Digital Logic Design

Computer Assignment 3

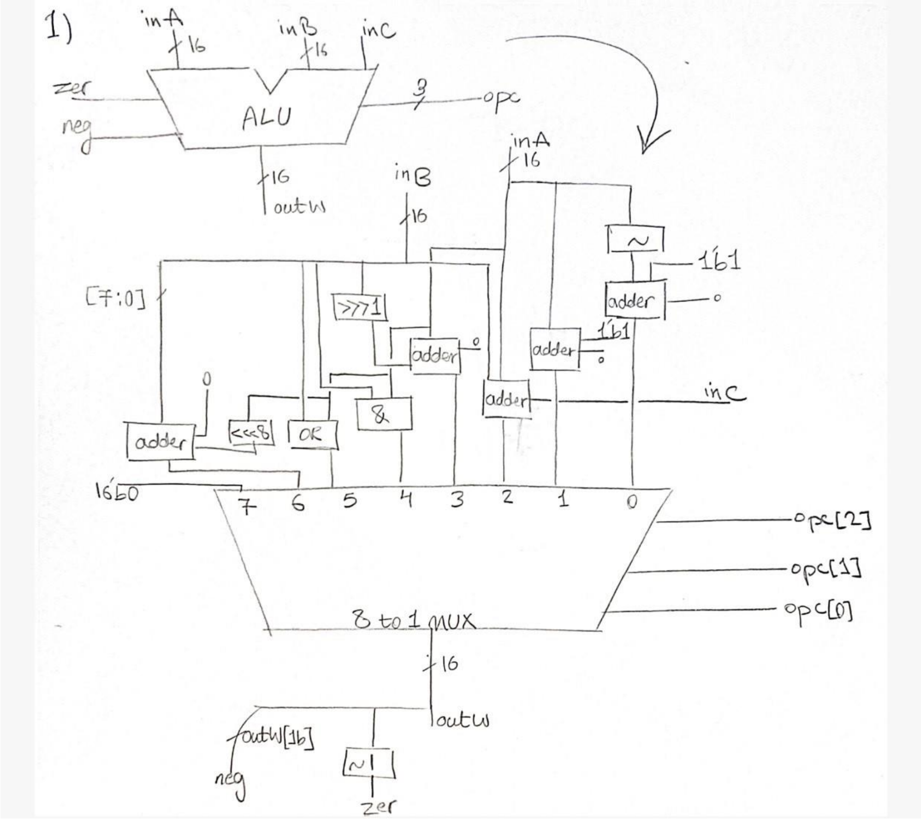
Report sheet

Niloufar mortazavi

220701096

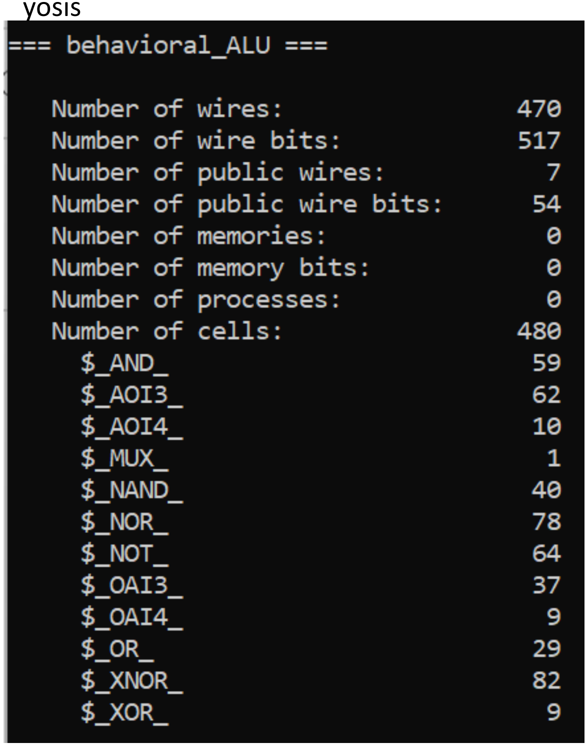
نیلوفر مرتضوی

Q1) B) First, I wrote the Verilog code for the behavioral ALU (pre synthesis) and then I gave the code to yosis to synthesis it and I stored as a separate file.



