

CMPE 315: Principles of VLSI Design

Lab Cover Page

Lab # : Project/ Lab 6
Lab Title : Cache Design Final

Name : Nitheesh Kumar Manjunath
Section : Computer Engineering

Date Submitted : 12/09/2018

TA / Grader Use Only:

Late Submission Deduction (20% per day late):

Other Deductions:

Final Lab Grade:

Comments to student:

Cache Design Project Final Report

Prepared for

Dr. Chintan Patel
Mr. Rachit Sood | Mr. Hasib-Al-Rashid

University of Maryland, Baltimore County
Department of CSEE

Prepared by

Nitheesh Kumar Manjunath

December 9, 2018

Table of Contents

List of Figures	3
1. Introduction.....	5
2. Status of the project	6
3. Block diagram of Cache.....	7
4. Chip Interface.....	8
5. Operations	10
Read Hit	10
Write Hit	11
Write Miss.....	11
Read Miss.....	11
6. State Machine.....	12
7. Simulation Procedure.....	16
8. Hierarchical breakup.....	16
8.1 Registers.....	16
1-bit register	16
8-bit register.....	17
8.2 Decoder.....	18
2 to 4 Decoder.....	18
3 to 8 Decoder.....	18
8.3 Cache.....	19
Cache cell.....	19
32-byte Cache	21
8.4 Hit/Miss block.....	23
Comparator	23
8.5 Multiplexer.....	24
8.6 Encoder	26
8.7 Tristate buffer.....	27
9. Chip.....	28
10. Miscellaneous layout view.....	30
11. Results.....	34
12. Area of the major modules in the layout.....	35
13. Enscript of Code and LVS match	35

List of Figures

Figure 1 Block diagram of the chip interface	7
Figure 2 Schematic of the Chip	8
Figure 3 CPU Address	10
Figure 4 State Machine of chip	13
Figure 5 State machine Schematic	13
Figure 6 Simulation of Read hit state in State Machine	13
Figure 7 Simulation of Read miss state in State Machine	14
Figure 8 Simulation of Write hit state in State Machine	14
Figure 9 Simulation of Write miss state in State Machine	14
Figure 10 Layout view of the state machine.....	15
Figure 11 1-bit register.....	16
Figure 12 8-bit register.....	17
Figure 13 Simulation of Register using D flip flop	17
Figure 14 Layout view of 8-bit register using d flip flop	17
Figure 15 2-4 Decoder	18
Figure 16 3-8 Decoder	18
Figure 17 Simulation of decoder.....	19
Figure 18 Layout view of 2-4 decoder.....	19
Figure 19 Layout view of 3-8 decoder.....	19
Figure 20 1-bit Cache cell.....	20
Figure 21 Simulation of 1-bit cache cell.....	20
Figure 22 Layout view of 1-bit cache cell	20
Figure 23 32-byte Cache.....	21
Figure 24 Simulation of 32-byte(256-bit) Cache.....	21
Figure 25 Layout view of 32-byte Cache	22
Figure 26 Comparator	23
Figure 27 Hit/Miss block with comparator and decoder	23
Figure 28 Simulation of Hit-Miss block	24
Figure 29 Layout view of hit-miss block.....	24
Figure 30 2-1 MUX	24
Figure 31 Simulation of 8-bit 2 to 1 multiplexer	25
Figure 32 Layout view of 8-bit 2-1 MUX	25
Figure 33 Encoder.....	26
Figure 34 Simulation of encoder.....	26
Figure 35Layout view of encoder	27
Figure 36 Tristate buffer	27
Figure 37 Simulation of tristate buffer.....	28
Figure 38 Layout view of 8-bit Tristate buffer	28
Figure 39 Schematic of chip	28
Figure 40 Layout view of the Chip	29
Figure 41 Layout view of and2	30

Figure 42 Layout view of busy1	30
Figure 43 Layout view of cac_dec_sgn	30
Figure 44 Layout view of cmprtr	30
Figure 45 Layout view of cache_32b.....	31
Figure 46 Layout view of col_sel_tgcm	31
Figure 47 Layout view of dlatch.....	31
Figure 48 Layout view of inverter	32
Figure 49 Layout view of mux2x1.....	32
Figure 50 Layout view of nand2	32
Figure 51 Layout view of nor2	33
Figure 52 Layout view of or2	33
Figure 53 Layout view of tag_cmp1	33
Figure 54 Layout view of xnor2	34
Figure 55 Simulation showing read miss state	34
Figure 56 Simulation showing write hit, read hit and write miss	34

1. Introduction

Cache is a hardware component that stores data so that future requests for that data can be served faster; the data stored in a cache might be the result of an earlier computation or a copy of data stored elsewhere. A *cache hit* occurs when the requested data can be found in a cache, while a *cache miss* occurs when it cannot. Cache hits are served by reading data from the cache, which is faster than recomputing a result or reading from a slower data store; thus, the more requests that can be served from the cache, the faster the system performs.

Hardware implements cache as a block of memory for temporary storage of data likely to be used again. Central processing units (CPUs) and hard disk drives (HDDs) frequently use a cache, as do web browsers and web servers. A cache is made up of a pool of entries. Each entry has associated *data*, which is a copy of the same data in some *backing store*. Each entry also has a *tag*, which specifies the identity of the data in the backing store of which the entry is a copy.

Common types of cache include the following

1. **Browser cache** - Most web browsers cache webpage data by default. For example, when you visit a webpage, the browser may cache the HTML, images, and any CSS or JavaScript files referenced by the page. When you browse through other pages on the site that use the same images, CSS, or JavaScript, your browser will not have to re-download the files. Instead, the browser can simply load them from the cache, which is stored on your local hard drive.
2. **Memory cache** - When an application is running, it may cache certain data in the system memory, or RAM. For example, if you are working on a video project, the video editor may load specific video clips and audio tracks from the hard drive into RAM. Since RAM can be accessed much more quickly than a hard drive, this reduces lag when importing and editing files.
3. **Disk cache** - Most HDDs and SSDs include a small amount of RAM that serves as a disk cache. A typical disk cache for a 1 terabyte hard drive is 32 megabytes, while a 2 TB hard drive may have a 64 MB cache. This small amount of RAM can make a big difference in the drive's performance. For example, when you open a folder with many files, the references to the files may be automatically saved in the disk cache. The

next time you open the folder, the list of files may load instantly instead of taking several seconds to appear.

4. **Processor cache** - Processor caches are even smaller than disk caches. This is because a processor cache contains tiny blocks of data, such as frequently used instructions, that can be accessed quickly by the CPU. Modern processors often contain an L1 cache that is right next to the processor and an L2 cache that is slightly further away. The L1 cache is the smallest (around 64 KB), while the L2 cache may be around 2 MB in size. Some high-end processors even include an L3 cache, which is larger than the L2 cache. When a processor accesses data from a higher-level cache, it may also move the data to the lower level cache for faster access next time.

A cache usually needs two address and data interfaces, one to the upper level cache/processor and another to lower level cache/memory. In this design, both these interfaces, 8-bit address and data connections are present to both the CPU and the memory.

2. Status of the project

Completed.

Both the test benches (chip_test and chip_full_test) are simulating as expected. In the chip_full_test make the reset high for two clock cycles as in the waveform given [Project Waveforms](#).

3. Block diagram of Cache

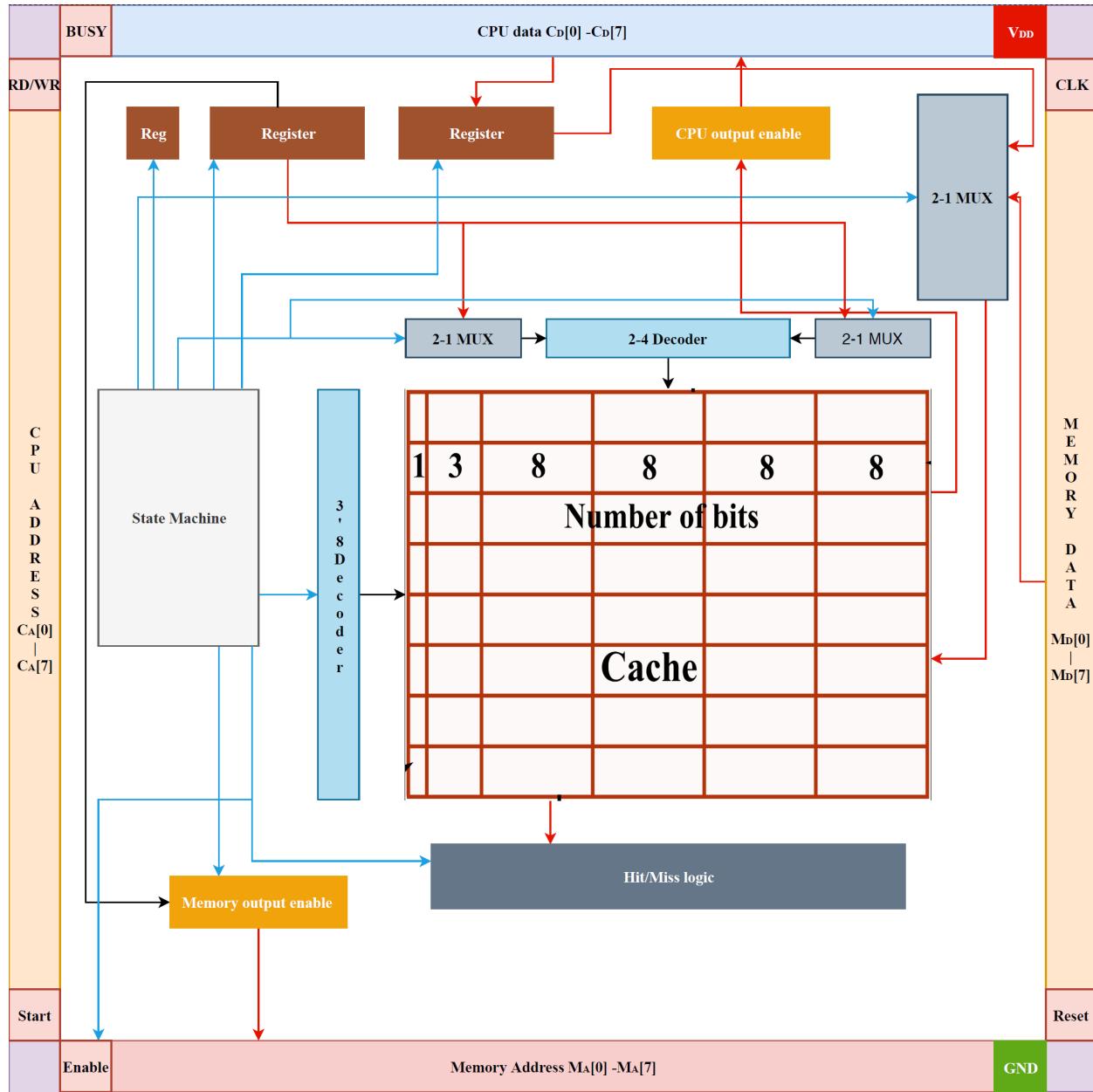


Figure 1 Block diagram of the chip interface

The above figure shows the block diagram of the chip interface of my design. The state machine controls the flow of the data to the different lower blocks. The description of the lower level blocks is given in the later stage of this report. The cache designed consists of 8 blocks so 3-bit block address is required to locate a block and 4 bytes so 2-bit byte address is required.

The reference block diagram from the project description was used to construct the design logic of my project. The schematic block diagram of the code is shown below.

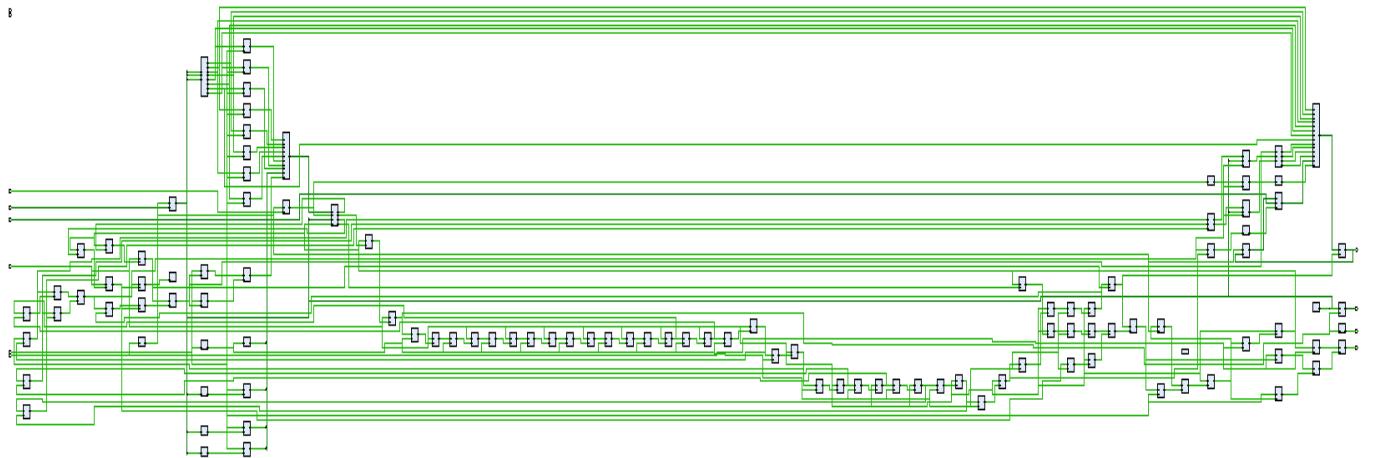


Figure 2 Schematic of the Chip

4. Chip Interface

CPU Address (CA): 8-bit address input from the CPU for both read and write requests. Address will be provided by the CPU on the rising edge of clock along with the start signal

CPU Data (CD): 8-bit input/output data bus. Used as input for write requests and output for read requests. Data will be provided on the rising edge of clock for write requests along with the start signal

Start: Handshaking signal indicating the start of a read/write request from the CPU, goes high on a positive edge. (Internally your chip works on the negative edge. The address, start, read_write control and data, will be setup half a clock cycle before you start operating on the negative edge)

Busy: Handshaking signal indicating to the CPU that your chip is processing the previous request

Read Write (RD/WR): If signal is high CPU is requesting a read operation, if low a write operation

Reset: Master reset to the chip. A high on reset should invalidate all the entries in the cache and reset your state machine to its reset state. Reset any other registers as required.

Clock (clk): Clock input to the chip. The CPU inputs/outputs and the memory inputs/outputs will be synchronized with this clock signal also

Memory Address (MA): Address output to the memory in case of read miss. The last two bits of the address should always be 00 i.e. you provide the address for the last word in the block. The memory controller will automatically increment the address four times and provide you data for the whole block (4 bytes)

Memory Data (MD): Data input from the memory. It takes the memory 8 clock cycles after getting the enable signal to provide the first byte of data. It will sequentially provide four bytes of data required for the whole block. The first data byte will become valid on the 8th negative edge after asserting enable and will stay stable for 2 clock cycles. The next byte will be provided on the 10th negative edge after asserting enable and will stay stable for 2 clock cycles. The last two data bytes will have similar timing and will be provided on the 12th and 14th negative edges.

Enable: The enable signal is an output requesting the memory to perform a read operation starting at the address provided on MA. The memory returns four bytes as explained above starting at the 8th clock cycle after asserting this signal.

5. Operations

The cache has 8 blocks, with four bytes per block. Therefore, you need 3 bits to select one of the 8 blocks and 2 bits to select the correct byte from the block. The remaining 3 bits in the address are used as a tag. The start signal from the CPU marks the beginning of an operation. The CPU provides the address for read operations and the address as well as data for write operations. All these signals are provided on the rising edge of clock. Your chip internally works at the falling edge of clock, so the signals are stable before you need to latch them thus avoiding setup time violations. The signals will be removed by the CPU on the next rising edge of clock. The busy signal triggers the removal of these signals. Busy is an output that should go high on the negative edge after receiving the start signal from the CPU.

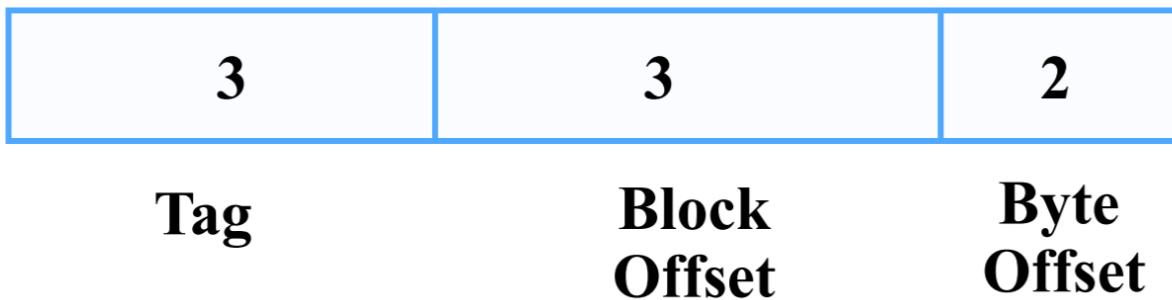


Figure 3 CPU Address

Read Hit

For a read operation, the CPU turns start high, provides the address and turns read/write signal high on the positive edge of clock. On the negative edge you need to latch all these required signals and turn on the busy signal. The inputs will be removed by the CPU on the positive edge once it receives a busy signal. You determine whether the data being referenced is in the cache, by comparing the tag bits of the address with the tag bits stored in the block given by the block offset in the address. Also, you need to check that the block is valid. Simultaneously you need to read the correct byte from the cache. If the tag and valid checking operation signals a hit, output enable should go high on the next negative edge, the data should be latched in the output register so that

it stays stable for one whole clock cycle and the busy signal should be turned off. The CPU will read the data off the data bus on the positive edge. The next operation could be requested on any subsequent clock cycle.

[Write Hit](#)

CPU signals follow the same timing as the read but read write is set to low and the data to be written is provided. The required inputs should be latched on the negative edge, busy should be turned on and tag/valid compare should be performed. If the result is a hit, on the next negative edge the data should be written to the correct byte in the selected block, busy should still stay high. On the second negative edge after receiving the write request busy should go low signaling the end of the write operation. The CPU will provide a new input on any subsequent clock cycle.

[Write Miss](#)

As we are using write through with no write allocate, you don't have to get the block from memory on a write miss. The timing and operations are like the write hit case. Only difference is that as you don't have a hit, no write operation should be performed on the cache. The busy should go low after two clocks from receiving the write request as in the hit case and the CPU can provide a new request after busy is low.

[Read Miss](#)

The signals are provided by the CPU as described in the read hit operation. On the second negative edge of the clock, you get a miss in this case i.e. the block is not present in the cache. On this edge you keep busy high, turn on the enable signal to the memory requesting a block read and provide the byte address for the last byte in the block that you need. The memory latches the address once it sees enable is high on a positive edge. Enable should be turned low after one clock cycle (third negative edge since receiving the request). The memory needs 8 clock cycles to access the data. After 8 clock cycles, the memory provides the data for the last byte on the negative edge. The data will stay stable for 2 clock cycles. You should setup the correct address for write and update the cache with the new data byte. During the first write operation you should also update the tag and turn on the valid bit. Write timing is very critical, if there are any glitches in the address lines or

the data lines while the write signal to the cache is enabled, you could write the wrong data or overwrite another location in the cache. Your write signal should be turned on exactly for one clock cycle from the positive edge after the data is valid to the positive edge before the new data byte is provided. The address should be stable for 2 clock cycles, like the data. The address should be stable for half a clock cycle before the write is turned on and for half a clock cycle after the write is turned off. Once the bytes are written, the correct byte requested by the CPU should be read and latched on the output register, busy should be turned off and the output enabled on a negative edge. Output enable should go low after one clock cycle as in the case of read hit. CPU will read the data on the positive edge after output enable goes high and will provide a new request on any subsequent clock cycle.

6. State Machine

As described in the previous section the operations carry on accordingly. The State machine is built using the [Negative edge triggered D Flip Flop](#). The schematic of the state machine is shown in figure 4. The state machine operates according to the hit/miss logic. Once start signal is applied by the CPU and reset signal is lifted state machine enters state 1. Based on the output from the hit miss logic the state machine operates. If the hit/miss block gives a read hit then the state machine runs only for two states, if it gives read miss then state machine runs for 19 states. For write hit and mis state machine runs for 3 states. Once the state machine enters the state 1 then busy signal is pulled high cache is set to read and tag compare is enabled. In read miss memory is enabled during the state 2 and CPU address is sent to memory. When the memory provides with the data then 2-4 decoder is enabled to select the byte to write in. In read hit in state 2 the cache memory and decoder are enabled to read from the respective block and byte. In case of write hit, the similar thing is done but read-write signal is set to write so the data is written. In write miss the state machine simply waits for 3 states and then goes to state 0. In every case before the last state the busy is turned off to signal the CPU that operation is done and is waiting for next operation.

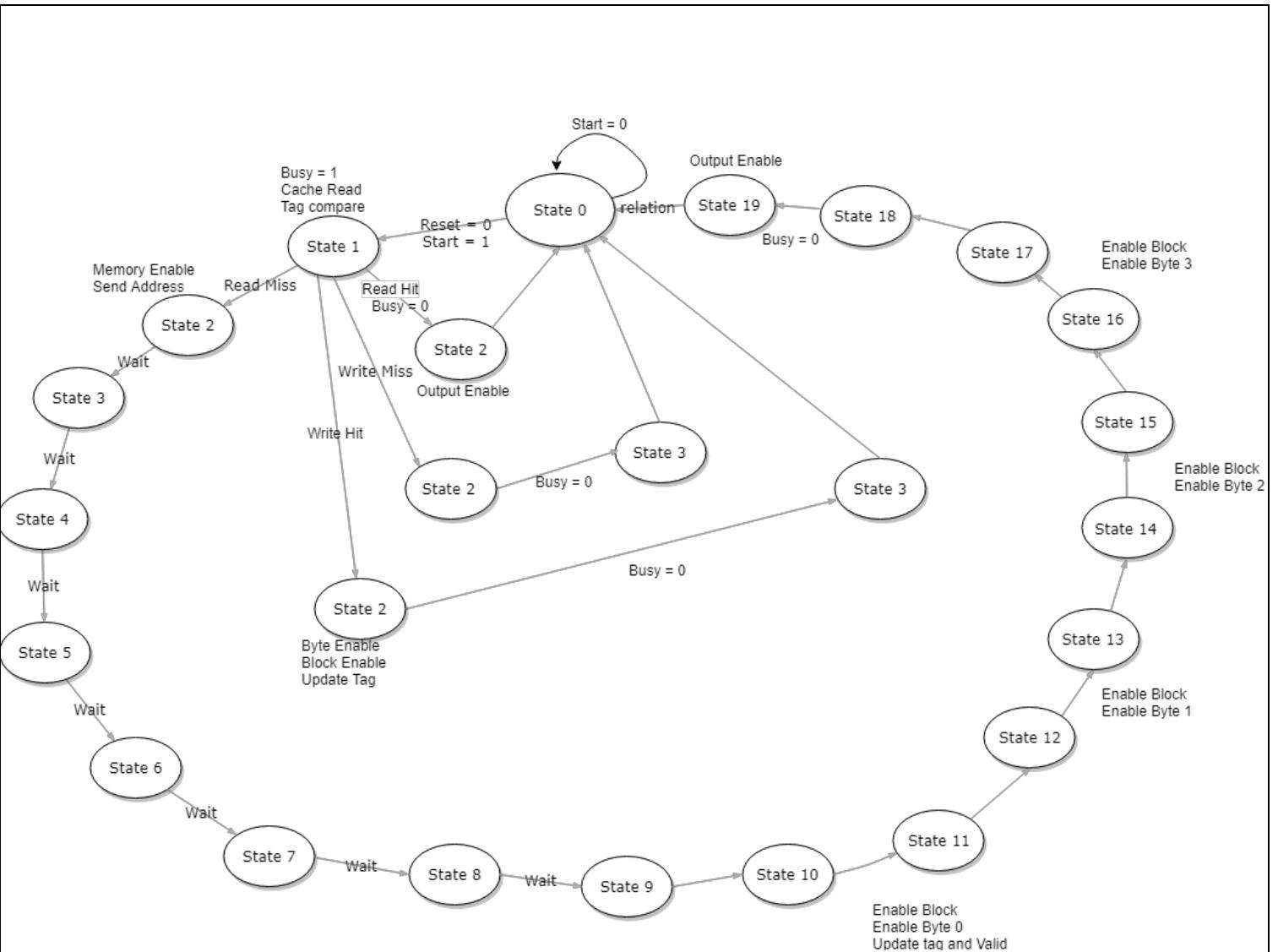


Figure 4 State Machine of chip

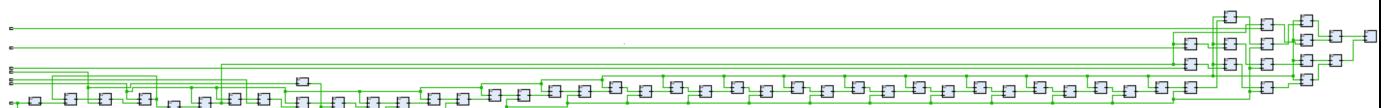


Figure 5 State machine Schematic

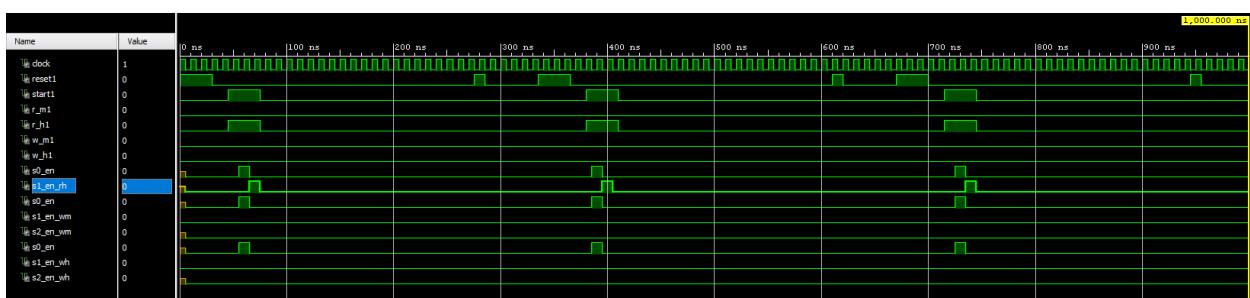


Figure 6 Simulation of Read hit state in State Machine

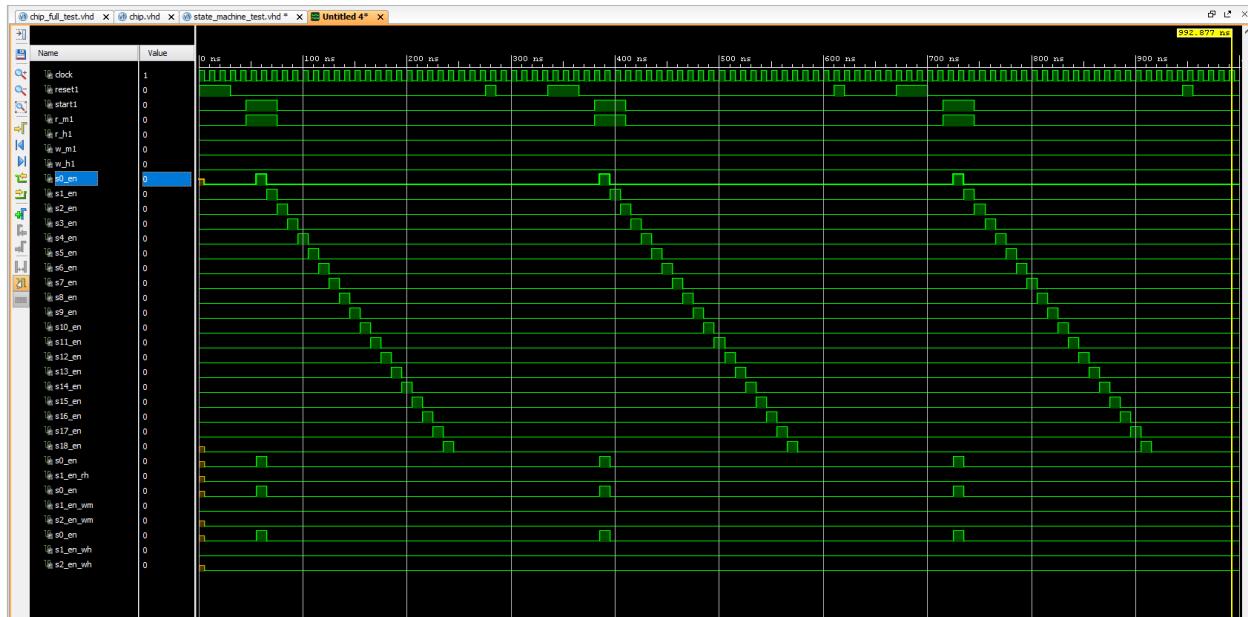


Figure 7 Simulation of Read miss state in State Machine

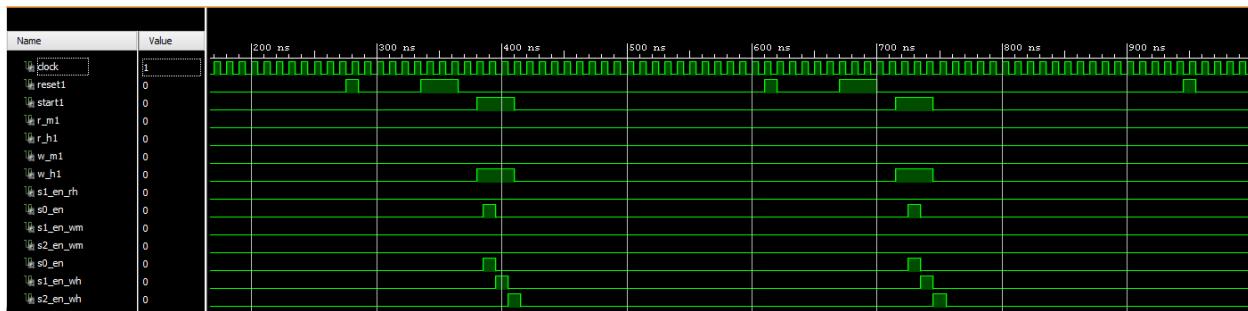


Figure 8 Simulation of Write hit state in State Machine

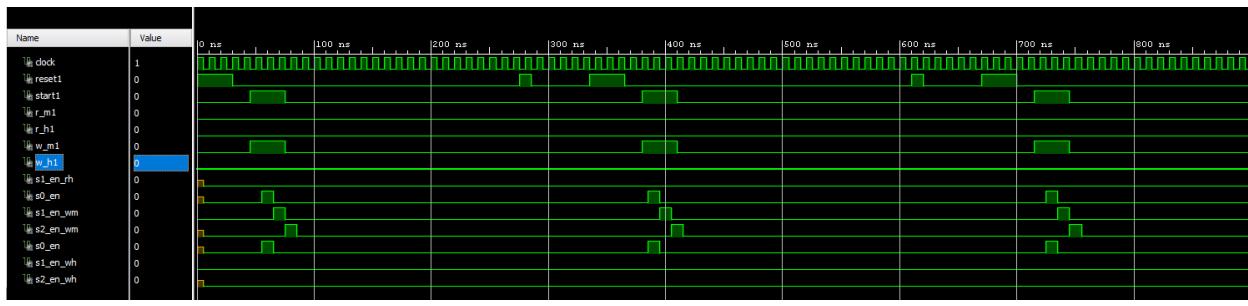


Figure 9 Simulation of Write miss state in State Machine

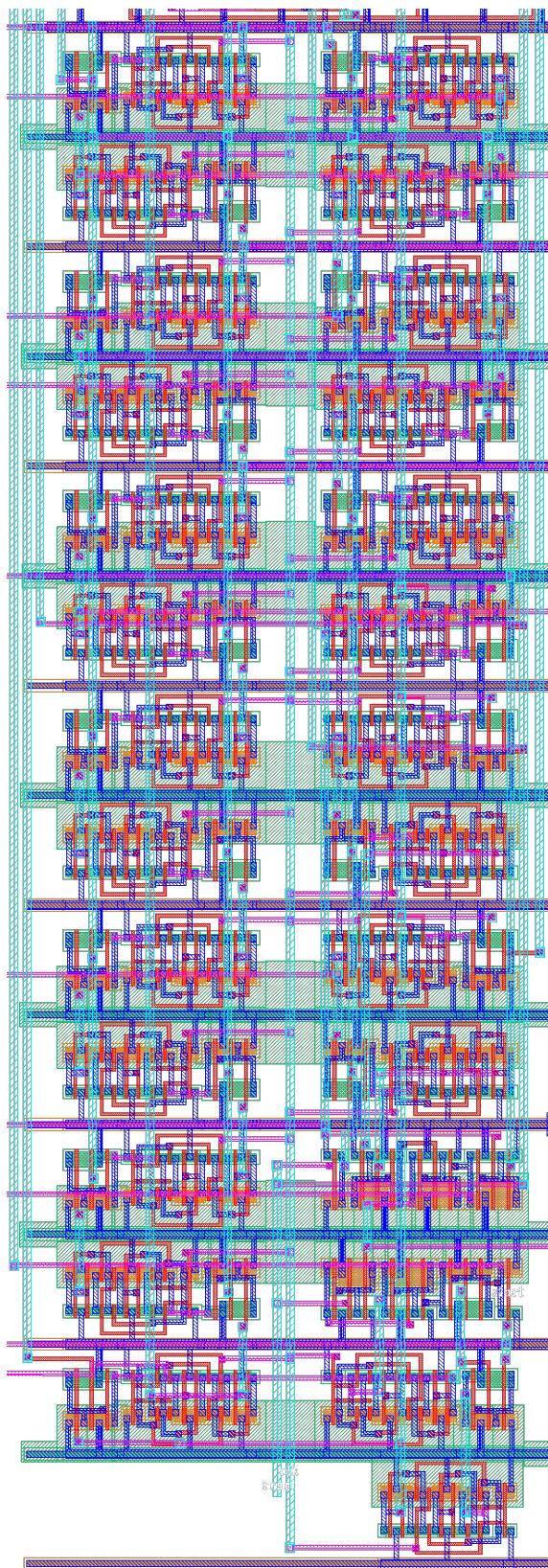


Figure 10 Layout view of the state machine

7. Simulation Procedure

Navigate to the directory where the cadence .tar file was unzipped in the setup using the change directory command i.e. cd /home/.../cadence.

