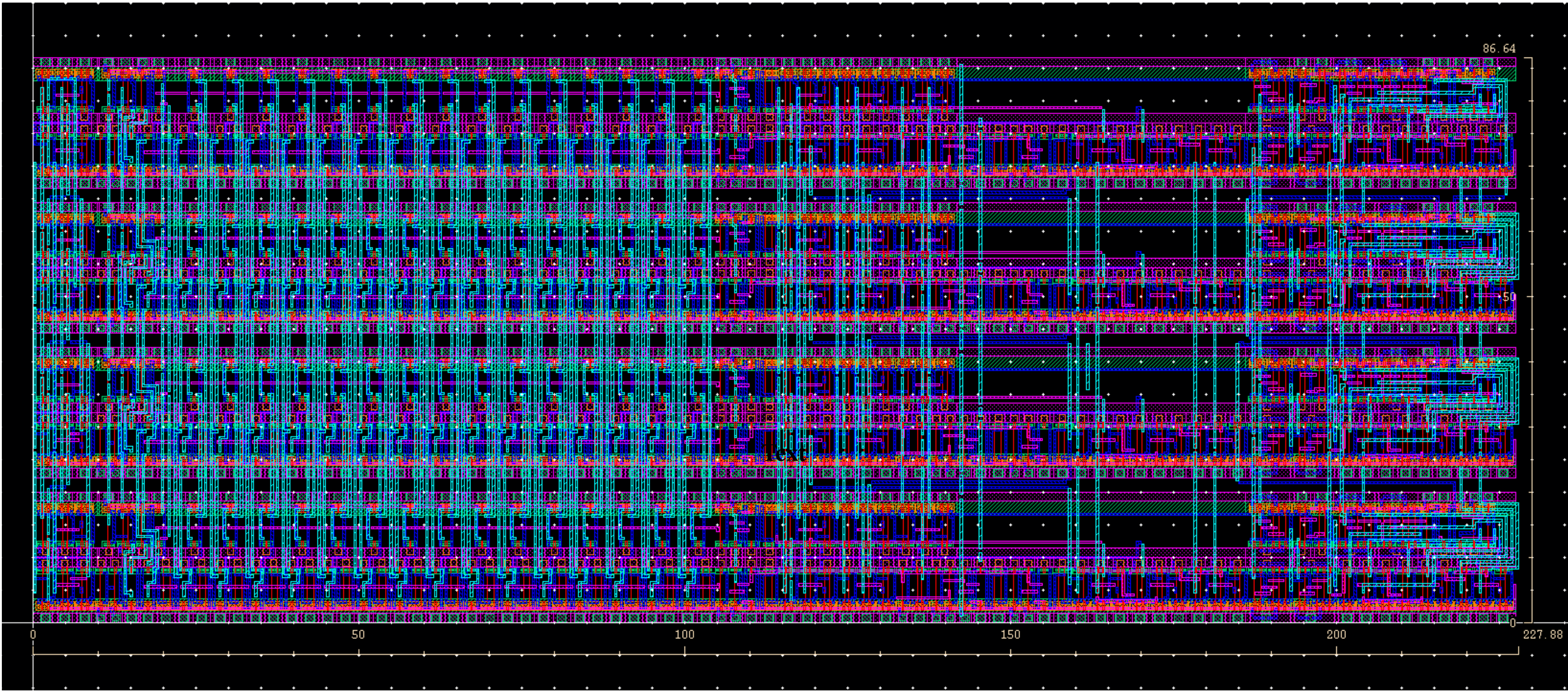
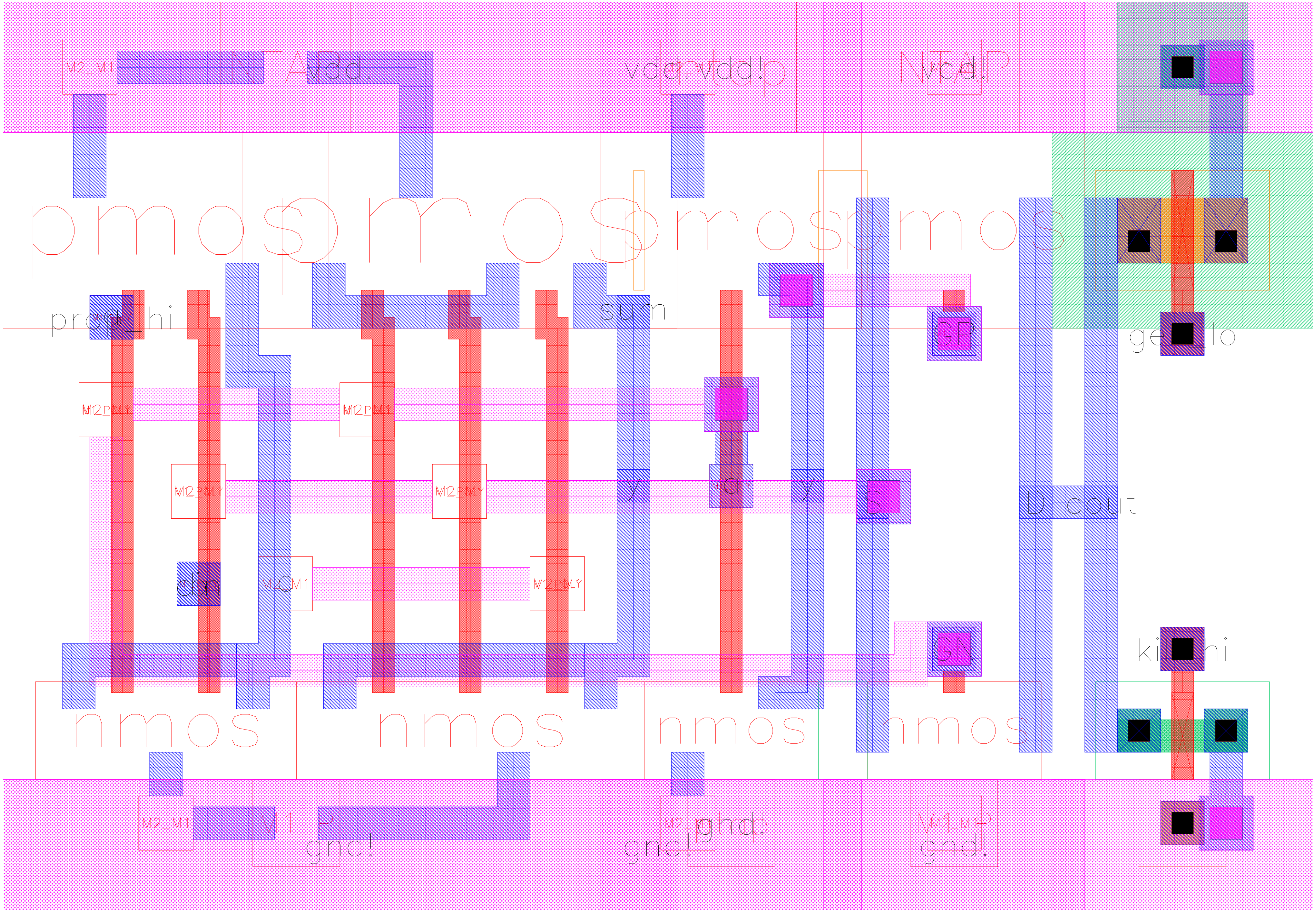