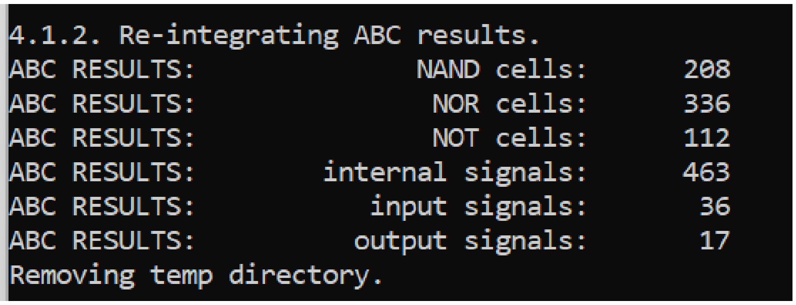
Then I wrote a testbench to compare these two synthesized and pre synthesized Verilog codes. I got help from mycells.lib and yosis libraries to compare those.

yosis



```
=== behavioral_ALU ===

   Number of wires:                470
   Number of wire bits:            517
   Number of public wires:           7
   Number of public wire bits:      54
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                480
     $_AND_                         59
     $_AOI3_                        62
     $_AOI4_                        10
     $_MUX_                          1
     $_NAND_                        40
     $_NOR_                         78
     $_NOT_                         64
     $_OAI3_                        37
     $_OAI4_                         9
     $_OR_                          29
     $_XNOR_                        82
     $_XOR_                          9
```

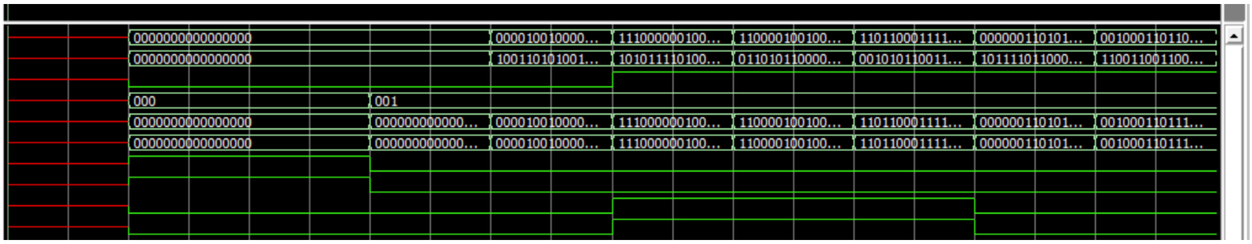
Mycells.lib

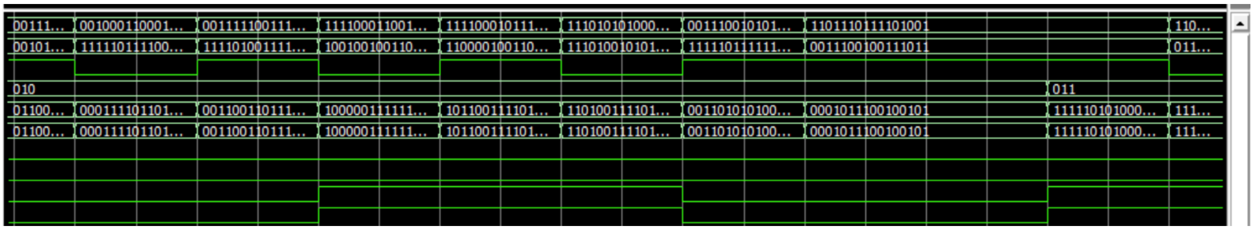


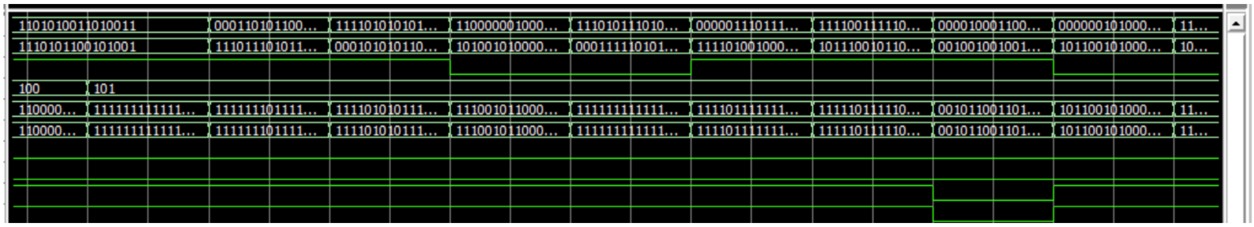
```
4.1.2. Re-integrating ABC results.
ABC RESULTS:             NAND cells:      208
ABC RESULTS:              NOR cells:      336
ABC RESULTS:              NOT cells:      112
ABC RESULTS:       internal signals:      463
ABC RESULTS:          input signals:       36
ABC RESULTS:         output signals:       17
Removing temp directory.
```

