Nametable Mapping

File: nametable_mapping.kicad_sch

CPU Address Bus

File: cpuAddressBus.kicad_sch

Register Access

File: register_access.kicad_sch

PRG Banking

File: prg_banking.kicad_sch

PPU Address Bus

File: ppu_address_bus.kicad_sch

Reset Detection

File: reset_detection.kicad_sch

CPU Data Bus

File: cpu_data_bus.kicad_sch

PCM DAC

File: pcm_dac.kicad_sch

Scanline Detection

File: scanline_detection.kicad_sch

V-Split

File: v_split.kicad_sch

PPU Fetch Counting

File: ppu_fetch_counting.kicad_sch

CHR A0,A1,A2

File: chr_a0_a1_a2.kicad_sch

ExRAM Mode

File: exram_mode.kicad_sch

Fill Mode

File: fill_mode.kicad_sch

Pulse Length Lookup Table

File: pulse_length_lookup_table.kicad_sch

Pulse Timer

File: pulse_timer.kicad_sch

Pulse Duty Envelope Volume

File: pulse_duty_envelope_volume.kicad_sch

Pulse DAC

File: Pulse DAC.kicad_sch

PPU Data Bus

File: ppu_data_bus.kicad_sch

CHR Bank Registers

File: chr_bank_registers.kicad_sch

CHR Bank Selection Logic

File: chr_bank_selection_logic.kicad_sch

CHR Address Bus 10 to 14

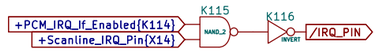
File: chr_address_bus_10_to_14.kicad_sch

CHR Address Bus 15 to 19

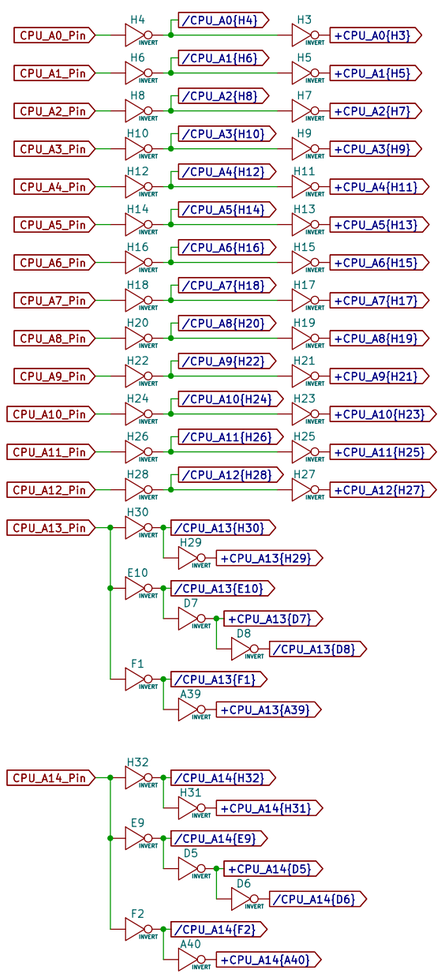
File: chr_address_bus_15_to_19.kicad_sch