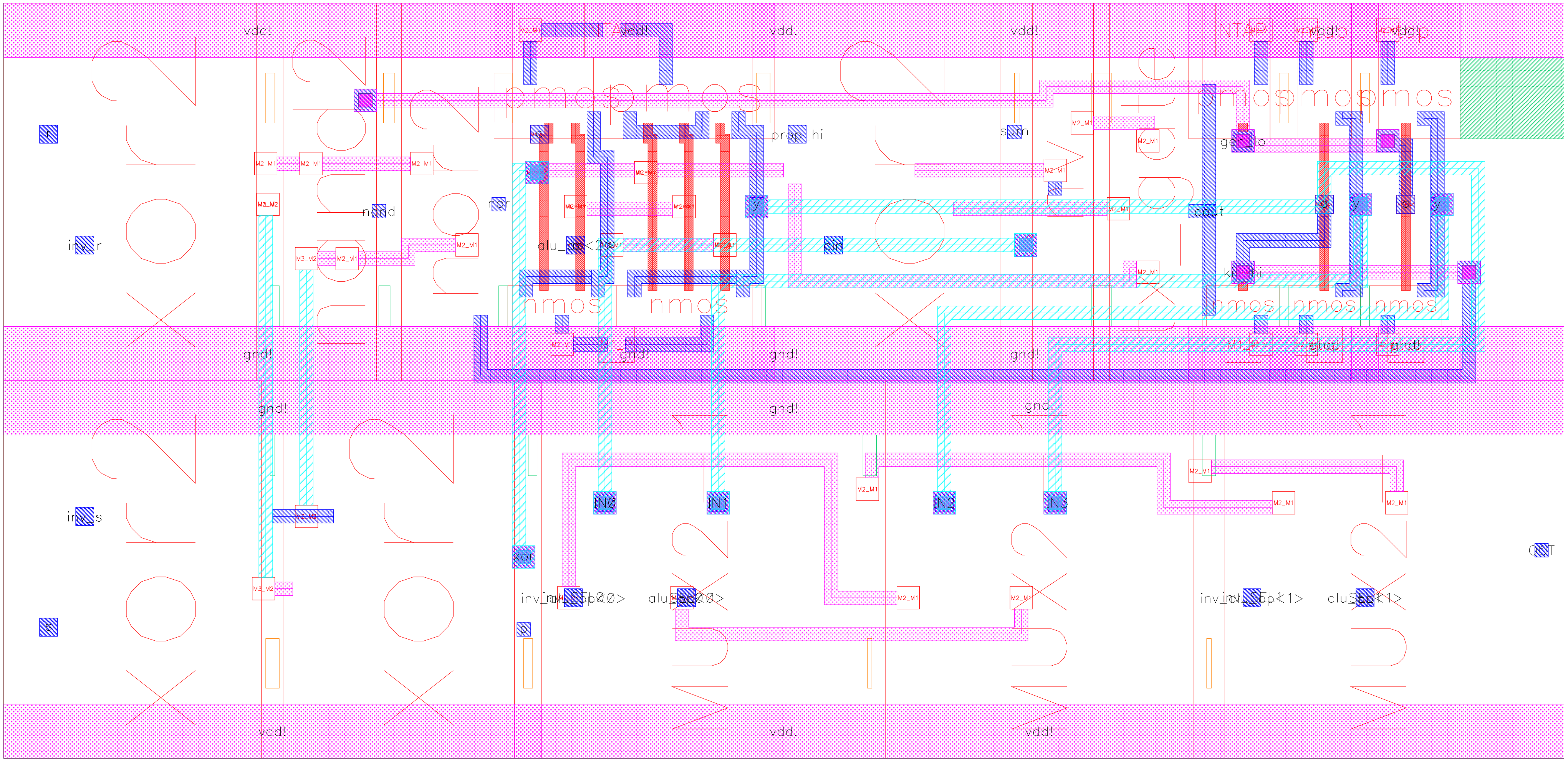


