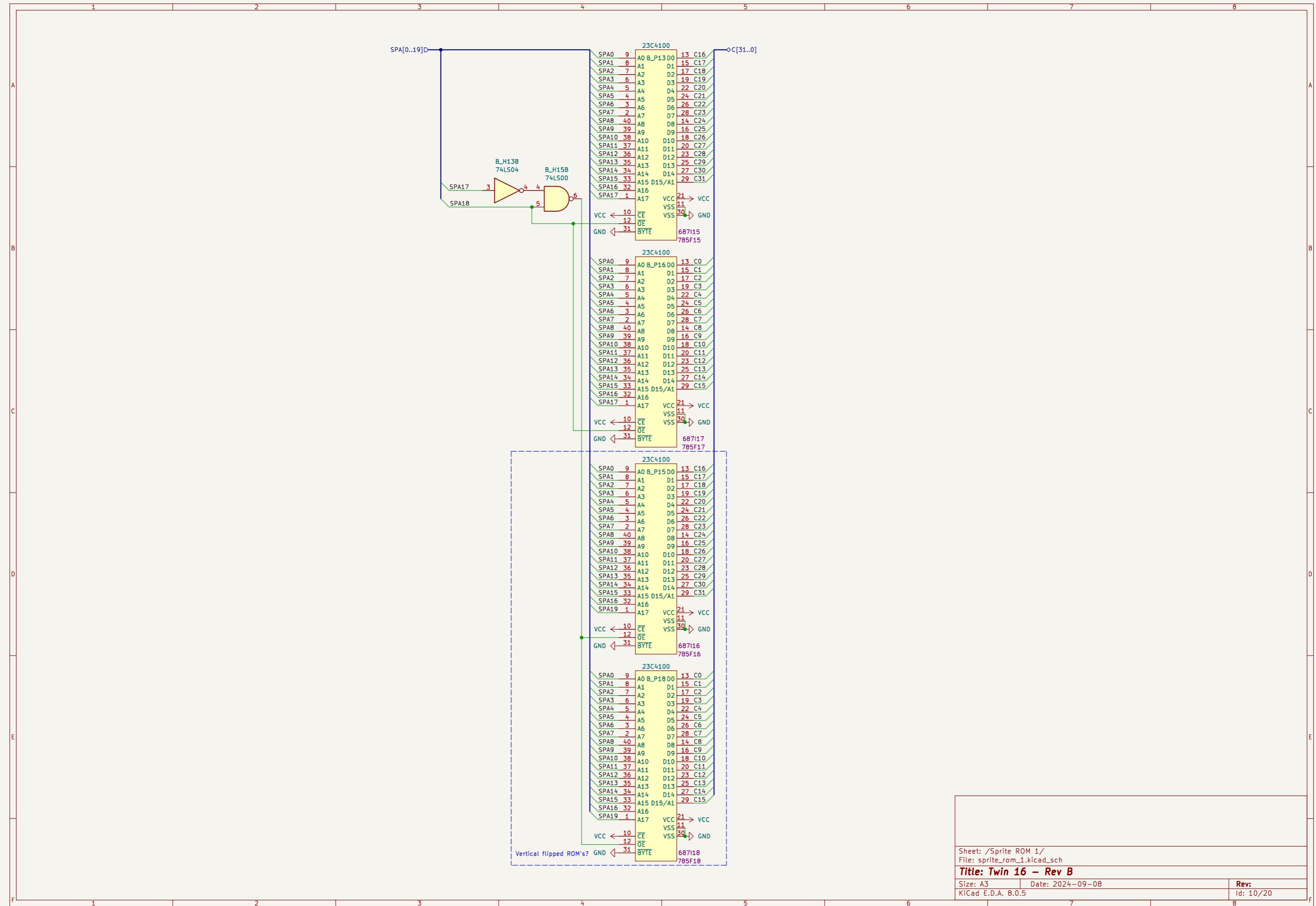


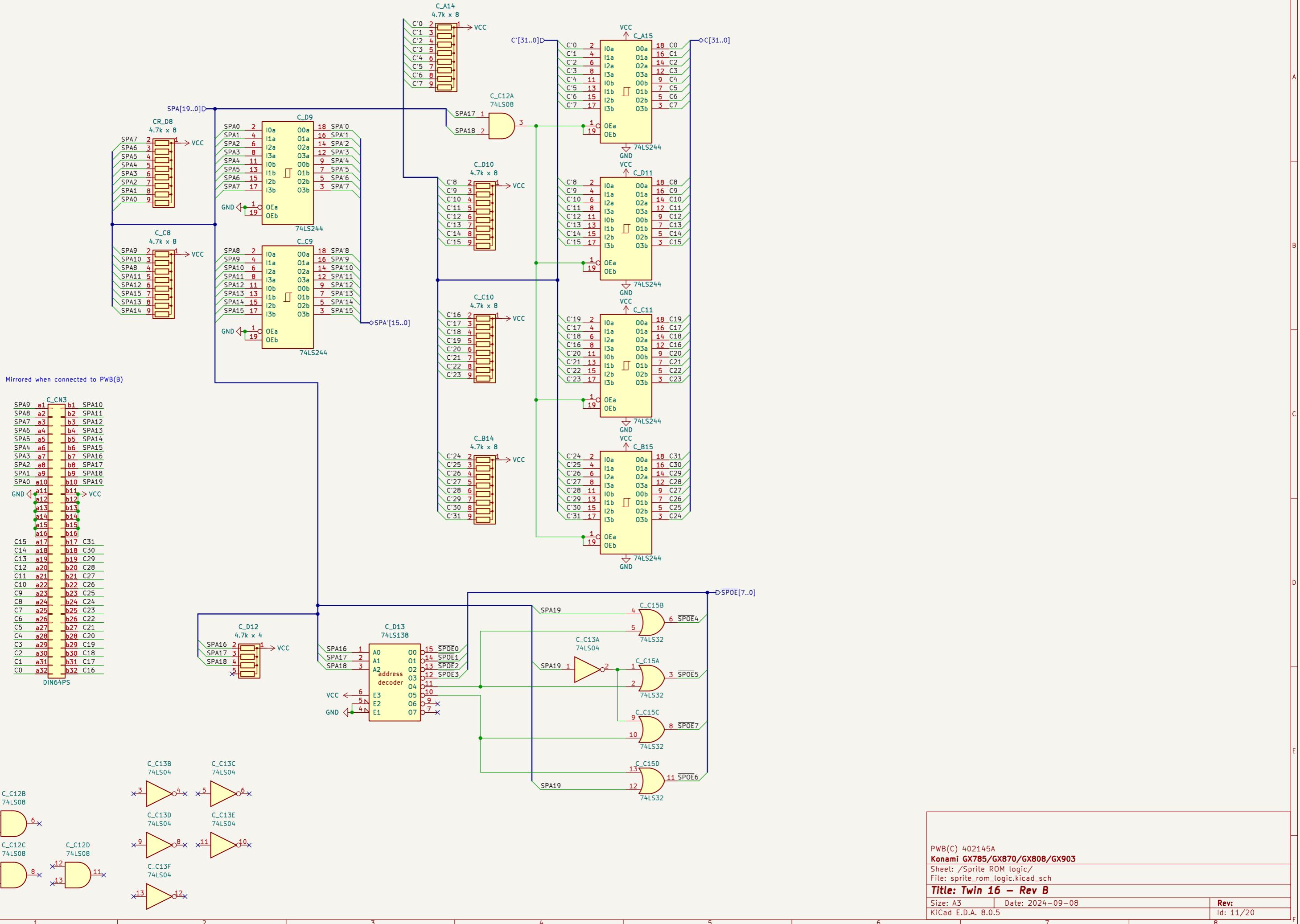




