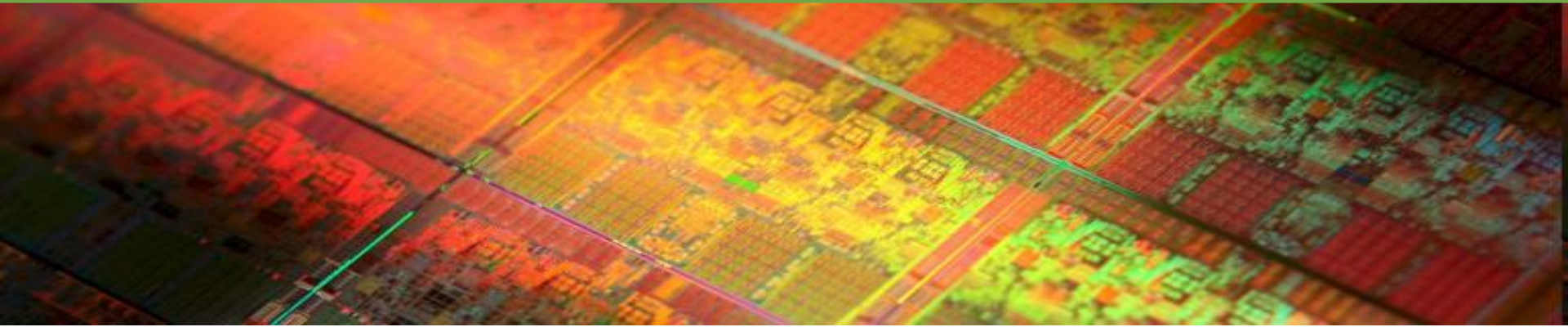


EE 2000

Logic Circuit Design (Fall 2018)



Tutorial 2

Simulate your Logic Design
Step-by-Step

Instructor: Dr. Ray C.C. Cheung

Objectives

- Learn how to use this design tool to simulate your logic design
- Simulate the first VHDL code
 - A few given examples
- Experience different Vivado implementation steps
- Try to modify the VHDL Code

Extension from Tutorial 1

The screenshot displays the Vivado 2018.2.1 IDE interface. The top menu bar includes File, Edit, Flow, Tools, Reports, Window, Layout, View, and Help. The left sidebar contains the Flow Navigator and a Project Manager pane. The Project Manager pane shows the project structure for 'Tuto2', including Design Sources (1) and Simulation Sources (1). The Source File Properties pane for 'Tuto2.vhd' shows it is enabled and located at 'C:/Users/racheung/Desktop/Tuto2/Tuto2.srcs/sources_1/new/Tuto2.vhd'. The main editor window displays the VHDL code for 'Tuto2.vhd'.

```
--use UNISIM.VComponents.all;

entity Tuto2 is
    Port ( A : in STD_LOGIC;
          B : in STD_LOGIC;
          S : out STD_LOGIC );
end Tuto2;

architecture Behavioral of Tuto2 is
begin
    S <= A and B;
end Behavioral;
```

The bottom pane shows the Design Runs table, which lists the synthesis and implementation runs for the project.

Name	Constraints	Status	WNS	TNS	WHS	THS	TPWS	Total Power	Failed Routes	LUT	FF	BRAMs	URAM	DSP	Start
synth_1	constrs_1	Not started													
impl_1	constrs_1	Not started													

Add Simulation Source

- Click -> Add Source -> Add or create simulation source -> Create File
- Create a new testbench file: AND_tb.vhd
- No need to specify the I/O Ports

```
entity AND_tb is  
  // empty  
end AND_tb;
```

Update the file

- Add the following lines before “begin”

```
component Tuto2
    port (A : in std_logic;
          B : in std_logic;
          S : out std_logic);
end component;
```

```
signal A : std_logic;
signal B : std_logic;
signal S : std_logic;
```

Unit Under Test (UUT)

- uut : Tuto2 port map (A => A, B => B, S => S);

The screenshot displays a VHDL development environment. On the left, the 'Objects' window shows a table of signals:

Name	Value	Data Type
A	1	Logic
B	1	Logic
S	1	Logic

On the right, the 'AND_tb.vhd' source file is open, showing the following VHDL code:

```
32  --use UNISIM.VComponents.all;
33
34  entity AND_tb is
35  end AND_tb;
36
37  architecture Behavioral of AND_tb is
38
39      component Tuto2
40      port (A : in std_logic;
41           B : in std_logic;
42           S : out std_logic);
43      end component;
44
45      signal A : std_logic;
46      signal B : std_logic;
47      signal S : std_logic;
48
49  begin
50
51      uut : Tuto2
52      port map (A => A,
53               B => B,
54               S => S);
55  end;
```