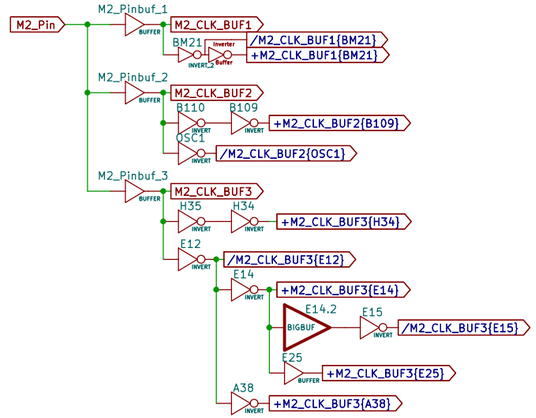
Artifacts

File: artifacts.kicad_sch

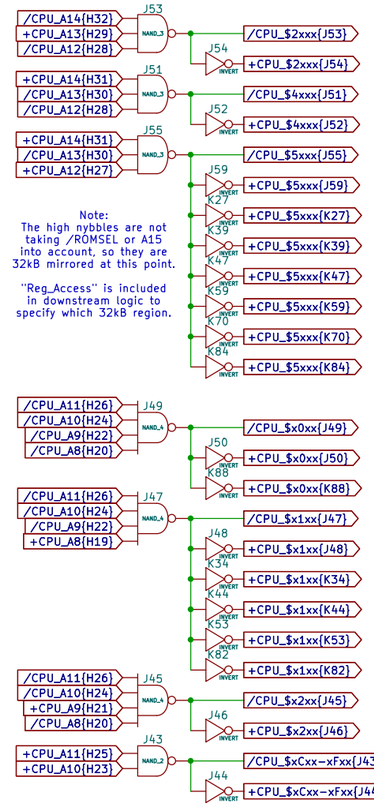


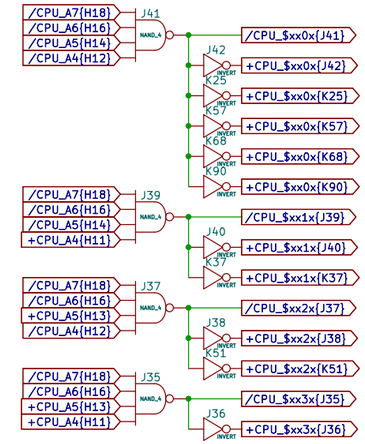
CPU Address Bus Buffering

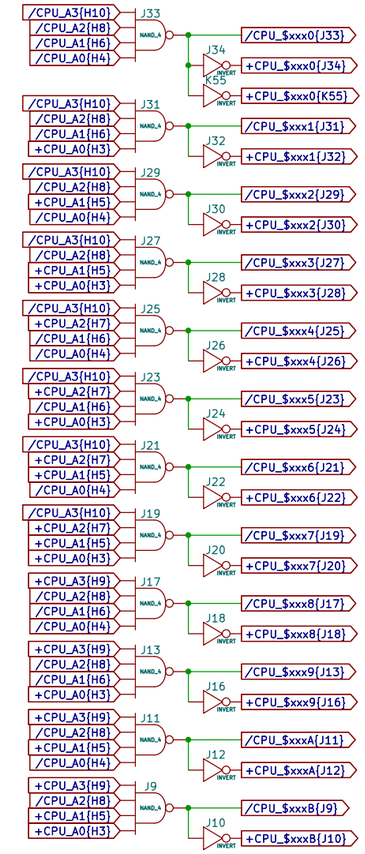


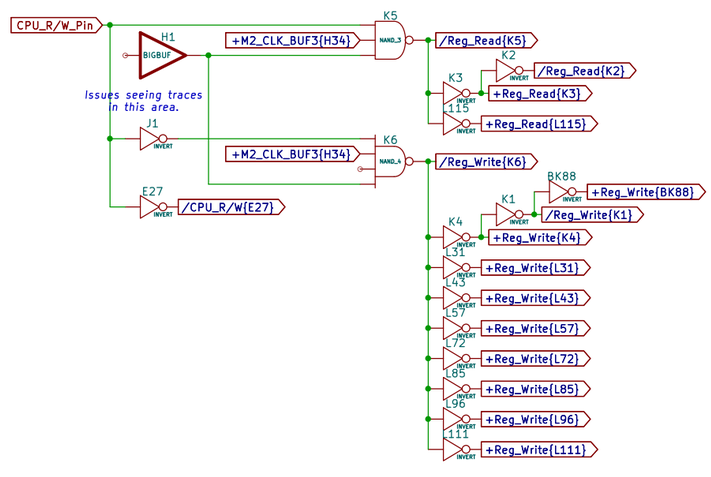


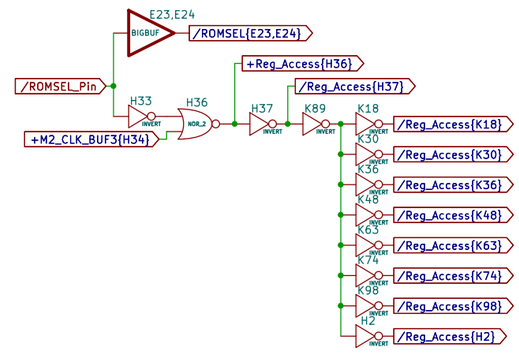
CPU Address Bus Nybble Decoding



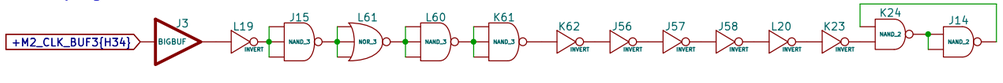






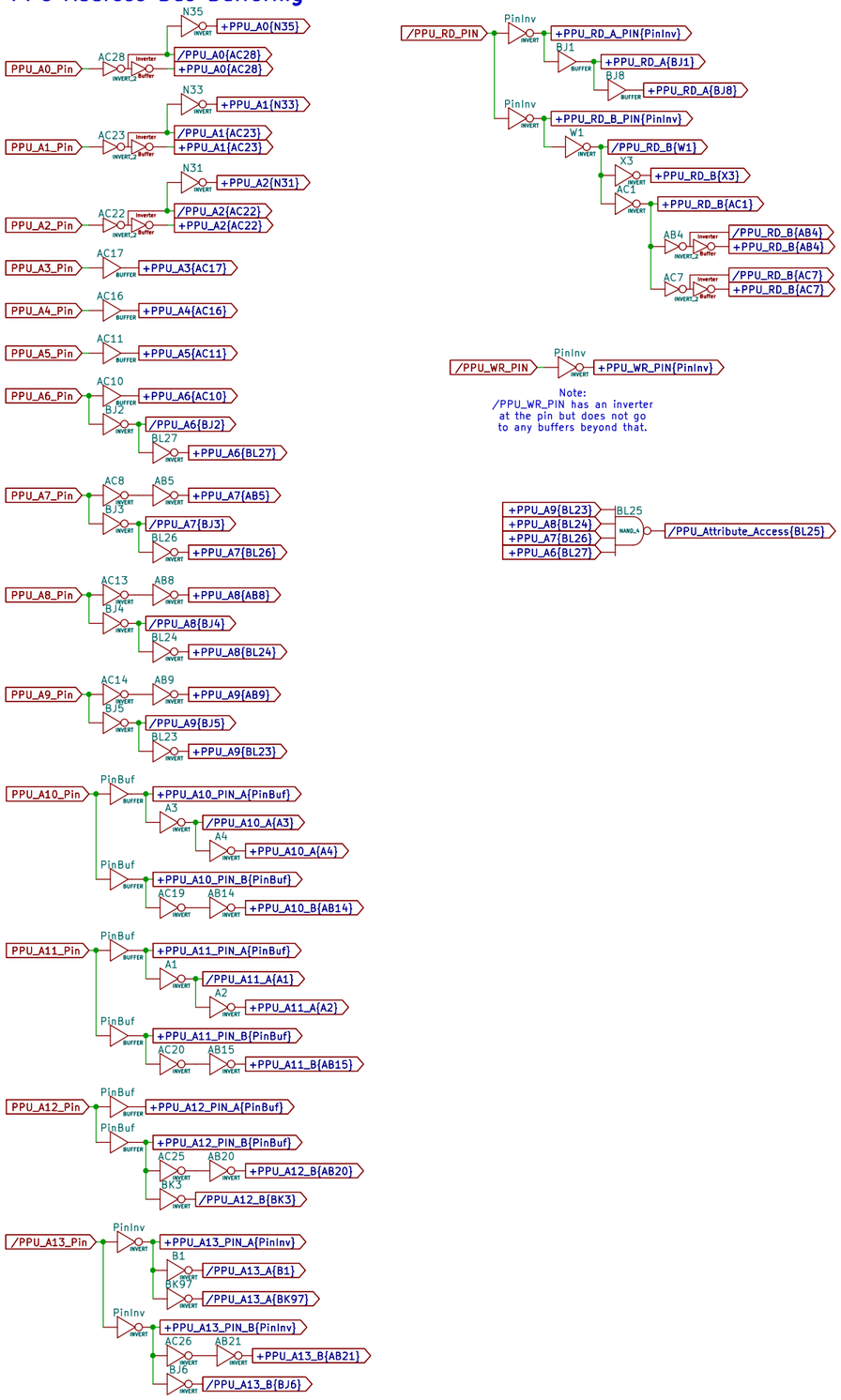


M2 Spaghetti That Goes Nowhere

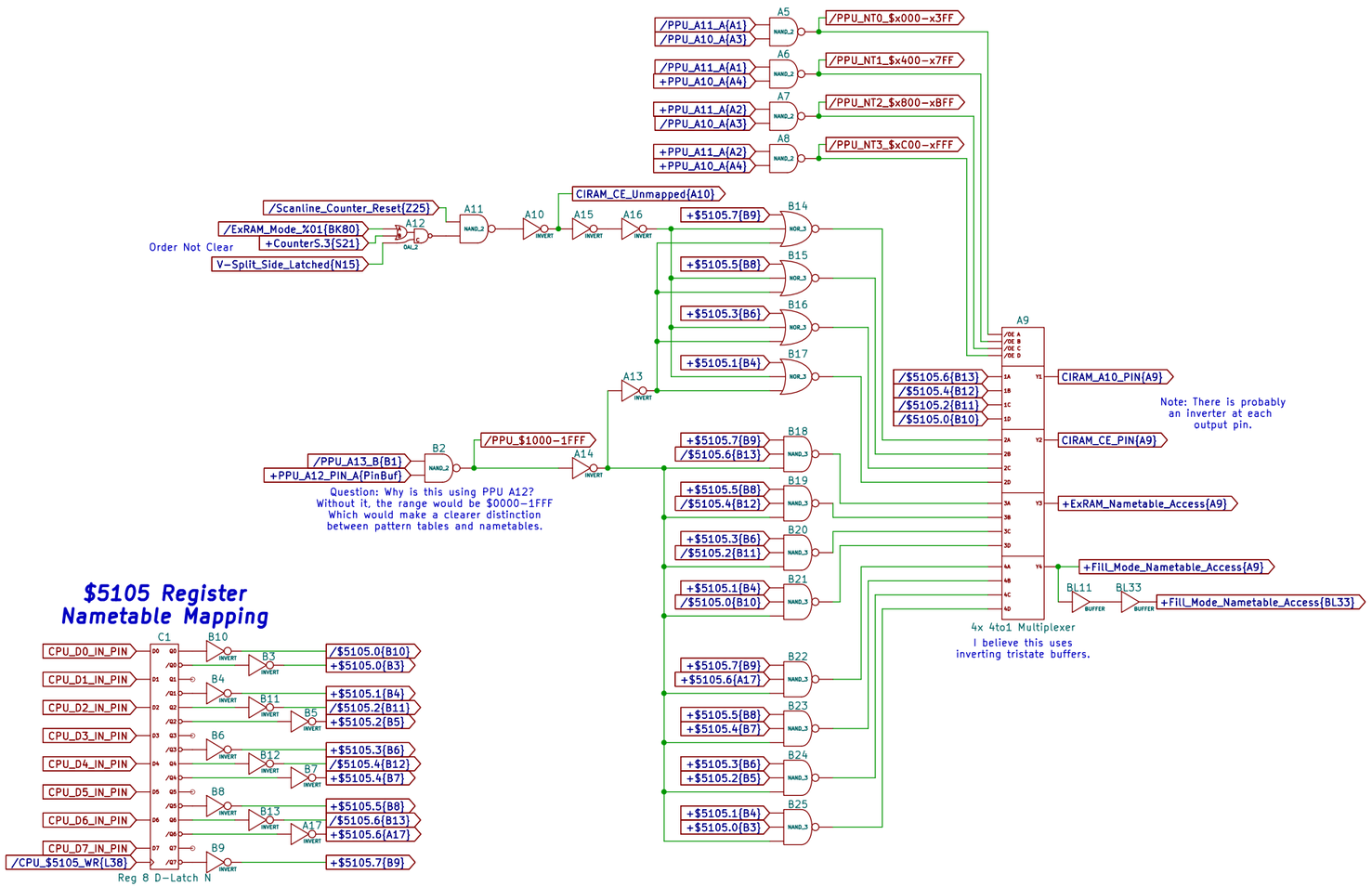


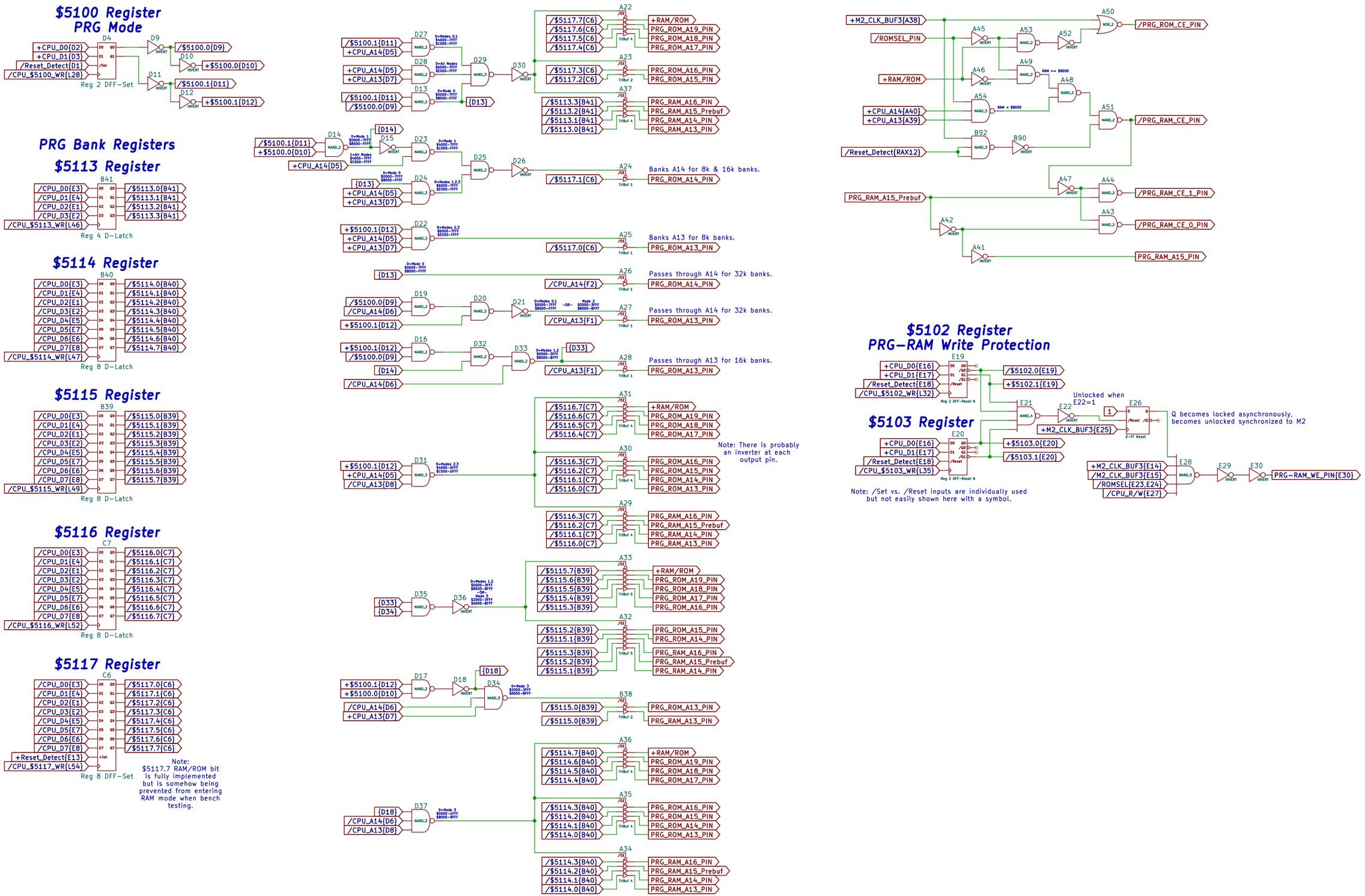
+CPU_$2xxx{J54}
+CPU_$x0xx{K88}
+CPU_$xx0x{K68}
+CPU_$xxx0{K55}
K87
NAND_A
/CPU_$2000{K87}
L106
NOR_2
/Reg_Access{H2}
+Reg_Write{L111}
L105
NAND_2
+CPU_$2000_Access{L106}
/CPU_$2000_WR{L105}
+CPU_$2xxx{J54}
+CPU_$x0xx{K88}
+CPU_$xx0x{K90}
+CPU_$xxx1{J32}
K91
NAND_A
/CPU_$2001{K91}
L108
NOR_2
/Reg_Access{K98}
+Reg_Write{L111}
L107
NAND_2
+CPU_$2001_Access{L107}
/CPU_$2001_WR{L107}
+CPU_$2xxx{J54}
+CPU_$x0xx{K88}
+CPU_$xx0x{K90}
+CPU_$xxx2{J30}
K92
NAND_A
/CPU_$2002{K92}
L109
NOR_2
/Reg_Access{K98}
+Reg_Read{L115}
L110
NAND_2
+CPU_$2002_Access{L109}
/CPU_$2002_RD{L110}
+CPU_$2xxx{J54}
+CPU_$x0xx{K88}
+CPU_$xx0x{K90}
+CPU_$xxx5{J24}
K93
NAND_A
/CPU_$2005{K93}
L112
NOR_2
/Reg_Access{K98}
+Reg_Write{L111}
L113
NAND_2
+CPU_$2005_Access{L112}
/CPU_$2005_WR{L113}
+CPU_$4xxx{J52}
+CPU_$x0xx{K88}
+CPU_$xx1x{K37}
+CPU_$xxx4{J26}
K94
NAND_A
/CPU_$4014{K94}
L114
NOR_2
/Reg_Access{K98}
+Reg_Write{L111}
L116
NAND_2
+CPU_$4014_Access{L114}
/CPU_$4014_WR{L116}
+CPU_$5xxx{J59}
+CPU_$x0xx{J50}
+CPU_$xx0x{J42}
+CPU_$xxx0{J34}
K8
NAND_A
/CPU_$5000{K8}
K9
NOR_2
/Reg_Access{K18}
+Reg_Write{K4}
K10
NAND_2
+CPU_$5000_Access{K9}
/CPU_$5000_WR{K10}
+CPU_$5xxx{J59}
+CPU_$x0xx{J50}
+CPU_$xx0x{J42}
+CPU_$xxx2{J30}
K11
NAND_A
/CPU_$5002{K11}
K12
NOR_2
/Reg_Access{K18}
+Reg_Write{K4}
K13
NAND_2
+CPU_$5002_Access{K12}
/CPU_$5002_WR{K13}
+CPU_$5xxx{J59}
+CPU_$x0xx{J50}
+CPU_$xx0x{J42}
+CPU_$xxx3{J28}
K14
NAND_A
/CPU_$5003{K14}
K15
NOR_2
/Reg_Access{K18}
+Reg_Write{K4}
K16
NAND_2
+CPU_$5003_Access{K15}
/CPU_$5003_WR{K16}
+CPU_$5xxx{K70}
+CPU_$x0xx{K88}
+CPU_$xx0x{K68}
+CPU_$xxx4{J26}
K71
NAND_A
/CPU_$5004{K71}
K72
NOR_2
/Reg_Access{K74}
+Reg_Write{L96}
K73
NAND_2
+CPU_$5004_Access{K72}
/CPU_$5004_WR{K73}
+CPU_$5xxx{K70}
+CPU_$x0xx{K88}
+CPU_$xx0x{K68}
+CPU_$xxx6{J22}
K75
NAND_A
/CPU_$5006{K75}
K76
NOR_2
/Reg_Access{K74}
+Reg_Write{L96}
K77
NAND_2
+CPU_$5006_Access{K76}
/CPU_$5006_WR{K77}
+CPU_$5xxx{K70}
+CPU_$x0xx{K88}
+CPU_$xx0x{K68}
+CPU_$xxx7{J20}
K78
NAND_A
/CPU_$5007{K78}
K79
NOR_2
/Reg_Access{H2}
+Reg_Write{L96}
K80
NAND_2
+CPU_$5007_Access{K79}
/CPU_$5007_WR{K80}
+CPU_$5xxx{J59}
+CPU_$x0xx{J50}
+CPU_$xx1x{J40}
+CPU_$xxx0{J34}
K17
NAND_A
/CPU_$5010{K17}
+Reg_Read{K3}
L22
NOR_2
/Reg_Access{K18}
+Reg_Write{K4}
L24
NAND_2
/CPU_$5010_RD{L24}
+CPU_$5010_Access{L22}
L23
NAND_2
/CPU_$5010_WR{L23}
+CPU_$5xxx{J59}
+CPU_$x0xx{J50}
+CPU_$xx1x{J40}
+CPU_$xxx1{J32}
K19
NAND_A
/CPU_$5011{K19}
L25
NOR_2
/Reg_Access{K18}
+Reg_Write{L31}
L26
NAND_2
+CPU_$5011_Access{L25}
/CPU_$5011_WR{L26}
+CPU_$5xxx{K84}
+CPU_$x0xx{K88}
+CPU_$xx1x{K37}
+CPU_$xxx5{J24}
K97
NAND_A
/CPU_$5015{K97}
+Reg_Read{L115}
L118
NOR_2
/Reg_Access{K98}
+Reg_Write{L111}
L120
NAND_2
/CPU_$5015_RD{L120}
+CPU_$5015_Access{L118}
L119
NAND_2
/CPU_$5015_WR{L119}
+CPU_$5xxx{J59}
+CPU_$x1xx{J48}
+CPU_$xx0x{J42}
+CPU_$xxx0{J34}
K20
NAND_A
/CPU_$5100{K20}
L27
NOR_2
/Reg_Access{K18}
+Reg_Write{L31}
L28
NAND_2
+CPU_$5100_Access{L27}
/CPU_$5100_WR{L28}
+CPU_$5xxx{K27}
+CPU_$x1xx{J48}
+CPU_$xx0x{J42}
+CPU_$xxx1{J32}
K21
NAND_A
/CPU_$5101{K21}
L30
NOR_2
/Reg_Access{K30}
+Reg_Write{L31}
L29
NAND_2
+CPU_$5101_Access{L30}
/CPU_$5101_WR{L29}
+CPU_$5xxx{K27}
+CPU_$x1xx{J48}
+CPU_$xx0x{K25}
+CPU_$xxx2{J30}
K22
NAND_A
/CPU_$5102{K22}
L33
NOR_2
/Reg_Access{K30}
+Reg_Write{L31}
L32
NAND_2
+CPU_$5102_Access{L33}
/CPU_$5102_WR{L32}
+CPU_$5xxx{K27}
+CPU_$x1xx{J48}
+CPU_$xx0x{K25}
+CPU_$xxx3{J28}
K26
NAND_A
/CPU_$5103{K26}
L34
NOR_2
/Reg_Access{K30}
+Reg_Write{L31}
L35
NAND_2
+CPU_$5103_Access{L34}
/CPU_$5103_WR{L35}
+CPU_$5xxx{K27}
+CPU_$x1xx{J48}
+CPU_$xx0x{K25}
+CPU_$xxx4{J26}
K28
NAND_A
/CPU_$5104{K28}
L36
NOR_2
/Reg_Access{K30}
+Reg_Write{L31}
L37
NAND_2
+CPU_$5104_Access{L36}
/CPU_$5104_WR{L37}
+CPU_$5xxx{K27}
+CPU_$x1xx{J48}
+CPU_$xx0x{K25}
+CPU_$xxx5{J24}
K29
NAND_A
/CPU_$5105{K29}
L39
NOR_2
/Reg_Access{K30}
+Reg_Write{L43}
L38
NAND_2
+CPU_$5105_Access{L39}
/CPU_$5105_WR{L38}
+CPU_$5xxx{K27}
+CPU_$x1xx{K34}
+CPU_$xx0x{K25}
+CPU_$xxx6{J22}
K31
NAND_A
/CPU_$5106{K31}
L41
NOR_2
/Reg_Access{K30}
+Reg_Write{L43}
L40
NAND_2
+CPU_$5106_Access{L41}
/CPU_$5106_WR{L40}
+CPU_$5xxx{K27}
+CPU_$x1xx{K34}
+CPU_$xx0x{K25}
+CPU_$xxx7{J20}
K32
NAND_A
/CPU_$5107{K32}
L42
NOR_2
/Reg_Access{K36}
+Reg_Write{L43}
L44
NAND_2
+CPU_$5107_Access{L42}
/CPU_$5107_WR{L44}
+CPU_$5xxx{K39}
+CPU_$x1xx{K34}
+CPU_$xx1x{J40}
+CPU_$xxx3{J28}
K33
NAND_A
/CPU_$5113{K33}
L45
NOR_2
/Reg_Access{K36}
+Reg_Write{L43}
L46
NAND_2
+CPU_$5113_Access{L45}
/CPU_$5113_WR{L46}
+CPU_$5xxx{K39}
+CPU_$x1xx{K34}
+CPU_$xx1x{J40}
+CPU_$xxx4{J26}
K35
NAND_A
/CPU_$5114{K35}
L48
NOR_2
/Reg_Access{K36}
+Reg_Write{L43}
L47
NAND_2
+CPU_$5114_Access{L48}
/CPU_$5114_WR{L47}
+CPU_$5xxx{K39}
+CPU_$x1xx{K34}
+CPU_$xx1x{K37}
+CPU_$xxx5{J24}
K38
NAND_A
/CPU_$5115{K38}
L50
NOR_2
/Reg_Access{K36}
+Reg_Write{L43}
L49
NAND_2
+CPU_$5115_Access{L50}
/CPU_$5115_WR{L49}
+CPU_$5xxx{K39}
+CPU_$x1xx{K34}
+CPU_$xx1x{K37}
+CPU_$xxx6{J22}
K40
NAND_A
/CPU_$5116{K40}
L51
NOR_2
/Reg_Access{K36}
+Reg_Write{L57}
L52
NAND_2
+CPU_$5116_Access{L51}
/CPU_$5116_WR{L52}
+CPU_$5xxx{K39}
+CPU_$x1xx{K34}
+CPU_$xx1x{K37}
+CPU_$xxx7{J20}
K41
NAND_A
/CPU_$5117{K41}
L53
NOR_2
/Reg_Access{K36}
+Reg_Write{L57}
L54
NAND_2
+CPU_$5117_Access{L53}
/CPU_$5117_WR{L54}
+CPU_$5xxx{K39}
+CPU_$x1xx{K44}
+CPU_$xx0x{J42}
+CPU_$xxx0{J34}
K42
NAND_A
/CPU_$5120{K42}
L55
NOR_2
/Reg_Access{K48}
+Reg_Write{L57}
L56
NAND_2
+CPU_$5120_Access{L55}
/CPU_$5120_WR{L56}
+CPU_$5xxx{K47}
+CPU_$x1xx{K44}
+CPU_$xx2x{J38}
+CPU_$xxx1{J32}
K43
NAND_A
/CPU_$5121{K43}
L59
NOR_2
/Reg_Access{K48}
+Reg_Write{L57}
L58
NAND_2
+CPU_$5121_Access{L59}
/CPU_$5121_WR{L58}
+CPU_$5xxx{K47}
+CPU_$x1xx{K44}
+CPU_$xx2x{J38}
+CPU_$xxx2{J30}
K45
NAND_A
/CPU_$5122{K45}
L63
NOR_2
/Reg_Access{K48}
+Reg_Write{L57}
L62
NAND_2
+CPU_$5122_Access{L63}
/CPU_$5122_WR{L62}
+CPU_$5xxx{K47}
+CPU_$x1xx{K44}
+CPU_$xx2x{J38}
+CPU_$xxx3{J28}
K46
NAND_A
/CPU_$5123{K46}
L65
NOR_2
/Reg_Access{K48}
+Reg_Write{L57}
L64
NAND_2
+CPU_$5123_Access{L65}
/CPU_$5123_WR{L64}
+CPU_$5xxx{K47}
+CPU_$x1xx{K44}
+CPU_$xx2x{J38}
+CPU_$xxx4{J26}
K49
NAND_A
/CPU_$5124{K49}
L67
NOR_2
/Reg_Access{K48}
+Reg_Write{L72}
L66
NAND_2
+CPU_$5124_Access{L67}
/CPU_$5124_WR{L66}
+CPU_$5xxx{K47}
+CPU_$x1xx{K53}
+CPU_$xx2x{J38}
+CPU_$xxx5{J24}
K50
NAND_A
/CPU_$5125{K50}
L68
NOR_2
/Reg_Access{K48}
+Reg_Write{L72}
L69
NAND_2
+CPU_$5125_Access{L68}
/CPU_$5125_WR{L69}
+CPU_$5xxx{K47}
+CPU_$x1xx{K53}
+CPU_$xx2x{J38}
+CPU_$xxx6{J22}
K52
NAND_A
/CPU_$5126{K52}
L71
NOR_2
/Reg_Access{K63}
+Reg_Write{L72}
L70
NAND_2
+CPU_$5126_Access{L71}
/CPU_$5126_WR{L70}
+CPU_$5xxx{K59}
+CPU_$x1xx{K53}
+CPU_$xx2x{K51}
+CPU_$xxx7{J20}
K54
NAND_A
/CPU_$5127{K54}
L73
NOR_2
/Reg_Access{K63}
+Reg_Write{L72}
L74
NAND_2
+CPU_$5127_Access{L73}
/CPU_$5127_WR{L74}
+CPU_$5xxx{K84}
+CPU_$x1xx{K53}
+CPU_$xx2x{K51}
+CPU_$xxx8{J18}
K81
NAND_A
/CPU_$5128{K81}
L97
NOR_2
/Reg_Access{H2}
+Reg_Write{L96}
L98
NAND_2
+CPU_$5128_Access{L97}
/CPU_$5128_WR{L98}
+CPU_$5xxx{K84}
+CPU_$x1xx{K82}
+CPU_$xx2x{K51}
+CPU_$xxx9{J16}
K83
NAND_A
/CPU_$5129{K83}
L100
NOR_2
/Reg_Access{H2}
+Reg_Write{L96}
L99
NAND_2
+CPU_$5129_Access{L100}
/CPU_$5129_WR{L99}
+CPU_$5xxx{K84}
+CPU_$x1xx{K82}
+CPU_$xx2x{K51}
+CPU_$xxxA{J12}
K85
NAND_A
/CPU_$512A{K85}
L101
NOR_2
/Reg_Access{H2}
+Reg_Write{L96}
L102
NAND_2
+CPU_$512A_Access{L101}
/CPU_$512A_WR{L102}
+CPU_$5xxx{K84}
+CPU_$x1xx{K82}
+CPU_$xx2x{K51}
+CPU_$xxxB{J10}
K86
NAND_A
/CPU_$512B{K86}
L103
NOR_2
/Reg_Access{H2}
+Reg_Write{L111}
L104
NAND_2
+CPU_$512B_Access{L103}
/CPU_$512B_WR{L104}
+CPU_$5xxx{K59}
+CPU_$x1xx{K53}
+CPU_$xx3x{J36}
+CPU_$xxx0{K55}
K56
NAND_A
/CPU_$5130{K56}
L75
NOR_2
/Reg_Access{K63}
+Reg_Write{L72}
L76
NAND_2
+CPU_$5130_Access{L75}
/CPU_$5130_WR{L76}
+CPU_$5xxx{K59}
+CPU_$x2xx{J46}
+CPU_$xx0x{K57}
+CPU_$xxx0{K55}
K58
NAND_A
/CPU_$5200{K58}
L77
NOR_2
/Reg_Access{K63}
+Reg_Write{L72}
L78
NAND_2
+CPU_$5200_Access{L77}
/CPU_$5200_WR{L78}
+CPU_$5xxx{K59}
+CPU_$x2xx{J46}
+CPU_$xx0x{K57}
+CPU_$xxx1{J32}
K60
NAND_A
/CPU_$5201{K60}
L80
NOR_2
/Reg_Access{K63}
+Reg_Write{L85}
L79
NAND_2
+CPU_$5201_Access{L80}
/CPU_$5201_WR{L79}
+CPU_$5xxx{K59}
+CPU_$x2xx{J46}
+CPU_$xx0x{K57}
+CPU_$xxx2{J30}
K64
NAND_A
/CPU_$5202{K64}
L81
NOR_2
/Reg_Access{K63}
+Reg_Write{L85}
L82
NAND_2
+CPU_$5202_Access{L81}
/CPU_$5202_WR{L82}
+CPU_$5xxx{K59}
+CPU_$x2xx{J46}
+CPU_$xx0x{K57}
+CPU_$xxx3{J28}
K65
NAND_A
/CPU_$5203{K65}
L84
NOR_2
/Reg_Access{K74}
+Reg_Write{L85}
L83
NAND_2
+CPU_$5203_Access{L84}
/CPU_$5203_WR{L83}
+CPU_$5xxx{K70}
+CPU_$x2xx{J46}
+CPU_$xx0x{K57}
+CPU_$xxx4{J26}
K66
NAND_A
/CPU_$5204{K66}
+Reg_Read{K3}
L88
NOR_2
/Reg_Access{K74}
+Reg_Write{L85}
L86
NAND_2
/CPU_$5204_RD{L86}
+CPU_$5204_Access{L88}
L87
NAND_2
/CPU_$5204_WR{L87}
+CPU_$5xxx{K70}
+CPU_$x2xx{J46}
+CPU_$xx0x{K57}
+CPU_$xxx5{J24}
K67
NAND_A
/CPU_$5205{K67}
+Reg_Read{K3}
L91
NOR_2
/Reg_Access{K74}
+Reg_Write{L85}
L90
NAND_2
/CPU_$5205_RD{L90}
+CPU_$5205_Access{L91}
L89
NAND_2
/CPU_$5205_WR{L89}
+CPU_$5xxx{K70}
+CPU_$x2xx{J46}
+CPU_$xx0x{K68}
+CPU_$xxx6{J22}
K69
NAND_A
/CPU_$5206{K69}
+Reg_Read{K3}
L92
NOR_2
/Reg_Access{K74}
+Reg_Write{L85}
L94
NAND_2
/CPU_$5206_RD{L94}
+CPU_$5206_Access{L92}
L93
NAND_2
/CPU_$5206_WR{L93}
+CPU_$5xxx{K84}
+CPU_$xCxx-xFxx{J44}
K95
NAND_2
/Reg_Access{K98}
L117
NOR_2
+CPU_ExRAM_Access{L117}
+CPU_A1{H5}
/CPU_A2{H8}
+CPU_A3{H9}
+CPU_A4{H11}
+CPU_A5{H13}
+CPU_A6{H15}
+CPU_A7{H17}
J4
NAND_7
+CPU_A8{H19}
+CPU_A9{H21}
+CPU_A10{H23}
+CPU_A11{H25}
+CPU_A12{H27}
+CPU_A13{H29}
+CPU_A14{H31}
J5
NAND_7
J6
NOR_2
J7
INVERT
+Reg_Access{H36}
J8
NOR_2
+Reg_Read{K3}
+CPU_$FFFA/B_Access{J8}
L21
NAND_2
/CPU_$FFFA/B_RD{L21}
/CPU_$5120_WR{L56}
/CPU_$5121_WR{L58}
/CPU_$5122_WR{L62}
/CPU_$5123_WR{L64}
/CPU_$5124_WR{L66}
/CPU_$5125_WR{L69}
/CPU_$5126_WR{L70}
/CPU_$5127_WR{L74}
AF1
NAND_8
AE1
INVERT
/CPU_$5120/1/2/3/4/5/6/7_WR{AE1}
/CPU_$5128_WR{L98}
/CPU_$5129_WR{L99}
/CPU_$512A_WR{L102}
/CPU_$512B_WR{L104}
AF2
NAND_A
AE2
INVERT
/CPU_$5128/9/A/B_WR{AE2}