Area of ‘datapath’ layout: 227.88 [units] * 86.64 [units] = 19743.5232 [units^2]

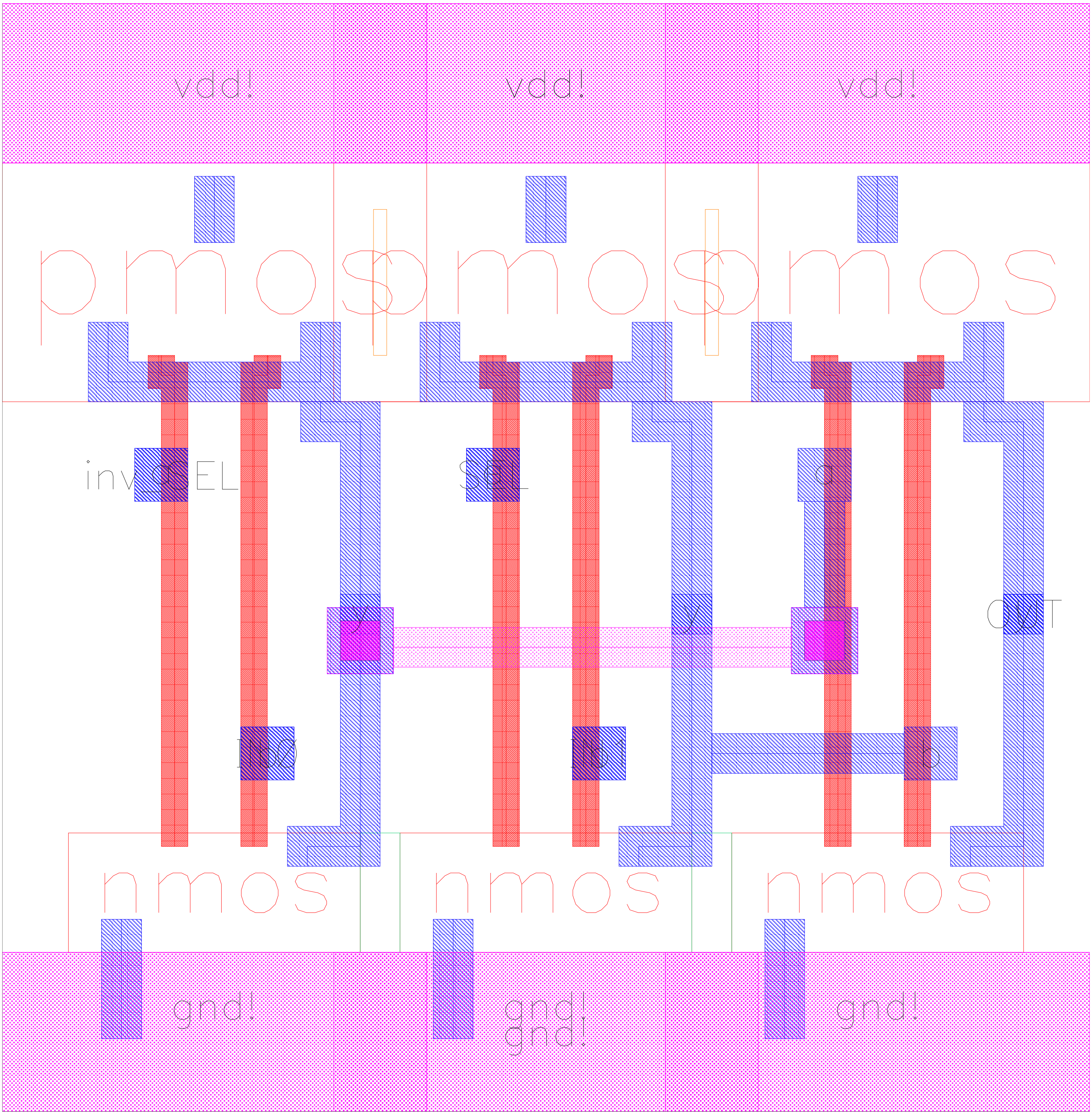
ADD Cell Layout (Level 1 Heirarchy)