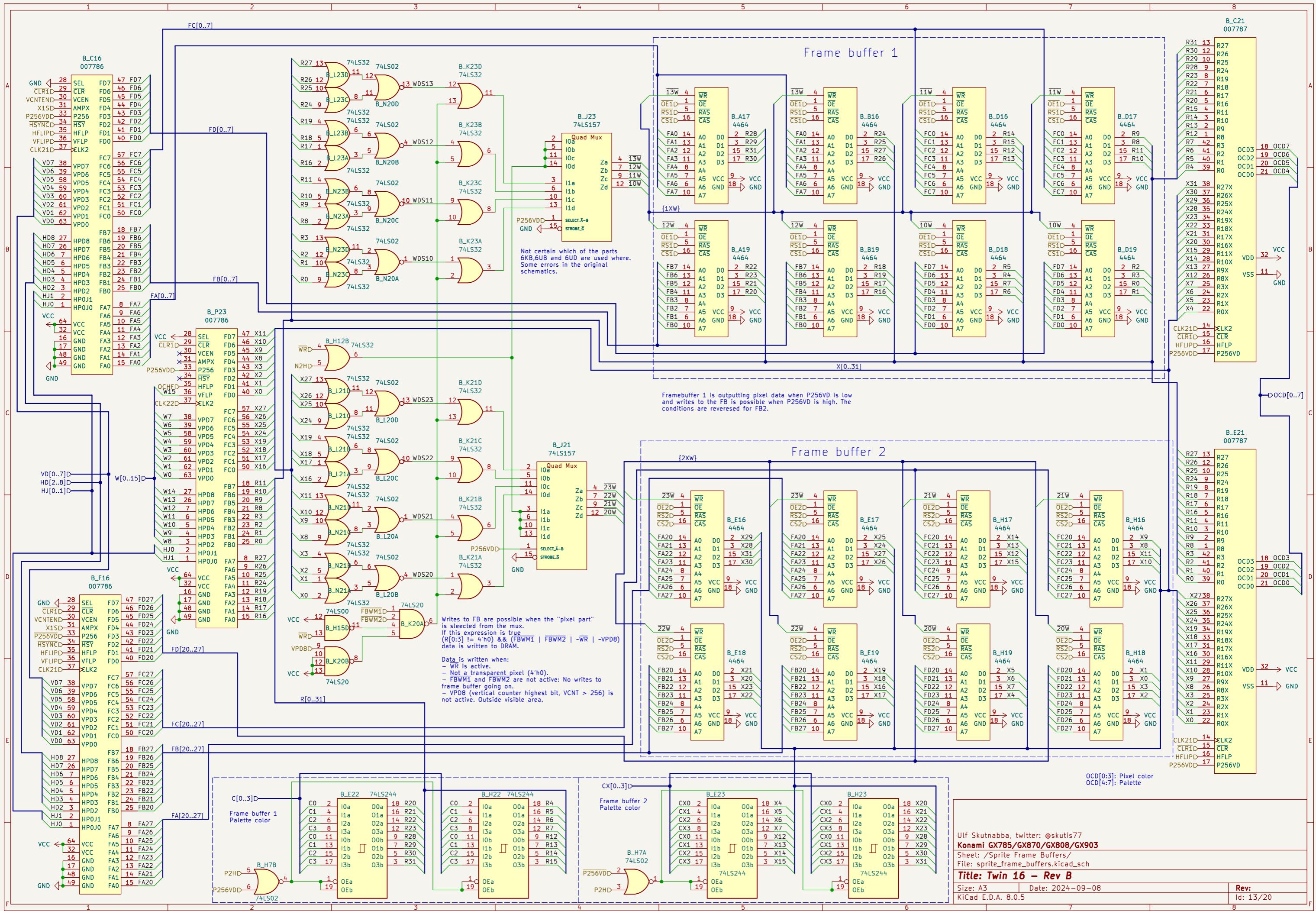
## Graphics ROM and RAM