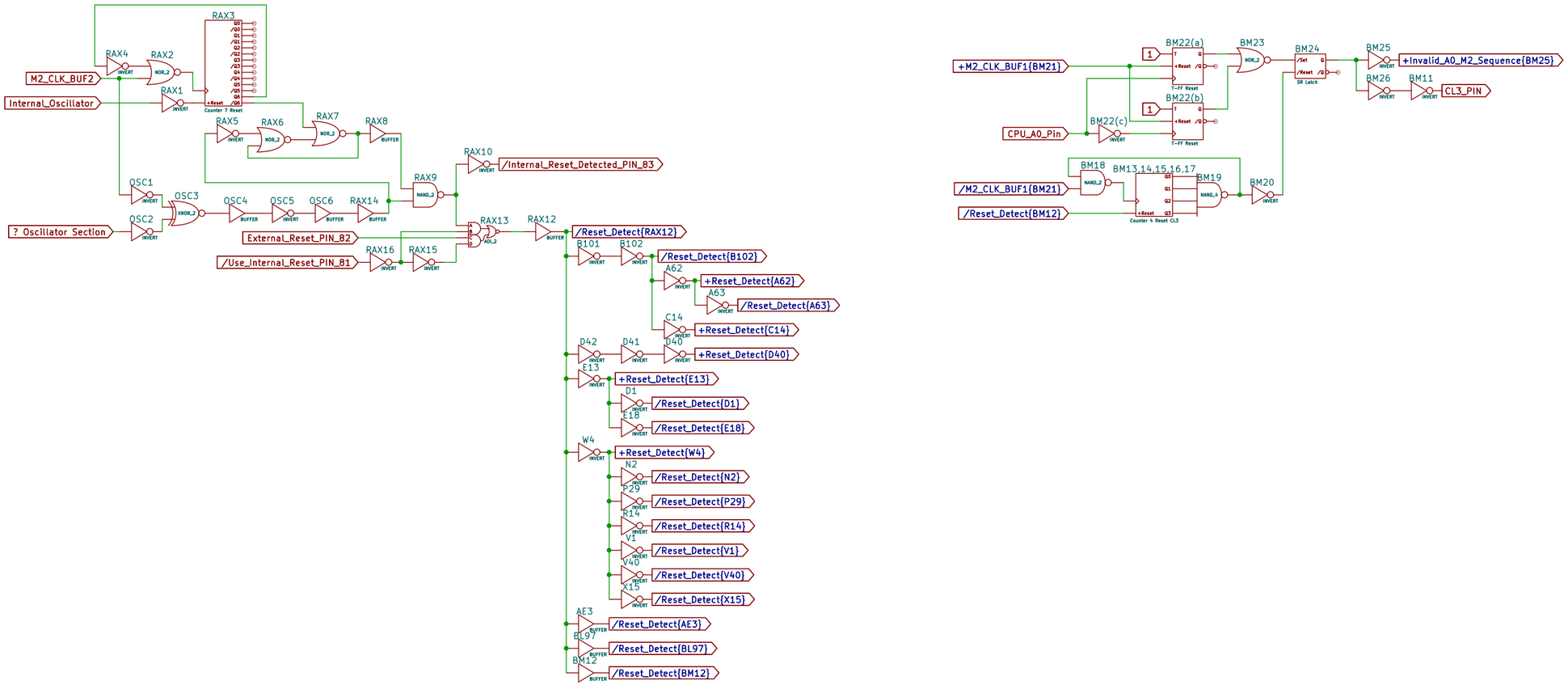
PPU Address Bus Buffering



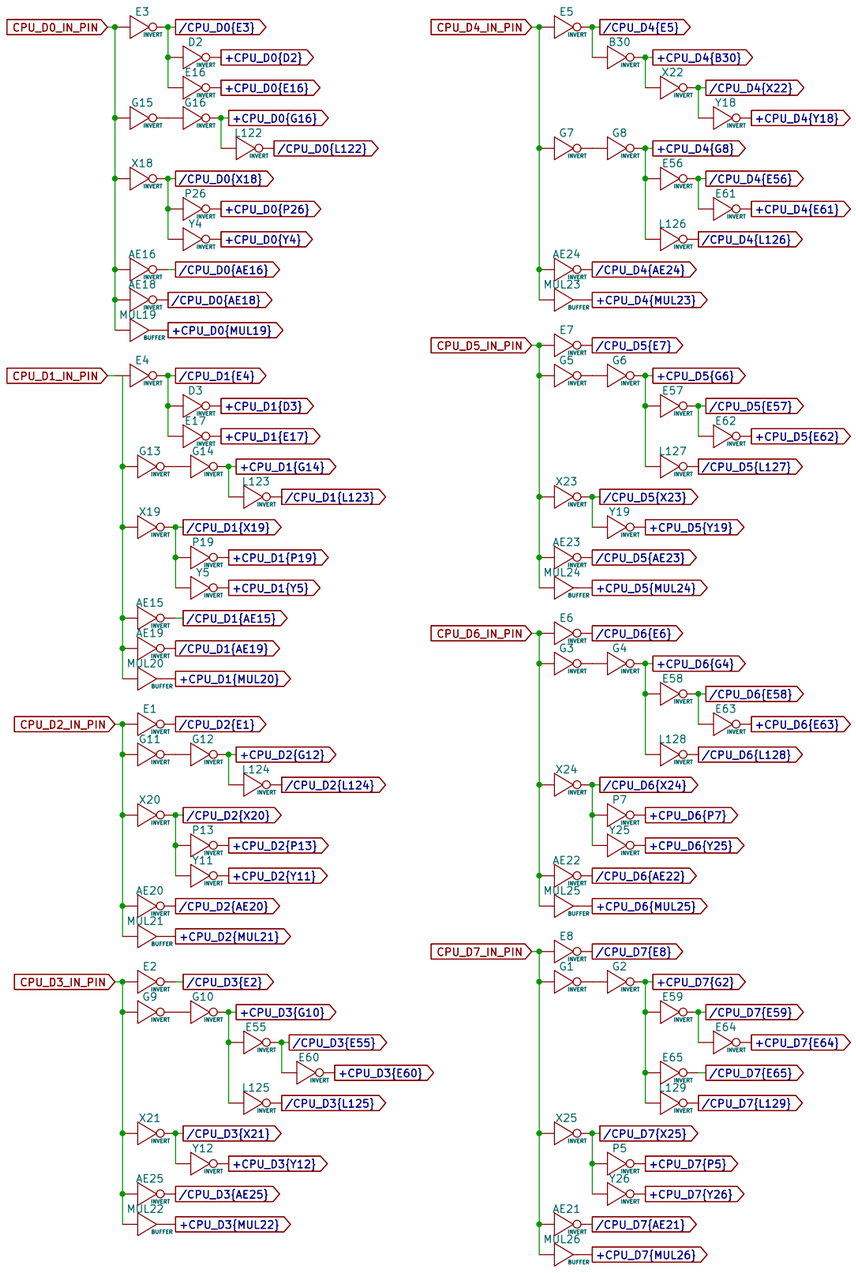
Note:
/PPU_WR_PIN has an inverter
at the pin but does not go
to any buffers beyond that.


$5105 Register
Nametable Mapping
CPU_D0_IN_PIN
CPU_D1_IN_PIN
CPU_D2_IN_PIN
CPU_D3_IN_PIN
CPU_D4_IN_PIN
CPU_D5_IN_PIN
CPU_D6_IN_PIN
CPU_D7_IN_PIN
/CPU_$5105_WR{L38}
C1
Reg 8 D-Latch N
B10
B3
B4
B11
B5
B6
B12
B7
B8
B13
A17
B9
/$5105.0{B10}
+$5105.0{B3}
+$5105.1{B4}
/$5105.2{B11}
+$5105.2{B5}
+$5105.3{B6}
/$5105.4{B12}
+$5105.4{B7}
+$5105.5{B8}
/$5105.6{B13}
+$5105.6{A17}
+$5105.7{B9}
/Scanline_Counter_Reset{Z25}
/ExRAM_Mode_%01{BK80}
+CounterS.3{S21}
V-Split_Side_Latched{N15}
Order Not Clear
A11
A12
A10
A15
A16
CIRAM_CE_Unmapped{A10}
/PPU_A13_B{B1}
+PPU_A12_PIN_A{PinBuf}
B2
/PPU_$1000-1FFF
A14
A13
Question: Why is this using PPU A12?
Without it, the range would be $0000-1FFF
Which would make a clearer distinction
between pattern tables and nametables.
/PPU_A11_A{A1}
/PPU_A10_A{A3}
+PPU_A10_A{A4}
+PPU_A11_A{A2}
A5
A6
A7
A8
/PPU_NT0_$x000-x3FF
/PPU_NT1_$x400-x7FF
/PPU_NT2_$x800-xBFF
/PPU_NT3_$xC00-xFFF
B14
B15
B16
B17
B18
B19
B20
B21
B22
B23
B24
B25
A9
/$5105.6{B13}
/$5105.4{B12}
/$5105.2{B11}
/$5105.0{B10}
CIRAM_A10_PIN{A9}
CIRAM_CE_PIN{A9}
+ExRAM_Nametable_Access{A9}
+Fill_Mode_Nametable_Access{A9}
BL11
BL33
+Fill_Mode_Nametable_Access{BL33}
Note: There is probably
an inverter at each
output pin.
4x 4to1 Multiplexer
I believe this uses
inverting tristate buffers.

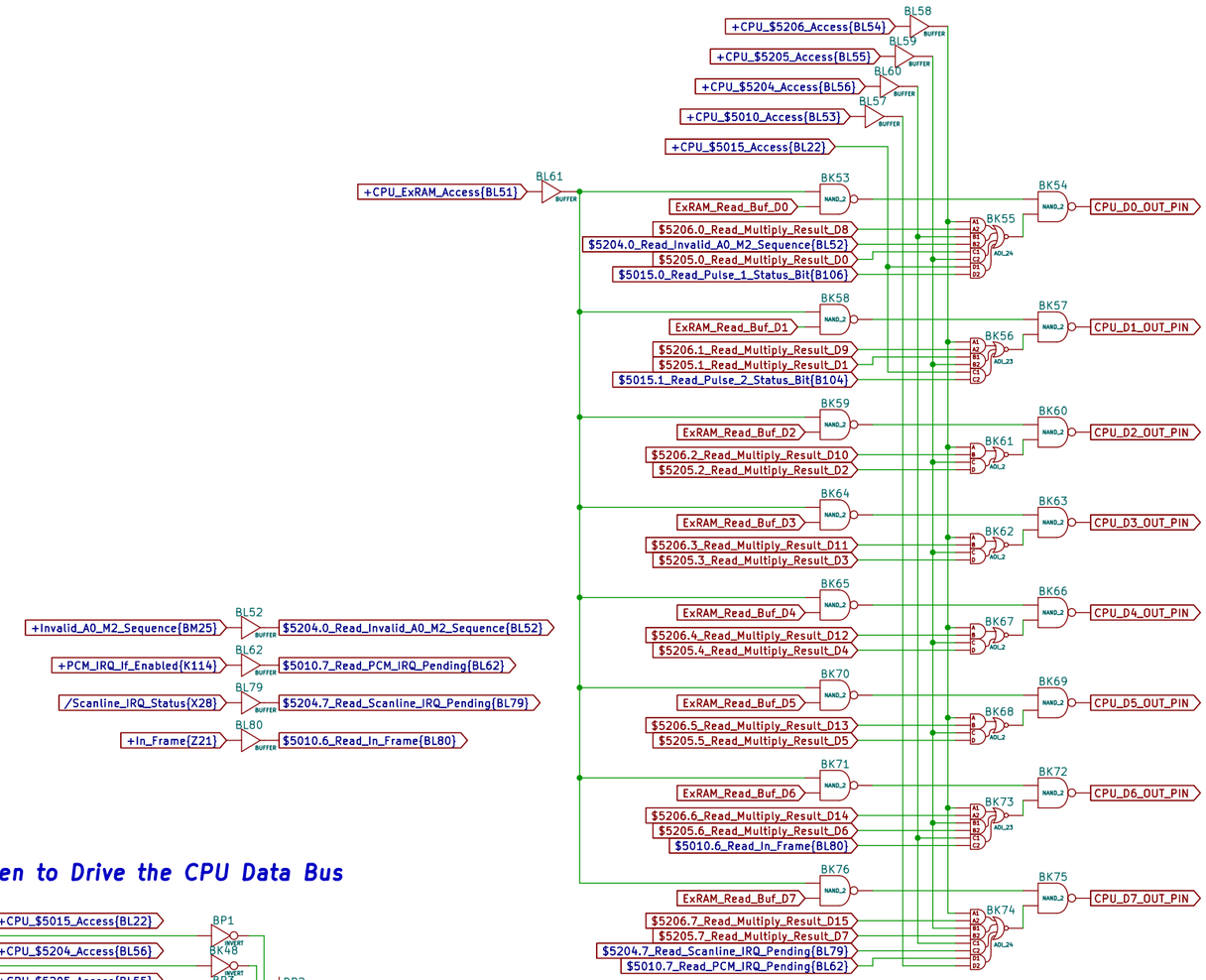

$5100 Register
PRG Mode
Reg 2 DFF-Set
PRG Bank Registers
$5113 Register
Reg 4 D-Latch
$5114 Register
Reg 8 D-Latch
$5115 Register
Reg 8 D-Latch
$5116 Register
Reg 8 D-Latch
$5117 Register
Reg 8 DFF-Set
Note: $5117.7 RAM/ROM bit is fully implemented but is somehow being prevented from entering RAM mode when bench testing.
Banks A14 for 8k & 16k banks.
Banks A13 for 8k banks.
Passes through A14 for 32k banks.
Passes through A14 for 32k banks.
Passes through A13 for 16k banks.
Note: There is probably an inverter at each output pin.
$5102 Register
PRG-RAM Write Protection
Reg 2 DFF-Reset N
Unlocked when E22=1
Q becomes locked asynchronously, becomes unlocked synchronized to M2
$5103 Register
Reg 2 DFF-Reset N
Note: /Set vs. /Reset inputs are individually used but not easily shown here with a symbol.


M2_CLK_BUF2
Internal_Oscillator
? Oscillator Section
RAX4
RAX2
RAX1
RAX3
Counter 7 Reset
RAX5
RAX6
RAX7
RAX8
OSC1
OSC2
OSC3
OSC4
OSC5
OSC6
RAX14
RAX9
RAX10
/Internal_Reset_Detected_PIN_83
External_Reset_PIN_82
/Use_Internal_Reset_PIN_81
RAX16
RAX15
RAX13
RAX12
/Reset_Detect{RAX12}
B101
B102
/Reset_Detect{B102}
A62
+Reset_Detect{A62}
A63
/Reset_Detect{A63}
C14
+Reset_Detect{C14}
D42
D41
D40
+Reset_Detect{D40}
E13
+Reset_Detect{E13}
D1
/Reset_Detect{D1}
E18
/Reset_Detect{E18}
W4
+Reset_Detect{W4}
N2
/Reset_Detect{N2}
P29
/Reset_Detect{P29}
R14
/Reset_Detect{R14}
V1
/Reset_Detect{V1}
V40
/Reset_Detect{V40}
X15
/Reset_Detect{X15}
AE3
/Reset_Detect{AE3}
BL97
/Reset_Detect{BL97}
BM12
/Reset_Detect{BM12}
+M2_CLK_BUF1{BM21}
CPU_A0_Pin
BM22(a)
BM22(b)
BM22(c)
T-FF Reset
BM23
BM24
SR Latch
BM25
+Invalid_A0_M2_Sequence{BM25}
BM26
BM11
CL3_PIN
/M2_CLK_BUF1{BM21}
/Reset_Detect{BM12}
BM18
BM13,14,15,16,17
Counter 4 Reset CL3
BM19
BM20

