PS9 Final Micro Processor

Mingyuan Zheng(mz3143), Linxiao Wu(lw3227)

PLA

Truth table:

opcode	Assembly	subtract	mux_cntl	drv_enable	mem_write	mem_read	shift_bypass	load_bus	store_bus
000	NOP	0	001	0	0	0	1	0	0
001	LOAD	0	001	0	1	0	1	1	0
010	STORE	0	001	0	0	1	1	0	1
011	GET	0	100	0	0	1	1	0	0
100	PUT	0	001	1	1	0	1	0	0
101	ADD	0	010	0	0	1	\	0	0
110	SUB	1	010	0	0	1	\	0	0
111	SHIFT	0	001	0	0	0	0	0	0

Espresso script:

```
.i 3
.o 10
.ilb op2 op1 op0
.ob subtract mux2 mux1 mux0 drv_enable mem_write mem_read shift_bypass load_bus store_bus

# opcode 000 → NOP
0 0 0  0 0 0 1 0 0 0 1 0 0

# opcode 001 → LOAD
0 0 1  0 0 0 1 0 1 0 1 1 0

# opcode 010 → STORE
0 1 0  0 0 0 1 0 0 1 1 0 1

# opcode 011 → GET
0 1 1  0 1 0 0 0 0 1 1 0 0

# opcode 100 → PUT
1 0 0  0 0 0 1 1 1 0 1 0 0

# opcode 101 → ADD
1 0 1  0 0 1 0 0 0 1 1 0 0

# opcode 110 → SUB
1 1 0  1 0 1 0 0 0 1 1 0 0

# opcode 111 → SHIFT
1 1 1  0 0 0 1 0 0 0 0 0 0

.e
```

After running espresso ps9.pla, the output of the min truth table is shown as follows:

```
.i 3
.o 10
.ilb op2 op1 op0
.ob subtract mux2 mux1 mux0 drv_enable mem_write mem_read shift_bypass load_bus store_bus
.p 8
111 0001000000
010 0000001001
011 0100001100
101 0010001100
100 0001110100
001 0001010110
110 1010001100
0-0 0001000100
.e
```

SOP results:

```
# opcode 000 → NOP
# opcode 001 → LOAD
# opcode 010 → STORE
# opcode 011 → GET
# opcode 100 → PUT
# opcode 101 → ADD
# opcode 110 → SUB
# opcode 111 → SHIFT
subtract = (op2&op1&!op0);

mux2 = (!op2&op1&op0);

mux1 = (op2&!op1&op0) | (op2&op1&!op0);

mux0 = (op2&op1&op0) | (op2&!op1&!op0) | (!op2&!op1&op0) | (!op2&!op0);

drv_enable = (op2&!op1&!op0);

mem_write = (op2&!op1&!op0) | (!op2&!op1&op0);

mem_read = (!op2&op1&!op0) | (!op2&op1&op0) | (op2&!op1&op0) | (op2&op1
    &!op0);

shift_bypass = (!op2&op1&op0) | (op2&!op1&op0) | (op2&!op1&!op0) | (!op2
    &!op1&op0) | (op2&op1&!op0) | (!op2&!op0);

load_bus = (!op2&!op1&op0);

store_bus = (!op2&op1&!op0);
```

// 1. subtract (Active only for P_110)
subtract' = ((op2' + op1' + op0)')'
i
// 2. mux2 (Active only for P_011)
mux2' = ((op2 + op1' + op0')')'

// 3. mux1 (Active for P_101, P_110)
mux1' = ((op2' + op1+ op0')' + (op2' + op1' + op0)')'

// 4. mux0 (Active for P_111, P_001, P_x00, P_010)
mux0' = ((op2' + op1' + op0')' + (op2 + op1 + op0')' + (op2 + op0)' + (op2 + op1' + op0)')'

// 5. drv_enable (Active for P_100)

drv_enable' = ((op2' + op1 + op0)')'

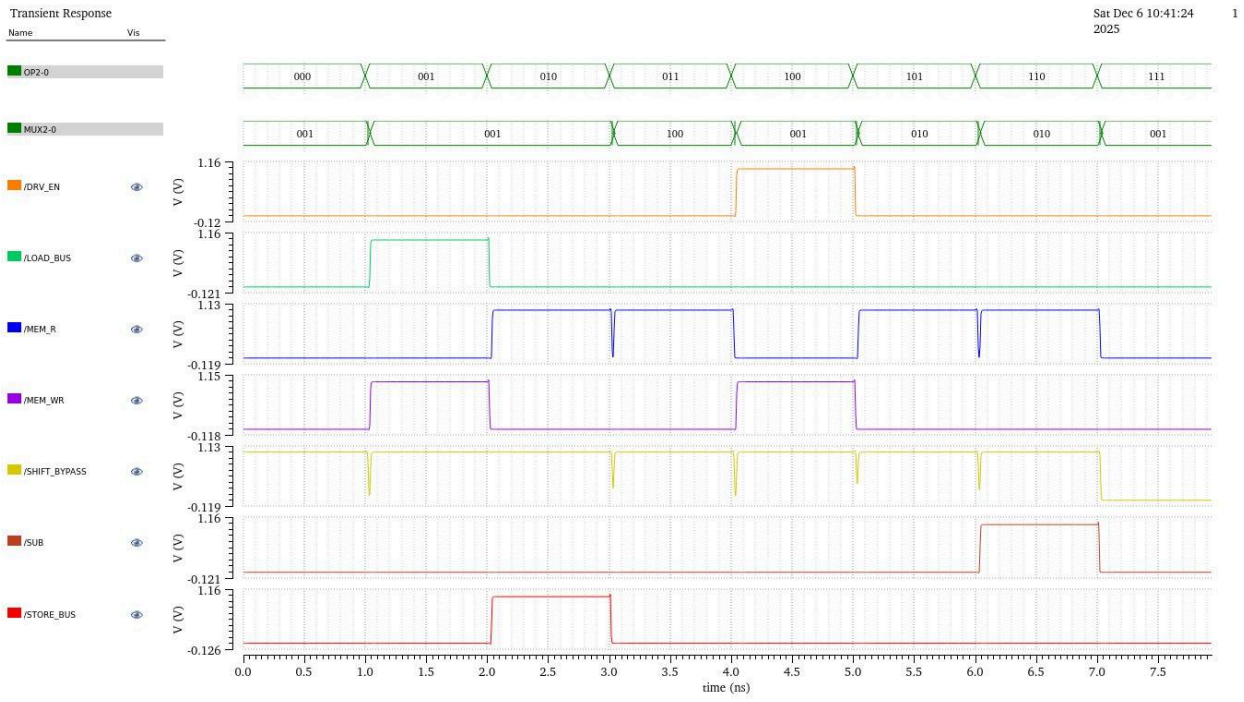
// 6. mem_write (Active for P_100, P_001)
mem_write' = ((op2' + op1 + op0)' + (op2 + op1 + op0')')'

// 7. mem_read (Active for P_011, P_101, P_110, P_010)
mem_read' = ((op2 + op1' + op0')' + (op2' + op1 + op0')' + (op2' + op1' + op0)' + (op2 + op1' + op0)')'

// 8. shift_bypass (Active for P_011, P_101, P_001, P_110, P_x00, P_010)
shift_bypass' = ((op2 + op1' + op0')' + (op2' + op1 + op0')' + (op2 + op1 + op0')' + (op2' + op1' + op0)' + (oP2 + op0)' + (op2 + op1' + op0)')'

