

ECGR 4146 Introduction to VHDL

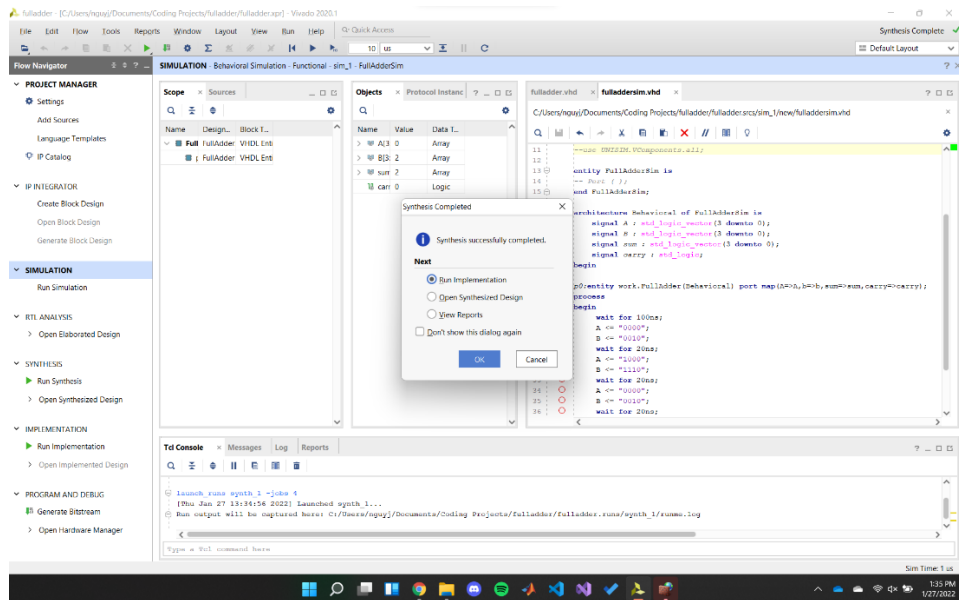
Lab 2

Jonathon Nguyen

ID: 801093003

1. Take a screenshot of Vivado screen after you have run synthesis, and the simulation

I added the full adder code to fulladder.vhd and added the testbench code to fulladdersim.vhd. I ran the synthesis on the code and screenshotted the completed screen. Then I ran the a simulation on the full adder code.

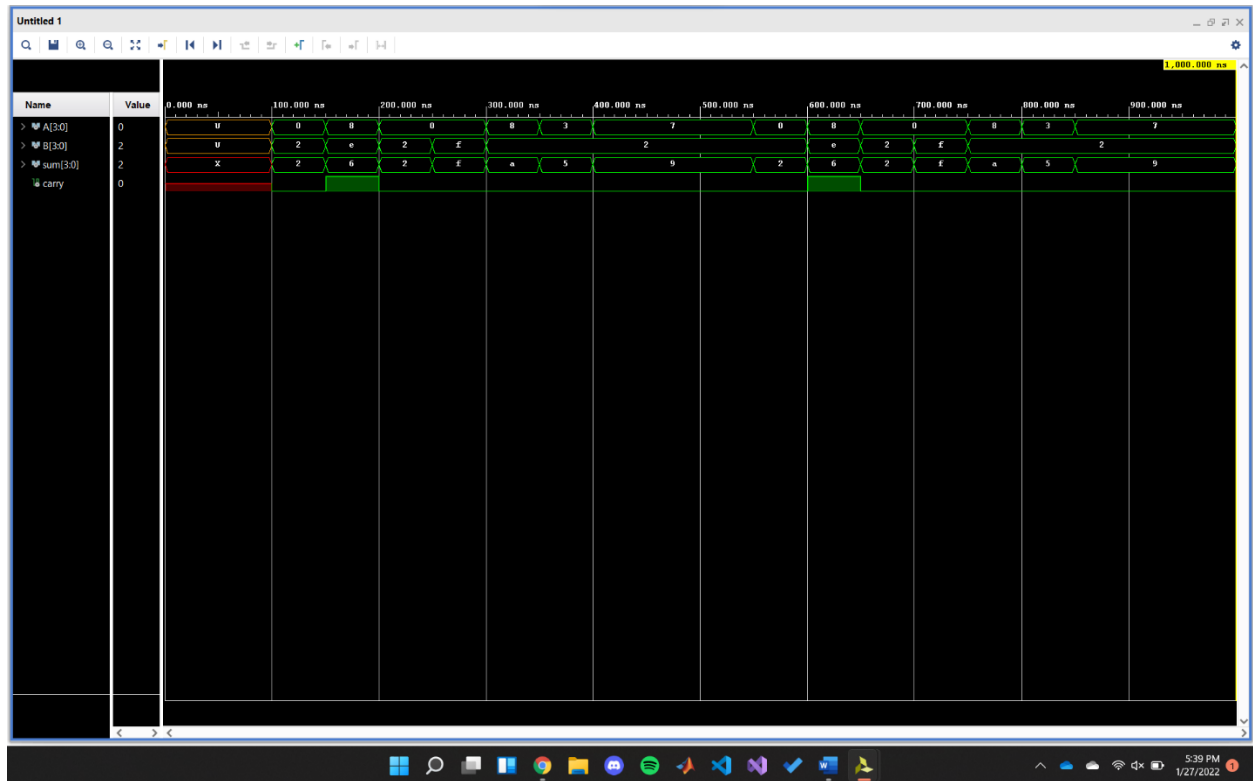


Synthesis screenshot



Simulation screenshot.

During the lab checkout, I changed the testbench to include more test and changed some of the old test. These are the simulation results.



