

# Design of a General Purpose Microprocessor

(Schematic and layout diagrams of main units -Perl)

*University of Southern California*

Aakash Barbhaya

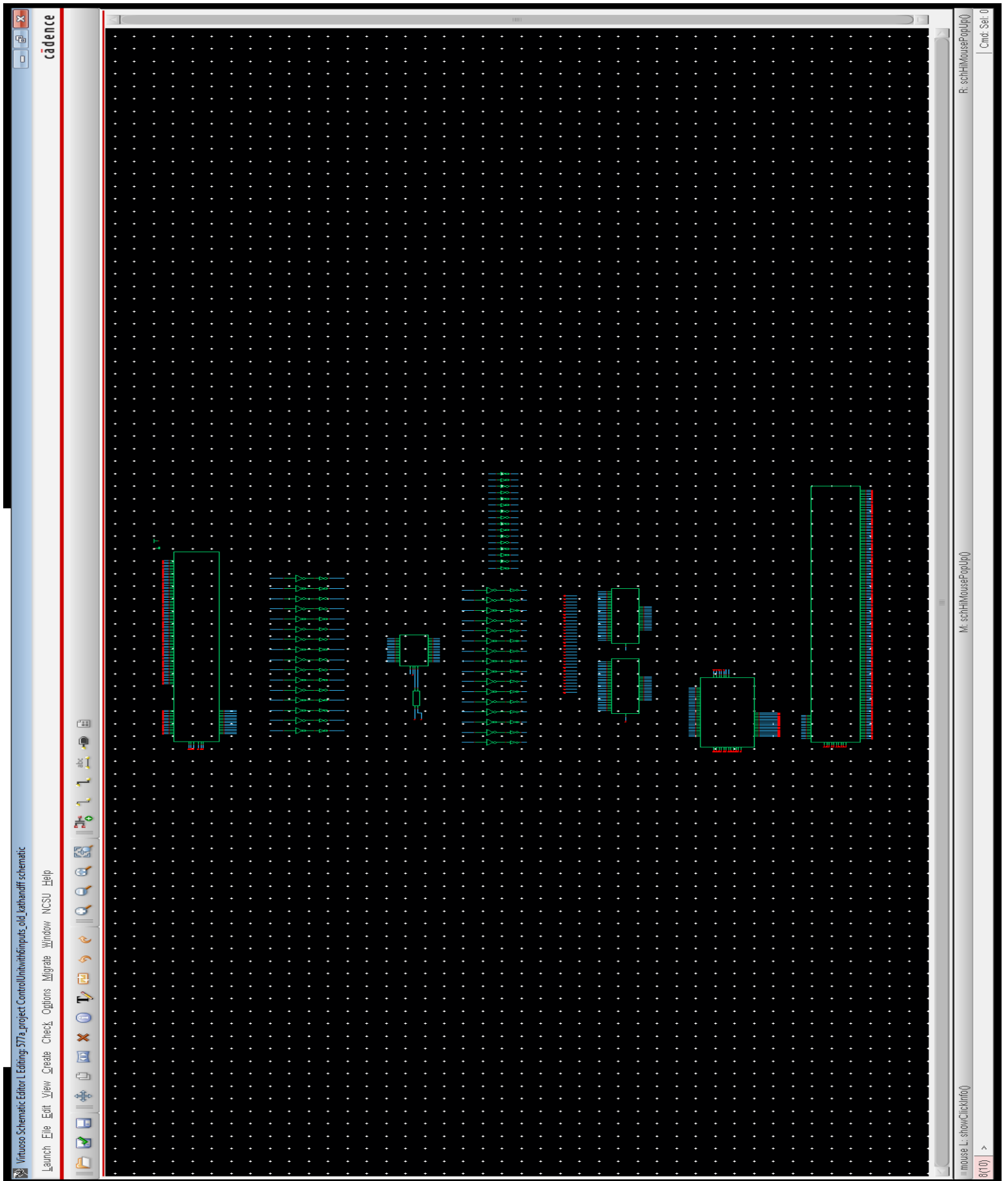
Abhishanga Upadhyay

Amruta Gadre

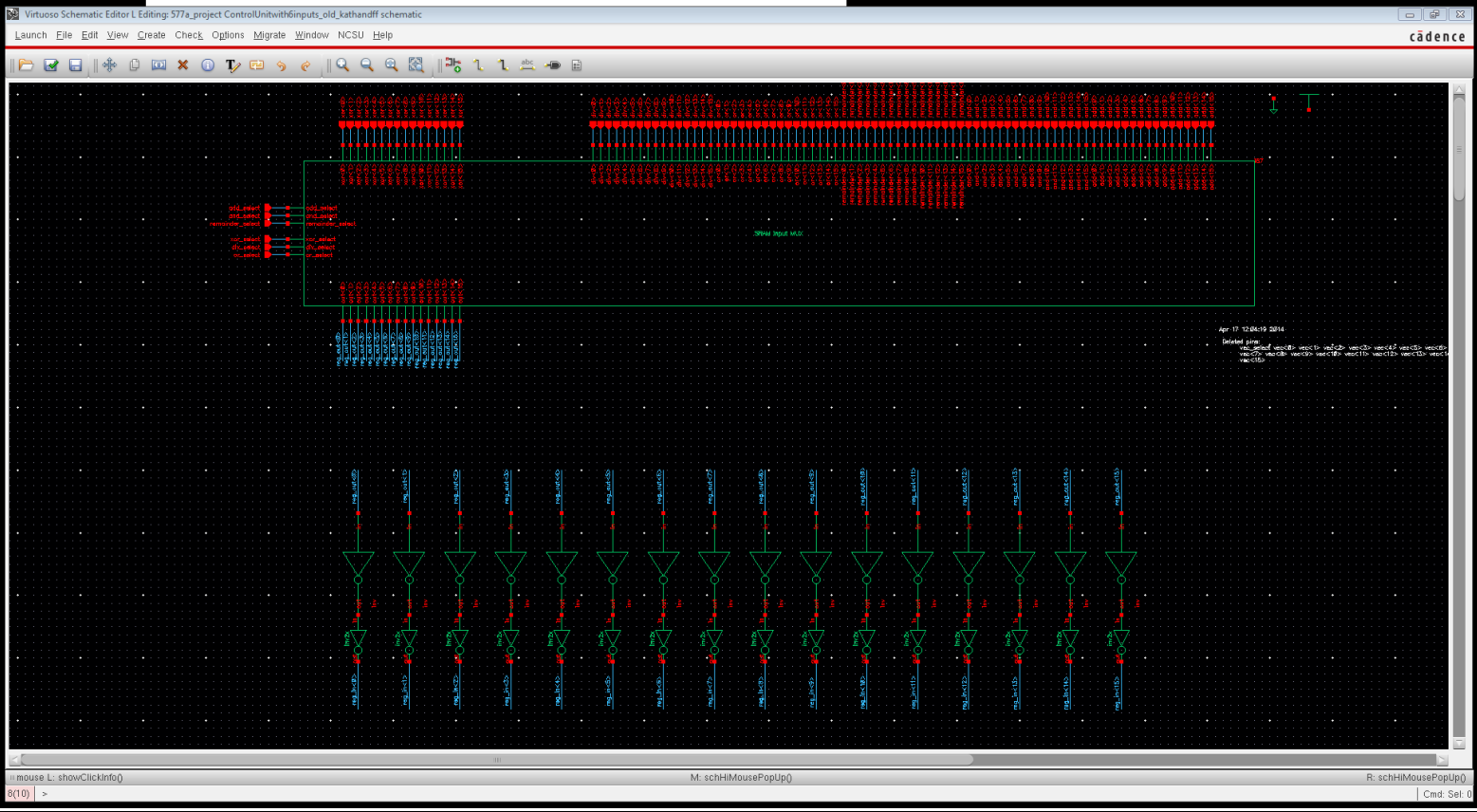
Kathan Shah

Malvika Goda Krishna