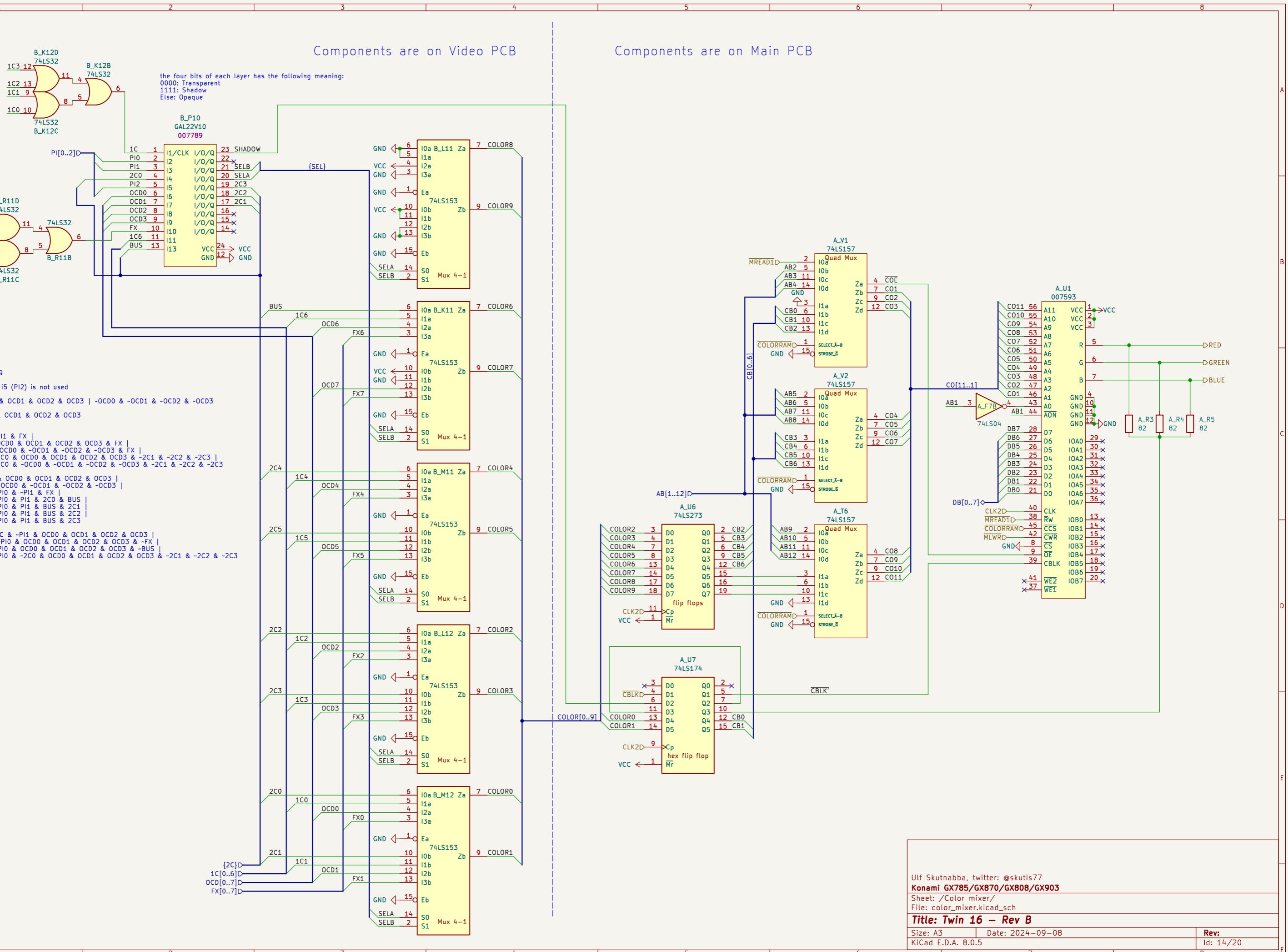












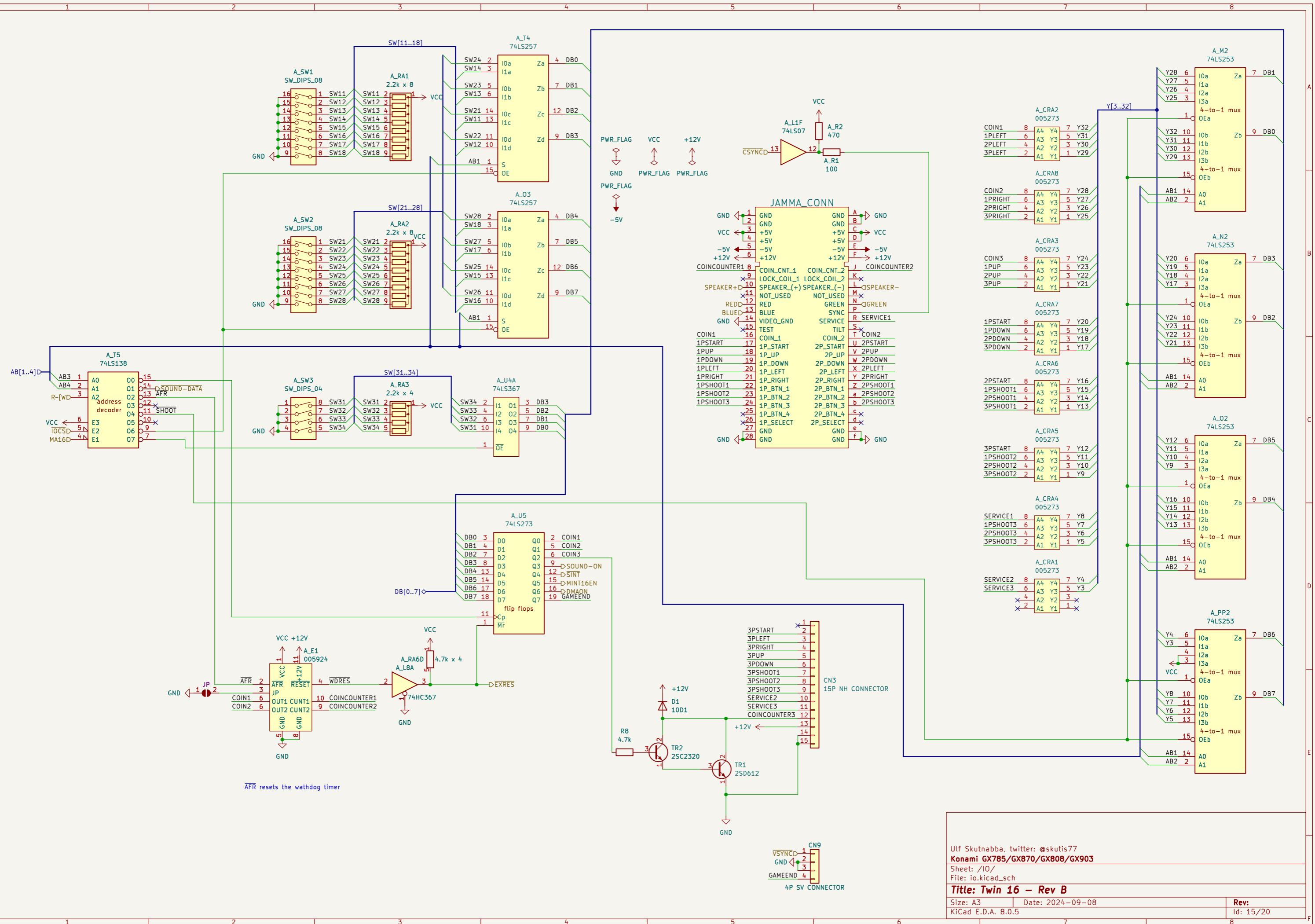
Ulf Skutnabba, twitter: @skutis77  
Konami GX785/GX870/GX808/GX903