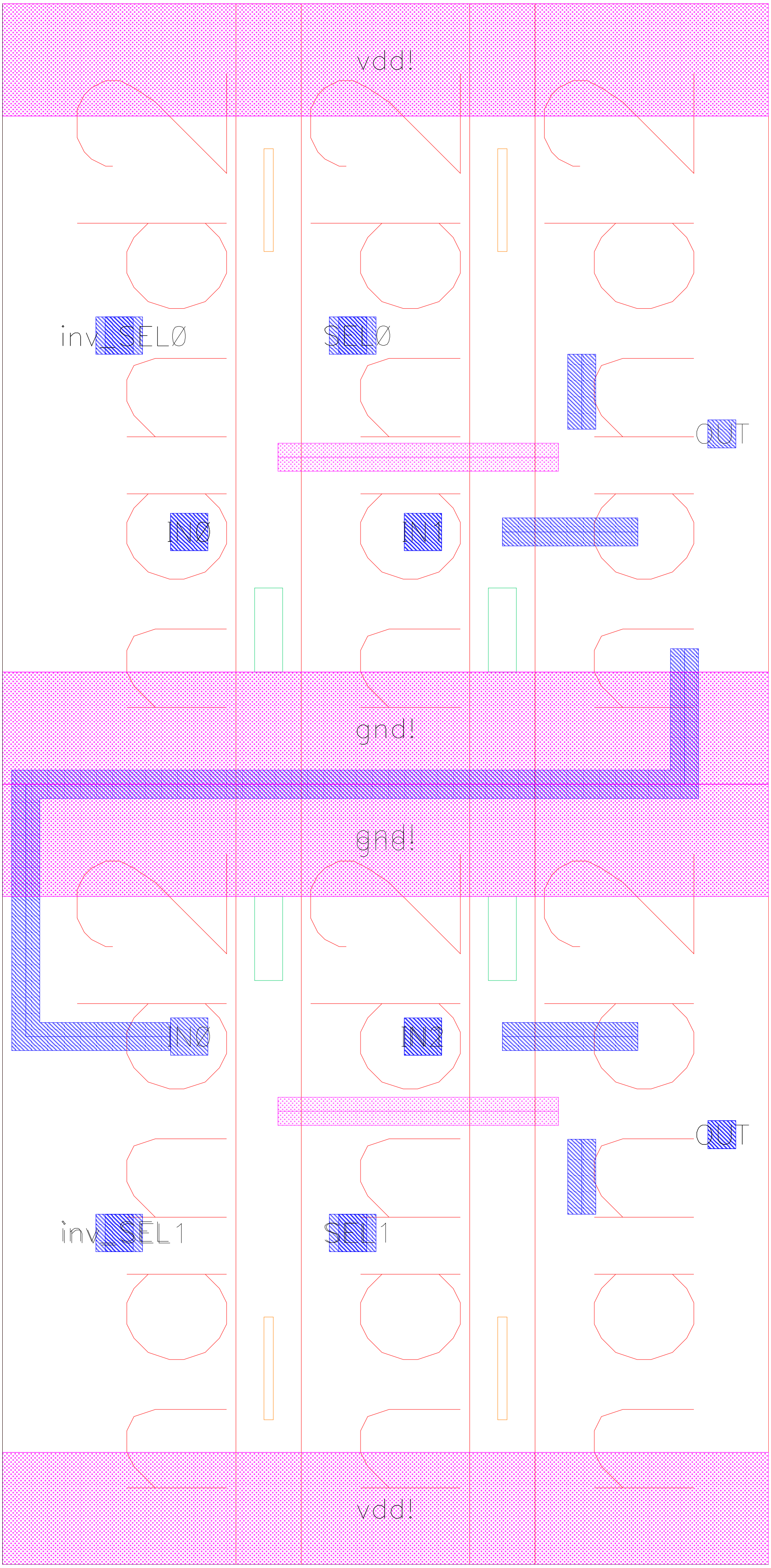
ALU Cell Layout (Level 1 Heirarchy)



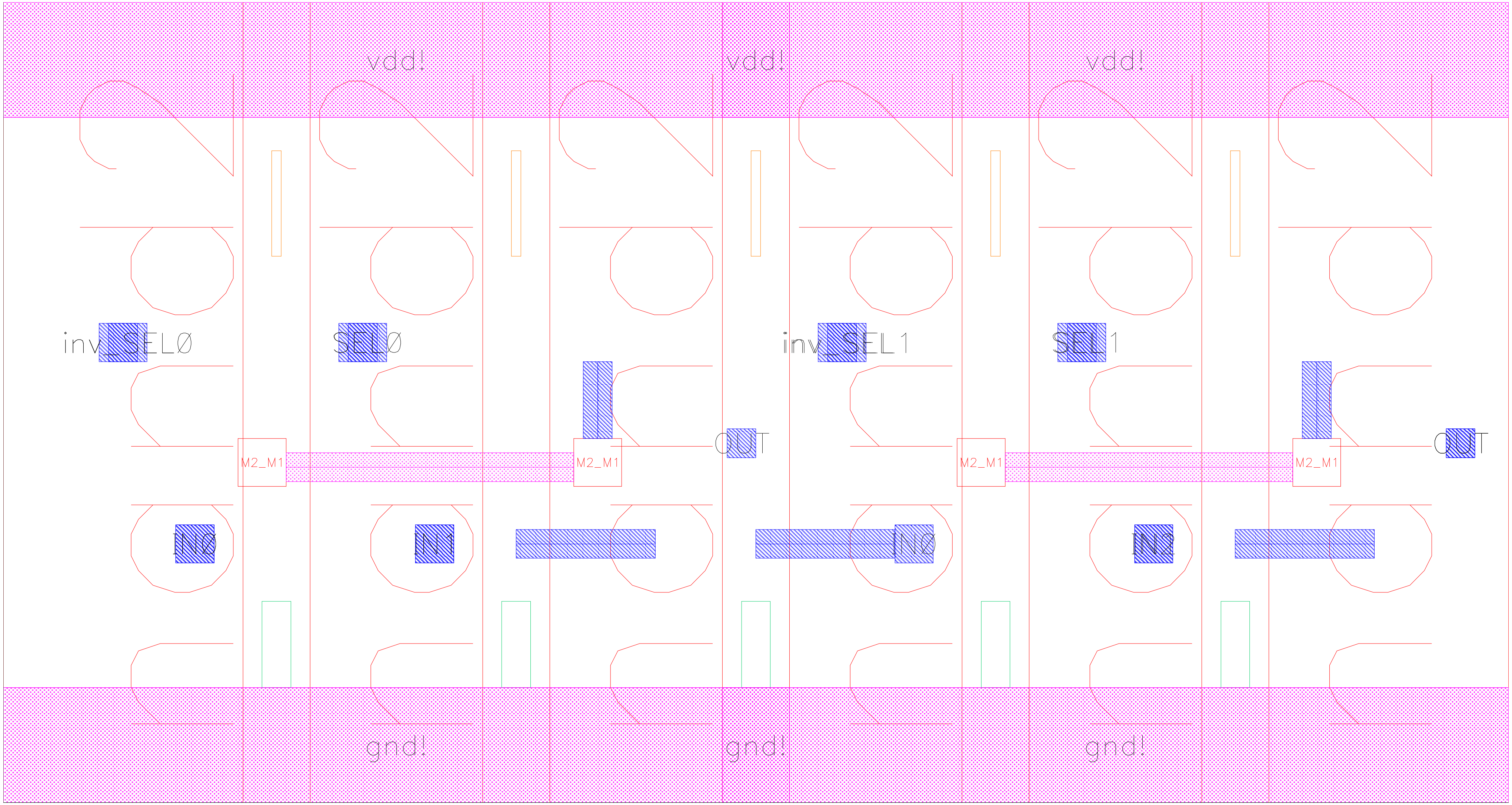
MUX2-1 Cell Layout (Level 1 Hierarchy)