Run the following command

```
run_ncvhdls.bash -messages -linedebug -cdslib <path to cds.lib> -hdlvar <path to hdl.var> -smartorder <path to filename.vhd>
```

Do the same for the test bench code also.

To elaborate the VHDL *test bench* code the following command is run in command window.

```
run_ncelab.bash -messages -access rwc -cdslib <path to cds.lib> -hdlvar <path to hdl.var> <filename>
```

To simulate the following command is run in command window, before that one has to verify if the *ncsim.run* file is stored in the cadence folder.

```
run_ncsim.bash -input ncsim.run -messages -cdslib <path to cds.lib> -hdlvar <path to hdl.var> <filename>
```

8. Hierarchical breakup

8.1 Registers

1-bit register

This is a 1-bit register which stores the read write signal from the CPU to be used once the read write signal is removed from the CPU.

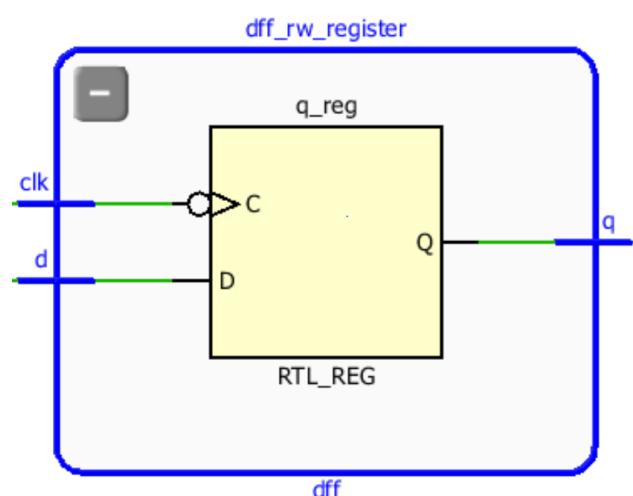


Figure 11 1-bit register

8-bit register

An 8-bit register is used to store the CPU address and CPU data. This is used whenever necessary.

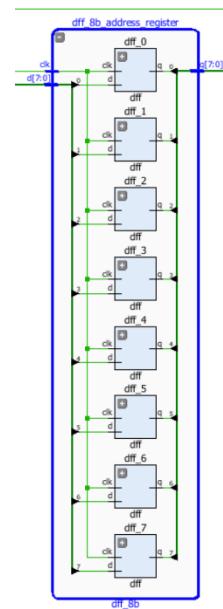


Figure 12 8-bit register

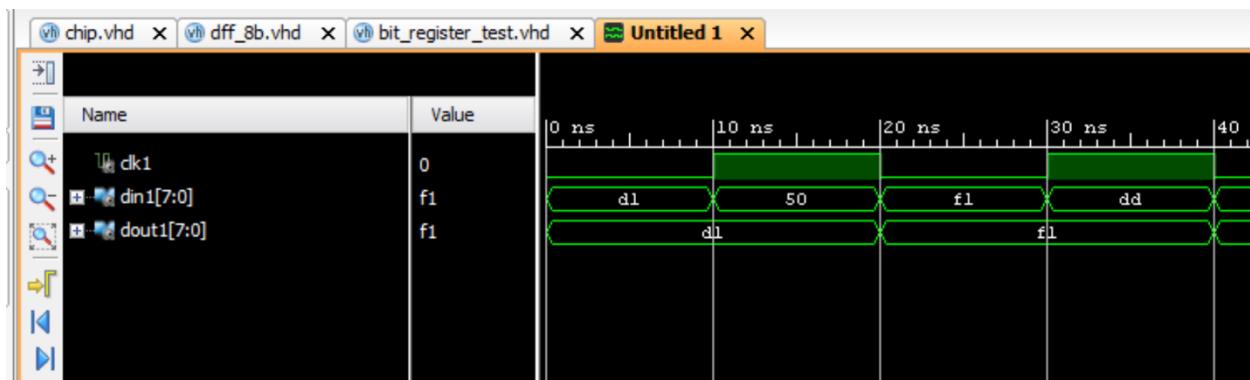


Figure 13 Simulation of Register using D flip flop

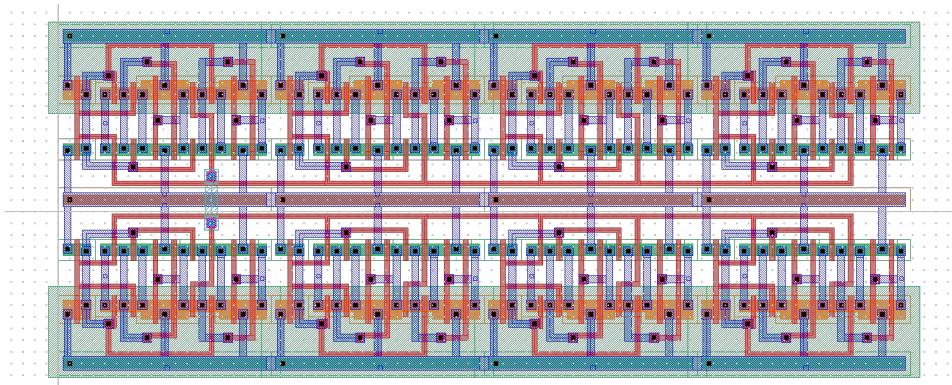


Figure 14 Layout view of 8-bit register using d flip flop

8.2 Decoder

2 to 4 Decoder

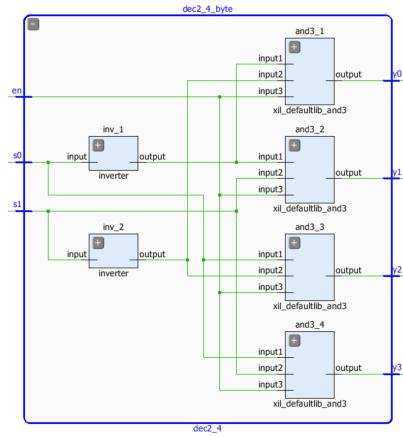


Figure 15 2-4 Decoder

In this project 2-4 decoder is used in two places. First it is used to choose the correct byte address to read from and write to, second it is used in Hit/Miss block decide if the given hit/miss is a read or write hit-miss.

3 to 8 Decoder

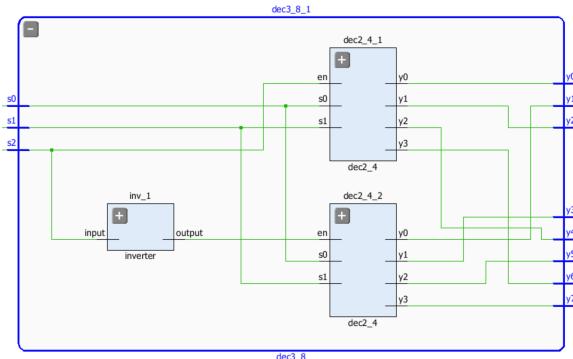


Figure 16 3-8 Decoder

This is used to select a block using the CPU block address. It is also used to read from and write to the tag and valid cache.

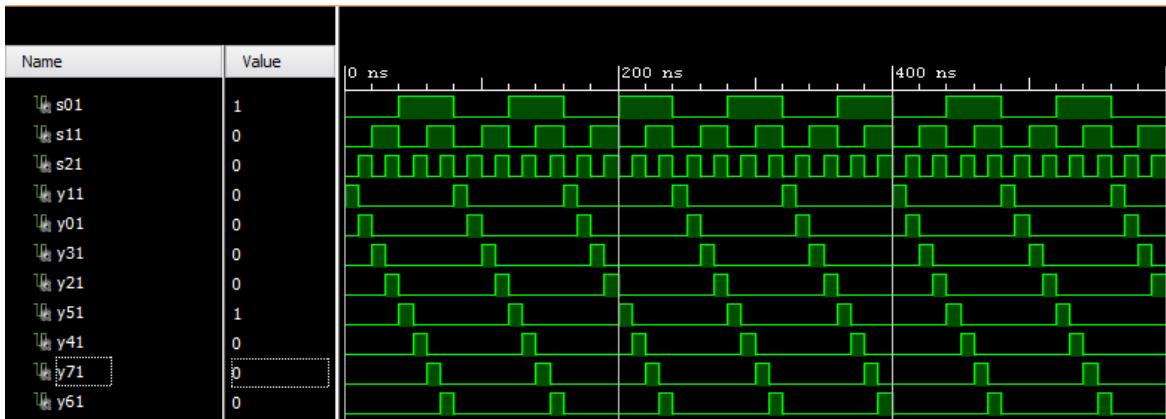


Figure 17 Simulation of decoder

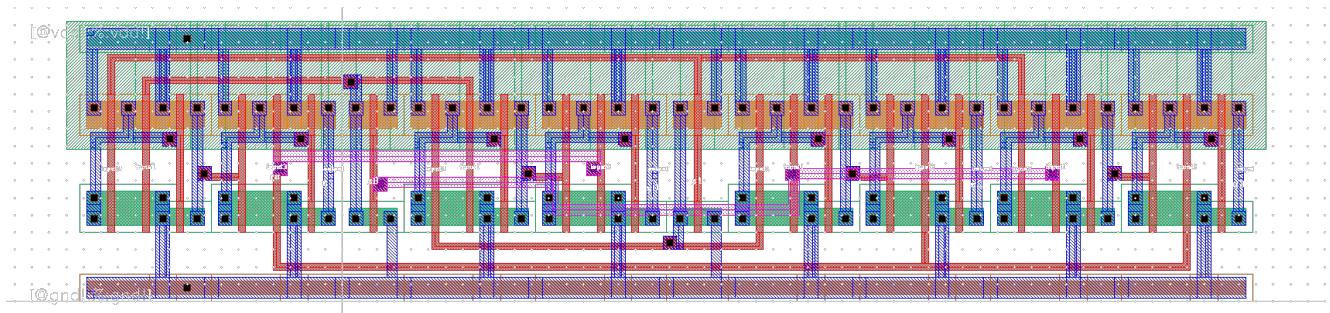


Figure 18 Layout view of 2-4 decoder

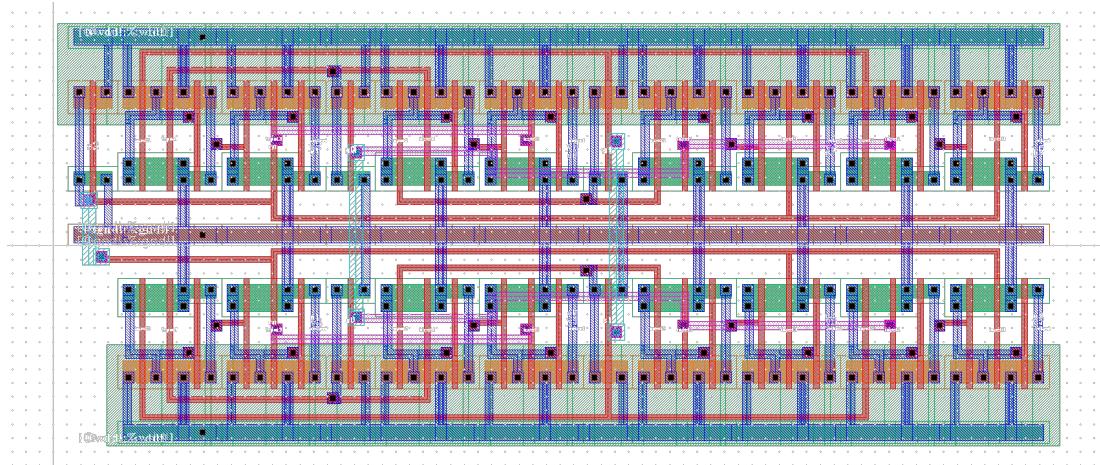


Figure 19 Layout view of 3-8 decoder

8.3 Cache

Cache cell

Cache cell is designed using a positive level sensitive latch and a transmission gate. It can store a 1-bit value.

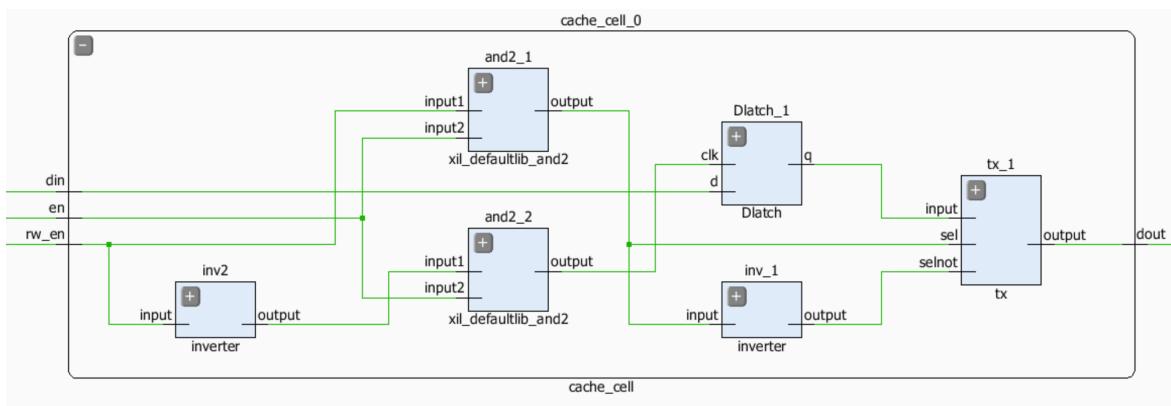


Figure 20 1-bit Cache cell

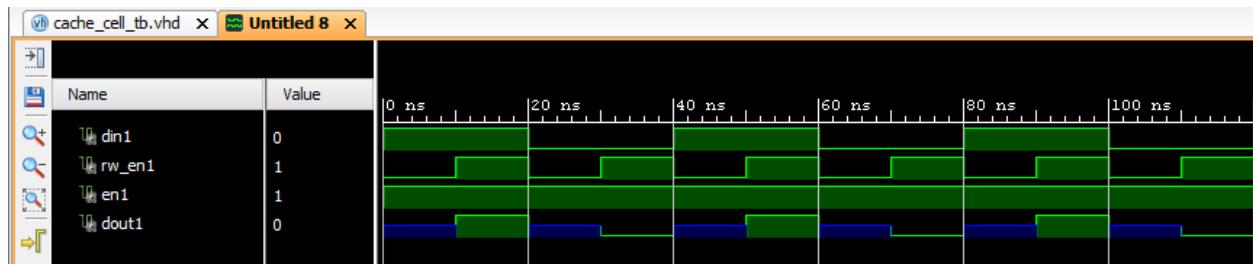


Figure 21 Simulation of 1-bit cache cell

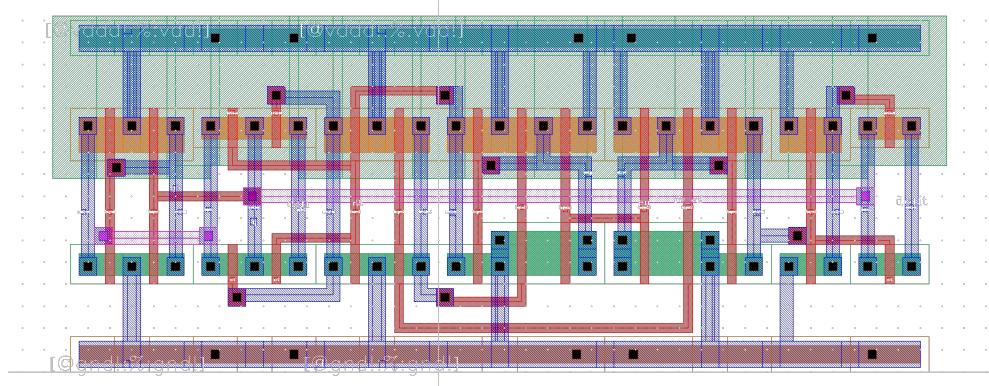


Figure 22 Layout view of 1-bit cache cell

32-byte Cache

32-byte cache consists of eight blocks of 32-bit cache which in turn consists of a four 8-bit cell made from latch and transmission gate.

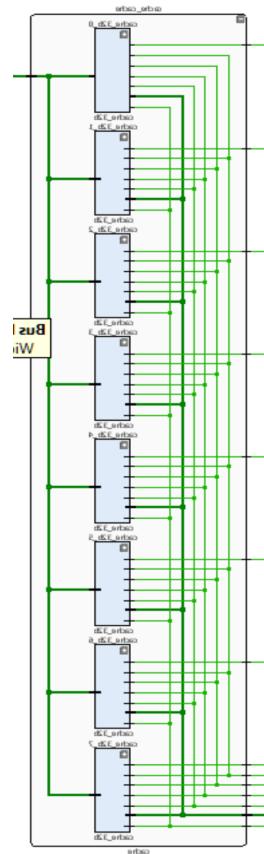


Figure 23 32-byte Cache

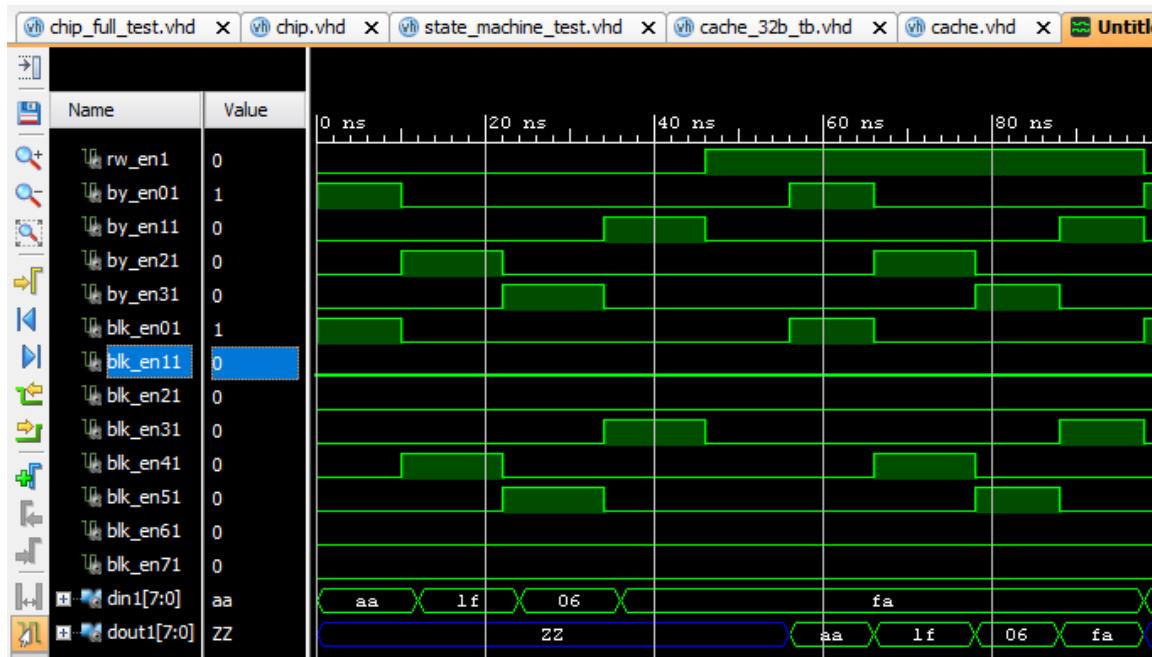


Figure 24 Simulation of 32-byte(256-bit) Cache

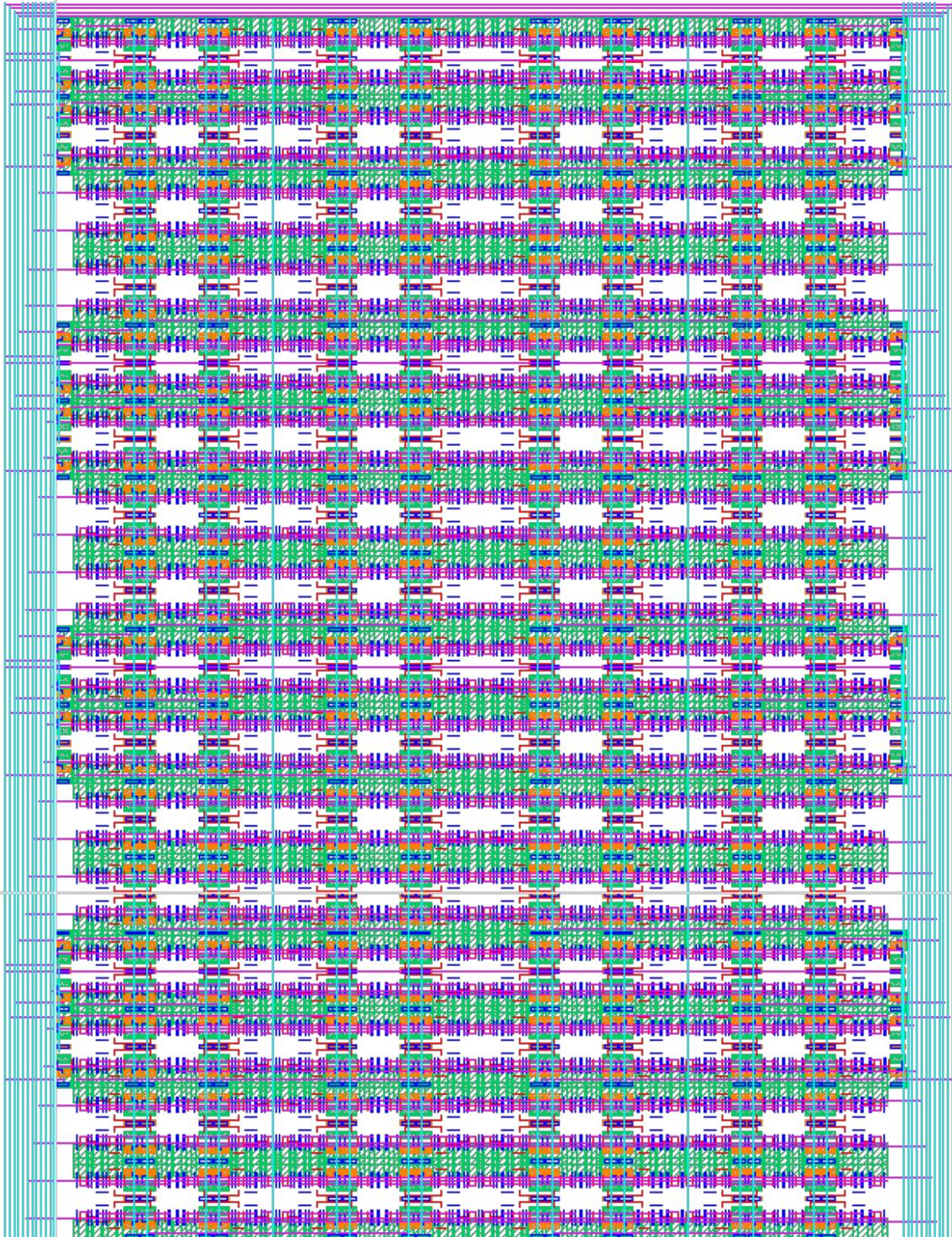


Figure 25 Layout view of 32-byte Cache

8.4 Hit/Miss block

Comparator

A 3-bit comparator is used in the design. It compares the tag bits from the CPU address with the tag bits in the Cache. The 4th bit in the input1 is the valid bit. This gives the hit or a miss logic.

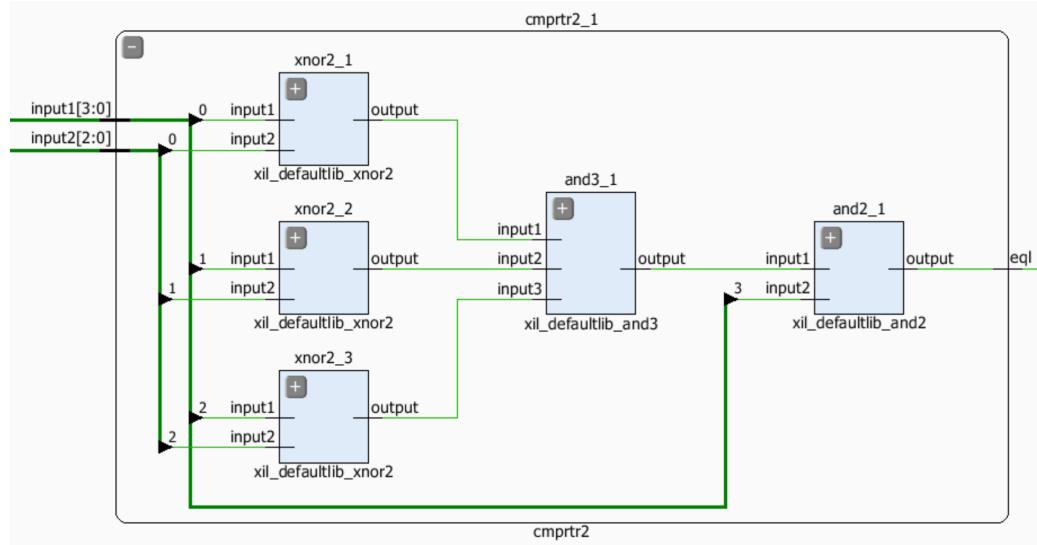


Figure 26 Comparator

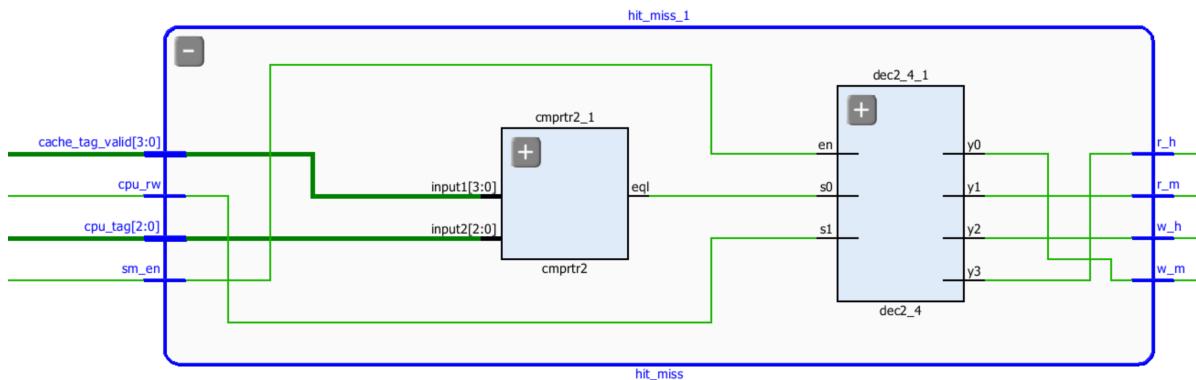


Figure 27 Hit/Miss block with comparator and decoder

The hit/miss logic from the comparator is given into the decoder which will decide the read hit, read miss, write hit and write miss logic.

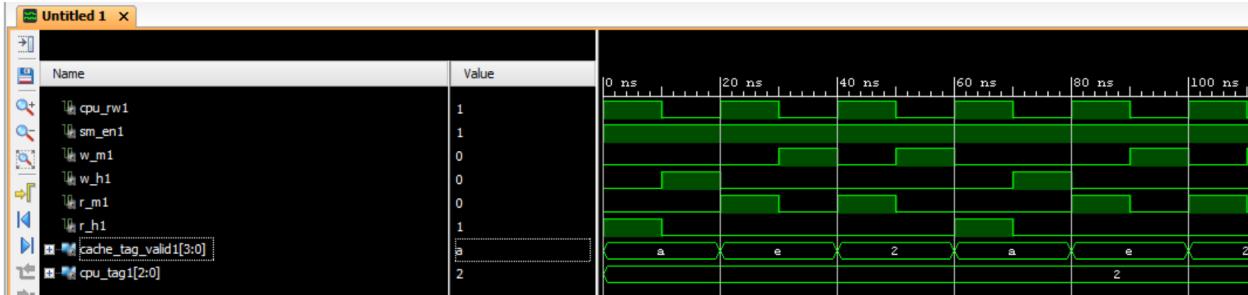


Figure 28 Simulation of Hit-Miss block

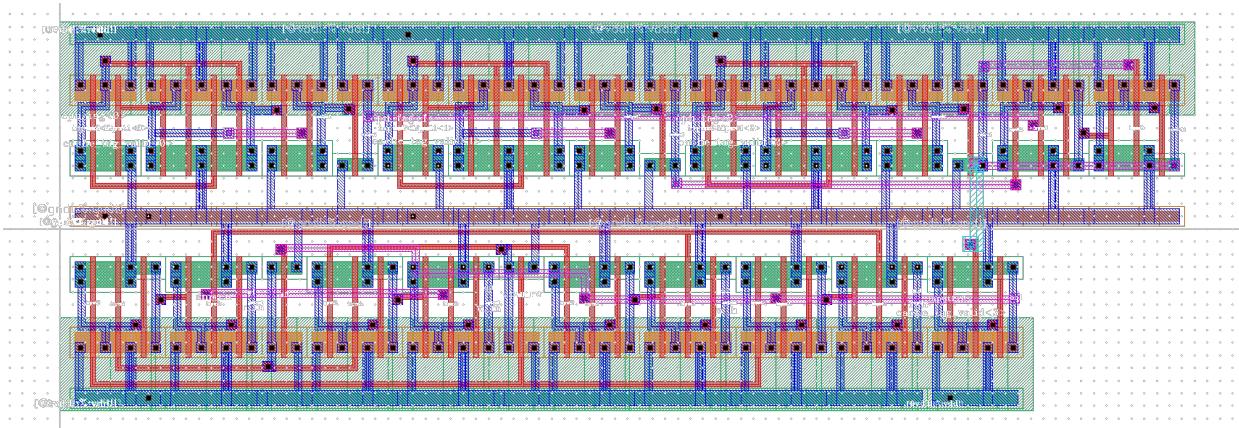


Figure 29 Layout view of hit-miss block

8.5 Multiplexer

The 2 to 1 mux is used in the design. Mux is used in the two places in design. An 8-bit 2 to 1 mux is used to select the CPU data or the Memory data. Two 1-bit 2 to 1 mux is used in selecting the byte to read from or write to the cache.