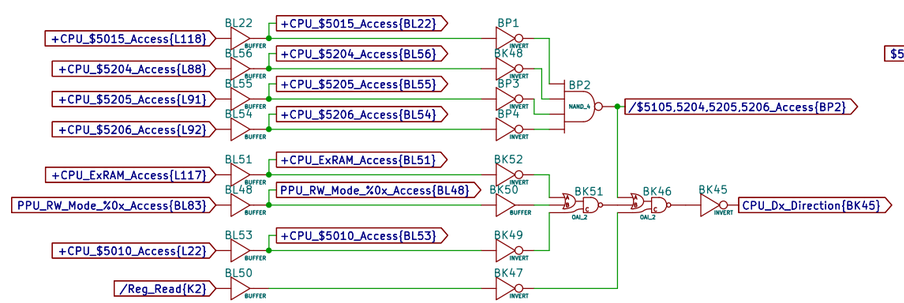
Incoming CPU Data Bus Buffering Bonanza



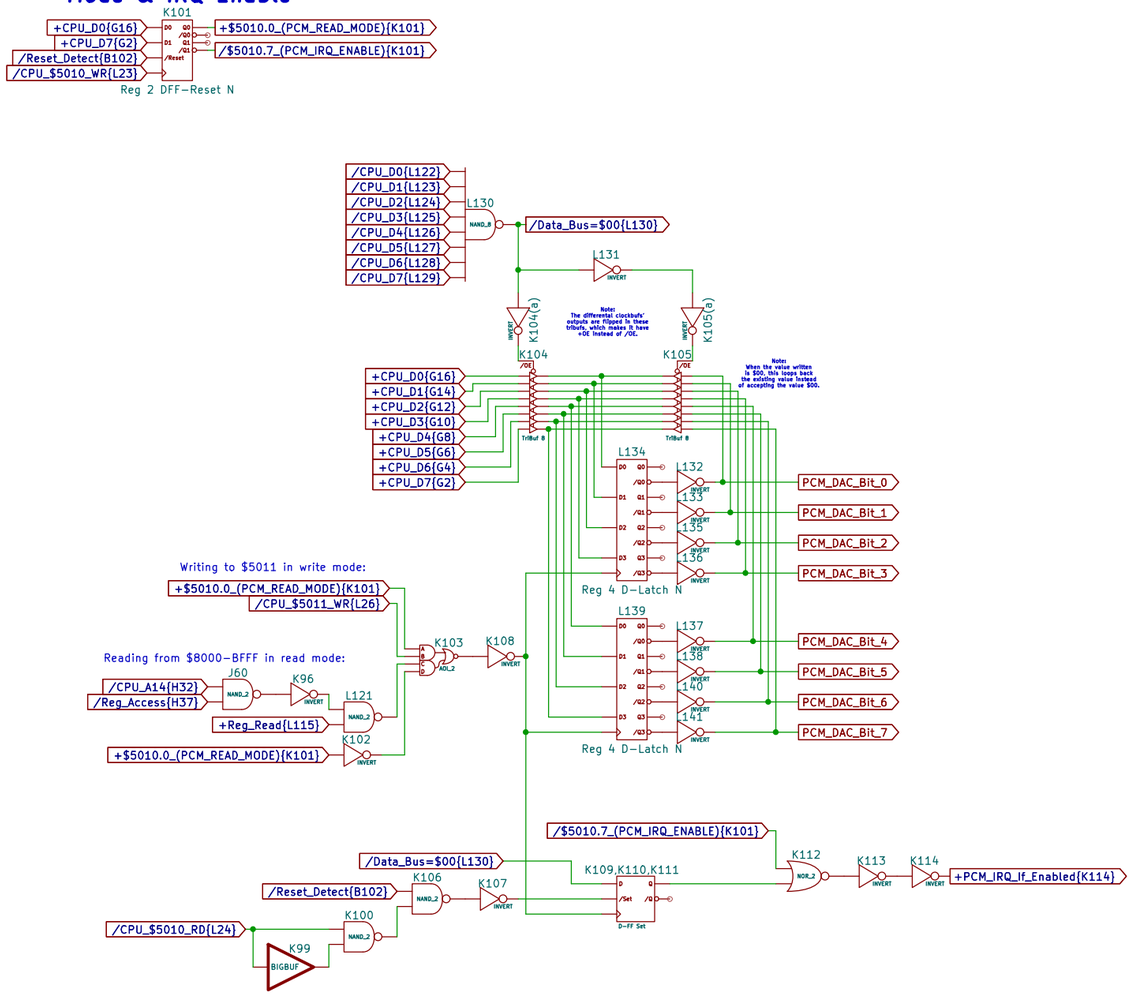
Multiplexing Data to be Driven onto the CPU Data Bus

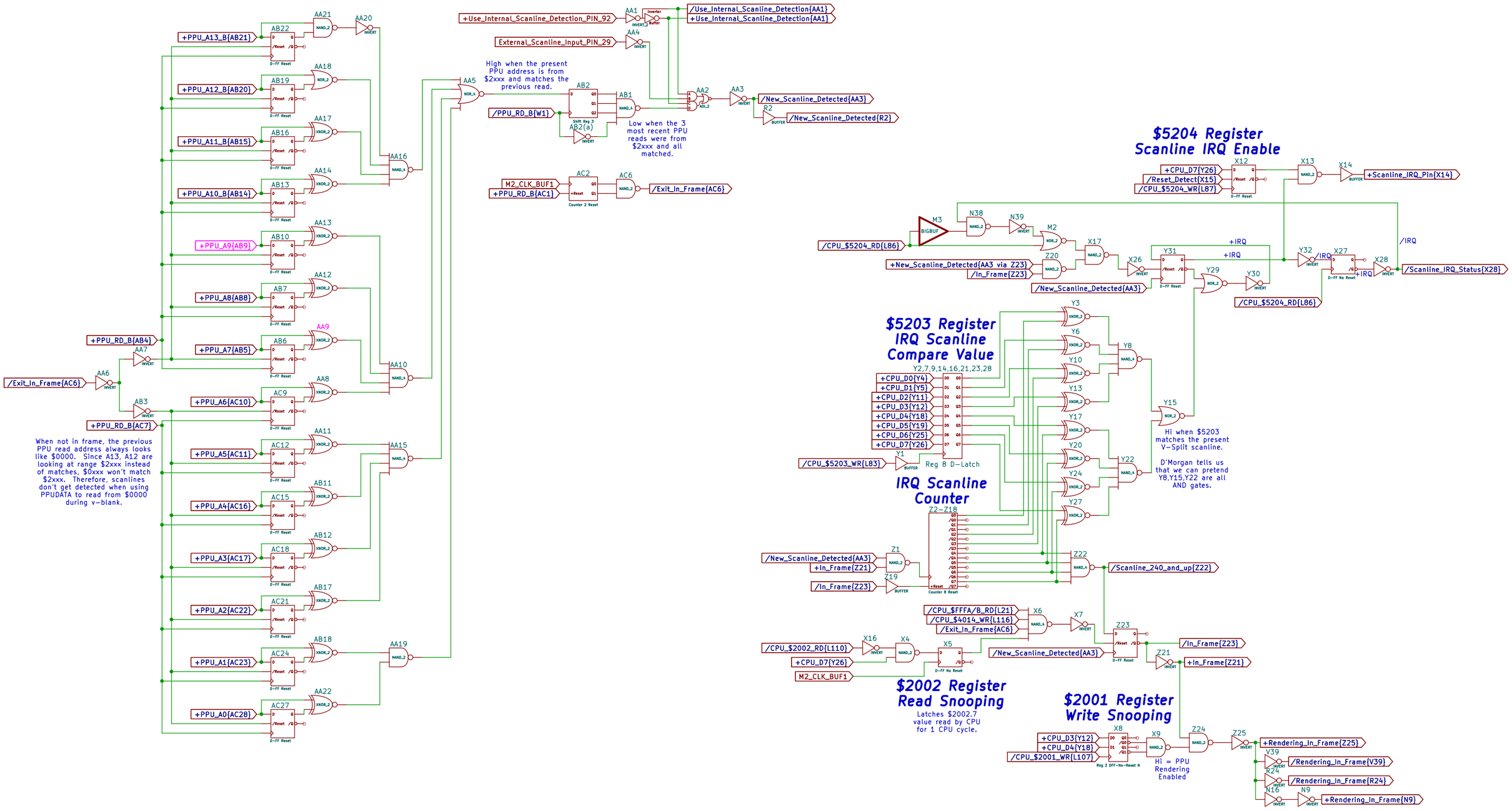


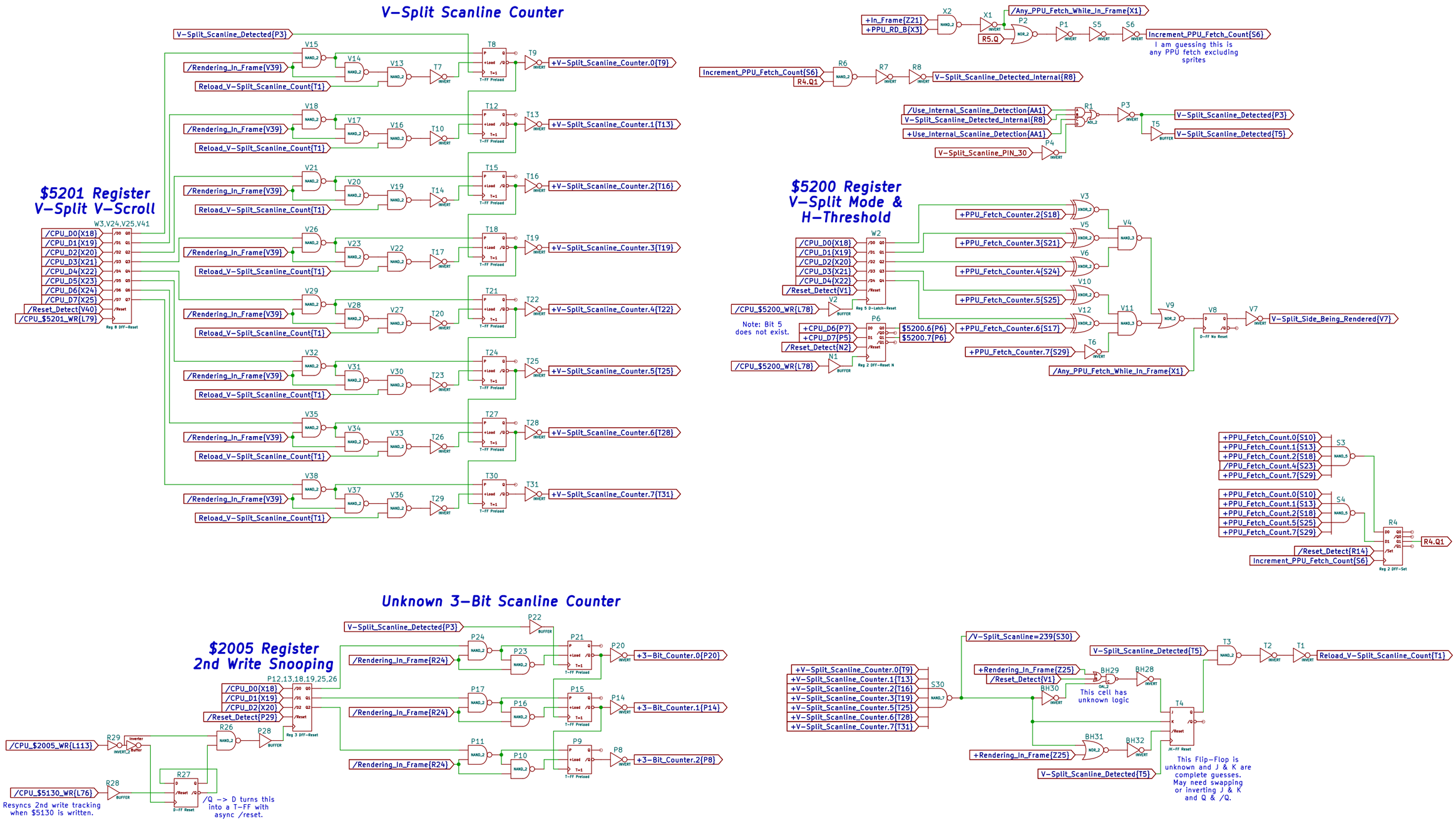
Decides When to Drive the CPU Data Bus



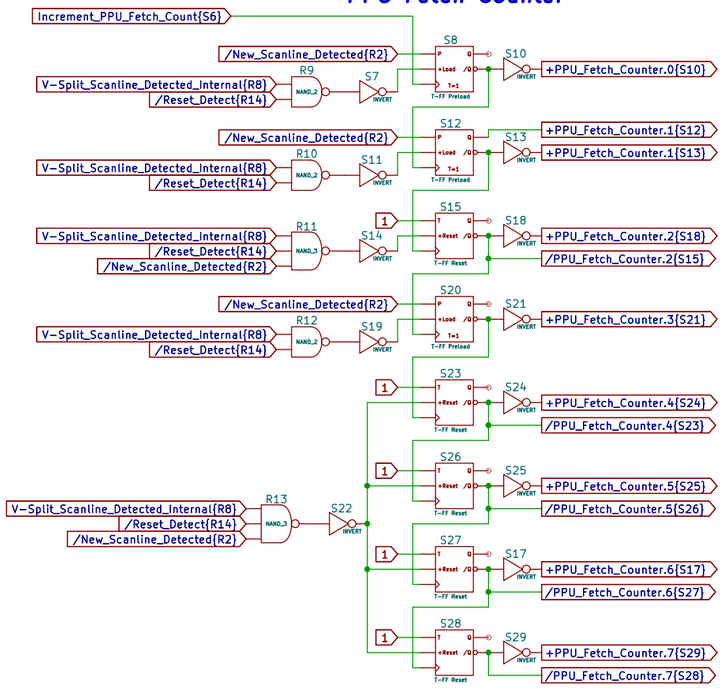
$5010 Register
Mode & IRQ Enable

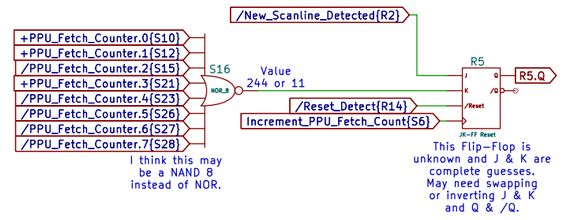


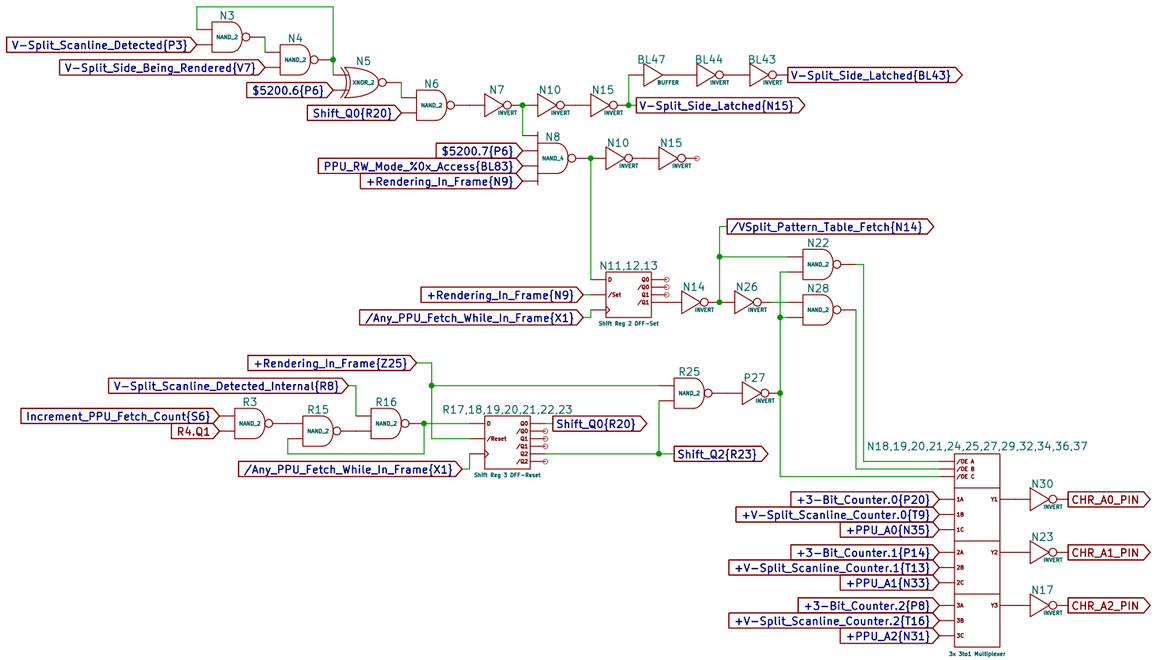

+PPU_A13_B{AB21}
+PPU_A12_B{AB20}
+PPU_A11_B{AB15}
+PPU_A10_B{AB14}
+PPU_A9{AB9}
+PPU_A8{AB8}
+PPU_A7{AB5}
+PPU_A6{AC10}
+PPU_A5{AC11}
+PPU_A4{AC16}
+PPU_A3{AC17}
+PPU_A2{AC22}
+PPU_A1{AC23}
+PPU_A0{AC28}
+PPU_RD_B{AB4}
+PPU_RD_B{AC7}
/Exit_In_Frame{AC6}
When not in frame, the previous PPU read address always looks like $0000. Since A13, A12 are looking at range $2xxx instead of matches, $0xxx won't match $2xxx. Therefore, scanlines don't get detected when using PPUDATA to read from $0000 during v-blank.
+Use_Internal_Scanline_Detection_PIN_92
External_Scanline_Input_PIN_29
/Use_Internal_Scanline_Detection{AA1}
+Use_Internal_Scanline_Detection{AA1}
/New_Scanline_Detected{AA3}
/New_Scanline_Detected{R2}
High when the present PPU address is from $2xxx and matches the previous read.
/PPU_RD_B{W1}
Low when the 3 most recent PPU reads were from $2xxx and all matched.
M2_CLK_BUF1
+PPU_RD_B{AC1}
/Exit_In_Frame{AC6}
$5204 Register
Scanline IRQ Enable
+CPU_D7{Y26}
/Reset_Detect{X15}
/CPU_$5204_WR{L87}
+Scanline_IRQ_Pin{X14}
/CPU_$5204_RD{L86}
+New_Scanline_Detected{AA3 via Z23}
/In_Frame{Z23}
/New_Scanline_Detected{AA3}
+IRQ
/IRQ
/Scanline_IRQ_Status{X28}
$5203 Register
IRQ Scanline
Compare Value
Y2,7,9,14,16,21,23,28
+CPU_D0{Y4}
+CPU_D1{Y5}
+CPU_D2{Y11}
+CPU_D3{Y12}
+CPU_D4{Y18}
+CPU_D5{Y19}
+CPU_D6{Y25}
+CPU_D7{Y26}
/CPU_$5203_WR{L83}
Reg 8 D-Latch
Hi when $5203 matches the present V-Split scanline.
D'Morgan tells us that we can pretend Y8,Y15,Y22 are all AND gates.
IRQ Scanline
Counter
Z2-Z18
/New_Scanline_Detected{AA3}
+In_Frame{Z21}
/In_Frame{Z23}
/Scanline_240_and_up{Z22}
/CPU_$FFFA/B_RD{L21}
/CPU_$4014_WR{L116}
/Exit_In_Frame{AC6}
/CPU_$2002_RD{L110}
+CPU_D7{Y26}
M2_CLK_BUF1
/New_Scanline_Detected{AA3}
/In_Frame{Z23}
+In_Frame{Z21}
$2002 Register
Read Snooping
Latches $2002.7 value read by CPU for 1 CPU cycle.
$2001 Register
Write Snooping
+CPU_D3{Y12}
+CPU_D4{Y18}
/CPU_$2001_WR{L107}
Hi = PPU Rendering Enabled
+Rendering_In_Frame{Z25}
/Rendering_In_Frame{V39}
/Rendering_In_Frame{R24}
+Rendering_In_Frame{N9}


V-Split Scanline Counter
$5201 Register
V-Split V-Scroll
W3,V24,V25,V41
/CPU_D0{X18}
/CPU_D1{X19}
/CPU_D2{X20}
/CPU_D3{X21}
/CPU_D4{X22}
/CPU_D5{X23}
/CPU_D6{X24}
/CPU_D7{X25}
/Reset_Detect{V40}
/CPU_$5201_WR{L79}
Reg 8 DFF-Reset
V-Split_Scanline_Detected{P3}
/Rendering_In_Frame{V39}
Reload_V-Split_Scanline_Count{T1}
+V-Split_Scanline_Counter.0{T9}
+V-Split_Scanline_Counter.1{T13}
+V-Split_Scanline_Counter.2{T16}
+V-Split_Scanline_Counter.3{T19}
+V-Split_Scanline_Counter.4{T22}
+V-Split_Scanline_Counter.5{T25}
+V-Split_Scanline_Counter.6{T28}
+V-Split_Scanline_Counter.7{T31}
+In_Frame{Z21}
+PPU_RD_B{X3}
/Any_PPU_Fetch_While_In_Frame{X1}
R5.Q
Increment_PPU_Fetch_Count{S6}
I am guessing this is any PPU fetch excluding sprites
Increment_PPU_Fetch_Count{S6}
R4.Q1
V-Split_Scanline_Detected_Internal{R8}
/Use_Internal_Scanline_Detection{AA1}
V-Split_Scanline_Detected_Internal{R8}
+Use_Internal_Scanline_Detection{AA1}
V-Split_Scanline_PIN_30
V-Split_Scanline_Detected{P3}
V-Split_Scanline_Detected{T5}
$5200 Register
V-Split Mode &
H-Threshold
/CPU_D0{X18}
/CPU_D1{X19}
/CPU_D2{X20}
/CPU_D3{X21}
/CPU_D4{X22}
/Reset_Detect{V1}
/CPU_$5200_WR{L78}
Reg 5 D-Latch-Reset
Note: Bit 5 does not exist.
+CPU_D6{P7}
+CPU_D7{P5}
/Reset_Detect{N2}
$5200.6{P6}
$5200.7{P6}
Reg 2 DFF-Reset N
+PPU_Fetch_Counter.2{S18}
+PPU_Fetch_Counter.3{S21}
+PPU_Fetch_Counter.4{S24}
+PPU_Fetch_Counter.5{S25}
+PPU_Fetch_Counter.6{S17}
+PPU_Fetch_Counter.7{S29}
/Any_PPU_Fetch_While_In_Frame{X1}
V-Split_Side_Being_Rendered{V7}
+PPU_Fetch_Count.0{S10}
+PPU_Fetch_Count.1{S13}
+PPU_Fetch_Count.2{S18}
/PPU_Fetch_Count.4{S23}
+PPU_Fetch_Count.7{S29}
+PPU_Fetch_Count.0{S10}
+PPU_Fetch_Count.1{S13}
+PPU_Fetch_Count.2{S18}
+PPU_Fetch_Count.5{S25}
+PPU_Fetch_Count.7{S29}
/Reset_Detect{R14}
Increment_PPU_Fetch_Count{S6}
Reg 2 DFF-Set
Unknown 3-Bit Scanline Counter
$2005 Register
2nd Write Snooping
P12,13,18,19,25,26
/CPU_D0{X18}
/CPU_D1{X19}
/CPU_D2{X20}
/Reset_Detect{P29}
Reg 3 DFF-Reset
/CPU_$2005_WR{L113}
/CPU_$5130_WR{L76}
Resyncs 2nd write tracking when $5130 is written.
/Q -> D turns this into a T-FF with async /reset.
+3-Bit_Counter.0{P20}
+3-Bit_Counter.1{P14}
+3-Bit_Counter.2{P8}
/V-Split_Scanline=239{S30}
+Rendering_In_Frame{Z25}
/Reset_Detect{V1}
This cell has unknown logic
V-Split_Scanline_Detected{T5}
Reload_V-Split_Scanline_Count{T1}
This Flip-Flop is unknown and J & K are complete guesses. May need swapping or inverting J & K and Q & /Q.