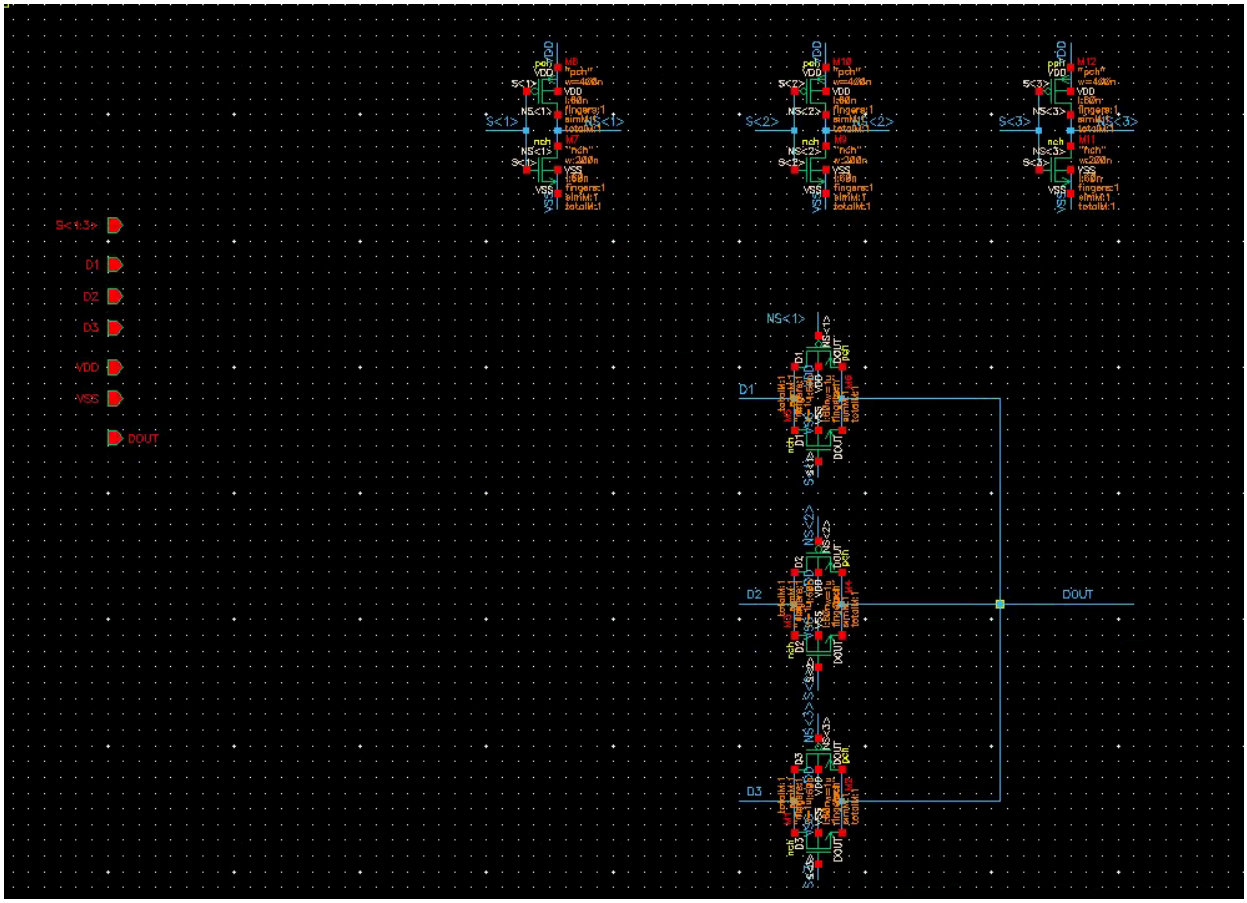
// 9. load_bus (Active for P_001)
load_bus' = ((op2 + op1 + op0')')'

// 10. store_bus (Active for P_010)
store_bus' = ((op2 + op1' + op0)')'