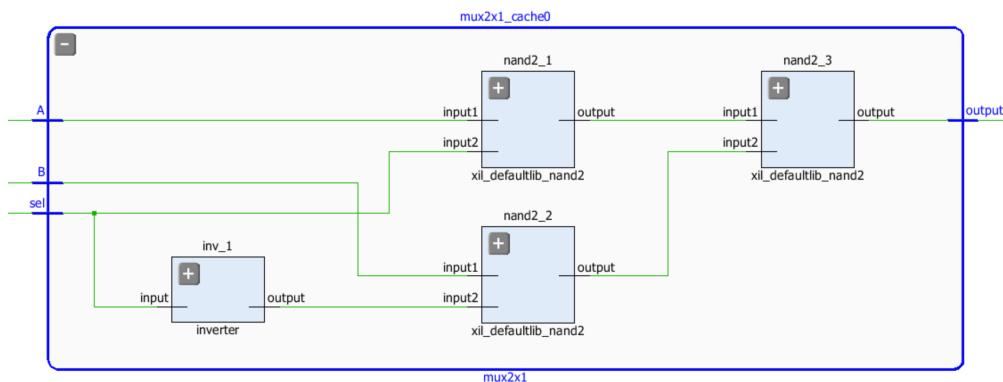


Figure 30 2-1 MUX

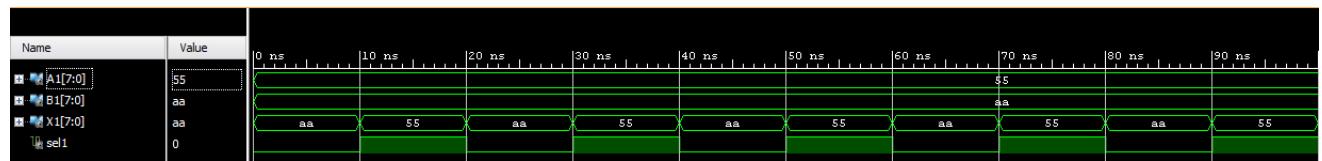


Figure 31 Simulation of 8-bit 2 to 1 multiplexer

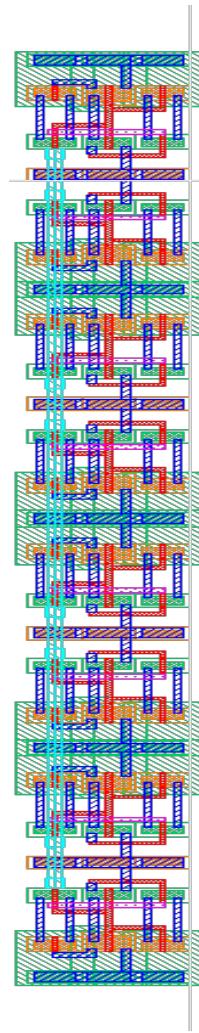


Figure 32 Layout view of 8-bit 2-1 MUX

8.6 Encoder

Encoder is used in design to encode the state machine signals 3-bit to 2-bit which is used to select the byte address with the help of the decoder.

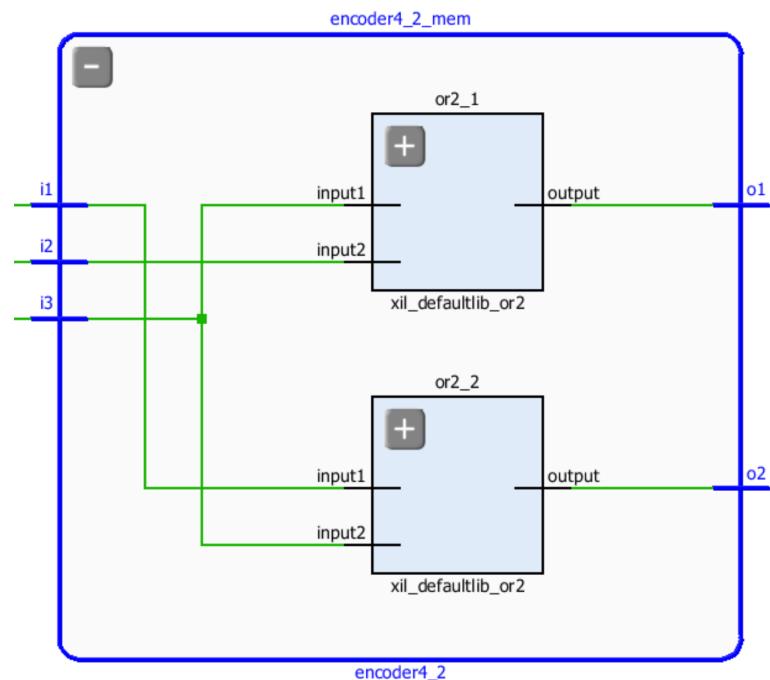


Figure 33 Encoder

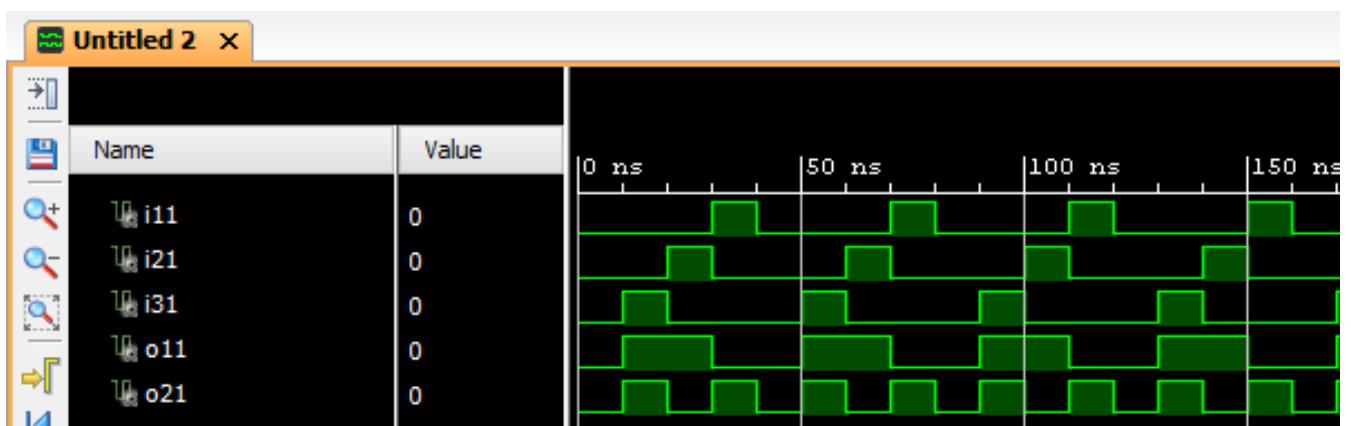


Figure 34 Simulation of encoder

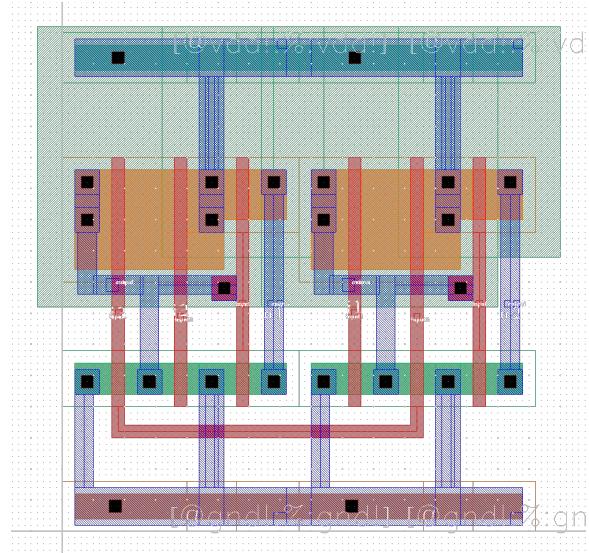


Figure 35 Layout view of encoder

8.7 Tristate buffer

Tristate buffer is used to allow one-way connection between the cache and CPU data. Since CPU data is in-out if you don't put a tristate buffer then errors occur.

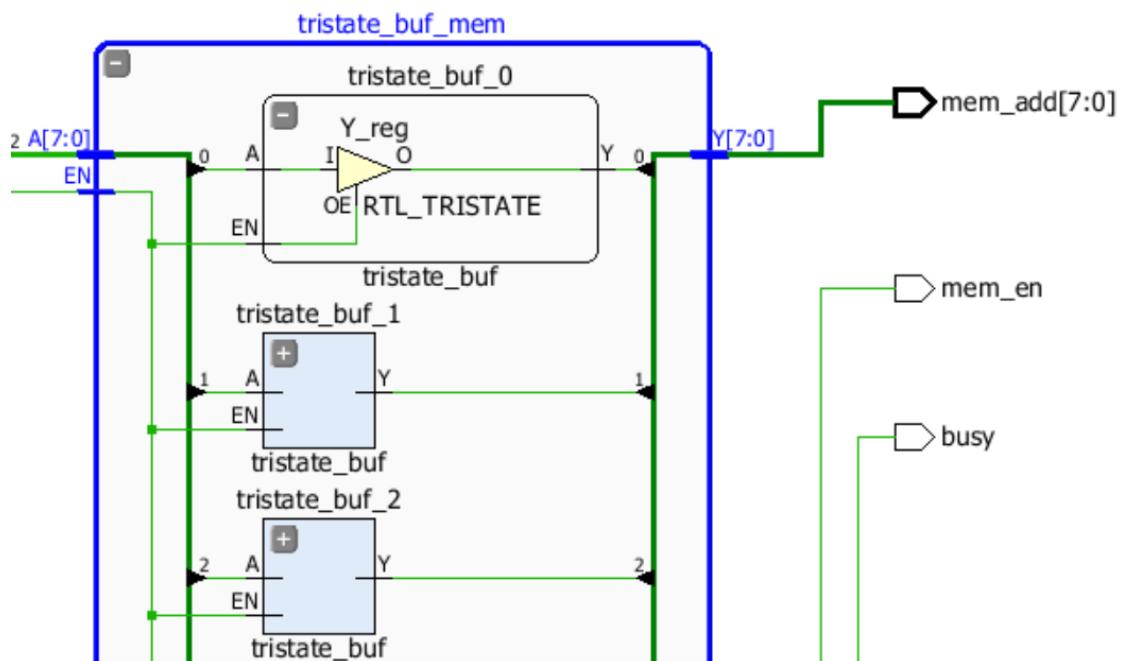


Figure 36 Tristate buffer

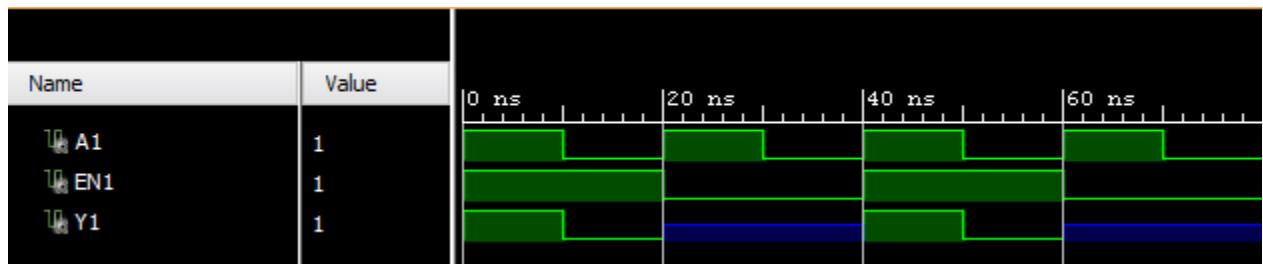


Figure 37 Simulation of tristate buffer

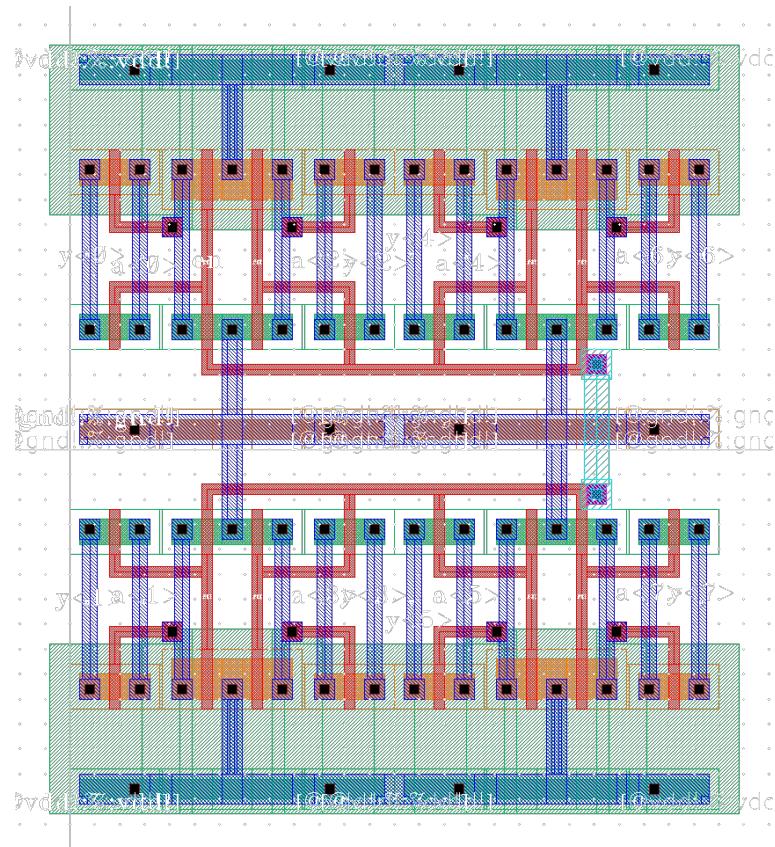


Figure 38 Layout view of 8-bit Tristate buffer

9. Chip

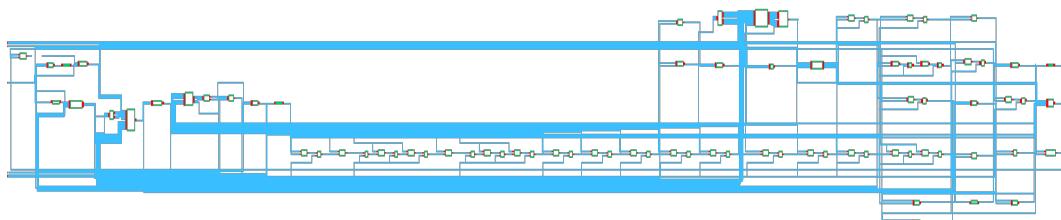


Figure 39 Schematic of chip

The layout of the top module (top most hierarchy), chip is shown in the figure below

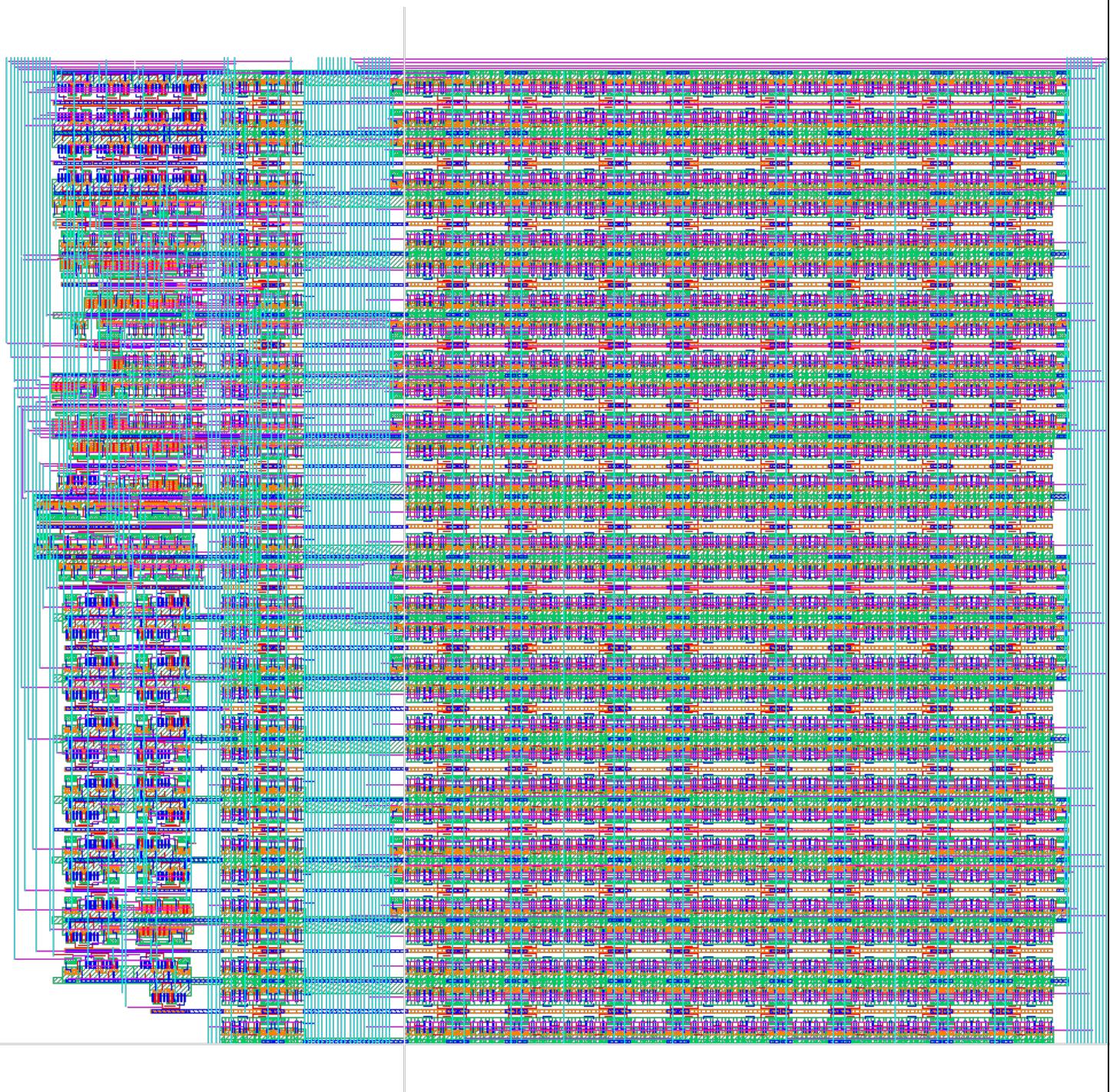


Figure 40 Layout view of the Chip

10. Miscellaneous layout view

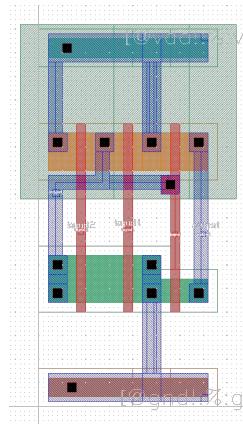


Figure 41 Layout view of *and2*

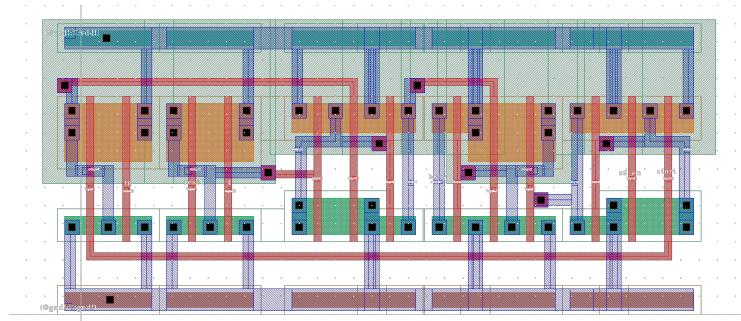


Figure 42 Layout view of *busy1*

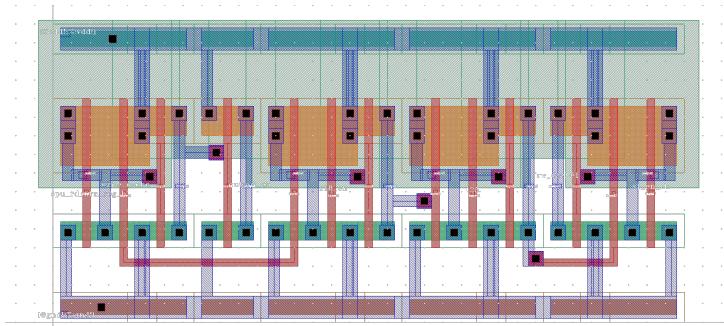


Figure 43 Layout view of *cac_dec_sgn*

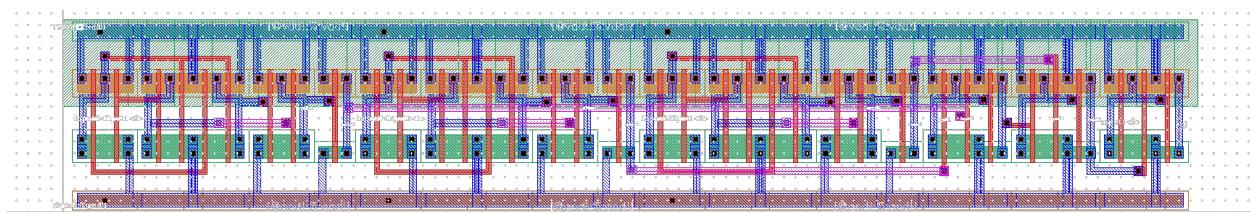


Figure 44 Layout view of *cmprtr*

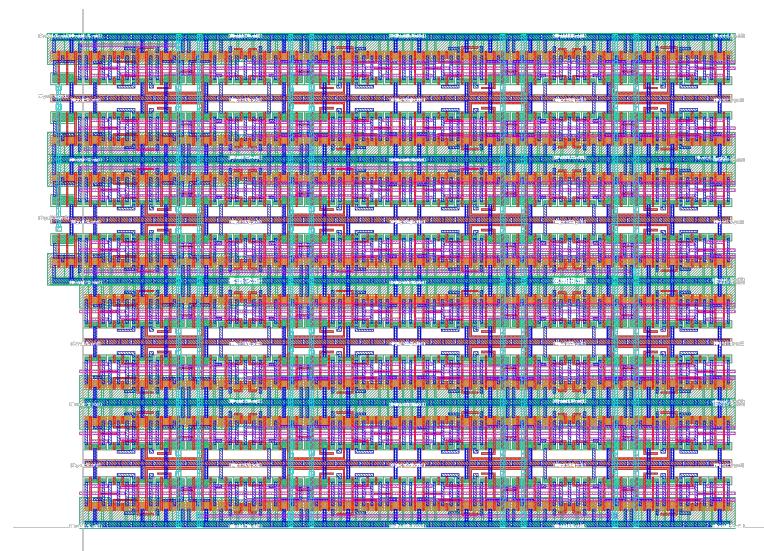


Figure 45 Layout view of *cache_32b*

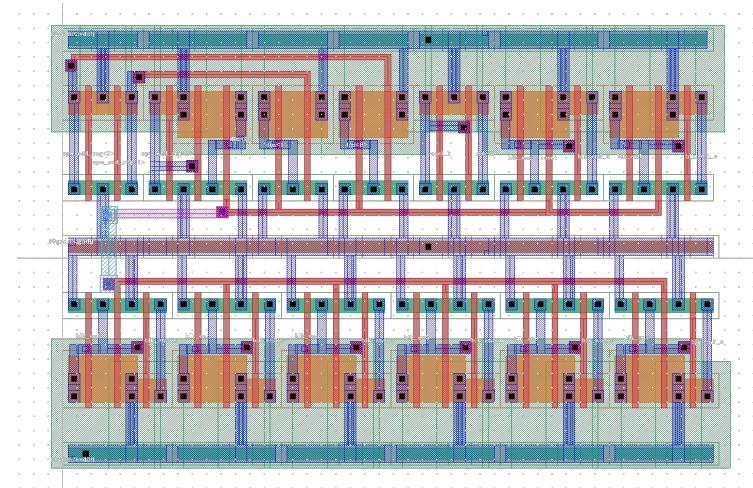


Figure 46 Layout view of *col_sel_tgcm*

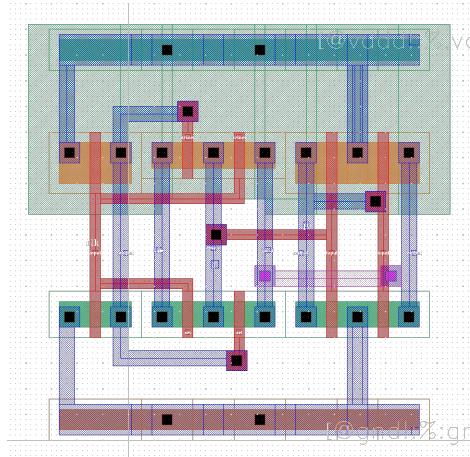


Figure 47 Layout view of *dlatch*

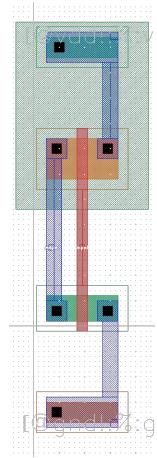


Figure 48 Layout view of inverter

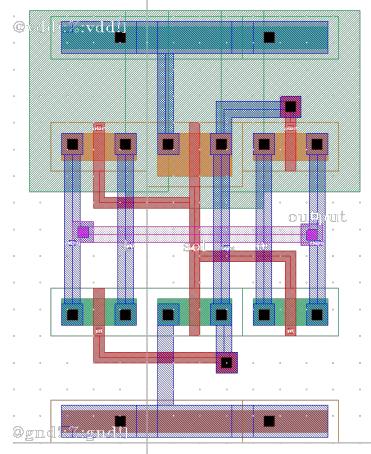


Figure 49 Layout view of mux2x1

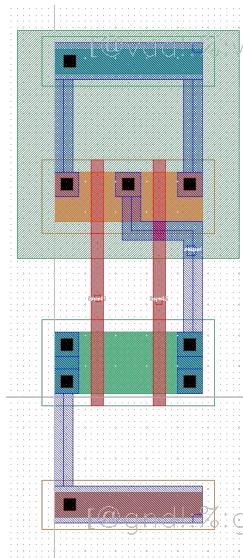


Figure 50 Layout view of nand2

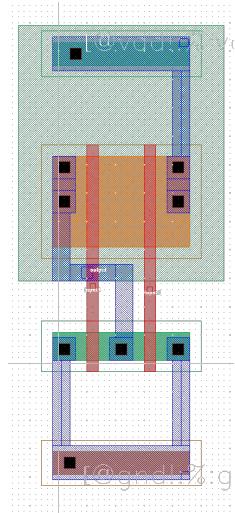


Figure 51 Layout view of nor2

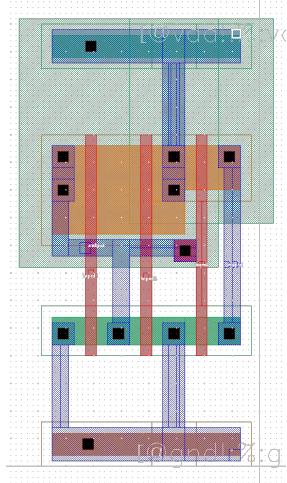


Figure 52 Layout view of or2

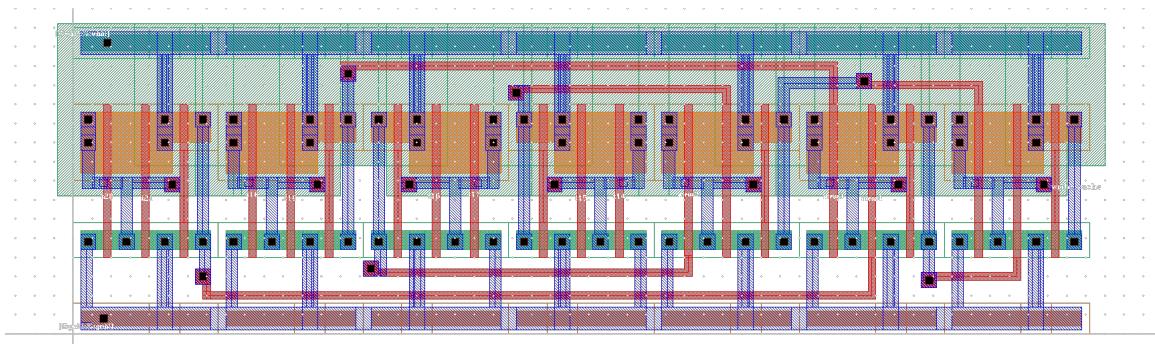


Figure 53 Layout view of tag_cmp1

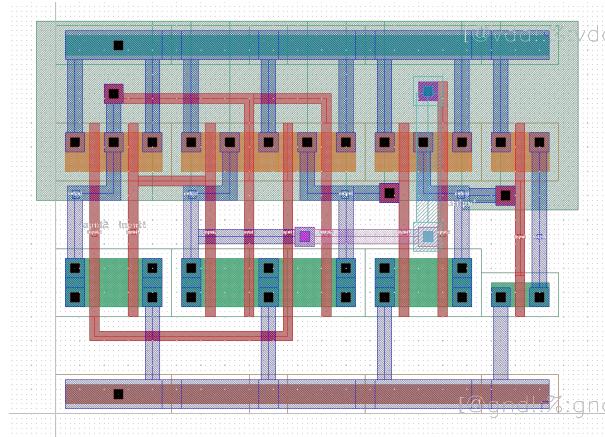


Figure 54 Layout view of xnor2

11. Results

Simulating with the two testbenches provided yield the expected output and match the output waveform given in the project description.

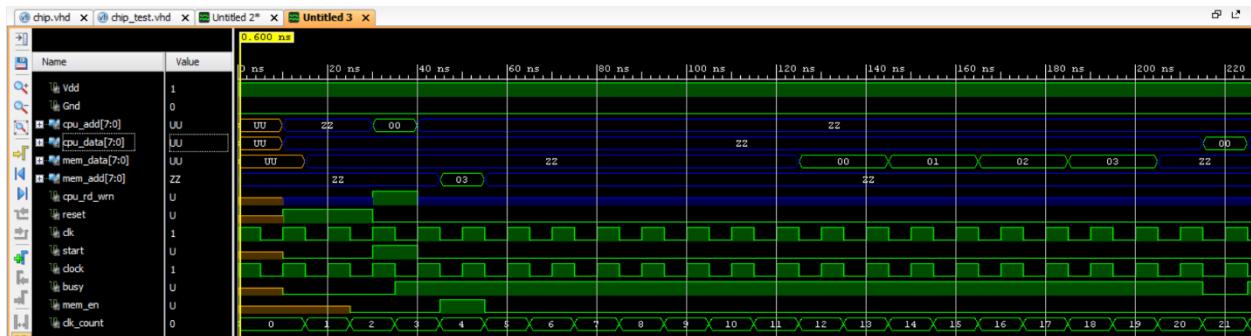


Figure 55 Simulation showing read miss state

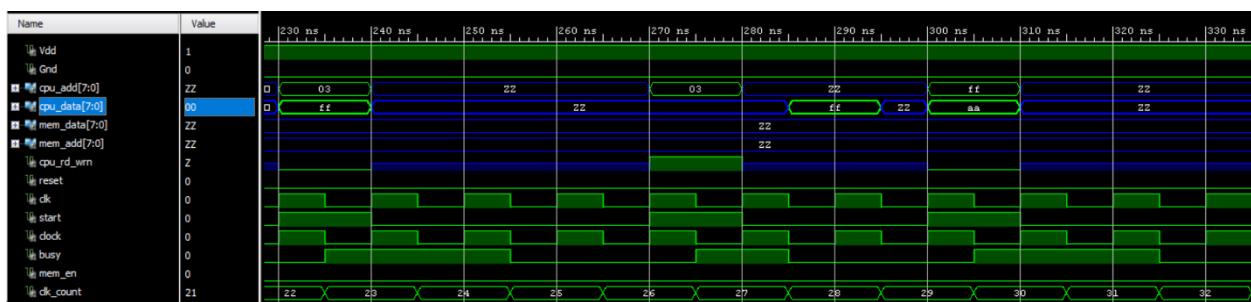


Figure 56 Simulation showing write hit, read hit and write miss

As seen in the above graphs that the busy is turning in the exact time. The output memory enable is also correct. The data that was written in the specified address can also be seen reading in read hit state. So, the code for cache design is working for all test cases.