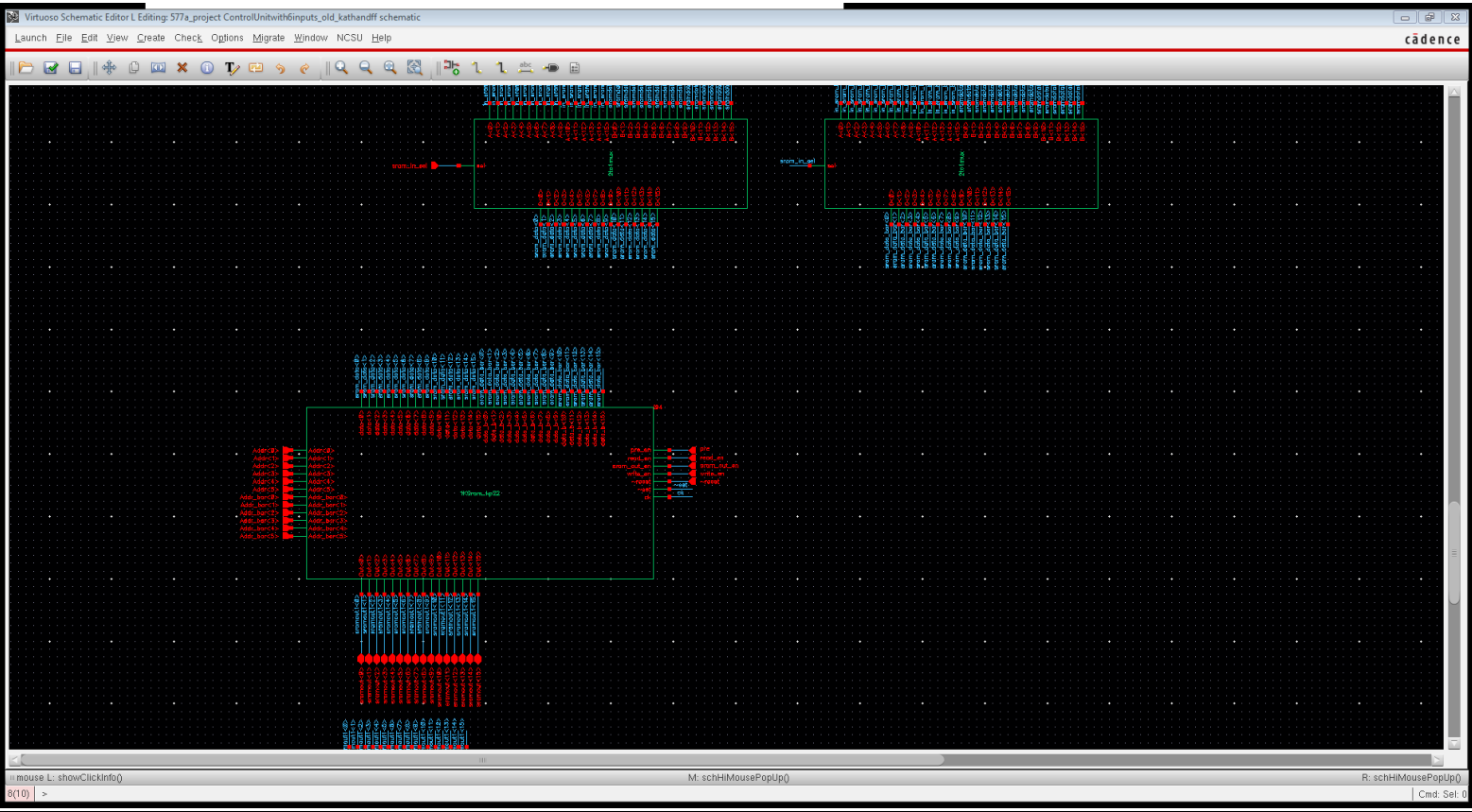
## Schematic of design



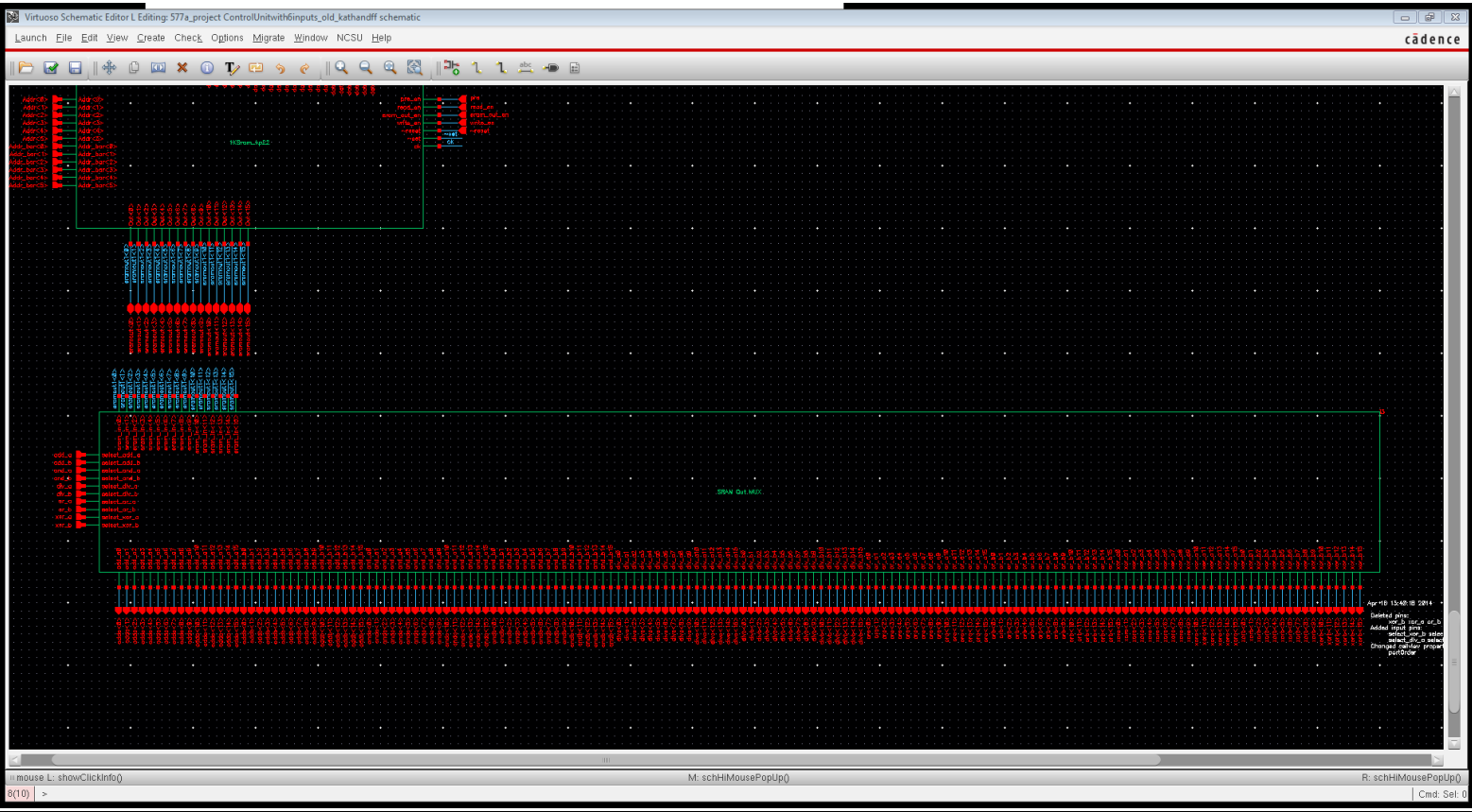
# SRAM input mux



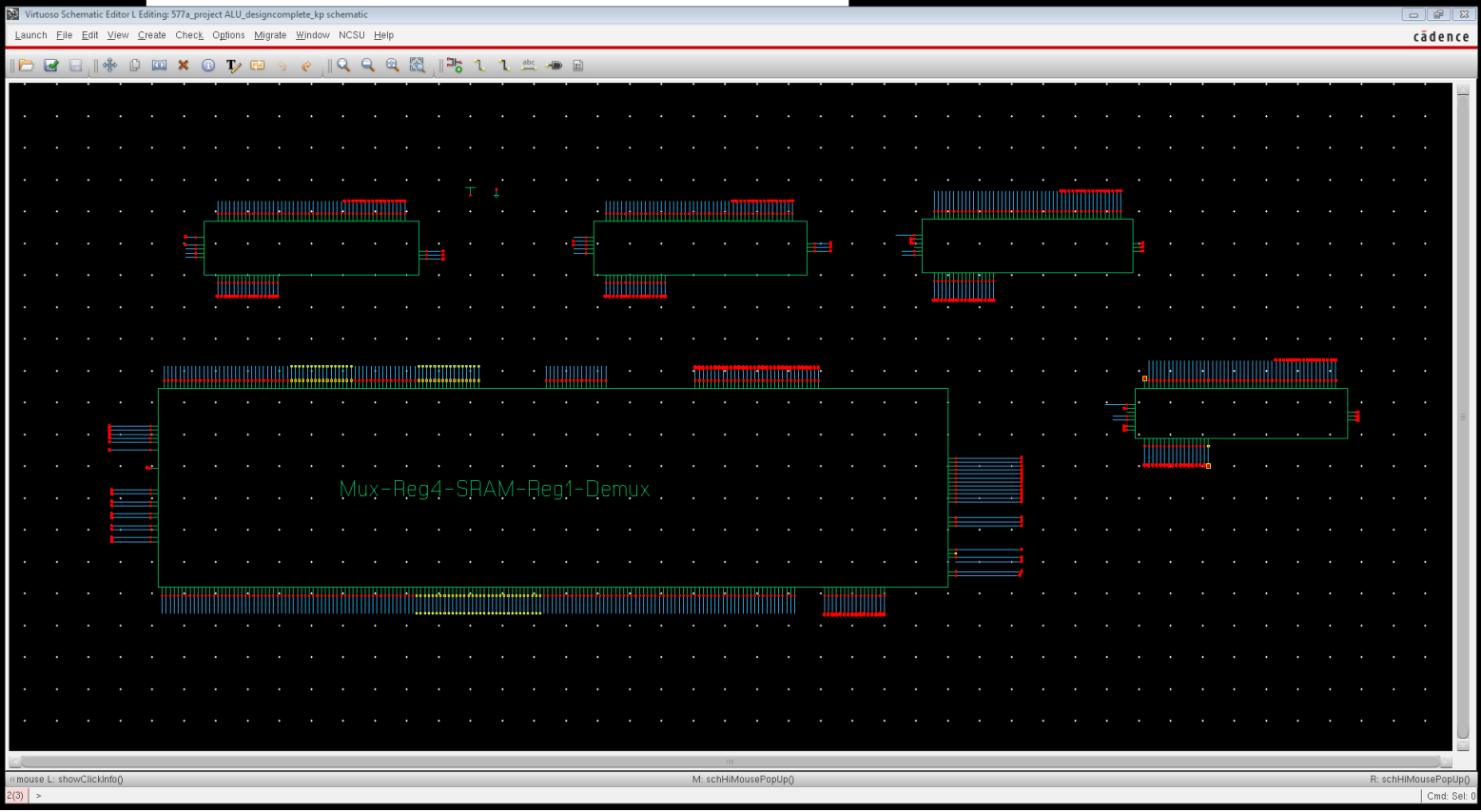