Add Your Simulation Logic

- This part is up to your own choice.

stimuli : process

begin

A <= '0';

B <= '0';

wait for 100 ns;

A <= '1';

wait for 100 ns;

B <= '1';

wait;

end process;

```
architecture Behavioral of AND_tb is
```

```
    component Tuto2
```

```
        port (A : in std_logic;
```

```
              B : in std_logic;
```

```
              S : out std_logic);
```

```
    end component;
```

```
    signal A : std_logic;
```

```
    signal B : std_logic;
```

```
    signal S : std_logic;
```

```
begin
```

```
    uut : Tuto2
```

```
    port map (A => A,
```

```
              B => B,
```

```
              S => S);
```

```
    stimuli : process
```

```
    begin
```

```
        A <= '0';
```

```
        B <= '0';
```

```
        wait for 100 ns;
```

```
        A <= '1';
```

```
        wait for 100 ns;
```

```
        B <= '1';
```

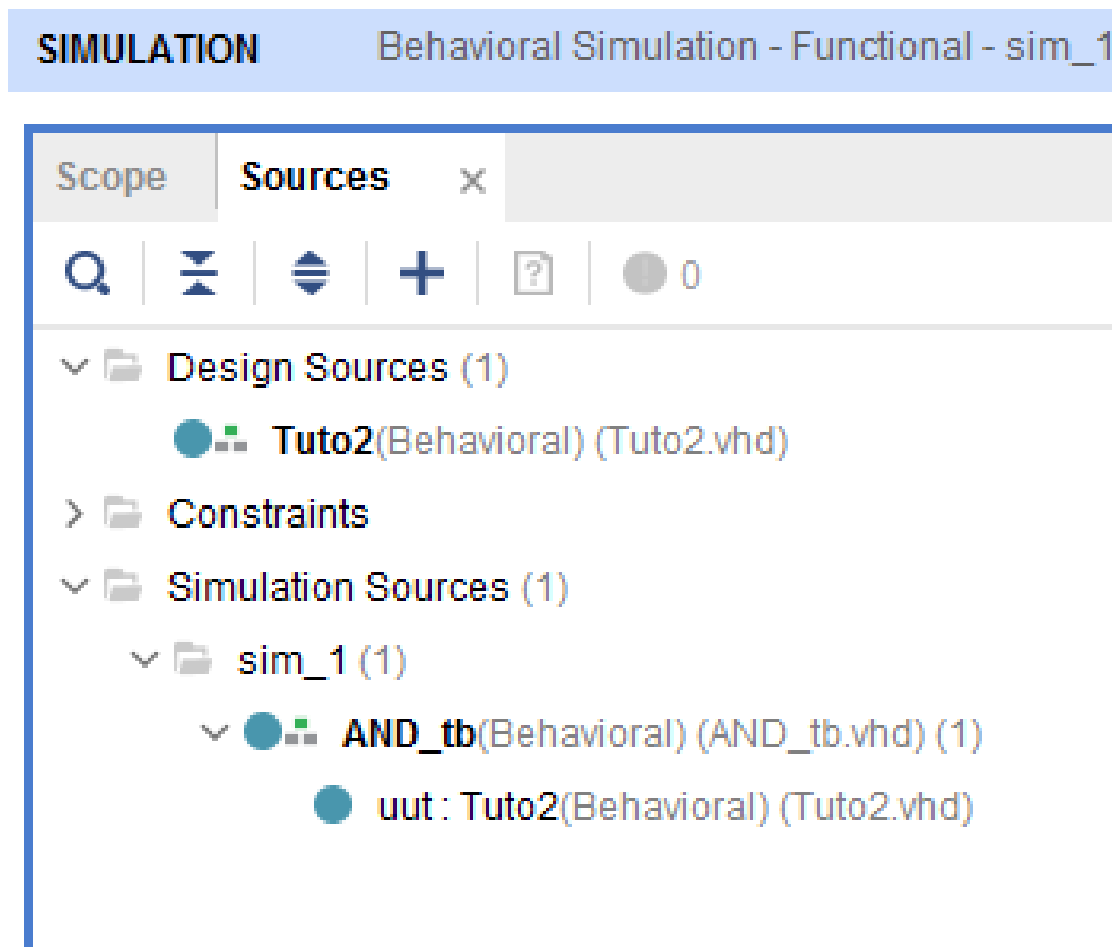
```
        wait;
```

```
    end process;
```

```
end Behavioral;
```