12. Area of the major modules in the layout

Module Name	Length (μm)	Width (μm)	Area (μm)
buff	24	8.4	201.6
busy1	24	53.1	1274.4
cac_dec_sgn	24	51	1224
cache	703.05	540.9	380279.745
col_sel_tgcm	45.6	68.4	3119.04
dec2_4	24	102	2448
dec3_8	45.6	107.4	4897.44
dff	24	27.45	658.8
dff_8b	45.6	107.4	4897.44
encoder	24	22.8	547.2
hit_miss	45.6	132.6	6046.56
mux8x1	175.2	16.2	2838.24
tag_cmp1	24	79.8	1915.2
tag_valid	703.05	77.25	54310.6125
tristate_buf_8b	45.6	39	1778.4
chip	703.05	786.3	552808.215

*Note: All the module names are exactly the same as in the project

13. Enscript of Code and LVS match

```
library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity and2 is
port(input1 : in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end and2;

architecture structural of and2 is
begin
    output <= input2 and input1;
end structural;
```

```
library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity and3 is
port(input1 : in std_logic;
      input2 : in std_logic;
      input3 : in std_logic;
      output : out std_logic);
end and3;

architecture structural of and3 is

component and2
port(input1 : in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end component;

for all : and2 use entity work.and2(structural);

signal t1: std_logic;

begin
and2_1 : and2 port map(input1, input2, t1);
and2_2 : and2 port map(input3, t1, output);
end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity buff is
port(input : in std_logic;
output : out std_logic);
end buff;

architecture structural of buff is

begin
output<= input;
end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity busy1 is
port(s0_en,start,t6,reset,q : in std_logic;
      busy : out std_logic
     );
end busy1;

architecture structural of busy1 is

component nor2
port(input1 : in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end component;
for all :nor2 use entity work.nor2(structural);

component and2
port(input1 : in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end component;
for all :and2 use entity work.and2(structural);

component or2
port(
      input1    : in std_logic;
      input2 : in std_logic;
      output  : out std_logic);
end component;

for all :or2 use entity work.or2(structural);

signal tbusy2,tbusy1,tbusy3,last_busy : std_logic;

begin

and2_busy1 : and2 port map(s0_en, start, tbusy1);
nor_busy2 : nor2 port map(start,t6, tbusy2);
nor_busy3 : nor2 port map(reset,q,tbusy3);
and2_busy2 : and2 port map(tbusy2, tbusy3,last_busy);
or2_busy3 : or2 port map(tbusy1,last_busy,busy);

end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity cac_dec_sgn is
port(cpu_rd_wrn_reg_b,write_cache,s18_en,t25,s1_en_rh : in std_logic;
      output_en,rw_cache1 :out std_logic);
end cac_dec_sgn;

architecture structural of cac_dec_sgn is

component inverter
port (
  input    : in std_logic;
  output   : out std_logic);
end component;

component or2
port(
  input1   : in std_logic;
  input2 : in std_logic;
  output  : out std_logic);
end component;

for all : inverter use entity work.inverter(structural);
for all : or2 use entity work.or2(structural);

signal cpu_op_en,rw_cache,rw_cache_en : std_logic;

begin

or2_13 :or2 port map(cpu_rd_wrn_reg_b,write_cache,cpu_op_en);
inv_cacheOUT : inverter port map(cpu_op_en,output_en);
or2_14 : or2 port map(write_cache,s18_en,rw_cache);
or2_15 : or2 port map(rw_cache,t25,rw_cache_en);
or2_RH : or2 port map(s1_en_rh,rw_cache_en,rw_cache1);

end structural;
```

```
-- Entity: cache cell 32b
-- Architecture : structural
-- Author: n67
--

library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity cache_32b is
port ( din : in std_logic_vector(7 downto 0);
      by_en0,by_en1,by_en2,by_en3 :in std_logic;
      blk_en : in std_logic;
      rw_en : in std_logic;
      dout : out std_logic_vector(7 downto 0));
end cache_32b;

architecture structural of cache_32b is

component cache_8b
port ( din : in std_logic_vector(7 downto 0);
       en : in std_logic;
       rw_en : in std_logic;
       dout : out std_logic_vector(7 downto 0));
end component;

component and2
port(input1 : in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end component;

for and2_1, and2_2, and2_3, and2_4 : and2 use entity work.and2(structural);
for cache_8b_1, cache_8b_2, cache_8b_3, cache_8b_4 : cache_8b use entity work.cache_8b
(structural);

signal t1,t2,t3,t4: std_logic;
begin

and2_1 : and2 port map(by_en0, blk_en, t1);
and2_2 : and2 port map(by_en1, blk_en, t2);
and2_3 : and2 port map(by_en2, blk_en, t3);
and2_4 : and2 port map(by_en3, blk_en, t4);

cache_8b_1 : cache_8b port map(din,t1,rw_en,dout);
cache_8b_2 : cache_8b port map(din,t2,rw_en,dout);
cache_8b_3 : cache_8b port map(din,t3,rw_en,dout);
cache_8b_4 : cache_8b port map(din,t4,rw_en,dout);

end structural;
```

```
-- Entity: cache cell 8b
-- Architecture : structural
-- Author: n67
--

library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity cache_4b is
port ( din : in std_logic_vector(3 downto 0);
      en : in std_logic;
      rw_en : in std_logic;
      dout : out std_logic_vector(3 downto 0));
end cache_4b;

architecture structural of cache_4b is

component cache_cell
port(      din      : in  std_logic;
            en       : in  std_logic;
            rw_en   : in  std_logic;
            dout    : out std_logic);
end component;

for cache_cell_0, cache_cell_1, cache_cell_2, cache_cell_3 : cache_cell use entity work.cache_cell(structural);

begin

cache_cell_0 : cache_cell port map(din(0),en,rw_en,dout(0));
cache_cell_1 : cache_cell port map(din(1),en,rw_en,dout(1));
cache_cell_2 : cache_cell port map(din(2),en,rw_en,dout(2));
cache_cell_3 : cache_cell port map(din(3),en,rw_en,dout(3));

end structural;
```

```
-- Entity: cache cell 8b
-- Architecture : structural
-- Author: n67
--

library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity cache_8b is
port ( din : in std_logic_vector(7 downto 0);
      en : in std_logic;
      rw_en : in std_logic;
      dout : out std_logic_vector(7 downto 0));
end cache_8b;

architecture structural of cache_8b is

component cache_4b
port ( din : in std_logic_vector(3 downto 0);
      en : in std_logic;
      rw_en : in std_logic;
      dout : out std_logic_vector(3 downto 0));
end component;

for all: cache_4b use entity work.cache_4b(structural);

begin

cache_4b_0 : cache_4b port map(din(3 downto 0),en,rw_en,dout(3 downto 0));
cache_4b_1 : cache_4b port map(din(7 downto 4),en,rw_en,dout(7 downto 4));

--cache_cell_0 : cache_cell port map(din(0),en,rw_en,dout(0));
--cache_cell_1 : cache_cell port map(din(1),en,rw_en,dout(1));
--cache_cell_2 : cache_cell port map(din(2),en,rw_en,dout(2));
--cache_cell_3 : cache_cell port map(din(3),en,rw_en,dout(3));
--cache_cell_4 : cache_cell port map(din(4),en,rw_en,dout(4));
--cache_cell_5 : cache_cell port map(din(5),en,rw_en,dout(5));
--cache_cell_6 : cache_cell port map(din(6),en,rw_en,dout(6));
--cache_cell_7 : cache_cell port map(din(7),en,rw_en,dout(7));

end structural;
```

```
-- Entity: cache cell
-- Architecture : structural
-- Author: n67
--

library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity cache_cell is
port ( din : in std_logic;
      en : in std_logic;
      rw_en : in std_logic;
      dout : out std_logic);
end cache_cell;

architecture structural of cache_cell is

component Dlatch
port ( d : in std_logic;
       clk : in std_logic;
       q : out std_logic;
       qb : out std_logic);
end component;

component tx
port ( sel : in std_logic;
       selnot: in std_logic;
       input : in std_logic;
       output:out std_logic);
end component;

component and2
port(input1 : in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end component;

component inverter
port (
      input : in std_logic;
      output : out std_logic);
end component;

for Dlatch_1 : Dlatch use entity work.Dlatch(structural);
for tx_1 : tx use entity work.tx(structural);
for inv_1,inv2 : inverter use entity work.inverter(structural);
for and2_1, and2_2 : and2 use entity work.and2(structural);

signal r_en, r_enb, w_en,rw_en_inv, temp1, temp2: std_logic;

begin
and2_1 : and2 port map(rw_en, en, r_en);
inv_1 : inverter port map(r_en, r_enb);
inv2 : inverter port map(rw_en,rw_en_inv);
and2_2 : and2 port map(rw_en_inv, en, w_en);
Dlatch_1 : Dlatch port map(din, w_en,temp1,temp2);
tx_1 : tx port map(r_en, r_enb,temp1,dout);
end structural;
```

```
-- Entity: cache cell 32b0
-- Architecture : structural
-- Author: n67
--

library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity cache is
port ( din : in std_logic_vector(7 downto 0);
      by_en0,by_en1,by_en2,by_en3 :in std_logic;
      blk_en0, blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7 : in std_logic
;
      rw_en : in std_logic;
      dout : out std_logic_vector(7 downto 0));
end cache;

architecture structural of cache is

component cache_32b
port ( din : in std_logic_vector(7 downto 0);
      by_en0,by_en1,by_en2,by_en3 :in std_logic;
      blk_en : in std_logic;
      rw_en : in std_logic;
      dout : out std_logic_vector(7 downto 0));
end component;

for cache_32b_0, cache_32b_1, cache_32b_2, cache_32b_3, cache_32b_4, cache_32b_5, cache_32b_6, cache_32b_7 : cache_32b use entity work.cache_32b(structural);

begin

cache_32b_0 : cache_32b port map(din, by_en0,by_en1,by_en2,by_en3, blk_en0,rw_en,dout)
;
cache_32b_1 : cache_32b port map(din, by_en0,by_en1,by_en2,by_en3, blk_en1,rw_en,dout)
;
cache_32b_2 : cache_32b port map(din, by_en0,by_en1,by_en2,by_en3, blk_en2,rw_en,dout)
;
cache_32b_3 : cache_32b port map(din, by_en0,by_en1,by_en2,by_en3, blk_en3,rw_en,dout)
;
cache_32b_4 : cache_32b port map(din, by_en0,by_en1,by_en2,by_en3, blk_en4,rw_en,dout)
;
cache_32b_5 : cache_32b port map(din, by_en0,by_en1,by_en2,by_en3, blk_en5,rw_en,dout)
;
cache_32b_6 : cache_32b port map(din, by_en0,by_en1,by_en2,by_en3, blk_en6,rw_en,dout)
;
cache_32b_7 : cache_32b port map(din, by_en0,by_en1,by_en2,by_en3, blk_en7,rw_en,dout)
;

end structural;
```

```

library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity chip is
port (
    cpu_add : in std_logic_vector(7 downto 0);
    cpu_data : inout std_logic_vector(7 downto 0);
    cpu_rd_wrn : in std_logic;
    start : in std_logic;
    clk : in std_logic;
    reset : in std_logic;
    mem_data : in std_logic_vector(7 downto 0);
    Vdd : in std_logic;
    Gnd : in std_logic;
    busy : out std_logic;
    mem_en : out std_logic;
    mem_addr : out std_logic_vector(7 downto 0));
end chip;

architecture structural of chip is

component nand2
port(input1 : in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end component;
for all :nand2 use entity work.nand2(structural);

--component nor2
--port(input1 : in std_logic;
--      input2 : in std_logic;
--      output : out std_logic);
--end component;
--for all :nor2 use entity work.nor2(structural);

component buff
port(input : in std_logic;
      output : out std_logic);
end component;
for all :buff use entity work.buff(structural);

component tristate_buf_8b
Port ( A : in STD_LOGIC_VECTOR(7 downto 0);
       EN : in STD_LOGIC;
       Y : out STD_LOGIC_VECTOR(7 downto 0));
end component;
for all :tristate_buf_8b use entity work.tristate_buf_8b(structural);

component encoder4_2
port( i1 : in std_logic; --LSB
      i2 : in std_logic;
      i3 : in std_logic;
      o1 : out std_logic; --MSB
      o2 : out std_logic);
end component;
for all :encoder4_2 use entity work.encoder4_2(structural);

component mux2x1
port( A : in std_logic;
      B : in std_logic;
      sel: in std_logic;
      output : out std_logic);
end component;
for all :mux2x1 use entity work.mux2x1(structural);

component mux2x1_8b
port (
    A : in std_logic_vector(7 downto 0);
    B : in std_logic_vector(7 downto 0);
    sel: in std_logic;
    X : out std_logic_vector(7 downto 0));
end component;
for all :mux2x1_8b use entity work.mux2x1_8b(structural);

--component xor2
--port(
--  input1 : in std_logic;
--  input2 : in std_logic;
--  output : out std_logic);
--end component;
--for all :xor2 use entity work.xor2(structural);

-----
--State Machine Init -----
component and2
port(input1 : in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end component;

component dff
port ( d : in std_logic;
        clk : in std_logic;
        q : out std_logic);
end component;

component inverter
port (
    input : in std_logic;
    output : out std_logic);
end component;

component or2
port(
    input1 : in std_logic;
    input2 : in std_logic;
    output : out std_logic);
end component;

for all : dff use entity work.dff(structural);
for all : and2 use entity work.and2(structural);
for all : inverter use entity work.inverter(structural);
for all : or2 use entity work.or2(structural);

signal t1,t2,q,t3,start_b,r_m_st,s0_en,s1_en,s2_en,s3_en,s4_en,s5_en,s6_en,s7_en,s8_en,
s9_en,s10_en,s11_en,s12_en,s13_en,s14_en,s15_en,s16_en,s17_en,s18_en:std_logic;
signal r_h_st,s1_en_rh,w_m_st,s1_en_wm,s2_en_wm,w_h_st,s1_en_wh,s2_en_wh,t4,t5,t6,s2_en_rh,s3_en_wm,s3_en_wh:std_logic;
signal reset_b,t7,t8,t9,t10,t11,t12,t13,t14,t15,t16,t17,t18,t19,t20,t21,t22,t23,t24,t25 : std_logic;
-----State Machine Init end -----

```

```

-----Registers-----
component dff_8b
port ( d : in std_logic_vector(7 downto 0);
      clk : in std_logic;
      q : out std_logic_vector(7 downto 0));
end component;

for all: dff_8b use entity work.dff_8b(structural);

-----Registers END-----
-----tie high/Low-----
-----tie High/Low END-----
-----cache-----
component cache
port ( din : in std_logic_vector(7 downto 0);
       by_en0,by_en1,by_en2,by_en3 :in std_logic;
       blk_en0,blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7 : in std_logic
      ;
       rw_en : in std_logic;
       dout : out std_logic_vector(7 downto 0));
end component;

for all : cache use entity work.cache(structural);

component tag_valid
port ( din : in std_logic_vector(3 downto 0);
       blk_en0,blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7 : in std_logic
      ;
       rw_en : in std_logic;
       dout : out std_logic_vector(3 downto 0));
end component;

for all : tag_valid use entity work.tag_valid(structural);

component dec2_4
port(s0,s1,en : in std_logic;
y0,y1,y2,y3 : out std_logic);
end component;

for all : dec2_4 use entity work.dec2_4(structural);

component dec3_8
port(s0,s1,s2 : in std_logic;
y0,y1,y2,y3,y4,y5,y6,y7 : out std_logic);
end component;

for all : dec3_8 use entity work.dec3_8(structural);

component hit_miss
port (
  cache_tag_valid : in std_logic_vector(3 downto 0);
  cpu_tag : in std_logic_vector(2 downto 0);
  cpu_rw : in std_logic;
  sm_en : in std_logic;
  w_m,r_m,w_h,r_h : out std_logic);
end component;

for all : hit_miss use entity work.hit_miss(structural);

component busy1
port(s0_en,start,t6,reset,q : in std_logic;
      busy : out std_logic
     );
end component;
for all : busy1 use entity work.busy1(structural);

component tag_cmp1
port(t14,t15,t16,t17,t18,t19,t20,t21: in std_logic;
      mem2,mem3,mem4 : inout std_logic;
      write_cache : out std_logic
     );
end component;
for all : tag_cmp1 use entity work.tag_cmp1(structural);

component col_sel_tgcm
Port ( blk_en0, blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7,reset,reset_b: in std_logic;
       cpu_add_reg : in std_logic_vector(2 downto 0);
       din : out std_logic_vector(3 downto 0);
       blk_en0_o,blk_en1_o,blk_en2_o,blk_en3_o,blk_en4_o,blk_en5_o,blk_en6_o,blk_en7_o : out std_logic
      );
end component;
for all : col_sel_tgcm use entity work.col_sel_tgcm(structural);

component cac_dec_sgn
port(cpu_rd_wrn_reg_b,write_cache,s18_en,t25,s1_en_rh : in std_logic;
      output_en,rw_cachel :out std_logic
     );
end component;
for all : cac_dec_sgn use entity work.cac_dec_sgn(structural);

-----signals-----
signal blk_en0, blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7:std_logic;
signal blk_en0_o,blk_en1_o,blk_en2_o,blk_en3_o,blk_en4_o,blk_en5_o,blk_en6_o,blk_en7_o : std_logic; --,blk1_en0,blk1_en1,blk1_en2,blk1_en3,blk1_en4,blk1_en5,blk1_en6,blk1_en7
signal cpu_rd_wrn_reg_b,cpu_rd_wrn_reg,tag_en,tag_en0,tag_en1 : std_logic;
signal cpu_rd_wrn_reg_when_RST,tag2,tag1,tag0: std_logic;
signal w_m,r_m,w_h,r_h : std_logic;
signal cpu_add_reg,cpu_data_reg :std_logic_vector(7 downto 0);
signal hi,lo:std_logic;
signal feedback,t26:std_logic;
--read from mem
signal mem1,mem2,mem3,mem4,b0,b1:std_logic;
--write mem to cache
signal write_cachel,write_cache0,write_cache,write_cache_2,write_cache_2_b :std_logic;
signal byte_select : std_logic_vector(1 downto 0);
signal by_en0,by_en1,by_en2,by_en3 :std_logic;
signal din_cache : std_logic_vector(7 downto 0); --data input for cache
--cache block
signal s_b,s,cpu_op_en,rw_cache,rw_cache_en,rw_cachel,output_en : std_logic;
signal dout_cache : std_logic_vector(7 downto 0);
--busy
signal tbusy1,tbusy2,last_busy,tbusy3: std_logic;