1kb SRAM with registers



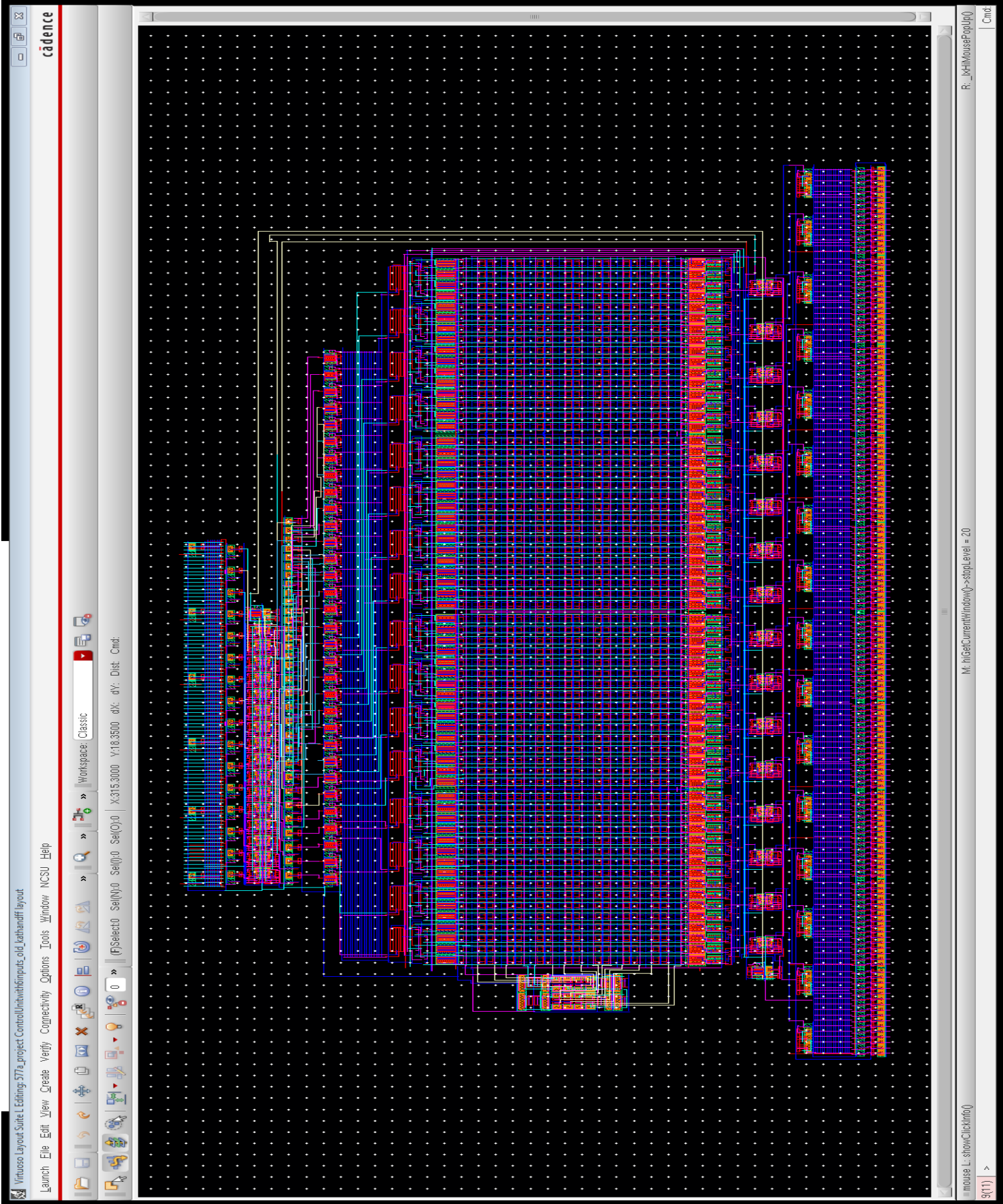
SRAM output mux



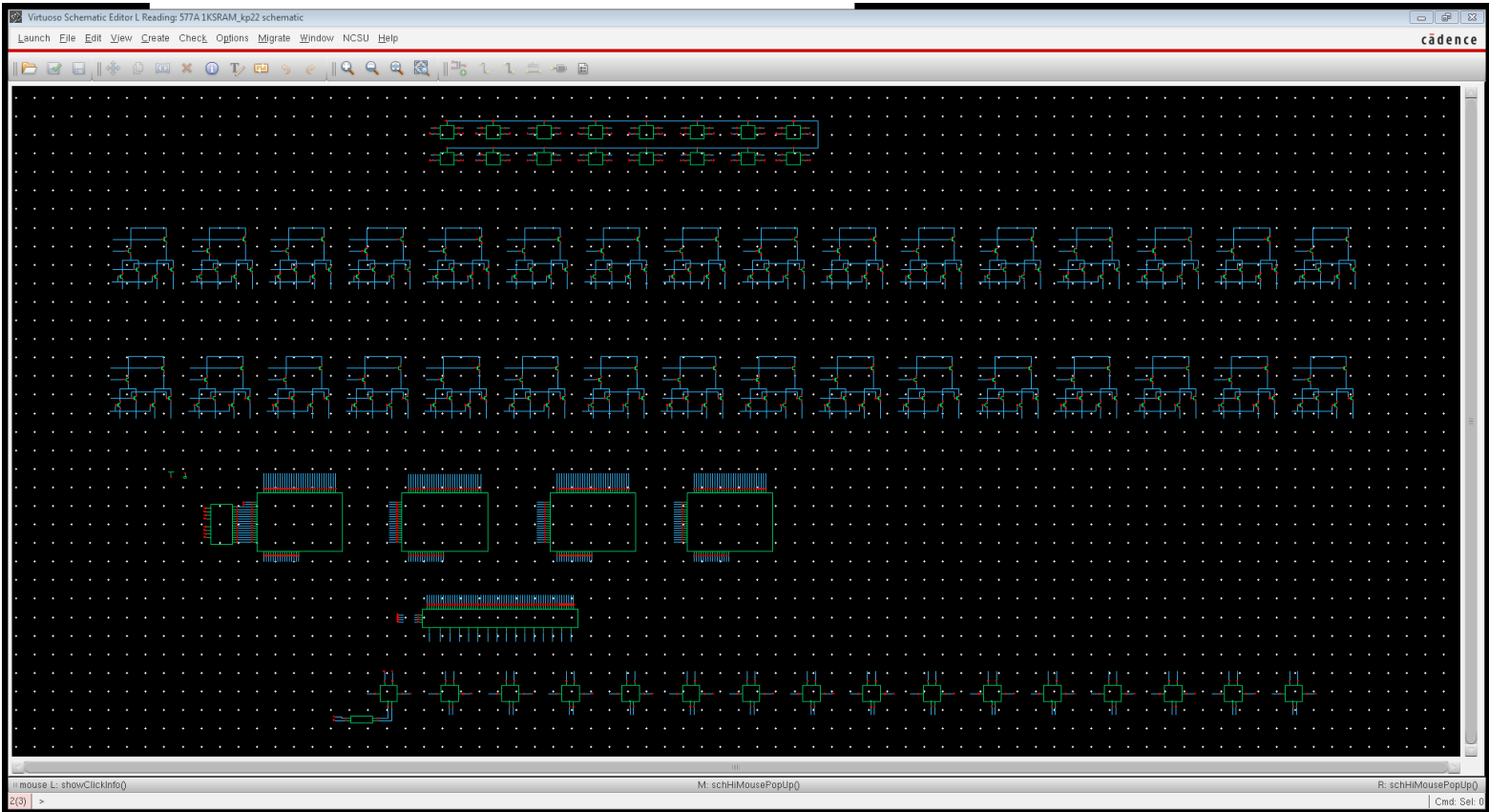
ALU



