

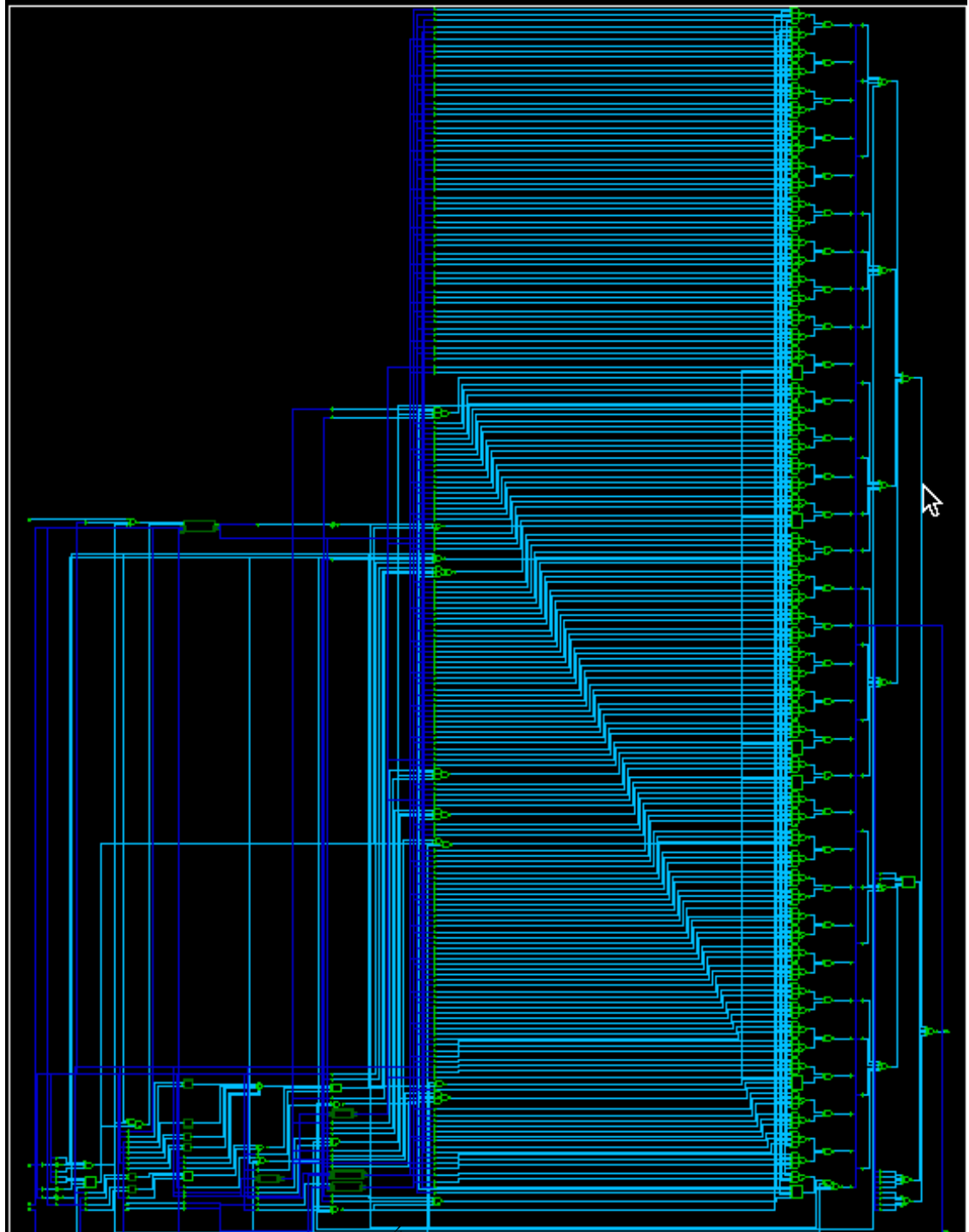
VHDL Lab2 Report

JHH-SIOU LI, 105021226

April 19, 2019

1 Design Chart

Here is my design chart synthesized by using Design Vision.