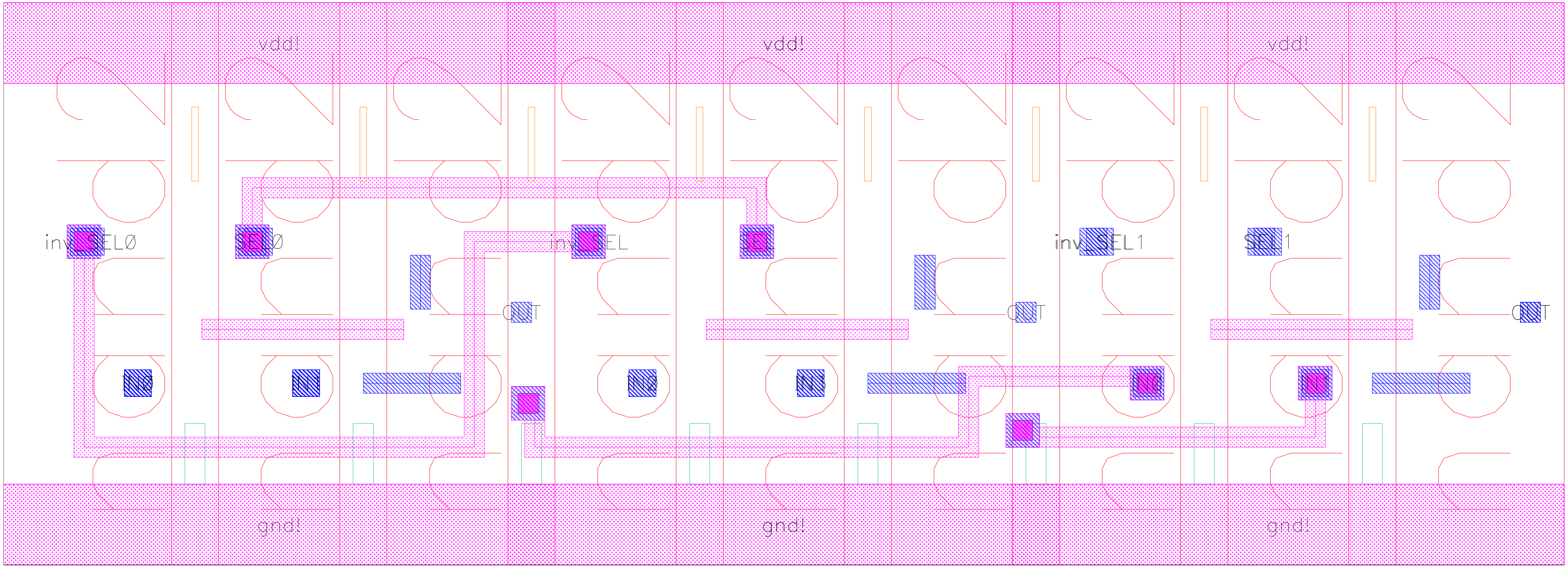
MUX3-1_v1 "Vertical" Cell Layout (Level 1 Hierarchy)



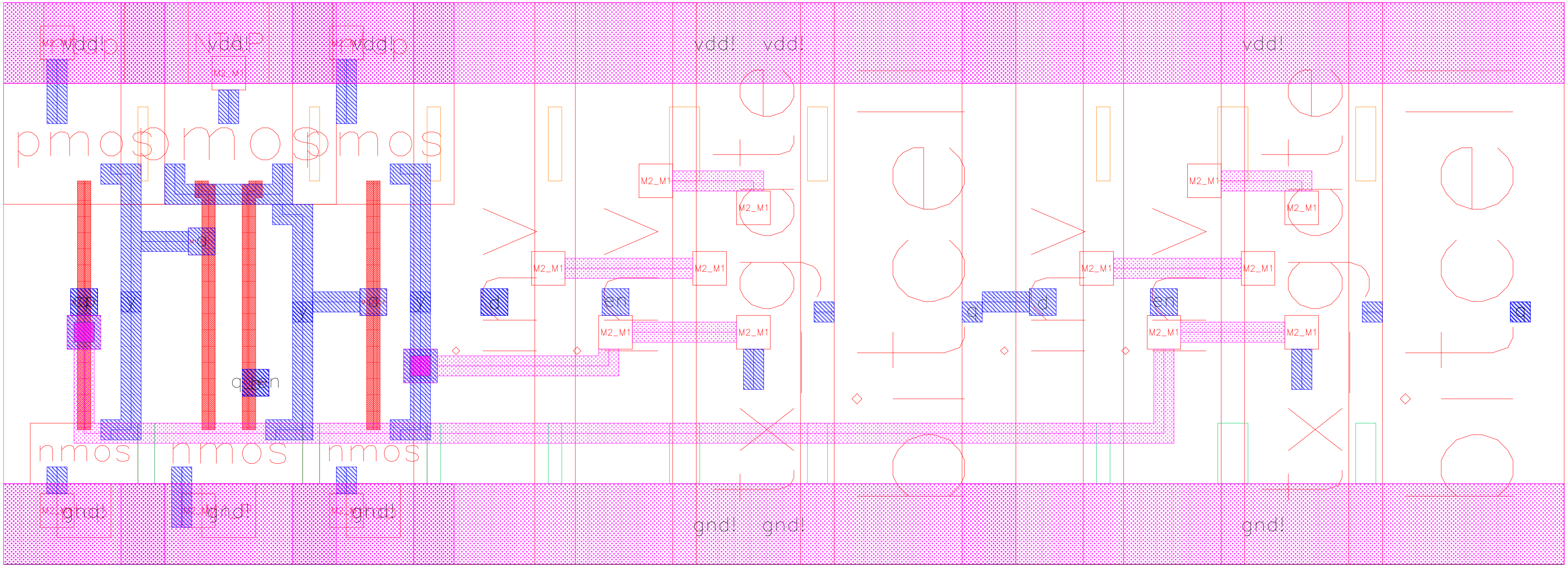
MUX3-1_v2 "Horizontal" Cell Layout (Level 1 Hierarchy)



