CMPE 315: Principles of VLSI Design
Project Cover Page

Project Title : **Cache Design: Final Submission**

Name : **Brendan Cain & Scott Boyd**
Section : **Friday @ 1:00pm**

Date Submitted : **12/8/2020**

TA / Grader Use Only:

Late Submission Deduction (20% per day late):

Other Deductions:

Final Lab Grade:

Comments to student:

Design Description:

Cache Design:

The Cache is made up of x8 cache blocks. Each cache block is made up of x4 bytes of cache. The cache block relies on the statemachine and state register to send the correct address, data, and signals at the correct time. The inputs and outputs to the cache memory are shown below

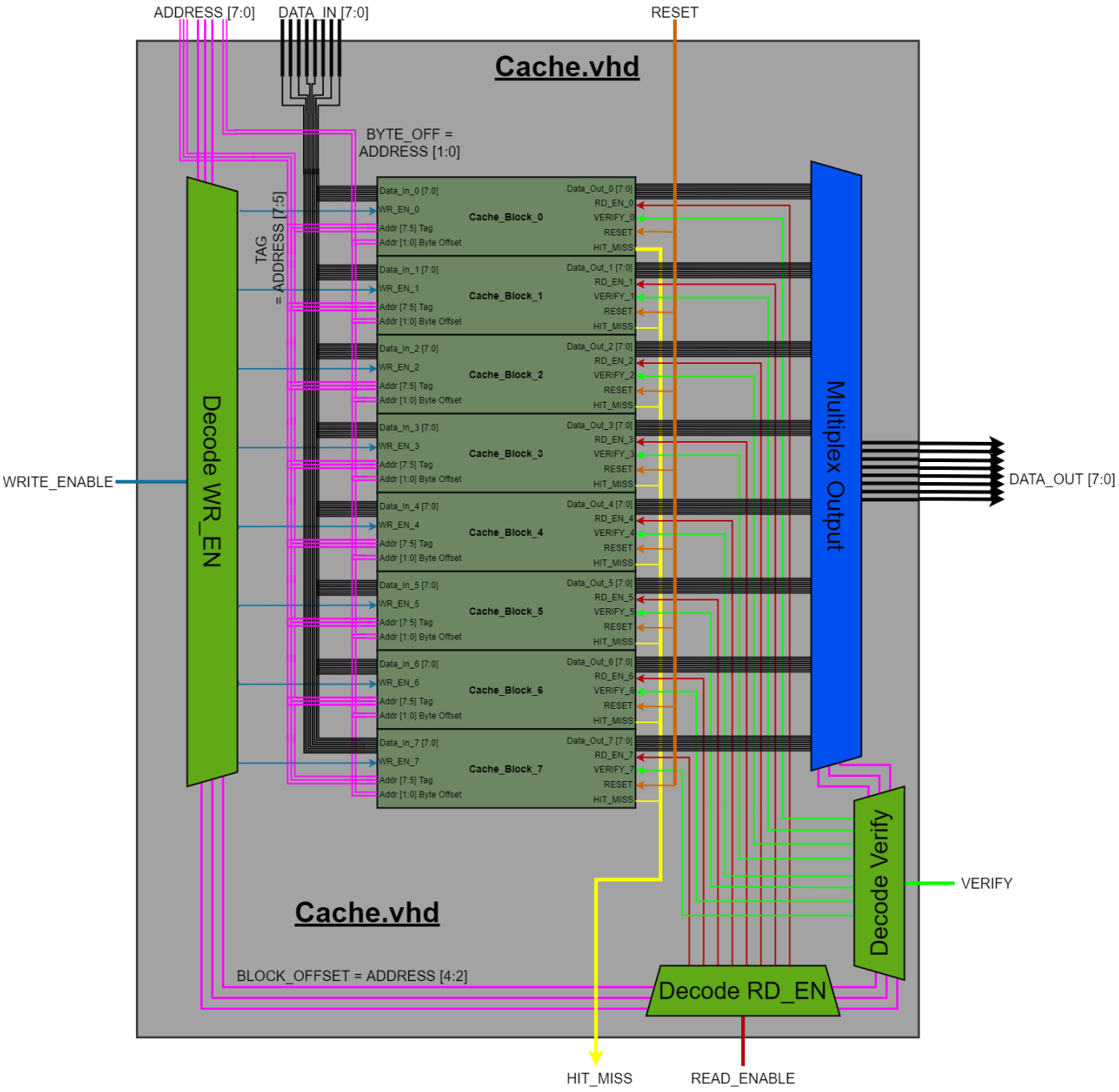
Inputs: Data_in[7:0], Addr[7:0], read_enable, write_enable, verify, reset
Outputs Data_Out[7:0], hit_miss

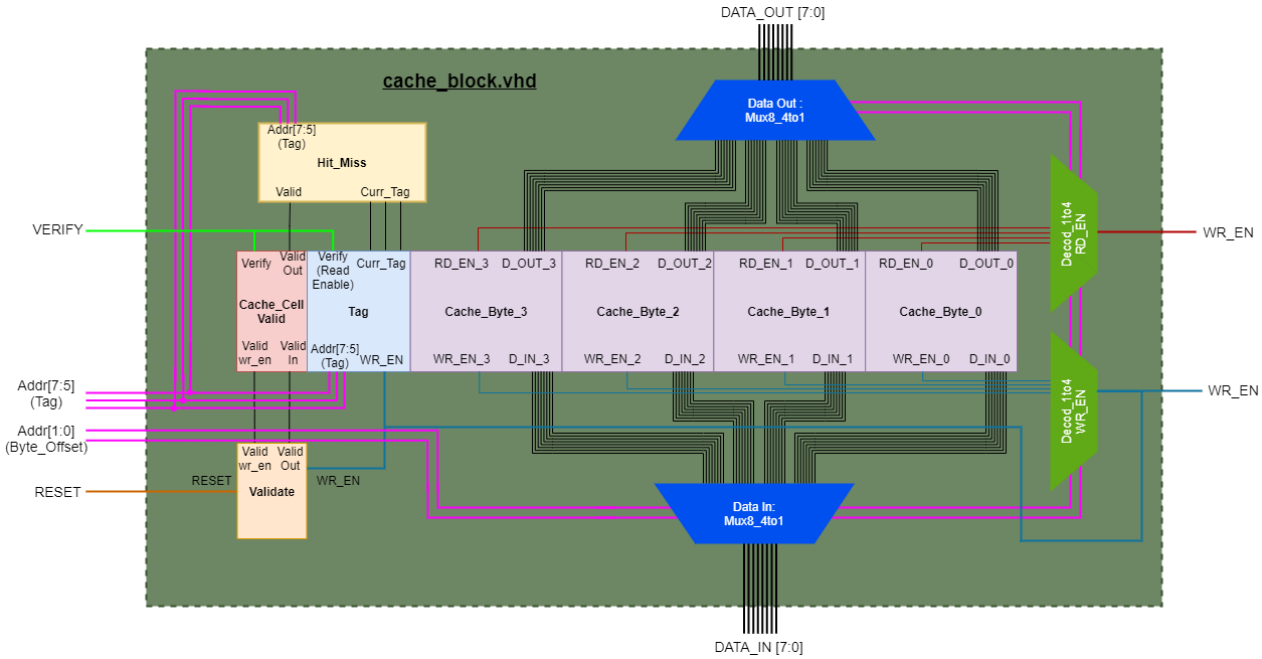
The cache memory design is fully able to verify that any of the 8 blocks in the cache is able to be written or read from provided there is a valid address in the address input. This is done via the hit or miss entity which compares the input tag and the tag that is currently saved in the block's tag along with the currently saved valid bit.

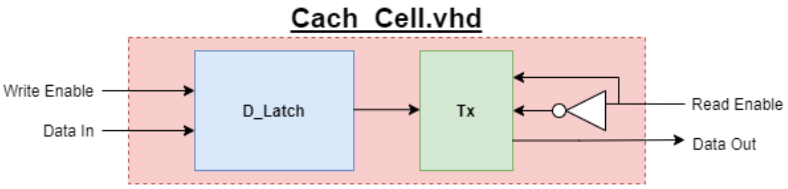
The cache memory entity assumes that if a read or write operation is occurring, then the address that it is occurring at is a hit. This is why before any operation occurs the hit or miss output from the cache entity must be latched by the state register/state machine. Since the verify input, which triggers the hit miss output, is tied to the start input coming from the cpu, it is able to return whether a hit or miss will occur before any state machine timer takes place.

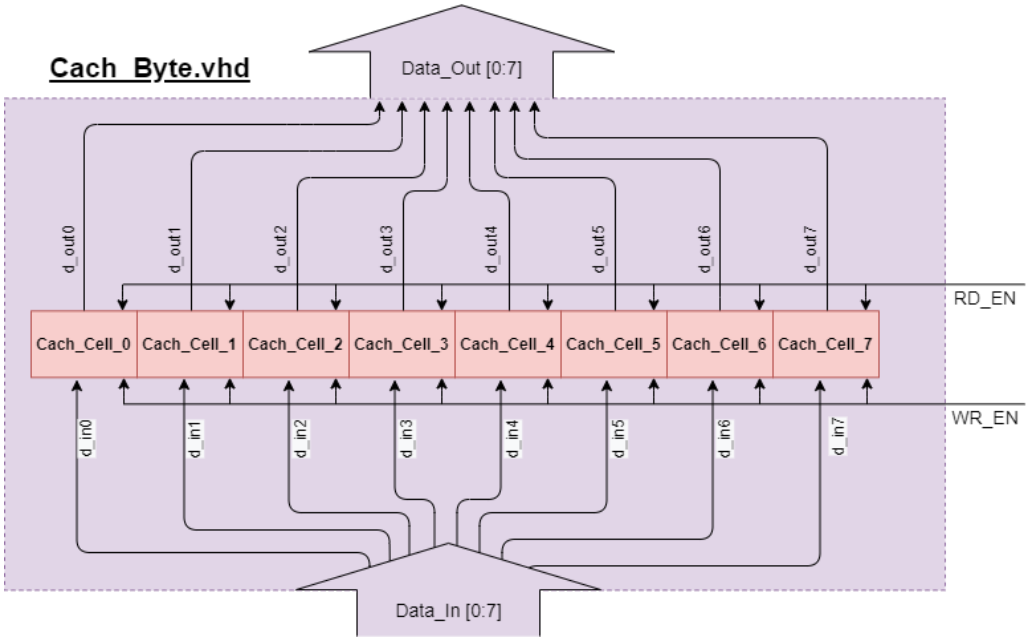
The reset input to the cache memory comes directly from the cpu and is tied to the reset input to each cache block. In the cache block entity, there is a validate entity which allows the writing to the valid bit on 2 occasions: One, a write operation occurs and thus the valid and tag are updated (valid being 1), or two, the entire cache is being reset and thus every valid bit will be set to 0.