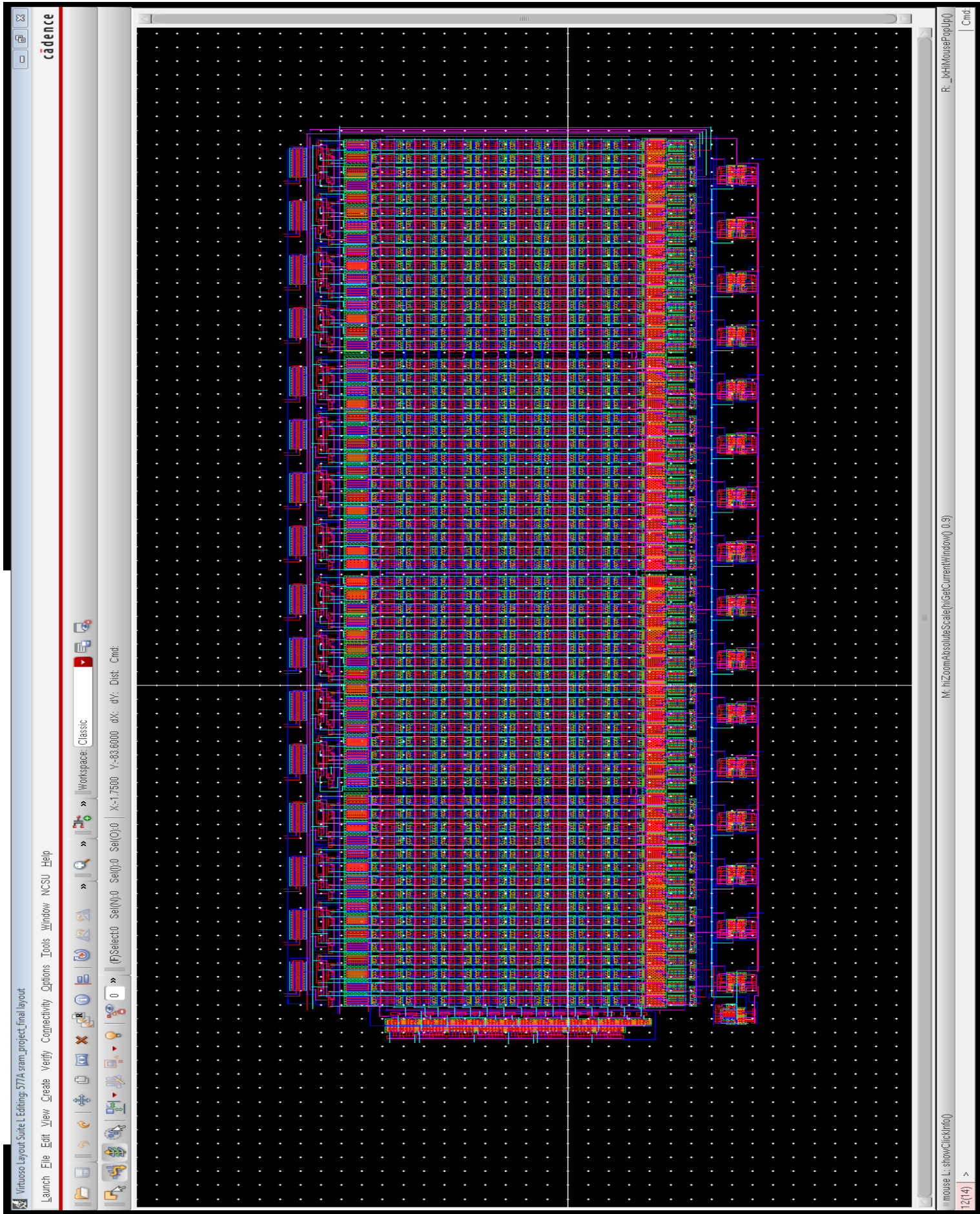
Layout of design



# SRAM

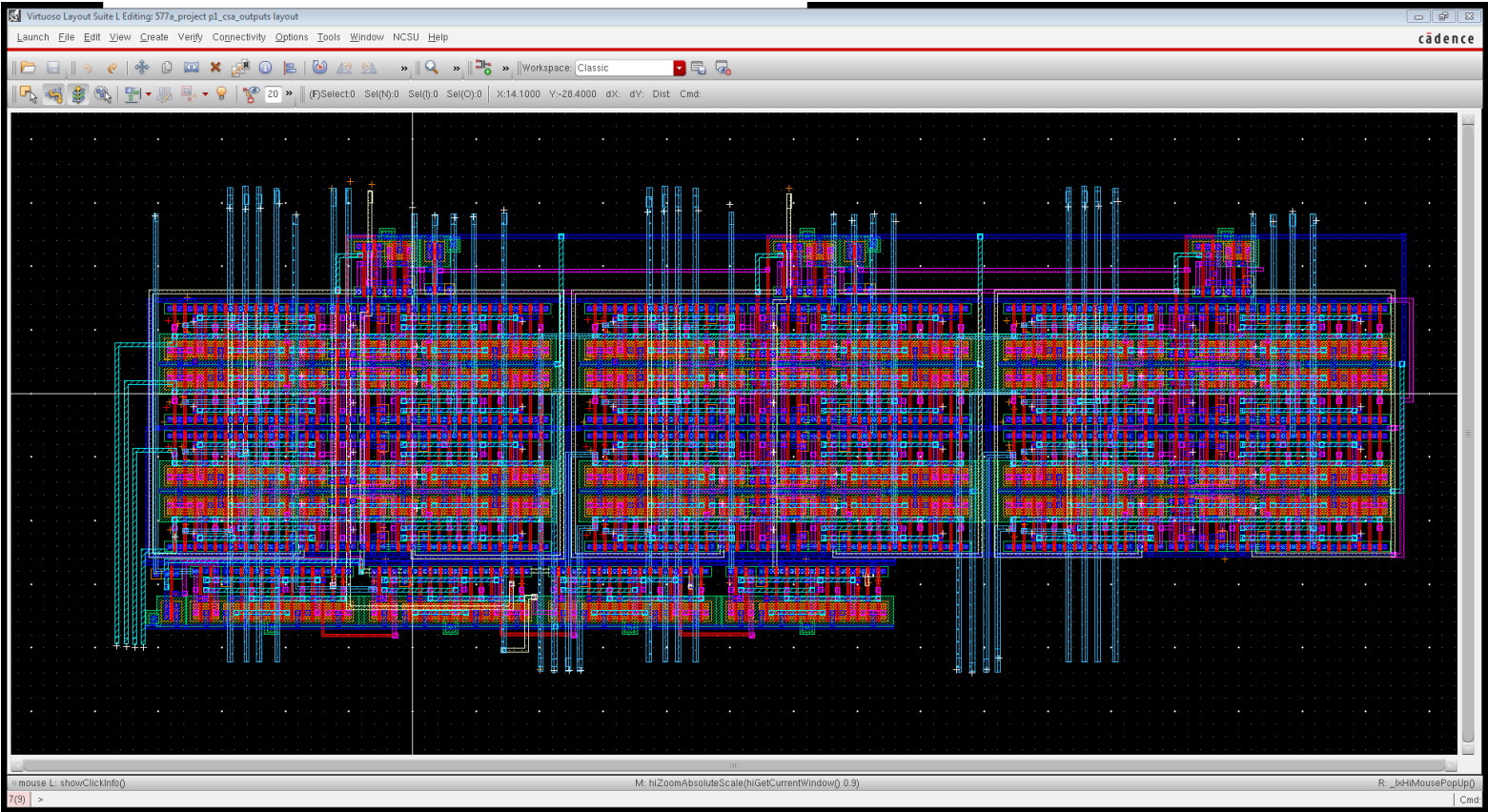
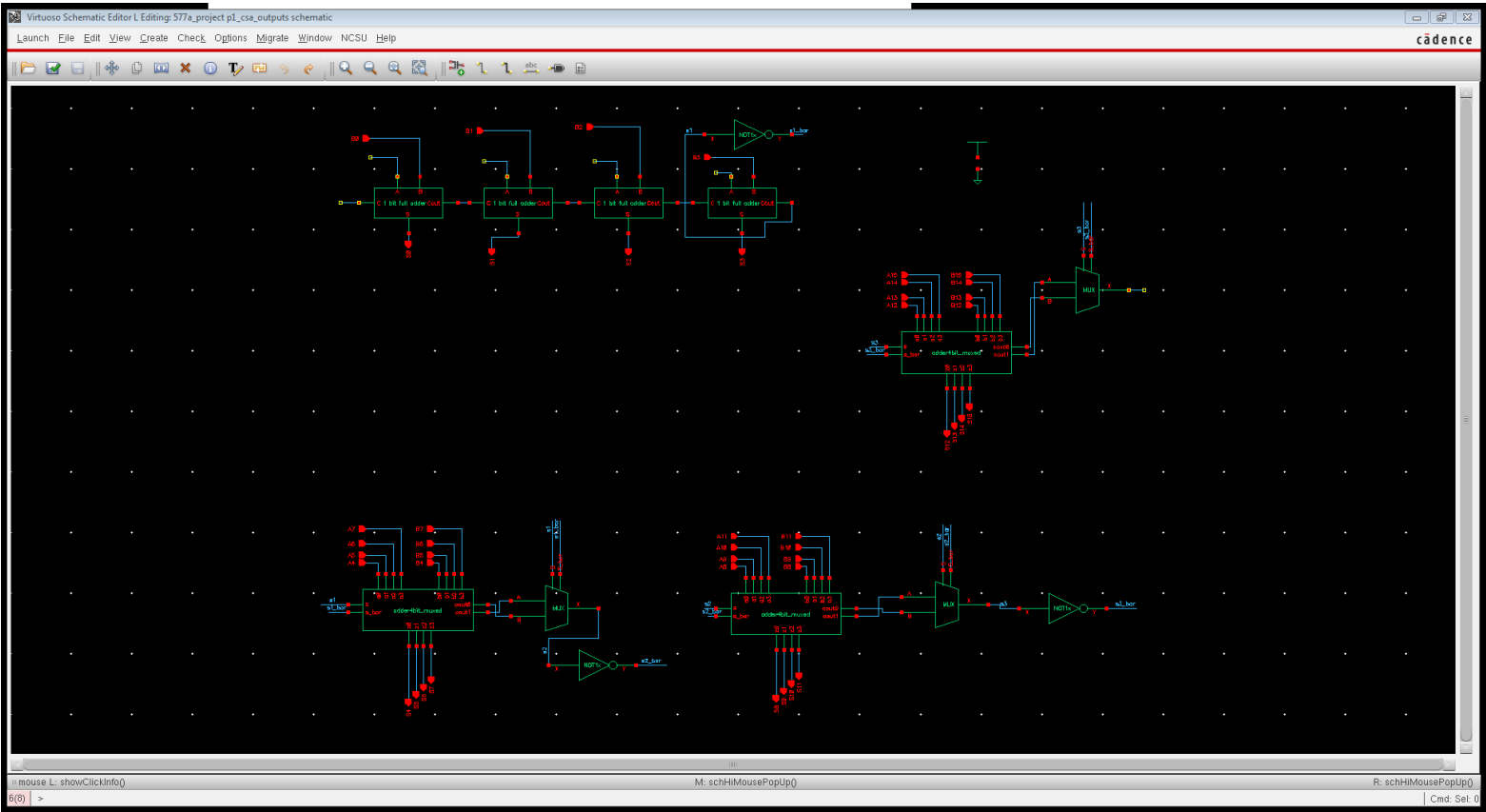