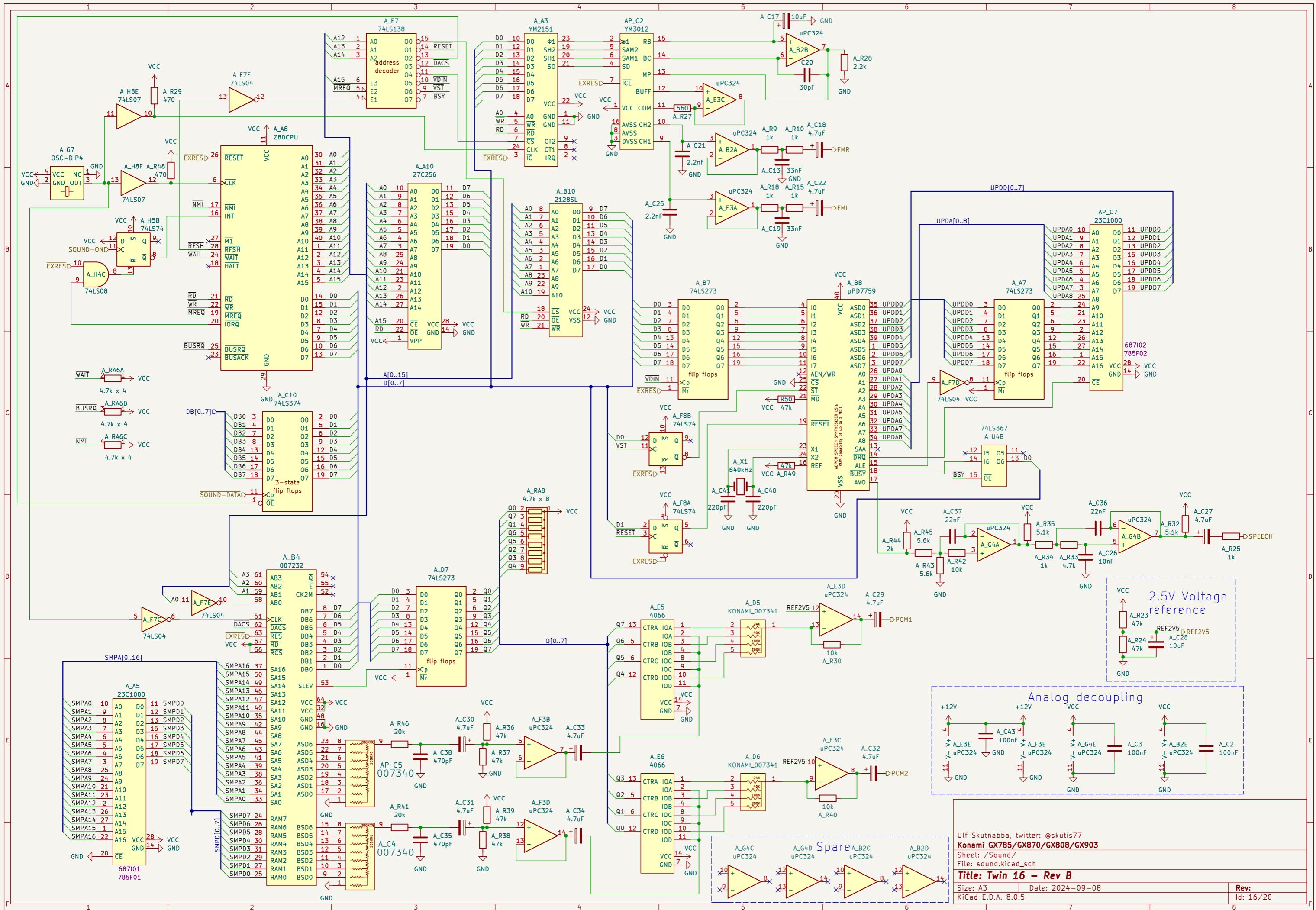
Sheet: /Color mixer/  
File: color\_mixer.kicad\_sch

