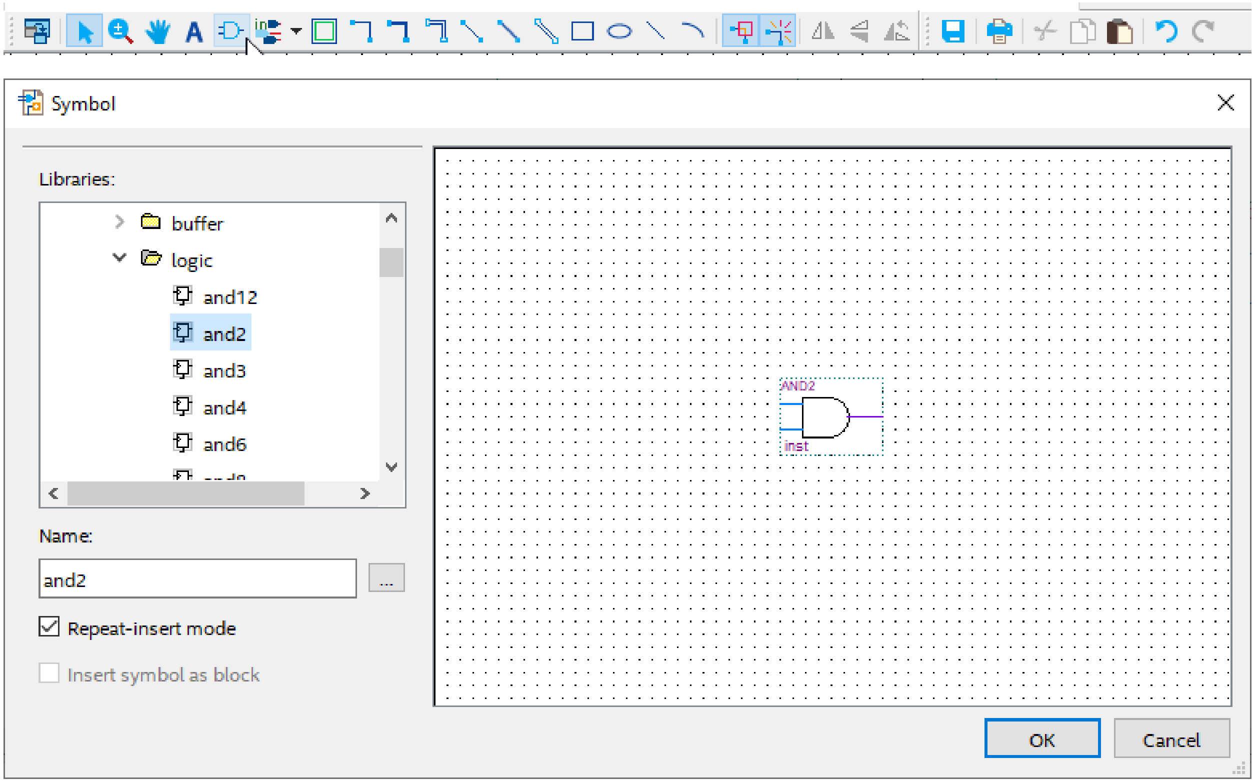
ZHANG_CHUE

Task 2: “2:1 Mux”


Symbol

Libraries:

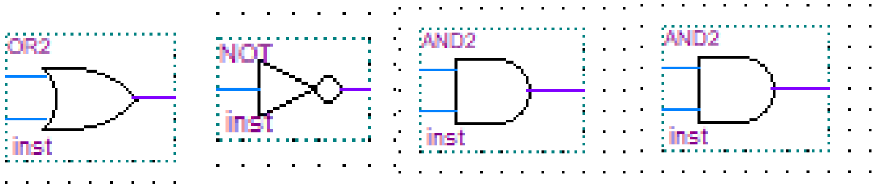
- buffer
- logic
  - and12
  - and2
  - and3
  - and4
  - and6

Name: and2

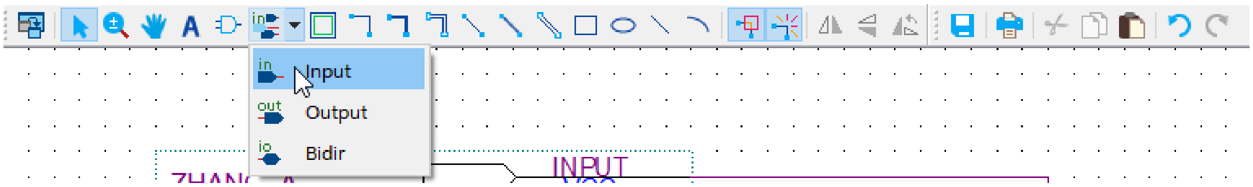
☑ Repeat-insert mode

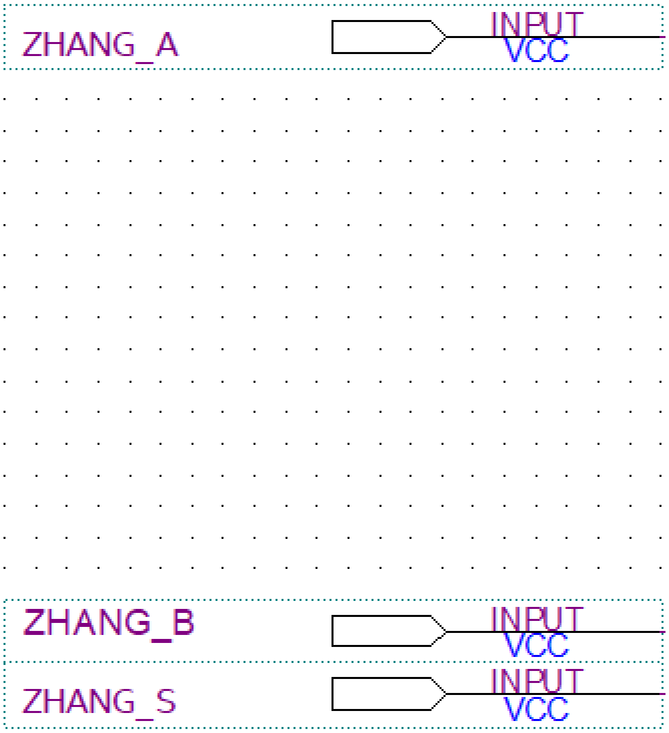
□ Insert symbol as block

I start by clicking the symbol icon in Quartus and picking out the logic gates. For the 2 to 1 Mux, I will be using two AND-gates, one OR-gate and one NOT-gate.

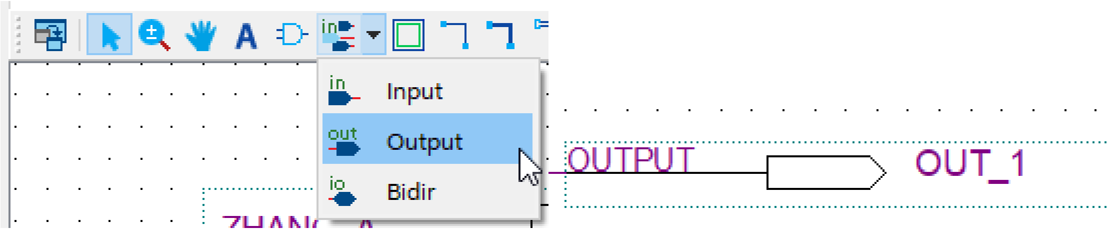


Next, I go to the pins tool dropdown menu and select input. For the MUX, I used 3 inputs

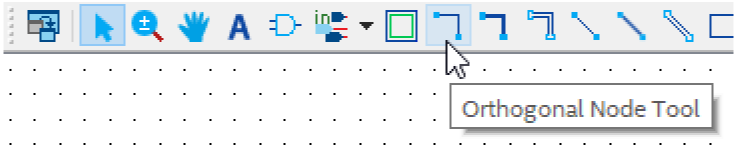


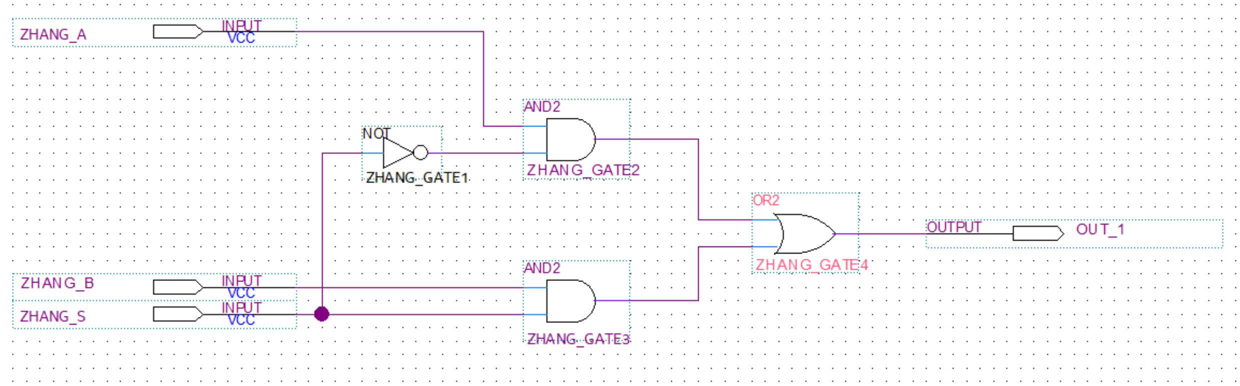


I repeat the same steps again for output Pins, however, I only use one output pin

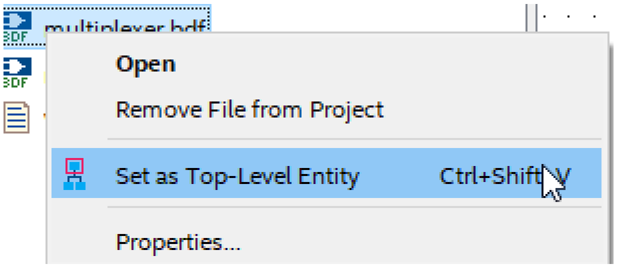


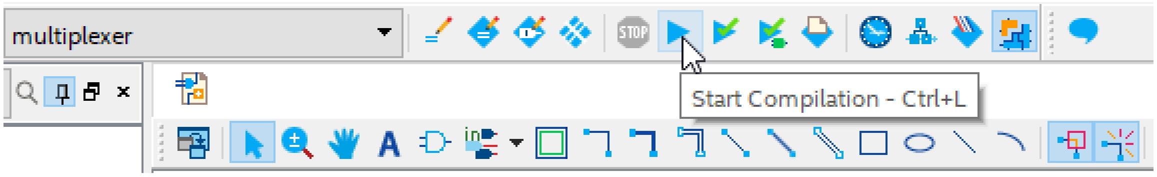
Then I use the Orthogonal Node tool to connect the inputs and outputs to the gates.