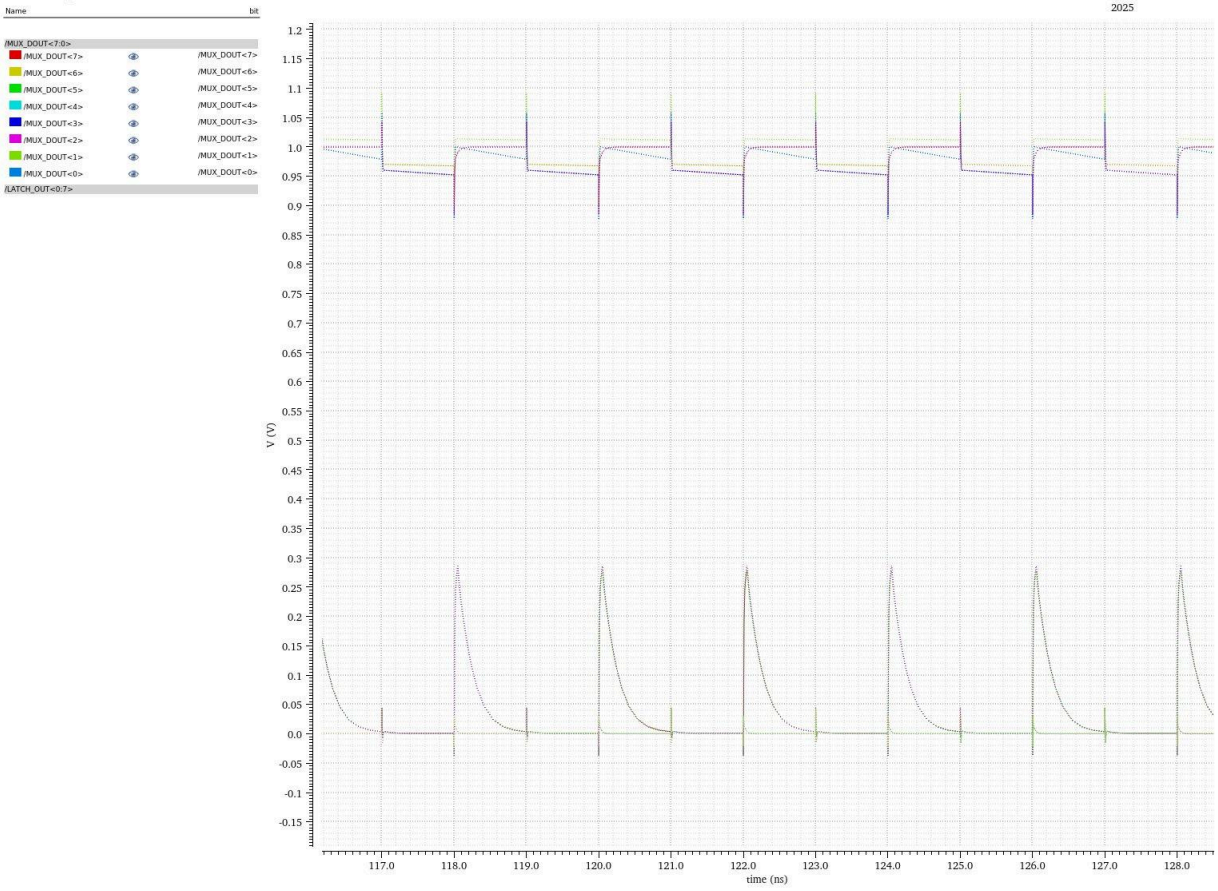


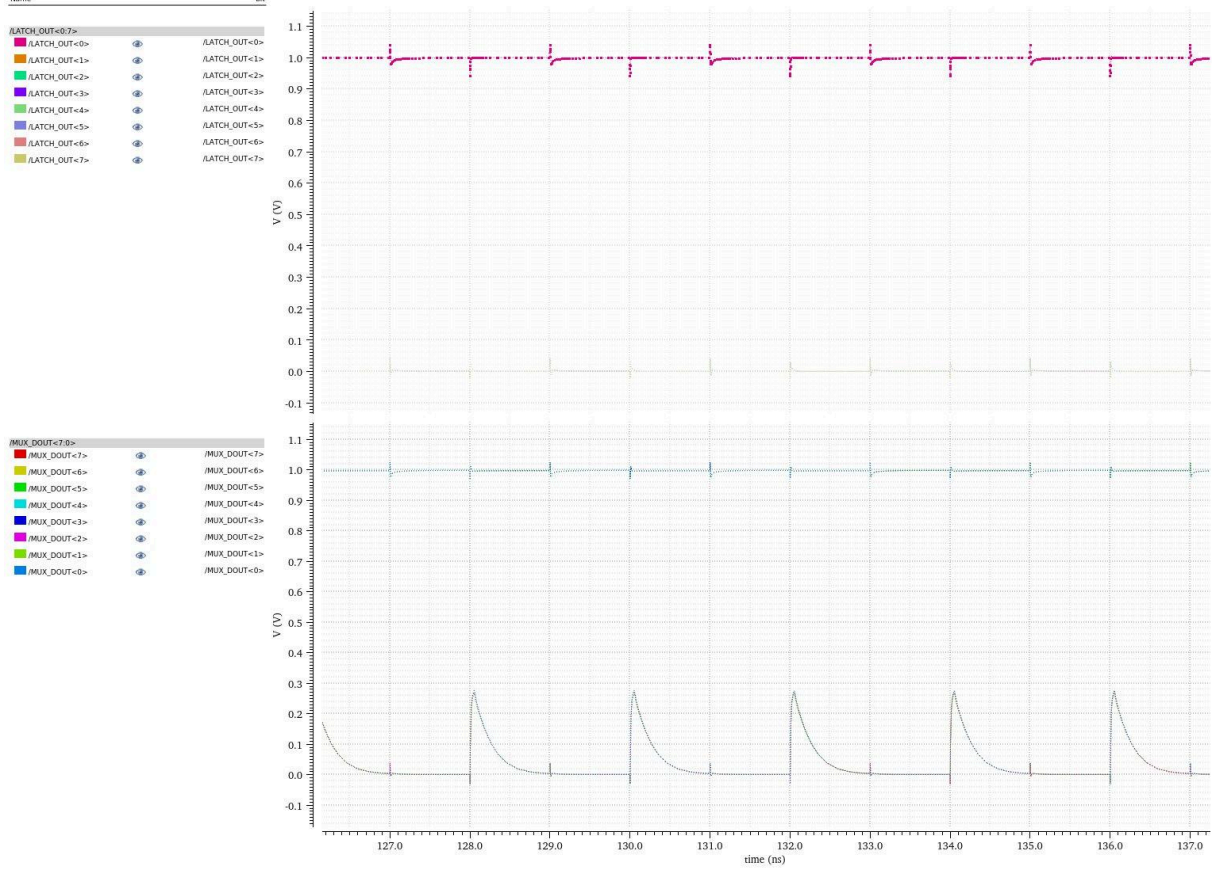
MUX

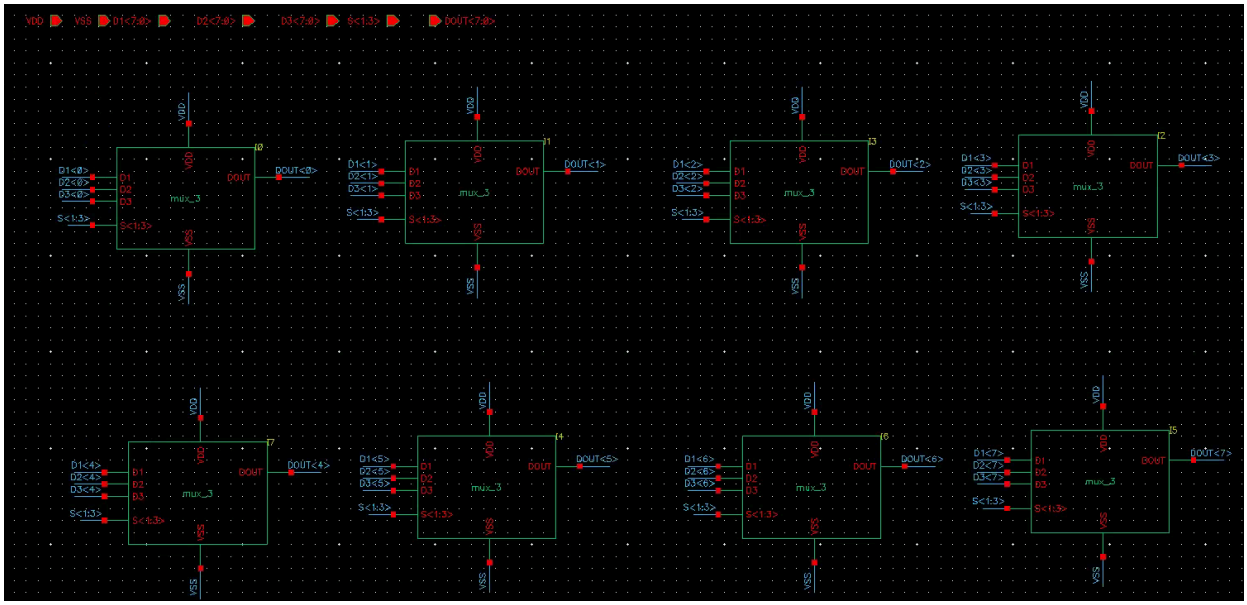
This is an 8-bit with 3-input MUX, which chooses one out of three signals, from SRAM, Adder or Shifter.



For the sizing, as the following figures showed, we swept through 200nm to 2um and found out that sizing the pass transistor to 1um and the inverter to 400nm/20nm has the best minimal rise/fall time.

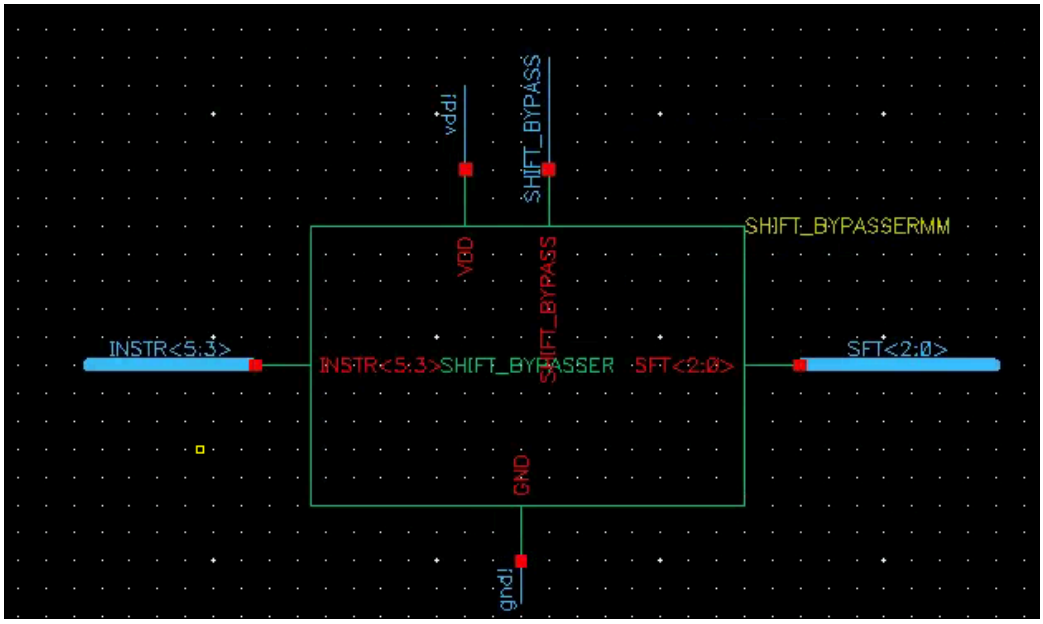

Name
bit
/MUX_DOUT<7:0>
/MUX_DOUT<7>
/MUX_DOUT<6>
/MUX_DOUT<5>
/MUX_DOUT<4>
/MUX_DOUT<3>
/MUX_DOUT<2>
/MUX_DOUT<1>
/MUX_DOUT<0>
/LATCH_OUT<0:7>
V (V)
time (ns)

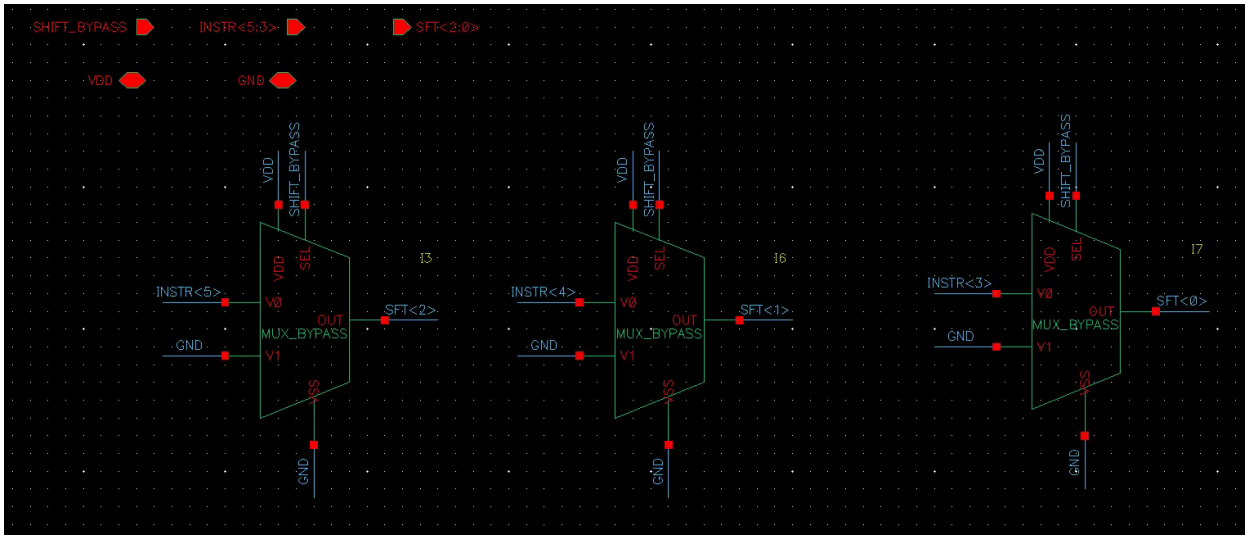

Name
bit
/LATCH_OUT<0:7>
/LATCH_OUT<0>
/LATCH_OUT<1>
/LATCH_OUT<2>
/LATCH_OUT<3>
/LATCH_OUT<4>
/LATCH_OUT<5>
/LATCH_OUT<6>
/LATCH_OUT<7>
/MUX_DOUT<7:0>
/MUX_DOUT<7>
/MUX_DOUT<6>
/MUX_DOUT<5>
/MUX_DOUT<4>
/MUX_DOUT<3>
/MUX_DOUT<2>
/MUX_DOUT<1>
/MUX_DOUT<0>
V (V)
time (ns)