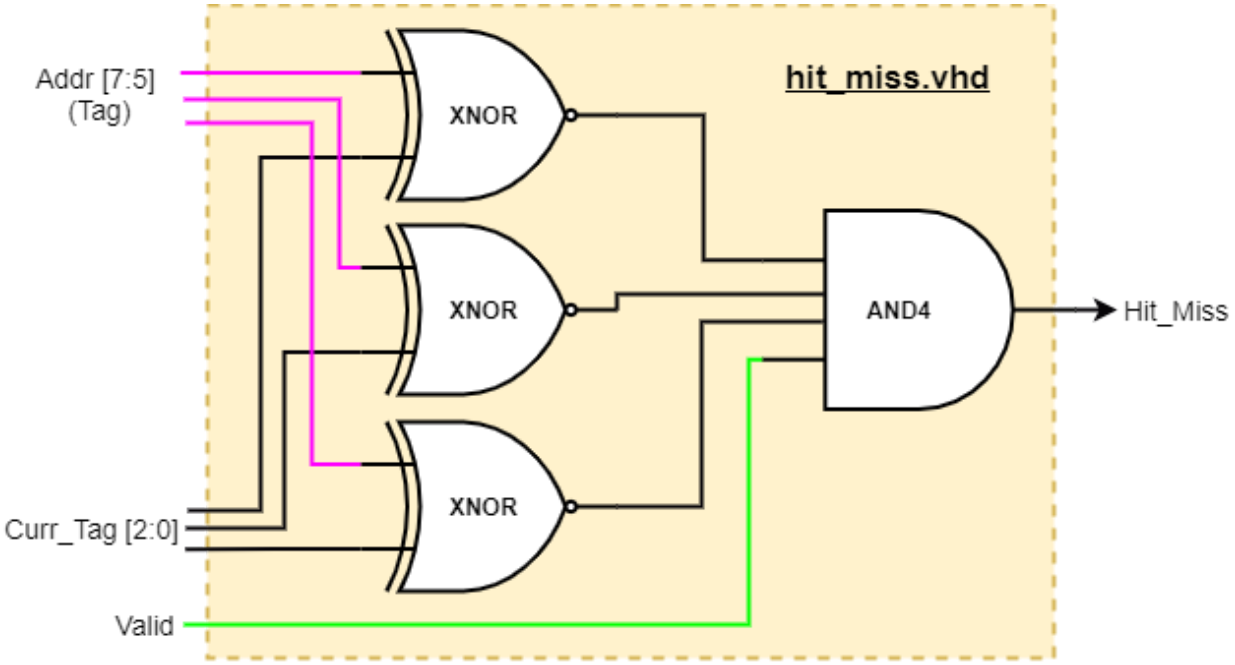
Cache Design Diagrams

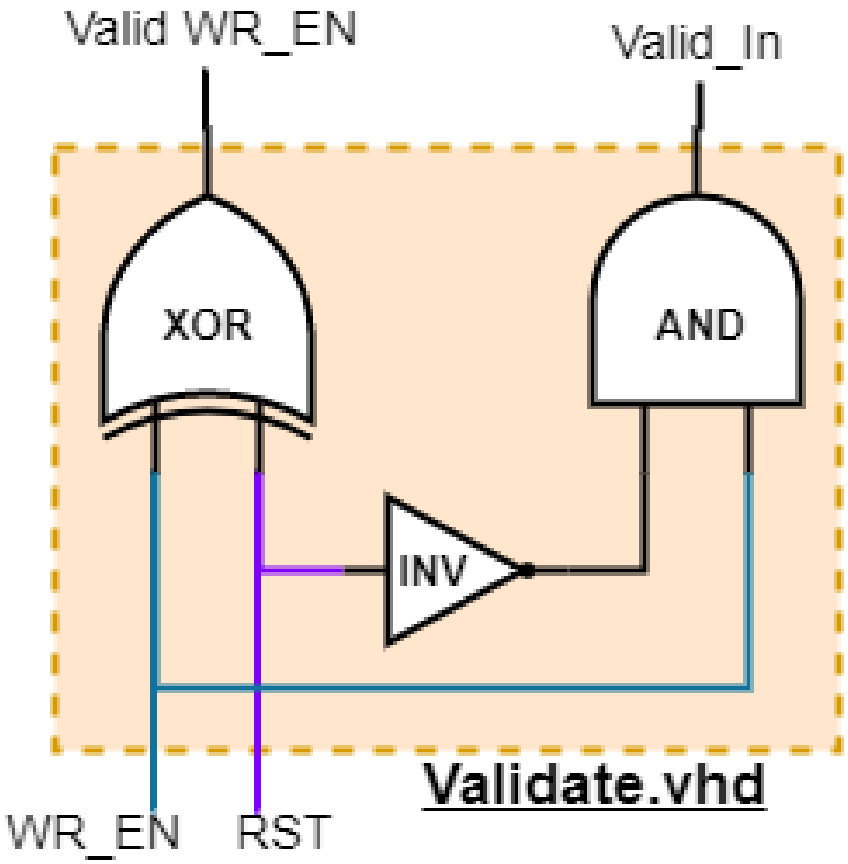



DATA_OUT [7:0]
cache_block.vhd
Data Out :
Mux8_4to1
Addr[7:5]
(Tag)
Hit_Miss
Valid
Curr_Tag
VERIFY
Verify
Valid
Out
Verify
(Read
Enable)
Curr_Tag
Cache_Cell
Valid
Tag
Valid
wr_en
Valid
In
Addr[7:5]
(Tag)
WR_EN
RD_EN_3
D_OUT_3
RD_EN_2
D_OUT_2
RD_EN_1
D_OUT_1
RD_EN_0
D_OUT_0
Cache_Byte_3
Cache_Byte_2
Cache_Byte_1
Cache_Byte_0
WR_EN_3
D_IN_3
WR_EN_2
D_IN_2
WR_EN_1
D_IN_1
WR_EN_0
D_IN_0
Decod_1to4
RD_EN
WR_EN
Decod_1to4
WR_EN
WR_EN
Addr[7:5]
(Tag)
Addr[1:0]
(Byte_Offset)
RESET
RESET
Valid
wr_en
Valid
Out
Validate
WR_EN
Data In:
Mux8_4to1
DATA_IN [7:0]

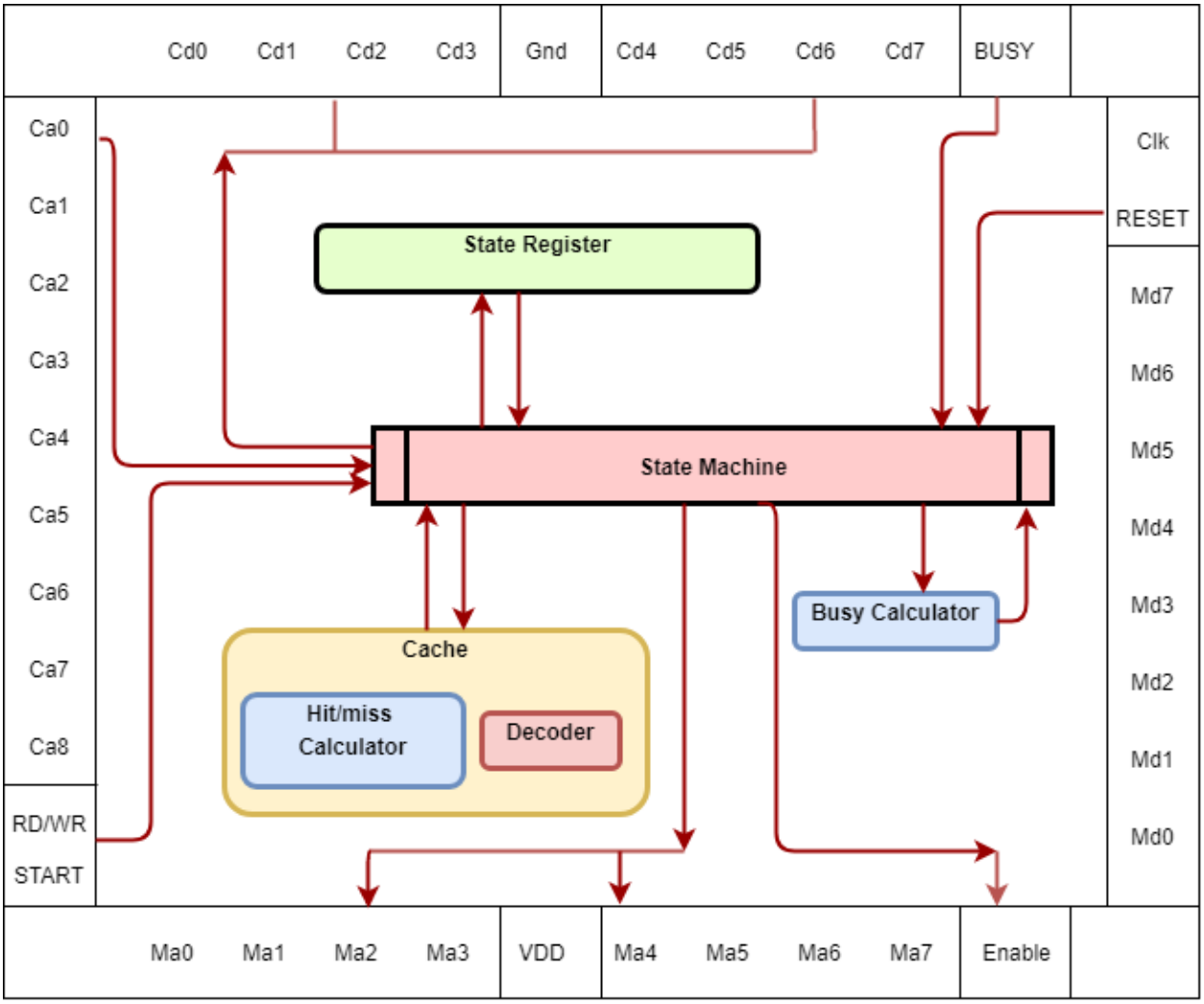

Cach_Cell.vhd
Write Enable
Data In
D_Latch
Tx
Read Enable
Data Out


Cach_Byte.vhd
Data_Out [0:7]
d_out0
d_out1
d_out2
d_out3
d_out4
d_out5
d_out6
d_out7
RD_EN
Cach_Cell_0
Cach_Cell_1
Cach_Cell_2
Cach_Cell_3
Cach_Cell_4
Cach_Cell_5
Cach_Cell_6
Cach_Cell_7
WR_EN
d_in0
d_in1
d_in2
d_in3
d_in4
d_in5
d_in6
d_in7
Data_In [0:7]


hit_miss.vhd
Addr [7:5]
(Tag)
XNOR
XNOR
XNOR
AND4
Hit_Miss
Curr_Tag [2:0]
Valid


Valid WR_EN
Valid_In
XOR
AND
INV
Validate.vhd
WR_EN
RST

Block Diagram:

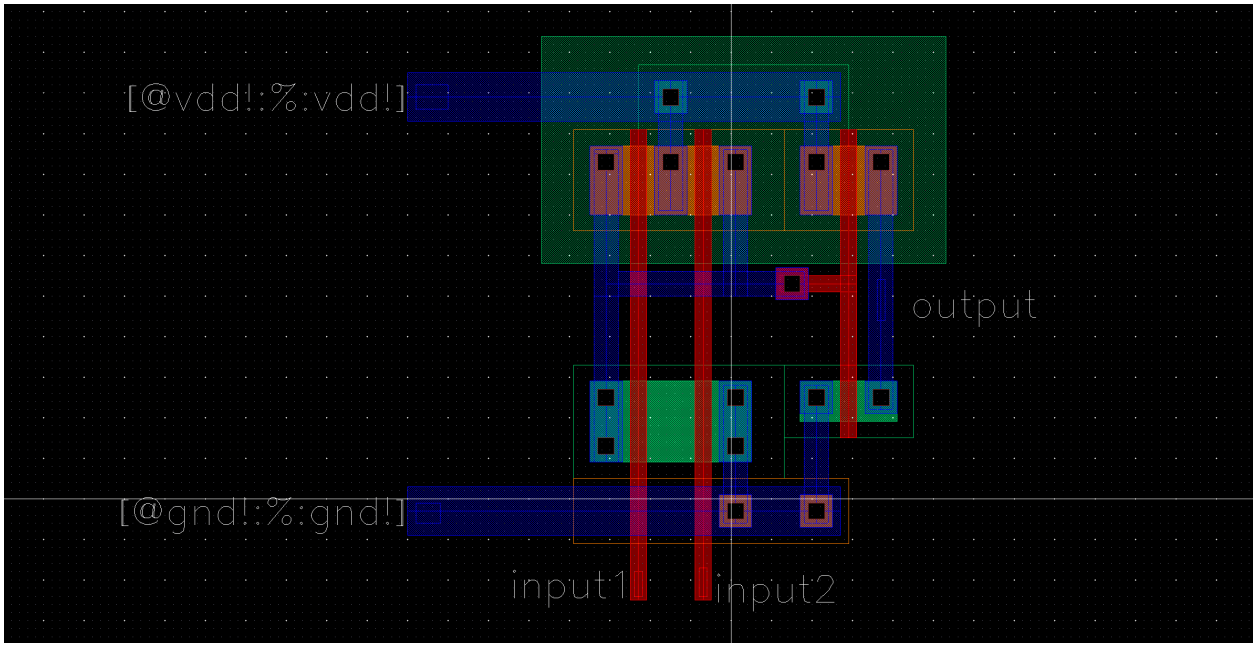


Design Hierarchy:

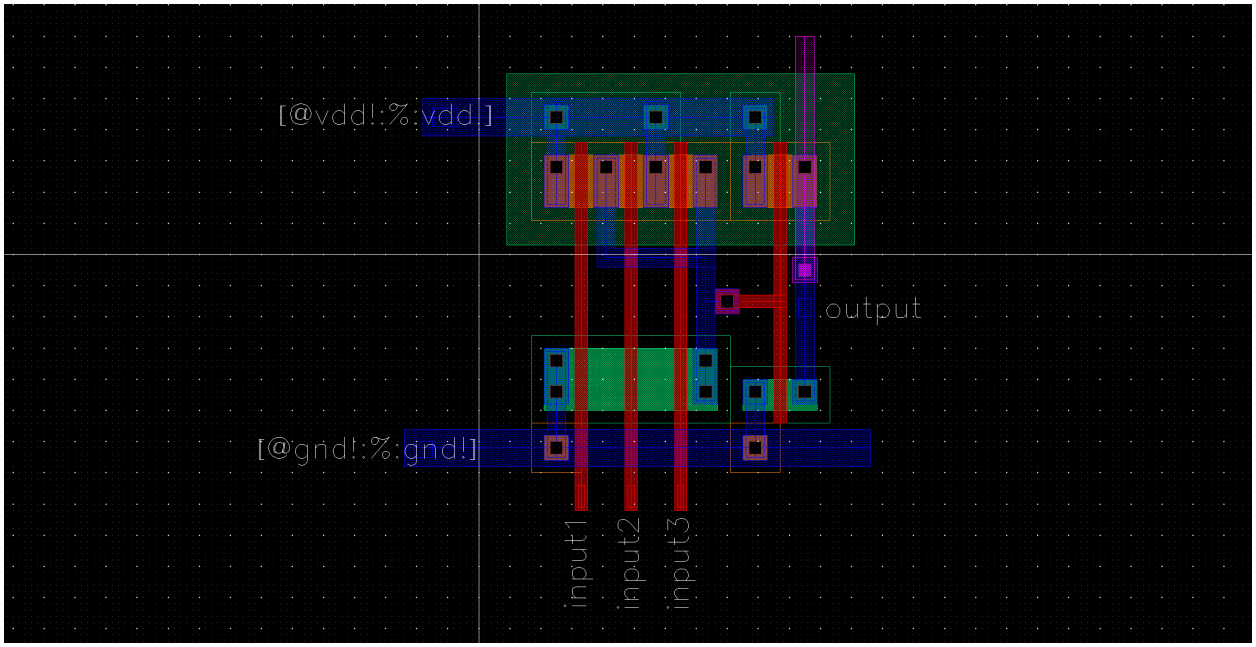
- Top.vhd
  - A. Cache.vhd
    - I. x8 cache_block.vhd
      1. x1 cache_byte.vhd
         - x4 cache_cell.vhd
           - i. D_latch.vhd
           - ii. tx.vhd
           - iii. inv.vhd
      2. x1 tag.vhd
         - x3 cache_cell.vhd
      3. x1 cache_cell.vhd (for valid)
      4. x1 hit_miss.vhd
         - x3 xnor2.vhd
         - x1 and4.vhd
      5. x1 validate.vhd
         - x1 and2.vhd
         - x1 xor.vhd
         - x1 inv.vhd
      6. x2 decod_1to4.vhd
         - x4 and3.vhd
         - x2 inv.vhd
      7. x2 mux8_4to1.vhd
         - x8 mux_4to1.vhd
    - II. x1mux8_8to1.vhd
      1. x8 mux_8to1.vhd
    - III. x3 decod_1to8.vhd
  - B. Statemachine.vhd
    - I. X2 counter.vhd
    - II. X4 and2.vhd
  - C. State_register.vhd
    - I. X5 inv.vhd
    - II. X1 and3.vhd
    - III. X4 and5.vhd
    - IV. X1 or5.vhd
  - D. Busy_calc.vhd
    - I. X5 and2.vhd
    - II. X7 and5.vhd
    - III. X3 or2.vhd
    - IV. X2 or3.vhd

Layouts:

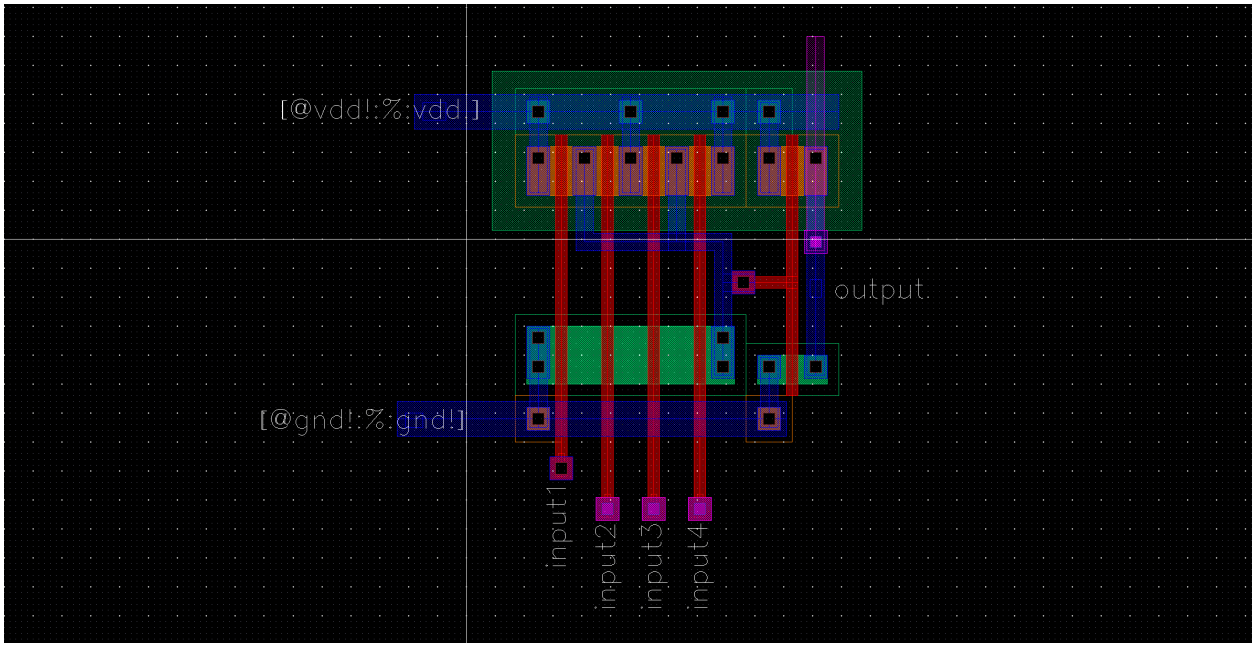
Below is a layout of the and2 schematic



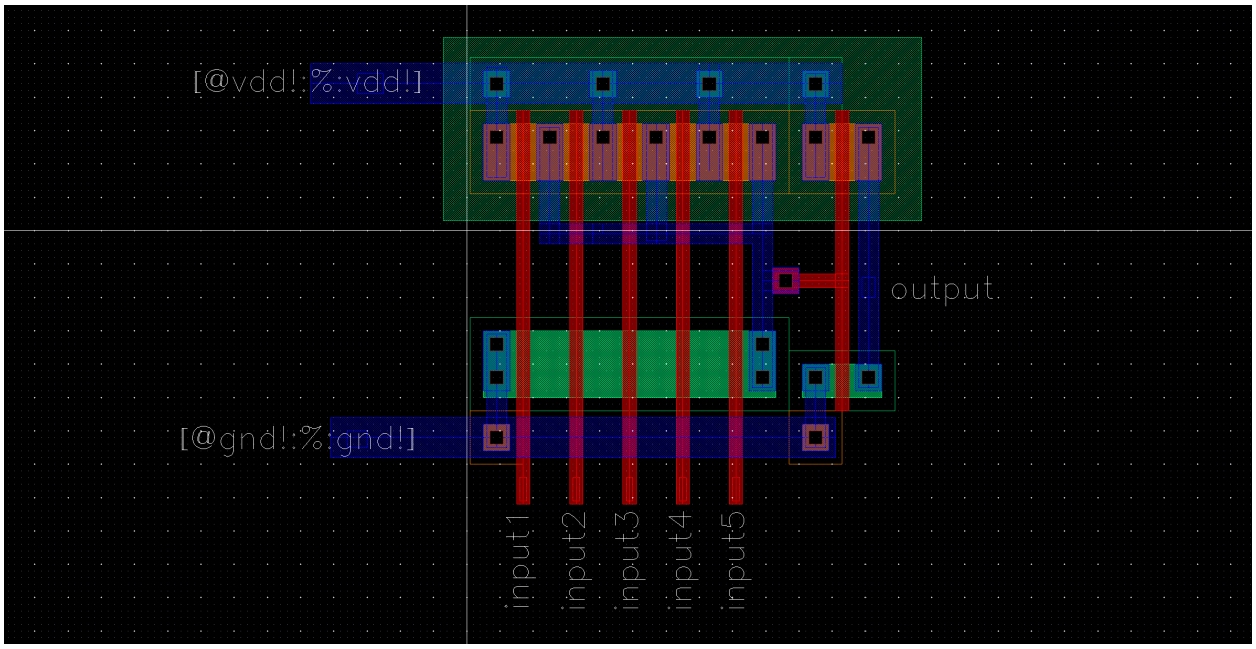
Below is a layout of the and3 schematic



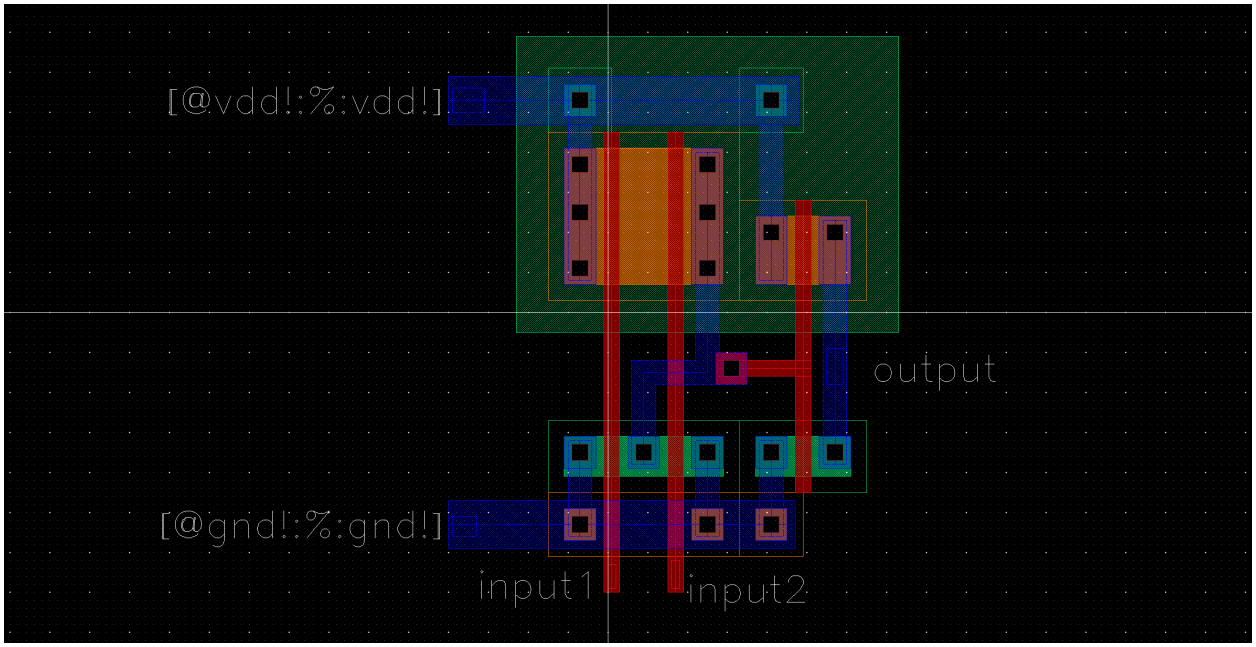
Below is a layout of the and4 schematic



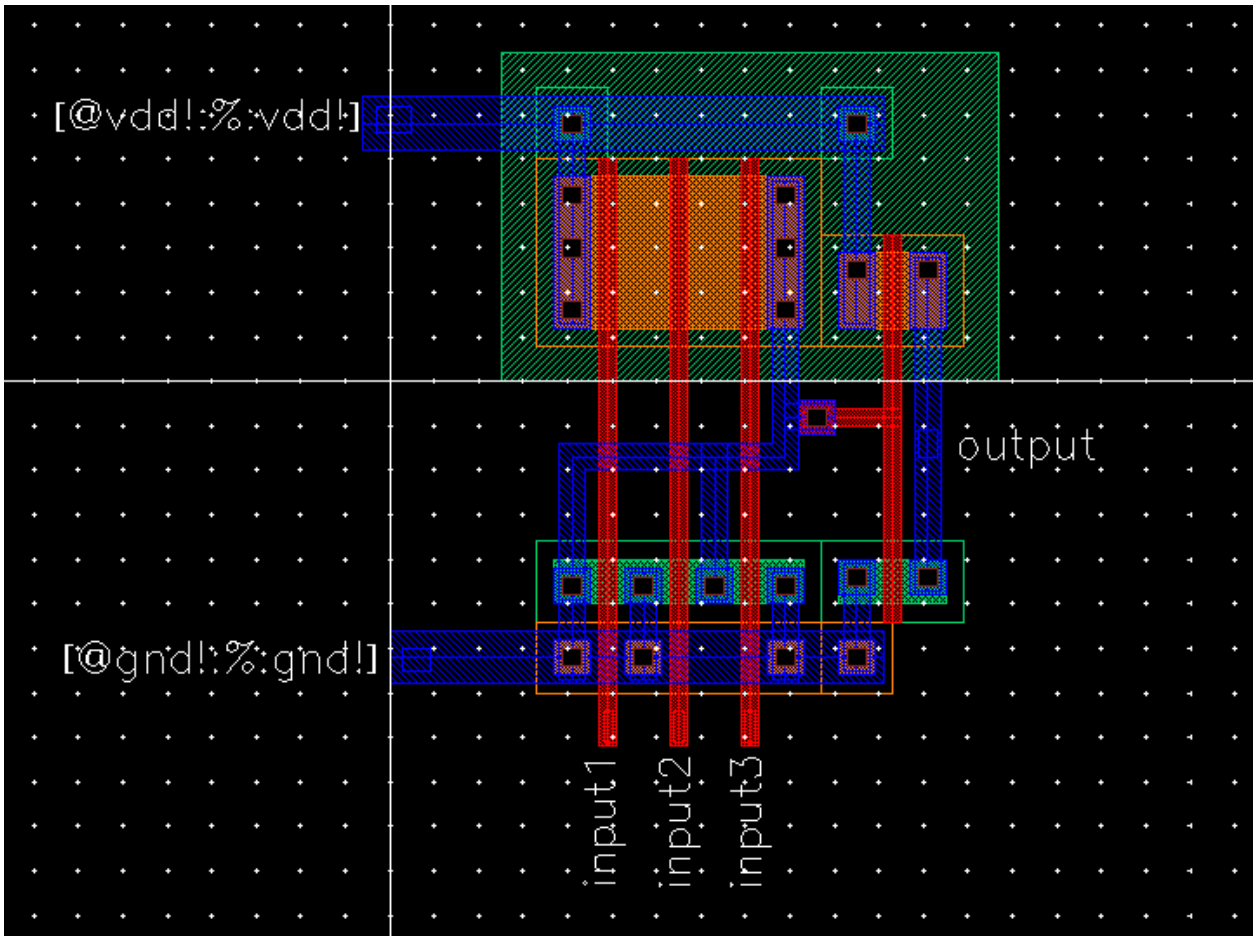
Below is a layout of the and5 schematic



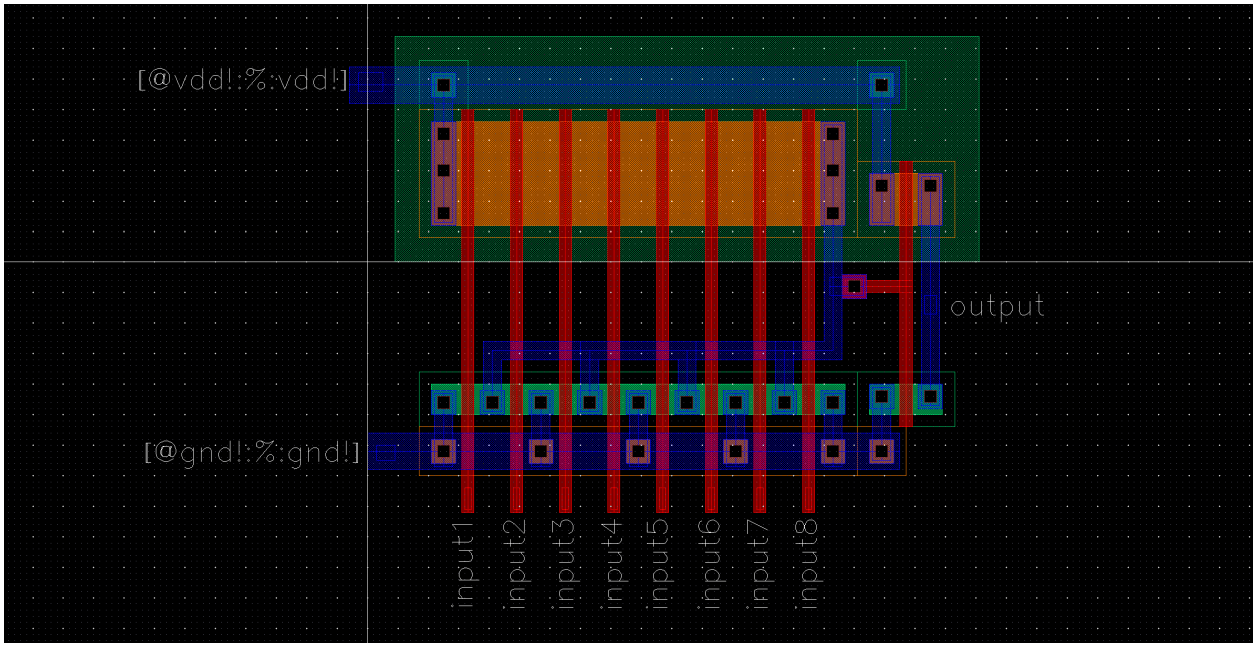
Below is a layout of the or2 schematic