```

```

--  

----temp assigns for test  

signal din,dout : std_logic_vector(3 downto 0);  

---  

begin  

-----State Machine processing-----  

inv_1 : inverter port map(start,start_b);  

inv_2 : inverter port map(reset,reset_b);  

or2_1 : or2 port map(t1,feedback,t2);  

and2_2 : and2 port map(q,start_b,t1);  

dff_1 : dff port map(t2,clk,q);  

and2_3 : and2 port map(q,start, t3);  

dff_2 : dff port map(t3,clk,s0_en);  

--Read Miss--  

and2_4 : and2 port map(r_m,s0_en,r_m_st);  

dff_3 : dff port map(r_m_st,clk,t7);  

and2_8 : and2 port map(reset_b,t7,s1_en);  

dff_4 : dff port map(s1_en,clk,t8);  

and2_9 : and2 port map(reset_b,t8,s2_en);  

dff_5 : dff port map(s2_en,clk,t9);  

and2_10 : and2 port map(reset_b,t9, s3_en);  

dff_6 : dff port map(s3_en,clk,t10);  

and2_11 : and2 port map(reset_b, t10, s4_en);  

dff_7 : dff port map(s4_en,clk,t11);  

and2_12 : and2 port map(reset_b,t11, s5_en);  

dff_8 : dff port map(s5_en,clk,t12);  

and2_13 : and2 port map(reset_b,t12, s6_en);  

dff_9 : dff port map(s6_en,clk,t13);  

and2_14 : and2 port map(reset_b,t13, s7_en);  

dff_10 : dff port map(s7_en,clk,t14);  

and2_15 : and2 port map(reset_b,t14, s8_en);  

dff_11 : dff port map(s8_en,clk,t15); -----  

and2_16 : and2 port map(reset_b,t15, s9_en);  

dff_12 : dff port map(s9_en,clk,t16);  

and2_17 : and2 port map(reset_b,t16, s10_en);  

dff_13 : dff port map(s10_en,clk,t17);  

and2_18 : and2 port map(reset_b,t17, s11_en);  

dff_14 : dff port map(s11_en,clk,t18);  

and2_19 : and2 port map(reset_b,t18, s12_en);  

dff_15 : dff port map(s12_en,clk,t19);  

and2_20 : and2 port map(reset_b,t19, s13_en);  

dff_16 : dff port map(s13_en,clk,t20);  

and2_21 : and2 port map(reset_b,t20, s14_en);  

dff_17 : dff port map(s14_en,clk,t21);  

and2_22 : and2 port map(reset_b,t21, s15_en);  

dff_18 : dff port map(s15_en,clk,t22);  

and2_23 : and2 port map(reset_b,t22, s16_en);  

dff_19 : dff port map(s16_en,clk,t23);  

and2_24 : and2 port map(reset_b,t23, s17_en);  

dff_20 : dff port map(s17_en,clk,s18_en);  

--read hit--  

and2_5 : and2 port map(s0_en,r_h,r_h_st);  

dff_22 : dff port map(r_h_st,clk,s1_en_rh);  

--write miss--
```

```

and2_6 : and2 port map(s0_en,w_m,w_m_st);  

dff_23 : dff port map(w_m_st, clk,t24);  

and2_25 : and2 port map(reset_b,t24,s1_en_wm);  

dff_24 : dff port map(s1_en_wm,clk,s2_en_wm);  

--write hit--  

and2_7 : and2 port map(s0_en,w_h,w_h_st);  

dff_25 : dff port map(w_h_st, clk,t25);  

and2_26 : and2 port map(reset_b,t25,s1_en_wh);  

dff_26 : dff port map(s1_en_wh, clk,s2_en_wh);  

--feedback--  

or2_2 : or2 port map(s18_en,s1_en_rh,t4);  

or2_3 : or2 port map(s2_en_wm,s2_en_wh,t5);  

or2_4 : or2 port map(t4,t5,t6);  

or2_feedback : or2 port map(t6, reset, feedback);  

-----State Machine processing end-----  

-----  

-----Latch cpu addr,data and rw_sig-----  

dff_rw_register : dff port map(cpu_rd_wrn,start_b,cpu_rd_wrn_reg);  

dff_8b_address_register : dff_8b port map (cpu_add,start_b,cpu_add_reg);  

dff_8b_data_register : dff_8b port map(cpu_data,start_b,cpu_data_reg);  

-----tag compare-----  

dec3_8_1 : dec3_8 port map (cpu_add_reg(4),cpu_add_reg(3),cpu_add_reg(2),blk_en1, blk_en0,blk_en3,blk_en2,blk_en5,blk_en4,blk_en7,blk_en6);  

-----  

--or2_8 : or2 port map(blk_en0,reset,blk_en0_o);  

--or2_9 : or2 port map(blk_en1,reset,blk_en1_o);  

--or2_10 : or2 port map(blk_en2,reset,blk_en2_o);  

--or2_11 : or2 port map(blk_en3,reset,blk_en3_o);  

--or2_12 : or2 port map(blk_en4,reset,blk_en4_o);  

--or2_5 : or2 port map(blk_en5,reset,blk_en5_o);  

--or2_6 : or2 port map(blk_en6,reset,blk_en6_o);  

--or2_7 : or2 port map(blk_en7,reset,blk_en7_o);  

--inv_13 : inverter port map(cpu_add_reg(7), tag2);  

--inv_14 : inverter port map(cpu_add_reg(6), tag1);  

--inv_15 : inverter port map(cpu_add_reg(5), tag0);  

--nor2_1 : nor2 port map(reset,tag2, din(2));  

--nor2_2 : nor2 port map(reset,tag1, din(1));  

--nor2_3 : nor2 port map(reset,tag0, din(0));  

--buff_1 : buff port map(reset_b,din(3));  

col_sel_tgcm_1 : col_sel_tgcm port map(blk_en0, blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7,reset,reset_b,cpu_add_reg(7 downto 5),din(3 downto 0),blk_en0_o,blk_en1_o,blk_en2_o,blk_en3_o,blk_en4_o,blk_en5_o,blk_en6_o,blk_en7_o);  

inv_12 : inverter port map(cpu_rd_wrn_reg,cpu_rd_wrn_reg_b);  

nand2_rwtag : nand2 port map(reset,rw_cache1,tag_en0);  

or2_rwtag : or2 port map(reset, rw_cache1,tag_en1);
```

```
nand2_rwtags : nand2 port map(tag_en0,tag_en1,tag_en);
tag_valid_1 : tag_valid port map(din,blk_en0_o,blk_en1_o,blk_en2_o,blk_en3_o,blk_en4_o
,blk_en5_o,blk_en6_o,blk_en7_o,tag_en,dout);
hit_miss_1 : hit_miss port map(dout,cpu_add_reg(7 downto 5),cpu_rd_wrn_reg,s0_en,w_m,r
_m,w_h,r_h);

-----memory enable -----
tristate_buf_mem : tristate_buf_8b port map(A(7 downto 2)=>cpu_add_reg(7 downto 2),A(1
)=>Gnd, A(0)=>Gnd
, EN=>t7, Y=>mem_add);
buf_mem : buff port map(t7,mem_en);
-----Column Select-----
--or2_mem1 : or2 port map(t14,t15,mem1);
--or2_mem2 : or2 port map(t17,t16,mem2);
--or2_mem3 : or2 port map(t18,t19,mem3);
--or2_mem4 : or2 port map(t20,t21,mem4);

--or2_byte0 : or2 port map(mem2,mem1,write_cache0);
--or2_byte1 : or2 port map(mem3,mem4,write_cache1);
--or2_byte : or2 port map(write_cache0, write_cache1,write_cache); --or2_cacheRow2 tem
pwcache

tag_cmp1_1 : tag_cmp1 port map(t14,t15,t16,t17,t18,t19,t20,t21,mem2,mem3,mem4,write_ca
che);
encoder4_2_mem : encoder4_2 port map(mem2, mem3, mem4, b0,b1);

mux2x1_cache0 : mux2x1 port map(b0,cpu_add_reg(1),write_cache,byte_select(0));
mux2x1_cache1 : mux2x1 port map(b1,cpu_add_reg(0),write_cache,byte_select(1));

dec2_4_byte : dec2_4 port map(byte_select(0),byte_select(1),rw_cache1,by_en0,by_en1,by
_en2,by_en3); --write_cache instead dec2_4_en
-----Write to Cache-----
or2_cache_write_en : or2 port map(write_cache,t22, write_cache_2);
inv_cache : inverter port map(write_cache_2,write_cache_2_b);
mux2x1_8b_cache_write : mux2x1_8b port map(cpu_data_reg,mem_data,write_cache_2_b,din_c
ache);
-----cache-----
--or2_13 :or2 port map(cpu_rd_wrn_reg_b,write_cache,cpu_op_en);
--inv_cacheOUT : inverter port map(cpu_op_en,output_en);
--or2_14 : or2 port map(write_cache,s18_en,rw_cache);
--or2_15 : or2 port map(rw_cache,t25,rw_cache_en);
--or2_RH : or2 port map(s1_en_rh,rw_cache_en,rw_cache1);

cac_dec_sgn_1 : cac_dec_sgn port map(cpu_rd_wrn_reg_b,write_cache,s18_en,t25,s1_en_rh,
output_en,rw_cache1);

cache_cache : cache port map(din_cache,by_en0,by_en1,by_en2,by_en3,blk_en0, blk_en1,bl
k_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7,output_en,dout_cache);
-----Write to CPU-----
tristate_buf_cpu : tristate_buf_8b port map (dout_cache,t4,cpu_data);
-----BUSY-----
--
--and2_busy1 : and2 port map(s0_en, start, tbusyl);
```

```
--  
-- Entity: cmpptr2  
-- Architecture : structural  
-- Author: n67  
--  
library STD;  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity cmpptr2 is  
    port (  
        input1  : in  std_logic_vector(3 downto 0);  
        input2  : in  std_logic_vector(2 downto 0);  
        eql    : out std_logic);  
end cmpptr2;  
  
architecture structural of cmpptr2 is  
  
component xnor2  
    port (  
        input1  : in  std_logic;  
        input2  : in  std_logic;  
        output  : out std_logic);  
end component;  
  
component and3  
port (  
    input1  : in  std_logic;  
    input2  : in  std_logic;  
    input3  : in  std_logic;  
    output  : out std_logic);  
end component;  
  
component and2  
port (  
    input1  : in  std_logic;  
    input2  : in  std_logic;  
    output  : out std_logic);  
end component;  
  
for and3_1 : and3 use entity work.and3(structural);  
for xnor2_1,xnor2_2,xnor2_3 : xnor2 use entity work.xnor2(structural);  
for and2_1 : and2 use entity work.and2(structural);  
  
signal t1,t2,t3,t4 : std_logic;  
  
begin  
  
    xnor2_1 : xnor2 port map (input1(0),input2(0),t1);  
    xnor2_2 : xnor2 port map (input1(1),input2(1),t2);  
    xnor2_3 : xnor2 port map (input1(2),input2(2),t3);  
    and3_1 : and3 port map (t1,t2,t3,t4);  
    and2_1 : and2 port map (t4, input1(3), eql);  
  
end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity col_sel_tgcm is
Port ( blk_en0, blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7,reset,reset_b:
in std_logic;
      cpu_add_reg : in std_logic_vector(2 downto 0);
      din : out std_logic_vector(3 downto 0);
      blk_en0_o,blk_en1_o,blk_en2_o,blk_en3_o,blk_en4_o,blk_en5_o,blk_en6_o,blk_en7_o
 : out std_logic
      );
end col_sel_tgcm;

architecture structural of col_sel_tgcm is

component inverter
port (
      input      : in  std_logic;
      output     : out std_logic);
end component;

component or2
port(
      input1    : in  std_logic;
      input2    : in  std_logic;
      output    : out std_logic);
end component;

for all : inverter use entity work.inverter(structural);
for all : or2 use entity work.or2(structural);

component buff
port(input : in std_logic;
      output : out std_logic);
end component;
for all :buff use entity work.buff(structural);

component nor2
port(input1 : in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end component;
for all :nor2 use entity work.nor2(structural);

signal tag0,tag1,tag2 : std_logic;

begin

or2_8 : or2 port map(blk_en0,reset,blk_en0_o);
or2_9 : or2 port map(blk_en1,reset,blk_en1_o);
or2_10 : or2 port map(blk_en2,reset,blk_en2_o);
or2_11 : or2 port map(blk_en3,reset,blk_en3_o);
or2_12 : or2 port map(blk_en4,reset,blk_en4_o);
or2_5 : or2 port map(blk_en5,reset,blk_en5_o);
or2_6 : or2 port map(blk_en6,reset,blk_en6_o);
or2_7 : or2 port map(blk_en7,reset,blk_en7_o);

inv_13 : inverter port map(cpu_add_reg(2), tag2);
inv_14 : inverter port map(cpu_add_reg(1), tag1);
inv_15 : inverter port map(cpu_add_reg(0), tag0);

nor2_1 : nor2 port map(reset,tag2, din(2));
nor2_2 : nor2 port map(reset,tag1, din(1));
nor2_3 : nor2 port map(reset,tag0, din(0));
buff_1 : buff port map(reset_b,din(3));

end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity dec2_4 is
port(s0,s1,en : in std_logic;
y0,y1,y2,y3 : out std_logic);
end dec2_4;

architecture structural of dec2_4 is

component and3
port(input1 : in std_logic;
      input2 : in std_logic;
      input3 : in std_logic;
      output : out std_logic);
end component;

component inverter
port(input : in std_logic;
      output : out std_logic);
end component;

for inv_1,inv_2 : inverter use entity work.inverter(structural);
for and3_1, and3_2, and3_3, and3_4 : and3 use entity work.and3(structural);

signal s0_b, s1_b : std_logic;

begin

inv_1 : inverter port map(input=>s0, output=>s0_b);
inv_2 : inverter port map(input=>s1, output=>s1_b);

and3_1 : and3 port map(input1=>s0_b, input2=>s1_b, input3=>en, output=>y0);
and3_2 : and3 port map(input1=>s0_b, input2=>s1, input3=>en, output=>y1);
and3_3 : and3 port map(input1=>s0, input2=>s1_b, input3=>en, output=>y2);
and3_4 : and3 port map(input1=>s0, input2=>s1, input3=>en, output=>y3);

end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity dec3_8 is
port(s0,s1,s2: in std_logic;
y0,y1,y2,y3,y4,y5,y6,y7 : out std_logic);
end dec3_8;

architecture structural of dec3_8 is

component dec2_4
port(s0,s1,en : in std_logic;
y0,y1,y2,y3 : out std_logic);
end component;

--component and2
--port(input1 : in std_logic;
--      input2 : in std_logic;
--      output : out std_logic);
--end component;

component inverter
port (
      input   : in  std_logic;
      output  : out std_logic);
end component;

for dec2_4_1,dec2_4_2 : dec2_4 use entity work.dec2_4(structural);
--for and2_1, and2_2 : and2 use entity work.and2(structural);
for inv_1 : inverter use entity work.inverter(structural);

signal s2b, en1,en2 :std_logic;

begin
inv_1 :inverter port map(s2, s2b);
--and2_1 :and2 port map(s2b, en, en1);
--and2_2 :and2 port map(s2, en, en2);

dec2_4_1 : dec2_4 port map(s0=>s0,s1=>s1,en=>s2,y0=>y0,y1=>y2,y2=>y4,y3=>y6);
dec2_4_2 : dec2_4 port map(s0=>s0,s1=>s1,en=>s2b,y0=>y1,y1=>y3,y2=>y5,y3=>y7);

end structural;
```

```
--  
-- Entity: dff  
-- Architecture : structural  
-- Author: n67  
--  
  
library STD;  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity dff_8b is  
  port ( d : in std_logic_vector(7 downto 0);  
         clk : in std_logic;  
         q : out std_logic_vector(7 downto 0));  
end dff_8b;  
  
architecture structural of dff_8b is  
  
component dff  
  port ( d : in std_logic;  
         clk : in std_logic;  
         q : out std_logic);  
end component;  
  
for all : dff use entity work.dff(structural);  
  
begin  
  
dff_0 : dff port map(d(0),clk,q(0));  
dff_1 : dff port map(d(1),clk,q(1));  
dff_2 : dff port map(d(2),clk,q(2));  
dff_3 : dff port map(d(3),clk,q(3));  
dff_4 : dff port map(d(4),clk,q(4));  
dff_5 : dff port map(d(5),clk,q(5));  
dff_6 : dff port map(d(6),clk,q(6));  
dff_7 : dff port map(d(7),clk,q(7));  
  
end structural;
```

```
--  
-- Entity: dff  
-- Architecture : structural  
-- Author: n67  
--  
  
library STD;  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity dff is  
  port ( d : in std_logic;  
        clk : in std_logic;  
        q : out std_logic);  
end dff;  
  
architecture structural of dff is  
begin  
  
  output: process  
  
  begin  
    wait until ( clk'EVENT and clk = '0' );  
    q <= d;  
  end process output;  
  
end structural;
```

```
-- Entity: dlatch
-- Architecture : structural
-- Author: n67
--

library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity Dlatch is
  port ( d  : in std_logic;
         clk : in std_logic;
         q   : out std_logic;
         qb  : out std_logic);
end Dlatch;

architecture structural of Dlatch is

begin

  output: process (d,clk)
  begin
    if clk = '1' then
      q <= d;
      qb <= not d;
    end if;
  end process output;
end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity encoder4_2 is
port( i1 : in std_logic; --LSB
      i2 : in std_logic;
      i3 : in std_logic;
      o1 : out std_logic; --MSB
      o2 : out std_logic);
end encoder4_2;

architecture structural of encoder4_2 is

component or2
port(input1  : in std_logic;
      input2  : in std_logic;
      output   : out std_logic);
end component;
for all : or2 use entity work.or2(structural);
begin
or2_1 : or2 port map(i3,i2,o1);
or2_2 : or2 port map(i1,i3,o2);
end structural;
```

```
--  
-- Entity: hit_miss  
-- Architecture : structural  
-- Author: n67  
--  
library STD;  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity hit_miss is  
    port (  
        cache_tag_valid : in std_logic_vector(3 downto 0);  
        cpu_tag : in std_logic_vector(2 downto 0);  
        cpu_rw : in std_logic;  
        sm_en : in std_logic;  
        w_m,r_m,w_h,r_h : out std_logic);  
end hit_miss;  
  
architecture structural of hit_miss is  
  
component cmprtr2  
    port (  
        input1 : in std_logic_vector(3 downto 0);  
        input2 : in std_logic_vector(2 downto 0);  
        eql : out std_logic);  
end component;  
  
component dec2_4  
    port(s0,s1,en : in std_logic;  
        y0,y1,y2,y3 : out std_logic);  
end component;  
  
for cmprtr2_1 : cmprtr2 use entity work.cmprtr2(structural);  
for dec2_4_1 : dec2_4 use entity work.dec2_4(structural);  
  
signal hit_or_miss : std_logic;  
  
begin  
  
cmprtr2_1 : cmprtr2 port map(cache_tag_valid, cpu_tag, hit_or_miss);  
dec2_4_1 : dec2_4 port map(hit_or_miss, cpu_rw, sm_en,w_m,r_m,w_h,r_h);  
  
end structural;
```

```
library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity inverter is

    port (
        input   : in std_logic;
        output  : out std_logic);
end inverter;

architecture structural of inverter is

begin

    output <= not (input);

end structural;
```

```
--  
-- Entity: mux2_1b  
-- Architecture : structural  
-- Author: n67  
-- Created On: 11/1/17 at 22:54  
--  
library STD;  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity mux2x1_8b is  
    port (  
        A : in std_logic_vector(7 downto 0);  
        B : in std_logic_vector(7 downto 0);  
        sel: in std_logic;  
        X : out std_logic_vector(7 downto 0));  
end mux2x1_8b;  
  
architecture structural of mux2x1_8b is  
  
component mux2x1  
    port( A : in std_logic;  
          B : in std_logic;  
          sel: in std_logic;  
          output : out std_logic);  
end component;  
  
for mux2x1_1,mux2x1_2,mux2x1_3,mux2x1_4,mux2x1_5,mux2x1_6,mux2x1_7,mux2x1_8 : mux2x1 u  
se entity work.mux2x1(structural);  
  
begin  
  
    mux2x1_1 : mux2x1 port map (A(0),B(0),sel,X(0));  
    mux2x1_2 : mux2x1 port map (A(1),B(1),sel,X(1));  
    mux2x1_3 : mux2x1 port map (A(2),B(2),sel,X(2));  
    mux2x1_4 : mux2x1 port map (A(3),B(3),sel,X(3));  
    mux2x1_5 : mux2x1 port map (A(4),B(4),sel,X(4));  
    mux2x1_6 : mux2x1 port map (A(5),B(5),sel,X(5));  
    mux2x1_7 : mux2x1 port map (A(6),B(6),sel,X(6));  
    mux2x1_8 : mux2x1 port map (A(7),B(7),sel,X(7));  
  
end structural;
```

```
--  
-- Entity: mux2_1b  
-- Architecture : structural  
-- Author: n67  
-- Created On: 11/1/17 at 22:54  
--  
library STD;  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity mux2x1 is  
  
port (  
    A : in std_logic;  
    B : in std_logic;  
    sel: in std_logic;  
    output : out std_logic  
)  
end mux2x1;  
  
architecture structural of mux2x1 is  
  
    component tx  
    port (    sel: in std_logic;  
              selnot: in std_logic;  
              input : in std_logic;  
              output:out std_logic);  
    end component;  
  
    component inverter  
    port( input : in std_logic;  
          output : out std_logic);  
    end component;  
  
--for nand2_1,nand2_2,nand2_3 : nand2 use entity work.nand2(structural);  
for inv_1 : inverter use entity work.inverter(structural);  
for all : tx use entity work.tx(structural);  
signal t1,t2,t3: std_logic;  
  
begin  
  
inv_1 : inverter port map(sel,t1);  
tx_1 : tx port map(sel,t1,A,output);  
tx_2 : tx port map(t1,sel,B,output);  
  
-- nand2_1 : nand2 port map(A,sel,t2);  
-- nand2_2 : nand2 port map(B,t1,t3);  
-- nand2_3 : nand2 port map(t2,t3,output);  
  
end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity nand2 is
port(input1 :in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end nand2;

architecture structural of nand2 is
begin
output <= input1 nand input2;
end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity nor2 is
port(input1 :in std_logic;
      input2 : in std_logic;
      output : out std_logic);
end nor2;

architecture structural of nor2 is
begin
output <= input1 nor input2;
end structural;
```

```
library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity or2 is

port (
    input1  : in std_logic;
    input2 : in std_logic;
    output  : out std_logic);
end or2;

architecture structural of or2 is

begin

    output <= input1  or input2;

end structural;
```

```
--  
-- Entity: state machine  
-- Architecture : structural  
-- Author: n67  
library STD;  
library IEEE;  
use IEEE.std_logic_1164.all;  
  
entity state_machine is  
port(clk : in std_logic;  
      reset : in std_logic;  
      start : in std_logic;  
      r_m,r_h,w_m,w_h : in std_logic  
--      state0: out std_logic  
      );  
end state_machine;  
  
architecture structural of state_machine is  
  
component and2  
port(input1 : in std_logic;  
      input2 : in std_logic;  
      output : out std_logic);  
end component;  
  
component dff  
port ( d : in std_logic;  
      clk : in std_logic;  
      q : out std_logic);  
end component;  
  
component inverter  
port (  
      input : in std_logic;  
      output : out std_logic);  
end component;  
  
component or2  
port(  
      input1 : in std_logic;  
      input2 : in std_logic;  
      output : out std_logic);  
end component;  
  
for all : dff use entity work.dff(structural);  
for all : and2 use entity work.and2(structural);  
for all : inverter use entity work.inverter(structural);  
for all : or2 use entity work.or2(structural);  
  
signal t1,t2,q,t3,start_b,r_m_st,s0_en,s1_en,s2_en,s3_en,s4_en,s5_en,s6_en,s7_en,s8_en  
,s9_en,s10_en,s11_en,s12_en,s13_en,s14_en,s15_en,s16_en,s17_en,s18_en:std_logic;  
signal r_h_st,s1_en_rh,w_m_st,s1_en_wm,s2_en_wm,w_h_st,s1_en_wh,s2_en_wh,t4,t5,t6,s2_e  
n_rh,s3_en_wm,s3_en_wh:std_logic;  
signal reset_b,t7,t8,t9,t10,t11,t12,t13,t14,t15,t16,t17,t18,t19,t20,t21,t22,t23,t24,t2  
5 : std_logic;  
  
begin  
  
inv_1 : inverter port map(start,start_b);  
inv_2 : inverter port map(reset, reset_b);  
  
or2_1 : or2 port map(t1,reset,t2);  
  
and2_2 : and2 port map(q,start_b,t1);  
dff_1 : dff port map(t2,clk,q);  
and2_3 : and2 port map(q,start, t3);  
dff_2 : dff port map(t3,clk,s0_en);  
  
--Read Miss--  
--and2_4 : and2 port map(s0_en,r_m,r_m_st);  
dff_3 : dff port map(s0_en,clk,t7);  
and2_8 : and2 port map(t7,r_m,s1_en);  
dff_4 : dff port map(s1_en,clk,t8);  
and2_9 : and2 port map(t8,reset_b,s2_en);  
dff_5 : dff port map(s2_en,clk,t9);  
and2_10 : and2 port map(t9, reset_b,s3_en);  
dff_6 : dff port map(s3_en,clk,t10);  
and2_11 : and2 port map(t10, reset_b, s4_en);  
dff_7 : dff port map(s4_en,clk,t11);  
and2_12 : and2 port map(t11, reset_b,s5_en);  
dff_8 : dff port map(s5_en,clk,t12);  
and2_13 : and2 port map(t12, reset_b,s6_en);  
dff_9 : dff port map(s6_en,clk,t13);  
and2_14 : and2 port map(t13, reset_b,s7_en);  
dff_10 : dff port map(s7_en,clk,t14);  
and2_15 : and2 port map(t14, reset_b,s8_en);  
dff_11 : dff port map(s8_en,clk,t15);  
and2_16 : and2 port map(t15, reset_b,s9_en);  
dff_12 : dff port map(s9_en,clk,t16);  
and2_17 : and2 port map(t16, reset_b,s10_en);  
dff_13 : dff port map(s10_en,clk,t17);  
and2_18 : and2 port map(t17, reset_b,s11_en);  
dff_14 : dff port map(s11_en,clk,t18);  
and2_19 : and2 port map(t18, reset_b,s12_en);  
dff_15 : dff port map(s12_en,clk,t19);  
and2_20 : and2 port map(t19, reset_b,s13_en);  
dff_16 : dff port map(s13_en,clk,t20);  
and2_21 : and2 port map(t20, reset_b,s14_en);  
dff_17 : dff port map(s14_en,clk,t21);  
and2_22 : and2 port map(t21, reset_b,s15_en);  
dff_18 : dff port map(s15_en,clk,t22);  
and2_23 : and2 port map(t22, reset_b,s16_en);  
dff_19 : dff port map(s16_en,clk,t23);  
and2_24 : and2 port map(t23, reset_b,s17_en);  
dff_20 : dff port map(s17_en,clk,s18_en);  
  
--read hit--  
and2_25 : and2 port map(s0_en,r_h,r_h_st);  
dff_22 : dff port map(r_h_st,clk,s1_en_rh);  
  
--write miss--  
and2_26 : and2 port map(s0_en,w_m,w_m_st);  
dff_23 : dff port map(w_m_st, clk,t24);  
and2_25 : and2 port map(t24,reset_b,s1_en_wm);  
dff_24 : dff port map(s1_en_wm,clk,s2_en_wm);  
  
--write hit--  
and2_27 : and2 port map(s0_en,w_h,w_h_st);  
dff_25 : dff port map(w_h_st, clk,t25);  
and2_26 : and2 port map(t25,reset_b,s1_en_wh);  
dff_26 : dff port map(s1_en_wh, clk,s2_en_wh);  
  
or2_2 : or2 port map(s18_en,s1_en_rh,t4);  
or2_3 : or2 port map(s2_en_wm,s2_en_wh,t5);
```

```
or2_4 : or2 port map(t4,t5,t6);  
end structural;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity tag_cmp1 is
port(t14,t15,t16,t17,t18,t19,t20,t21: in std_logic;
      mem2,mem3,mem4 : inout std_logic;
      write_cache : out std_logic
     );
end tag_cmp1;

architecture structural of tag_cmp1 is

component or2
port(
  input1  : in std_logic;
  input2 : in std_logic;
  output  : out std_logic);
end component;

for all : or2 use entity work.or2(structural);

signal mem1,write_cache0,write_cachel : std_logic;

begin

or2_mem1 : or2 port map(t14,t15,mem1);
or2_mem2 : or2 port map(t17,t16,mem2);
or2_mem3 : or2 port map(t18,t19,mem3);
or2_mem4 : or2 port map(t20,t21,mem4);
or2_byte0 : or2 port map(input1=>mem2,input2=>mem1,output=>write_cache0);
or2_byte1 : or2 port map(input1=>mem3,input2=>mem4,output=>write_cachel);
or2_byte : or2 port map(write_cache0, write_cachel,write_cache);

end structural;
```

```
-- Entity: tag_valid
-- Architecture : structural
-- Author: n67
--

library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity tag_valid is
port ( din : in std_logic_vector(3 downto 0);
      blk_en0,blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7 : in std_logic
;
      rw_en : in std_logic;
      dout : out std_logic_vector(3 downto 0));
end tag_valid;

architecture structural of tag_valid is

component cache_4b
port ( din : in std_logic_vector(3 downto 0);
       en : in std_logic;
       rw_en : in std_logic;
       dout : out std_logic_vector(3 downto 0));
end component;

for cache_4b_0, cache_4b_1, cache_4b_2, cache_4b_3, cache_4b_4, cache_4b_5, cache_4b_6
, cache_4b_7 : cache_4b use entity work.cache_4b(structural);

begin

cache_4b_0 : cache_4b port map(din, blk_en0,rw_en,dout);
cache_4b_1 : cache_4b port map(din, blk_en1,rw_en,dout);
cache_4b_2 : cache_4b port map(din, blk_en2,rw_en,dout);
cache_4b_3 : cache_4b port map(din, blk_en3,rw_en,dout);
cache_4b_4 : cache_4b port map(din, blk_en4,rw_en,dout);
cache_4b_5 : cache_4b port map(din, blk_en5,rw_en,dout);
cache_4b_6 : cache_4b port map(din, blk_en6,rw_en,dout);
cache_4b_7 : cache_4b port map(din, blk_en7,rw_en,dout);

end structural;
```

```
library STD;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity tristate_buf_8b is

    Port ( A      : in  STD_LOGIC_VECTOR(7 downto 0);
           EN     : in  STD_LOGIC;
           Y      : out STD_LOGIC_VECTOR(7 downto 0));
end tristate_buf_8b;

architecture structural of tristate_buf_8b is

component tristate_buf
    Port ( A      : in  STD_LOGIC;
           EN     : in  STD_LOGIC;
           Y      : out STD_LOGIC);
end component;

for all : tristate_buf use entity work.tristate_buf(structural);

begin
    tristate_buf_0 : tristate_buf port map(A(0),EN,Y(0));
    tristate_buf_1 : tristate_buf port map(A(1),EN,Y(1));
    tristate_buf_2 : tristate_buf port map(A(2),EN,Y(2));
    tristate_buf_3 : tristate_buf port map(A(3),EN,Y(3));
    tristate_buf_4 : tristate_buf port map(A(4),EN,Y(4));
    tristate_buf_5 : tristate_buf port map(A(5),EN,Y(5));
    tristate_buf_6 : tristate_buf port map(A(6),EN,Y(6));
    tristate_buf_7 : tristate_buf port map(A(7),EN,Y(7));
end structural;
```

```
library STD;
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity tristate_buf is

    Port ( A      : in  STD_LOGIC;
           EN     : in  STD_LOGIC;
           Y      : out STD_LOGIC);
end tristate_buf;

architecture structural of tristate_buf is

component tx
    port ( sel    : in std_logic;
           selnot: in std_logic;
           input  : in std_logic;
           output : out std_logic);
end component;

component inverter
port (input : in std_logic;
output : out std_logic);
end component;

for all : tx use entity work.tx(structural);
for all : inverter use entity work.inverter(structural);

signal EN_b:std_logic;
begin

    inv_1 : inverter port map (EN, EN_b);
    tx_1 : tx port map (EN, EN_b,A,Y);
    --Y <= A when (EN = '1') else 'Z';

end structural;
```

```
-- Entity: tx
-- Architecture : structural
-- Author: n67
--

library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity tx is
  port ( sel    : in std_logic;
         selnot: in std_logic;
         input : in std_logic;
         output:out std_logic);
end tx;

architecture structural of tx is
begin

  txprocess: process (sel, selnot, input)
  begin
    if (sel = '1' and selnot = '0') then
      output <= input;
    else
      output <= 'Z';
    end if;
  end process txprocess;

end structural;
```

```
library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity xnor2 is
  port(
    input1 : in std_logic;
    input2 : in std_logic;
    output : out std_logic);
end xnor2;

architecture structural of xnor2 is

component inverter
port(input : in std_logic;
      output : out std_logic);
end component;

component xor2
port(
  input1 : in std_logic;
  input2 : in std_logic;
  output : out std_logic);
end component;

for inv_1 : inverter use entity work.inverter(structural);
for xor2_1 : xor2 use entity work.xor2(structural);

signal templ:std_logic;

begin

xor2_1 : xor2 port map(input1, input2, templ);
inv_1 : inverter port map(templ, output);

end structural;
```

```
library STD;
library IEEE;
use IEEE.std_logic_1164.all;

entity xor2 is
  port(
    input1 : in std_logic;
    input2 : in std_logic;
    output : out std_logic);
end xor2;

architecture structural of xor2 is

component nand2
port(
  input1 : in std_logic;
  input2 : in std_logic;
  output : out std_logic);
end component;

for all: nand2 use entity work.nand2(structural);

signal temp1, temp2, temp3, temp4 :std_logic;

begin
nand2_1 : nand2 port map(input1,input2,temp1);
nand2_2 : nand2 port map(input1,temp1,temp2);
nand2_3 : nand2 port map(input2,temp1,temp3);
nand2_4 : nand2 port map(temp2,temp3,output);

end structural;
```

```
1:
2: library STD;
3: library IEEE;
4: use IEEE.std_logic_1164.all;
5: use IEEE.std_logic_textio.all;
6: use STD.textio.all;
7:
8: entity cache_32b_tb is
9:
10: end cache_32b_tb;
11:
12: architecture test of cache_32b_tb is
13:
14: component cache_32b
15: port ( din : in std_logic_vector(7 downto 0);
16:         by_en0,by_en1,by_en2,by_en3 :in std_logic;
17:         blk_en :in std_logic;
18:         rw_en :in std_logic;
19:         dout :out std_logic_vector(7 downto 0));
20: end component;
21:
22: for cache_32b_1: cache_32b use entity work.cache_32b(structural);
23:
24: signal rw_en1, by_en01,by_en11,by_en21,by_en31, blk_en1: std_logic;
25: signal din1, dout1 :std_logic_vector(7 downto 0);
26:
27: procedure print_output is
28:   variable out_line: line;
29:
30: begin
31:   write (out_line, string'(" din1:"));
32:   write (out_line, din1);
33:   -- write (out_line, string'(" en1:"));
34:   -- write (out_line, en1);
35:   write (out_line, string'(" rw_en1:"));
36:   write (out_line, rw_en1);
37:   writeline(output, out_line);
38:   write (out_line, string'(" dout:"));
39:   write (out_line, dout1);
40:   writeline(output, out_line);
41: end print_output;
42:
43: begin
44:
45: cache_32b_1 : cache_32b port map (din1,by_en01,by_en11,by_en21,by_en31, blk_en
1, rw_en1, dout1);
46:
47: io_process: process
48:   variable out_line: line;
49:
50: begin
51:
52:   din1 <= "10100000";
53:   by_en01 <= '1';
54:   by_en11 <= '0';
55:   by_en21 <= '0';
56:   by_en31 <= '0';
57:   blk_en1 <= '1';
58:   rw_en1<= '0';
59:
60:   wait for 10 ns;
61:
```

```
62:   by_en01 <= '0';
63:   by_en11 <= '1';
64:   by_en21 <= '0';
65:   by_en31 <= '0';
66:   blk_en1 <= '1';
67:   rw_en1<= '0';
68:
69:   wait for 2 ns;
70:   din1 <= "11111111";
71:   wait for 10 ns;
72:
73:   by_en01 <= '0';
74:   by_en11 <= '0';
75:   by_en21 <= '1';
76:   by_en31 <= '0';
77:   blk_en1 <= '1';
78:   rw_en1<= '0';
79:
80:   wait for 2 ns;
81:   din1 <= "11000110";
82:   wait for 10 ns;
83:
84:   print_output;
85:
86:   by_en01 <= '0';
87:   by_en11 <= '0';
88:   by_en21 <= '0';
89:   by_en31 <= '1';
90:   blk_en1 <= '1';
91:   rw_en1<= '0';
92:
93:   wait for 2 ns;
94:   din1 <= "11101010";
95:   wait for 10 ns;
96:
97:
98:   print_output;
99:
100:  --din1 <= "11010111";
101:  by_en01 <= '1';
102:  by_en11 <= '0';
103:  by_en21 <= '0';
104:  by_en31 <= '0';
105:  blk_en1 <= '1';
106:  rw_en1<= '1';
107:  wait for 10 ns;
108:
109:  print_output;
110:  rw_en1<= '1';
111:  --din1 <= "11010111";
112:  by_en01 <= '0';
113:  by_en11 <= '1';
114:  by_en21 <= '0';
115:  by_en31 <= '0';
116:  blk_en1 <= '1';
117:
118:  wait for 10 ns;
119:
120:  print_output;
121:  rw_en1<= '1';
122:  --din1 <= "11010111";
123:  by_en01 <= '0';
```

```
124: by_en11 <= '0';
125: by_en21 <= '1';
126: by_en31 <= '0';
127: blk_en1 <= '1';
128:
129:
130: wait for 10 ns;
131:
132: print_output;
133:
134: --din1 <= "11010111";
135: by_en01 <= '0';
136: by_en11 <= '0';
137: by_en21 <= '0';
138: by_en31 <= '1';
139: blk_en1 <= '1';
140: rw_en1<= '1';
141:
142: wait for 10 ns;
143:
144: print_output;
145: end process;
146:
147: end test;
```

```
1:
2: library STD;
3: library IEEE;
4: use IEEE.std_logic_1164.all;
5: use IEEE.std_logic_textio.all;
6: use STD.textio.all;
7:
8: entity cache_8b_tb is
9:
10: end cache_8b_tb;
11:
12: architecture test of cache_8b_tb is
13:
14: component cache_8b
15: port ( din : in std_logic_vector(7 downto 0);
16:         en : in std_logic;
17:         rw_en : in std_logic;
18:         dout : out std_logic_vector(7 downto 0));
19: end component;
20:
21: for cache_8b_1: cache_8b use entity work.cache_8b(structural);
22:
23: signal rw_en1, en1: std_logic;
24: signal din1, dout1 :std_logic_vector(7 downto 0);
25:
26: procedure print_output is
27:     variable out_line: line;
28:
29: begin
30:     write (out_line, string'(" din1:" ));
31:     write (out_line, din1);
32:     write (out_line, string'(" en1:" ));
33:     write (out_line, en1);
34:     write (out_line, string'(" rw_en1:" ));
35:     write (out_line, rw_en1);
36:     writeline(output, out_line);
37:     write (out_line, string'(" dout:" ));
38:     write (out_line, dout1);
39:     writeline(output, out_line);
40: end print_output;
41:
42: begin
43:
44: cache_8b_1 : cache_8b port map (din1, en1, rw_en1, dout1);
45:
46: io_process: process
47:     variable out_line: line;
48:
49: begin
50:     din1 <= "11010001";
51:     en1 <= '1';
52:     rw_en1<= '0';
53:
54:     wait for 10 ns;
55:
56:     print_output;
57:
58:     en1 <= '1';
59:     rw_en1 <= '1';
60:
61:     wait for 10 ns;
62:     print_output;
```

```
1: library STD;
2: library IEEE;
3: use IEEE.std_logic_1164.all;
4: use IEEE.std_logic_textio.all;
5: use STD.textio.all;
6:
7: entity cache_cell_tb is
8:
9: end cache_cell_tb;
10:
11: architecture test of cache_cell_tb is
12:
13: component cache_cell
14:     port(
15:             din      : in std_logic;
16:             en       : in std_logic;
17:             rw_en   : in std_logic;
18:             dout    : out std_logic);
19: end component;
20:
21: for cache_cell_1: cache_cell use entity work.cache_cell(structural);
22:
23: signal din1, rw_en1, en1, dout1 : std_logic;
24:
25: procedure print_output is
26:     variable out_line: line;
27:
28: begin
29:     write (out_line, string'(" din1:" ));
30:     write (out_line, din1);
31:     write (out_line, string'(" en1:" ));
32:     write (out_line, en1);
33:     write (out_line, string'(" rw_en1:" ));
34:     write (out_line, rw_en1);
35:     writeline(output, out_line);
36:     write (out_line, string'(" dout:" ));
37:     write (out_line, dout1);
38:     writeline(output, out_line);
39: end print_output;
40:
41: begin
42:
43: cache_cell_1 : cache_cell port map (din1, en1, rw_en1, dout1);
44:
45: io_process: process
46:     variable out_line: line;
47:
48: begin
49:     din1 <= '1';
50:     en1 <= '1';
51:     rw_en1<= '0';
52:
53:     wait for 10 ns;
54:
55:     print_output;
56:
57:     en1 <= '1';
58:     rw_en1 <= '1';
59:
60:     wait for 10 ns;
61:     print_output;
62:
```

```
1:
2: library STD;
3: library IEEE;
4: use IEEE.std_logic_1164.all;
5: use IEEE.std_logic_textio.all;
6: use STD.textio.all;
7:
8: entity cache_tb is
9:
10: end cache_tb;
11:
12: architecture test of cache_tb is
13:
14: component cache
15: port ( din : in std_logic_vector(7 downto 0);
16:         by_en0,by_en1,by_en2,by_en3 : in std_logic;
17:         blk_en0,blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7 : in s
td_logic;
18:         rw_en : in std_logic;
19:         dout : out std_logic_vector(7 downto 0));
20: end component;
21:
22: for cache_1: cache use entity work.cache(structural);
23: signal rw_en1, by_en01,by_en11,by_en21,by_en31,blk_en01, blk_en11,blk_en21,blk
_en31,blk_en41,blk_en51,blk_en61,blk_en71 : std_logic;
24: signal din1, dout1 : std_logic_vector(7 downto 0);
25:
26: procedure print_output is
27:     variable out_line: line;
28:
29: begin
30:     write (out_line, string'(" din1:"));
31:     write (out_line, din1);
32: --     write (out_line, string'(" en1:"));
33: --     write (out_line, en1);
34:     write (out_line, string'(" rw_en1:"));
35:     write (out_line, rw_en1);
36:     writeline(output, out_line);
37:     write (out_line, string'(" dout:"));
38:     write (out_line, dout1);
39:     writeline(output, out_line);
40: end print_output;
41:
42: begin
43:
44: cache_1 : cache port map (din1,by_en01,by_en11,by_en21,by_en31,blk_en01, blk_e
n11,blk_en21,blk_en31,blk_en41,blk_en51,blk_en61,blk_en71, rw_en1, dout1);
45:
46: io_process: process
47:     variable out_line: line;
48:
49: begin
50:
51: din1 <= "10100000";
52: rw_en1 <= '0';
53: by_en01 <= '1';
54: by_en11 <= '0';
55: by_en21 <= '0';
56: by_en31 <= '0';
57: blk_en01      <= '1';
58: blk_en11      <= '0';
59: blk_en21      <= '0';
60: blk_en31      <= '0';
61: blk_en41      <= '0';
62: blk_en51      <= '0';
63: blk_en61      <= '0';
64: blk_en71      <= '0';
65:
66: wait for 10 ns;
67:
68: print_output;
69:
70: rw_en1 <= '0';
71: by_en01 <= '0';
72: by_en11 <= '0';
73: by_en21 <= '1';
74: by_en31 <= '0';
75: blk_en01      <= '0';
76: blk_en11      <= '0';
77: blk_en21      <= '0';
78: blk_en31      <= '0';
79: blk_en41      <= '1';
80: blk_en51      <= '0';
81: blk_en61      <= '0';
82: blk_en71      <= '0';
83:
84: wait for 2 ns;
85: din1 <= "10101010";
86: wait for 10 ns;
87:
88: print_output;
89:
90:
91: rw_en1 <= '0';
92: by_en01 <= '0';
93: by_en11 <= '0';
94: by_en21 <= '0';
95: by_en31 <= '1';
96: blk_en01      <= '0';
97: blk_en11      <= '0';
98: blk_en21      <= '0';
99: blk_en31      <= '0';
100: blk_en41     <= '0';
101: blk_en51     <= '1';
102: blk_en61     <= '0';
103: blk_en71     <= '0';
104: wait for 2 ns;
105: din1 <= "11000110";
106: wait for 10 ns;
107:
108: print_output;
109:
110:
111: rw_en1 <= '0';
112: by_en01 <= '0';
113: by_en11 <= '1';
114: by_en21 <= '0';
115: by_en31 <= '0';
116: blk_en01      <= '0';
117: blk_en11      <= '0';
118: blk_en21      <= '0';
119: blk_en31      <= '1';
120: blk_en41      <= '0';
121: blk_en51      <= '0';
```

```
122: blk_en61      <= '0';
123: blk_en71      <= '0';
124:
125: wait for 2 ns;
126: din1 <= "11101010";
127: wait for 10 ns;
128:
129: print_output;
130: rw_en1 <= '1';
131: by_en01 <= '0';
132: by_en11 <= '0';
133: by_en21 <= '0';
134: by_en31 <= '0';
135: blk_en01      <= '0';
136: blk_en11      <= '0';
137: blk_en21      <= '0';
138: blk_en31      <= '0';
139: blk_en41      <= '0';
140: blk_en51      <= '0';
141: blk_en61      <= '0';
142: blk_en71      <= '0';
143: wait for 10 ns;
144:
145: by_en01 <= '1';
146: by_en11 <= '0';
147: by_en21 <= '0';
148: by_en31 <= '0';
149: blk_en01      <= '1';
150: blk_en11      <= '0';
151: blk_en21      <= '0';
152: blk_en31      <= '0';
153: blk_en41      <= '0';
154: blk_en51      <= '0';
155: blk_en61      <= '0';
156: blk_en71      <= '0';
157: wait for 10 ns;
158:
159: print_output;
160:
161: rw_en1 <= '1';
162: by_en01 <= '0';
163: by_en11 <= '0';
164: by_en21 <= '1';
165: by_en31 <= '0';
166: blk_en01      <= '0';
167: blk_en11      <= '0';
168: blk_en21      <= '0';
169: blk_en31      <= '0';
170: blk_en41      <= '1';
171: blk_en51      <= '0';
172: blk_en61      <= '0';
173: blk_en71      <= '0';
174: wait for 2 ns;
175:
176: wait for 10 ns;
177:
178: print_output;
179:
180: rw_en1 <= '1';
181: by_en01 <= '0';
182: by_en11 <= '0';
183: by_en21 <= '0';

184: by_en31 <= '1';
185: blk_en01      <= '0';
186: blk_en11      <= '0';
187: blk_en21      <= '0';
188: blk_en31      <= '0';
189: blk_en41      <= '0';
190: blk_en51      <= '1';
191: blk_en61      <= '0';
192: blk_en71      <= '0';
193:
194: wait for 10 ns;
195:
196: print_output;
197:
198: rw_en1 <= '1';
199: by_en01 <= '0';
200: by_en11 <= '1';
201: by_en21 <= '0';
202: by_en31 <= '0';
203: blk_en01      <= '0';
204: blk_en11      <= '0';
205: blk_en21      <= '0';
206: blk_en31      <= '1';
207: blk_en41      <= '0';
208: blk_en51      <= '0';
209: blk_en61      <= '0';
210: blk_en71      <= '0';
211:
212: wait for 10 ns;
213:
214: print_output;
215:
216: end process;
217:
218: end test;
```