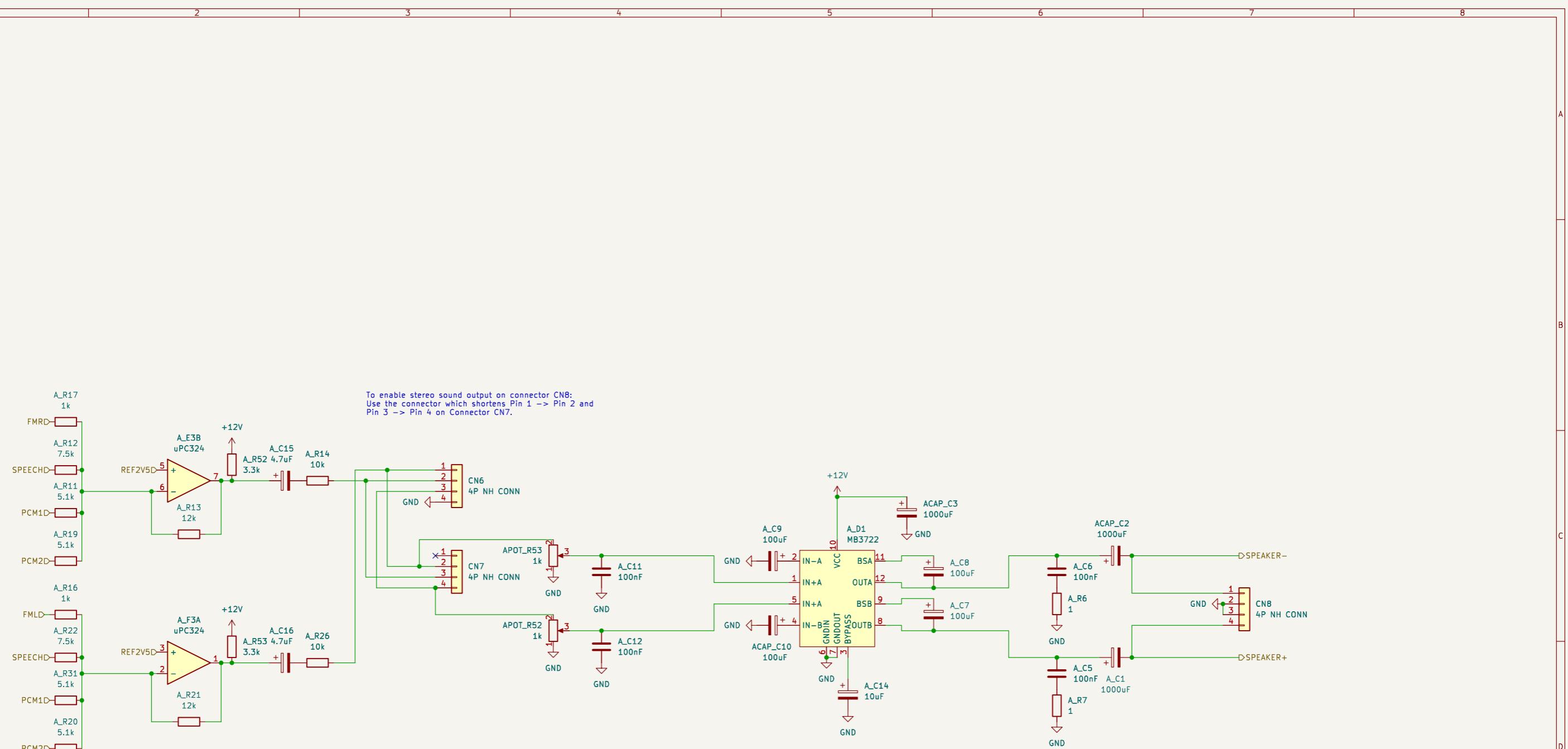
**Title: Twin 16 – Rev B**

Size: A3 Date: 2024-09-08  
KiCad E.D.A. 8.0.5

Rev: Id: 14/20







Ulf Skutnabba, twitter: @skutis77  
**Konami GX785/GX870/GX808/GX903**

Sheet: /Sound mixer/  
File: sound\_mixer.kicad\_sch

Title: Twin 16 - Rev B

Size: A3 Date: 2024  
KiCad EDA 8.0.5

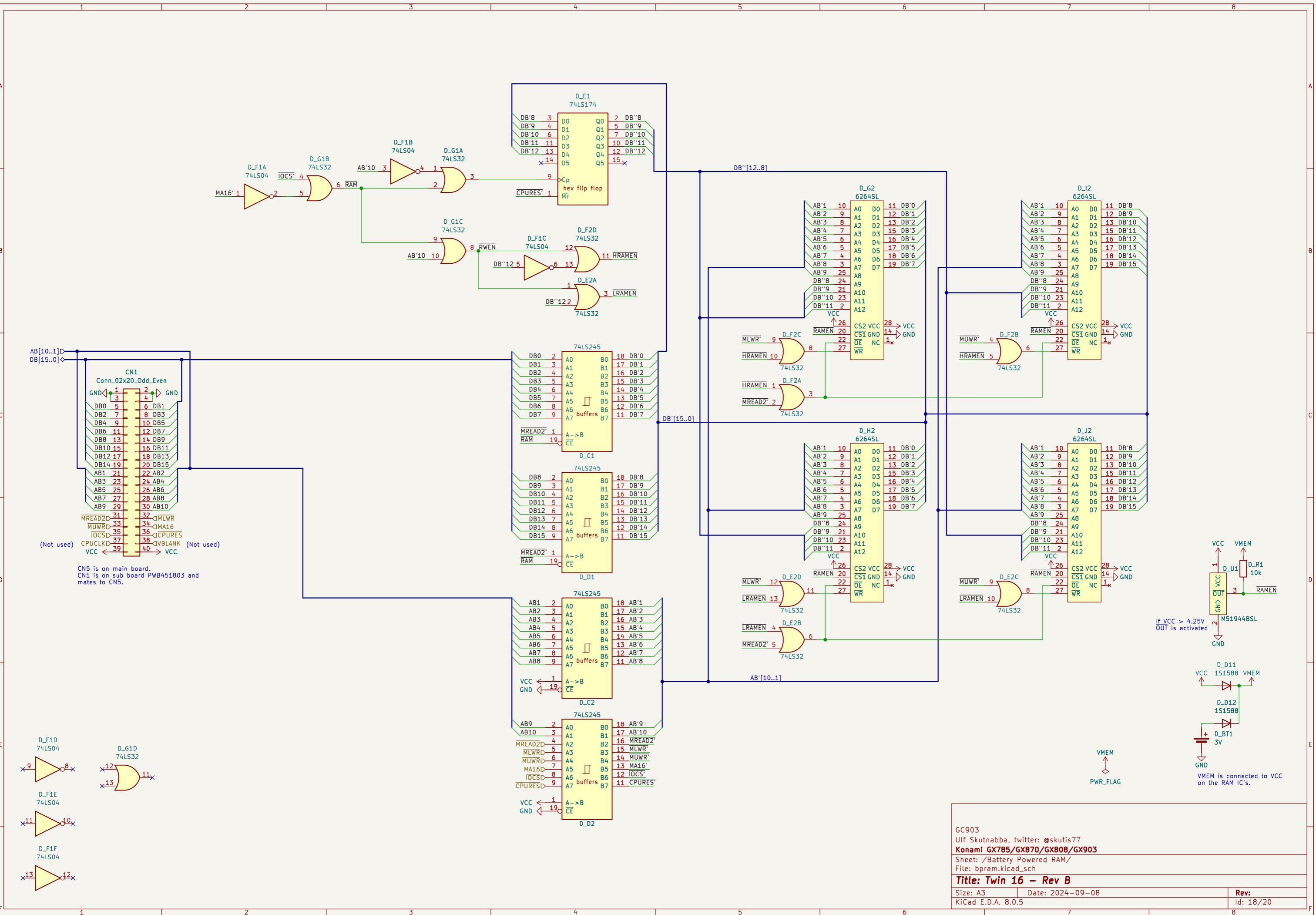
Ricau E.D.A. 0.0.3

Digitized by srujanika@gmail.com

Rev: 1

10.