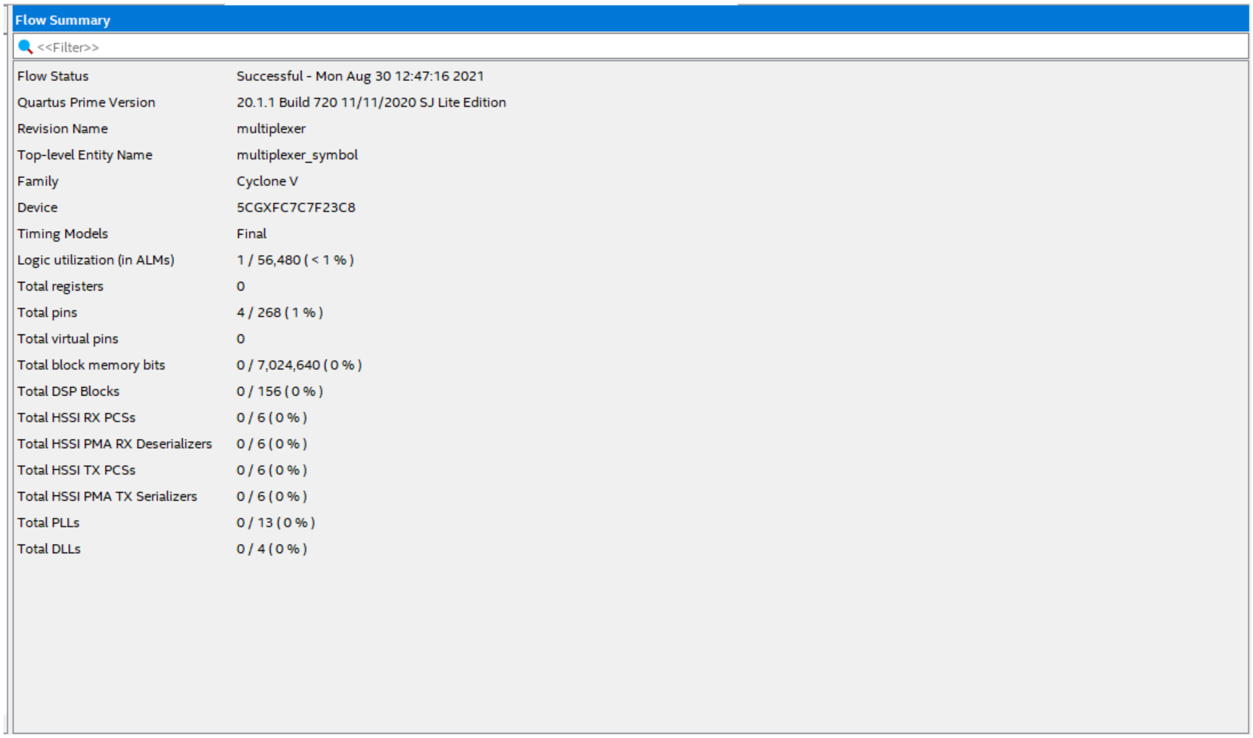




In the figure above, what is shown is a 2:1 MUX where we have 3 inputs. Input A, Input B and a Selector input. The selector is inputted in the NOT-gate which is then inputted in the ZHANG_GATE2 with ZHANG_A input. It is also inputted into the ZHANG_GATE3 with ZHANG_B input and finally both ZHANG_GATE2 and ZHANG_GATE3 and inputted together into ZHANG_GATE4 (OR-gate) where the result will be outputted.

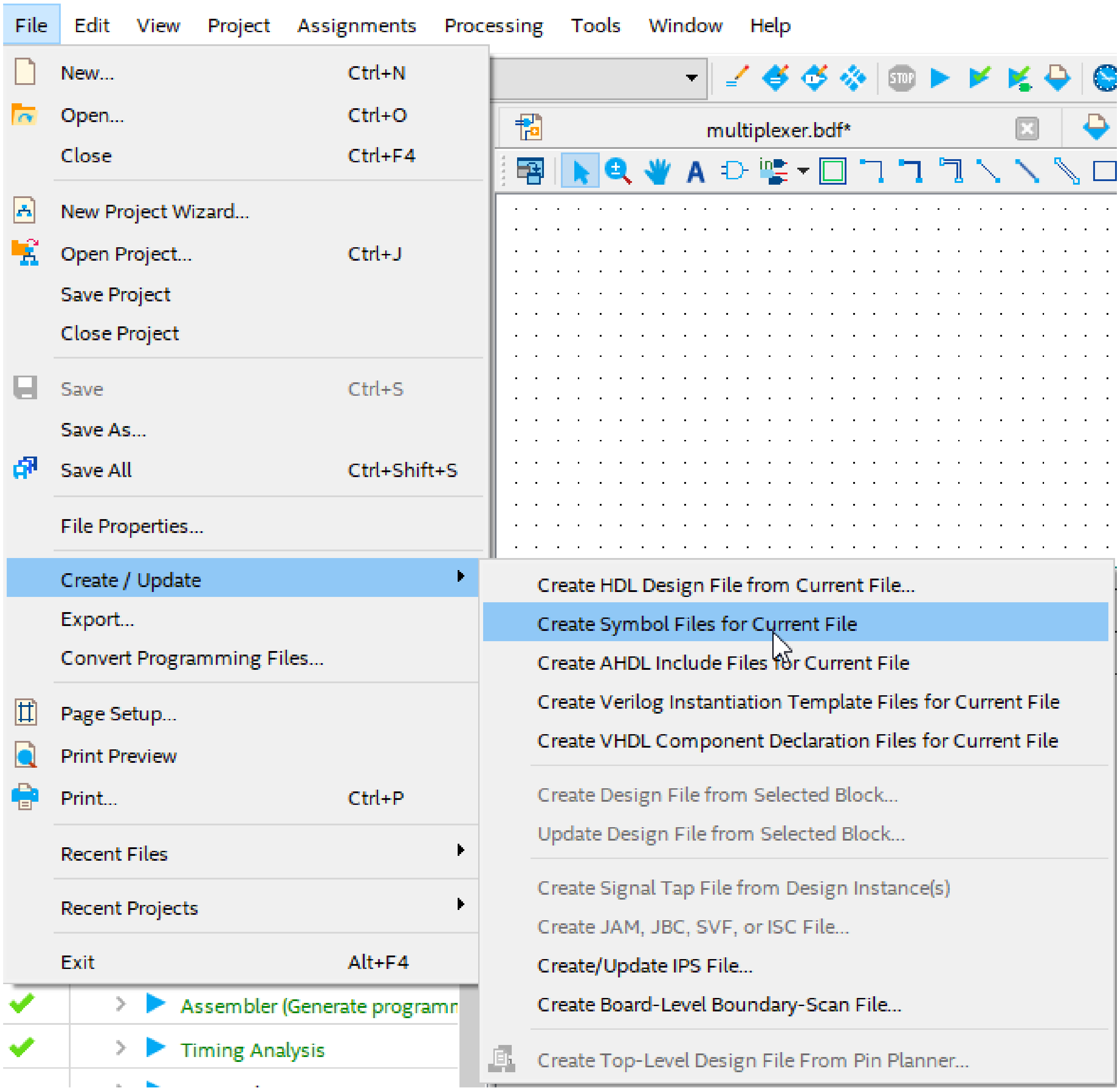


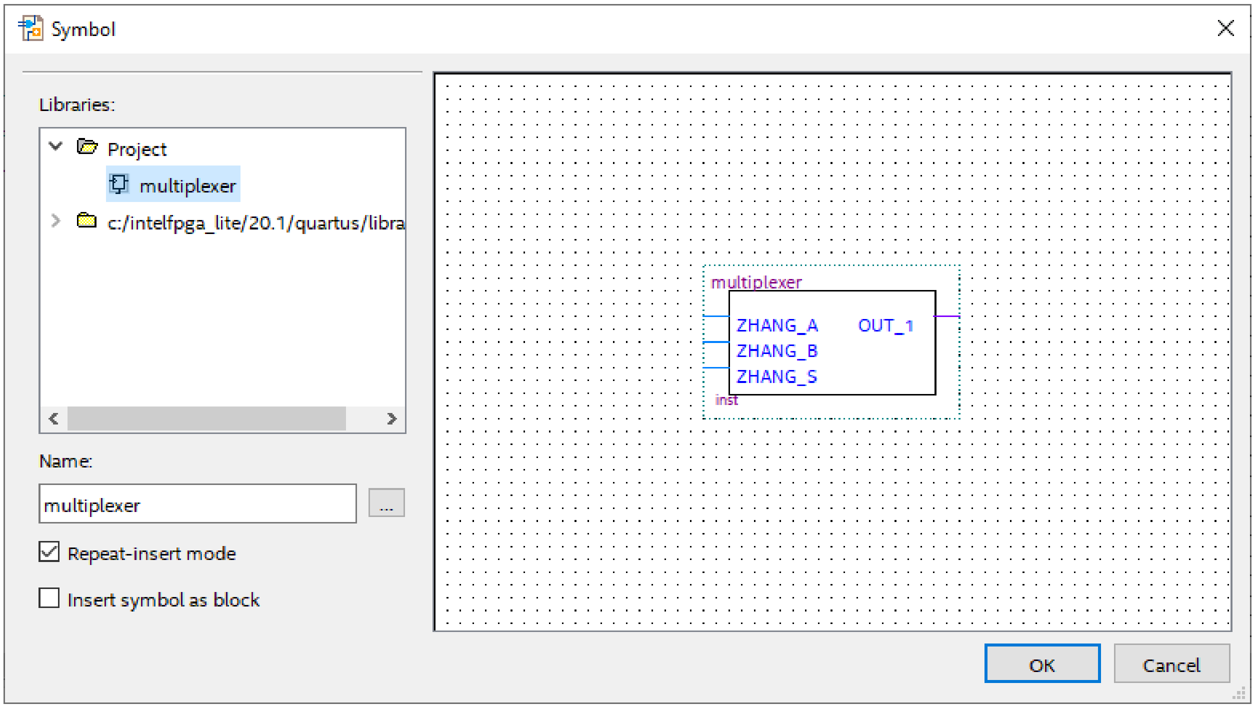




Flow Status	Successful - Mon Aug 30 12:47:16 2021
Quartus Prime Version	20.1.1 Build 720 11/11/2020 SJ Lite Edition
Revision Name	multiplexer
Top-level Entity Name	multiplexer_symbol
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	1 / 56,480 (< 1 %)
Total registers	0
Total pins	4 / 268 (1 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

In this next few screenshot, what I did was I set the MUX bdf file to top level where I then compile to ensure that everything is working properly.