2 Some other charts and descriptions

Since this time we don't need to build the circuit from drawing a K-Map, it is far more easier to build a ALU.

For the logic gate part, I just need to write down the sum of products and make it by stacking up the NAND gate (like this):

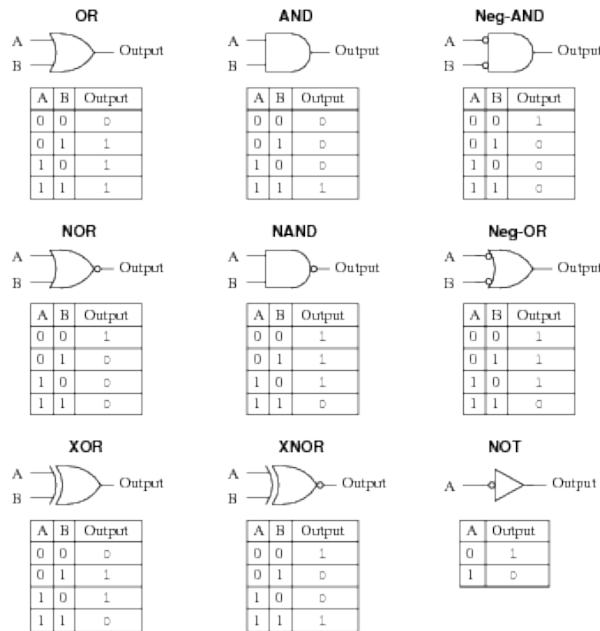
```
module AND (a, b, c);
    input  a, b;
    output c;
    wire nand_0;
    // IMPLEMENTATION
    nand nand1(nand_0, a, b);
    NOT N1(nand_0, c);
endmodule

module OR (a, b, c);
    input  a, b;
    output c;
    wire not_0, not_1;
    // IMPLEMENTATION
    NOT N1(a, not_0);
    NOT N2(b, not_1);
    nand(c, not_0, not_1);
endmodule

module NOT (a, b);
    input  a;
    output b;
    // IMPLEMENTATION
    nand(b, a, a);
endmodule
```

Once we got these basic logic gates done by using NAND gate, we can then use them to build up more complex gates such as NOR, XOR, XNOR gates.

Here are the truth tables:



Since the Lab2 didn't require us to draw the K-Maps, I'm going to omit them here.

3 Problems

The main problem in my Verilog design were 2 important errors:

1. Putting another always block inside a always block.
This caused me to seriously delayed the progress of this assignment for few days and I had no clue about what is going wrong.
2. Wrong array size in my wire variables.
With this typo, I spent about 3 hrs to find the error in my Decoder and Arbiter and found nothing.(here)

```

wire [32-1:0] dec_1;
wire [5-1:0] arb_1;
...
4'b1110: Y = dec_1;
4'b1111: Y = arb_1;