1



GC903  
Ulf Skutnabba, twitter: @skutis77  
**Konami GX785/GX870/GX808/GX903**

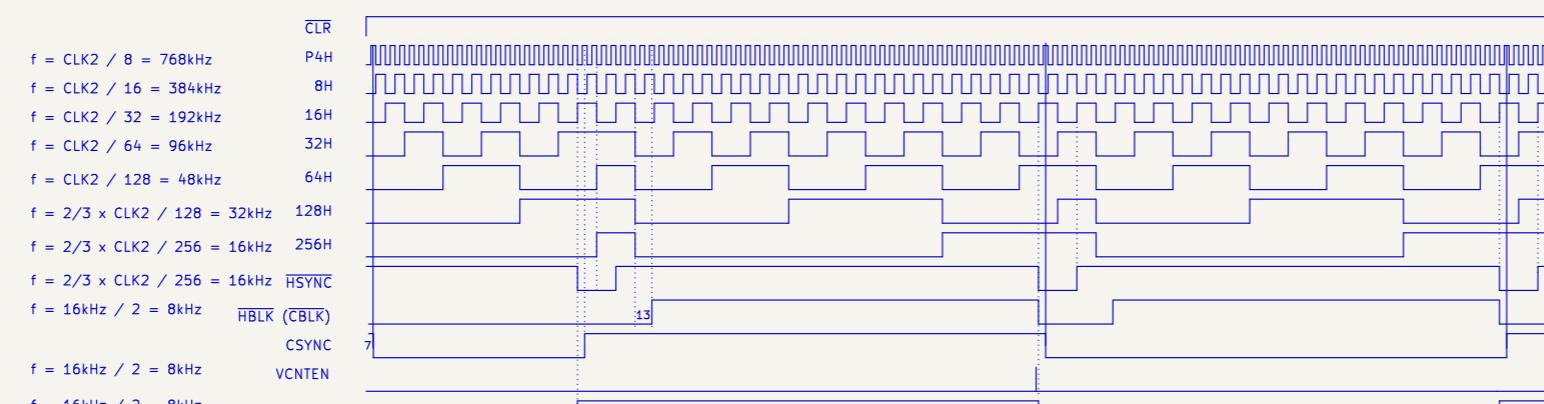
Sheet: /Battery Powered RAM/  
File: bpram.kicad\_sch

**Title: Twin 16 – Rev B**

Size: A3 Date: 2024-09-08  
KiCad E.D.A. 8.0.5

Rev:  
Id: 18/20

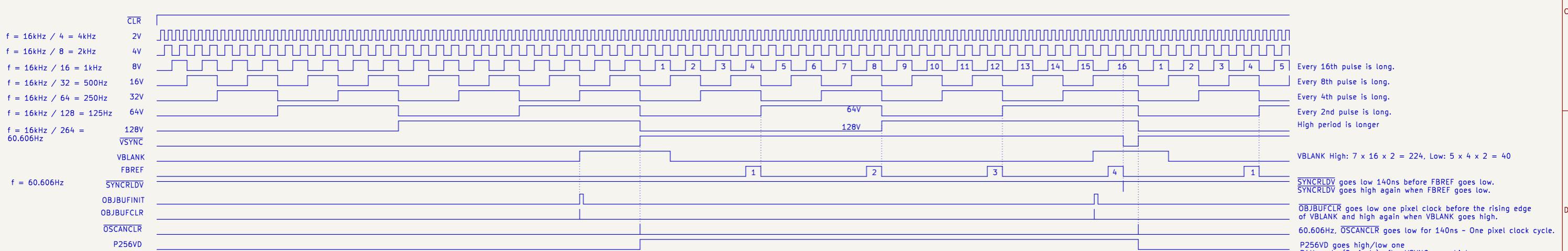
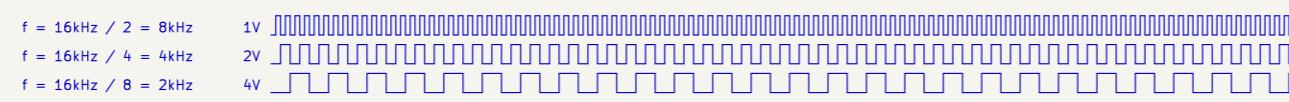
## Horizontal signals



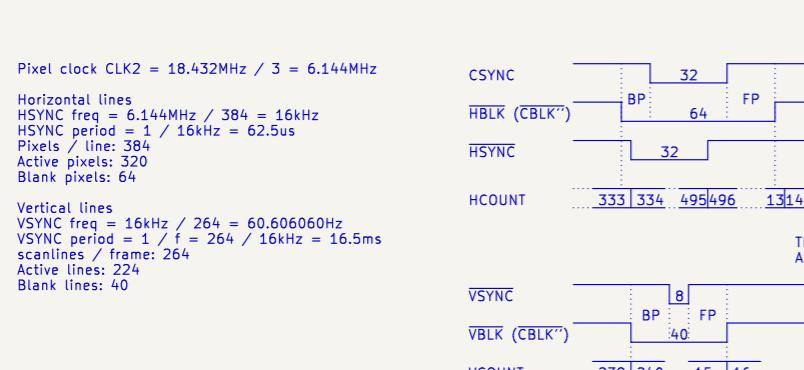
– The first VCNTEN is skipped after reset.  
It goes low 140ns before HSYNC goes low,  
and high again when HSYNC goes low.  
VCNTEN is active right before every second falling edge of  
HSYNC.