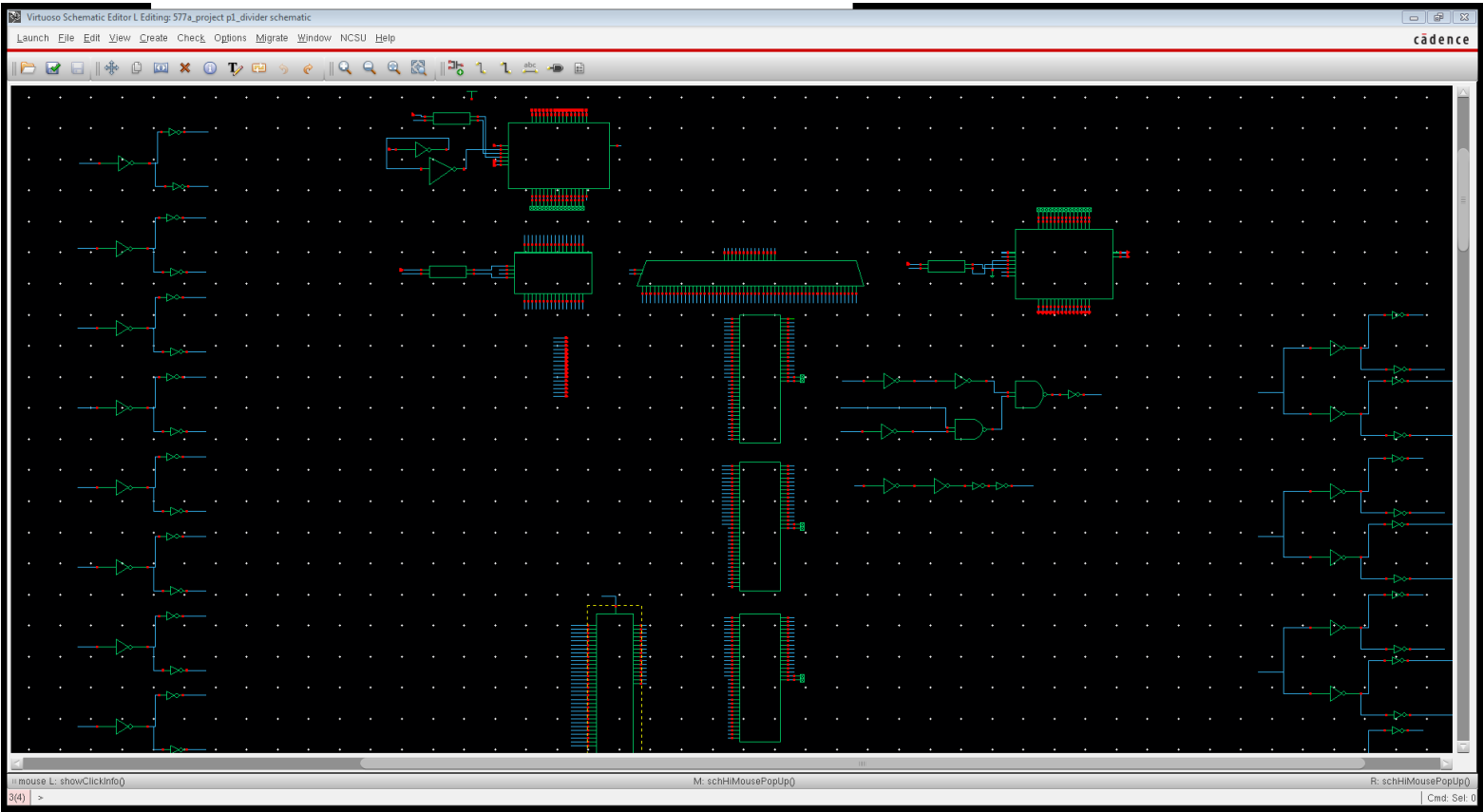
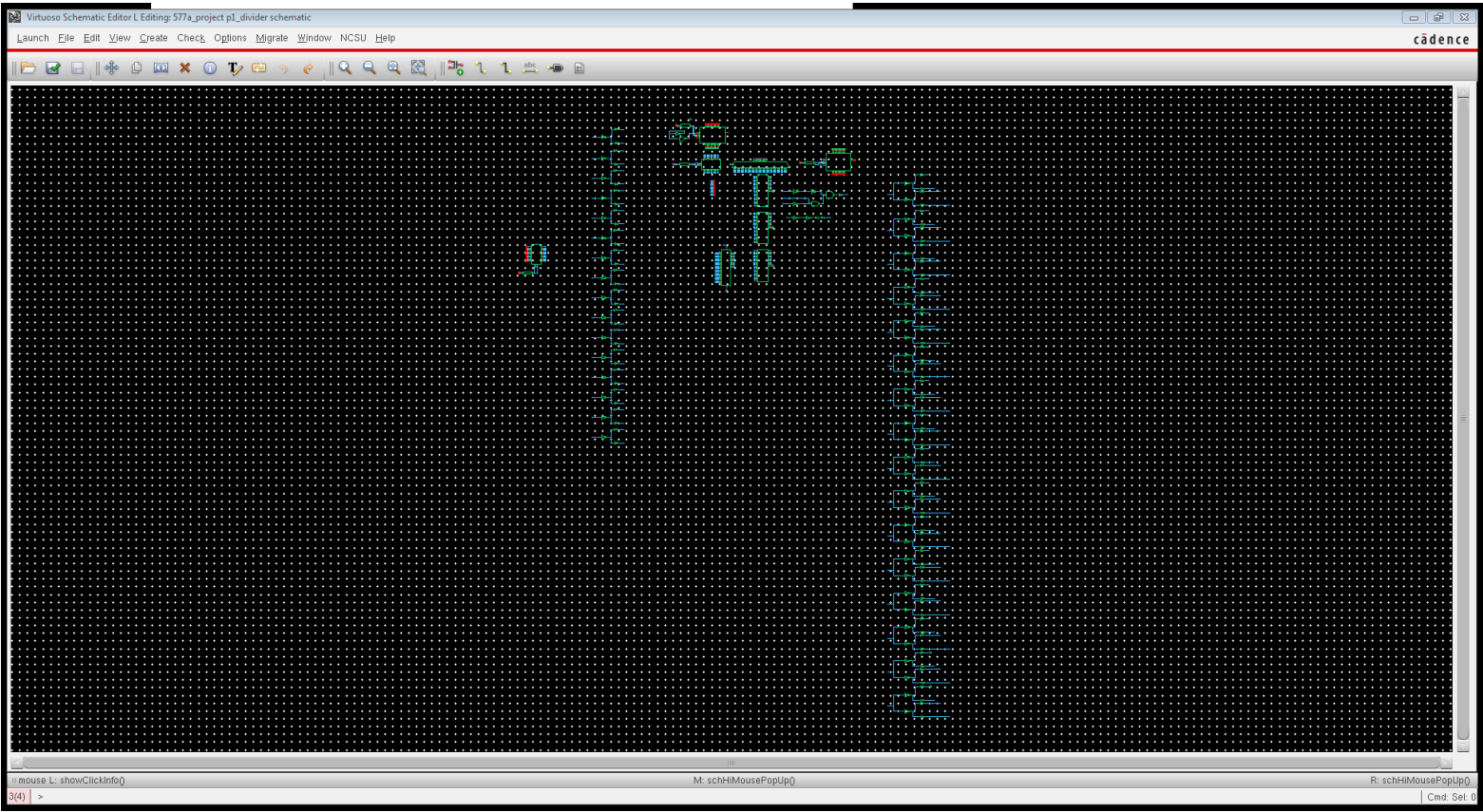


Execution units (ALU)

Adder



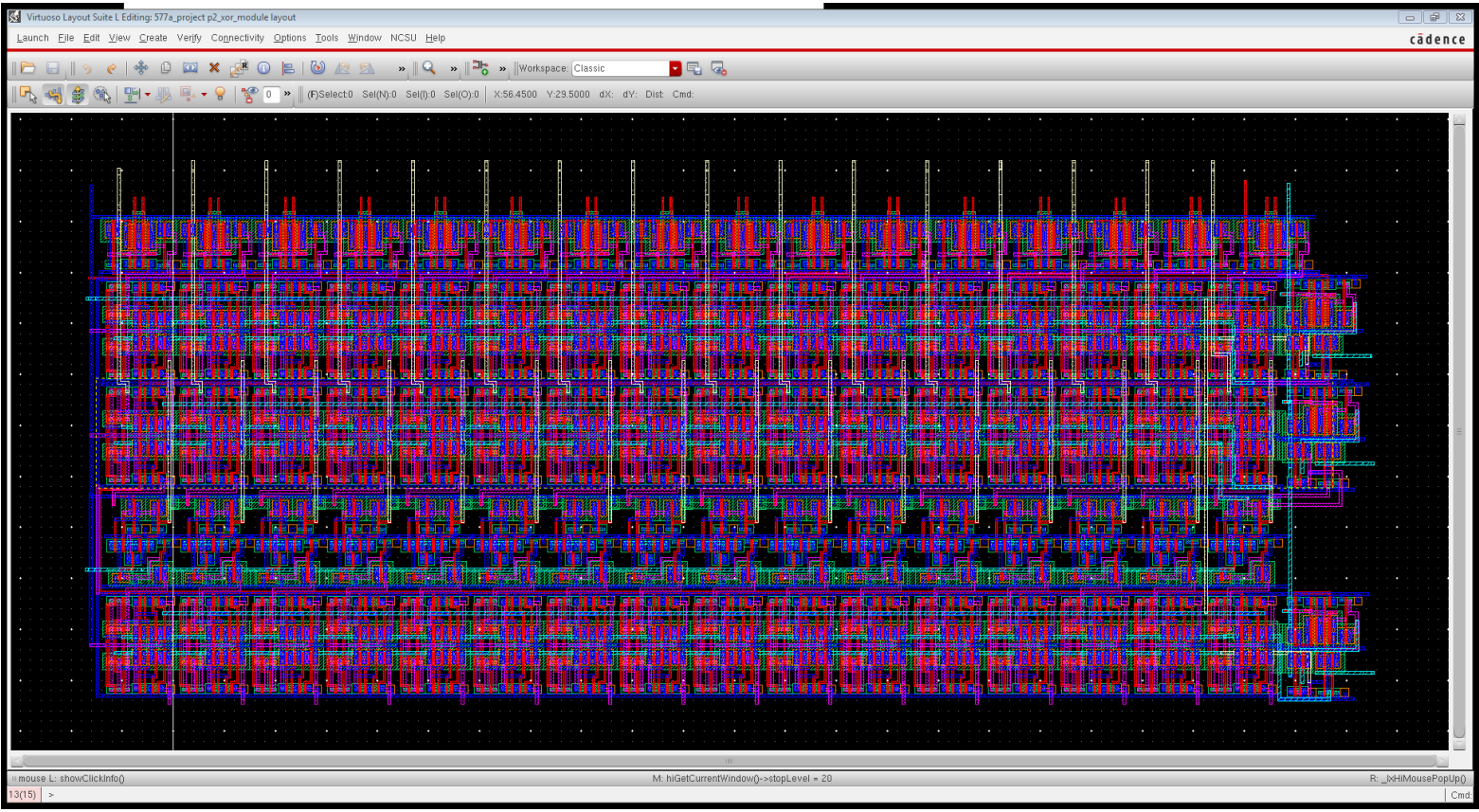
Divider (not integrated into design/separate module)