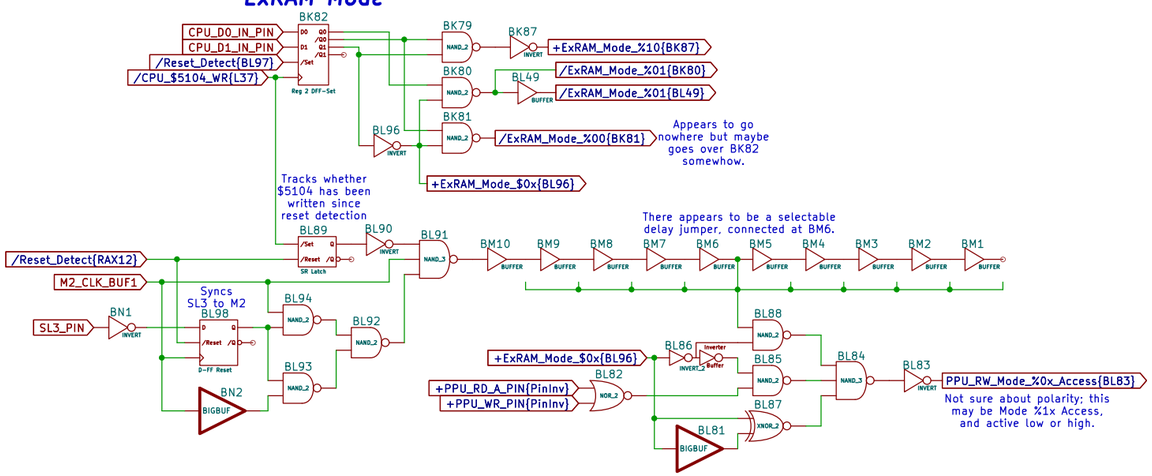
PPU Fetch Counter

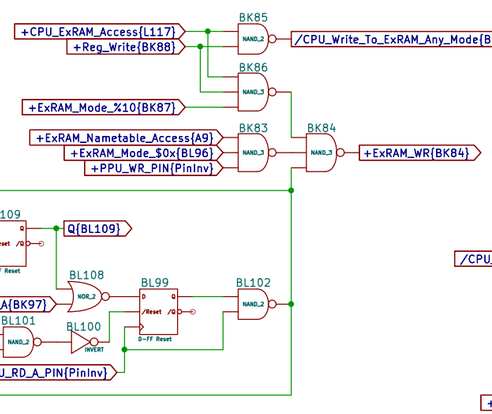


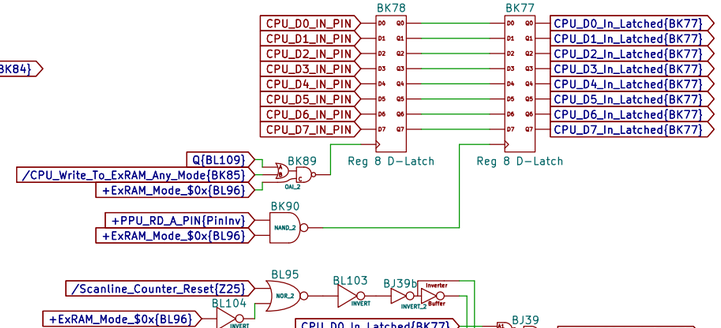


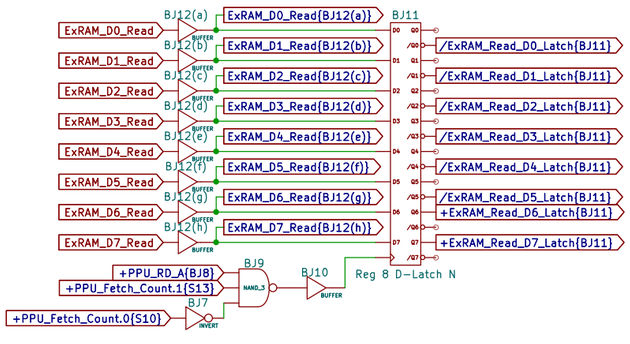

N3
NAND_2
V-Split_Scanline_Detected{P3}
N4
NAND_2
V-Split_Side_Being_Rendered{V7}
N5
XNOR_2
$5200.6{P6}
N6
NAND_2
Shift_Q0{R20}
N7
INVERT
N10
INVERT
N15
INVERT
BL47
BUFFER
BL44
INVERT
BL43
INVERT
V-Split_Side_Latched{BL43}
V-Split_Side_Latched{N15}
N8
NAND_4
$5200.7{P6}
PPU_RW_Mode_%0x_Access{BL83}
+Rendering_In_Frame{N9}
N10
INVERT
N15
INVERT
/VSplit_Pattern_Table_Fetch{N14}
N22
NAND_2
N11,12,13
D
/Set
Q0
/Q0
Q1
/Q1
Shift Reg 2 DFF-Set
+Rendering_In_Frame{N9}
/Any_PPU_Fetch_While_In_Frame{X1}
N14
INVERT
N26
INVERT
N28
NAND_2
+Rendering_In_Frame{Z25}
V-Split_Scanline_Detected_Internal{R8}
Increment_PPU_Fetch_Count{S6}
R4.Q1
R3
NAND_2
R15
NAND_2
R16
NAND_2
R17,18,19,20,21,22,23
D
/Reset
Q0
/Q0
Q1
/Q1
Q2
/Q2
Shift Reg 3 DFF-Reset
Shift_Q0{R20}
Shift_Q2{R23}
/Any_PPU_Fetch_While_In_Frame{X1}
R25
NAND_2
P27
INVERT
N18,19,20,21,24,25,27,29,32,34,36,37
/OE A
/OE B
/OE C
+3-Bit_Counter.0{P20}
+V-Split_Scanline_Counter.0{T9}
+PPU_A0{N35}
+3-Bit_Counter.1{P14}
+V-Split_Scanline_Counter.1{T13}
+PPU_A1{N33}
+3-Bit_Counter.2{P8}
+V-Split_Scanline_Counter.2{T16}
+PPU_A2{N31}
1A
1B
1C
2A
2B
2C
3A
3B
3C
Y1
Y2
Y3
3x 3to1 Multiplexer
N30
INVERT
CHR_A0_PIN
N23
INVERT
CHR_A1_PIN
N17
INVERT
CHR_A2_PIN

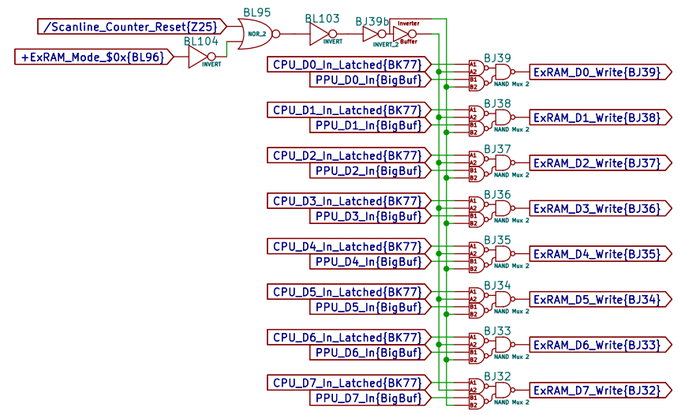
$5104 Register
ExRAM Mode



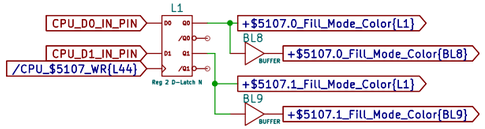




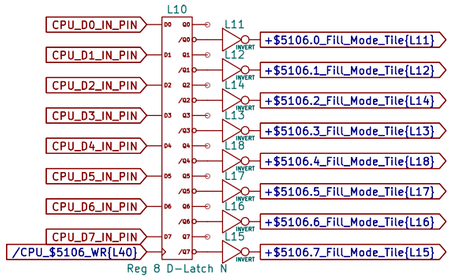




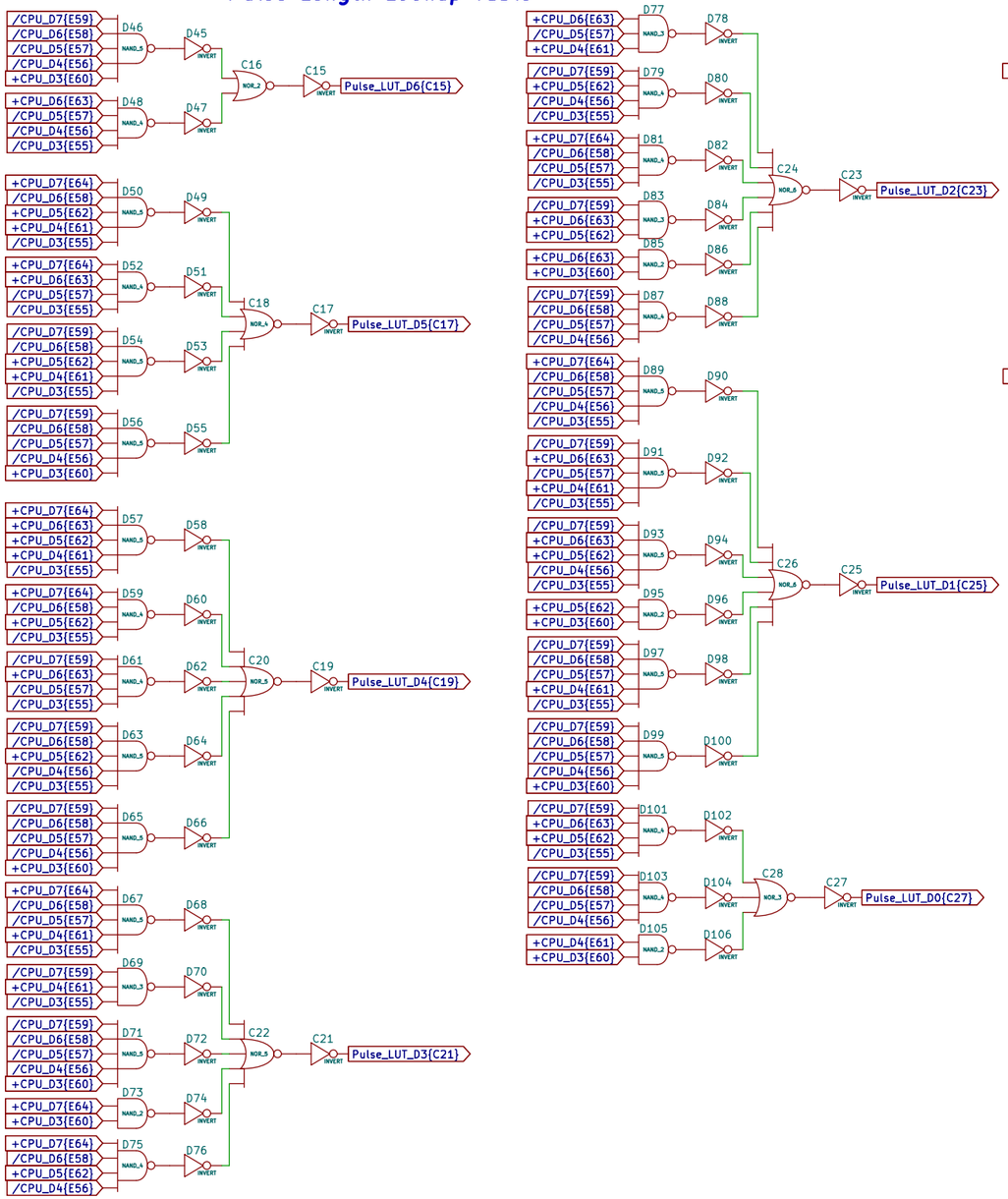
$5107 Register
Fill-Mode Color



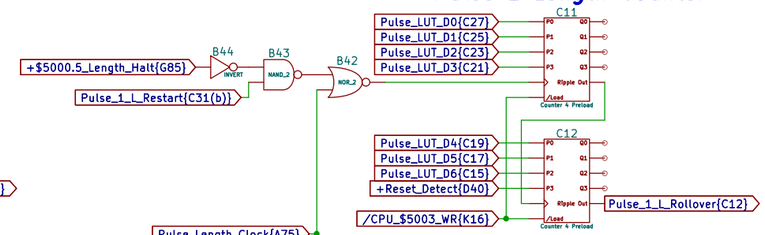
$5106 Register
Fill-Mode Tile



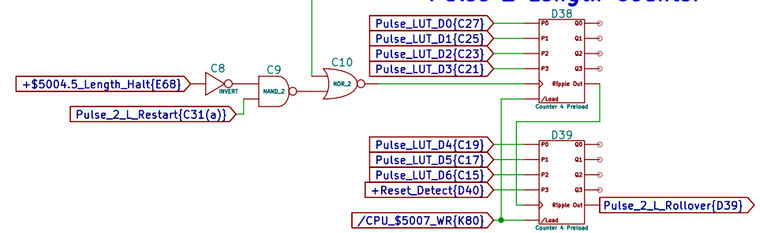
5-Bit to 7-Bit Pulse Length Lookup Table