2. Once done with the simulation you need to make some changes in the adder.vhd code to generate the bitstream.

Here I made the adjustments to the code. The changes are found on the right side of the screenshot. The changes that was made was to change the both input to 2 bits and the output to 2 bits. Also in the architecture the temp signal was uses 3 bits instead of 5 bits.

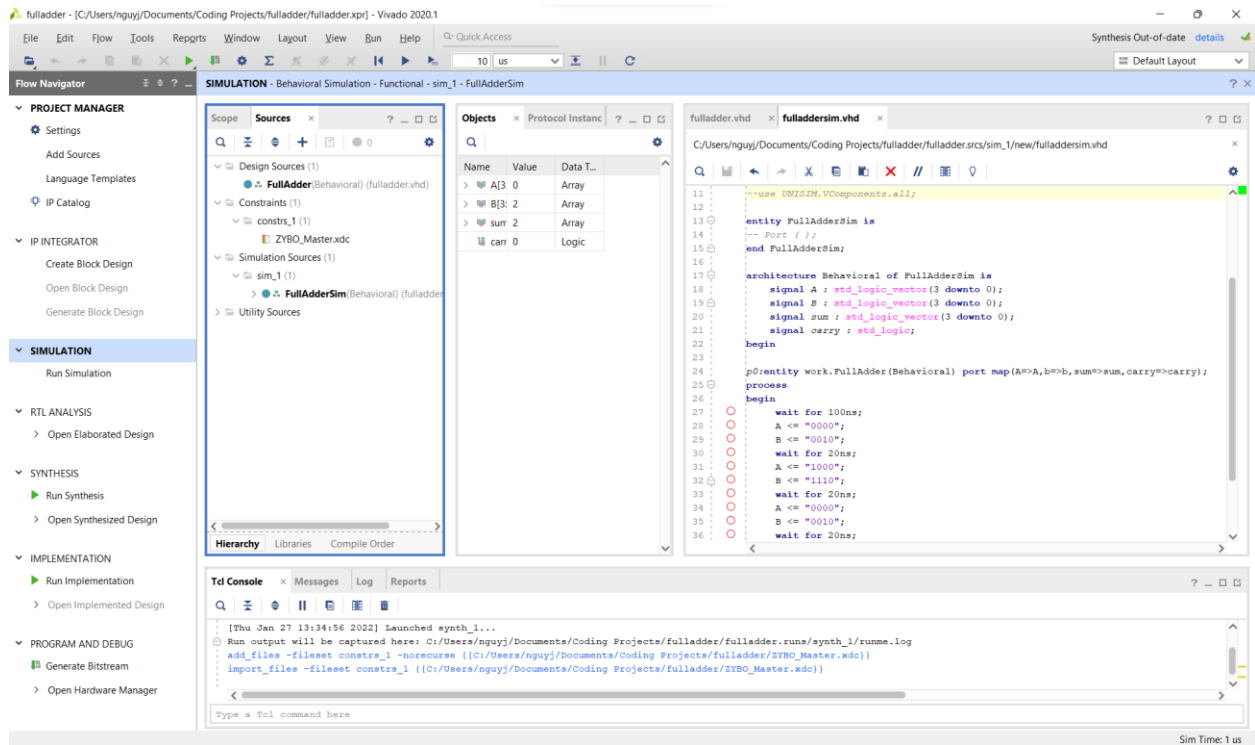
The screenshot displays the Vivado IDE interface. On the left, the 'Sources' pane shows the project structure with 'FullAdder(Behavioral) (fulladder.vhd)' selected. Below it, the 'Source File Properties' pane for 'fulladdersim.vhd' shows it is enabled and located at 'C:/Users/nguyj/Documents/Coding Projects/fulladder/fulladder.srcs/sim'. The main editor window shows the 'fulladder.vhd' code. The code defines an entity 'FullAdder' with two 1-bit inputs 'A' and 'B', and two 1-bit outputs 'sum' and 'carry'. The architecture 'Behavioral of FullAdder' implements the logic using 'std_logic_vector' and 'std_logic' types. The 'temp' signal is declared as a 2-bit vector. The 'Design Runs' table at the bottom shows the synthesis and implementation status.

```
10 library UNISIM;
11 use UNISIM.VComponents.all;
12
13 entity FullAdder is
14     Port (
15         A: in std_logic_vector(1 downto 0);
16         B: in std_logic_vector(1 downto 0);
17         sum: out std_logic_vector(1 downto 0);
18         carry: out std_logic
19     );
20 end FullAdder;
21
22 architecture Behavioral of FullAdder is
23     signal temp : std_logic_vector(2 downto 0);
24     begin
25         sum <= std_logic_vector( unsigned(A) + unsigned(B) );
26         temp <= std_logic_vector ("0"&unsigned(A) + unsigned(B));
27         carry <= temp(2);
28     end Behavioral;
29
30 end Behavioral;
31
```

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAM	URAM	DSP	Start	Elapsed	Run Strategy
synth_1	constrs_1	Synthesis Out-of-date								4	0	0.0	0	0	1/27/22, 1:35 PM	00:00:10	Vivado Synthesis Defaults (Vivado)
impl_1	constrs_1	Not started															Vivado Implementation Defaults (Vivado)

3. Add the constraint file provided in the constraint tab of vivado.

I downloaded the constraint file from canvas. Then went to add sources and added the constraint file. This is a screenshot of the constraint file in the project tab on the left of the picture.



4. Run synthesis, post place and route, and Generate Bitstream successfully.

I ran the synthesis, implementation and generated the bitstream. This is a screenshot of the completion screen of the generating the bitstream.