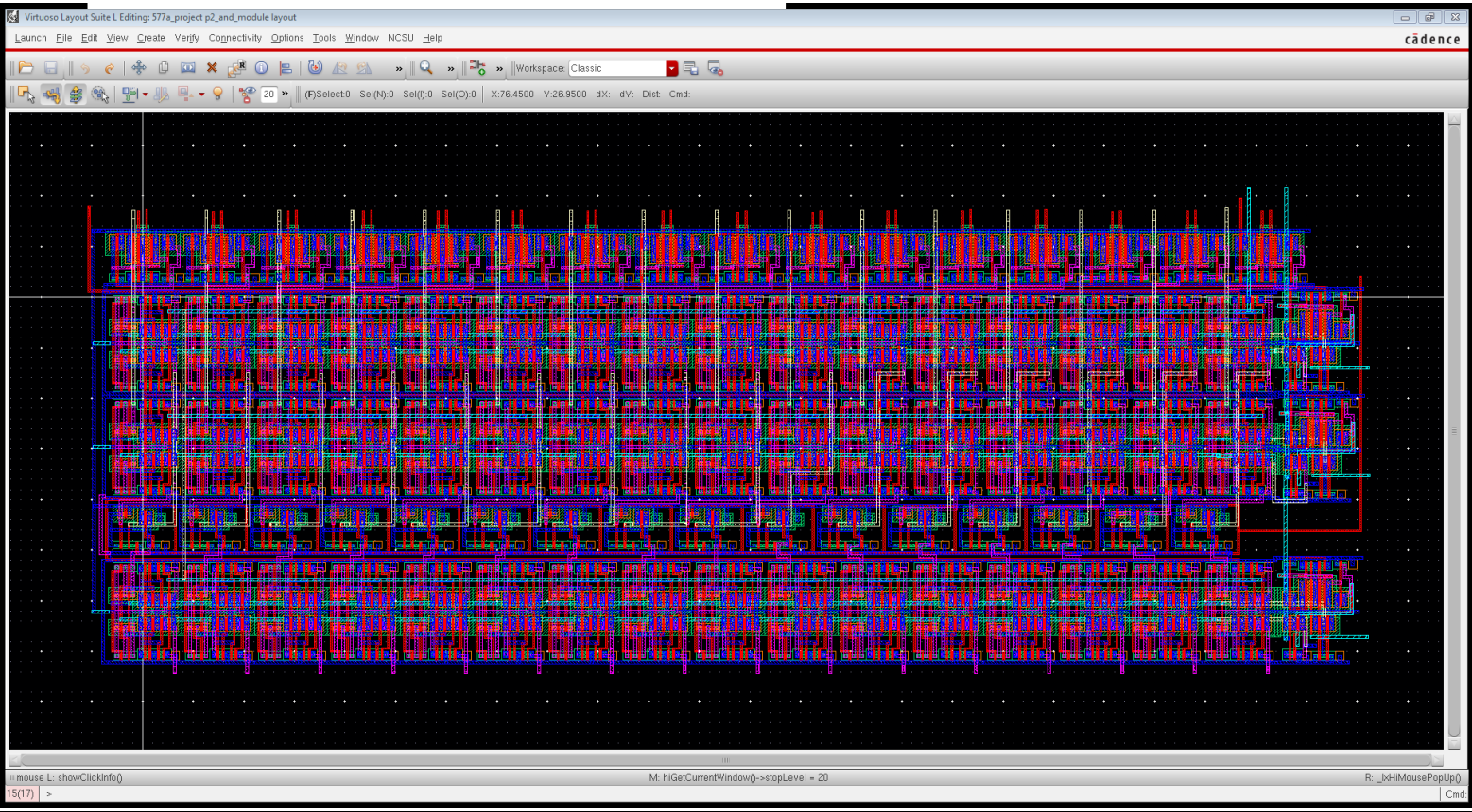




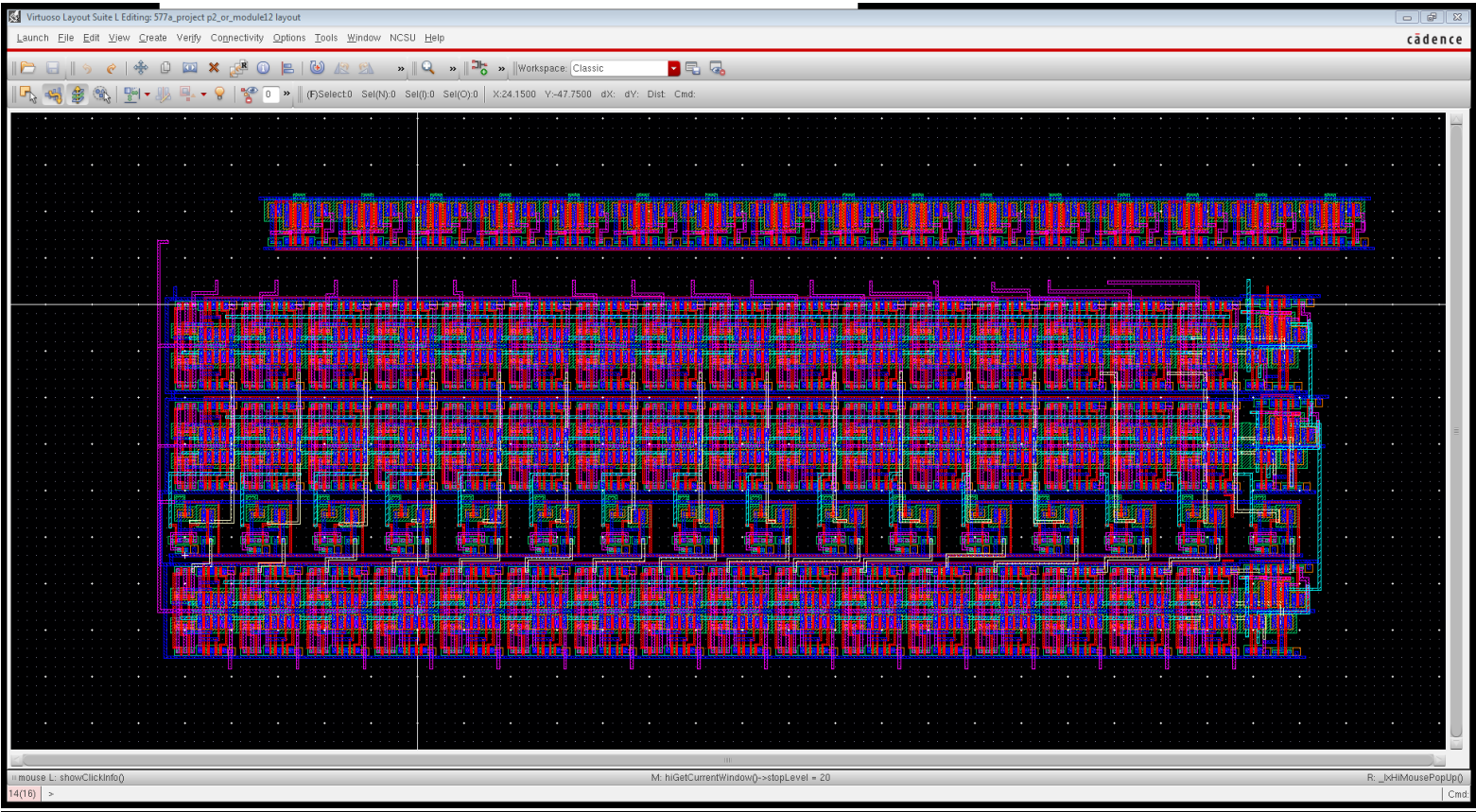
XOR dynamic



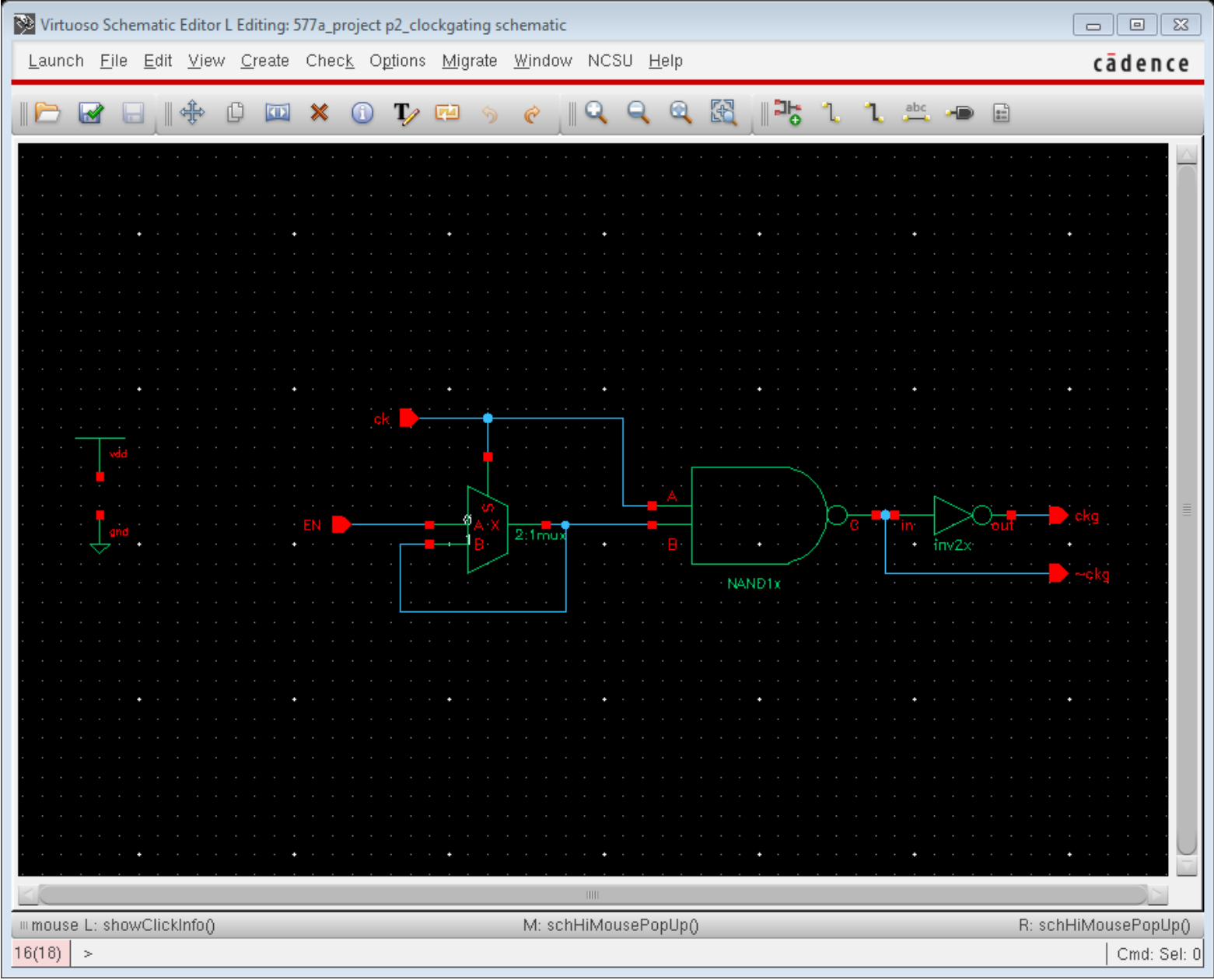
AND dynamic



OR dynamic



Clock gating for power minimization





### Vector file generated by perl (Snapshot)

```
vector.vup.ec - /cdfs/simfiles_kp - gedit
File Edit View Search Tools Documents Help
New Open Save Print... Undo Redo Cut Copy Paste Find Replace
vectormaker.pl | vector.vup.vec
radix 111 2 2 4 4 4444 4444 1111111111 11 11 11 111 4444 4444 4444 4444 4444 111111 111 111 111 111 111 111
vname pre_read_en write_en Addr=[5:4] Addr_bar=[5:4] Addr=[3:0] Addr_bar=[3:0] insram_bar=[0:3] insram_bar=[4:7] insram_bar=[8:11] insram_bar=[12:15] insram_bar=[0:3] insram_bar=[4:7] insram_bar=[8:11] insram_bar=[12:15] add_a
add_b_and_a_and_b_div_a_div_b_or_a_or_b_xor_a_xor_b_and_phi_select_andi_or_phi_select_ori_xor_phi_select_xori_previous_dep add_latch_en select_addi_andi=[15:12] andi=[11:8] andi=[7:4] andi=[3:0] add_select
ori=[15:12] ori=[7:4] or=[11:8] xor=[7:4] addi=[11:8] addi=[3:0] divi=[11:8] divi=[7:4] divi=[3:0] sram_in_en add_b_en add_out_en sram_in_en add_b_en add_out_en sram_in_sel
slope 0.1
vsh tunit 1.8
ns
;time sram addr data datab demux anddyn ordyn xordyn add andi ori xori addi divi mux enable
0 000 030 F ABCD 5432 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 000001 0000000000000010
0.8 101 030 F ABCD 5432 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 000001 0000000000000010 ; writing for STORE 00 ABCD
3 000 031 E CDEF 3210 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 000001 0000000000000010
3.8 101 031 E CDEF 3210 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 000001 0000000000000010 ; writing for STORE 01 CDEF
6 000 030 F 0000FFFF 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1000000000000010
6.8 100 030 F 0000FFFF 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1000000000000010
7.2 030 F 0000FFFF 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1000000000000010 ; reading
9 000 034 B 0000FFFF 001000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1100000000000010
9.8 100 034 B 0000FFFF 001000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1100000000000010