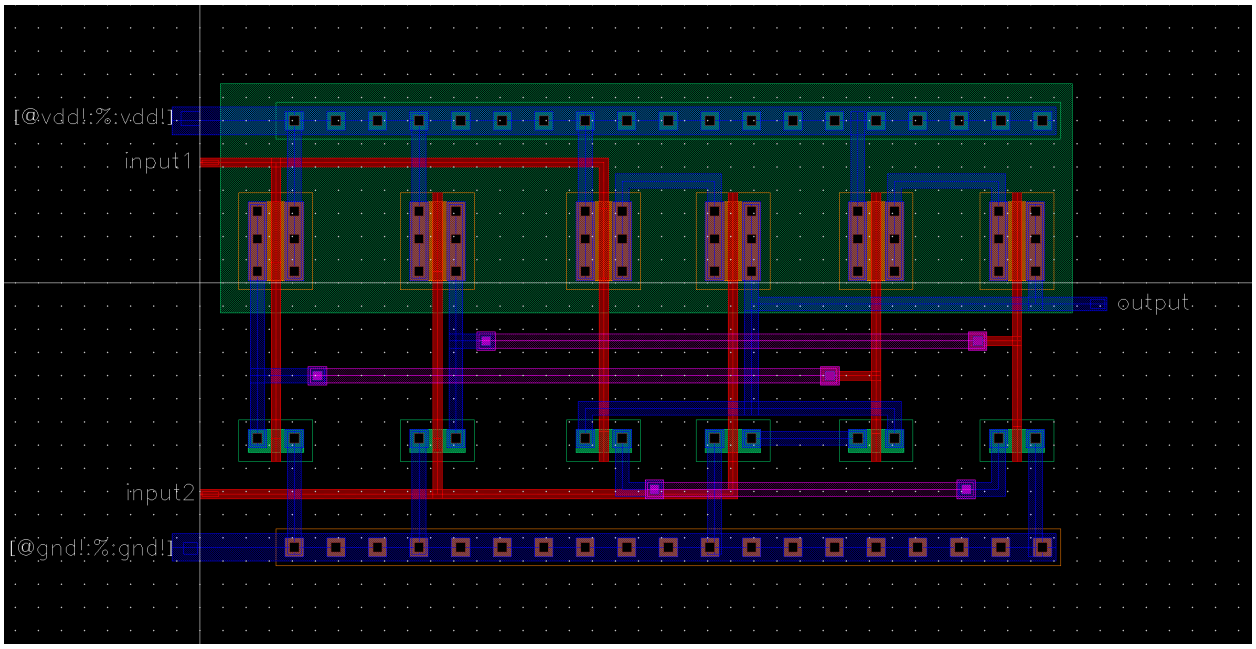
Below is a layout of the or3 schematic



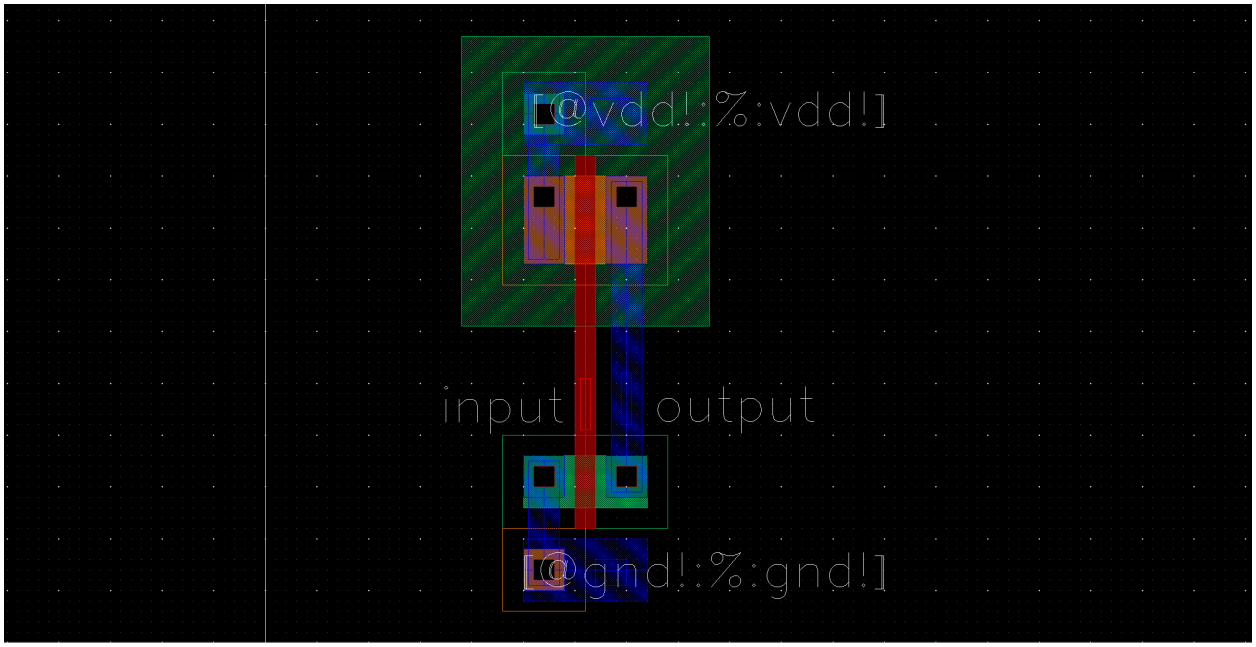
Below is a layout of the or8 schematic



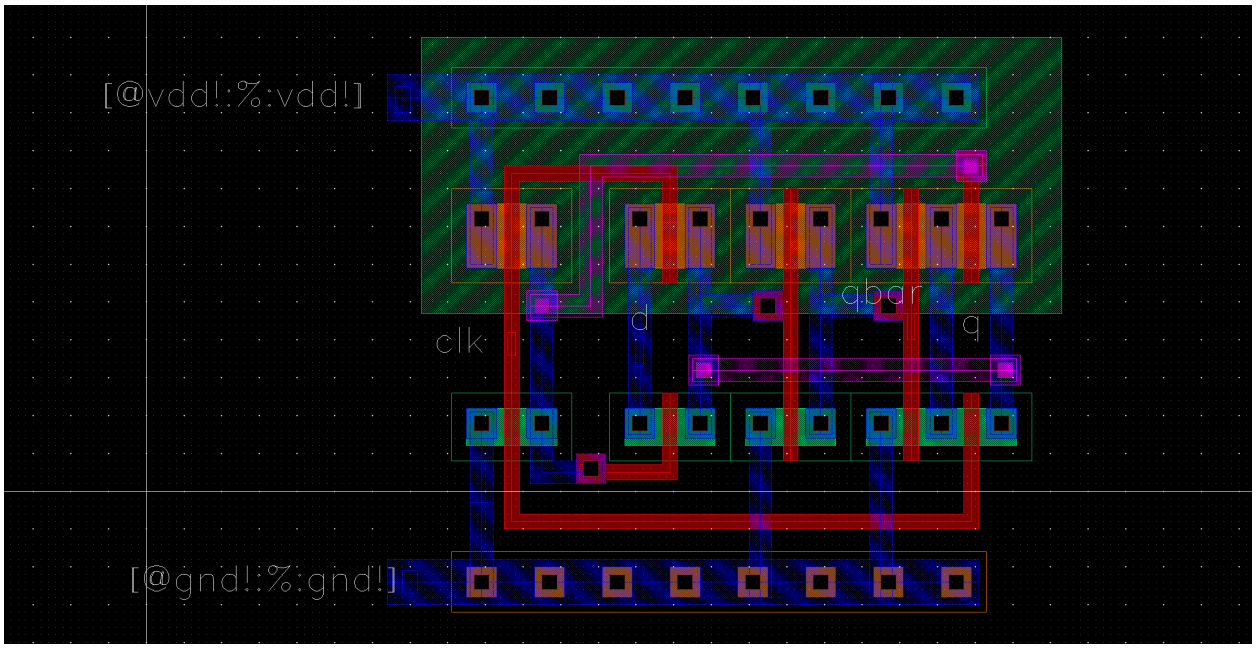
Below is a layout of the xnor2 schematic



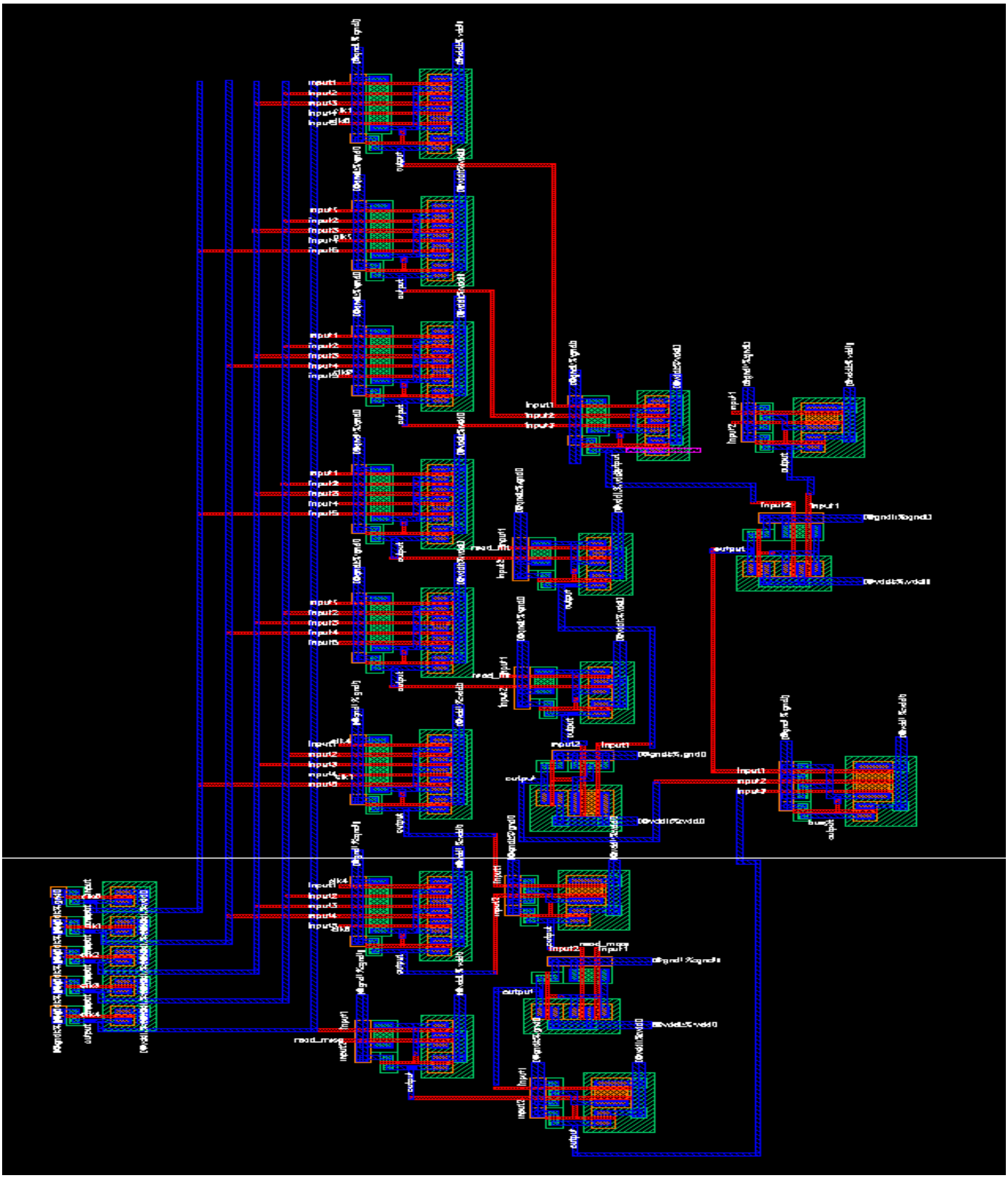
Below is a layout of the inverter schematic



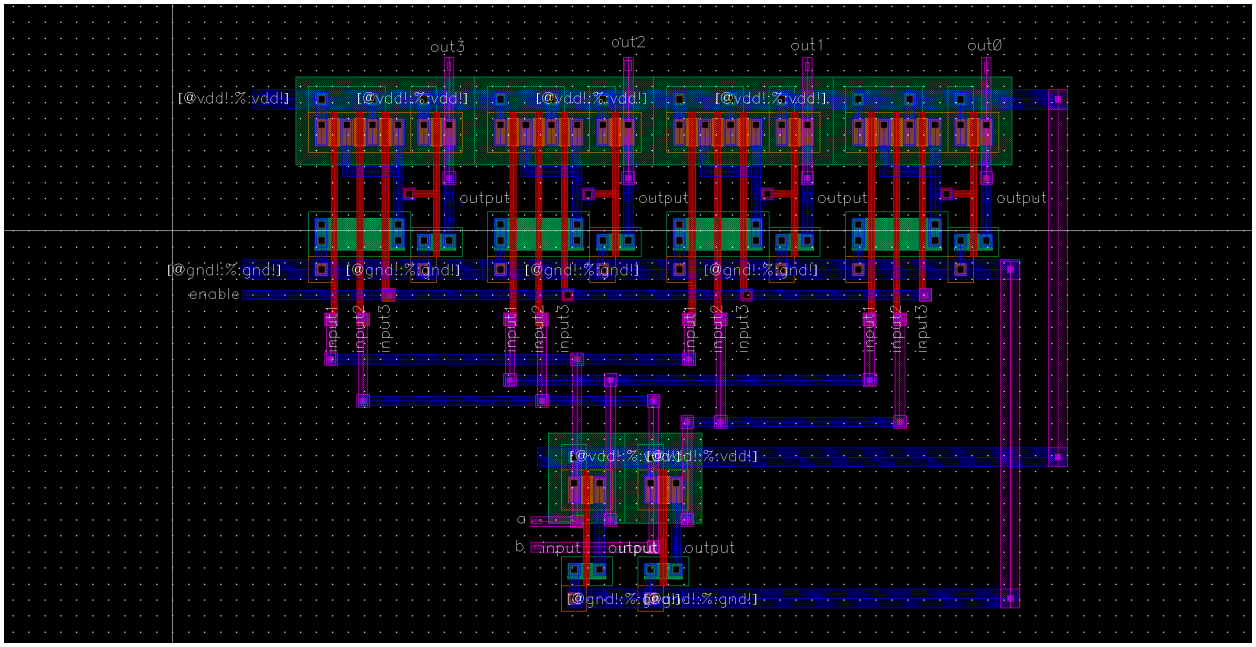
Below is a layout of the dlatch schematic