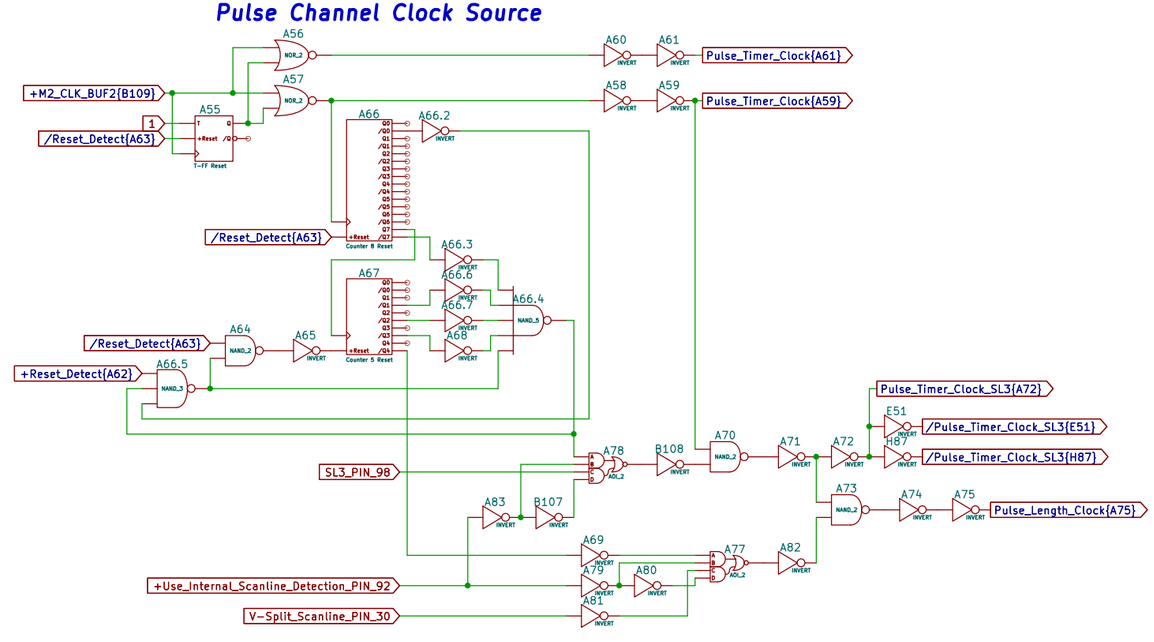
$5003 Register Pulse 1 Length Counter

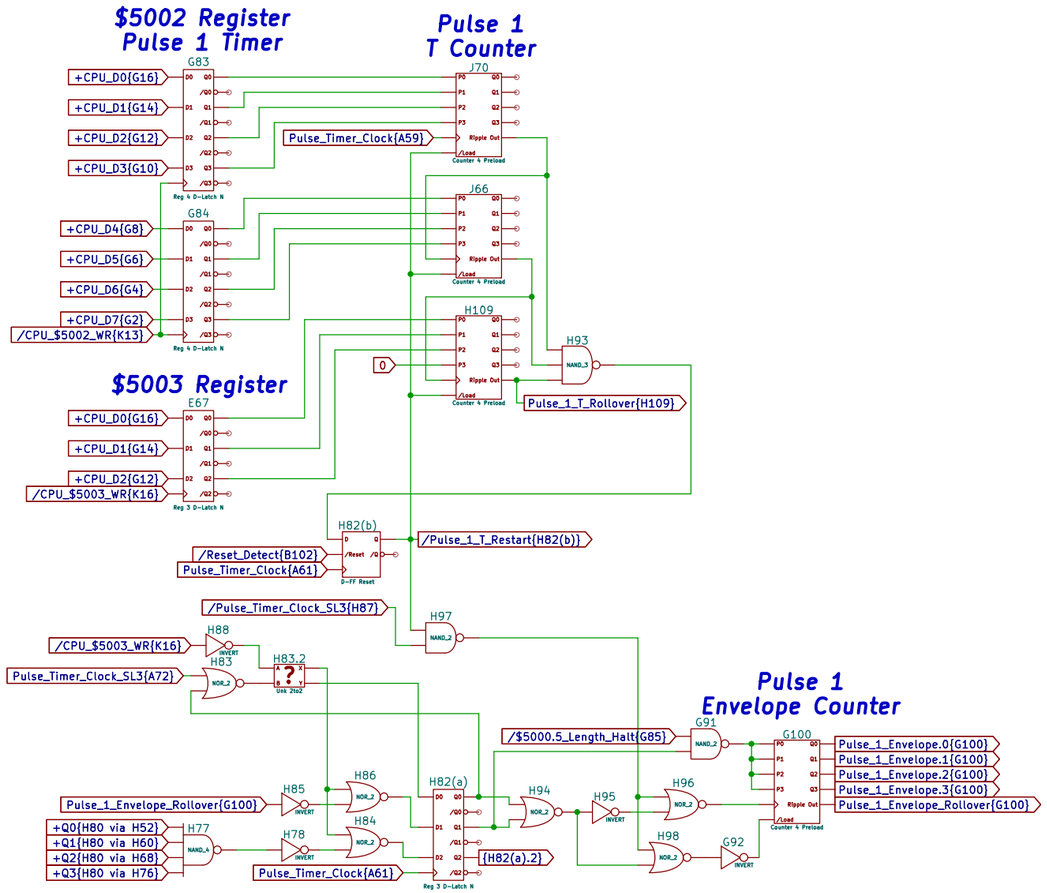


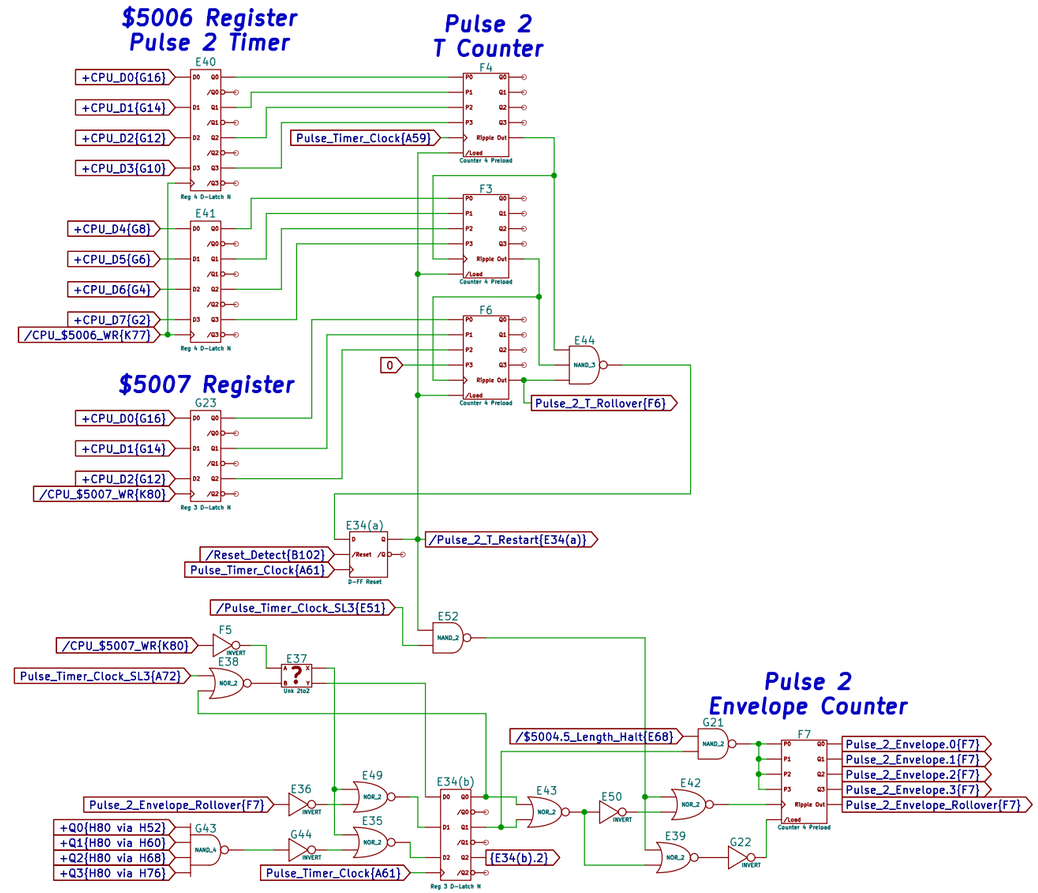
$5007 Register Pulse 2 Length Counter



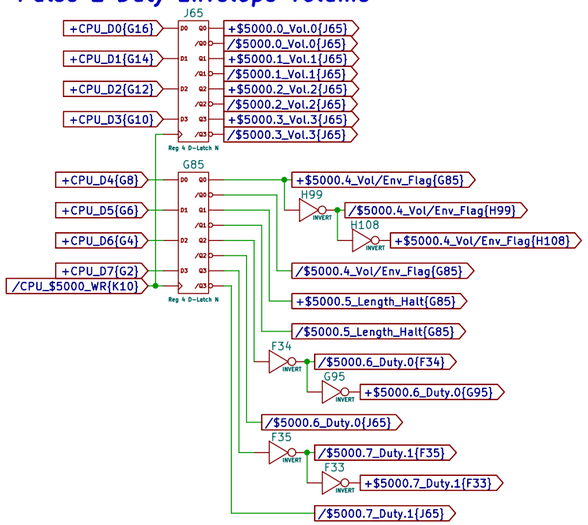
Pulse Channel Clock Source



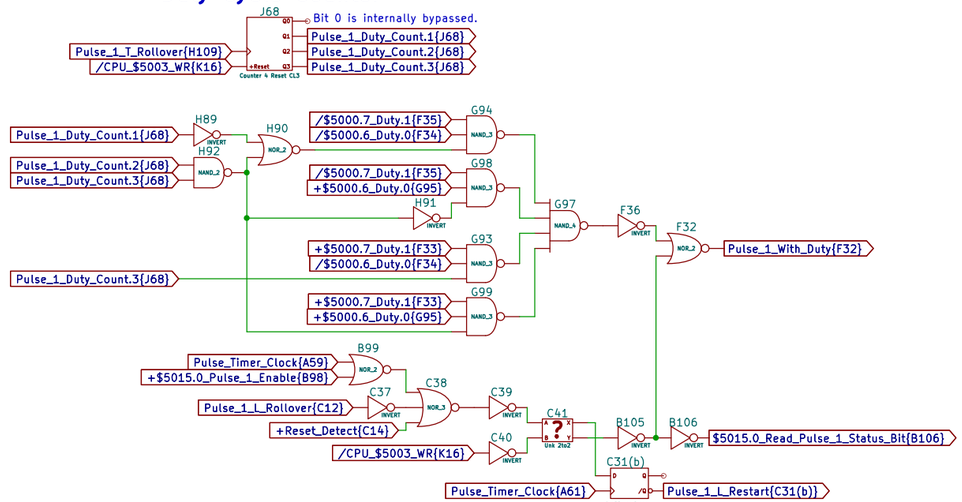

$5002 Register
Pulse 1 Timer
Pulse 1
T Counter
$5003 Register
Pulse 1
Envelope Counter
G83
G84
J70
J66
H109
H93
E67
H82(b)
H97
H88
H83
H83.2
G91
G100
H85
H86
H84
H82(a)
H94
H95
H96
H98
G92
H77
H78
+CPU_D0{G16}
+CPU_D1{G14}
+CPU_D2{G12}
+CPU_D3{G10}
+CPU_D4{G8}
+CPU_D5{G6}
+CPU_D6{G4}
+CPU_D7{G2}
/CPU_$5002_WR{K13}
/CPU_$5003_WR{K16}
Pulse_Timer_Clock{A59}
0
Pulse_1_T_Rollover{H109}
/Reset_Detect{B102}
Pulse_Timer_Clock{A61}
/Pulse_1_T_Restart{H82(b)}
/Pulse_Timer_Clock_SL3{H87}
/CPU_$5003_WR{K16}
Pulse_Timer_Clock_SL3{A72}
/$5000.5_Length_Halt{G85}
Pulse_1_Envelope.0{G100}
Pulse_1_Envelope.1{G100}
Pulse_1_Envelope.2{G100}
Pulse_1_Envelope.3{G100}
Pulse_1_Envelope_Rollover{G100}
Pulse_1_Envelope_Rollover{G100}
+Q0{H80 via H52}
+Q1{H80 via H60}
+Q2{H80 via H68}
+Q3{H80 via H76}
{H82(a).2}
Pulse_Timer_Clock{A61}


$5006 Register
Pulse 2 Timer
Pulse 2
T Counter
$5007 Register
Pulse 2
Envelope Counter
E40
E41
F4
F3
F6
E44
G23
E34(a)
E52
F5
E38
E37
G21
F7
E36
E49
E35
E34(b)
E43
E50
E42
E39
G22
G43
G44
+CPU_D0{G16}
+CPU_D1{G14}
+CPU_D2{G12}
+CPU_D3{G10}
+CPU_D4{G8}
+CPU_D5{G6}
+CPU_D6{G4}
+CPU_D7{G2}
/CPU_$5006_WR{K77}
/CPU_$5007_WR{K80}
Pulse_Timer_Clock{A59}
0
Pulse_2_T_Rollover{F6}
/Reset_Detect{B102}
Pulse_Timer_Clock{A61}
/Pulse_2_T_Restart{E34(a)}
/Pulse_Timer_Clock_SL3{E51}
/CPU_$5007_WR{K80}
Pulse_Timer_Clock_SL3{A72}
/$5004.5_Length_Halt{E68}
Pulse_2_Envelope.0{F7}
Pulse_2_Envelope.1{F7}
Pulse_2_Envelope.2{F7}
Pulse_2_Envelope.3{F7}
Pulse_2_Envelope_Rollover{F7}
Pulse_2_Envelope_Rollover{F7}
+Q0{H80 via H52}
+Q1{H80 via H60}
+Q2{H80 via H68}
+Q3{H80 via H76}
{E34(b).2}
Pulse_Timer_Clock{A61}

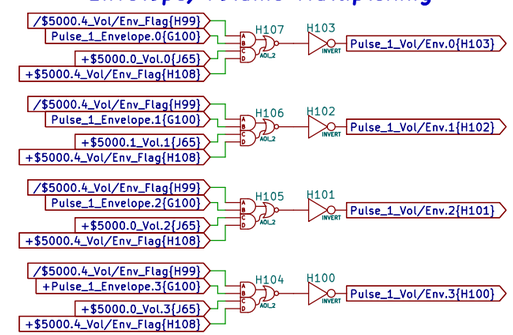
$5000 Register
Pulse 1 Duty Envelope Volume



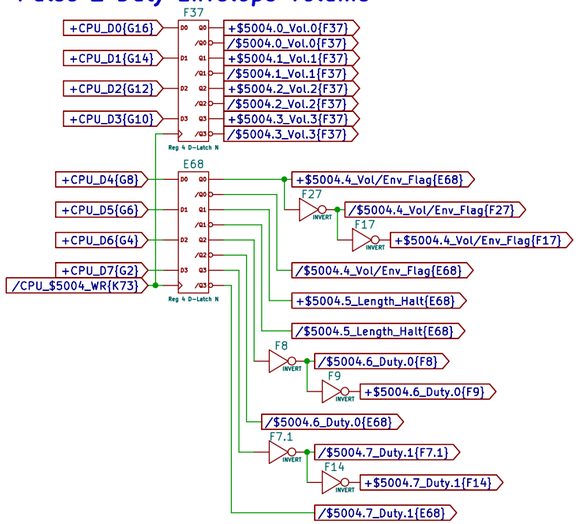
Pulse 1
Duty Cycle Counter



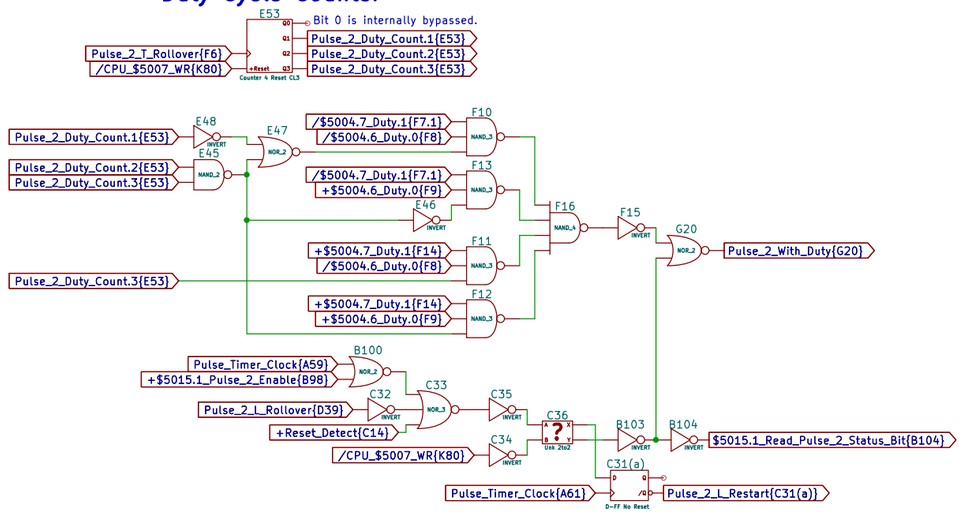
Pulse 1
Envelope/Volume Multiplexing



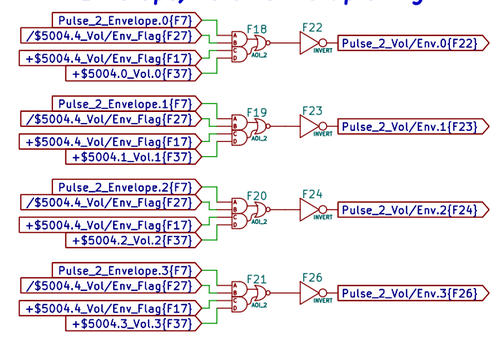
$5004 Register
Pulse 2 Duty Envelope Volume