File
Edit
View
Project
Assignments
Processing
Tools
Window
Help
New... Ctrl+N
Open... Ctrl+O
Close Ctrl+F4
New Project Wizard...
Open Project... Ctrl+J
Save Project
Close Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
File Properties...
Create / Update
Export...
Convert Programming Files...
Page Setup...
Print Preview
Print... Ctrl+P
Recent Files
Recent Projects
Exit Alt+F4
multiplexer.bdf*
Create HDL Design File from Current File...
Create Symbol Files for Current File
Create AHDL Include Files for Current File
Create Verilog Instantiation Template Files for Current File
Create VHDL Component Declaration Files for Current File
Create Design File from Selected Block...
Update Design File from Selected Block...
Create Signal Tap File from Design Instance(s)
Create JAM, JBC, SVF, or ISC File...
Create/Update IPS File...
Create Board-Level Boundary-Scan File...
Create Top-Level Design File From Pin Planner...
Assembler (Generate programm
Timing Analysis



Symbol

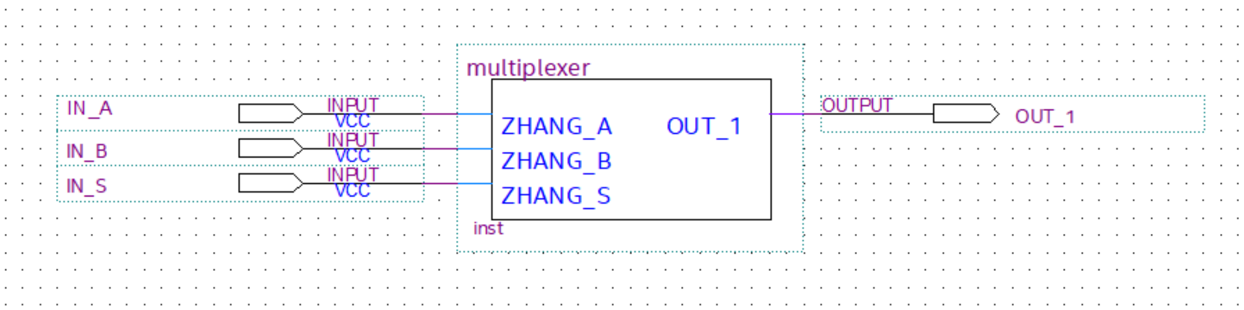
Libraries:

- Project
  - multiplexer
- c:/intelfpga_lite/20.1/quartus/libra

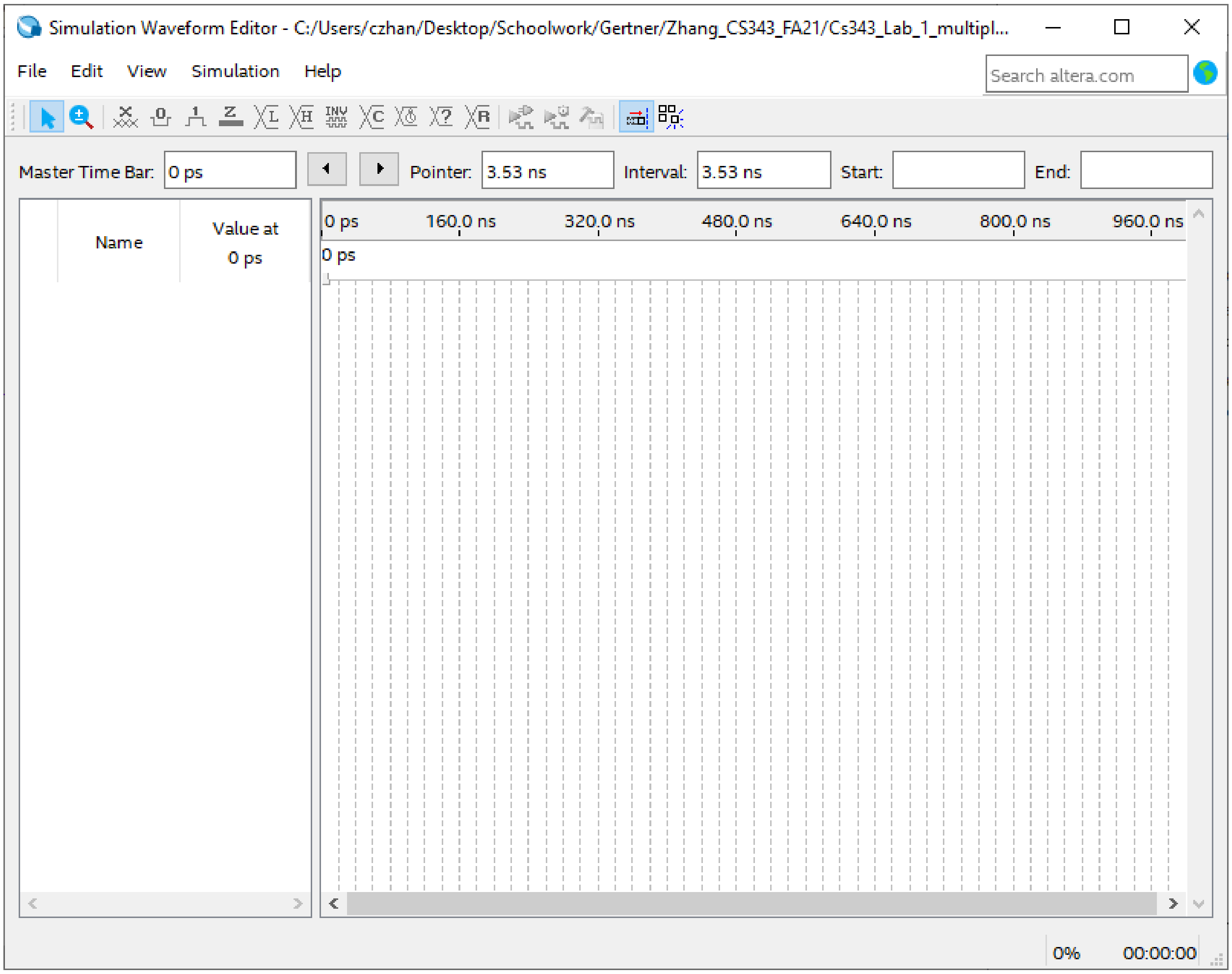
Name: multiplexer

☑ Repeat-insert mode

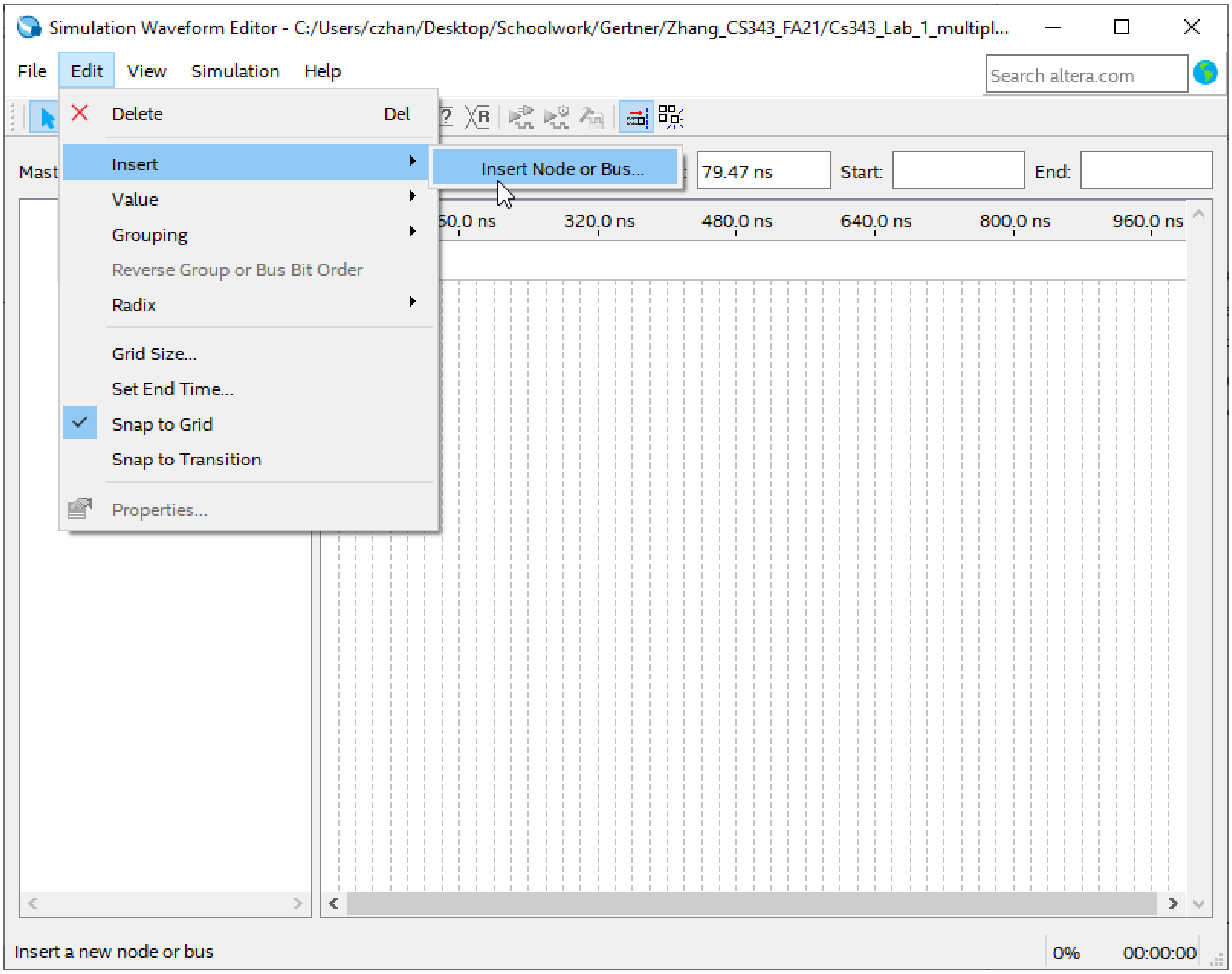
☐ Insert symbol as block



Next, I create a symbol file for the current file and in another block design file, I go into the symbols tool in the project dropdown, I can see the design for my 2:1 MUX. I press OK and put it in the new block design file with 3 inputs and 1 output under the names of IN_A, IN_B, IN_S, OUT_1 Respectively.

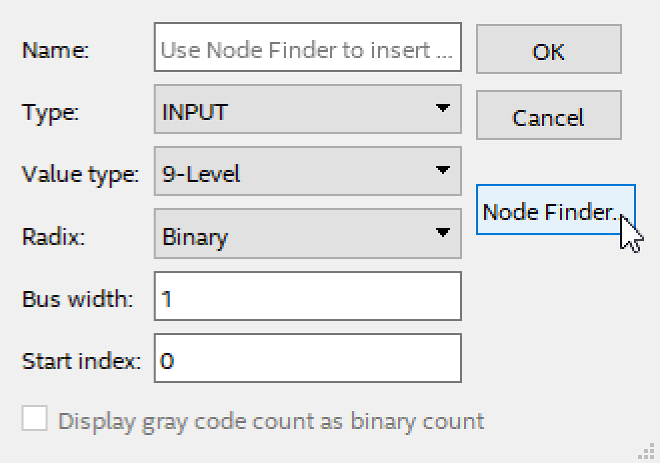


I make a new “University Program VWF” file



Insert a new node or bus

I press Edit -> Insert -> Insert Node or Bus and a pop up should show up.



Name: Use Node Finder to insert ...