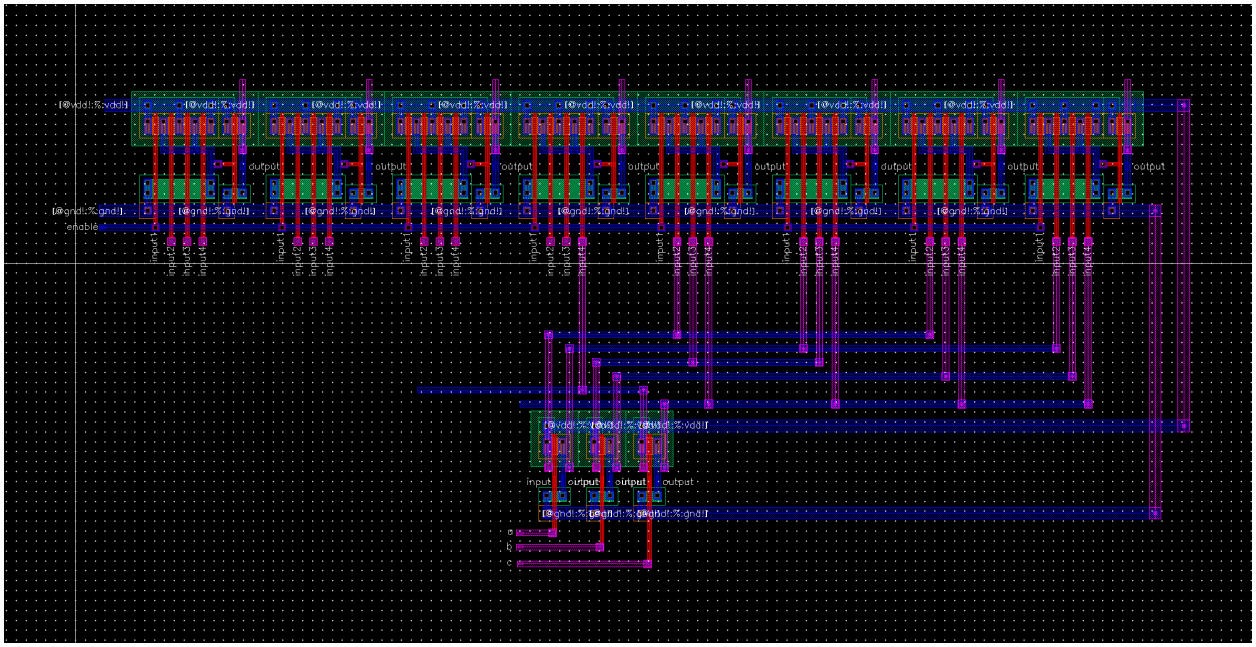
Below is a layout of the busy calculator schematic



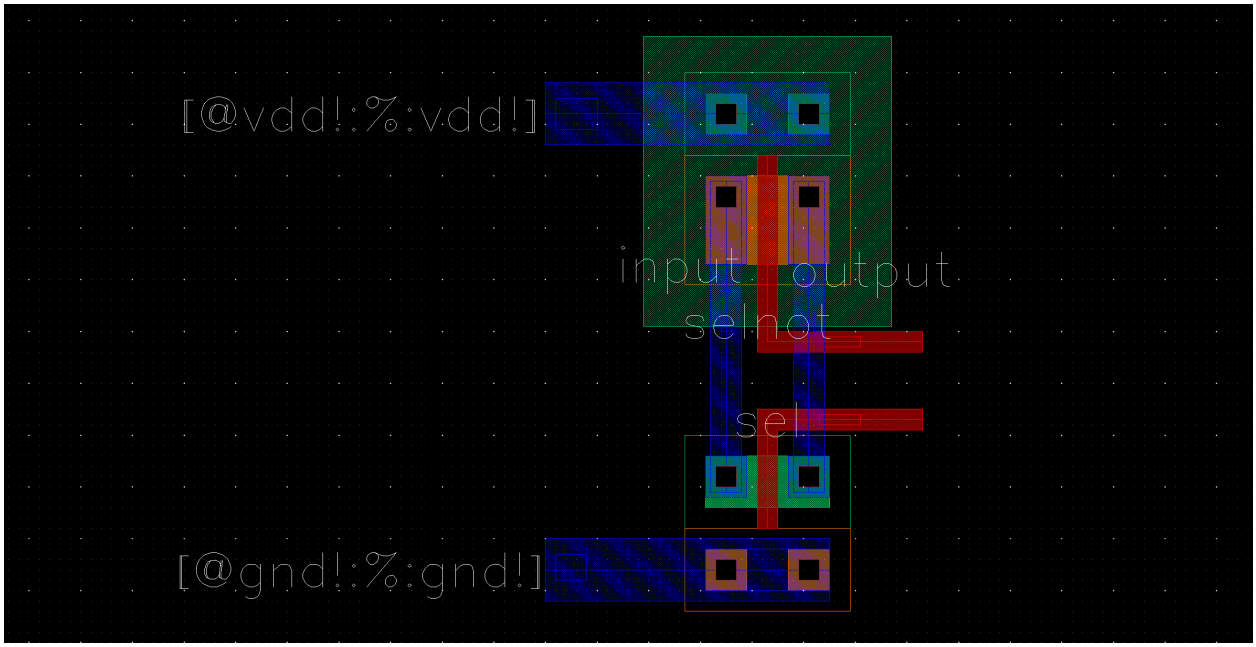
Below is a layout of the decoder 1to4 schematic



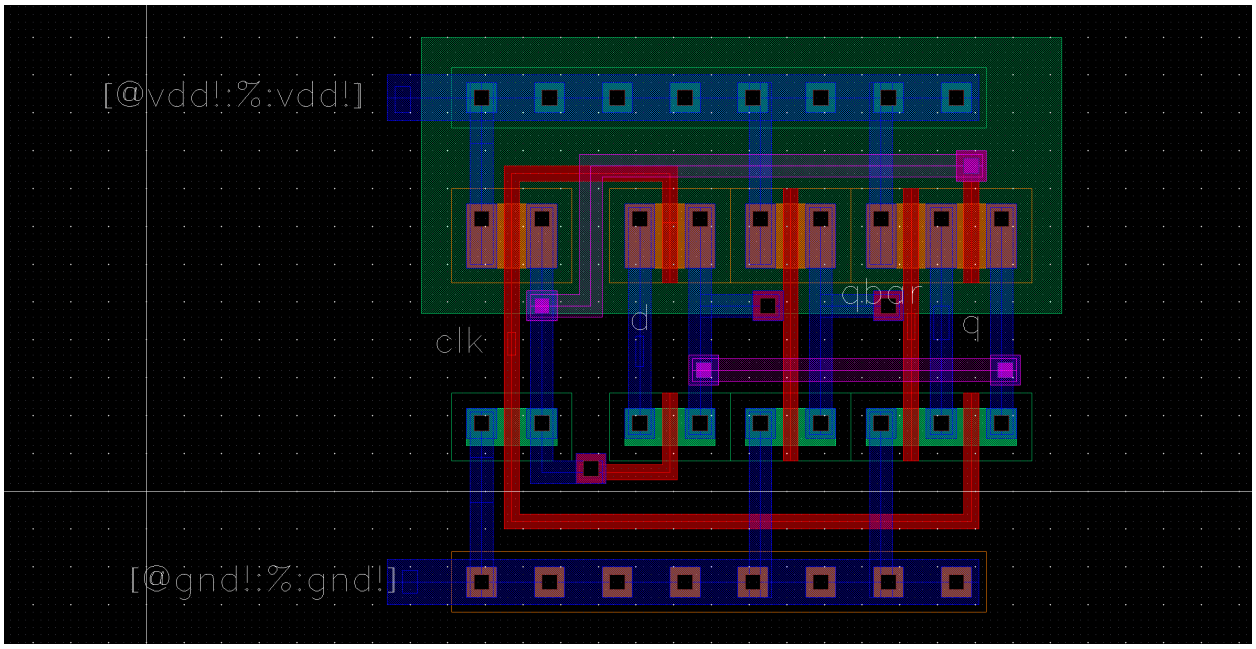
Below is a layout of the decoder 1to8 schematic