VDD
VSS
D1<7:0>
D2<7:0>
D3<7:0>
S<1:3>
DOUT<7:0>
mux_3

Shift Bypasser

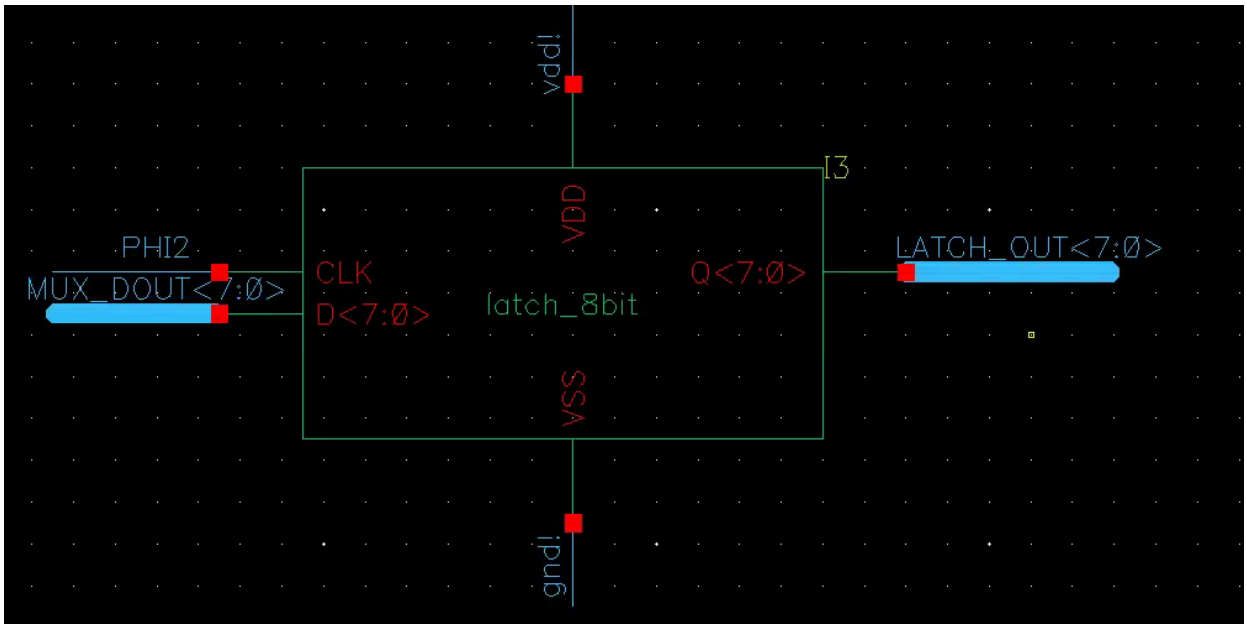
We designed a new component called ‘Shift Bypasser’ for this micro processor, which plays a role like a switch . When the signal SHIFT_BYPASS is 0, the SFT<2:0> equals to INSTR<5:3> and passes to the shifter. When the signal SHIFT_BYPASS is 1, the mux inside this component chooses GND and it shows to the shifter that it doesn’t need to shift. The INSTR<5:3> then act as a pure MEM Address.

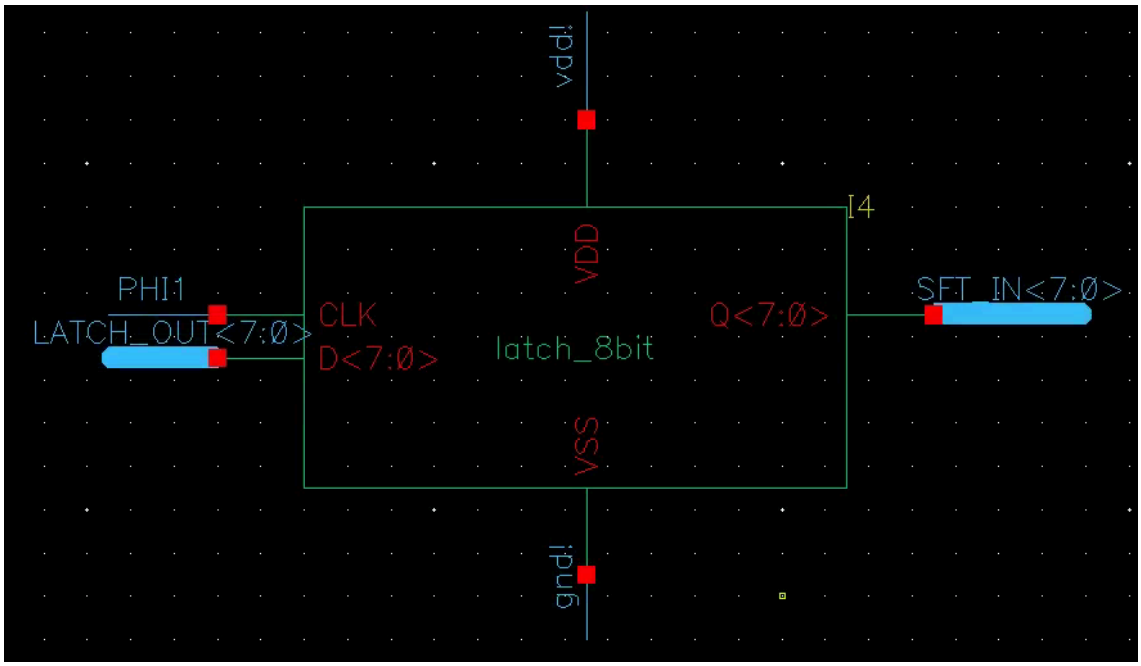




Accumulator

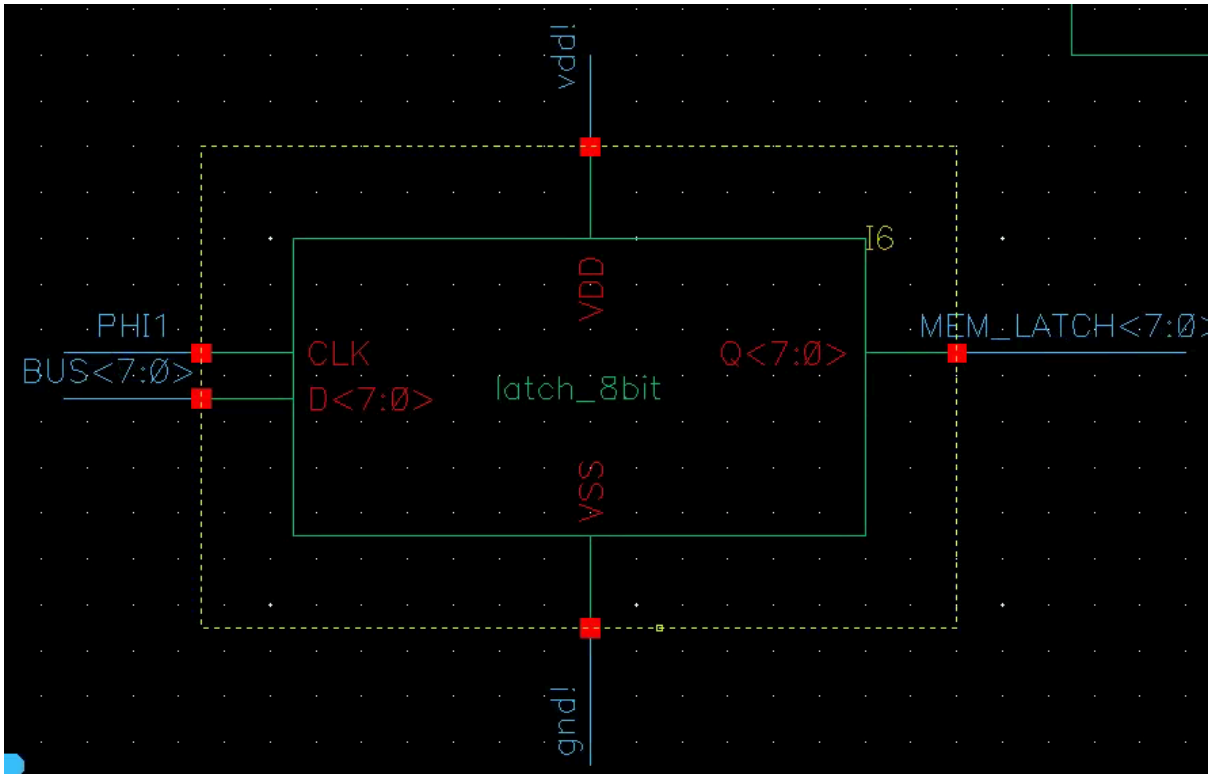
Two latches act as a Flip-Flop and it’s the accumulator in the micro processor.