Type: INPUT

Value type: 9-Level

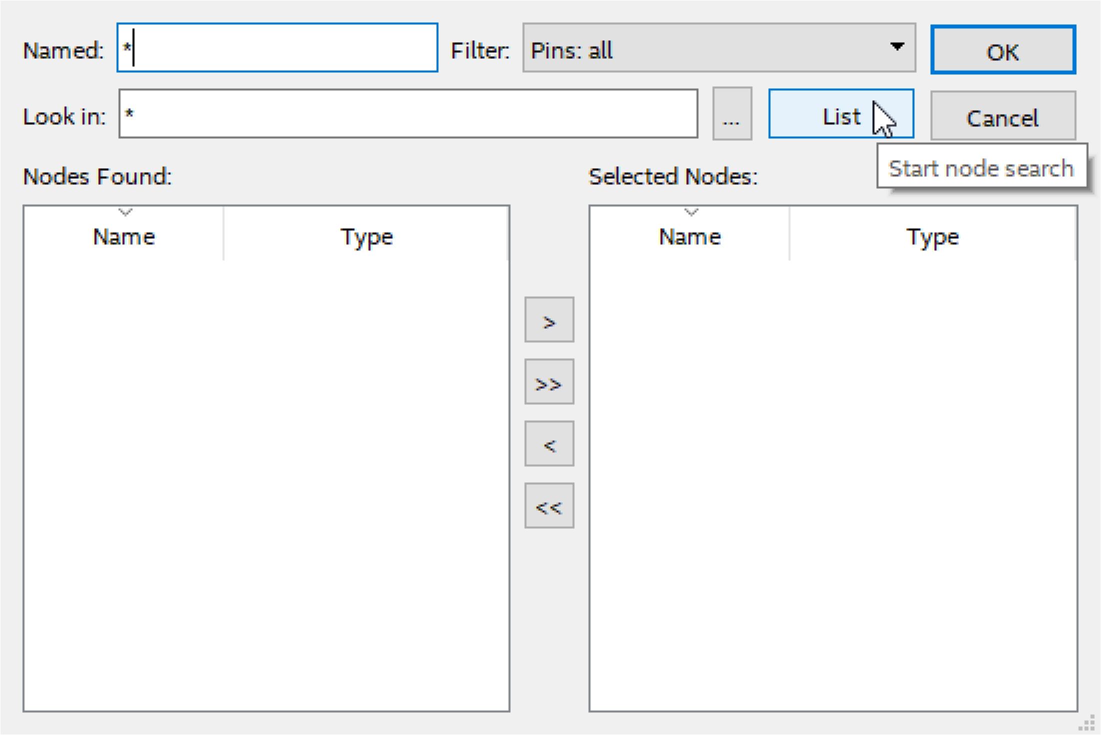
Radix: Binary

Bus width: 1

Start index: 0

□ Display gray code count as binary count

In the popup, press Node Finder and another popup should show up



Named: *

Filter: Pins: all

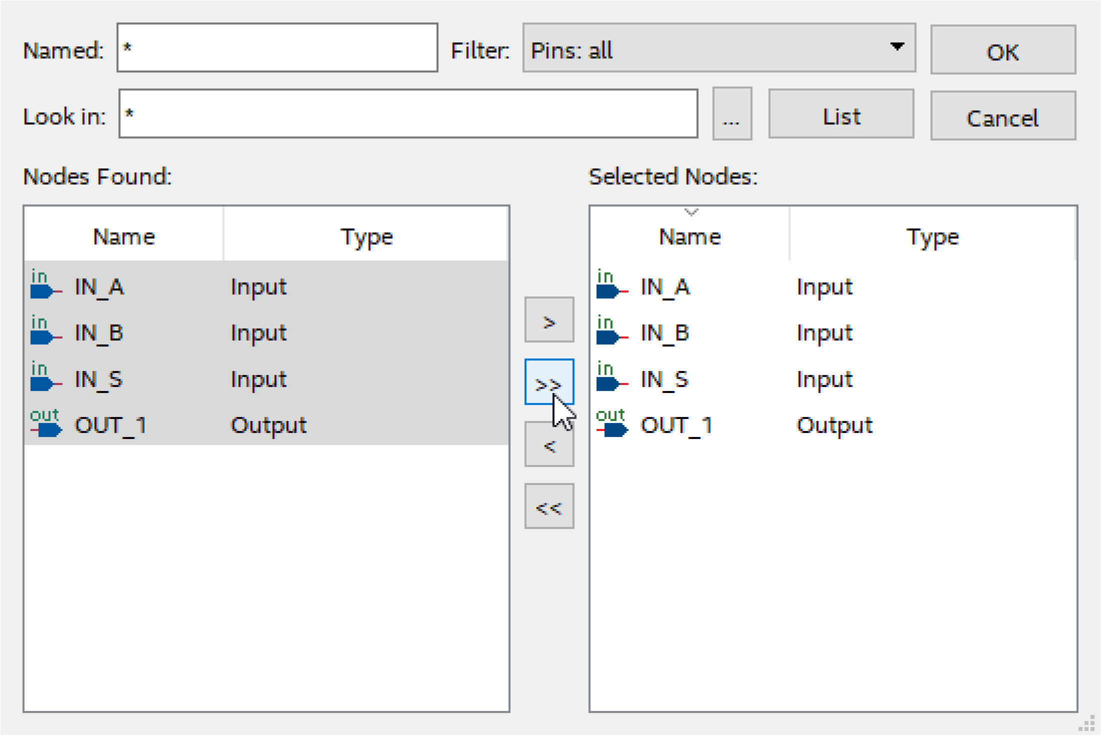
Look in: *

Nodes Found:

Name	Type

Selected Nodes:

Name	Type



Named: *

Filter: Pins: all

Look in: *

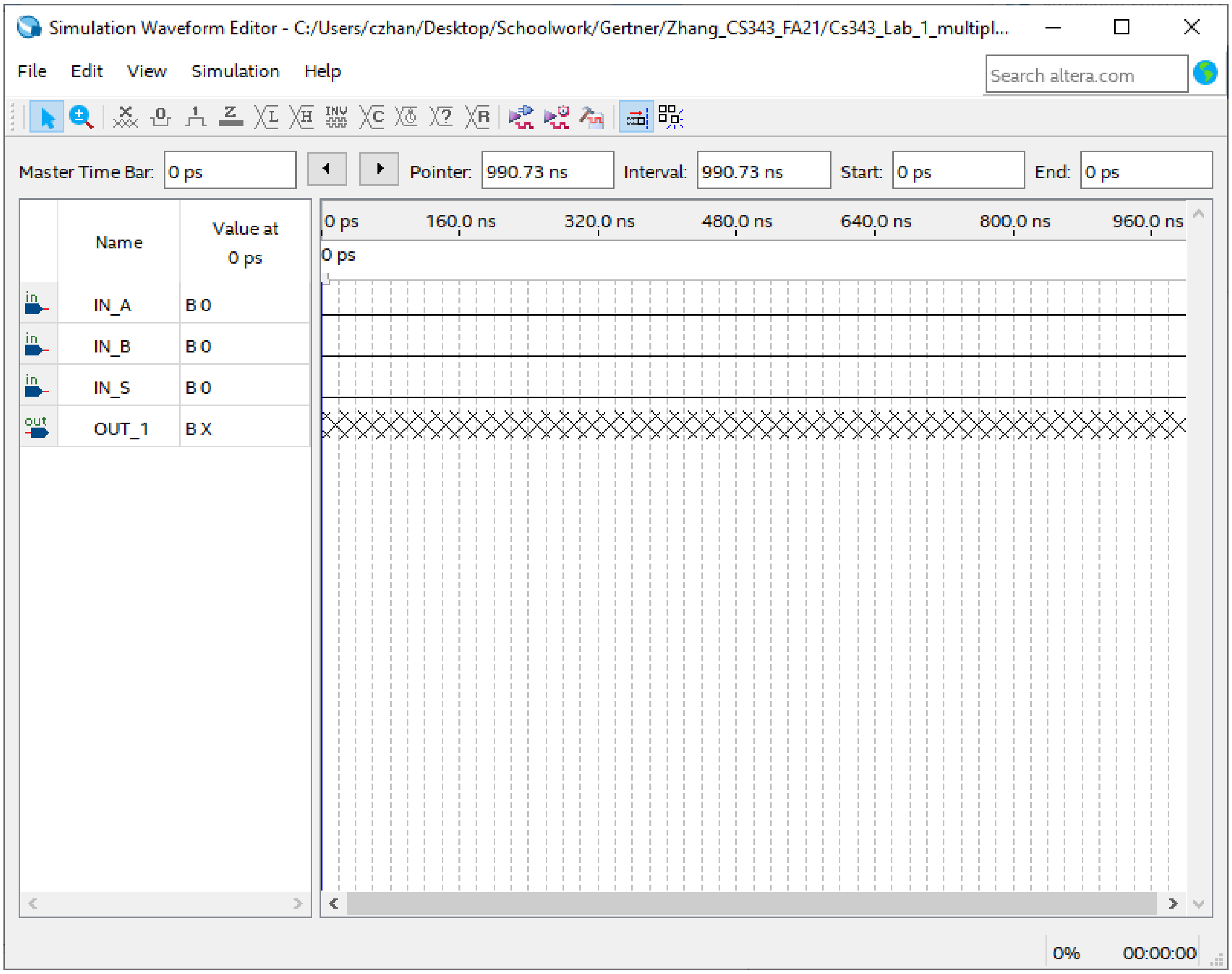
Nodes Found:

Name	Type
IN_A	Input
IN_B	Input
IN_S	Input
OUT_1	Output

Selected Nodes:

Name	Type
IN_A	Input
IN_B	Input
IN_S	Input
OUT_1	Output

In the new popup, press list and a list on the left with all inputs and outputs should show up, then press the >> symbol to select all nodes and press OK.

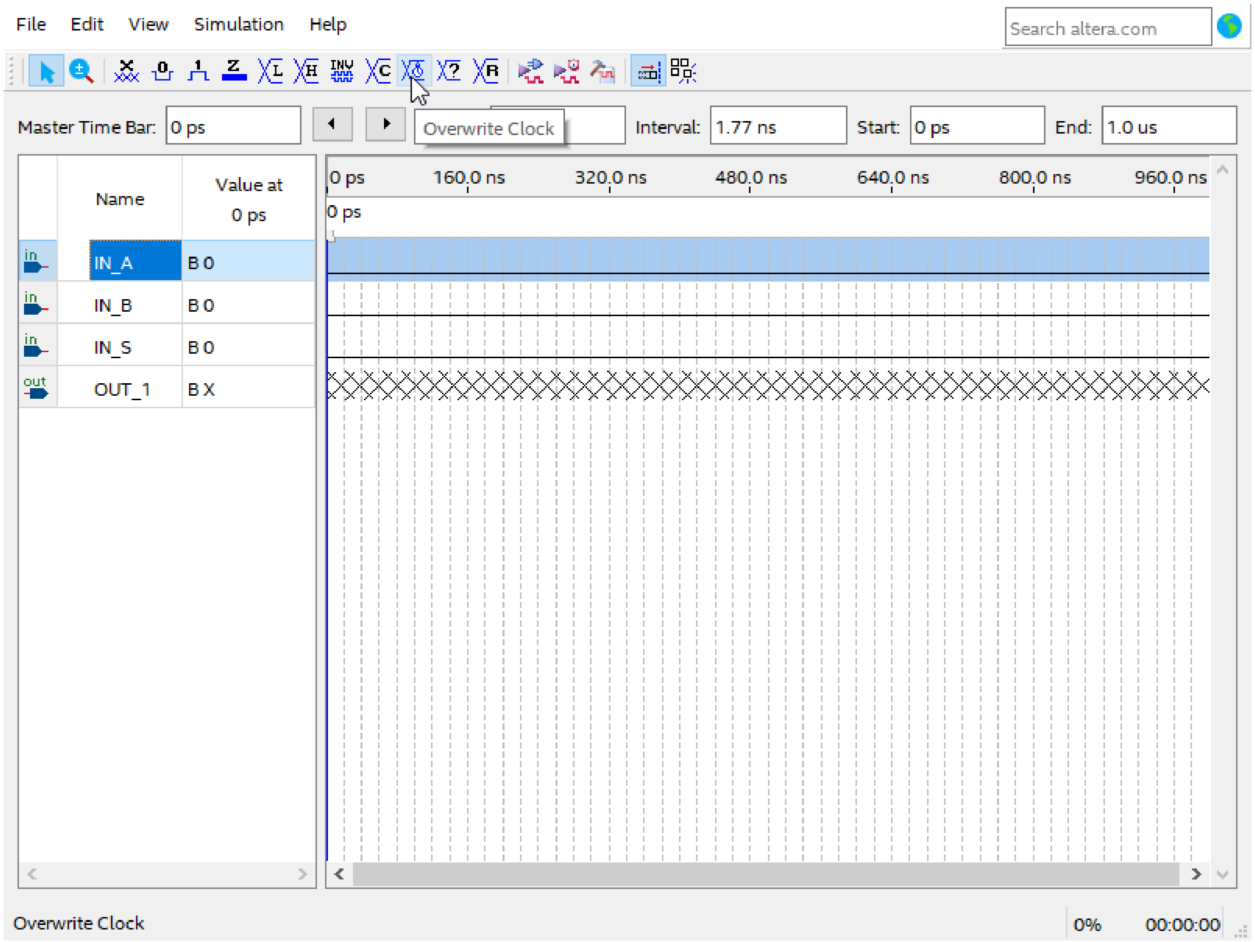


Simulation Waveform Editor - C:/Users/czhan/Desktop/Schoolwork/Gertner/Zhang_CS343_FA21/Cs343_Lab_1_multipl...
File
Edit
View
Simulation
Help
Search altera.com
0 ps
160.0 ns
320.0 ns
480.0 ns
640.0 ns
800.0 ns
960.0 ns
0 ps
0%
00:00:00

Master Time Bar: 0 ps
Pointer: 990.73 ns
Interval: 990.73 ns
Start: 0 ps
End: 0 ps

	Name	Value at 0 ps
in	IN_A	B 0
in	IN_B	B 0
in	IN_S	B 0
out	OUT_1	B X

This is what you should see and next I will start manually inputting values for the inputs so that we can generate the appropriate output for the MUX.