C) I erased the gate delays in mycells.lib because my code was without delays and I wanted to check if yosis synthesis my code correctly or not. The wave from both synthesized and pre synthesized codes are similar so that I found out yosis did a great job synthesizing the Verilog code.

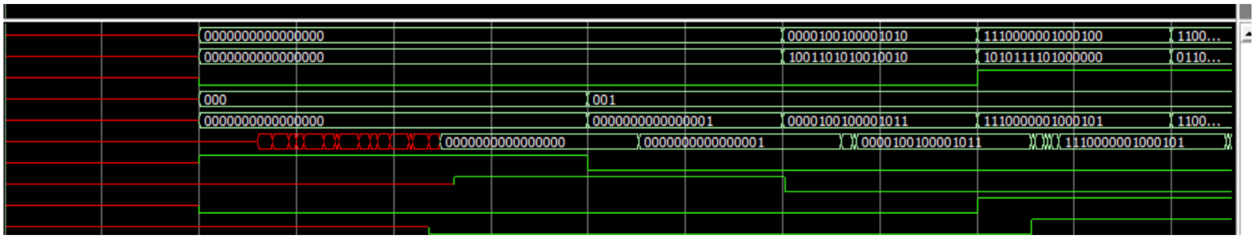
Some part of testbench wave forms for behavioral ALU:





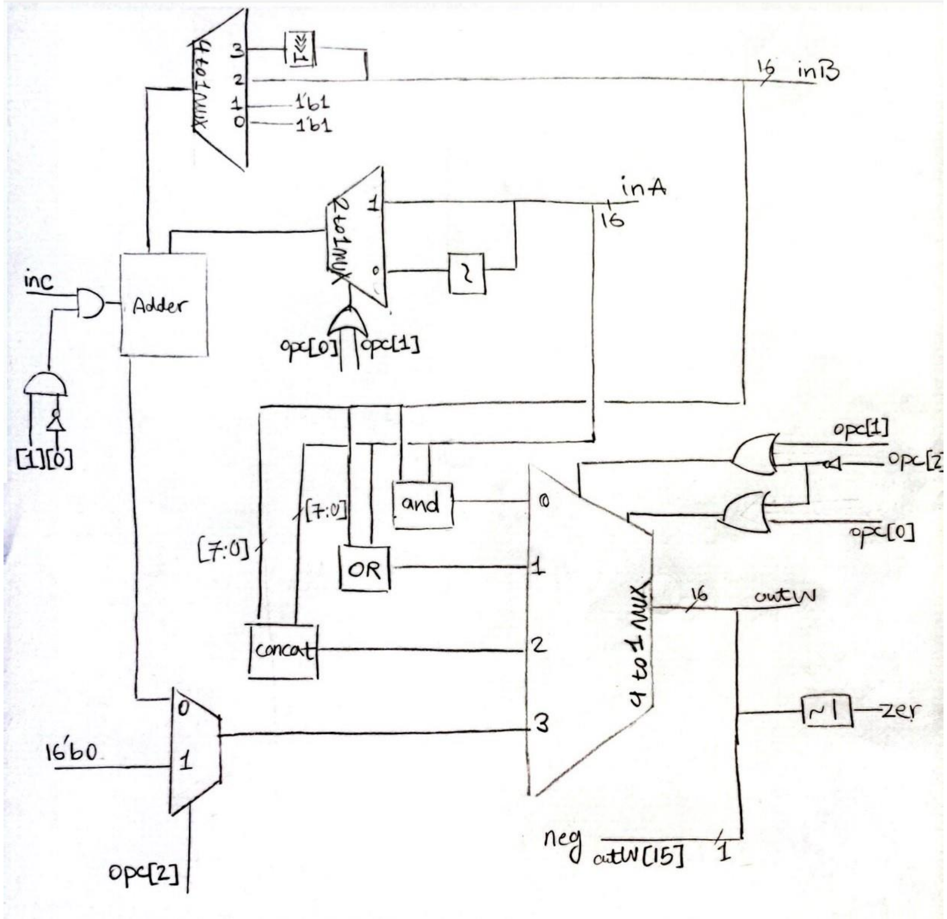


D) if we consider the delays the second wave form would have some differences with the first one as we can see below:

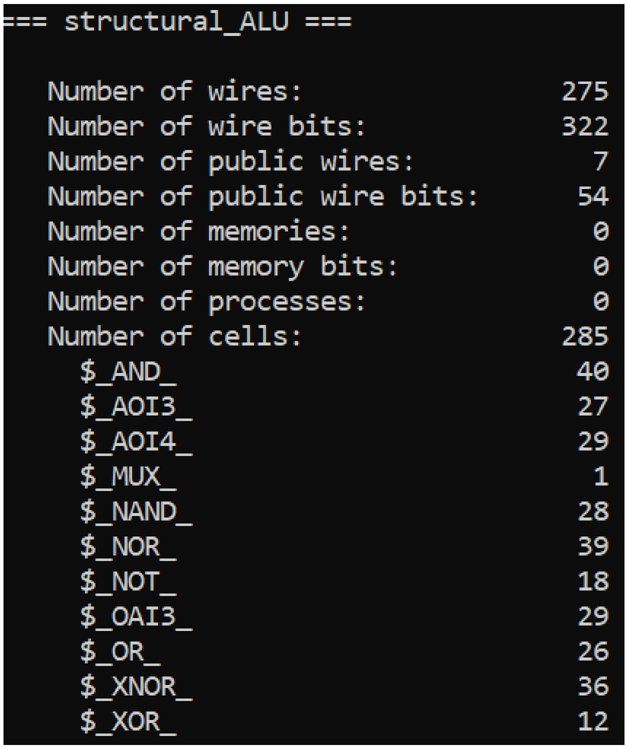


And because of the delays the bits might have been recived in different times so we may have wrong outputs. Also by observing the waves we can see that by having too much loops the pre synthesized code would be more efficient because of the less simulation run time.

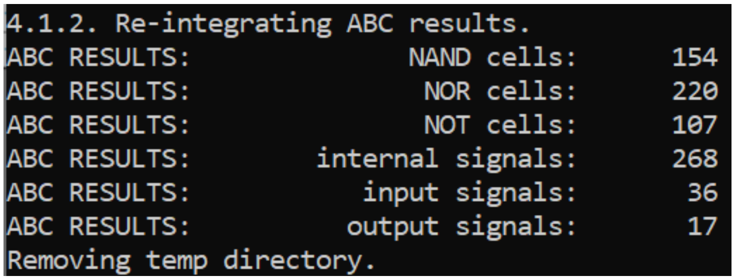
Q2) B) I did exactly like Q1 but this time for a structural ALU:



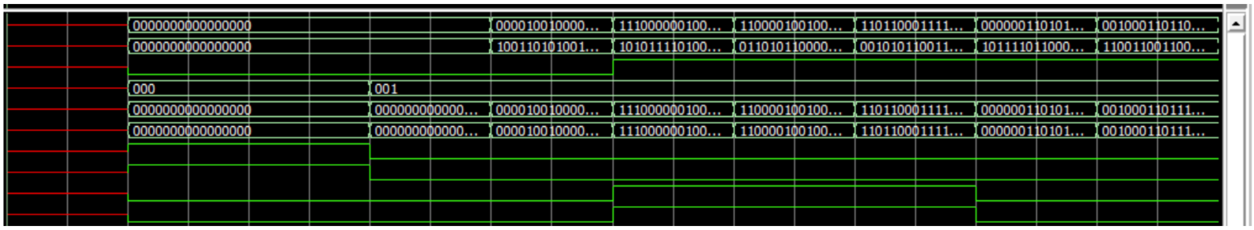
yosis

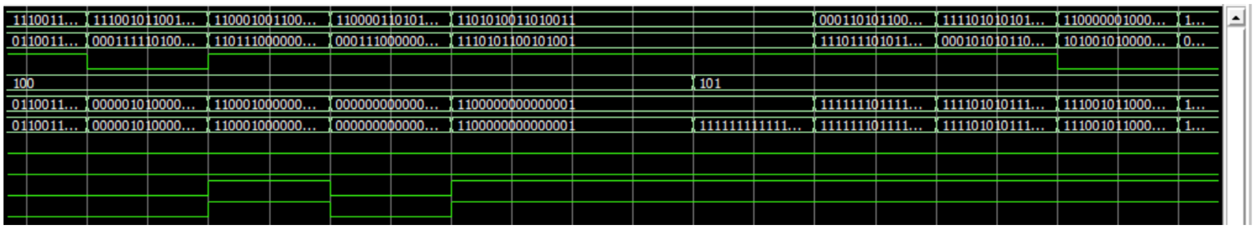

```
=== structural_ALU ===

   Number of wires:                275
   Number of wire bits:            322
   Number of public wires:           7
   Number of public wire bits:      54
   Number of memories:               0
   Number of memory bits:            0
   Number of processes:              0
   Number of cells:                285
     $_AND_                         40
     $_AOI3_                        27
     $_AOI4_                        29
     $_MUX_                          1
     $_NAND_                        28
     $_NOR_                         39
     $_NOT_                         18
     $_OAI3_                        29
     $_OR_                          26
     $_XNOR_                        36
     $_XOR_                         12
```

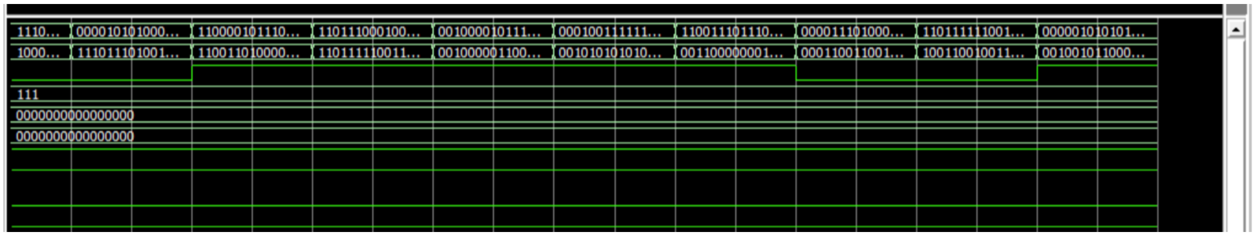
Mycells.lib


```
4.1.2. Re-integrating ABC results.
ABC RESULTS:               NAND cells:      154
ABC RESULTS:                NOR cells:      220
ABC RESULTS:                NOT cells:      107
ABC RESULTS:        internal signals:       268
ABC RESULTS:           input signals:        36
ABC RESULTS:          output signals:        17
Removing temp directory.
```

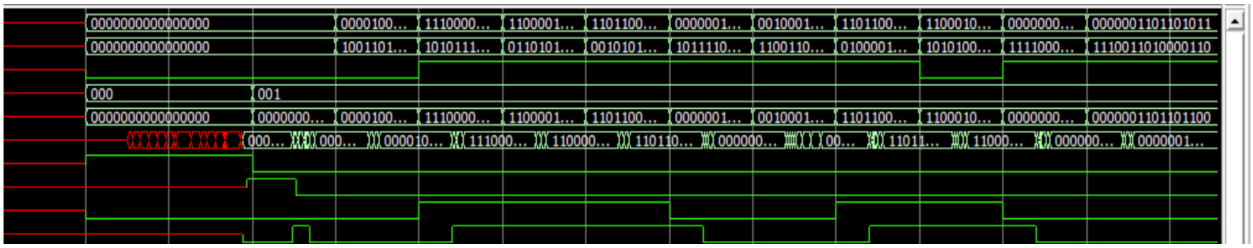
C) exactly as Q1 the wave forms from both will be similar:







D) as part D in Q1 if we consider the delays the waveforms would be different:



Q3) for this part i wrote a testbench to compare pre synthesized behavioral ALU with the structural one. As you can see they are the same so we come to conclusion that both of our verilogs work allright.