Cache Design Testbench
chip_full_test.vhd

```
1: -- Entity: chip_full_test
2: -- Architecture : test
3: -- Author: cpatel2
4: -- Created On: 11/01/05
5: --
6: library IEEE;
7: use IEEE.std_logic_1164.all;
8: use IEEE.std_logic_textio.all;
9: use IEEE.std_logic_arith.all;
10: use STD.textio.all;
11:
12: entity chip_full_test is
13: end chip_full_test;
15:
16: architecture test of chip_full_test is
17:
18: component chip
19: port (
20:     cpu_add    : in std_logic_vector(7 downto 0);
21:     cpu_data   : inout std_logic_vector(7 downto 0);
22:     cpu_rd_wrn : in std_logic;
23:     start      : in std_logic;
24:     clk        : in std_logic;
25:     reset      : in std_logic;
26:     mem_data   : in std_logic_vector(7 downto 0);
27:     Vdd        : in std_logic;
28:     Gnd        : in std_logic;
29:     busy       : out std_logic;
30:     mem_en    : out std_logic;
31:     mem_add   : out std_logic_vector(7 downto 0));
32: end component;
33:
34:
35:
36: for cl : chip use entity work.chip(structural);
37:
38: signal Vdd, Gnd: std_logic;
39: signal cpu_add, cpu_data, mem_data, mem_add: std_logic_vector(7 downto 0);
40: signal cpu_rd_wrn, reset, clk, start, clock, busy, mem_en: std_logic;
41:
42: signal clk_count: integer:=0;
43:
44: procedure print_output is
45:     variable out_line: line;
46:
47: begin
48:     write (out_line, string' (" Clock: "));
49:     write (out_line, clk_count);
50:     write (out_line, string'(" Start: "));
51:     write (out_line, start);
52:     write (out_line, string'(" Cpu Read/Write: "));
53:     write (out_line, cpu_rd_wrn);
54:     write (out_line, string'(" Reset: "));
55:     write (out_line, reset);
56:     writeline(output, out_line);
57:
58:     write (out_line, string' (" CPU address: "));
59:     write (out_line, cpu_add);
60:     write (out_line, string'(" CPU data: "));
61:     write (out_line, cpu_data);
62:     writeline(output, out_line);
```

```
63:
64:         write (out_line, string'" Memory data: "));
65:         write (out_line, mem_data);
66:         writeline(output, out_line);
67:         writeline(output, out_line);
68:
69:         write (out_line, string'" Busy: "));
70:         write (out_line, busy);
71:         write (out_line, string'" Memory Enable: "));
72:         write (out_line, mem_en);
73:         writeline(output, out_line);
74:
75:         write (out_line, string'" Memory Address: "));
76:         write (out_line, mem_add);
77:         writeline(output, out_line);
78:
79:         write (out_line, string'" -----"));
80:     );
81:     writeline(output, out_line);
82:
83: end print_output;
84:
85:
86:
87: begin
88:
89:     Vdd <= '1';
90:     Gnd <= '0';
91:     clk <= clock;
92:
93:     cl : chip port map (cpu_add, cpu_data, cpu_rd_wrn, start, clk, reset, mem_da-
ta, Vdd, Gnd, busy, mem_en, mem_add);
94:
95:     clking : process
96:     begin
97:         clock<= '1', '0' after 5 ns;
98:         wait for 10 ns;
99:     end process clking;
100:
101:    io_process: process
102:
103:        file infile : text is in "./chip_full_in.txt";
104:        variable buf: line;
105:        variable value: std_logic_vector(7 downto 0);
106:        variable valuel: std_logic;
107:
108:    begin
109:
110:        while not (endfile(infile)) loop
111:
112:            wait until rising_edge(clock);
113:
114:            readline(infile, buf);
115:
116:            readline(infile, buf);
117:            read(buf, value);
118:            cpu_add <= value;
119:
120:            readline(infile, buf);
121:            read(buf, value);
122:            cpu_data <= value;
```

```
123:      readline(infile, buf);
124:      read(buf, value1);
125:      cpu_rd_wrn <= value1;
126:
127:
128:      readline(infile, buf);
129:      read(buf, value1);
130:      start <= value1;
131:
132:      readline(infile, buf);
133:      read(buf, value1);
134:      reset <= value1;
135:
136:      clk_count <= clk_count+1;
137:
138:      wait until falling_edge(clock);
139:
140:      readline(infile, buf);
141:      read(buf, value);
142:      mem_data <= value;
143:
144: end loop;
145: wait;
146:
147: end process io_process;
148:
149: print_process: process
150:
151:   variable out_line: line;
152:
153: begin
154:
155:   wait until ((falling_edge(clock) and start ='1') or busy'EVENT or mem_en'EVE
NT);
156:   wait for 1 ns;
157:   print_output;
158:
159: end process print_process;
160:
161: end test;
```

```
1: -- Entity: chip_test
2: -- Architecture : test
3: -- Author: cpatel2
4: -- Created On: 11/01/05
5: --
6: library IEEE;
7: use IEEE.std_logic_1164.all;
8: use IEEE.std_logic_textio.all;
9: use IEEE.std_logic_arith.all;
10: use STD.textio.all;
11:
12: entity chip_test is
13: end chip_test;
15:
16: architecture test of chip_test is
17:
18: component chip
19: port (
20:     cpu_add    : in std_logic_vector(7 downto 0);
21:     cpu_data   : inout std_logic_vector(7 downto 0);
22:     cpu_rd_wrn : in std_logic;
23:     start      : in std_logic;
24:     clk        : in std_logic;
25:     reset      : in std_logic;
26:     mem_data   : in std_logic_vector(7 downto 0);
27:     Vdd        : in std_logic;
28:     Gnd        : in std_logic;
29:     busy       : out std_logic;
30:     mem_en    : out std_logic;
31:     mem_add   : out std_logic_vector(7 downto 0));
32: end component;
33:
34:
35:
36: for cl : chip use entity work.chip(structural);
37:
38: signal Vdd, Gnd: std_logic;
39: signal cpu_add, cpu_data, mem_data, mem_add: std_logic_vector(7 downto 0);
40: signal cpu_rd_wrn, reset, clk, start, clock, busy, mem_en: std_logic;
41:
42: signal clk_count: integer:=0;
43:
44: procedure print_output is
45:     variable out_line: line;
46:
47: begin
48:     write (out_line, string' (" Clock: "));
49:     write (out_line, clk_count);
50:     write (out_line, string'(" Start: "));
51:     write (out_line, start);
52:     write (out_line, string'(" Cpu Read/Write: "));
53:     write (out_line, cpu_rd_wrn);
54:     write (out_line, string'(" Reset: "));
55:     write (out_line, reset);
56:     writeline(output, out_line);
57:
58:     write (out_line, string' (" CPU address: "));
59:     write (out_line, cpu_add);
60:     write (out_line, string'(" CPU data: "));
61:     write (out_line, cpu_data);
62:     writeline(output, out_line);
63:
64:         write (out_line, string'" Memory data: "'));
65:         write (out_line, mem_data);
66:         writeline(output, out_line);
67:         writeline(output, out_line);
68:
69:         write (out_line, string'" Busy: "'));
70:         write (out_line, busy);
71:         write (out_line, string'" Memory Enable: "'));
72:         write (out_line, mem_en);
73:         writeline(output, out_line);
74:
75:         write (out_line, string'" Memory Address: "'));
76:         write (out_line, mem_add);
77:         writeline(output, out_line);
78:
79:         write (out_line, string'" -----"');
80:         writeline(output, out_line);
81:
82:
83: end print_output;
84:
85:
86:
87: begin
88:
89:     Vdd <= '1';
90:     Gnd <= '0';
91:     clk <= clock;
92:
93:     cl : chip port map (cpu_add, cpu_data, cpu_rd_wrn, start, clk, reset, mem_da
ta, Vdd, Gnd, busy, mem_en, mem_add);
94:
95:     clking : process
96:     begin
97:         clock<= '1', '0' after 5 ns;
98:         wait for 10 ns;
99:     end process clking;
100:
101:    io_process: process
102:
103:        file infile : text is in "./chip_in.txt";
104:        variable out_line: line;
105:        variable buf: line;
106:        variable value: std_logic_vector(7 downto 0);
107:        variable value1: std_logic;
108:
109:        begin
110:
111:            while not (endfile(infile)) loop
112:
113:                wait until rising_edge(clock);
114:                print_output;
115:
116:                readline(infile, buf);
117:                read(buf, value);
118:                cpu_add <= value;
119:
120:                readline(infile, buf);
121:                read(buf, value);
122:                cpu_data <= value;
```

```
123:      readline(infile, buf);
124:      read(buf, value1);
125:      cpu_rd_wrn <= value1;
126:
127:
128:      readline(infile, buf);
129:      read(buf, value1);
130:      start <= value1;
131:
132:      readline(infile, buf);
133:      read(buf, value1);
134:      reset <= value1;
135:
136:      wait until falling_edge(clock);
137:
138:      readline(infile, buf);
139:      read(buf, value);
140:      mem_data <= value;
141:
142:      clk_count <= clk_count+1;
143:
144:      print_output;
145:
146:  end loop;
147:  wait;
148:
149: end process io_process;
150:
151:
152: end test;
153:
```

```
1: library STD;
2: library IEEE;
3: use IEEE.std_logic_1164.all;
4: use IEEE.std_logic_textio.all;
5: use STD.textio.all;
6:
7: entity cmprtr2_test is
8:
9: end cmprtr2_test;
10:
11: architecture test of cmprtr2_test is
12:
13: component cmprtr2
14:   port (
15:     input1 : in std_logic_vector(2 downto 0);
16:     input2 : in std_logic_vector(2 downto 0);
17:     eql : out std_logic);
18: end component;
19:
20: for cmprtr2_1: cmprtr2 use entity work.cmprtr2(structural);
21:
22: signal input11, input21 : std_logic_vector(2 downto 0);
23: signal eq11 : std_logic;
24:
25: procedure print_output is
26:   variable out_line: line;
27:
28: begin
29:   write (out_line, string'(" input11:" ));
30:   write (out_line, input11);
31:   write (out_line, string'(" input21:" ));
32:   write (out_line, input21);
33:   writeline(output, out_line);
34:   write (out_line, string'(" eql1:" ));
35:   write (out_line, eq11);
36:   writeline(output, out_line);
37: end print_output;
38:
39: begin
40:
41: cmprtr2_1 : cmprtr2 port map (input11,input21,eql1);
42:
43: io_process: process
44:   variable out_line: line;
45:
46: begin
47:
48:   input11 <= "101";
49:   input21 <= "101";
50:
51:   wait for 10 ns;
52:   print_output;
53:
54:   input11 <= "101";
55:   input21 <= "111";
56:
57:   wait for 10 ns;
58:   print_output;
59:
60: end process;
61:
62: end test;
```

```
1:
2: library STD;
3: library IEEE;
4: use IEEE.std_logic_1164.all;
5: use IEEE.std_logic_textio.all;
6: use STD.textio.all;
7:
8: entity dec2_4_test is
9:
10: end dec2_4_test;
11:
12: architecture test of dec2_4_test is
13:
14: component dec2_4
15: port(s0,s1,en : in std_logic;
16: y0,y1,y2,y3 : out std_logic);
17: end component;
18:
19: for dec2_4_1: dec2_4 use entity work.dec2_4(structural);
20: signal s01,s11,en1,y01,y11,y21,y31: std_logic;
21:
22: procedure print_output is
23:     variable out_line: line;
24:
25: begin
26: --    write (out_line, string'(" din1:"));
27: --    write (out_line, din1);
28: ----    write (out_line, string'(" en1:"));
29: ----    write (out_line, en1);
30: --    write (out_line, string'(" rw_en1:"));
31: --    write (out_line, rw_en1);
32: --    writeline(output, out_line);
33: --    write (out_line, string'(" dout:"));
34: --    write (out_line, dout1);
35: --    writeline(output, out_line);
36: end print_output;
37:
38: begin
39:
40: dec2_4_1 : dec2_4 port map (s01,s11,en1,y01,y11,y21,y31);
41:
42:
43: io_process: process
44:     variable out_line: line;
45:
46: begin
47:
48: s01 <= '1';
49: s11 <= '1';
50: en1 <= '0';
51:
52: wait for 10 ns;
53:
54: s01 <= '0';
55: s11 <= '0';
56: en1 <= '1';
57:
58: wait for 10 ns;
59:
60: s01 <= '0';
61: s11 <= '1';
62: en1 <= '1';
63:
64: wait for 10 ns;
65:
66: s01 <= '1';
67: s11 <= '0';
68: en1 <= '1';
69:
70: wait for 10 ns;
71:
72: s01 <= '1';
73: s11 <= '1';
74: en1 <= '1';
75:
76: wait for 10 ns;
77:
78: print_output;
79:
80: end process;
81:
82: end test;
83:
```

```
1:
2: library STD;
3: library IEEE;
4: use IEEE.std_logic_1164.all;
5: use IEEE.std_logic_textio.all;
6: use STD.textio.all;
7:
8: entity dec3_8_test is
9:
10: end dec3_8_test;
11:
12: architecture test of dec3_8_test is
13:
14: component dec3_8
15: port(s0,s1,s2: in std_logic;
16: y0,y1,y2,y3,y4,y5,y6,y7 : out std_logic);
17: end component;
18:
19: for dec3_8_1: dec3_8 use entity work.dec3_8(structural);
20: signal s01,s11,s21,y01,y11,y21,y31,y41,y51,y61,y71 : std_logic;
21:
22: procedure print_output is
23:     variable out_line: line;
24:
25: begin
26:    -- write (out_line, string'(" din1:"));
27:    -- write (out_line, din1);
28:    ---- write (out_line, string'(" en1:"));
29:    ---- write (out_line, en1);
30:    -- write (out_line, string'(" rw_en1:"));
31:    -- write (out_line, rw_en1);
32:    writeline(output, out_line);
33:    -- write (out_line, string'(" dout:"));
34:    -- write (out_line, dout1);
35:    writeline(output, out_line);
36: end print_output;
37:
38: begin
39:
40: dec3_8_1 : dec3_8 port map (s01,s11,s21,y01,y11,y21,y31,y41,y51,y61,y71);
41:
42:
43: io_process: process
44:     variable out_line: line;
45:
46: begin
47:
48:    --en1<='0';
49:    s01 <= '0';
50:    s11 <= '0';
51:    s21 <= '0';
52:
53:
54:    wait for 10 ns;
55:    --en1<='1';
56:    s01 <= '0';
57:    s11 <= '0';
58:    s21 <= '1';
59:
60:
61:    wait for 10 ns;
62:    --en1<='1';
63:    s01 <= '0';
64:    s11 <= '1';
65:    s21 <= '0';
66:
67:
68:    wait for 10 ns;
69:    --en1<='1';
70:    s01 <= '0';
71:    s11 <= '1';
72:    s21 <= '1';
73:
74:
75:    wait for 10 ns;
76:    --en1<='1';
77:    s01 <= '1';
78:    s11 <= '0';
79:    s21 <= '0';
80:
81:
82:    wait for 10 ns;
83:    --en1<='1';
84:    s01 <= '1';
85:    s11 <= '0';
86:    s21 <= '1';
87:
88:
89:    wait for 10 ns;
90:    --en1<='1';
91:    s01 <= '1';
92:    s11 <= '1';
93:    s21 <= '0';
94:
95:
96:    wait for 10 ns;
97:    --en1<='1';
98:    s01 <= '1';
99:    s11 <= '1';
100:   s21 <= '1';
101:
102:
103:   wait for 10 ns;
104:
105:
106:   print_output;
107:
108: end process;
109:
110: end test;
111:
```

```
1: library STD;
2: library IEEE;
3: use IEEE.std_logic_1164.all;
4: use IEEE.std_logic_textio.all;
5: use STD.textio.all;
6:
7: entity Dlatch_8b_test is
8:
9: end Dlatch_8b_test;
10:
11: architecture test of Dlatch_8b_test is
12:
13: component Dlatch_8b
14:   port ( d  : in std_logic_vector(7 downto 0);
15:          clk : in std_logic;
16:          q   : out std_logic_vector(7 downto 0));
17: end component;
18:
19: for Dlatch_8b_1: Dlatch_8b use entity work.Dlatch_8b(structural);
20:
21: signal d1,q1 : std_logic_vector( 7  downto 0);
22: signal clk1 : std_logic;
23:
24: procedure print_output is
25:   variable out_line: line;
26:
27: begin
28:   -- write (out_line, string'(" din1:" ));
29:   -- write (out_line, din1);
30:   -- write (out_line, string'(" en1:" ));
31:   -- write (out_line, en1);
32:   -- write (out_line, string'(" rw_en1:" ));
33:   -- write (out_line, rw_en1);
34:   -- writeline(output, out_line);
35:   -- write (out_line, string'(" dout:" ));
36:   -- write (out_line, dout1);
37:   -- writeline(output, out_line);
38: end print_output;
39:
40: begin
41:
42: Dlatch_8b_1 : Dlatch_8b port map (d1,clk1,q1);
43:
44: io_process: process
45:   variable out_line: line;
46:
47: begin
48:
49: clk1<= '0';
50:
51: wait for 10ns;
52:
53: d1 <= "10101010";
54: clk1 <= '1';
55:
56: wait for 10 ns;
57:
58: print_output;
59:
60: clk1 <= '0';
61: d1 <= "10101110";
62: wait for 10 ns;
63: print_output;
64:
65: d1 <= "10101100";
66: clk1 <= '1';
67:
68: wait for 10 ns;
69:
70: print_output;
71: d1 <= "10101110";
72: clk1 <= '1';
73:
74: wait for 10 ns;
75:
76: clk1 <= '0';
77: wait for 10 ns;
78:
79: clk1 <= '1';
80: wait for 10ns;
81: end process;
82:
83: end test;
```

```
1: library STD;
2: library IEEE;
3: use IEEE.std_logic_1164.all;
4: use IEEE.std_logic_textio.all;
5: use STD.textio.all;
6:
7: entity encoder4_2_test is
8:
9: end encoder4_2_test;
10:
11: architecture test of encoder4_2_test is
12:
13: component encoder4_2
14: port( i1 : in std_logic;
15:        i2 : in std_logic;
16:        i3 : in std_logic;
17:        o1 : out std_logic;
18:        o2 : out std_logic);
19: end component;
20:
21: for encoder4_2_1: encoder4_2 use entity work.encoder4_2(structural);
22: signal i11,i21,i31,o11,o21: std_logic;
23:
24: procedure print_output is
25:     variable out_line: line;
26:
27: begin
28: --    write (out_line, string'(" din1:"));
29: --    write (out_line, din1);
30: ----    write (out_line, string'(" en1:"));
31: ----    write (out_line, en1);
32: --    write (out_line, string'(" rw_en1:"));
33: --    write (out_line, rw_en1);
34: --    writeline(output, out_line);
35: --    write (out_line, string'(" dout:"));
36: --    write (out_line, dout1);
37: --    writeline(output, out_line);
38: end print_output;
39:
40: begin
41:
42: encoder4_2_1 : encoder4_2 port map (i11,i21,i31,o11,o21);
43:
44: io_process: process
45:     variable out_line: line;
46:
47: begin
48:
49: i11<='0';
50: i21<='0';
51: i31<='0';
52:
53: wait for 10 ns;
54:
55: print_output;
56:
57: i11<='0';
58: i21<='0';
59: i31<='1';
60:
61:
62: wait for 10 ns;
```

```
63:
64: print_output;
65:
66:
67: i11<='0';
68: i21<='1';
69: i31<='0';
70:
71: wait for 10 ns;
72:
73: print_output;
74:
75:
76: i11<='1';
77: i21<='0';
78: i31<='0';
79:
80:
81: wait for 10 ns;
82:
83:
84: print_output;
85:
86: end process;
87:
88: end test;
```

```
1: library STD;
2: library IEEE;
3: use IEEE.std_logic_1164.all;
4: use IEEE.std_logic_textio.all;
5: use STD.textio.all;
6:
7: entity hit_miss_test is
8:
9: end hit_miss_test;
10:
11: architecture test of hit_miss_test is
12:
13: component hit_miss
14:   port (
15:     cache_tag_valid : in std_logic_vector(3 downto 0);
16:     cpu_tag : in std_logic_vector(2 downto 0);
17:     cpu_rw : in std_logic;
18:     sm_en : in std_logic;
19:     w_m,r_m,w_h,r_h : out std_logic);
20: end component;
21:
22: for hit_miss_1: hit_miss use entity work.hit_miss(structural);
23: signal cpu_rwl,sm_en1,w_m1,w_h1,r_m1,r_h1: std_logic;
24: signal cache_tag_valid1 :std_logic_vector(3 downto 0);
25: signal cpu_tag1 : std_logic_vector(2 downto 0);
26:
27: procedure print_output is
28:   variable out_line: line;
29:
30: begin
31:   -- write (out_line, string'(" din1:")); 
32:   -- write (out_line, din1);
33:   -- write (out_line, string'(" en1:")); 
34:   -- write (out_line, en1);
35:   -- write (out_line, string'(" rw_en1:")); 
36:   -- write (out_line, rw_en1);
37:   writeline(output, out_line);
38:   -- write (out_line, string'(" dout:")); 
39:   -- write (out_line, dout1);
40:   writeline(output, out_line);
41: end print_output;
42:
43: begin
44:
45: hit_miss_1 : hit_miss port map (cache_tag_valid1,cpu_tag1,cpu_rwl,sm_en1,w_m1,
r_m1,w_h1,r_h1);
46:
47: io_process: process
48:   variable out_line: line;
49:
50: begin
51:
52: cache_tag_valid1 <= "1010";
53: cpu_tag1 <= "010";
54: cpu_rwl <= '1';
55: sm_en1 <= '1';
56:
57: wait for 10 ns;
58:
59: print_output;
60:
61: cache_tag_valid1 <= "1010";
62: cpu_tag1 <= "010";
63: cpu_rwl <= '0';
64: sm_en1 <= '1';
65:
66:
67: wait for 10 ns;
68:
69: print_output;
70:
71:
72: cache_tag_valid1 <= "1110";
73: cpu_tag1 <= "010";
74: cpu_rwl <= '1';
75: sm_en1 <= '1';
76:
77: wait for 10 ns;
78:
79: print_output;
80:
81:
82: cache_tag_valid1 <= "1110";
83: cpu_tag1 <= "010";
84: cpu_rwl <= '0';
85: sm_en1 <= '1';
86:
87: wait for 10 ns;
88:
89: cache_tag_valid1 <= "0010";
90: cpu_tag1 <= "010";
91: cpu_rwl <= '1';
92: sm_en1 <= '1';
93:
94: wait for 10 ns;
95:
96:
97: cache_tag_valid1 <= "0010";
98: cpu_tag1 <= "010";
99: cpu_rwl <= '0';
100: sm_en1 <= '1';
101:
102: wait for 10 ns;
103:
104:
105: print_output;
106:
107: end process;
108:
109: end test;
110: end test;
```

```
1: library STD;
2: library IEEE;
3: use IEEE.std_logic_1164.all;
4: use IEEE.std_logic_textio.all;
5: use STD.textio.all;
6:
7: entity mux2x1_8b_test is
8:
9: end mux2x1_8b_test;
10:
11: architecture test of mux2x1_8b_test is
12:
13: component mux2x1_8b
14: port (
15:     A : in std_logic_vector(7 downto 0);
16:     B : in std_logic_vector(7 downto 0);
17:     sel : in std_logic;
18:     X : out std_logic_vector(7 downto 0)
19: );
20: end component;
21:
22: for mux2x1_8b_1 : mux2x1_8b use entity work.mux2x1_8b(structural);
23:
24: signal A1 : std_logic_vector(7 downto 0);
25: signal B1 : std_logic_vector(7 downto 0);
26: signal X1 : std_logic_vector(7 downto 0);
27: signal sell : std_logic;
28:
29: procedure print_output is
30:     variable out_line: line;
31:
32: begin
33: --    write (out_line, string'(" Memory "));
34: --    write (out_line, mem);
35: --    writeline(output, out_line);
36:
37: --    write (out_line, string'(" CPU "));
38: --    write (out_line, cpu);
39: --    writeline(output, out_line);
40:
41: --    write (out_line, string'(" Select "));
42: --    write (out_line, selectbit);
43: --    writeline(output, out_line);
44:
45: --    write (out_line, string'(" Output "));
46: --    write (out_line, outb);
47: --    writeline(output, out_line);
48:
49: end print_output;
50:
51: begin
52:
53: mux2x1_8b_1 : mux2x1_8b port map (A1, B1, sell, X1);
54:
55: io_process : process
56:     variable out_line: line;
57:
58: begin
59:     A1 <= "01010101";
60:     B1 <= "10101010";
61:     sell <= '0';
62:
63:     wait for 10 ns;
64:
65:     print_output;
66:
67:     A1 <= "01010101";
68:     B1 <= "10101010";
69:     sell <= '1';
70:
71:     wait for 10 ns;
72:     print_output;
73:
74: end process io_process;
75:
76: end test;
```

```
1:
2: library STD;
3: library IEEE;
4: use IEEE.std_logic_1164.all;
5: use IEEE.std_logic_textio.all;
6: use STD.textio.all;
7:
8: entity state_machine_test is
9:
10: end state_machine_test;
11:
12: architecture test of state_machine_test is
13:
14: component state_machine
15: port ( clk : in std_logic;
16:         reset : in std_logic;
17:         start : in std_logic;
18:         r_m,r_h,w_m,w_h : in std_logic);
19: end component;
20:
21: for state_machine_1: state_machine use entity work.state_machine(structural);
22: signal clk1,reset1,start1,r_ml,r_hl,w_ml,w_hl,clock: std_logic;
23:
24: procedure print_output is
25:     variable out_line: line;
26:
27: begin
28: --    write (out_line, string'(" din1:" ));
29: --    write (out_line, din1);
30: ----    write (out_line, string'(" en1:" ));
31: ----    write (out_line, en1);
32: --    write (out_line, string'(" rw_en1:" ));
33: --    write (out_line, rw_en1);
34: --    writeline(output, out_line);
35: --    write (out_line, string'(" dout:" ));
36: --    write (out_line, dout1);
37: --    writeline(output, out_line);
38: end print_output;
39:
40: begin
41:
42:    clk1 <= clock;
43:    state_machine_1 : state_machine port map (clk1,reset1,start1,r_ml,r_hl,w_ml,w_
h1);
44:
45:    clking : process
46:    begin
47:        clock<= '1', '0' after 5 ns;
48:        wait for 10 ns;
49:    end process clking;
50:
51:    io_process: process
52:        variable out_line: line;
53:
54:    begin
55:
56:        reset1 <= '1';
57:        start1 <= '0';
58:        r_ml <= '0';
59:        r_hl <= '0';
60:        w_ml <= '0';
61:        w_hl <= '0';
62:    wait for 20 ns;
63:
64:        reset1 <= '1';
65:
66:    wait for 10 ns;
67:
68:        reset1 <= '0';
69:
70:    wait for 15 ns;
71:
72:        start1 <= '1';
73:        r_ml <= '1';
74:    wait for 10 ns;
75:
76:        start1 <= '1';
77:    wait for 10 ns;
78:        start1 <= '1';
79:        r_ml <= '1';
80:    wait for 10 ns;
81:        start1 <= '0';
82:        r_ml<='0';
83:    wait for 200 ns;
84:        reset1 <= '1';
85:    wait for 10 ns;
86:        reset1 <= '0';
87:
88:    wait for 50 ns;
89:
90:        print_output;
91:
92:    end process;
93:
94: end test;
```

```
1: library STD;
2: library IEEE;
3: use IEEE.std_logic_1164.all;
4: use IEEE.std_logic_textio.all;
5: use STD.textio.all;
6:
7: entity tag_valid_test is
8:
9: end tag_valid_test;
10:
11: architecture test of tag_valid_test is
12:
13: component tag_valid
14: port ( din : in std_logic_vector(3 downto 0);
15:         blk_en0, blk_en1,blk_en2,blk_en3,blk_en4,blk_en5,blk_en6,blk_en7 : in s
td_logic;
16:         rw_en : in std_logic;
17:         dout : out std_logic_vector(3 downto 0));
18: end component;
19:
20: for tag_valid_1: tag_valid use entity work.tag_valid(structural);
21: signal rw_en1,blk_en01, blk_en11,blk_en21,blk_en31,blk_en41,blk_en51,blk_en61,
blk_en71 : std_logic;
22: signal din1, dout1 :std_logic_vector(3 downto 0);
23:
24: procedure print_output is
25:     variable out_line: line;
26:
27: begin
28:     write (out_line, string'(" din1:" ));
29:     write (out_line, din1);
30: --     write (out_line, string'(" en1:" ));
31: --     write (out_line, en1);
32:     write (out_line, string'(" rw_en1:" ));
33:     write (out_line, rw_en1);
34:     writeline(output, out_line);
35:     write (out_line, string'(" dout:" ));
36:     write (out_line, dout1);
37:     writeline(output, out_line);
38: end print_output;
39:
40: begin
41:
42: tag_valid_1 : tag_valid port map (din1,blk_en01, blk_en11,blk_en21,blk_en31,bl
k_en41,blk_en51,blk_en61,blk_en71, rw_en1, dout1);
43:
44: io_process: process
45:     variable out_line: line;
46:
47: begin
48:
49: din1 <= "1010";
50: rw_en1 <= '0';
51: blk_en01      <= '1';
52: blk_en11      <= '0';
53: blk_en21      <= '0';
54: blk_en31      <= '0';
55: blk_en41      <= '0';
56: blk_en51      <= '0';
57: blk_en61      <= '0';
58: blk_en71      <= '0';
59:
60: wait for 10 ns;
61:
62: print_output;
63:
64: rw_en1 <= '0';
65: blk_en01      <= '0';
66: blk_en11      <= '0';
67: blk_en21      <= '0';
68: blk_en31      <= '0';
69: blk_en41      <= '1';
70: blk_en51      <= '0';
71: blk_en61      <= '0';
72: blk_en71      <= '0';
73:
74: wait for 2 ns;
75: din1 <= "1010";
76: wait for 10 ns;
77:
78: print_output;
79:
80:
81: rw_en1 <= '0';
82: blk_en01      <= '0';
83: blk_en11      <= '0';
84: blk_en21      <= '0';
85: blk_en31      <= '0';
86: blk_en41      <= '0';
87: blk_en51      <= '1';
88: blk_en61      <= '0';
89: blk_en71      <= '0';
90: wait for 2 ns;
91: din1 <= "1100";
92: wait for 10 ns;
93:
94: print_output;
95:
96:
97: rw_en1 <= '0';
98:
99: blk_en01      <= '0';
100: blk_en11      <= '0';
101: blk_en21      <= '0';
102: blk_en31      <= '1';
103: blk_en41      <= '0';
104: blk_en51      <= '0';
105: blk_en61      <= '0';
106: blk_en71      <= '0';
107:
108: wait for 2 ns;
109: din1 <= "1110";
110: wait for 10 ns;
111:
112: print_output;
113: rw_en1 <= '1';
114:
115: blk_en01      <= '0';
116: blk_en11      <= '0';
117: blk_en21      <= '0';
118: blk_en31      <= '0';
119: blk_en41      <= '0';
120: blk_en51      <= '0';
121: blk_en61      <= '0';
```

```
122: blk_en71      <= '0';
123: wait for 10 ns;
124:
125:
126: blk_en01      <= '1';
127: blk_en11      <= '0';
128: blk_en21      <= '0';
129: blk_en31      <= '0';
130: blk_en41      <= '0';
131: blk_en51      <= '0';
132: blk_en61      <= '0';
133: blk_en71      <= '0';
134: wait for 10 ns;
135:
136: print_output;
137:
138: rw_en1 <= '1';
139:
140: blk_en01      <= '0';
141: blk_en11      <= '0';
142: blk_en21      <= '0';
143: blk_en31      <= '0';
144: blk_en41      <= '1';
145: blk_en51      <= '0';
146: blk_en61      <= '0';
147: blk_en71      <= '0';
148: wait for 2 ns;
149:
150: wait for 10 ns;
151:
152: print_output;
153:
154: rw_en1 <= '1';
155:
156: blk_en01      <= '0';
157: blk_en11      <= '0';
158: blk_en21      <= '0';
159: blk_en31      <= '0';
160: blk_en41      <= '0';
161: blk_en51      <= '1';
162: blk_en61      <= '0';
163: blk_en71      <= '0';
164:
165: wait for 10 ns;
166:
167: print_output;
168:
169: rw_en1 <= '1';
170: blk_en01      <= '0';
171: blk_en11      <= '0';
172: blk_en21      <= '0';
173: blk_en31      <= '1';
174: blk_en41      <= '0';
175: blk_en51      <= '0';
176: blk_en61      <= '0';
177: blk_en71      <= '0';
178:
179: wait for 10 ns;
180:
181: print_output;
182:
183: end process;
```

```
1: library STD;
2: library IEEE;
3: use IEEE.std_logic_1164.all;
4: use IEEE.std_logic_textio.all;
5: use STD.textio.all;
6:
7: entity tristate_buf_test is
8:
9: end tristate_buf_test;
10:
11: architecture test of tristate_buf_test is
12:
13: component tristate_buf
14:     Port ( A      : in STD_LOGIC;
15:             EN     : in STD_LOGIC;
16:             Y      : out STD_LOGIC);
17: end component;
18:
19: for tristate_buf_1: tristate_buf use entity work.tristate_buf(structural);
20:
21: signal A1,EN1,Y1 : std_logic;
22:
23: procedure print_output is
24:     variable out_line: line;
25:
26: begin
27: --    write (out_line, string'(" din1:"));
28: --    write (out_line, din1);
29: --    write (out_line, string'(" en1:"));
30: --    write (out_line, en1);
31: --    write (out_line, string'(" rw_en1:"));
32: --    write (out_line, rw_en1);
33: --    writeline(output, out_line);
34: --    write (out_line, string'(" dout:"));
35: --    write (out_line, dout1);
36: --    writeline(output, out_line);
37: end print_output;
38:
39: begin
40:
41: tristate_buf_1 : tristate_buf port map (A1,EN1,Y1);
42:
43: io_process: process
44:     variable out_line: line;
45:
46: begin
47: A1 <= '1';
48: EN1 <= '1';
49:
50: wait for 10 ns;
51:
52: print_output;
53:
54: A1 <= '0';
55: EN1 <= '1';
56:
57: wait for 10 ns;
58: print_output;
59:
60: A1 <= '1';
61: EN1 <= '0';
62:
63: wait for 10 ns;
64:
65: print_output;
66:
67: A1 <= '0';
68: EN1 <= '0';
69:
70: wait for 10 ns;
71:
72: end process;
73:
74: end test;
```