Mem Latch

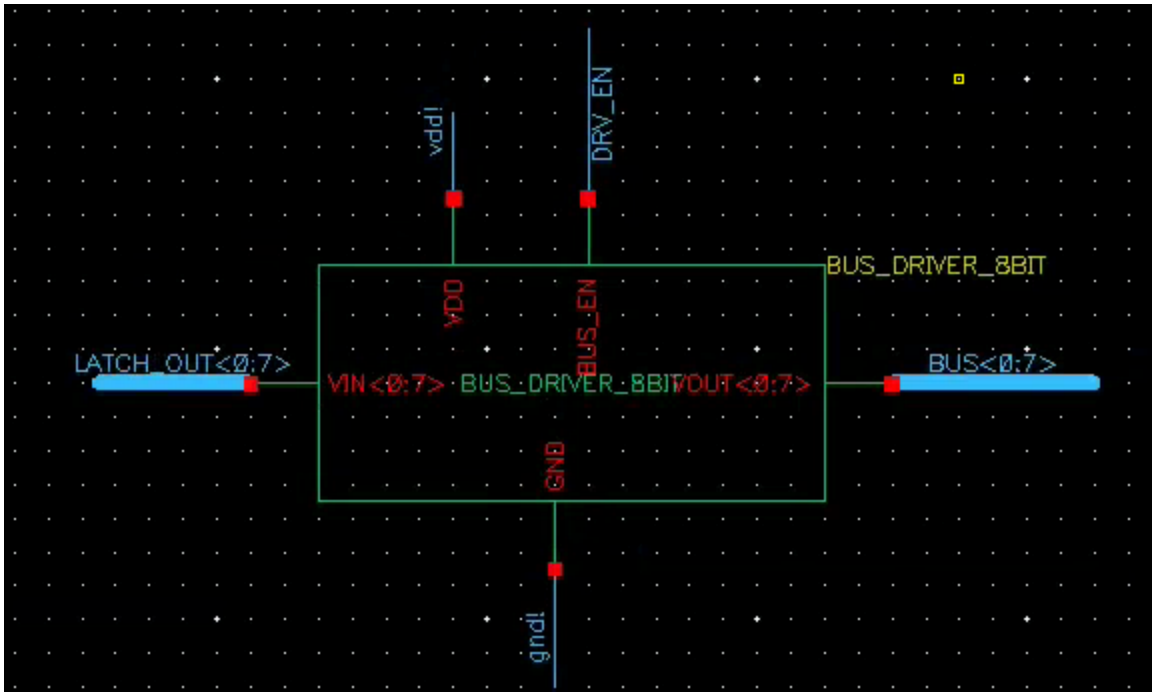
This latch is to make the BUS<7:0> stable and comes out as MEM_LATCH<7:0>.



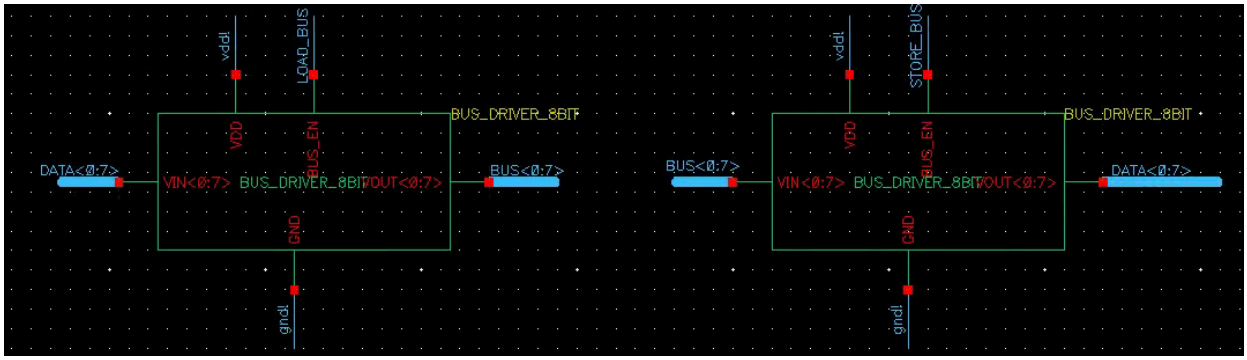
Bus Driver

Internal Bus Driver

This bus driver is the internal driver that is controlled by DRV_EN if it can drive the inside data to outside BUS<0:7>.



External Bus Driver

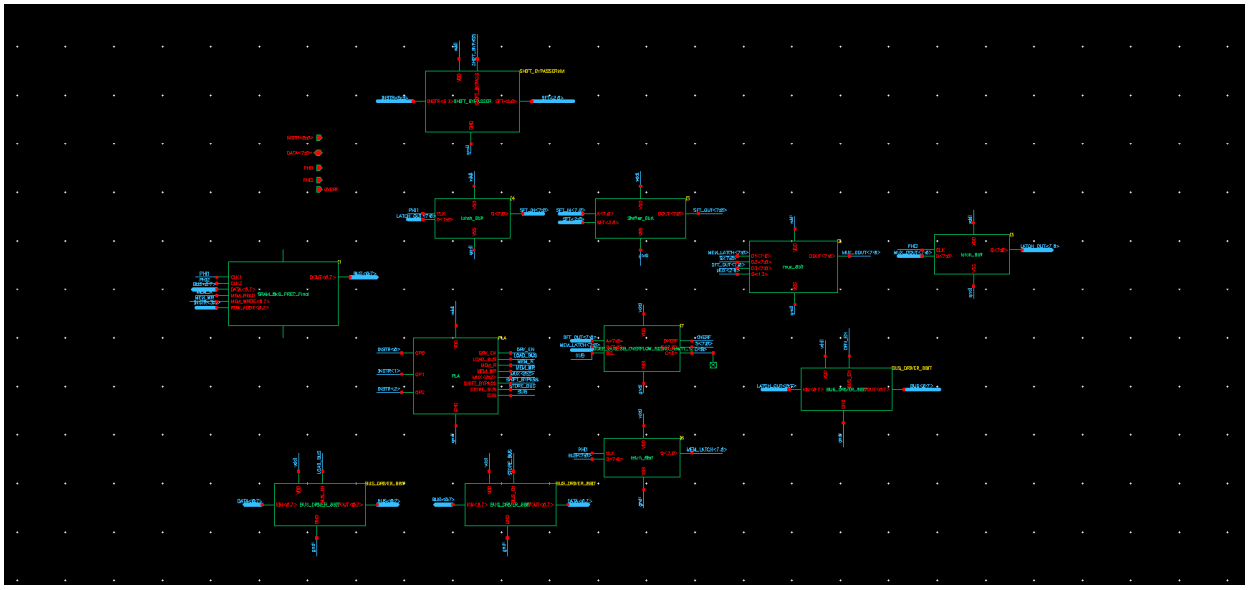


Other Components

Shifter, Adder,MEM are drawn and specified in former reports.