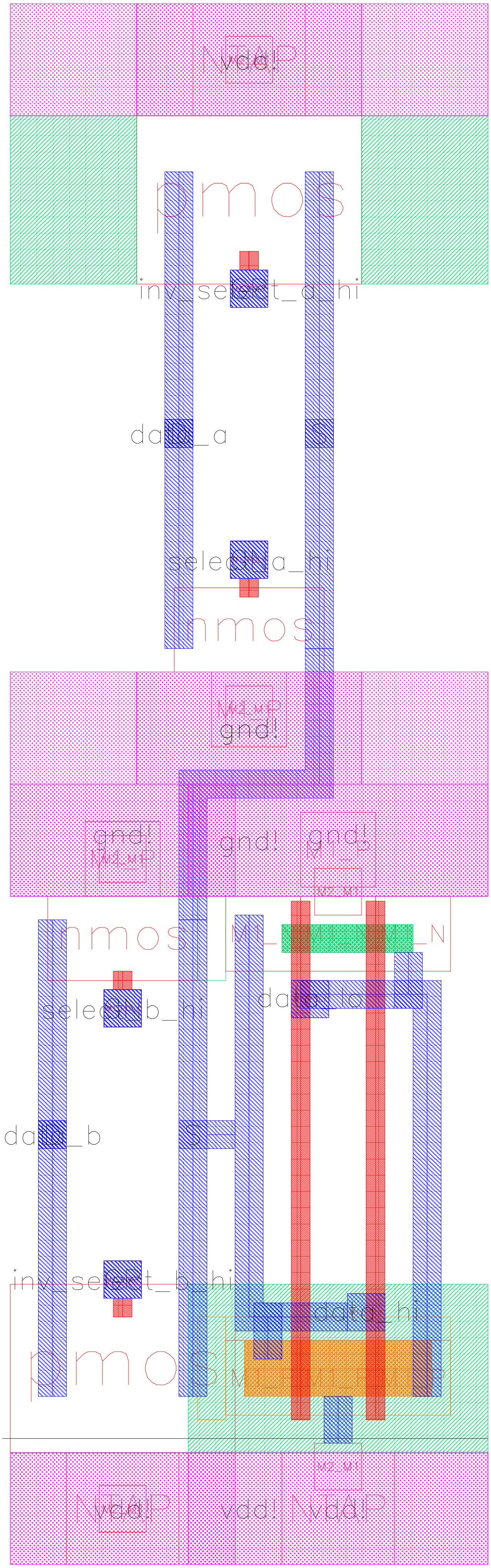
MUX4-1 Cell Layout (Level 1 Hierarchy)



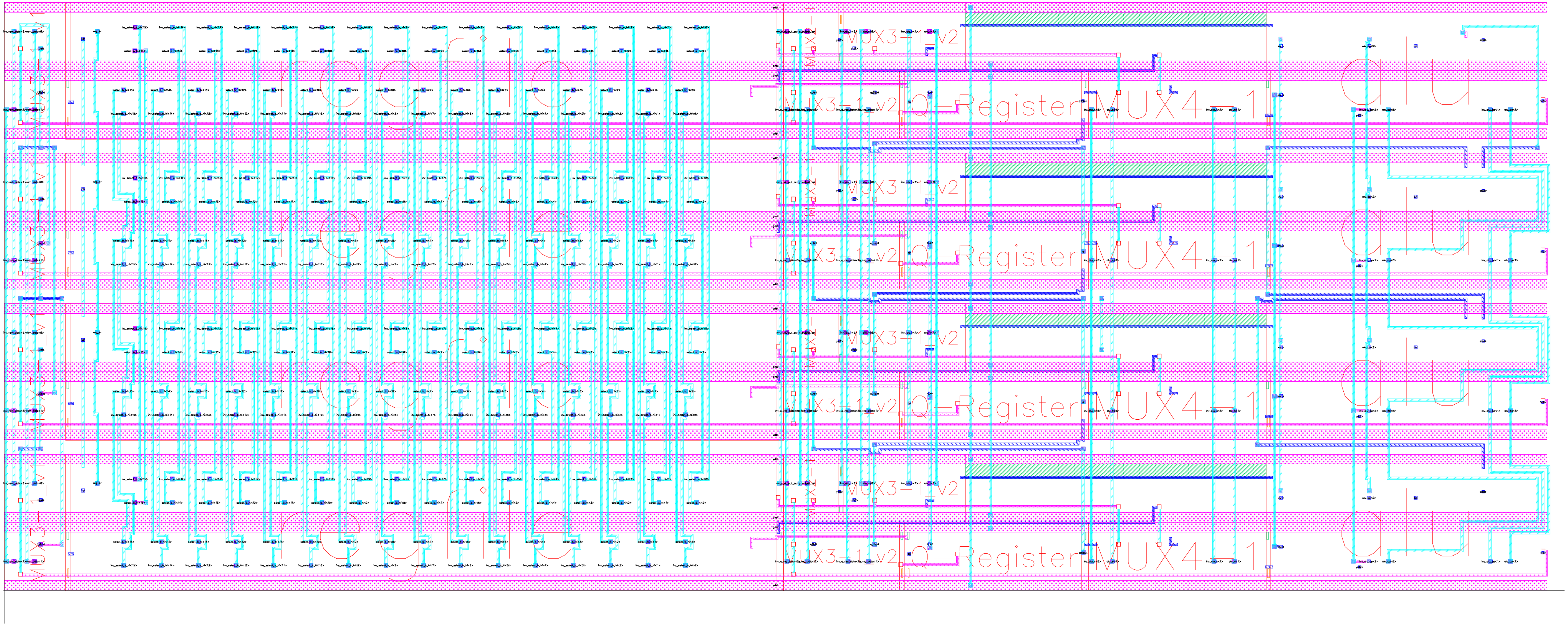
Q-Register Cell Layout (Level 1 Hierarchy)