Master Time Bar: 0 ps

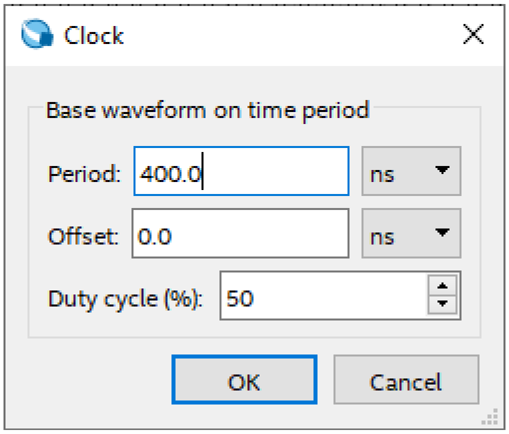
Interval: 1.77 ns

Start: 0 ps

End: 1.0 us

	Name	Value at 0 ps
in	IN_A	B 0
in	IN_B	B 0
in	IN_S	B 0
out	OUT_1	B X

In the same file, I press on IN_A -> overwrite Clock symbol which allows me to change the values of the input at a set interval. (A POPUP SHOULD SHOW UP)

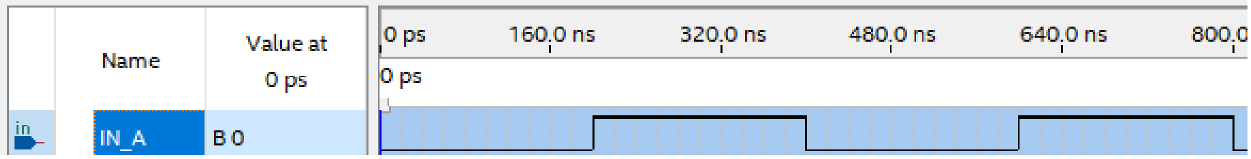


Base waveform on time period

Period: 400.0 ns

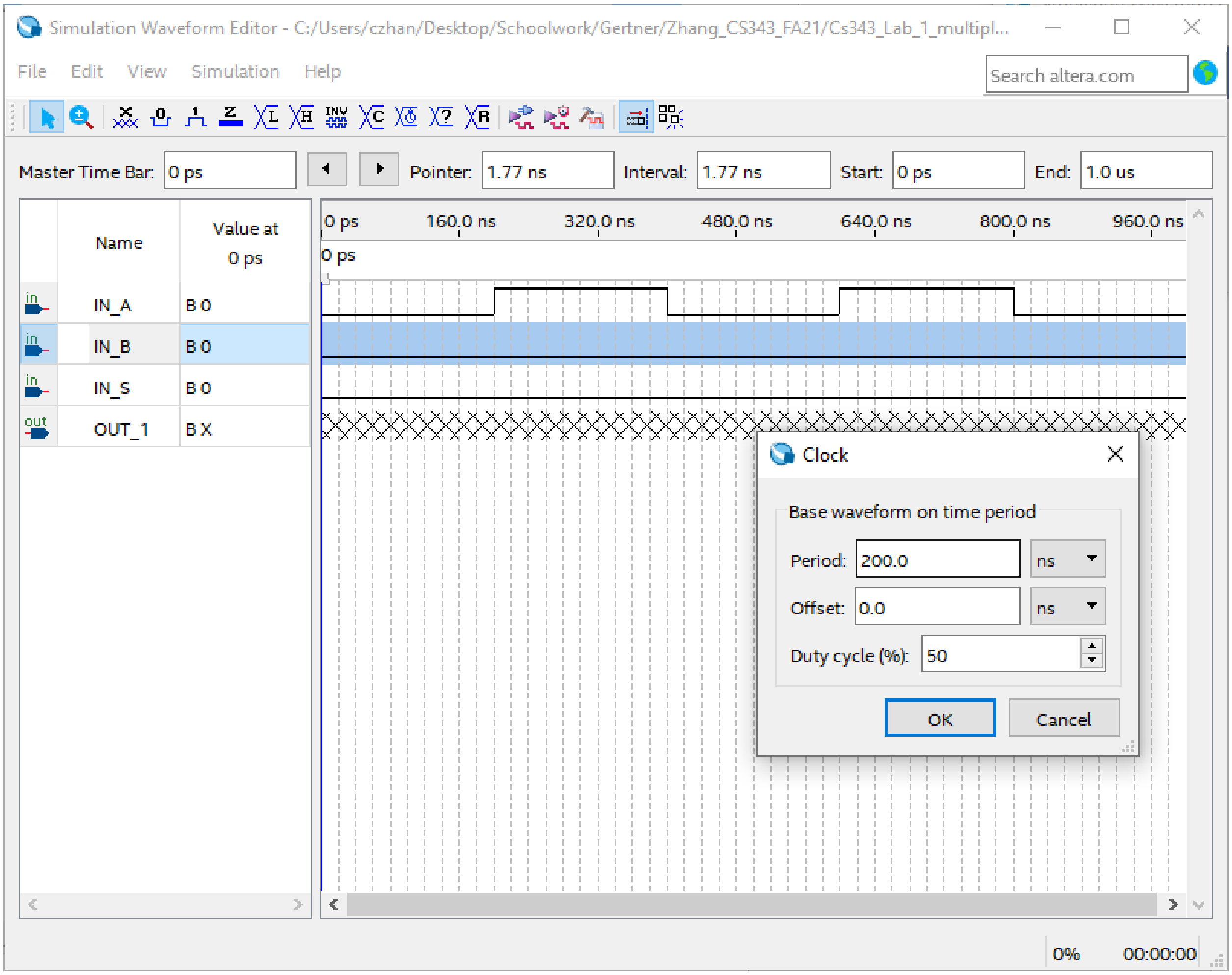
Offset: 0.0 ns

Duty cycle (%): 50



Name	Value at 0 ps
IN_A	B 0

In the popup window, I set the period to 400 ns which makes it so that the value is at 1 at 200ns-400ns and 1 at 600ns-800ns.



Master Time Bar: 0 ps Pointer: 1.77 ns Interval: 1.77 ns Start: 0 ps End: 1.0 us

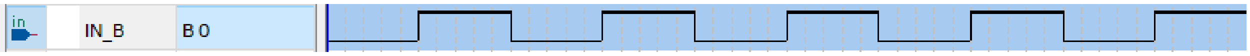
Name	Value at 0 ps
IN_A	B 0
IN_B	B 0
IN_S	B 0
OUT_1	B X

Base waveform on time period

Period: 200.0 ns

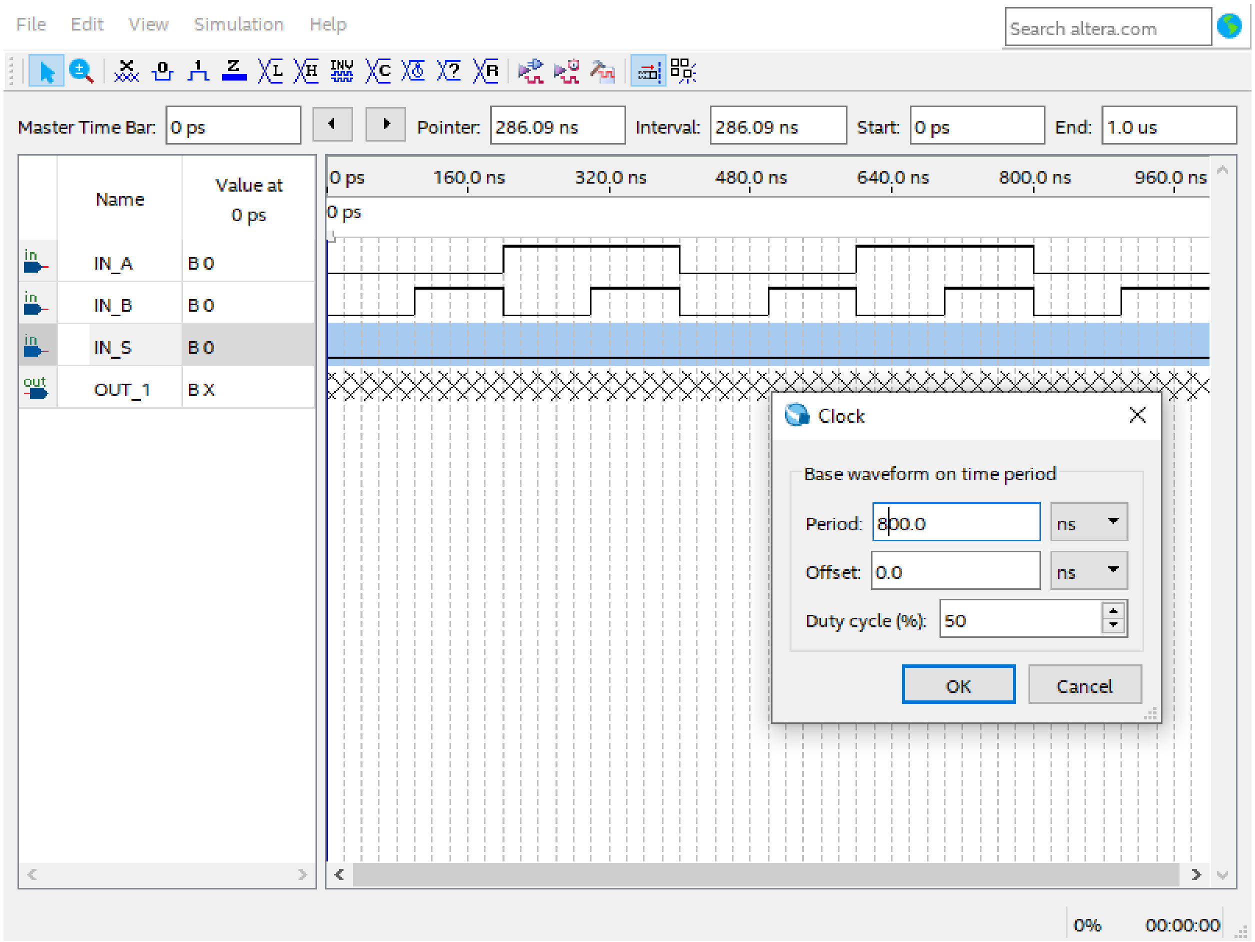
Offset: 0.0 ns

Duty cycle (%): 50



IN_B B 0

I repeat the same steps as above for IN_B but instead of 400ns, I do 200ns.