```

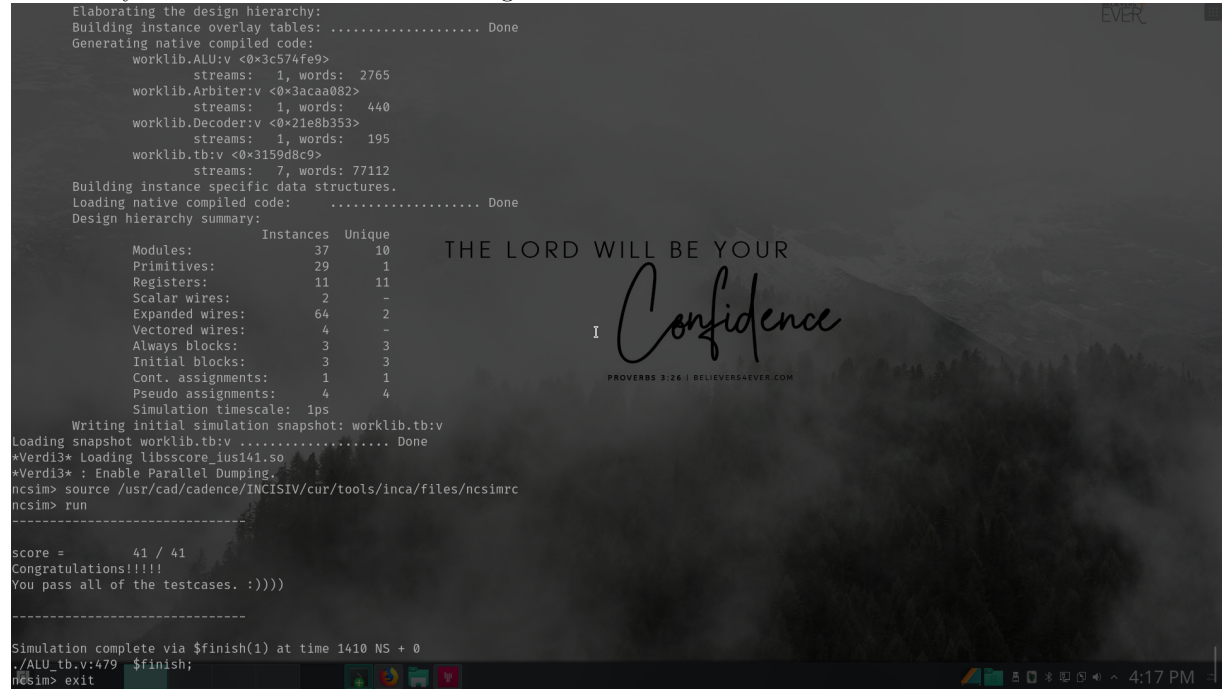
If we assign the input with the size that is too small, then it will fill up Y's remaining blanks with 'z's, which is not our desired '0's.

To sum it up, it was nothing but careless typos and conceptual inaccuracy.

4 Simulation Outcome

Here is my simulation outcome of ncvverilog:

```
Elaborating the design hierarchy:
Building instance overlay tables: ..... Done
Generating native compiled code:
  worklib.ALU:v <0x3c574fe9>
    streams: 1, words: 2765
  worklib.Arbitrator:v <0x3acaa082>
    streams: 1, words: 440
  worklib.Decoder:v <0x21e8b353>
    streams: 1, words: 195
  worklib.tb:v <0x3159d8c9>
    streams: 7, words: 77112
Building instance specific data structures.
Loading native compiled code: ..... Done
Design hierarchy summary:
  Instances Unique
  Modules: 37 10
  Primitives: 29 1
  Registers: 11 11
  Scalar wires: 2 -
  Expanded wires: 64 2
  Vectored wires: 4 -
  Always blocks: 3 3
  Initial blocks: 3 3
  Cont. assignments: 1 1
  Pseudo assignments: 4 4
  Simulation timescale: 1ps
Writing initial simulation snapshot: worklib.tb:v
Loading snapshot worklib.tb:v ..... Done
*Verdi3* Loading libsscore_ius141.so
*Verdi3* : Enable Parallel Dumping.
ncsim source /usr/cad/cadence/INCISIV/cur/tools/inca/files/ncsimrc
ncsim run
-----
score = 41 / 41
Congratulations!!!!
You pass all of the testcases. :)))
-----
Simulation complete via $finish(1) at time 1410 NS + 0
./ALU.tb.v:479 $finish;
ncsim> exit
```



5 Report Area

The content inside my report area txt file.

```
*****
Report : area
Design : ALU
Version: K-2015.06-SP1
Date   : Fri Apr 19 14:46:52 2019
*****
Library(s) Used:

slow (File: /theda21_2/CBDK_IC_Contest/cur/SynopsysDC/db/slow.db)

Number of ports:          489
Number of nets:          1492
Number of cells:          930
Number of combinational cells: 890
Number of sequential cells: 2
Number of macros/black boxes: 0
Number of buf/inv:        147
Number of references:      36

Combinational area:      12183.937274
Buf/Inv area:            692.539189
Noncombinational area:    0.000000
Macro/Black Box area:    0.000000
Net Interconnect area:    undefined (No wire load specified)

Total cell area:         12183.937274
Total area:              undefined
```