regbit Cell Layout (Level 1 Hierarchy)



datapath Cell Layout (Level 1 Hierarchy)



```
@(#)$CDS: LVS version 6.1.8-64b 10/08/2019 19:38 (ip-172-18-22-52) $

Command line: /software/cadence-Feb2019/IC618/tools.lnx86/dfII/bin/64bit/LVS -dir /home
/jrarndt2/ece425.work/LVS -l -s -t /home/jrarndt2/ece425.work/LVS/layout /home/jrarndt2
/ece425.work/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.

    Net-list summary for /home/jrarndt2/ece425.work/LVS/layout/netlist
       count
        787             nets
        121             terminals
        828             pmos
        828             nmos

    Net-list summary for /home/jrarndt2/ece425.work/LVS/schematic/netlist
       count
        787             nets
        121             terminals
        828             pmos
        828             nmos


    Terminal correspondence points
    N730      N24       alu_op<0>
    N717      N41       alu_op<1>
    N701      N40       alu_op<2>
    N768      N63       alu_r<0>
    N755      N39       alu_r<1>
    N712      N55       alu_s<0>
    N695      N85       alu_s<1>
    N710      N45       c<0>
    N694      N6        c<1>
    N685      N25       c<2>
    N672      N20       c<3>
    N771      N46       cin
    N738      N31       cp
    N783      N101      d<0>
    N766      N5        d<1>
    N751      N75       d<2>
    N737      N48       d<3>
    N773      N11       f0_in
    N731      N33       f3_in
    N673      N76       f<0>
    N784      N110      f<1>
    N767      N57       f<2>
    N752      N56       f<3>
    N678      N0        gnd!
    N674      N12       inv_alu_op<0>
    N785      N13       inv_alu_op<1>
    N775      N78       inv_alu_r<0>
    N759      N37       inv_alu_r<1>
    N719      N7        inv_alu_s<0>
    N704      N8        inv_alu_s<1>
    N725      N96       inv_q_reg_data<0>
    N711      N91       inv_q_reg_data<1>
    N782      N38       inv_r
    N761      N59       inv_ram_data<0>
    N746      N83       inv_ram_data<1>
    N778      N22       inv_s
    N684      N58       inv_select_a_hi<0>
    N702      N64       inv_select_a_hi<10>
    N686      N28       inv_select_a_hi<11>
    N676      N90       inv_select_a_hi<12>
    N786      N14       inv_select_a_hi<13>
    N769      N109      inv_select_a_hi<14>
    N756      N108      inv_select_a_hi<15>
    N671      N53       inv_select_a_hi<1>
    N780      N65       inv_select_a_hi<2>
    N764      N61       inv_select_a_hi<3>
    N749      N60       inv_select_a_hi<4>
    N733      N62       inv_select_a_hi<5>
    N720      N66       inv_select_a_hi<6>
    N705      N87       inv_select_a_hi<7>
    N689      N73       inv_select_a_hi<8>
    N679      N92       inv_select_a_hi<9>
    N669      N19       inv_select_b_hi<0>
    N754      N106      inv_select_b_hi<10>
    N739      N86       inv_select_b_hi<11>
    N724      N50       inv_select_b_hi<12>
    N709      N29       inv_select_b_hi<13>
    N692      N2        inv_select_b_hi<14>
    N682      N27       inv_select_b_hi<15>
    N777      N102      inv_select_b_hi<1>
    N762      N100      inv_select_b_hi<2>
    N747      N9        inv_select_b_hi<3>
    N732      N95       inv_select_b_hi<4>
    N718      N99       inv_select_b_hi<5>
    N703      N105      inv_select_b_hi<6>
    N687      N89       inv_select_b_hi<7>
    N677      N93       inv_select_b_hi<8>
    N787      N104      inv_select_b_hi<9>
    N753      N36       inv_y_output_sel
    N741      N82       p<0>
    N726      N4        p<1>
    N713      N71       p<2>
    N696      N54       p<3>
    N740      N30       q0_in
    N707      N74       q0_out
    N697      N21       q3_in
    N675      N26       q3_out
    N700      N23       q_en
    N781      N16       q_reg_data<0>
    N765      N34       q_reg_data<1>
    N735      N43       ram_data<0>
    N722      N44       ram_data<1>
    N772      N35       reg_wr
    N750      N122      select_a_hi<0>
    N770      N112      select_a_hi<10>
    N757      N111      select_a_hi<11>
    N742      N80       select_a_hi<12>
    N727      N79       select_a_hi<13>
    N714      N77       select_a_hi<14>
    N698      N51       select_a_hi<15>
    N736      N121      select_a_hi<1>
    N723      N120      select_a_hi<2>
    N708      N119      select_a_hi<3>
    N691      N118      select_a_hi<4>
    N681      N117      select_a_hi<5>
    N668      N116      select_a_hi<6>
    N776      N115      select_a_hi<7>
    N760      N114      select_a_hi<8>
    N745      N113      select_a_hi<9>
    N734      N18       select_b_hi<0>
    N693      N72       select_b_hi<10>
    N683      N98       select_b_hi<11>
    N670      N97       select_b_hi<12>
    N779      N94       select_b_hi<13>
    N763      N10       select_b_hi<14>
    N748      N49       select_b_hi<15>
    N721      N52       select_b_hi<1>
    N706      N47       select_b_hi<2>
    N690      N69       select_b_hi<3>
    N680      N67       select_b_hi<4>
    N667      N70       select_b_hi<5>
    N774      N32       select_b_hi<6>
    N758      N88       select_b_hi<7>
    N744      N42       select_b_hi<8>
    N729      N81       select_b_hi<9>
    N716      N1        vdd!
    N743      N68       y<0>
    N728      N3        y<1>
    N715      N17       y<2>
    N699      N15       y<3>
    N688      N107      y_output_sel

Devices in the rules but not in the netlist:
        nfet pfet nmos4 pmos4 cap

The net-lists match.

                             layout  schematic
                                instances
        un-matched              0       0
        rewired                 0       0
        size errors             0       0
        pruned                  0       0
        active                  1656    1656
        total                   1656    1656

                                  nets
        un-matched              0       0
        merged                  0       0
        pruned                  0       0
        active                  787     787
        total                   787     787

                                terminals
        un-matched              0       0
        matched but
        different type          0       0
        total                   121     121


Probe files from /home/jrarndt2/ece425.work/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:
```

```
audit.out:


Probe files from /home/jrarndt2/ece425.work/LVS/layout

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:
```