Master Time Bar: 0 ps

Pointer: 286.09 ns

Interval: 286.09 ns

Start: 0 ps

End: 1.0 us

Name	Value at 0 ps
IN_A	B 0
IN_B	B 0
IN_S	B 0
OUT_1	B X

Clock

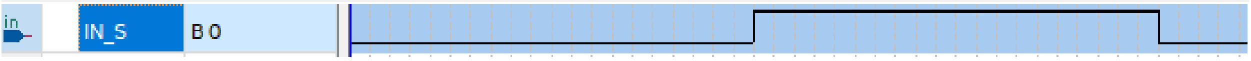
Base waveform on time period

Period: 800.0 ns

Offset: 0.0 ns

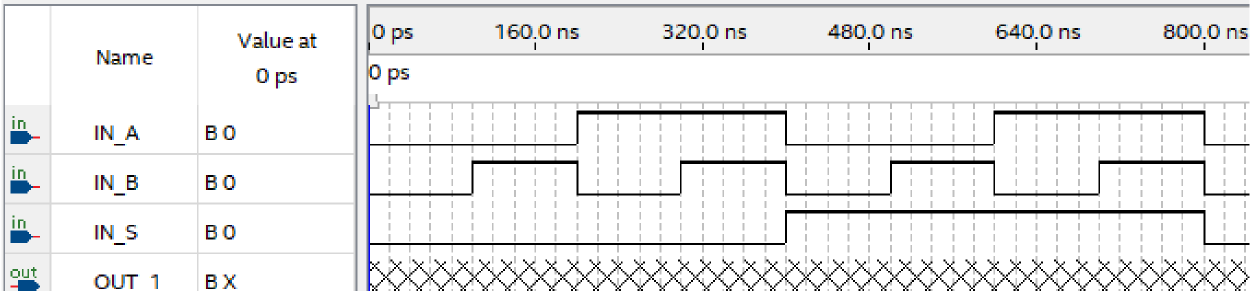
Duty cycle (%): 50

OK Cancel



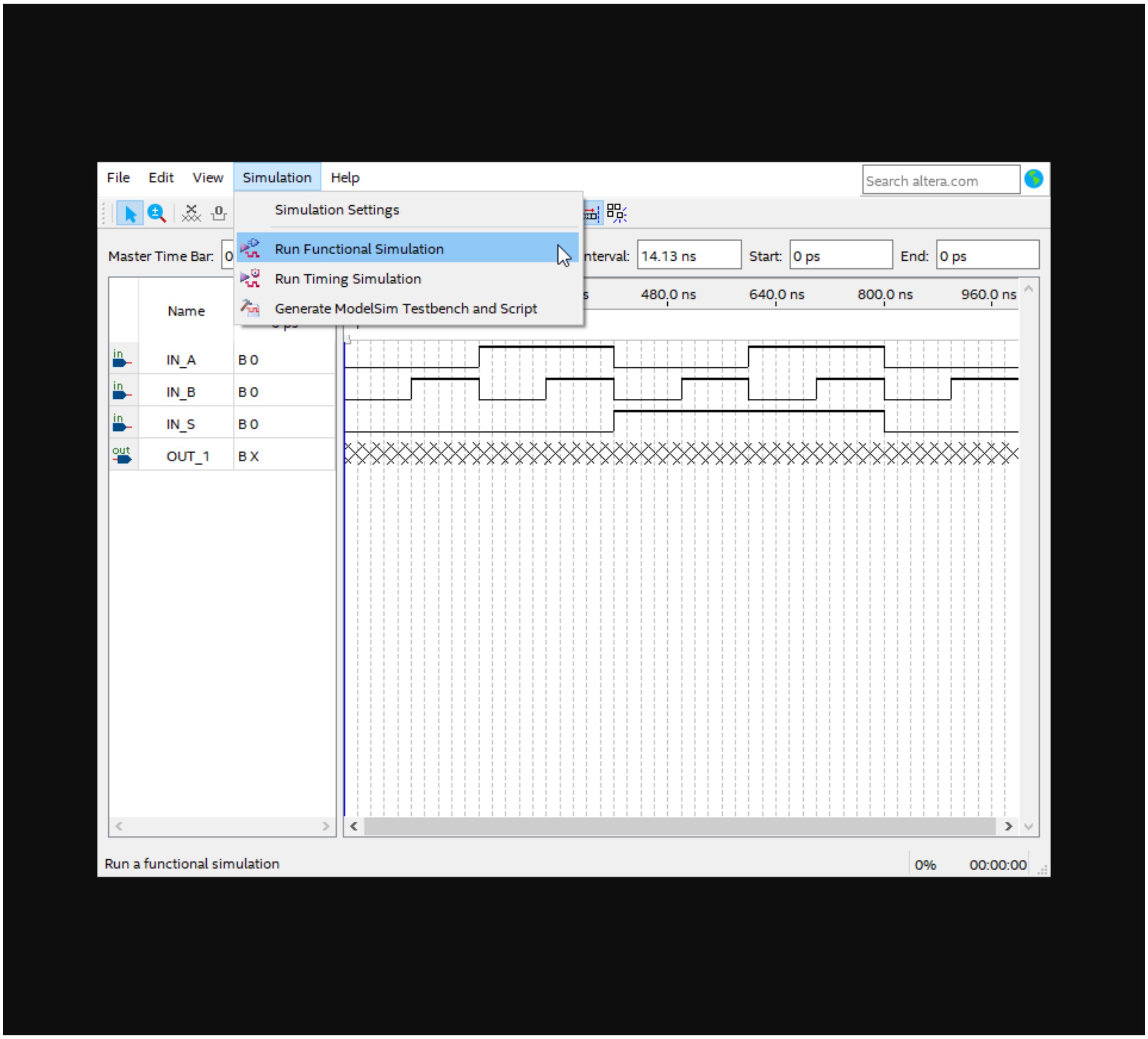
IN_S B 0

I repeat the same step again for IN_S but with 800ns.



Name	Value at 0 ps
IN_A	B 0
IN_B	B 0
IN_S	B 0
OUT_1	B X

My WVF now looks like this and I can begin the simulation

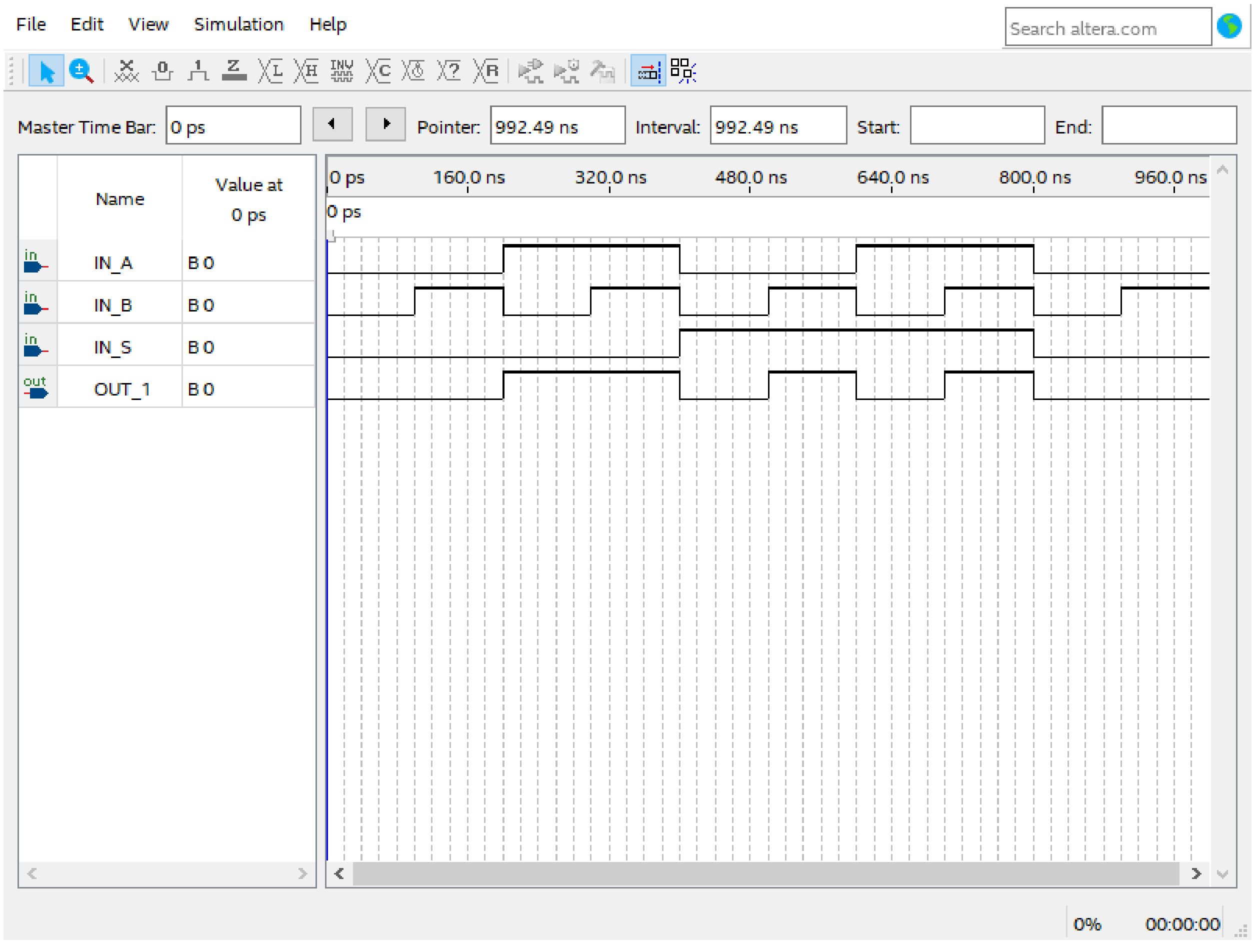


Master Time Bar: 0 nterval: 14.13 ns Start: 0 ps End: 0 ps

Name	
IN_A	B 0
IN_B	B 0
IN_S	B 0
OUT_1	B X

Run a functional simulation 0% 00:00:00

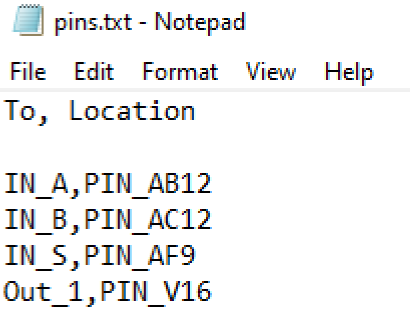
I go to the simulation drop down and press run functional simulation



Master Time Bar: 0 ps   Pointer: 992.49 ns   Interval: 992.49 ns   Start: End:

Name	Value at 0 ps
IN_A	B 0
IN_B	B 0
IN_S	B 0
OUT_1	B 0

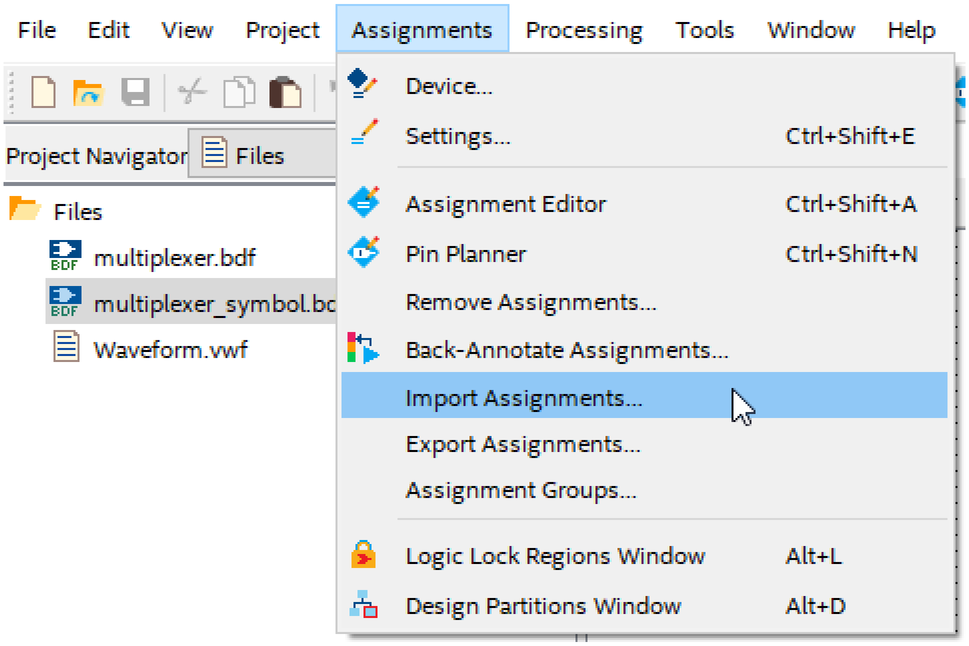
Afterwards, this should be the result. A WVF file with a proper line output instead of X's.

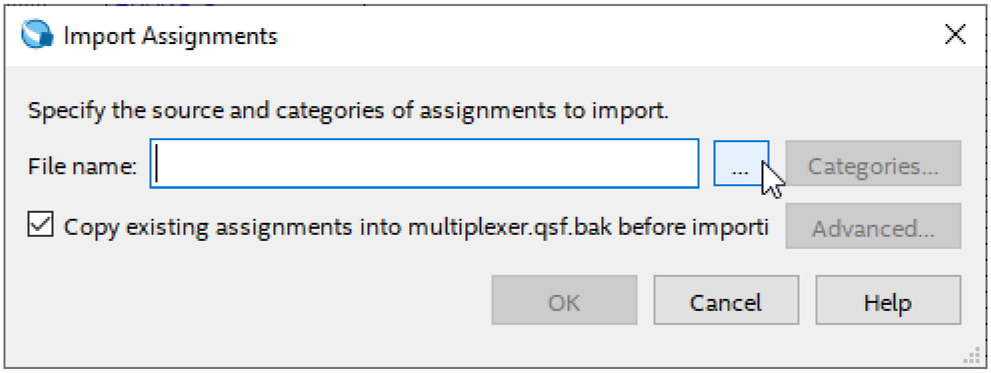


```
To, Location

IN_A,PIN_AB12
IN_B,PIN_AC12
IN_S,PIN_AF9
Out_1,PIN_V16
```

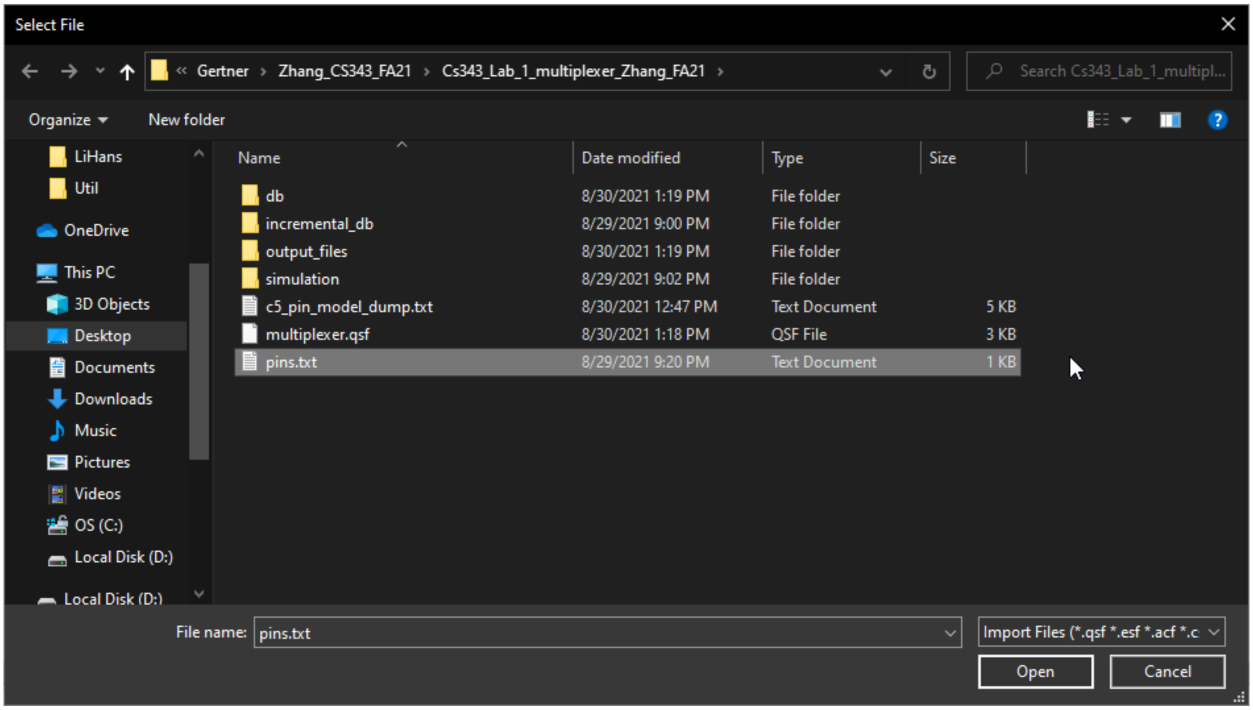
Next I create a text file with the naming convention "To, Location" where the To describes where the pin is assigned to and which pin to assign.





Specify the source and categories of assignments to import.

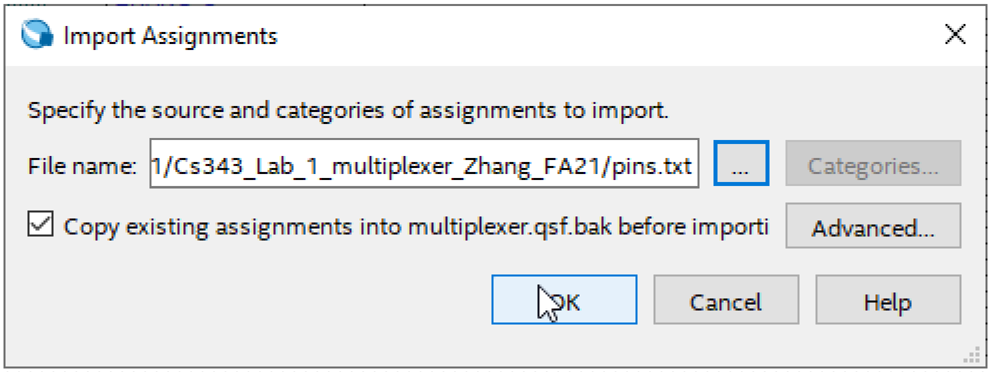
☑ Copy existing assignments into multiplexer.qsf.bak before importi



Name	Date modified	Type	Size
db	8/30/2021 1:19 PM	File folder	
incremental_db	8/29/2021 9:00 PM	File folder	
output_files	8/30/2021 1:19 PM	File folder	
simulation	8/29/2021 9:02 PM	File folder	
c5_pin_model_dump.txt	8/30/2021 12:47 PM	Text Document	5 KB
multiplexer.qsf	8/30/2021 1:18 PM	QSF File	3 KB
pins.txt	8/29/2021 9:20 PM	Text Document	1 KB

File name: pins.txt

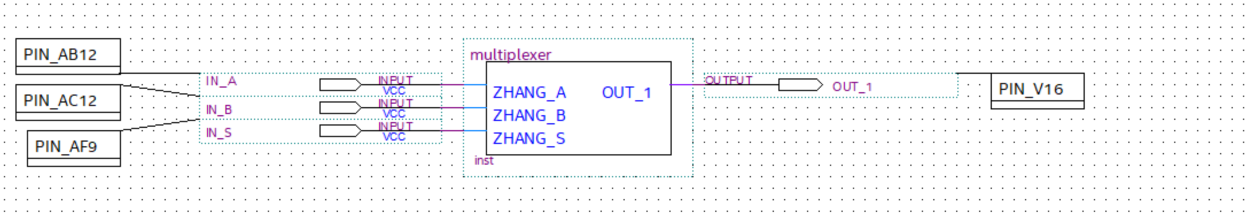
Import Files (*.qsf *.esf *.acf *.c



Specify the source and categories of assignments to import.

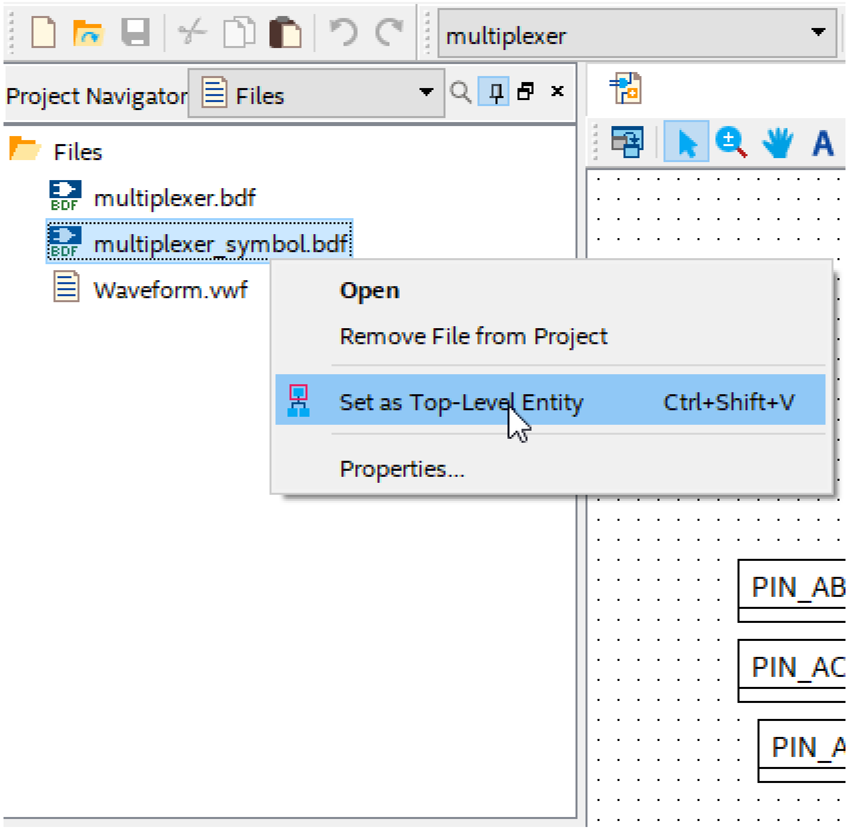
File name: 1/Cs343_Lab_1_multiplexer_Zhang_FA21/pins.txt