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      7          nets  
12:      5          terminals  
13:      3          pmos  
14:      3          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      7          nets  
19:      5          terminals  
20:      3          pmos  
21:      3          nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N2      N1      gnd!  
26:     N4      N4      input1  
27:     N3      N5      input2  
28:     N5      N2      output  
29:     N6      N0      vdd!  
30:  
31: Devices in the netlist but not in the rules:  
32:     pcapacitor  
33: Devices in the rules but not in the netlist:  
34:     cap nfet pfet nmos4 pmos4  
35:  
36: The net-lists match.  
37:  
38:           layout schematic  
39:           instances  
40:     un-matched      0      0  
41:     rewired        0      0  
42:     size errors    0      0  
43:     pruned         0      0  
44:     active         6      6  
45:     total          6      6  
46:  
47:           nets  
48:     un-matched      0      0  
49:     merged          0      0  
50:     pruned          0      0  
51:     active          7      7  
52:     total           7      7  
53:  
54:           terminals  
55:     un-matched      0      0  
56:     matched but  
57:     different type  0      0  
58:           total          5      5  
59:  
60:  
61: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
62:  
63: devbad.out:  
64:  
65: netbad.out:  
66:  
67: mergenet.out:  
68:  
69: termbad.out:  
70:  
71: prunenet.out:  
72:  
73: prunedev.out:  
74:  
75: audit.out:  
76:  
77:  
78: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
79:  
80: devbad.out:  
81:  
82: netbad.out:  
83:  
84: mergenet.out:  
85:  
86: termbad.out:  
87:  
88: prunenet.out:  
89:  
90: prunedev.out:  
91:  
92: audit.out:
```

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      11      nets  
12:      6       terminals  
13:      6       pmos  
14:      6       nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      11      nets  
19:      6       terminals  
20:      6       pmos  
21:      6       nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N5      N1      gnd!  
26:     N8      N2      input1  
27:     N7      N3      input2  
28:     N6      N4      input3  
29:     N9      N6      output  
30:     N10     N0      vdd!  
31:  
32: Devices in the netlist but not in the rules:  
33:     pcapacitor  
34: Devices in the rules but not in the netlist:  
35:     cap nfet pfet nmos4 pmos4  
36:  
37: The net-lists match.  
38:  
39:             layout schematic  
40:                 instances  
41:     un-matched      0      0  
42:     rewired        0      0  
43:     size errors    0      0  
44:     pruned         0      0  
45:     active         12     12  
46:     total          12     12  
47:  
48:                 nets  
49:     un-matched      0      0  
50:     merged          0      0  
51:     pruned          0      0  
52:     active          11     11  
53:     total           11     11  
54:  
55:             terminals  
56:     un-matched      0      0  
57:     matched but
```

	58:	different type	0	0
	59:	total	6	6
	60:			
	61:			
	62:	Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic		
	ic			
	63:			
	64:	devbad.out:		
	65:			
	66:	netbad.out:		
	67:			
	68:	mergenet.out:		
	69:			
	70:	termbad.out:		
	71:			
	72:	prunenet.out:		
	73:			
	74:	prunedev.out:		
	75:			
	76:	audit.out:		
	77:			
	78:			
	79:	Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout		
	80:			
	81:	devbad.out:		
	82:			
	83:	netbad.out:		
	84:			
	85:	mergenet.out:		
	86:			
	87:	termbad.out:		
	88:			
	89:	prunenet.out:		
	90:			
	91:	prunedev.out:		
	92:			
	93:	audit.out:		

Layout vs Schematic
buff.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout/netlist  
10:    count  
11:      5          nets  
12:      4          terminals  
13:      2          pmos  
14:      2          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic/netlist  
17:    count  
18:      5          nets  
19:      4          terminals  
20:      2          pmos  
21:      2          nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N1      N1      gnd!  
26:     N3      N2      input  
27:     N2      N4      output  
28:     N4      N0      vdd!  
29:  
30: Devices in the netlist but not in the rules:  
31:     pcapacitor  
32: Devices in the rules but not in the netlist:  
33:     cap nfet pfet nmos4 pmos4  
34:  
35: The net-lists match.  
36:  
37:           layout schematic  
38:           instances  
39:     un-matched      0      0  
40:     rewired        0      0  
41:     size errors    0      0  
42:     pruned         0      0  
43:     active         4      4  
44:     total          4      4  
45:  
46:           nets  
47:     un-matched    0      0  
48:     merged         0      0  
49:     pruned         0      0  
50:     active         5      5  
51:     total          5      5  
52:  
53:           terminals  
54:     un-matched    0      0  
55:     matched but  
56:     different type 0      0  
57:     total          4      4  
58:  
59:  
60: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
61:  
62: devbad.out:  
63:  
64: netbad.out:  
65:  
66: mergenet.out:  
67:  
68: termbad.out:  
69:  
70: prunenet.out:  
71:  
72: prunedev.out:  
73:  
74: audit.out:  
75:  
76:  
77: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
78:  
79: devbad.out:  
80:  
81: netbad.out:  
82:  
83: mergenet.out:  
84:  
85: termbad.out:  
86:  
87: prunenet.out:  
88:  
89: prunedev.out:  
90:  
91: audit.out:
```

busy1.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout/netlist  
10:    count  
11:      20      nets  
12:      8       terminals  
13:      13      pmos  
14:      13      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic/netlist  
17:    count  
18:      20      nets  
19:      8       terminals  
20:      13      pmos  
21:      13      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N22   N2    busy  
26: N20   N1    gnd!  
27: N27   N7    q  
28: N25   N8    reset  
29: N24   N11   s0_en  
30: N26   N10   start  
31: N21   N9    t6  
32: N23   N0    vdd!  
33:  
34: Devices in the netlist but not in the rules:  
35:     pcapacitor  
36: Devices in the rules but not in the netlist:  
37:     cap nfet pfet nmos4 pmos4  
38:  
39: The net-lists match.  
40:  
41:           layout schematic  
42:           instances  
43: un-matched      0      0  
44: rewired        0      0  
45: size errors    0      0  
46: pruned         0      0  
47: active          26     26  
48: total           26     26  
49:  
50:           nets  
51: un-matched      0      0  
52: merged          0      0  
53: pruned          0      0  
54: active          20     20  
55: total           20     20  
56:  
57:           terminals
```

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      20      nets  
12:      9       terminals  
13:      13      pmos  
14:      13      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      20      nets  
19:      9       terminals  
20:      13      pmos  
21:      13      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N27   N11   cpu_rd_wrn_reg_b  
26: N20   N1    gnd!  
27: N19   N6    output_en  
28: N26   N2    rw_cach1  
29: N21   N9    s18_en  
30: N22   N7    s1_en_rh  
31: N24   N8    t25  
32: N25   N0    vdd!  
33: N23   N10   write_cache  
34:  
35: Devices in the netlist but not in the rules:  
36:     pcapacitor  
37: Devices in the rules but not in the netlist:  
38:     cap nfet pfet nmos4 pmos4  
39:  
40: The net-lists match.  
41:  
42:           layout schematic  
43:           instances  
44: un-matched      0      0  
45: rewired        0      0  
46: size errors    0      0  
47: pruned         0      0  
48: active          26     26  
49: total           26     26  
50:  
51:           nets  
52: un-matched      0      0  
53: merged          0      0  
54: pruned          0      0  
55: active          20     20  
56: total           20     20  
57:  
58:                                     terminals  
59: un-matched      0      0  
60: matched but    0      0  
61: different type  0      0  
62: total           9      9  
63:  
64:  
65: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
66:  
67: devbad.out:  
68:  
69: netbad.out:  
70:  
71: mergenet.out:  
72:  
73: termbad.out:  
74:  
75: prunenet.out:  
76:  
77: prunedev.out:  
78:  
79: audit.out:  
80:  
81:  
82: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
83:  
84: devbad.out:  
85:  
86: netbad.out:  
87:  
88: mergenet.out:  
89:  
90: termbad.out:  
91:  
92: prunenet.out:  
93:  
94: prunedev.out:  
95:  
96: audit.out:
```

Layout vs Schematic
cache_32b.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      420      nets  
12:      24      terminals  
13:      460      pmos  
14:      460      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      420      nets  
19:      24      terminals  
20:      460      pmos  
21:      460      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N403   N17   blk_en  
26: N410   N13   by_en0  
27: N409   N14   by_en1  
28: N408   N15   by_en2  
29: N407   N16   by_en3  
30: N413   N12   din<0>  
31: N411   N11   din<1>  
32: N405   N10   din<2>  
33: N401   N9    din<3>  
34: N399   N8    din<4>  
35: N397   N7    din<5>  
36: N419   N6    din<6>  
37: N416   N5    din<7>  
38: N396   N26   dout<0>  
39: N417   N25   dout<1>  
40: N415   N24   dout<2>  
41: N414   N23   dout<3>  
42: N412   N22   dout<4>  
43: N406   N21   dout<5>  
44: N402   N20   dout<6>  
45: N400   N19   dout<7>  
46: N398   N1    gnd!  
47: N418   N18   rw_en  
48: N404   N0    vdd!  
49:  
50: Devices in the netlist but not in the rules:  
51:      pcapacitor  
52: Devices in the rules but not in the netlist:  
53:      cap nfet pfet nmos4 pmos4  
54:  
55: The net-lists match.  
56:  
57: layout schematic
```

		instances	
58:	un-matched	0	0
59:	rewired	0	0
60:	size errors	0	0
61:	pruned	0	0
62:	active	920	920
63:	total	920	920
64:			
65:			
66:		nets	
67:	un-matched	0	0
68:	merged	0	0
69:	pruned	0	0
70:	active	420	420
71:	total	420	420
72:			
73:		terminals	
74:	un-matched	0	0
75:	matched but		
76:	different type	0	0
77:	total	24	24
78:			
79:		80: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat	
ic			
81:			
82:	devbad.out:		
83:			
84:	netbad.out:		
85:			
86:	mergenet.out:		
87:			
88:	termbad.out:		
89:			
90:	prunenet.out:		
91:			
92:	prunedev.out:		
93:			
94:	audit.out:		
95:			
96:			
97:	Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout		
98:			
99:	devbad.out:		
100:			
101:	netbad.out:		
102:			
103:	mergenet.out:		
104:			
105:	termbad.out:		
106:			
107:	prunenet.out:		
108:			
109:	prunedev.out:		
110:			
111:	audit.out:		

Layout vs Schematic
cache_4b.txt

```
1: #@(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      60      nets  
12:      12      terminals  
13:      56      pmos  
14:      56      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      60      nets  
19:      12      terminals  
20:      56      pmos  
21:      56      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N55   N11   din<0>  
26: N54   N10   din<1>  
27: N53   N9    din<2>  
28: N50   N4    din<3>  
29: N48   N8    dout<0>  
30: N58   N7    dout<1>  
31: N57   N6    dout<2>  
32: N56   N5    dout<3>  
33: N51   N3    en  
34: N49   N1    gnd!  
35: N59   N2    rw_en  
36: N52   N0    vdd!  
37:  
38: Devices in the netlist but not in the rules:  
39:     pcapacitor  
40: Devices in the rules but not in the netlist:  
41:     cap nfet pfet nmos4 pmos4  
42:  
43: The net-lists match.  
44:  
45:             layout schematic  
46:                 instances  
47:     un-matched      0      0  
48:     rewired        0      0  
49:     size errors    0      0  
50:     pruned         0      0  
51:     active         112    112  
52:     total          112    112  
53:  
54:             nets  
55:     un-matched    0      0  
56:     merged         0      0  
57:     pruned         0      0  
58:     active         60     60  
59:     total          60     60  
60:  
61:                     terminals  
62:     un-matched    0      0  
63:     matched but   0      0  
64:     different type 0      0  
65:     total          12     12  
66:  
67:  
68: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
69:  
70: devbad.out:  
71:  
72: netbad.out:  
73:  
74: mergenet.out:  
75:  
76: termbad.out:  
77:  
78: prunenet.out:  
79:  
80: prunedev.out:  
81:  
82: audit.out:  
83:  
84:  
85: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
86:  
87: devbad.out:  
88:  
89: netbad.out:  
90:  
91: mergenet.out:  
92:  
93: termbad.out:  
94:  
95: prunenet.out:  
96:  
97: prunedev.out:  
98:  
99: audit.out:
```

Layout vs Schematic
cache_8b.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/h  
ome/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      116      nets  
12:      20       terminals  
13:      112      pmos  
14:      112      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      116      nets  
19:      20       terminals  
20:      112      pmos  
21:      112      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N109   N19   din<0>  
26: N107   N18   din<1>  
27: N105   N9    din<2>  
28: N101   N8    din<3>  
29: N99    N7    din<4>  
30: N97    N6    din<5>  
31: N115   N5    din<6>  
32: N112   N4    din<7>  
33: N96    N17   dout<0>  
34: N113   N16   dout<1>  
35: N111   N15   dout<2>  
36: N110   N14   dout<3>  
37: N108   N13   dout<4>  
38: N106   N12   dout<5>  
39: N102   N11   dout<6>  
40: N100   N10   dout<7>  
41: N103   N3    en  
42: N98    N1    gnd!  
43: N114   N2    rw_en  
44: N104   N0    vdd!  
45:  
46: Devices in the netlist but not in the rules:  
47:     pcapacitor  
48: Devices in the rules but not in the netlist:  
49:     cap nfet pfet nmos4 pmos4  
50:  
51: The net-lists match.  
52:  
53:             layout schematic  
54:                 instances  
55:     un-matched      0      0  
56:     rewired        0      0  
57:     size errors    0      0  
58:     pruned         0      0  
59:     active          224   224  
60:     total           224   224  
61:  
62:                     nets  
63:     un-matched      0      0  
64:     merged          0      0  
65:     pruned          0      0  
66:     active          116   116  
67:     total           116   116  
68:  
69:                     terminals  
70:     un-matched      0      0  
71:     matched but  
72:     different type  2      2  
73:     total           20    20  
74:  
75:  
76: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
77:  
78: devbad.out:  
79:  
80: netbad.out:  
81:  
82: mergenet.out:  
83:  
84: termbad.out:  
85: ? Terminal en's type in the schematic: input, in the layout: output  
86: ? Terminal rw_en's type in the schematic: input, in the layout: output  
87:  
88: prunenet.out:  
89:  
90: prunedev.out:  
91:  
92: audit.out:  
93:  
94:  
95: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
96:  
97: devbad.out:  
98:  
99: netbad.out:  
100:  
101: mergenet.out:  
102:  
103: termbad.out:  
104: ? Terminal en's type in the layout: output, in the schematic: input  
105: ? Terminal rw_en's type in the layout: output, in the schematic: input  
106:  
107: prunenet.out:  
108:  
109: prunedev.out:  
110:  
111: audit.out:
```

Layout vs Schematic
cache_cell.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      18      nets  
12:      6      terminals  
13:      14      pmos  
14:      14      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      18      nets  
19:      6      terminals  
20:      14      pmos  
21:      14      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N14      N6      din  
26: N13      N10     dout  
27: N15      N8      en  
28: N12      N1      gnd!  
29: N17      N9      rw_en  
30: N16      N0      vdd!  
31:  
32: Devices in the netlist but not in the rules:  
33:      pcapacitor  
34: Devices in the rules but not in the netlist:  
35:      cap nfet pfet nmos4 pmos4  
36:  
37: The net-lists match.  
38:  
39:          layout schematic  
40:          instances  
41: un-matched      0      0  
42: rewired        0      0  
43: size errors    0      0  
44: pruned         0      0  
45: active         28     28  
46: total          28     28  
47:  
48:          nets  
49: un-matched      0      0  
50: merged          0      0  
51: pruned         0      0  
52: active         18     18  
53: total          18     18  
54:  
55:          terminals  
56: un-matched      0      0  
57: matched but
```

	58:	different type	0	0
	59:	total	6	6
	60:			
	61:			
	62:	Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic		
	63:			
	64:	devbad.out:		
	65:			
	66:	netbad.out:		
	67:			
	68:	mergenet.out:		
	69:			
	70:	termbad.out:		
	71:			
	72:	prunenet.out:		
	73:			
	74:	prunedev.out:		
	75:			
	76:	audit.out:		
	77:			
	78:			
	79:	Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout		
	80:			
	81:	devbad.out:		
	82:			
	83:	netbad.out:		
	84:			
	85:	mergenet.out:		
	86:			
	87:	termbad.out:		
	88:			
	89:	prunenet.out:		
	90:			
	91:	prunedev.out:		
	92:			
	93:	audit.out:		

Layout vs Schematic
cache.txt

```
1: #@($CD$ LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      3199      nets  
12:      31      terminals  
13:      3680      pmos  
14:      3680      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      3199      nets  
19:      31      terminals  
20:      3680      pmos  
21:      3680      nmos  
22:  
23: Terminal correspondence points  
24: N3177  N14  blk_en0  
25: N3176  N15  blk_en1  
26: N3174  N16  blk_en2  
27: N3173  N17  blk_en3  
28: N3172  N18  blk_en4  
29: N3171  N19  blk_en5  
30: N3170  N20  blk_en6  
31: N3169  N21  blk_en7  
32: N3189  N10  by_en0  
33: N3188  N11  by_en1  
34: N3187  N12  by_en2  
35: N3186  N13  by_en3  
36: N3192  N9   din<0>  
37: N3190  N8   din<1>  
38: N3184  N7   din<2>  
39: N3181  N6   din<3>  
40: N3179  N5   din<4>  
41: N3175  N4   din<5>  
42: N3198  N3   din<6>  
43: N3195  N2   din<7>  
44: N3168  N30  dout<0>  
45: N3196  N29  dout<1>  
46: N3194  N28  dout<2>  
47: N3193  N27  dout<3>  
48: N3191  N26  dout<4>  
49: N3185  N25  dout<5>  
50: N3182  N24  dout<6>  
51: N3180  N23  dout<7>  
52: N3178  N1   gnd!  
53: N3197  N22  rw_en  
54: N3183  N0   vdd!  
55:  
56: Devices in the rules but not in the netlist:  
58:     cap nfet pfet nmos4 pmos4  
59:  
60: The net-lists match.  
61:  
62:                                     layout schematic  
63:                                     instances  
64:     un-matched                      0      0  
65:     rewired                         0      0  
66:     size errors                     0      0  
67:     pruned                          0      0  
68:     active                          7360   7360  
69:     total                           7360   7360  
70:  
71:                                     nets  
72:     un-matched                     0      0  
73:     merged                          0      0  
74:     pruned                         0      0  
75:     active                         3199   3199  
76:     total                          3199   3199  
77:  
78:                                     terminals  
79:     un-matched                     0      0  
80:     matched but                   0      0  
81:     different type                0      0  
82:     total                          31      31  
83:  
84:  
85: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
86:  
87: devbad.out:  
88:  
89: netbad.out:  
90:  
91: mergenet.out:  
92:  
93: termbad.out:  
94:  
95: prunenet.out:  
96:  
97: prunedev.out:  
98:  
99: audit.out:  
100:  
101:  
102: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
103:  
104: devbad.out:  
105:  
106: netbad.out:  
107:  
108: mergenet.out:  
109:  
110: termbad.out:  
111:  
112: prunenet.out:  
113:  
114: prunedev.out:  
115:  
116: audit.out:
```

Layout vs Schematic
chip.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      4268      nets  
12:      41       terminals  
13:      4903      pmos  
14:      4903      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      4268      nets  
19:      41       terminals  
20:      4903      pmos  
21:      4903      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N4240  N133  busy  
26: N4250  N143  clk  
27: N4251  N162  cpu_add<0>  
28: N4247  N169  cpu_add<1>  
29: N4241  N168  cpu_add<2>  
30: N4235  N167  cpu_add<3>  
31: N4231  N166  cpu_add<4>  
32: N4227  N165  cpu_add<5>  
33: N4262  N164  cpu_add<6>  
34: N4258  N163  cpu_add<7>  
35: N4244  N152  cpu_data<0>  
36: N4238  N153  cpu_data<1>  
37: N4233  N154  cpu_data<2>  
38: N4228  N155  cpu_data<3>  
39: N4266  N156  cpu_data<4>  
40: N4260  N157  cpu_data<5>  
41: N4256  N158  cpu_data<6>  
42: N4253  N159  cpu_data<7>  
43: N4239  N145  cpu_rd_wrn  
44: N4245  N63   gnd  
45: N4230  N1   gnd!  
46: N4265  N124  mem_add<0>  
47: N4259  N125  mem_add<1>  
48: N4255  N126  mem_add<2>  
49: N4252  N127  mem_add<3>  
50: N4248  N128  mem_add<4>  
51: N4242  N129  mem_add<5>  
52: N4236  N130  mem_add<6>  
53: N4232  N131  mem_add<7>  
54: N4229  N134  mem_data<0>  
55: N4267  N135  mem_data<1>  
56: N4261  N136  mem_data<2>  
57: N4257  N137  mem_data<3>  
58:      N4254  N138  mem_data<4>  
59:      N4249  N139  mem_data<5>  
60:      N4243  N140  mem_data<6>  
61:      N4237  N141  mem_data<7>  
62:      N4234  N132  mem_en  
63:      N4263  N142  reset  
64:      N4264  N144  start  
65:      N4246  N0    vdd!  
66:  
67: Devices in the netlist but not in the rules:  
68:      pcapacitor  
69: Devices in the rules but not in the netlist:  
70:      cap nfet pfet nmos4 pmos4  
71:  
72: The net-lists match.  
73:  
74:                                     layout schematic  
75:                                     instances  
76: un-matched                      0      0  
77: rewired                         0      0  
78: size errors                     0      0  
79: pruned                          0      0  
80: active                          9806   9806  
81: total                           9806   9806  
82:  
83:                                     nets  
84: un-matched                      0      0  
85: merged                          0      0  
86: pruned                          0      0  
87: active                          4268   4268  
88: total                           4268   4268  
89:  
90:                                     terminals  
91: un-matched                      0      0  
92: matched but                     0      0  
93: different type                  0      0  
94: total                           41    41  
95:  
96:  
97: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
98:  
99: devbad.out:  
100:  
101: netbad.out:  
102:  
103: mergenet.out:  
104:  
105: termbad.out:  
106:  
107: prunenet.out:  
108:  
109: prunedev.out:  
110:  
111: audit.out:  
112:  
113:  
114: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
115:  
116: devbad.out:  
117:  
118: netbad.out:
```

```
119:  
120: mergenet.out:  
121:  
122: termbad.out:  
123:  
124: prunenet.out:  
125:  
126: prunedev.out:  
127:  
128: audit.out:
```

cmprtr2.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      45      nets  
12:      10      terminals  
13:      36      pmos  
14:      36      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      45      nets  
19:      10      terminals  
20:      36      pmos  
21:      36      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N44   N4    eq1  
26: N36   N1    gnd!  
27: N43   N13   input1<0>  
28: N42   N12   input1<1>  
29: N41   N3    input1<2>  
30: N39   N2    input1<3>  
31: N38   N11   input2<0>  
32: N37   N10   input2<1>  
33: N35   N9    input2<2>  
34: N40   N0    vdd!  
35:  
36: Devices in the rules but not in the netlist:  
37:     cap nfet pfet nmos4 pmos4  
38:  
39: The net-lists match.  
40:  
41:             layout schematic  
42:                 instances  
43:     un-matched      0      0  
44:     rewired        0      0  
45:     size errors    0      0  
46:     pruned         0      0  
47:     active         72     72  
48:     total          72     72  
49:  
50:             nets  
51:     un-matched    0      0  
52:     merged         0      0  
53:     pruned         0      0  
54:     active         45     45  
55:     total          45     45  
56:  
57:             terminals
```

Layout vs Schematic
col_sel_tgcm.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      50      nets  
12:      27      terminals  
13:      35      pmos  
14:      35      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      50      nets  
19:      27      terminals  
20:      35      pmos  
21:      35      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N41   N10   blk_en0  
26: N44   N4    blk_en0_o  
27: N40   N11   blk_en1  
28: N56   N5    blk_en1_o  
29: N39   N12   blk_en2  
30: N45   N6    blk_en2_o  
31: N38   N13   blk_en3  
32: N57   N7    blk_en3_o  
33: N37   N14   blk_en4  
34: N47   N25   blk_en4_o  
35: N36   N15   blk_en5  
36: N58   N26   blk_en5_o  
37: N35   N16   blk_en6  
38: N48   N27   blk_en6_o  
39: N34   N17   blk_en7  
40: N59   N28   blk_en7_o  
41: N53   N22   cpu_add_reg<0>  
42: N51   N21   cpu_add_reg<1>  
43: N46   N20   cpu_add_reg<2>  
44: N54   N24   din<0>  
45: N52   N23   din<1>  
46: N50   N9    din<2>  
47: N43   N8    din<3>  
48: N42   N1    gnd!  
49: N60   N18   reset  
50: N55   N19   reset_b  
51: N49   N0    vdd!  
52:  
53: Devices in the netlist but not in the rules:  
54:     pcapacitor  
55: Devices in the rules but not in the netlist:  
56:     cap nfet pfet nmos4 pmos4  
57:
```

	layout	schematic	instances
60:			
61: un-matched	0	0	
63: rewired	0	0	
64: size errors	0	0	
65: pruned	0	0	
66: active	70	70	
67: total	70	70	
68:			
69:			nets
70: un-matched	0	0	
71: merged	0	0	
72: pruned	0	0	
73: active	50	50	
74: total	50	50	
75:			
76:			terminals
77: un-matched	0	0	
78: matched but			
79: different type	0	0	
80: total	27	27	
81:			
82:			
83: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat			
ic			
84:			
85: devbad.out:			
86:			
87: netbad.out:			
88:			
89: mergenet.out:			
90:			
91: termbad.out:			
92:			
93: prunenet.out:			
94:			
95: prunedev.out:			
96:			
97: audit.out:			
98:			
99:			
100: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout			
101:			
102: devbad.out:			
103:			
104: netbad.out:			
105:			
106: mergenet.out:			
107:			
108: termbad.out:			
109:			
110: prunenet.out:			
111:			
112: prunedev.out:			
113:			
114: audit.out:			

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      31      nets  
12:      9       terminals  
13:      26      pmos  
14:      26      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      31      nets  
19:      9       terminals  
20:      26      pmos  
21:      26      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N25   N5    en  
26: N22   N1    gnd!  
27: N24   N3    s0  
28: N23   N4    s1  
29: N26   N0    vdd!  
30: N30   N6    y0  
31: N29   N7    y1  
32: N28   N8    y2  
33: N27   N9    y3  
34:  
35: Devices in the rules but not in the netlist:  
36:     cap nfet pfet nmos4 pmos4  
37:  
38: The net-lists match.  
39:  
40:             layout schematic  
41:                 instances  
42: un-matched      0      0  
43: rewired         0      0  
44: size errors     0      0  
45: pruned          0      0  
46: active          52     52  
47: total           52     52  
48:  
49:             nets  
50: un-matched      0      0  
51: merged           0      0  
52: pruned          0      0  
53: active          31     31  
54: total           31     31  
55:  
56:             terminals  
57: un-matched      0      0  
58:     matched but  
59:     different type      0      0  
60:     total           9      9  
61:  
62:  
63: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
ic  
64:  
65: devbad.out:  
66:  
67: netbad.out:  
68:  
69: mergenet.out:  
70:  
71: termbad.out:  
72:  
73: prunenet.out:  
74:  
75: prunedev.out:  
76:  
77: audit.out:  
78:  
79:  
80: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
81:  
82: devbad.out:  
83:  
84: netbad.out:  
85:  
86: mergenet.out:  
87:  
88: termbad.out:  
89:  
90: prunenet.out:  
91:  
92: prunedev.out:  
93:  
94: audit.out:
```

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      58      nets  
12:      13      terminals  
13:      53      pmos  
14:      53      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      58      nets  
19:      13      terminals  
20:      53      pmos  
21:      53      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N45   N1      gnd!  
26: N48   N13     s0  
27: N47   N12     s1  
28: N46   N11     s2  
29: N49   N0      vdd!  
30: N57   N10     y0  
31: N56   N9      y1  
32: N55   N8      y2  
33: N54   N7      y3  
34: N53   N6      y4  
35: N52   N5      y5  
36: N51   N4      y6  
37: N50   N2      y7  
38:  
39: Devices in the rules but not in the netlist:  
40:     cap nfet pfet nmos4 pmos4  
41:  
42: The net-lists match.  
43:  
44:             layout schematic  
45:                 instances  
46: un-matched      0      0  
47: rewired         0      0  
48: size errors     0      0  
49: pruned          0      0  
50: active          106    106  
51: total           106    106  
52:  
53:             nets  
54: un-matched      0      0  
55: merged           0      0  
56: pruned          0      0  
57: active          58     58  
58:     total          58     58  
59:  
60:                     terminals  
61: un-matched      0      0  
62: matched but  
63: different type  0      0  
64: total           13    13  
65:  
66:  
67: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
68:  
69: devbad.out:  
70:  
71: netbad.out:  
72:  
73: mergenet.out:  
74:  
75: termbad.out:  
76:  
77: prunenet.out:  
78:  
79: prunedev.out:  
80:  
81: audit.out:  
82:  
83:  
84: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
85:  
86: devbad.out:  
87:  
88: netbad.out:  
89:  
90: mergenet.out:  
91:  
92: termbad.out:  
93:  
94: prunenet.out:  
95:  
96: prunedev.out:  
97:  
98: audit.out:
```

dff_8b.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      67      nets  
12:      19      terminals  
13:      72      pmos  
14:      72      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      67      nets  
19:      19      terminals  
20:      72      pmos  
21:      72      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N58   N2      clk  
26: N65   N16     d<0>  
27: N63   N15     d<1>  
28: N61   N14     d<2>  
29: N59   N13     d<3>  
30: N56   N12     d<4>  
31: N53   N11     d<5>  
32: N52   N10     d<6>  
33: N50   N9      d<7>  
34: N49   N0      gnd!  
35: N51   N18     q<0>  
36: N48   N17     q<1>  
37: N66   N8      q<2>  
38: N64   N7      q<3>  
39: N62   N6      q<4>  
40: N60   N5      q<5>  
41: N57   N4      q<6>  
42: N54   N3      q<7>  
43: N55   N1      vdd!  
44:  
45: Devices in the netlist but not in the rules:  
46:     pcapacitor  
47: Devices in the rules but not in the netlist:  
48:     cap nfet pfet nmos4 pmos4  
49:  
50: The net-lists match.  
51:  
52:           layout schematic  
53:                   instances  
54:     un-matched      0      0  
55:     rewired         0      0  
56:     size errors     0      0  
57:     pruned          0      0  
58:     active          144    144  
59:     total           144    144  
60:  
61:           nets  
62:     un-matched      0      0  
63:     merged          0      0  
64:     pruned          0      0  
65:     active          67    67  
66:     total           67    67  
67:  
68:           terminals  
69:     un-matched      0      0  
70:     matched but    0      0  
71:     different type 0      0  
72:     total           19    19  
73:  
74:  
75: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
ic  
76: devbad.out:  
78:  
79: netbad.out:  
80:  
81: mergenet.out:  
82:  
83: termbad.out:  
84:  
85: prunenet.out:  
86:  
87: prunedev.out:  
88:  
89: audit.out:  
90:  
91:  
92: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
93:  
94: devbad.out:  
95:  
96: netbad.out:  
97:  
98: mergenet.out:  
99:  
100: termbad.out:  
101:  
102: prunenet.out:  
103:  
104: prunedev.out:  
105:  
106: audit.out:
```