10.2 110 034 B 0000FFFF 001000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1100000000000010 ; reading
12 000 0345 A345A345 000100000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 0110000000000010 ; demux
15 000 0345 A345A345 000100000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 0010000000000110
15.6 000 0345 A345A345 000100000000 10 00 00 000 A345 A345 A345 A345 A345 A345 010000 0010000000000110 ; AND operation
18 000 0345 A345A345 000100000000 10 00 00 000 A345 A345 A345 A345 A345 A345 010000 0000000000000111 ; mux
21 000 120 F FFFF0000 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 0000000000000111
21.8 101 120 F FFFF0000 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 0000000000000111 ; writing
24 000 031 E 0000FFFF 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1000000000000010
24.8 100 031 E 0000FFFF 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1000000000000010
25.2 110 031 E 0000FFFF 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1000000000000010 ; reading
27 000 035 A 0000FFFF 001000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1100000000000010
27.8 100 035 A 0000FFFF 001000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1100000000000010
28.2 110 035 A 0000FFFF 001000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 1100000000000010 ; reading
30 000 0345 A345A345 000100000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 0110000000000010 ; demux
33 000 0345 A345A345 000100000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 0010000000000110
33.6 000 0345 A345A345 000100000000 10 00 00 000 A345 A345 A345 A345 A345 A345 010000 0010000000000110 ; AND operation
36 000 0345 A345A345 000100000000 10 00 00 000 A345 A345 A345 A345 A345 A345 010000 0000000000000111 ; mux
39 000 121 E FFFF0000 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 0000000000000111
39.8 101 121 E FFFF0000 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010000 0000000000000111 ; writing
42 000 120 F 0000FFFF 000000000000 00 00 00 000 A345 A345 A345 A345 A345 A345 010001 0000000000000111
```

Ln 28, Col 27 IN\$

## Perl program snippet (Snapshot)

[illegible]