9. So far, you have finished the hand-made part of you AM2901 microprocessor. Are there any difficulties you have encountered in your design and layout? List three most important ones and simply describe what they are and how you solved them.

Cell Standardization and Cell Placement/Floorplaning

To preserve the modularity of the design, we took the approach of making the "rail-to-rail" distance approximately ~10um (i.e. cell height) across all cells. Note that in our layouts for the MUX3-1_v1, regfile, and ALU cells, we implemented a "double row" design, in which cell instances are "stacked" and share a common ground rail. This deliberate choice was made for these cells specifically to alleviate routing difficulties; however, while still maintaining a standardized cell shape and size. This choice proved to be critical when routing and placing cells in the upper levels of the hierarchy of the overall design. Nonetheless, we still did run into difficulties during floorplaning and cell placement, specifically—bitslice. From the datapath cell layout screenshot and printout above, there is a noticeable "gap" of unused area that exists in each bitslice instance. This was obviously undesirable, yet seemingly unavoidable. We tried many different configurations in the bitslice layout, yet the current configuration proved to be most optimal with respect to routing considerations. For future designs, it would be better to incorporate this unused area into the design as to increase density, while reducing area.

Signal Routing

As in any great design process, signal routing is one of if not the most important considerations during layout. Being limited to only 3 metal layers on such a large design became increasingly difficult as we moved up the hierarchy. Luckily, the modularity of our design allowed for much easier routing on datapath. Further, we became more conscious to the design rules, which helped in quickly identifying valid and invalid signal routes and connections. Though, there are other aspects of routing that we must consider. For example, the output of the ALU, f, needed to be routed to two 3-1 MUXes and one 2-1 MUX in bitslice. To do so, we routed this signal along a long wire, spanning the entire width of the cell. In practice, this is not a good design choice as the long metal wire could introduce large amounts of delay due to interconnect resistance and capacitance, which could have a substantial impact of system functionality and performance.

Area Constraints and Considerations

Given that the upper limit constraint of this design is 20K units, we had to restart our design approach multiple times. At first, each cell was laid out naively, in which area optimization was not considered. After discussing with other peers and Yizhen at his office hours, we began to implement some design optimization techniques. For example, each instance in each cell was placed as "near as possible" to each other in the horizontal direction, without violating DRC. This proved to recover a large amount of area that would have otherwise been used before as the result of poor layout. Further, as touched on before, we sacrificed a significant amount of recoverable area in each of our bitslices. Clearly, area is a function of floorplanning, cell placement, and signal routing and as a result, we could have further improved our area if these design aspects were considered more carefully.

10. Briefly answer the following questions:

a. What is the distance between *vdd* and *gnd* in your layout? Why do you set this value?

As mentioned in the previous response, we standardized each cell height to be 10um between *vdd* and *gnd* in order to preserve the modularity of the design. This value was primarily chosen arbitrarily—with justification. We found that 10um was an adequate cell height such that each of the cell instances can more easily be connected (i.e. sharing/connecting power and ground rails) and maintain a "rectangular" cell shape, while leaving ample space between the n- and p-diffusions (in the vertical direction) for routing considerations.

b. Some gates are sized to be larger than the minimum size (e.g., I55 and I56 in the *regfile* or I0 and I1 in the *logic*). Take one example to explain why it has to be set larger than the minimum size? (Hint: different examples may have different reasons.)

For example, the two inverter gates in series (that buffers the input data_b), I55 and I66, in *regfile* are sized to be larger than minimum size as to minimize the delay from the input of the buffer to the input of the transmission gate.