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      11      nets  
12:      5       terminals  
13:      9       pmos  
14:      9       nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      11      nets  
19:      5       terminals  
20:      9       pmos  
21:      9       nmos  
22:  
23:  
24:     Terminal correspondence points  
25:   N9      N5      clk  
26:   N7      N9      d  
27:   N6      N0      gnd!  
28:   N10     N6      q  
29:   N8      N1      vdd!  
30:  
31: Devices in the rules but not in the netlist:  
32:     cap nfet pfet nmos4 pmos4  
33:  
34: The net-lists match.  
35:  
36:           layout schematic  
37:           instances  
38: un-matched      0      0  
39: rewired         0      0  
40: size errors     0      0  
41: pruned          0      0  
42: active          18     18  
43: total           18     18  
44:  
45:           nets  
46: un-matched      0      0  
47: merged           0      0  
48: pruned          0      0  
49: active          11     11  
50: total           11     11  
51:  
52:           terminals  
53: un-matched      0      0  
54: matched but  
55: different type   0      0  
56: total           5      5  
57:
```

```
58:  
59: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
60:  
61: devbad.out:  
62:  
63: netbad.out:  
64:  
65: mergenet.out:  
66:  
67: termbad.out:  
68:  
69: prunenet.out:  
70:  
71: prunedev.out:  
72:  
73: audit.out:  
74:  
75:  
76: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
77:  
78: devbad.out:  
79:  
80: netbad.out:  
81:  
82: mergenet.out:  
83:  
84: termbad.out:  
85:  
86: prunenet.out:  
87:  
88: prunedev.out:  
89:  
90: audit.out:
```

Layout vs Schematic

dlatch.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      8          nets  
12:      6          terminals  
13:      5          pmos  
14:      5          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      8          nets  
19:      6          terminals  
20:      5          pmos  
21:      5          nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N6      N4      clk  
26:     N2      N7      d  
27:     N3      N0      gnd!  
28:     N7      N3      q  
29:     N4      N6      qb  
30:     N5      N1      vdd!  
31:  
32: Devices in the netlist but not in the rules:  
33:     pcapacitor  
34: Devices in the rules but not in the netlist:  
35:     cap nfet pfet nmos4 pmos4  
36:  
37: The net-lists match.  
38:  
39:             layout  schematic  
40:                 instances  
41:     un-matched      0      0  
42:     rewired        0      0  
43:     size errors    0      0  
44:     pruned         0      0  
45:     active         10     10  
46:     total          10     10  
47:  
48:             nets  
49:     un-matched      0      0  
50:     merged          0      0  
51:     pruned          0      0  
52:     active          8      8  
53:     total           8      8  
54:  
55:             terminals  
56:     un-matched      0      0  
57:     matched but
```

Layout vs Schematic
encoder4_2.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      11      nets  
12:      5       terminals  
13:      6       pmos  
14:      6       nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      11      nets  
19:      7       terminals  
20:      6       pmos  
21:      6       nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N8      N6      i1  
26:     N7      N5      i2  
27:     N6      N4      i3  
28:     N10     N3      o1  
29:     N9      N2      o2  
30:  
31: Devices in the netlist but not in the rules:  
32:     pcapacitor  
33: Devices in the rules but not in the netlist:  
34:     cap nfet pfet nmos4 pmos4  
35:  
36: The net-lists match.  
37:  
38:           layout schematic  
39:           instances  
40:     un-matched      0      0  
41:     rewired        0      0  
42:     size errors    0      0  
43:     pruned         0      0  
44:     active         12     12  
45:     total          12     12  
46:  
47:           nets  
48:     un-matched      0      0  
49:     merged          0      0  
50:     pruned          0      0  
51:     active          11     11  
52:     total           11     11  
53:  
54:           terminals  
55:     un-matched      0      0  
56:     matched but  
57:     different type  0      0  
58:           total      5      7  
59:  
60:  
61: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
62:  
63: devbad.out:  
64:  
65: netbad.out:  
66:  
67: mergenet.out:  
68:  
69: termbad.out:  
70:  
71: prunenet.out:  
72:  
73: prunedev.out:  
74:  
75: audit.out:  
76:  
77:  
78: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
79:  
80: devbad.out:  
81:  
82: netbad.out:  
83:  
84: mergenet.out:  
85:  
86: termbad.out:  
87:  
88: prunenet.out:  
89:  
90: prunedev.out:  
91:  
92: audit.out:
```

Layout vs Schematic
hit_miss.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      73      nets  
12:      15      terminals  
13:      62      pmos  
14:      62      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      73      nets  
19:      15      terminals  
20:      62      pmos  
21:      62      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N66   N15   cache_tag_valid<0>  
26: N64   N14   cache_tag_valid<1>  
27: N62   N13   cache_tag_valid<2>  
28: N61   N12   cache_tag_valid<3>  
29: N70   N8    cpu_rw  
30: N72   N11   cpu_tag<0>  
31: N68   N10   cpu_tag<1>  
32: N65   N9    cpu_tag<2>  
33: N58   N1    gnd!  
34: N71   N2    r_h  
35: N69   N4    r_m  
36: N67   N6    sm_en  
37: N63   N0    vdd!  
38: N60   N3    w_h  
39: N59   N5    w_m  
40:  
41: Devices in the netlist but not in the rules:  
42:     pcapacitor  
43: Devices in the rules but not in the netlist:  
44:     cap nfet pfet nmos4 pmos4  
45:  
46: The net-lists match.  
47:  
48:             layout schematic  
49:                 instances  
50: un-matched      0      0  
51: rewired        0      0  
52: size errors    0      0  
53: pruned         0      0  
54: active          124    124  
55: total           124    124  
56:  
57:             nets  
58:      un-matched      0      0  
59:      merged          0      0  
60:      pruned          0      0  
61:      active          73    73  
62:      total           73    73  
63:  
64:      terminals  
65:      un-matched      0      0  
66:      matched but  
67:      different type  0      0  
68:      total            15    15  
69:  
70:  
71: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
72:  
73: devbad.out:  
74:  
75: netbad.out:  
76:  
77: mergenet.out:  
78:  
79: termbad.out:  
80:  
81: prunenet.out:  
82:  
83: prunedev.out:  
84:  
85: audit.out:  
86:  
87:  
88: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
89:  
90: devbad.out:  
91:  
92: netbad.out:  
93:  
94: mergenet.out:  
95:  
96: termbad.out:  
97:  
98: prunenet.out:  
99:  
100: prunedev.out:  
101:  
102: audit.out:
```

Layout vs Schematic
inverter.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      4          nets  
12:      4          terminals  
13:      1          pmos  
14:      1          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      4          nets  
19:      4          terminals  
20:      1          pmos  
21:      1          nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N0      N1      gnd!  
26:     N2      N2      input  
27:     N1      N5      output  
28:     N3      N0      vdd!  
29:  
30: Devices in the netlist but not in the rules:  
31:     pcapacitor  
32: Devices in the rules but not in the netlist:  
33:     cap nfet pfet nmos4 pmos4  
34:  
35: The net-lists match.  
36:  
37:           layout schematic  
38:           instances  
39:     un-matched      0      0  
40:     rewired        0      0  
41:     size errors    0      0  
42:     pruned         0      0  
43:     active         2      2  
44:     total          2      2  
45:  
46:           nets  
47:     un-matched      0      0  
48:     merged          0      0  
49:     pruned          0      0  
50:     active          4      4  
51:     total          4      4  
52:  
53:           terminals  
54:     un-matched      0      0  
55:     matched but  
56:     different type  0      0  
57:     total          4      4  
58:  
59:  
60: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
61:  
62: devbad.out:  
63:  
64: netbad.out:  
65:  
66: mergenet.out:  
67:  
68: termbad.out:  
69:  
70: prunenet.out:  
71:  
72: prunedev.out:  
73:  
74: audit.out:  
75:  
76:  
77: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
78:  
79: devbad.out:  
80:  
81: netbad.out:  
82:  
83: mergenet.out:  
84:  
85: termbad.out:  
86:  
87: prunenet.out:  
88:  
89: prunedev.out:  
90:  
91: audit.out:
```

Layout vs Schematic
mux2x1_8b.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      35      nets  
12:      27      terminals  
13:      24      pmos  
14:      24      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      35      nets  
19:      27      terminals  
20:      24      pmos  
21:      24      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N15      N23      a<0>  
26: N13      N22      a<1>  
27: N9       N21      a<2>  
28: N33      N20      a<3>  
29: N30      N19      a<4>  
30: N27      N18      a<5>  
31: N24      N17      a<6>  
32: N21      N16      a<7>  
33: N31      N24      b<0>  
34: N28      N15      b<1>  
35: N25      N14      b<2>  
36: N22      N13      b<3>  
37: N17      N12      b<4>  
38: N14      N11      b<5>  
39: N12      N10      b<6>  
40: N8       N9       b<7>  
41: N11      N1       gnd!  
42: N19      N2       sel  
43: N20      N0       vdd!  
44: N10      N26      x<0>  
45: N34      N25      x<1>  
46: N32      N8       x<2>  
47: N29      N7       x<3>  
48: N26      N6       x<4>  
49: N23      N5       x<5>  
50: N18      N4       x<6>  
51: N16      N3       x<7>  
52:  
53: Devices in the rules but not in the netlist:  
54:     cap nfet pfet nmos4 pmos4  
55:  
56: The net-lists match.  
57:
```

		layout	schematic
		instances	
58:			
59:	un-matched	0	0
60:	rewired	0	0
61:	size errors	0	0
62:	pruned	0	0
63:	active	48	48
64:	total	48	48
65:			
66:			
67:		nets	
68:	un-matched	0	0
69:	merged	0	0
70:	pruned	0	0
71:	active	35	35
72:	total	35	35
73:			
74:		terminals	
75:	un-matched	0	0
76:	matched but		
77:	different type	0	0
78:	total	27	27
79:			
80:			
81:	Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat		
ic			
82:			
83:	devbad.out:		
84:			
85:	netbad.out:		
86:			
87:	mergenet.out:		
88:			
89:	termbad.out:		
90:			
91:	prunenet.out:		
92:			
93:	prunedev.out:		
94:			
95:	audit.out:		
96:			
97:			
98:	Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout		
99:			
100:	devbad.out:		
101:			
102:	netbad.out:		
103:			
104:	mergenet.out:		
105:			
106:	termbad.out:		
107:			
108:	prunenet.out:		
109:			
110:	prunedev.out:		
111:			
112:	audit.out:		

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      7          nets  
12:      6          terminals  
13:      3          pmos  
14:      3          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      7          nets  
19:      6          terminals  
20:      3          pmos  
21:      3          nmos  
22:  
23:  
24:     Terminal correspondence points  
25:   N3      N6      a  
26:   N2      N5      b  
27:   N1      N1      gnd!  
28:   N4      N2      output  
29:   N5      N4      sel  
30:   N6      N0      vdd!  
31:  
32: Devices in the rules but not in the netlist:  
33:     cap nfet pfet nmos4 pmos4  
34:  
35: The net-lists match.  
36:  
37:           layout schematic  
38:           instances  
39:   un-matched      0      0  
40:   rewired        0      0  
41:   size errors    0      0  
42:   pruned         0      0  
43:   active          6      6  
44:   total           6      6  
45:  
46:           nets  
47:   un-matched      0      0  
48:   merged          0      0  
49:   pruned         0      0  
50:   active          7      7  
51:   total           7      7  
52:  
53:           terminals  
54:   un-matched      0      0  
55:   matched but  
56:   different type  0      0  
57:   total           6      6  
58:  
59:  
60: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
61:  
62: devbad.out:  
63:  
64: netbad.out:  
65:  
66: mergenet.out:  
67:  
68: termbad.out:  
69:  
70: prunenet.out:  
71:  
72: prunedev.out:  
73:  
74: audit.out:  
75:  
76:  
77: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
78:  
79: devbad.out:  
80:  
81: netbad.out:  
82:  
83: mergenet.out:  
84:  
85: termbad.out:  
86:  
87: prunenet.out:  
88:  
89: prunedev.out:  
90:  
91: audit.out:
```

Layout vs Schematic

nand2.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout/netlist  
10:    count  
11:      6          nets  
12:      5          terminals  
13:      2          pmos  
14:      2          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic/netlist  
17:    count  
18:      6          nets  
19:      5          terminals  
20:      2          pmos  
21:      2          nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N1      N1      gnd!  
26:     N3      N3      input1  
27:     N2      N6      input2  
28:     N4      N7      output  
29:     N5      N0      vdd!  
30:  
31: Devices in the rules but not in the netlist:  
32:     cap nfet pfet nmos4 pmos4  
33:  
34: The net-lists match.  
35:  
36:             layout schematic  
37:                 instances  
38:     un-matched      0      0  
39:     rewired        0      0  
40:     size errors    0      0  
41:     pruned         0      0  
42:     active         4      4  
43:     total          4      4  
44:  
45:             nets  
46:     un-matched      0      0  
47:     merged          0      0  
48:     pruned          0      0  
49:     active          6      6  
50:     total          6      6  
51:  
52:             terminals  
53:     un-matched      0      0  
54:     matched but  
55:     different type  0      0  
56:     total          5      5  
57:  
58:  
59: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
60:  
61: devbad.out:  
62:  
63: netbad.out:  
64:  
65: mergenet.out:  
66:  
67: termbad.out:  
68:  
69: prunenet.out:  
70:  
71: prunedev.out:  
72:  
73: audit.out:  
74:  
75:  
76: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
77:  
78: devbad.out:  
79:  
80: netbad.out:  
81:  
82: mergenet.out:  
83:  
84: termbad.out:  
85:  
86: prunenet.out:  
87:  
88: prunedev.out:  
89:  
90: audit.out:
```

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      6          nets  
12:      5          terminals  
13:      2          pmos  
14:      2          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      6          nets  
19:      5          terminals  
20:      2          pmos  
21:      2          nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N1      N1      gnd!  
26:     N3      N3      input1  
27:     N2      N7      input2  
28:     N4      N2      output  
29:     N5      N0      vdd!  
30:  
31: Devices in the netlist but not in the rules:  
32:     pcapacitor  
33: Devices in the rules but not in the netlist:  
34:     cap nfet pfet nmos4 pmos4  
35:  
36: The net-lists match.  
37:  
38:           layout schematic  
39:           instances  
40:     un-matched      0      0  
41:     rewired        0      0  
42:     size errors    0      0  
43:     pruned         0      0  
44:     active         4      4  
45:     total          4      4  
46:  
47:           nets  
48:     un-matched      0      0  
49:     merged          0      0  
50:     pruned         0      0  
51:     active         6      6  
52:     total          6      6  
53:  
54:           terminals  
55:     un-matched      0      0  
56:     matched but  
57:     different type  0      0  
58:           total          5      5  
59:  
60:  
61: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
62:  
63: devbad.out:  
64:  
65: netbad.out:  
66:  
67: mergenet.out:  
68:  
69: termbad.out:  
70:  
71: prunenet.out:  
72:  
73: prunedev.out:  
74:  
75: audit.out:  
76:  
77:  
78: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
79:  
80: devbad.out:  
81:  
82: netbad.out:  
83:  
84: mergenet.out:  
85:  
86: termbad.out:  
87:  
88: prunenet.out:  
89:  
90: prunedev.out:  
91:  
92: audit.out:
```

Layout vs Schematic
or2.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      7          nets  
12:      5          terminals  
13:      3          pmos  
14:      3          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      7          nets  
19:      5          terminals  
20:      3          pmos  
21:      3          nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N2      N1      gnd!  
26:     N4      N2      input1  
27:     N3      N3      input2  
28:     N5      N5      output  
29:     N6      N0      vdd!  
30:  
31: Devices in the netlist but not in the rules:  
32:     pcapacitor  
33: Devices in the rules but not in the netlist:  
34:     cap nfet pfet nmos4 pmos4  
35:  
36: The net-lists match.  
37:  
38:           layout schematic  
39:           instances  
40:     un-matched      0      0  
41:     rewired        0      0  
42:     size errors    0      0  
43:     pruned         0      0  
44:     active         6      6  
45:     total          6      6  
46:  
47:           nets  
48:     un-matched      0      0  
49:     merged          0      0  
50:     pruned          0      0  
51:     active          7      7  
52:     total           7      7  
53:  
54:           terminals  
55:     un-matched      0      0  
56:     matched but  
57:     different type  0      0
```

```
58:           total      5      5  
59:  
60:  
61: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
62:  
63: devbad.out:  
64:  
65: netbad.out:  
66:  
67: mergenet.out:  
68:  
69: termbad.out:  
70:  
71: prunenet.out:  
72:  
73: prunedev.out:  
74:  
75: audit.out:  
76:  
77:  
78: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
79:  
80: devbad.out:  
81:  
82: netbad.out:  
83:  
84: mergenet.out:  
85:  
86: termbad.out:  
87:  
88: prunenet.out:  
89:  
90: prunedev.out:  
91:  
92: audit.out:
```

Layout vs Schematic

tag_cmp1.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      31      nets  
12:      14      terminals  
13:      21      pmos  
14:      21      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      31      nets  
19:      14      terminals  
20:      21      pmos  
21:      21      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N29   N1      gnd!  
26: N42   N12     mem2  
27: N41   N13     mem3  
28: N40   N14     mem4  
29: N38   N4      t14  
30: N37   N5      t15  
31: N36   N6      t16  
32: N35   N7      t17  
33: N34   N8      t18  
34: N33   N9      t19  
35: N32   N10     t20  
36: N31   N11     t21  
37: N39   N0      vdd!  
38: N30   N15     write_cache  
39:  
40: Devices in the netlist but not in the rules:  
41:     pcapacitor  
42: Devices in the rules but not in the netlist:  
43:     cap nfet pfet nmos4 pmos4  
44:  
45: The net-lists match.  
46:  
47:           layout schematic  
48:           instances  
49: un-matched      0      0  
50: rewired        0      0  
51: size errors    0      0  
52: pruned         0      0  
53: active          42     42  
54: total           42     42  
55:  
56:           nets  
57: un-matched      0      0  
58: merged          0      0  
59: pruned          0      0  
60: active          31     31  
61: total           31     31  
62:  
63:               terminals  
64: un-matched      0      0  
65: matched but  
66: different type  0      0  
67: total           14     14  
68:  
69:  
70: Probe files from /afs/umbc.edu/users/n/n67/home/nitheesh/cadence/LVS/schematic  
ic  
71:  
72: devbad.out:  
73:  
74: netbad.out:  
75:  
76: mergenet.out:  
77:  
78: termbad.out:  
79:  
80: prunenet.out:  
81:  
82: prunedev.out:  
83:  
84: audit.out:  
85:  
86:  
87: Probe files from /afs/umbc.edu/users/n/n67/home/nitheesh/cadence/LVS/layout  
88:  
89: devbad.out:  
90:  
91: netbad.out:  
92:  
93: mergenet.out:  
94:  
95: termbad.out:  
96:  
97: prunenet.out:  
98:  
99: prunedev.out:  
100:  
101: audit.out:
```

Layout vs Schematic

tag_valid.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      403      nets  
12:      19       terminals  
13:      448      pmos  
14:      448      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      403      nets  
19:      19       terminals  
20:      448      pmos  
21:      448      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N392   N6   blk_en0  
26: N391   N7   blk_en1  
27: N390   N8   blk_en2  
28: N389   N9   blk_en3  
29: N388   N10  blk_en4  
30: N387   N11  blk_en5  
31: N386   N12  blk_en6  
32: N385   N13  blk_en7  
33: N398   N5   din<0>  
34: N397   N4   din<1>  
35: N396   N3   din<2>  
36: N394   N2   din<3>  
37: N384   N18  dout<0>  
38: N401   N17  dout<1>  
39: N400   N16  dout<2>  
40: N399   N15  dout<3>  
41: N393   N1   gnd!  
42: N402   N14  rw_en  
43: N395   N0   vdd!  
44:  
45: Devices in the rules but not in the netlist:  
46:     cap nfet pfet nmos4 pmos4  
47:  
48: The net-lists match.  
49:  
50:           layout schematic  
51:                   instances  
52:     un-matched      0      0  
53:     rewired         0      0  
54:     size errors     0      0  
55:     pruned          0      0  
56:     active          896    896  
57:     total           896    896  
58:  
59:           nets  
60:     un-matched      0      0  
61:     merged          0      0  
62:     pruned          0      0  
63:     active          403    403  
64:     total           403    403  
65:  
66:           terminals  
67:     un-matched      0      0  
68:     matched but     0      0  
69:     different type  0      0  
70:     total           19     19  
71:  
72:  
73: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
74:  
75: devbad.out:  
76:  
77: netbad.out:  
78:  
79: mergenet.out:  
80:  
81: termbad.out:  
82:  
83: prunenet.out:  
84:  
85: prunedev.out:  
86:  
87: audit.out:  
88:  
89:  
90: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
91:  
92: devbad.out:  
93:  
94: netbad.out:  
95:  
96: mergenet.out:  
97:  
98: termbad.out:  
99:  
100: prunenet.out:  
101:  
102: prunedev.out:  
103:  
104: audit.out:
```

Layout vs Schematic
tristate_buf_8b.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      27      nets  
12:      19      terminals  
13:      16      pmos  
14:      16      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      27      nets  
19:      19      terminals  
20:      16      pmos  
21:      16      nmos  
22:  
23:  
24: Terminal correspondence points  
25: N13   N16   a<0>  
26: N11   N15   a<1>  
27: N8    N14   a<2>  
28: N25   N13   a<3>  
29: N23   N12   a<4>  
30: N22   N11   a<5>  
31: N20   N10   a<6>  
32: N18   N9    a<7>  
33: N15   N2    en  
34: N10   N0    gnd!  
35: N17   N1    vdd!  
36: N21   N18   y<0>  
37: N19   N17   y<1>  
38: N16   N8    y<2>  
39: N14   N7    y<3>  
40: N12   N6    y<4>  
41: N9    N5    y<5>  
42: N26   N4    y<6>  
43: N24   N3    y<7>  
44:  
45: Devices in the netlist but not in the rules:  
46:     pcapacitor  
47: Devices in the rules but not in the netlist:  
48:     cap nfet pfet nmos4 pmos4  
49:  
50: The net-lists match.  
51:  
52:             layout schematic  
53:                 instances  
54:     un-matched      0      0  
55:     rewired         0      0  
56:     size errors     0      0  
57:     pruned          0      0  
58:     active          32     32  
59:     total           32     32  
60:  
61:                         nets  
62:     un-matched      0      0  
63:     merged          0      0  
64:     pruned          0      0  
65:     active          27     27  
66:     total           27     27  
67:  
68:                         terminals  
69:     un-matched      0      0  
70:     matched         0      0  
71:     different type  0      0  
72:     total           19     19  
73:  
74:  
75: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
76:  
77: devbad.out:  
78:  
79: netbad.out:  
80:  
81: mergenet.out:  
82:  
83: termbad.out:  
84:  
85: prunenet.out:  
86:  
87: prunedev.out:  
88:  
89: audit.out:  
90:  
91:  
92: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
93:  
94: devbad.out:  
95:  
96: netbad.out:  
97:  
98: mergenet.out:  
99:  
100: termbad.out:  
101:  
102: prunenet.out:  
103:  
104: prunedev.out:  
105:  
106: audit.out:
```

Layout vs Schematic
tristate_buf.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout/netlist  
10:    count  
11:      6          nets  
12:      5          terminals  
13:      2          pmos  
14:      2          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic/netlist  
17:    count  
18:      6          nets  
19:      5          terminals  
20:      2          pmos  
21:      2          nmos  
22:  
23:  
24:     Terminal correspondence points  
25:   N2      N5      a  
26:   N3      N4      en  
27:   N1      N0      gnd!  
28:   N4      N1      vdd!  
29:   N5      N2      y  
30:  
31: Devices in the netlist but not in the rules:  
32:     pcapacitor  
33: Devices in the rules but not in the netlist:  
34:     cap nfet pfet nmos4 pmos4  
35:  
36: The net-lists match.  
37:  
38:           layout schematic  
39:           instances  
40:   un-matched      0      0  
41:   rewired        0      0  
42:   size errors    0      0  
43:   pruned         0      0  
44:   active          4      4  
45:   total           4      4  
46:  
47:           nets  
48:   un-matched      0      0  
49:   merged          0      0  
50:   pruned          0      0  
51:   active          6      6  
52:   total           6      6  
53:  
54:           terminals  
55:   un-matched      0      0  
56:   matched but  
57:   different type  0      0
```

```
58:           total      5      5  
59:  
60:  
61: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
62:  
63: devbad.out:  
64:  
65: netbad.out:  
66:  
67: mergenet.out:  
68:  
69: termbad.out:  
70:  
71: prunenet.out:  
72:  
73: prunedev.out:  
74:  
75: audit.out:  
76:  
77:  
78: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
79:  
80: devbad.out:  
81:  
82: netbad.out:  
83:  
84: mergenet.out:  
85:  
86: termbad.out:  
87:  
88: prunenet.out:  
89:  
90: prunedev.out:  
91:  
92: audit.out:
```

Layout vs Schematic

tx.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout/netlist  
10:    count  
11:      6          nets  
12:      6          terminals  
13:      1          pmos  
14:      1          nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic/netlist  
17:    count  
18:      6          nets  
19:      6          terminals  
20:      1          pmos  
21:      1          nmos  
22:  
23:  
24:     Terminal correspondence points  
25: N0      N0      gnd!  
26: N2      N6      input  
27: N1      N4      output  
28: N4      N5      sel  
29: N3      N7      selnot  
30: N5      N1      vdd!  
31:  
32: Devices in the rules but not in the netlist:  
33:     cap nfet pfet nmos4 pmos4  
34:  
35: The net-lists match.  
36:  
37:           layout schematic  
38:           instances  
39: un-matched      0      0  
40: rewired        0      0  
41: size errors    0      0  
42: pruned         0      0  
43: active          2      2  
44: total           2      2  
45:  
46:           nets  
47: un-matched      0      0  
48: merged          0      0  
49: pruned         0      0  
50: active          6      6  
51: total           6      6  
52:  
53:           terminals  
54: un-matched      0      0  
55: matched but  
56: different type   0      0  
57: total           6      6  
58:  
59:  
60: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
61:  
62: devbad.out:  
63:  
64: netbad.out:  
65:  
66: mergenet.out:  
67:  
68: termbad.out:  
69:  
70: prunenet.out:  
71:  
72: prunedev.out:  
73:  
74: audit.out:  
75:  
76:  
77: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
78:  
79: devbad.out:  
80:  
81: netbad.out:  
82:  
83: mergenet.out:  
84:  
85: termbad.out:  
86:  
87: prunenet.out:  
88:  
89: prunedev.out:  
90:  
91: audit.out:
```

xnor2.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      13      nets  
12:      5      terminals  
13:      9      pmos  
14:      9      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      13      nets  
19:      5      terminals  
20:      9      pmos  
21:      9      nmos  
22:  
23:  
24:     Terminal correspondence points  
25:     N8      N1      gnd!  
26:     N10     N5      input1  
27:     N9      N4      input2  
28:     N11     N2      output  
29:     N12     N0      vdd!  
30:  
31: Devices in the netlist but not in the rules:  
32:     pcapacitor  
33: Devices in the rules but not in the netlist:  
34:     cap nfet pfet nmos4 pmos4  
35:  
36: The net-lists match.  
37:  
38:             layout schematic  
39:             instances  
40:     un-matched      0      0  
41:     rewired        0      0  
42:     size errors    0      0  
43:     pruned         0      0  
44:     active         18     18  
45:     total          18     18  
46:  
47:             nets  
48:     un-matched      0      0  
49:     merged          0      0  
50:     pruned          0      0  
51:     active          13     13  
52:     total           13     13  
53:  
54:             terminals  
55:     un-matched      0      0  
56:     matched but  
57:     different type  0      0  
58:             total      5      5  
59:  
60:  
61: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schematic  
62:  
63: devbad.out:  
64:  
65: netbad.out:  
66:  
67: mergenet.out:  
68:  
69: termbad.out:  
70:  
71: prunenet.out:  
72:  
73: prunedev.out:  
74:  
75: audit.out:  
76:  
77:  
78: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
79:  
80: devbad.out:  
81:  
82: netbad.out:  
83:  
84: mergenet.out:  
85:  
86: termbad.out:  
87:  
88: prunenet.out:  
89:  
90: prunedev.out:  
91:  
92: audit.out:
```

xor2.txt

```
1: @(#)SCDS: LVS version 6.1.7-64b 09/27/2016 19:41 (sjfhw305) $  
2:  
3: Command line: /afs/umbc.edu/software/cadence/install/IC617/tools.lnx86/dfII/b  
in/64bit/LVS -dir /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS -l -s -t /afs/  
umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout /afs/umbc.edu/users/n/n/n67/ho  
me/nitheesh/cadence/LVS/schematic  
4: Like matching is enabled.  
5: Net swapping is enabled.  
6: Using terminal names as correspondence points.  
7: Compiling Diva LVS rules...  
8:  
9:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/layout/netlist  
10:    count  
11:      12      nets  
12:      5      terminals  
13:      8      pmos  
14:      8      nmos  
15:  
16:     Net-list summary for /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS  
/schematic/netlist  
17:    count  
18:      12      nets  
19:      5      terminals  
20:      8      pmos  
21:      8      nmos  
22:  
23:  
24:     Terminal correspondence points  
25: N7      N1      gnd!  
26: N9      N7      input1  
27: N8      N6      input2  
28: N10     N2      output  
29: N11     N0      vdd!  
30:  
31: Devices in the netlist but not in the rules:  
32:     pcapacitor  
33: Devices in the rules but not in the netlist:  
34:     cap nfet pfet nmos4 pmos4  
35:  
36: The net-lists match.  
37:  
38:           layout schematic  
39:           instances  
40: un-matched      0      0  
41: rewired        0      0  
42: size errors    0      0  
43: pruned         0      0  
44: active          16     16  
45: total           16     16  
46:  
47:           nets  
48: un-matched      0      0  
49: merged          0      0  
50: pruned          0      0  
51: active          12     12  
52: total           12     12  
53:  
54:           terminals  
55: un-matched      0      0  
56: matched but  
57: different type  0      0  
58:     total          5      5  
59:  
60:  
61: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/schemat  
ic  
62:  
63: devbad.out:  
64:  
65: netbad.out:  
66:  
67: mergenet.out:  
68:  
69: termbad.out:  
70:  
71: prunenet.out:  
72:  
73: prunedev.out:  
74:  
75: audit.out:  
76:  
77:  
78: Probe files from /afs/umbc.edu/users/n/n/n67/home/nitheesh/cadence/LVS/layout  
79:  
80: devbad.out:  
81:  
82: netbad.out:  
83:  
84: mergenet.out:  
85:  
86: termbad.out:  
87:  
88: prunenet.out:  
89:  
90: prunedev.out:  
91:  
92: audit.out:
```