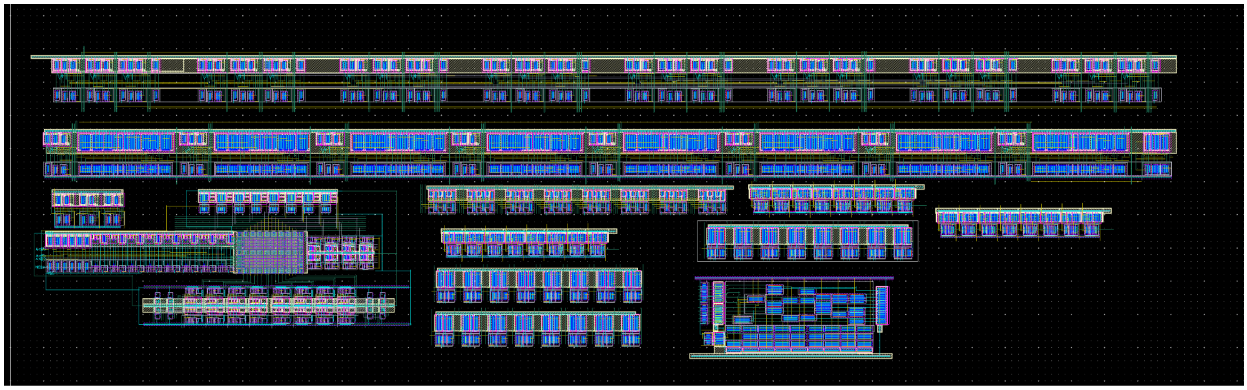
Micro Processor

Schematic

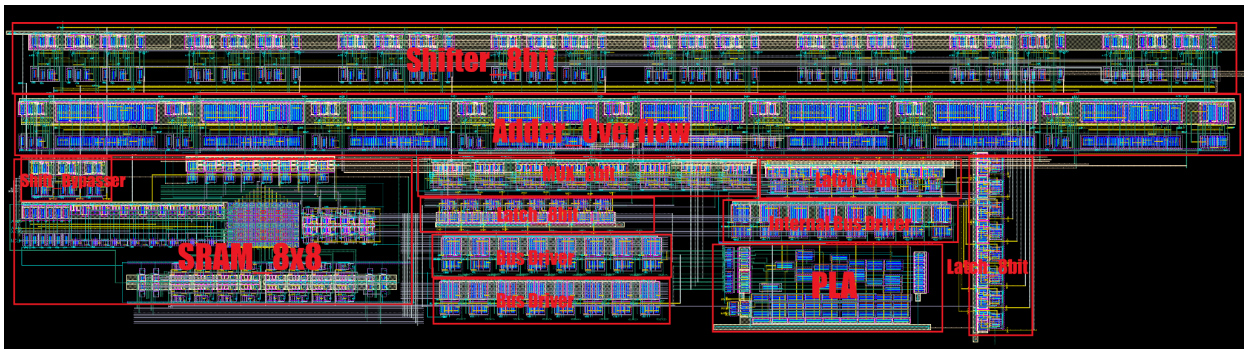


LAYOUT

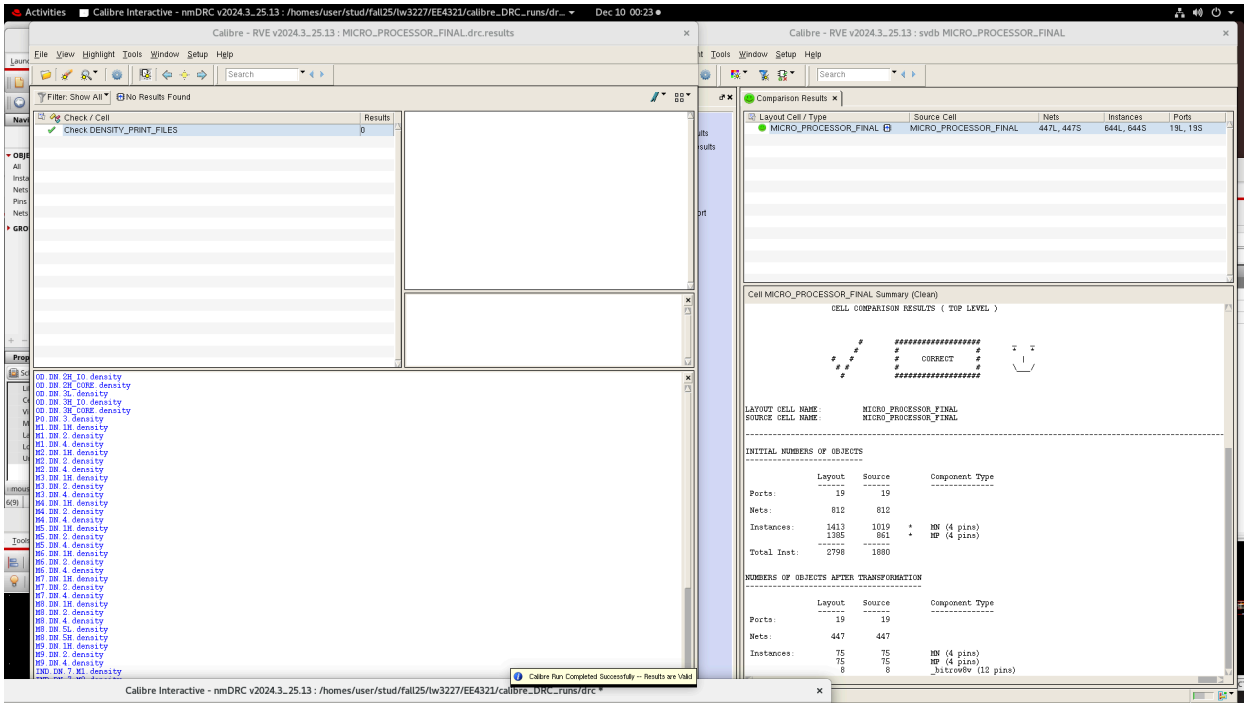
Before place and route



After Place and Route:

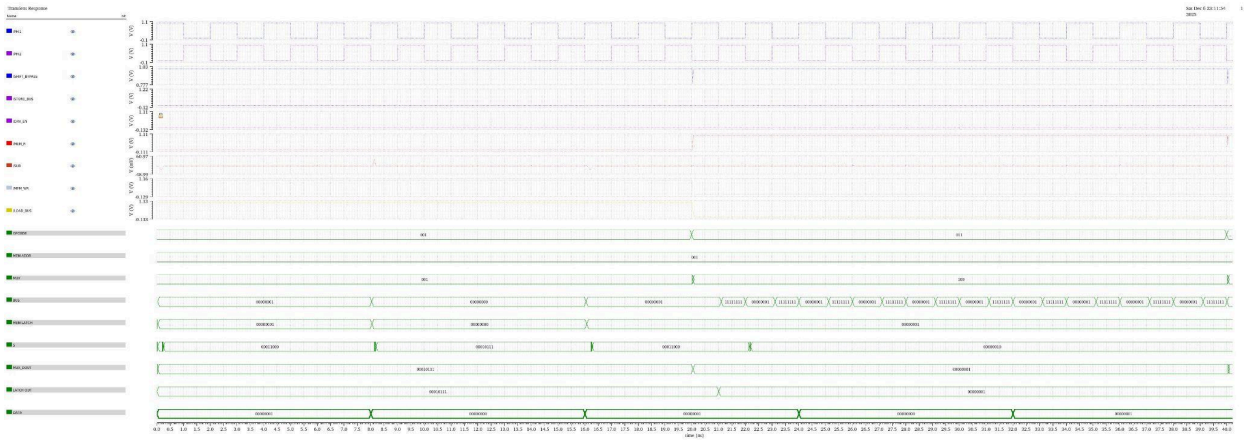


DRC and LVS clean:



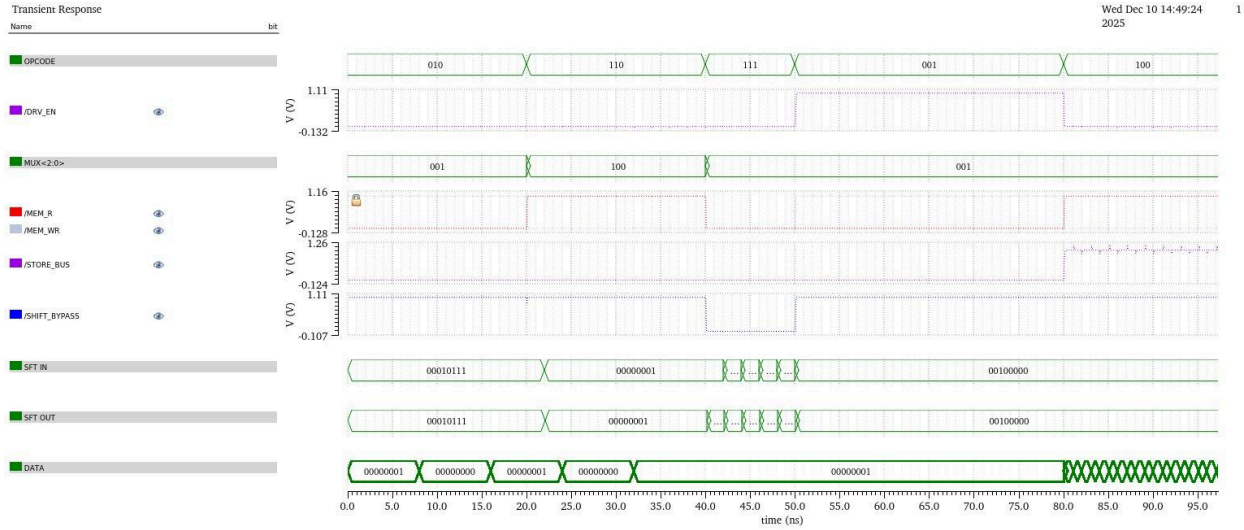
Pre-layout Simulation

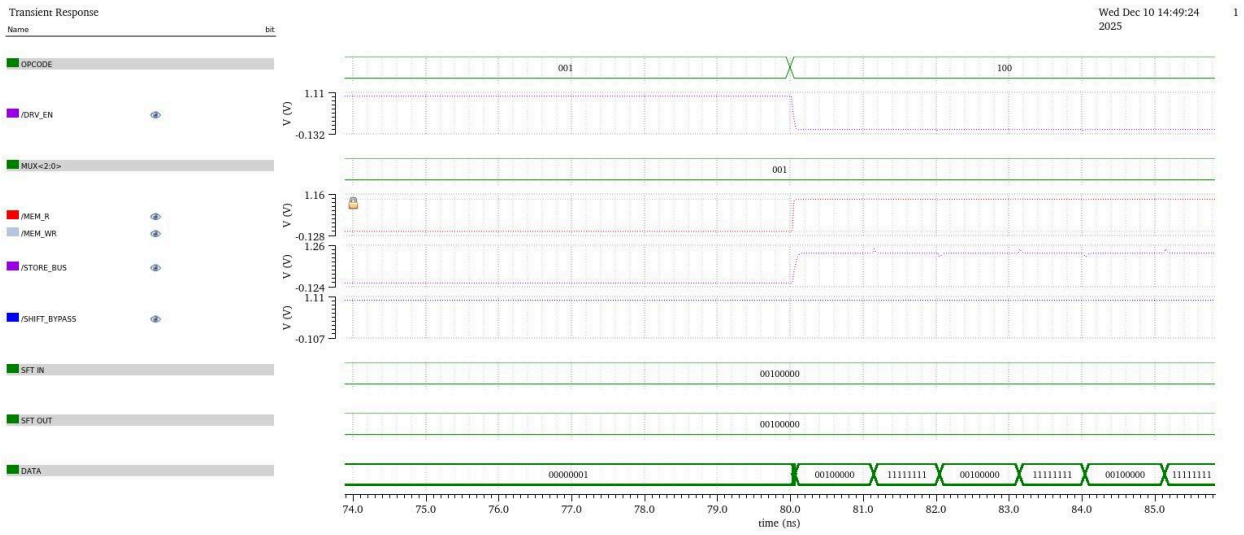
1. ADD calculation



2. SHIFT

When setting the opcode to 010>110->111>001>100, it conducts shifting after loading from MEM[0], then stores the result back to MEM[1], and at last reads it out from the external BUS. As it's shown in the figure, after shifting, SFT_OUT becomes 00100000, and finally DATA reads out correctly as 00100000 in the PUT stage.

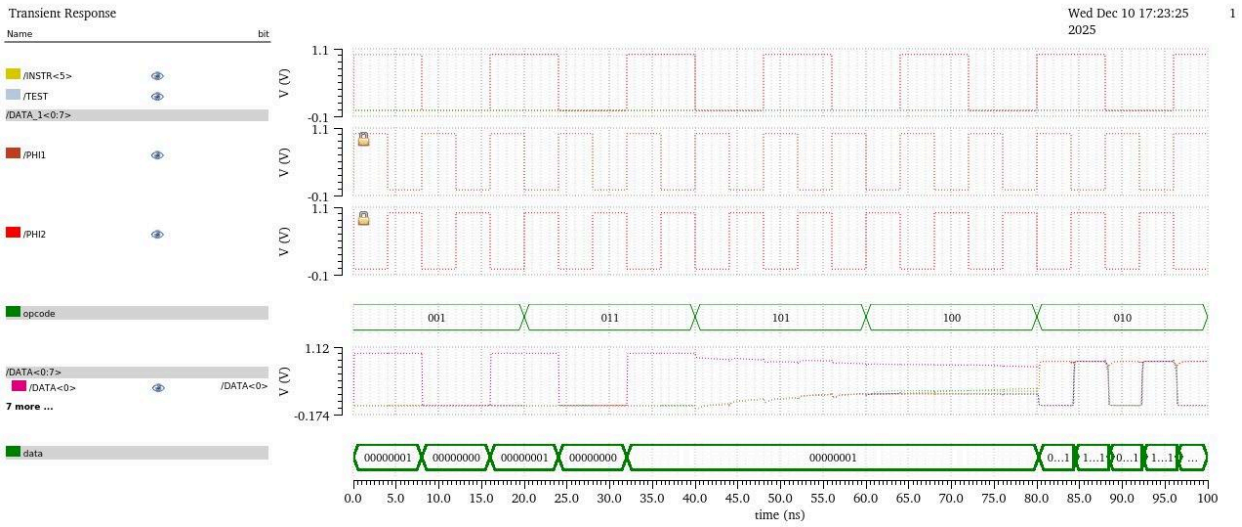


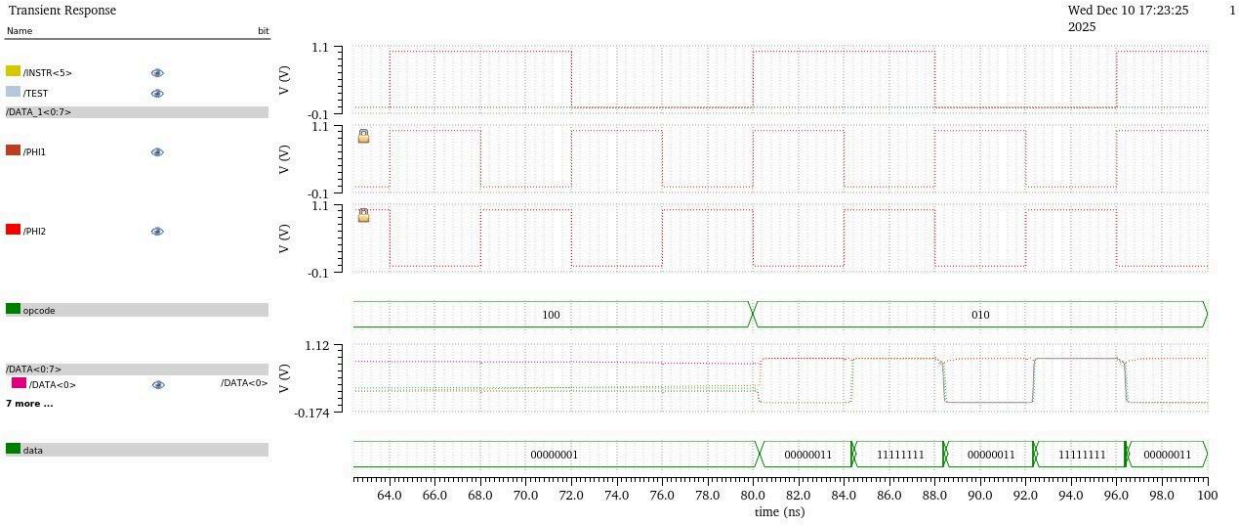


Post-layout Simulation

ADD:

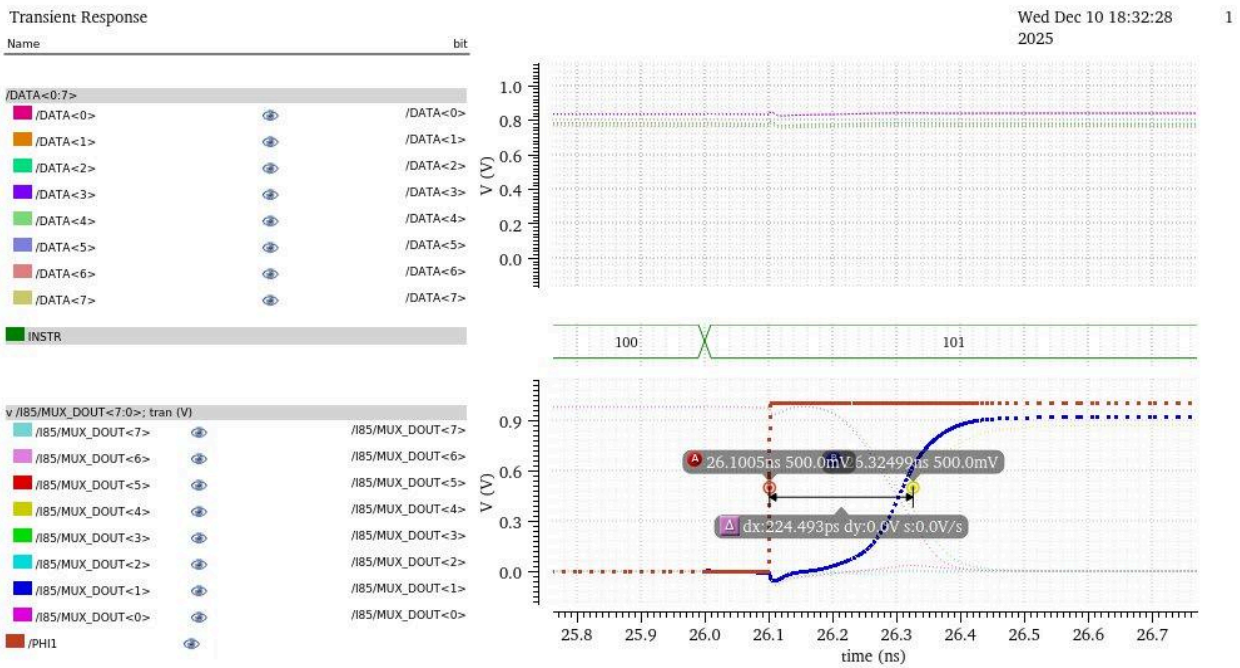
To shorten the simulation time, we modify the clock cycle to 8ns, so the adding goes less and it adds to 0000_0011. The final DATA results show the correct results.

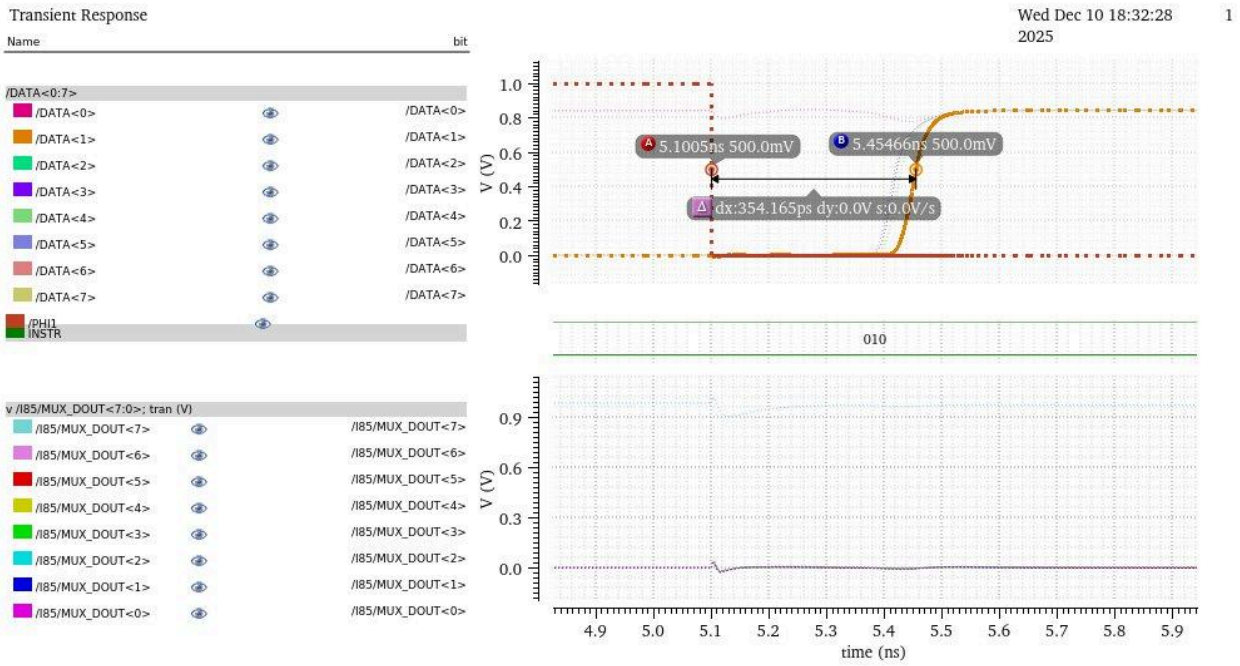




Critical Path

After testing, as is shown below, the delay of Adding is 224ps, while the delay of LOAD is 354ps. All other instructions has lower delay, making LOAD the critical path, which delay is 354ps.



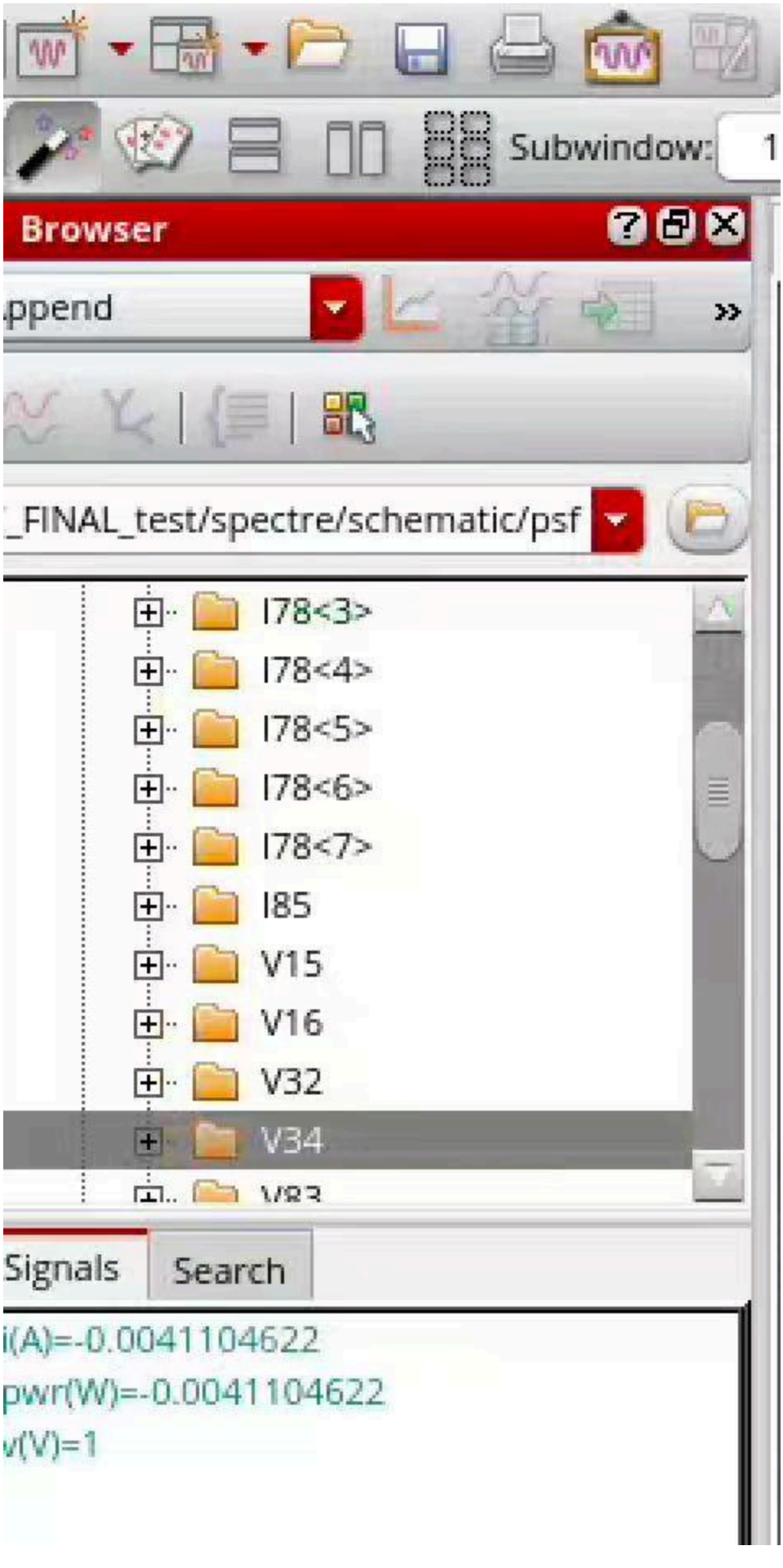


Area

172.195 um x 42.205um = 7267.49um^2

Power

When setting the clk cycle to 500ps, the power looks like:


Subwindow:
Browser
ppend
_FINAL_test/spectre/schematic/psf
I78<3>
I78<4>
I78<5>
I78<6>
I78<7>
I85
V15
V16
V32
V34
Signals
Search
i(A)=-0.0041104622
pwr(W)=-0.0041104622
v(V)=1