☑ Copy existing assignments into multiplexer.qsf.bak before importi

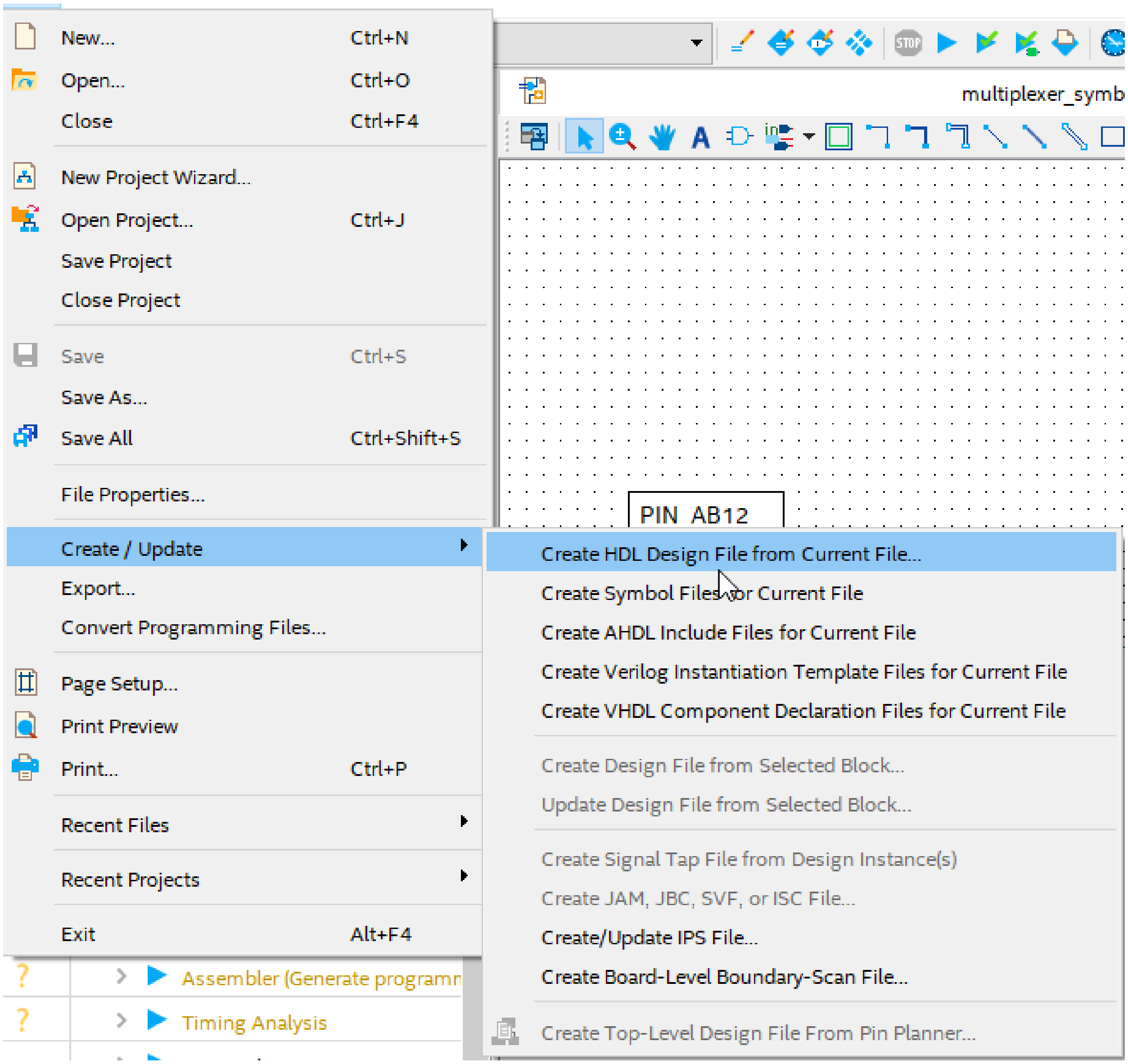


The next step is to go to the assignments and import assignments. Then press the … when the popup shows which should allow you to select a file. Select the pins files that was created before and the output should show in the design file.

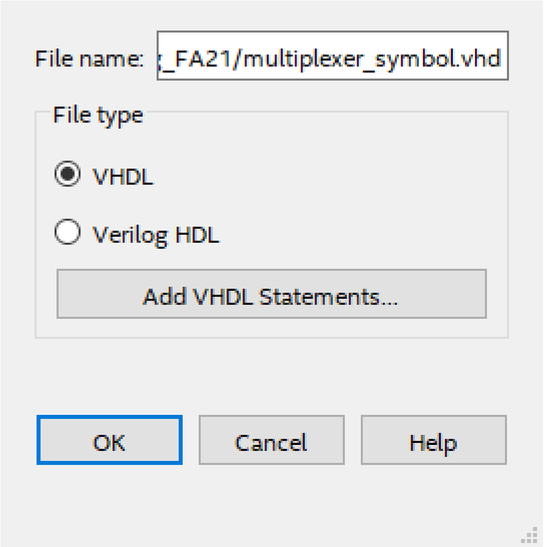
Task 3



It is important to set the correct top level entity, in my case it is the multiplexer_symbol.bdf which is basically the MUX as a symbol.


New... Ctrl+N
Open... Ctrl+O
Close Ctrl+F4
New Project Wizard...
Open Project... Ctrl+J
Save Project
Close Project
Save Ctrl+S
Save As...
Save All Ctrl+Shift+S
File Properties...
Create / Update
Export...
Convert Programming Files...
Page Setup...
Print Preview
Print... Ctrl+P
Recent Files
Recent Projects
Exit Alt+F4
Assembler (Generate programm
Timing Analysis
multiplexer_symb
PIN AB12
Create HDL Design File from Current File...
Create Symbol Files for Current File
Create AHDL Include Files for Current File
Create Verilog Instantiation Template Files for Current File
Create VHDL Component Declaration Files for Current File
Create Design File from Selected Block...
Update Design File from Selected Block...
Create Signal Tap File from Design Instance(s)
Create JAM, JBC, SVF, or ISC File...
Create/Update IPS File...
Create Board-Level Boundary-Scan File...
Create Top-Level Design File From Pin Planner...

Next go to files -> create/Update -> create HDL Design File from Current File…



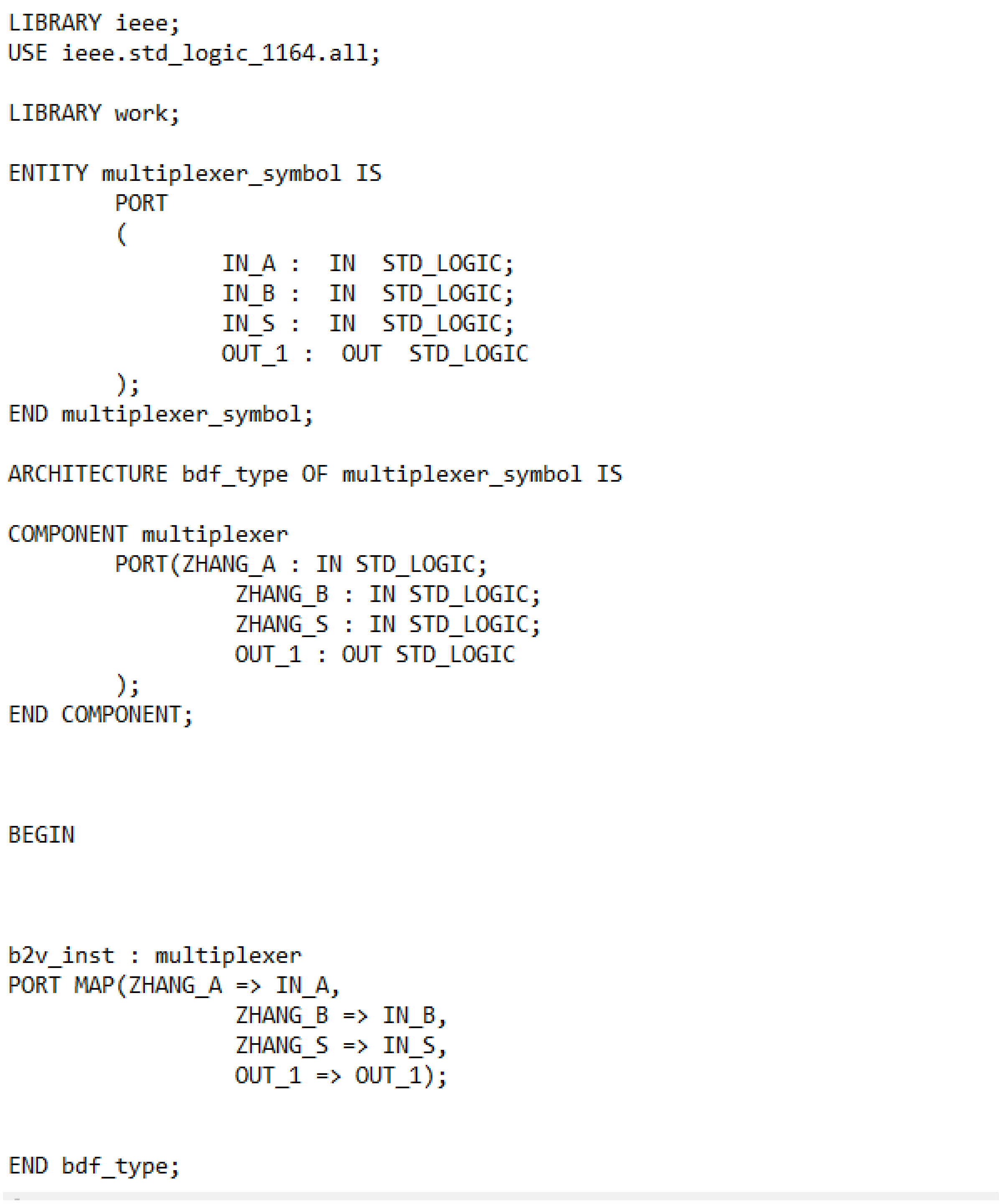
File name: ;_FA21/multiplexer_symbol.vhd

File type

● VHDL

○ Verilog HDL

A popup should show looking like this. Make sure the file type is in VHDL and not Verilog HDL



```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

LIBRARY work;

ENTITY multiplexer_symbol IS
        PORT
        (
                IN_A :  IN  STD_LOGIC;
                IN_B :  IN  STD_LOGIC;
                IN_S :  IN  STD_LOGIC;
                OUT_1 :  OUT  STD_LOGIC
        );
END multiplexer_symbol;

ARCHITECTURE bdf_type OF multiplexer_symbol IS

COMPONENT multiplexer
        PORT(ZHANG_A : IN STD_LOGIC;
                 ZHANG_B : IN STD_LOGIC;
                 ZHANG_S : IN STD_LOGIC;
                 OUT_1 : OUT STD_LOGIC
        );
END COMPONENT;



BEGIN



b2v_inst : multiplexer
PORT MAP(ZHANG_A => IN_A,
                 ZHANG_B => IN_B,
                 ZHANG_S => IN_S,
                 OUT_1 => OUT_1);


END bdf_type;
```

A vhd file should be generated now in your directory. Go into your directory and open the vhd file. I chose to use notepad to observe my vhd code.