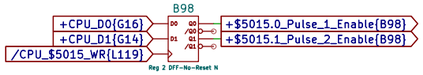
Pulse 2
Duty Cycle Counter



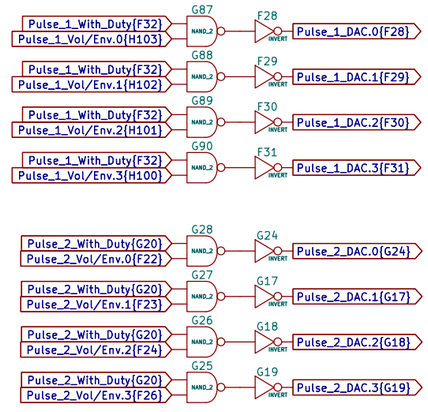
Pulse 2
Envelope/Volume Multiplexing



$5015 Write Register
Pulse 1&2 Status

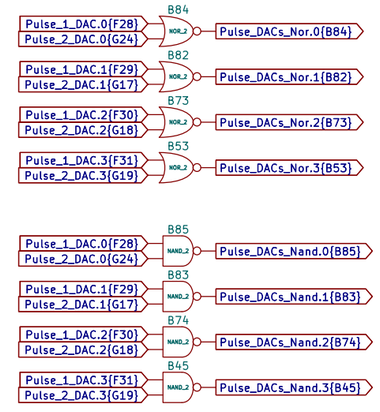


Pulse 1&2
Individual 4-bit DAC Outputs

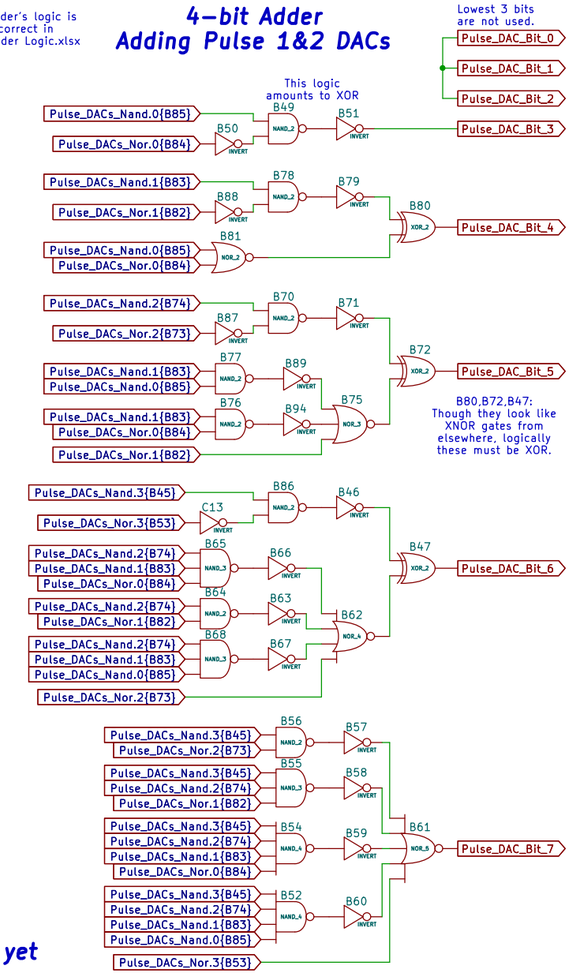


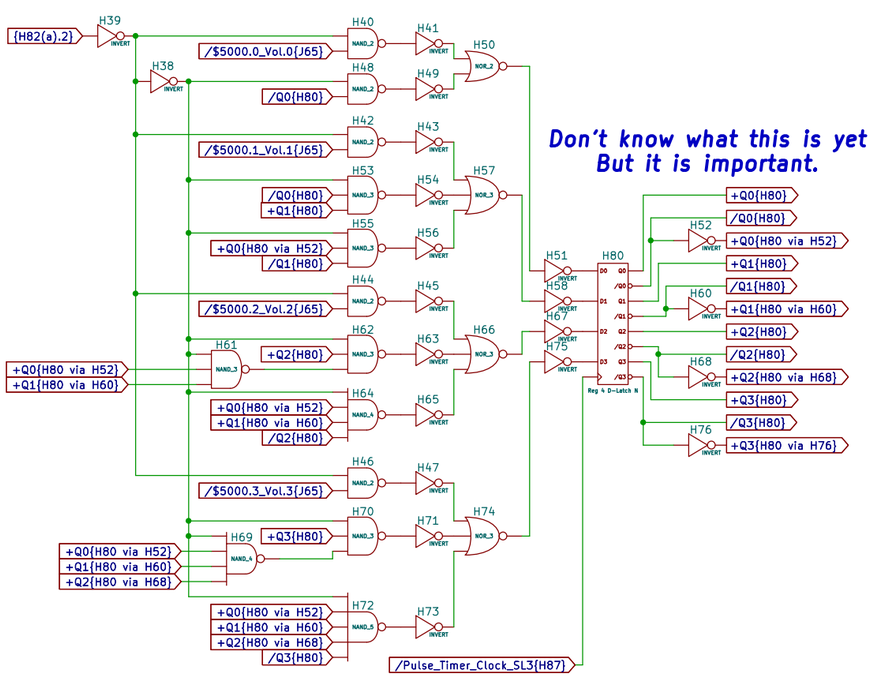
Components for
4-bit Adder

Note: The adder's logic is verified correct in Pulse DAC Adder Logic.xlsx

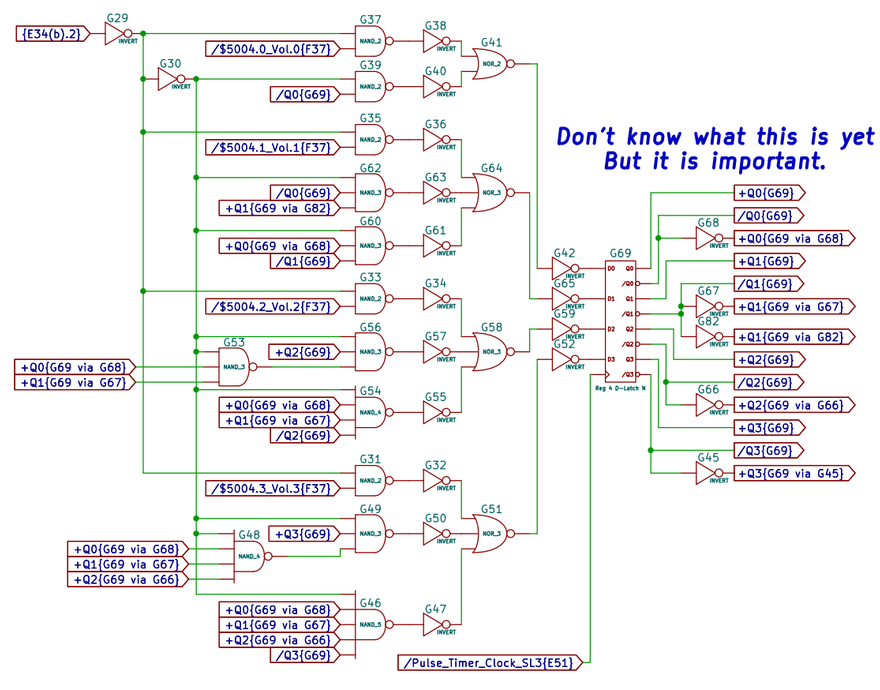


4-bit Adder
Adding Pulse 1&2 DACs

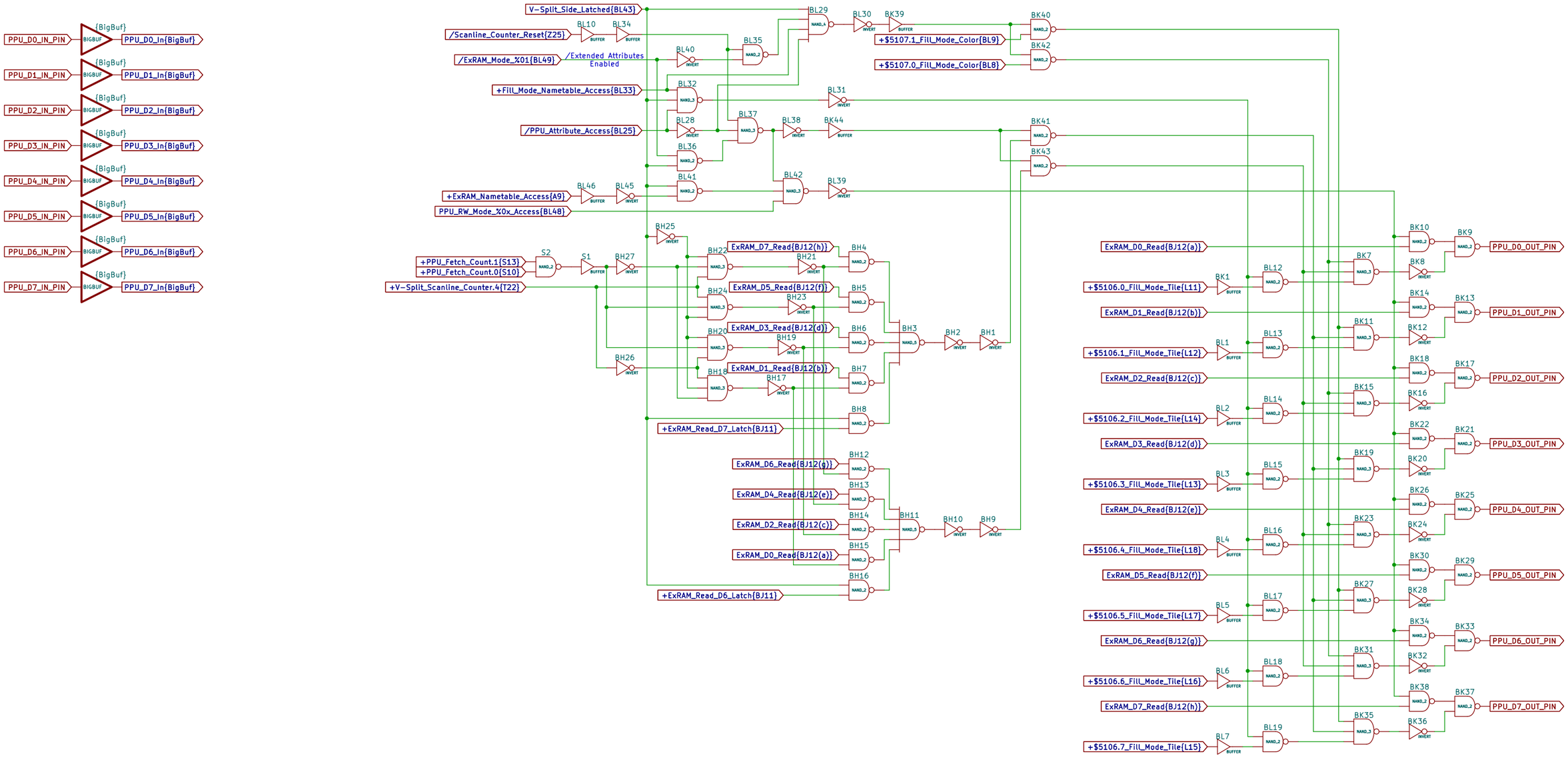


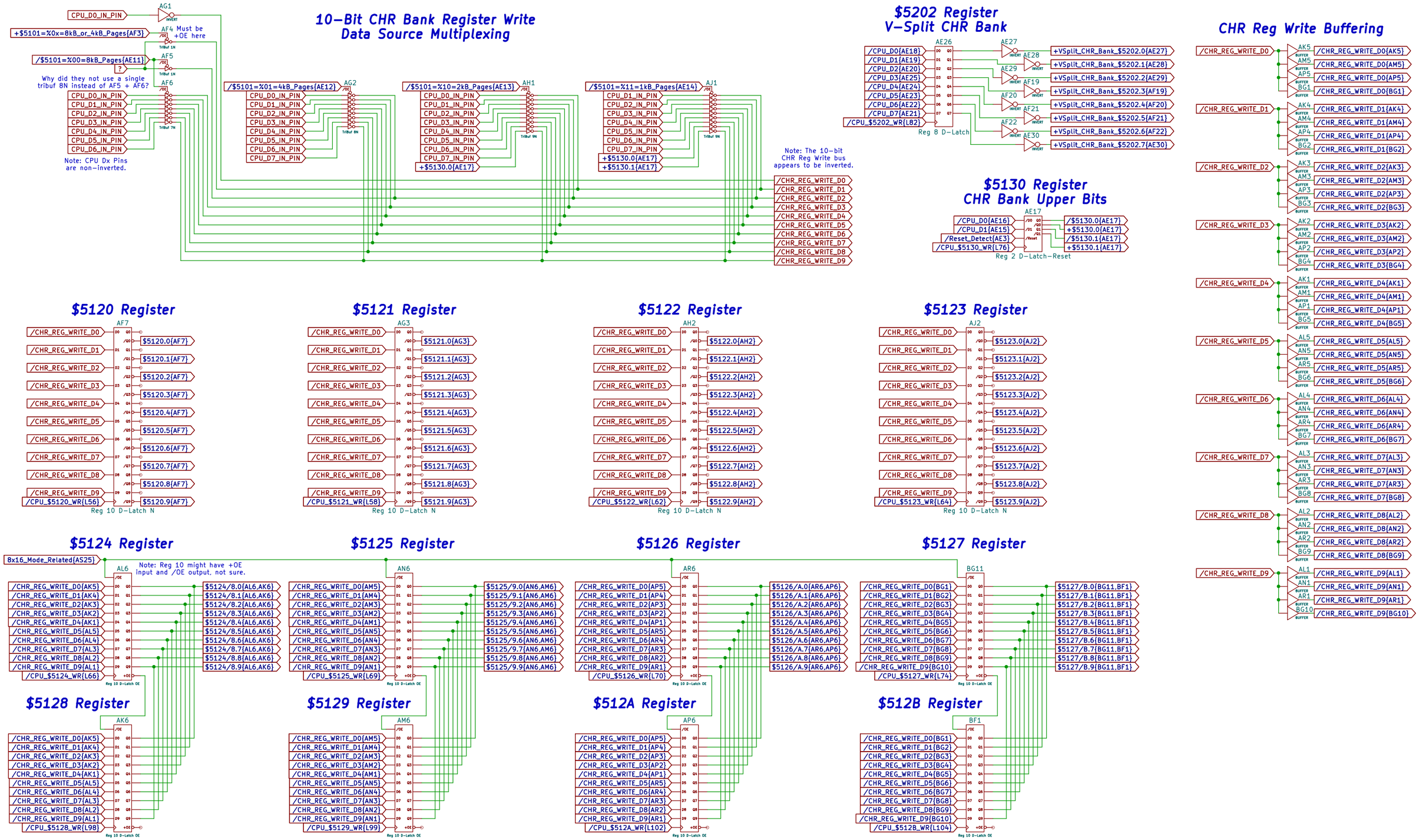


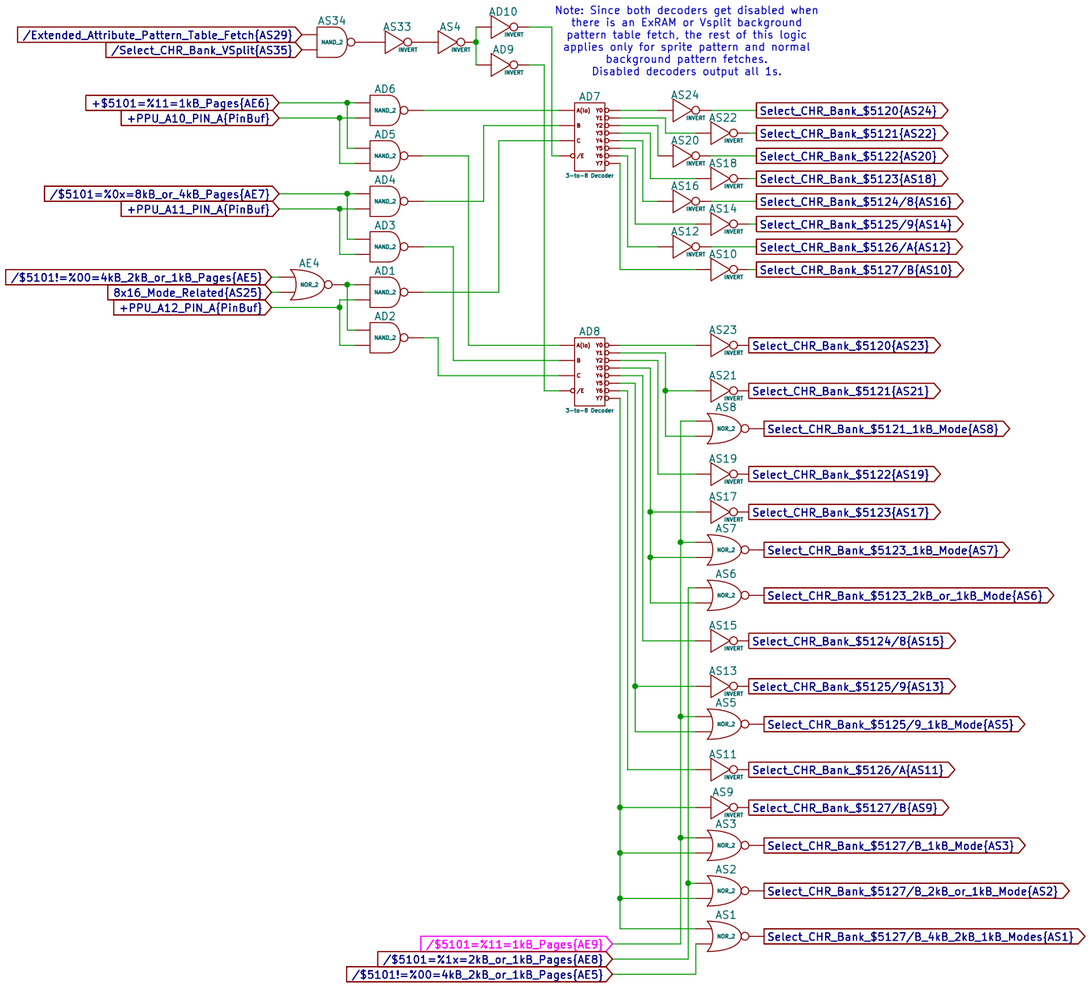
Don't know what this is yet
But it is important.



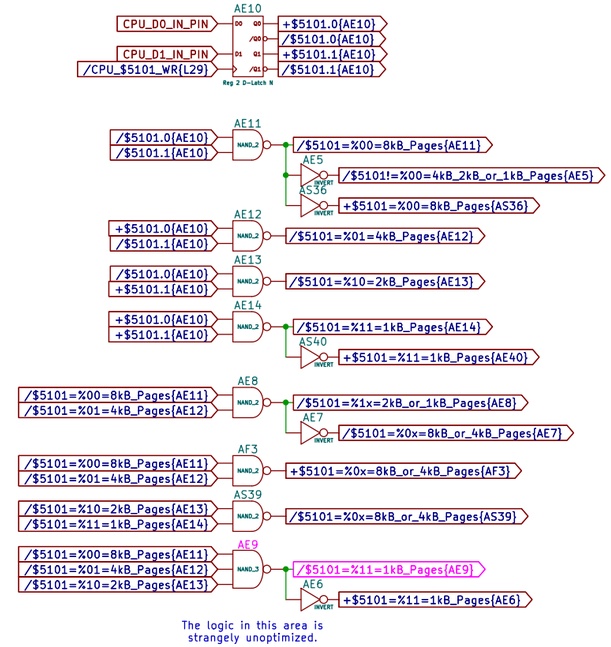
Don't know what this is yet
But it is important.

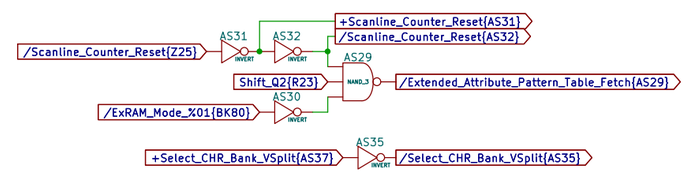

PPU_D0_IN_PIN
PPU_D1_IN_PIN
PPU_D2_IN_PIN
PPU_D3_IN_PIN
PPU_D4_IN_PIN
PPU_D5_IN_PIN
PPU_D6_IN_PIN
PPU_D7_IN_PIN
{BigBuf}
PPU_D0_In{BigBuf}
PPU_D1_In{BigBuf}
PPU_D2_In{BigBuf}
PPU_D3_In{BigBuf}
PPU_D4_In{BigBuf}
PPU_D5_In{BigBuf}
PPU_D6_In{BigBuf}
PPU_D7_In{BigBuf}
V-Split_Side_Latched{BL43}
/Scanline_Counter_Reset{Z25}
/ExRAM_Mode_%01{BL49}
/Extended Attributes Enabled
+Fill_Mode_Nametable_Access{BL33}
/PPU_Attribute_Access{BL25}
+ExRAM_Nametable_Access{A9}
PPU_RW_Mode_%0x_Access{BL48}
+$5107.1_Fill_Mode_Color{BL9}
+$5107.0_Fill_Mode_Color{BL8}
+PPU_Fetch_Count.1{S13}
+PPU_Fetch_Count.0{S10}
+V-Split_Scanline_Counter.4{T22}
+ExRAM_Read_D7_Latch{BJ11}
+ExRAM_Read_D6_Latch{BJ11}
ExRAM_D7_Read{BJ12(h)}
ExRAM_D5_Read{BJ12(f)}
ExRAM_D3_Read{BJ12(d)}
ExRAM_D1_Read{BJ12(b)}
ExRAM_D6_Read{BJ12(g)}
ExRAM_D4_Read{BJ12(e)}
ExRAM_D2_Read{BJ12(c)}
ExRAM_D0_Read{BJ12(a)}
+$5106.0_Fill_Mode_Tile{L11}
+$5106.1_Fill_Mode_Tile{L12}
+$5106.2_Fill_Mode_Tile{L14}
+$5106.3_Fill_Mode_Tile{L13}
+$5106.4_Fill_Mode_Tile{L18}
+$5106.5_Fill_Mode_Tile{L17}
+$5106.6_Fill_Mode_Tile{L16}
+$5106.7_Fill_Mode_Tile{L15}
PPU_D0_OUT_PIN
PPU_D1_OUT_PIN
PPU_D2_OUT_PIN
PPU_D3_OUT_PIN
PPU_D4_OUT_PIN
PPU_D5_OUT_PIN
PPU_D6_OUT_PIN
PPU_D7_OUT_PIN