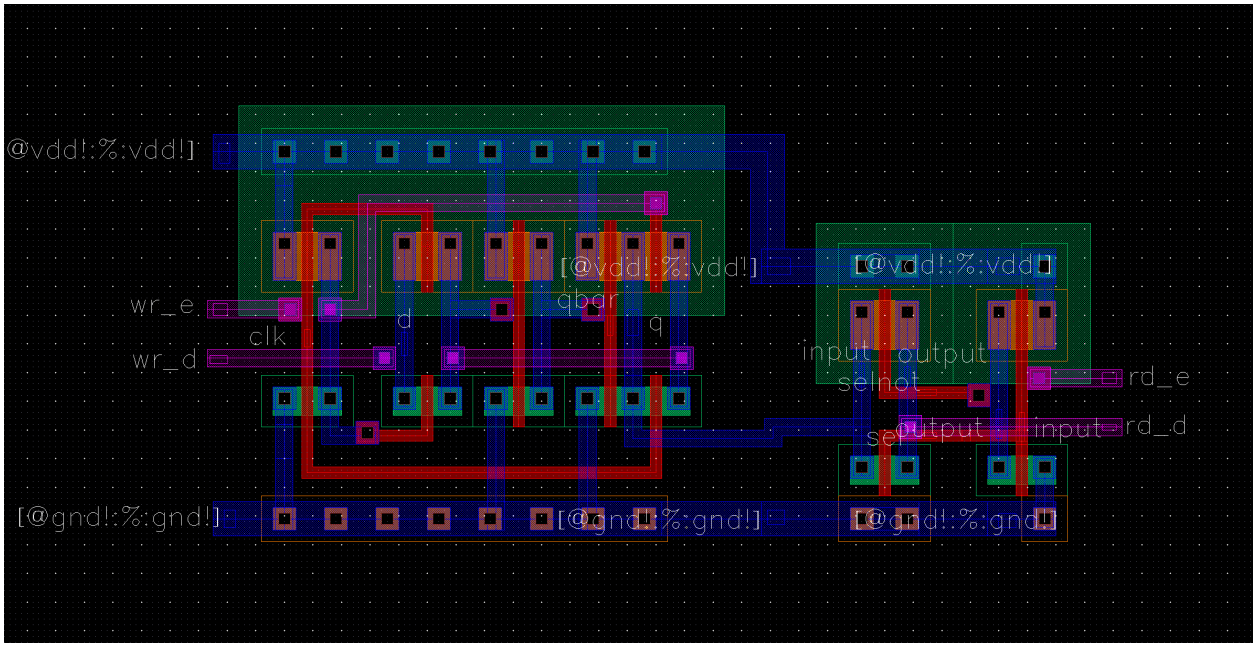
Below is a layout of the tx schematic



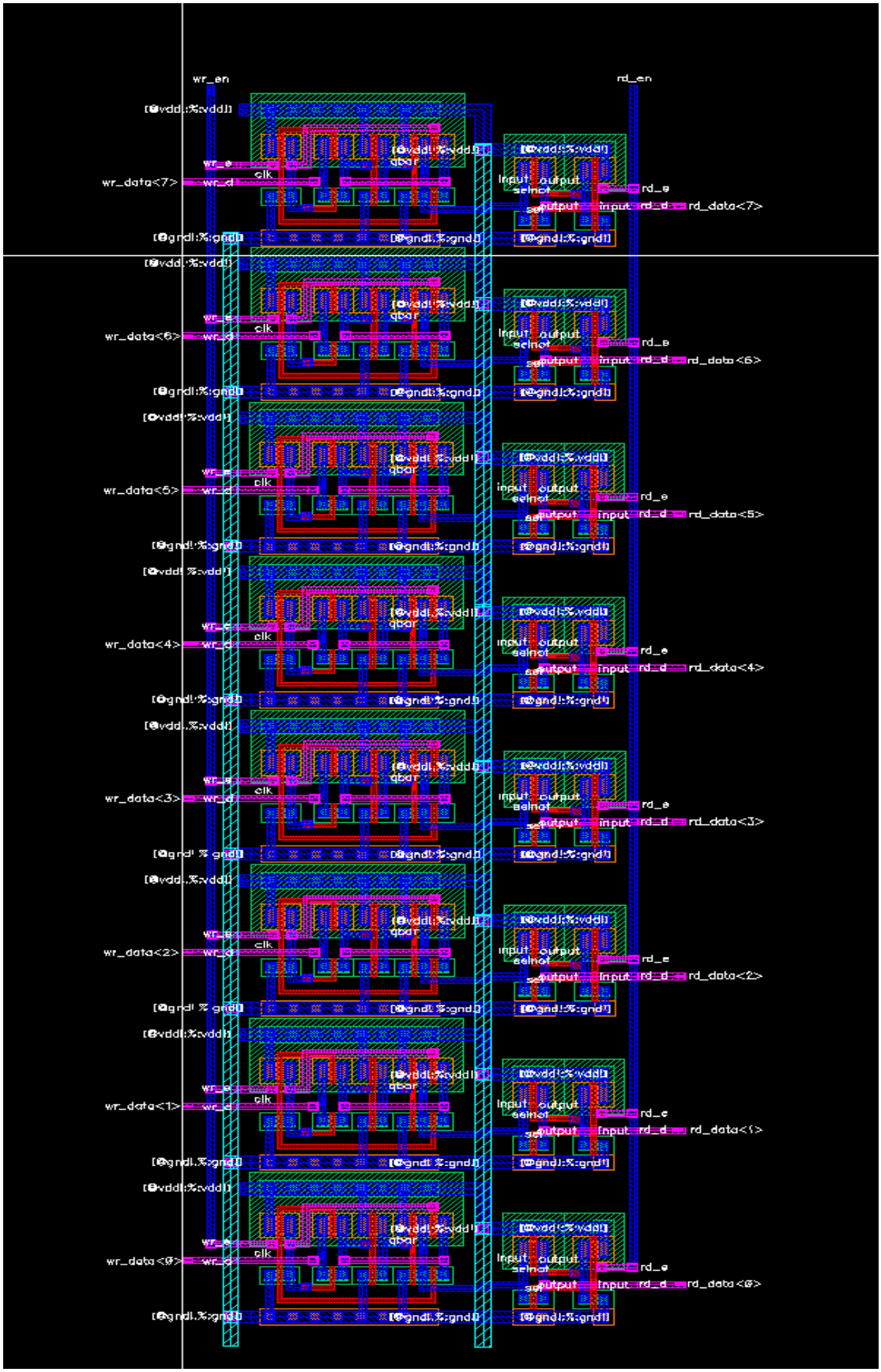
Below is a layout of the Dlatch schematic



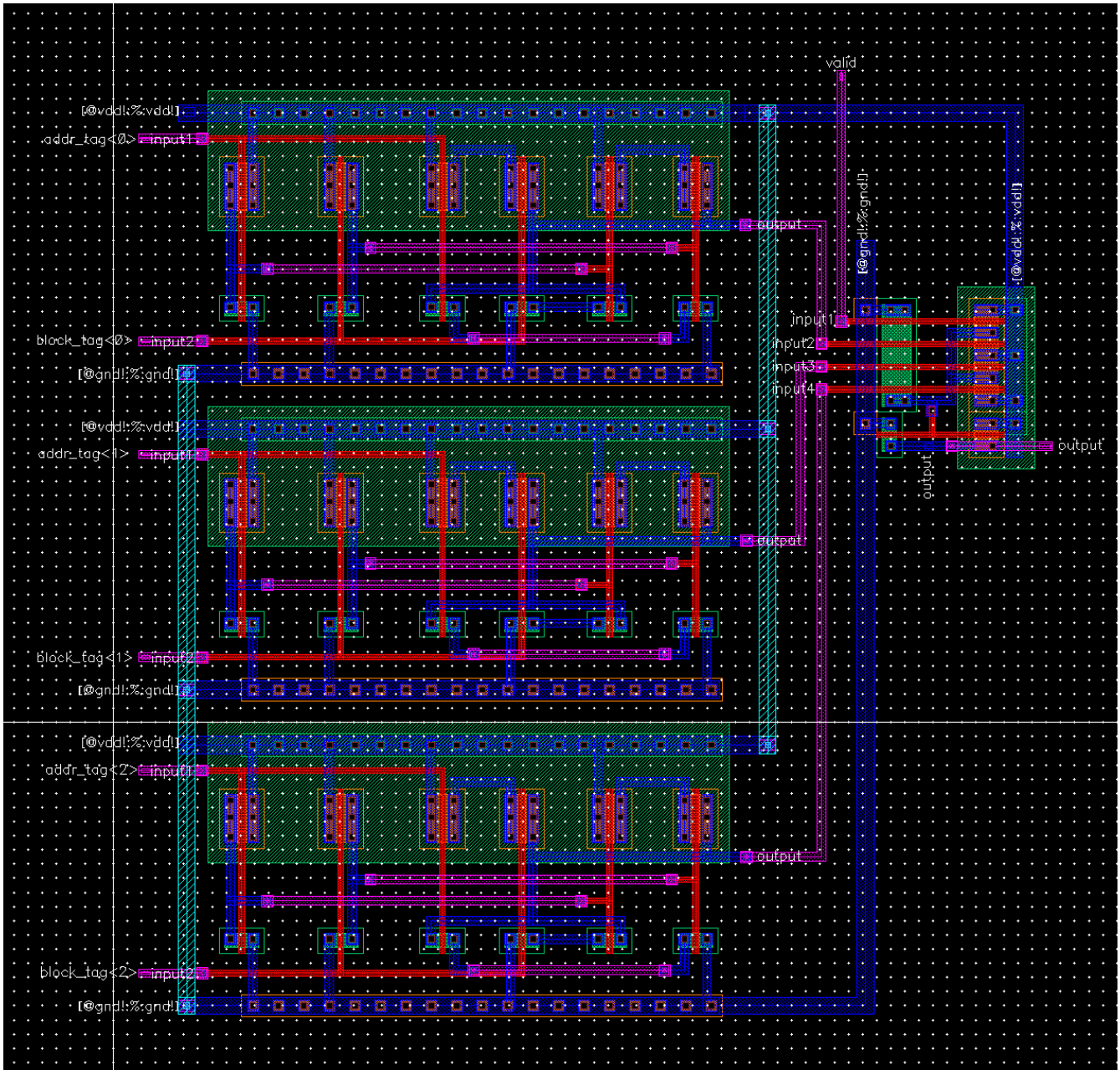
Below is a layout of the cache cell schematic



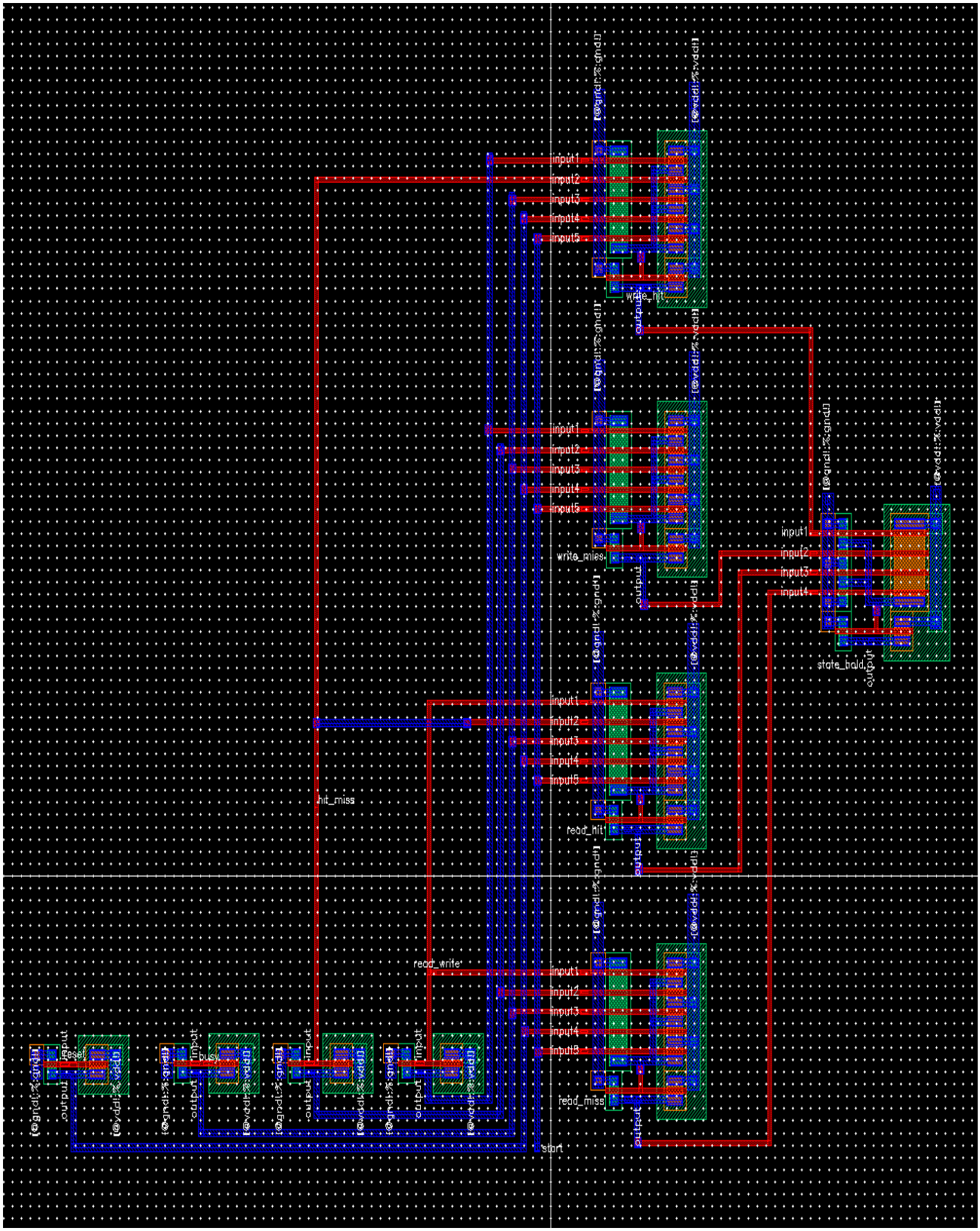
Below is a layout of the cache byte schematic