– CPURES goes high, and stays high, on the seventh falling edge  
of HSYNC.

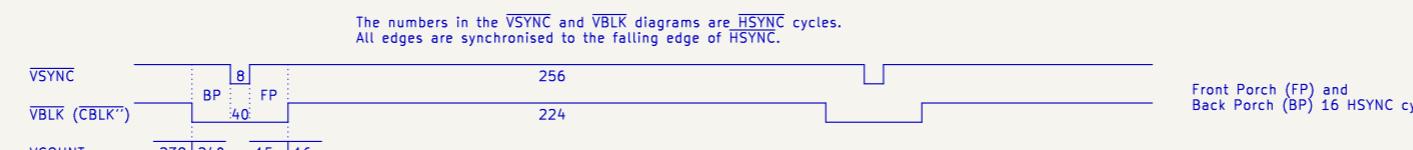
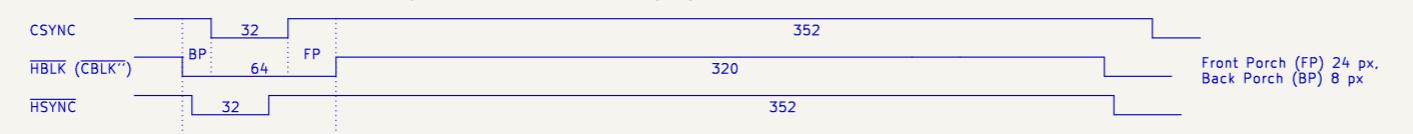
## Vertical signals



## Horizontal and vertical synch timing diagrams



The numbers in the HSYNC and HBLK diagrams are HSYNC cycles.  
All edges are synchronised to the rising edge of CLK2.



HCOUNT is bits [256H, 128H, 64H, 32H, 16H, 8H, P4H, P2H, P1H]  
VCOUNT is bits [128V, 64V, 32V, 16V, 8V, 4V, 2V, 1V]

CBLK'' is at the output of color mixer 007593.  
CBLK' is shifted inside the 007593 one pixel clock.

Ulf Skutnabba, twitter: @skutis77  
Konami GX785/GX870/GX808/GX903

Sheet: /Timing diagrams/  
File: timing\_diagrams.kicad\_sch

Title: Twin 16 – Rev B

Size: A3 Date: 2024-09-08  
KiCad E.D.A. 8.0.5

Rev: Id: 19/20

A

B

C

D

E

F

A

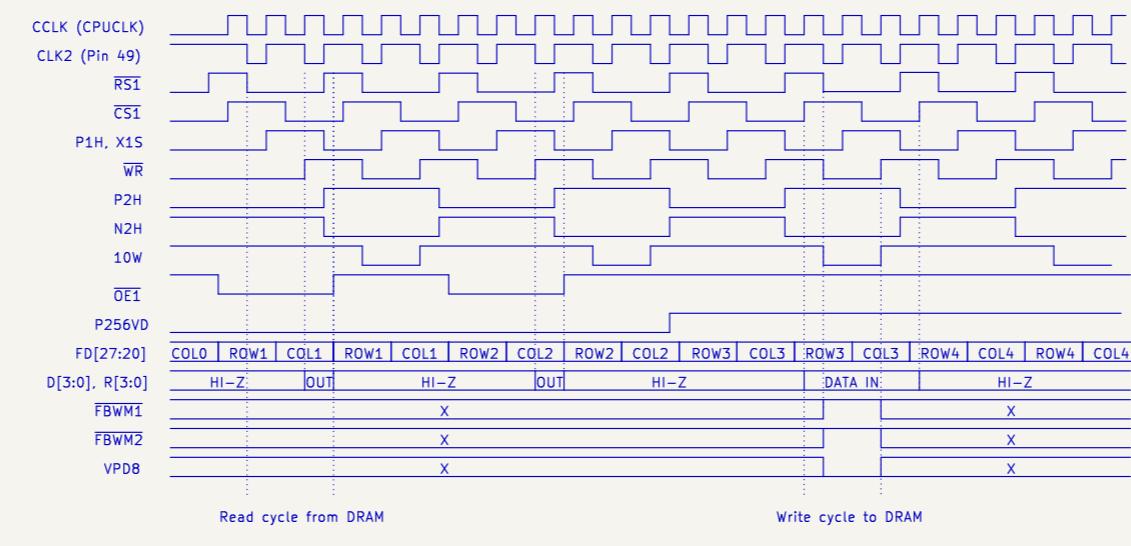
B

C

D

E

F



Ulf Skutnabba, twitter: @skutis77  
**Konami GX785/GX870/GX808/GX903**

Sheet: /Sprite timing diagrams/  
File: sprite\_timing\_diagrams.kicad\_sch

**Title: Twin 16 – Rev B**

Size: A3 | Date: 2024-09-08  
KiCad E.D.A. 8.0.5

Rev:  
Id: 20/20