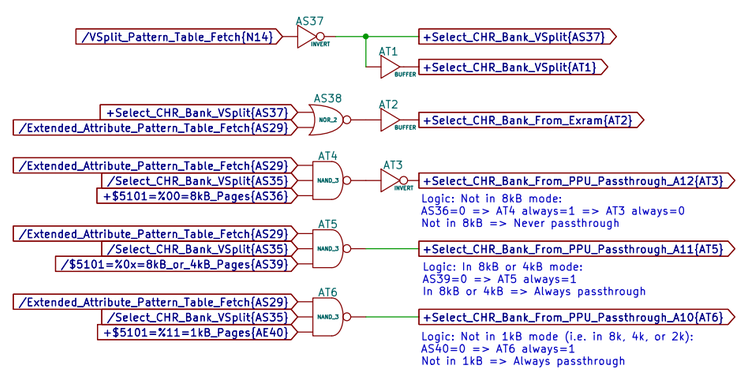

10-Bit CHR Bank Register Write Data Source Multiplexing
CPU_D0_IN_PIN
AG1
+$5101=%0x=8kB_or_4kB_Pages{AF3}
AF4 Must be +OE here
/$5101=%00=8kB_Pages{AE11}
AF5
?
Why did they not use a single tribuf 8N instead of AF5 + AF6?
AF6
CPU_D0_IN_PIN
CPU_D1_IN_PIN
CPU_D2_IN_PIN
CPU_D3_IN_PIN
CPU_D4_IN_PIN
CPU_D5_IN_PIN
CPU_D6_IN_PIN
Note: CPU Dx Pins are non-inverted.
/$5101=%01=4kB_Pages{AE12}
AG2
CPU_D7_IN_PIN
/$5101=%10=2kB_Pages{AE13}
AH1
+$5130.0{AE17}
/$5101=%11=1kB_Pages{AE14}
AJ1
+$5130.1{AE17}
Note: The 10-bit CHR Reg Write bus appears to be inverted.
/CHR_REG_WRITE_D0
/CHR_REG_WRITE_D1
/CHR_REG_WRITE_D2
/CHR_REG_WRITE_D3
/CHR_REG_WRITE_D4
/CHR_REG_WRITE_D5
/CHR_REG_WRITE_D6
/CHR_REG_WRITE_D7
/CHR_REG_WRITE_D8
/CHR_REG_WRITE_D9
$5202 Register
V-Split CHR Bank
AE26
/CPU_D0{AE18}
/CPU_D1{AE19}
/CPU_D2{AE20}
/CPU_D3{AE25}
/CPU_D4{AE24}
/CPU_D5{AE23}
/CPU_D6{AE22}
/CPU_D7{AE21}
/CPU_$5202_WR{LB2}
Reg 8 D-Latch
AE27
AE28
AE29
AF19
AF20
AF21
AF22
AE30
+VSplit_CHR_Bank_$5202.0{AE27}
+VSplit_CHR_Bank_$5202.1{AE28}
+VSplit_CHR_Bank_$5202.2{AE29}
+VSplit_CHR_Bank_$5202.3{AF19}
+VSplit_CHR_Bank_$5202.4{AF20}
+VSplit_CHR_Bank_$5202.5{AF21}
+VSplit_CHR_Bank_$5202.6{AF22}
+VSplit_CHR_Bank_$5202.7{AE30}
$5130 Register
CHR Bank Upper Bits
AE17
/CPU_D0{AE16}
/CPU_D1{AE15}
/Reset_Detect{AE3}
/CPU_$5130_WR{L76}
Reg 2 D-Latch-Reset
/$5130.0{AE17}
+$5130.0{AE17}
/$5130.1{AE17}
+$5130.1{AE17}
CHR Reg Write Buffering
/CHR_REG_WRITE_D0
AK5
AM5
AP5
BG1
/CHR_REG_WRITE_D1
AK4
AM4
AP4
BG2
/CHR_REG_WRITE_D2
AK3
AM3
AP3
BG3
/CHR_REG_WRITE_D3
AK2
AM2
AP2
BG4
/CHR_REG_WRITE_D4
AK1
AM1
AP1
BG5
/CHR_REG_WRITE_D5
AL5
AN5
AR5
BG6
/CHR_REG_WRITE_D6
AL4
AN4
AR4
BG7
/CHR_REG_WRITE_D7
AL3
AN3
AR3
BG8
/CHR_REG_WRITE_D8
AL2
AN2
AR2
BG9
/CHR_REG_WRITE_D9
AL1
AN1
AR1
BG10
BUFFER
$5120 Register
AF7
/CPU_$5120_WR{L56}
Reg 10 D-Latch N
$5121 Register
AG3
/CPU_$5121_WR{L58}
Reg 10 D-Latch N
$5122 Register
AH2
/CPU_$5122_WR{L62}
Reg 10 D-Latch N
$5123 Register
AJ2
/CPU_$5123_WR{L64}
Reg 10 D-Latch N
$5124 Register
8x16_Mode_Related{AS25}
AL6
Note: Reg 10 might have +OE input and /OE output, not sure.
/CPU_$5124_WR{L66}
Reg 10 D-Latch OE
$5125 Register
AN6
/CPU_$5125_WR{L69}
Reg 10 D-Latch OE
$5126 Register
AR6
/CPU_$5126_WR{L70}
Reg 10 D-Latch OE
$5127 Register
BG11
/CPU_$5127_WR{L74}
Reg 10 D-Latch OE
$5128 Register
AK6
/CPU_$5128_WR{L98}
Reg 10 D-Latch OE
$5129 Register
AM6
/CPU_$5129_WR{L99}
Reg 10 D-Latch OE
$512A Register
AP6
/CPU_$512A_WR{L102}
Reg 10 D-Latch OE
$512B Register
BF1
/CPU_$512B_WR{L104}
Reg 10 D-Latch OE



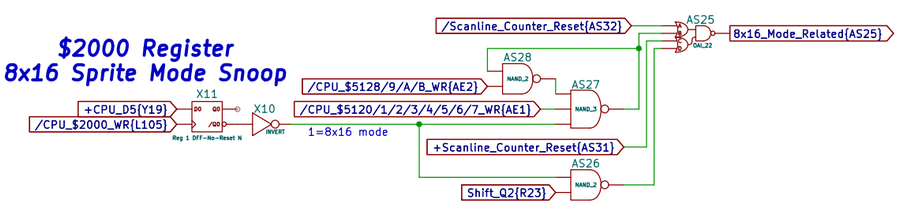
$5101 Register
CHR Mode



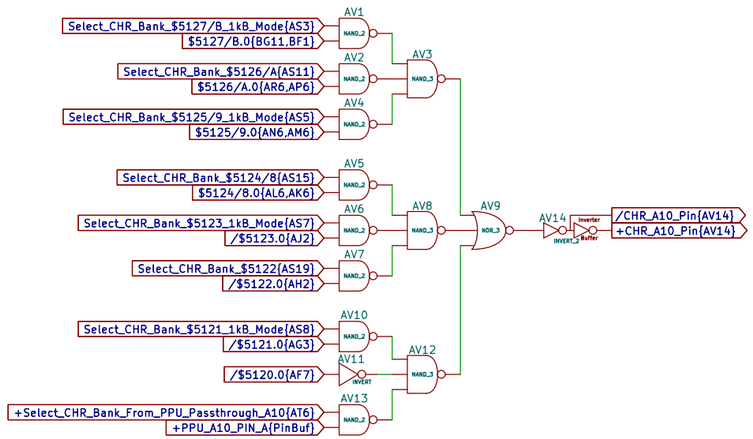




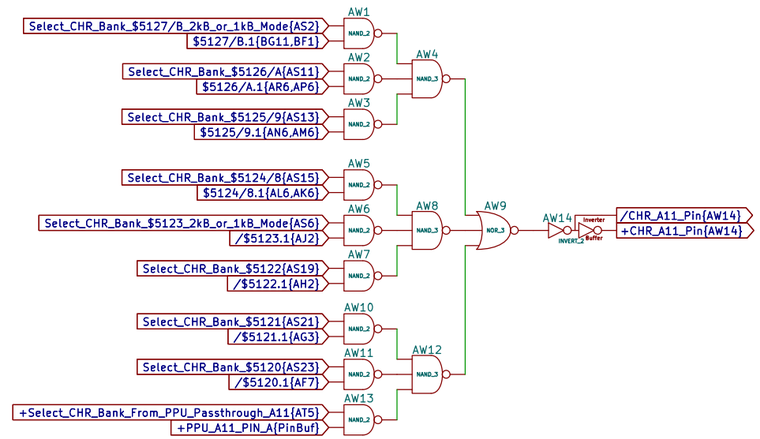
$2000 Register
8x16 Sprite Mode Snoop



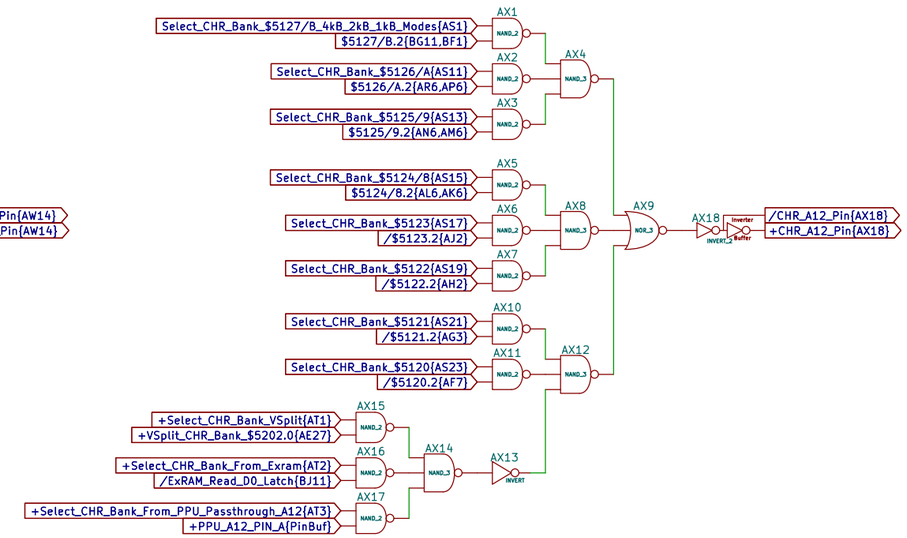
CHR A10



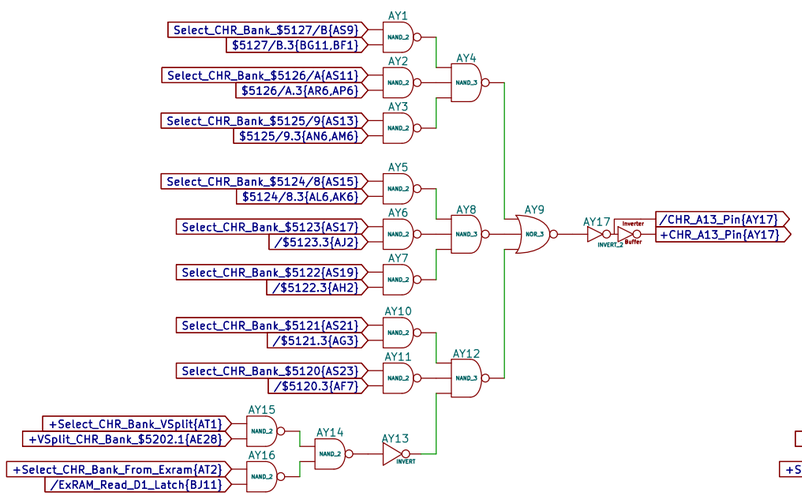
CHR A11



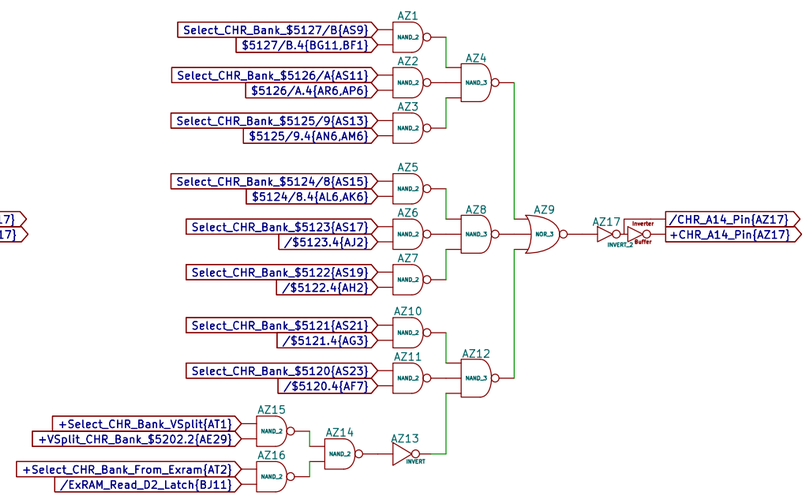
CHR A12



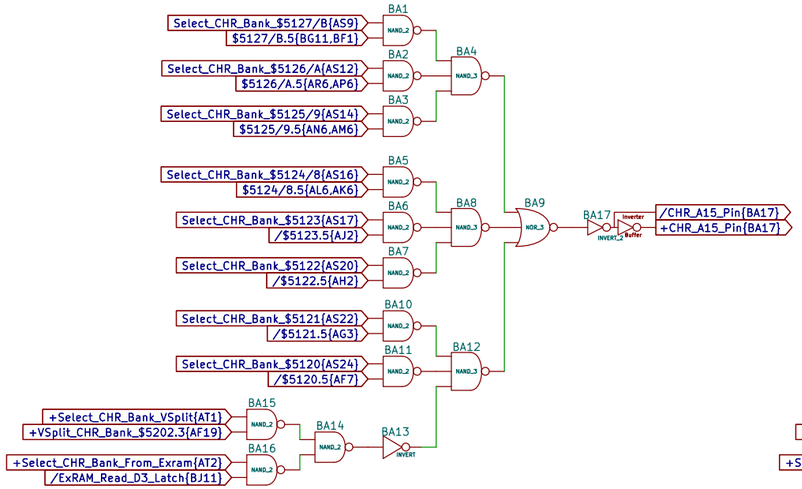
CHR A13



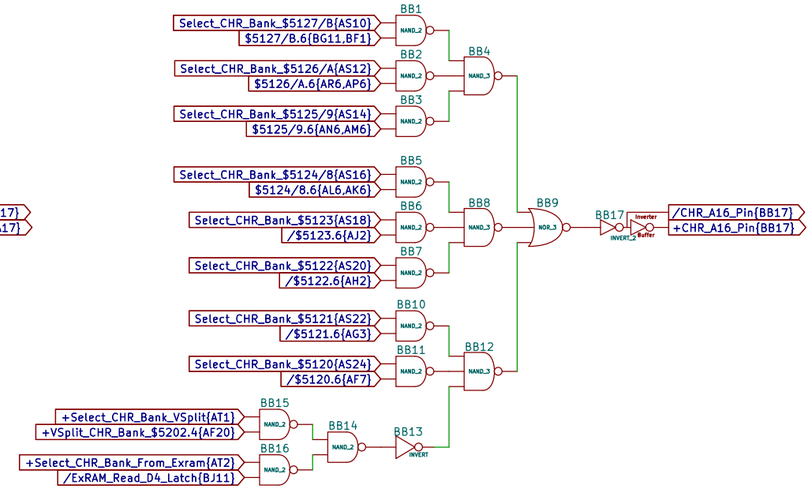
CHR A14



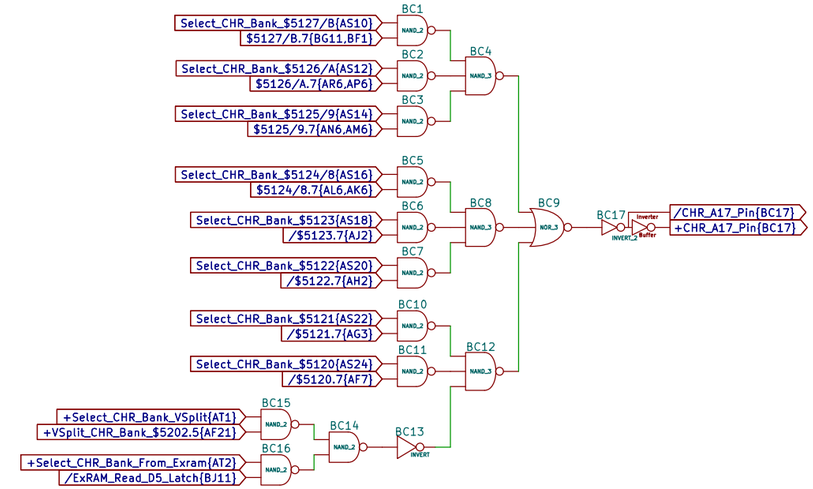
CHR A15



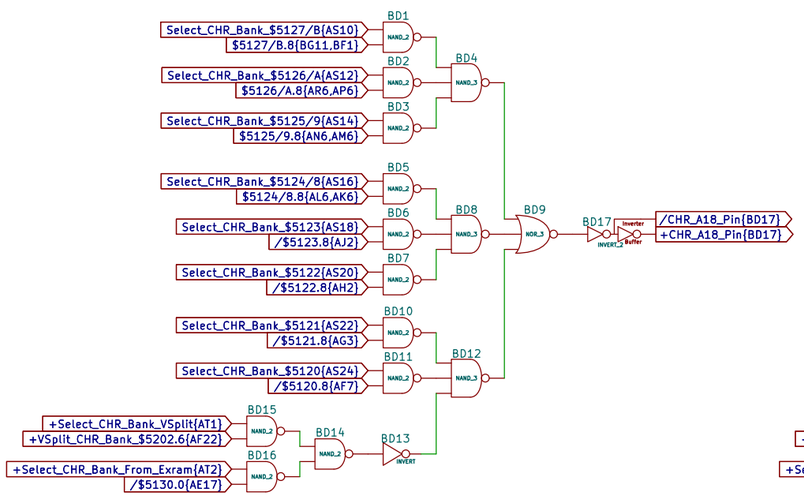
CHR A16



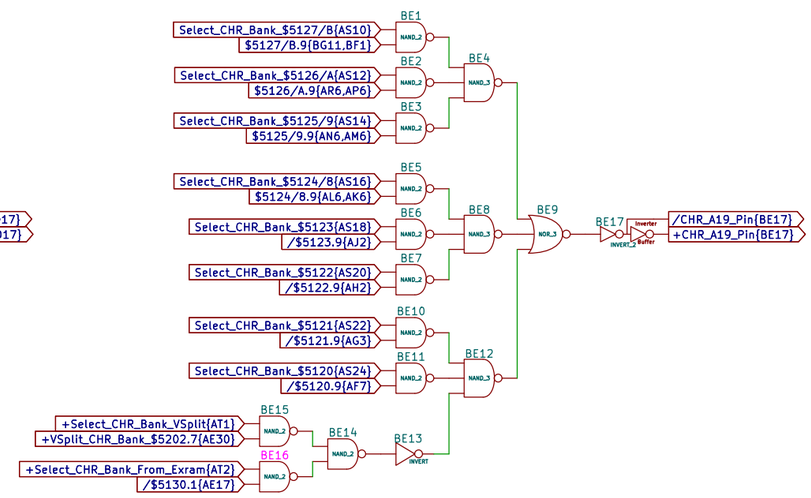
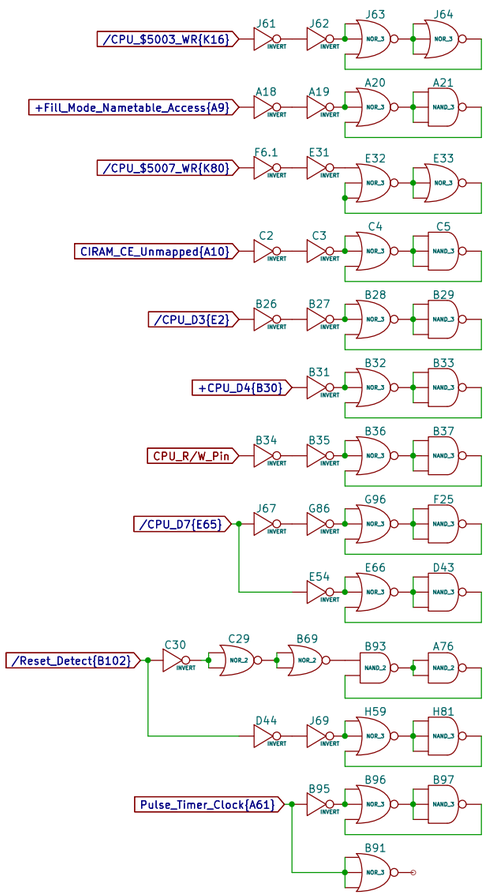
CHR A17



CHR A18



CHR A19



This sheet contains all residual items that don't serve any apparent purpose.