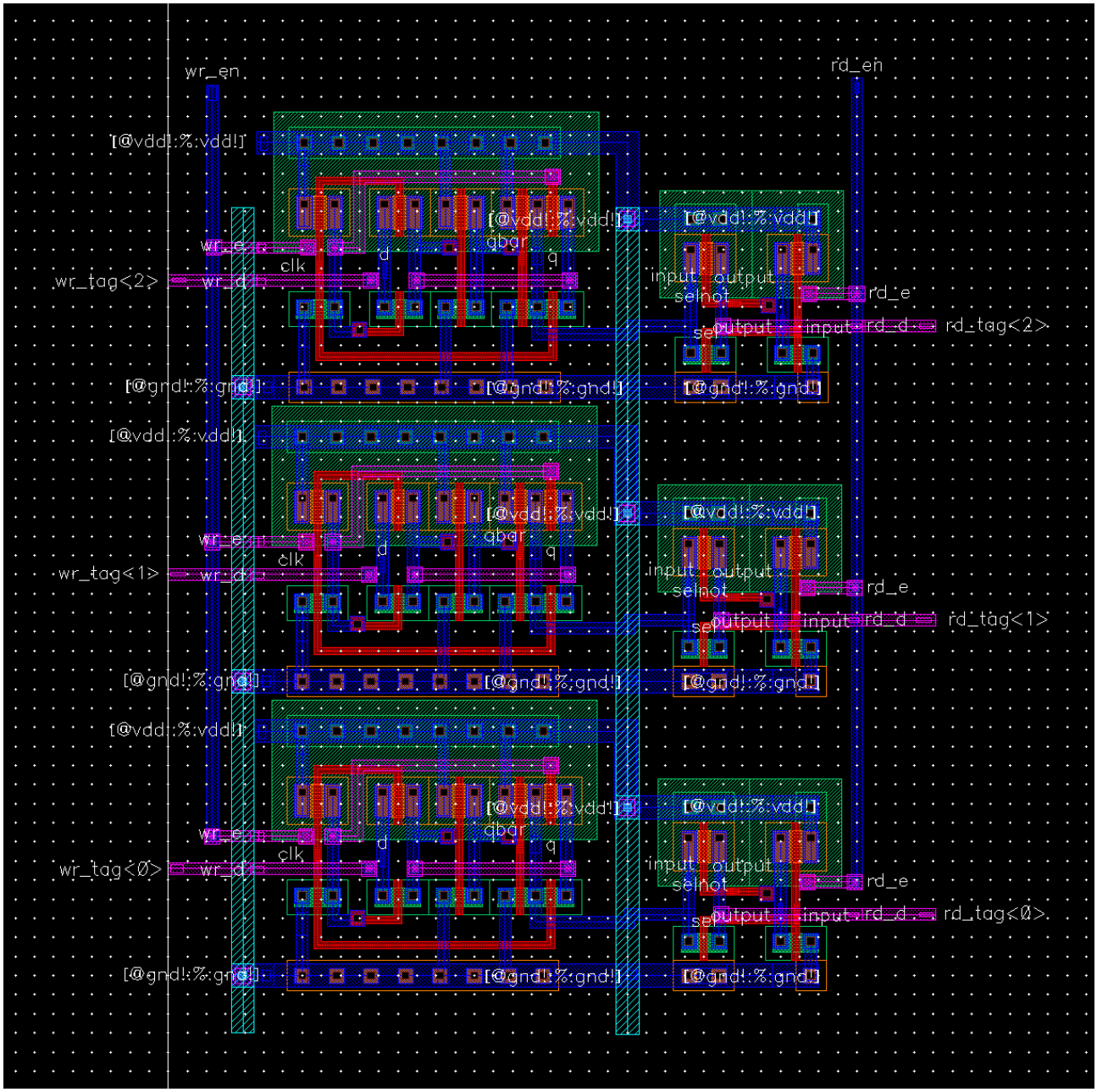
Below is a layout of the hit miss schematic



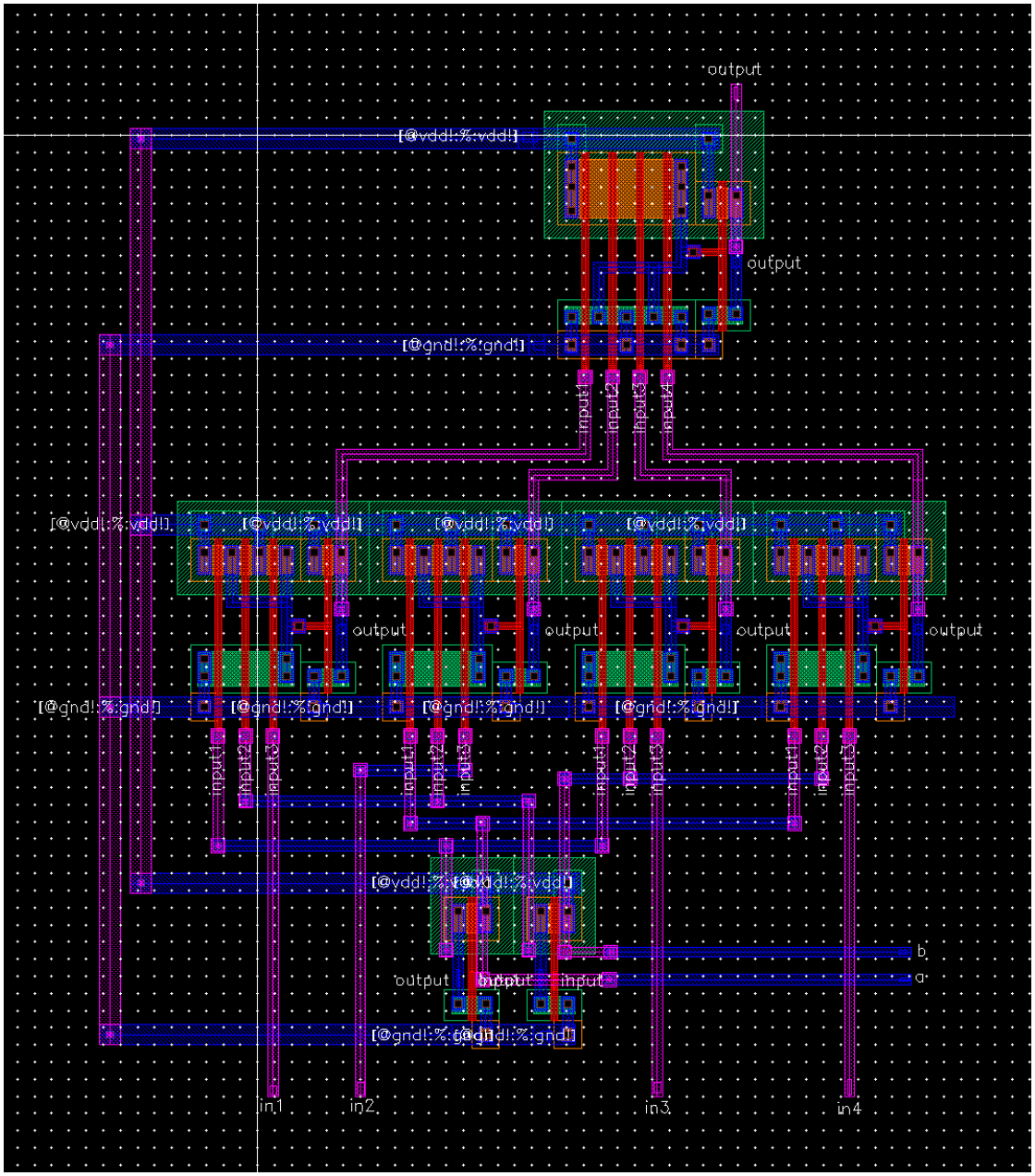
Below is a layout of the state register schematic



Below is a layout of the cache tag schematic



Below is a layout of the mux4to1 schematic



Area Of Major Components:

Component Name	Area of Component (μm^2)
Dlatch	174.96
Cache Cell	991.44
Cache Byte	9,958.545
Decode 1 to 4	2,950.425
Decode 1 to 8	7,740.9
Mux 4 to 1	1,155.06
Tag	3,318.52
Validate	621.81
Hit Miss	88.92
State Register	6,498.1125
Busy Calculator	2,520.54

Running LVS:

LVS was run for the following entities: dlatch, cache_cell, cache_byte, decod_1to4, decod_1to8, mux_4to1, tag, validate, and hit_miss. Every netlist matched appropriately. Their runs are in the “LVS_runs” folder of the final submission zip file.

Conclusion:

Overall, this project was very successful at teaching the vlsi design process. A lot was learned regarding cadence, vhdl, schematic & layout creation, and extraction/verification testing. LVS was performed at every level including the gate level logic to ensure that the device was working properly. However a lot of time was spent trying to figure out issues with the dlatch and the xnor schematic and layout. First, a dflipflop was made and that had to be changed and then for the xnor schematic a xor schematic was made instead and so that had to be changed. As a result of this and lack of time we were unable to finish all layouts.

Work Breakup:

Brendan		
Date	Worked On	Started / In Progress / Completed
10/28/2020	Took Notes on Specs	Completed
	Took Notes on Deliverables	Completed
	Completed Report Outline/Skeleton	Completed
10/30/2020	Finished Taking Notes on Deliverables	Completed
11/9/2020	Cache Cell	Completed
	Created a Make File for Compiling, Elaborating, and Simulating a single entity	Completed
11/10/2020	Simulated Cache Cell	Completed
	Created and Simulated Cache Byte	Completed
11/11/2020	finalized cache block Design	Completed
11/11/2020	Created cache block vhdl	Completed
11/12/2020	tested and verified cache_block	completed
11/12/2020	designed full 8block cache	Completed
11/12/2020	created and tested: reset & hitmiss calculator	Completed
11/12/2020	created a decoder for enable symbols	Completed
11/12/2020	tested and verified decoder	Completed
11/12/2020	added multiplexers to cache	completed
11/13/2020	Created Schematic graphics for report	completed
11/13/2020	simulated cache entity with 2 different test benches	completed
11/13/2020	verified cache entity with ta	completed
11/13/2020	put simulation, code, and observations into report	completed
12/1/2020	fixed unnoticed vhdl error (or4 entity)	completed
12/1/2020	Imported Cache VHDL code into Cadence	completed
12/1/2020	created schematics for all low level gates	completed
12/1/2020	created schematics for cache_cell, tag, and cache_byte	completed

12/2/2020	created schematics for multiplexers and decoders	completed
12/6/2020	Created cache block schematic	completed
12/6/2020	created cache memory schematic (for cache.vhdl)	completed
12/7/2020	created layout and extracted views for all lower level gates	completed
12/8/2020	having issues with xnor2 and dlatch layout, going to office hours	completed
12/8/2020	created cache_cell layout	completed
12/8/2020	cache_byte layout complete	completed
12/8/2020	finish all decoder layouts	completed
12/8/2020	finished hit miss calculation layout	completed
12/8/2020	finished mux_4to1 layout	completed
12/8/2020	finished validate layout	completed
12/8/2020	verified every completed layout with LVS	completed
12/8/2020	started 8 bit multiplexers	started
12/8/2020	started cache_block layout	started
12/8/2020	started top cache layout	started

Scott		
Date	Worked On	Started / In Progress / Completed
10/28/2020	Researched Statemachine	Completed
11/4/2020	Reaseached and went of Off hours	Completed
11/10/2020	Started Statemachine Design	Completed
11/11/2020	Worked on state machine	Completed
11/12/2020	Attempted to finish state machine	Completed
12/4/2020	Finished State Machine Code	Completed
12/7/2020	Completed State Register Layout	Completed
12/8/2020	Completed Busy Calculator Layout	Completed
12/8/2020	Completed Documentation for the project	Completed