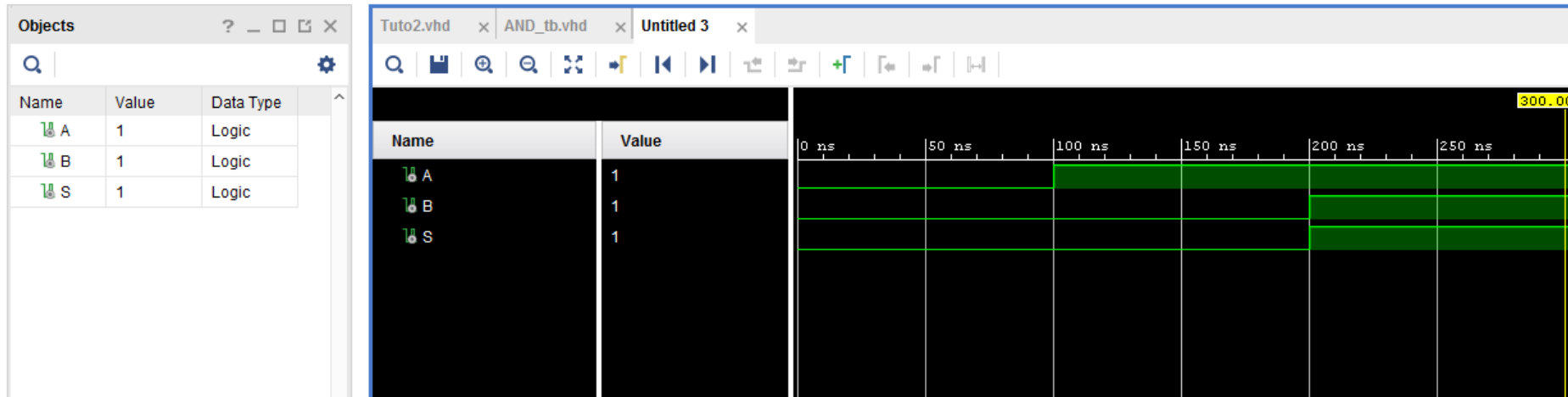
Important - Please Check

- Check the file hierarchy, where is UUT?



Let's Simulate

- Go to the Flow Manager -> Simulation -> Run Simulation -> Run Behavioral Simulation
- Make sure you Zoom to the right level



A Simple Boolean Function in VHDL

Instruction

1. Prepare your own 3-input logic function, a simple Boolean function $D = f(A, B, C)$;
2. you can draw any f you like, such as

$$D = A \ \&\& \ B \ || \ !C$$

3. Now please prepare your own VHDL source code

A Simple Boolean Function in Truth Table

Instruction

Draw the Truth Table according to the function f you choose

(The truth table should be like the one shown on the right side if

$D = A \& \& B \ || \ ! C$);

Truth Table

ABC	D
000	1
001	0
010	1
011	0
100	1
101	0
110	0
111	1

Add Your Simple Design to a Project

Instruction

4. Start Vivado; create a new project;
5. Create new source file;
6. Create your own `boolean.vhd` to the newly create .vhd file in Vivado;
7. Create a testbench file by following the previous steps, name it as '`boolean_tb.vhd`'

Observe Result

8. You can observe the simulation result in the Console window or the waveform window;



Check Results



Instruction

9. Check the result shown in the simulation results; compare it with the content in the Truth Table in step 3.

Anything goes wrong? Then you should learn how to debug your HDL codes.

If the results are the same, Congratulations!

Additional Xilinx Video

- Logic Simulator
- <https://www.xilinx.com/video/hardware/logic-simulation.html>
- Testbench Generator:
- <https://vhdl.lapinoo.net/testbench/>

Tutorial - Remarks

- Please raise your hand if you have questions!
 - 9 course tutors + course Instructors
 - If you finish the tutorial quickly, please try different settings in the Vivado tool.
- You can also refer to the course Canvas homepage.
- Get familiar with the Design and Simulation tool.
- Using BRAM and Distributed RAM Blocks will be covered in the next tutorial.
- Get ready for the upcoming Lab sessions.

