Lab # 8
Introduction to Xilinx

Introduction

This is our first hardware lab, we are going to use Xilinx program and Spartan-3E starter kit in our work.

Special thanks to Bahaa Eddin El Aila for providing us with his own kit.

Kit parts for this lab
We will use the three parts above in this lab and they are: USB connection table, power connection, knobs and LEDs.

Note that each knob or led has a name and pin name written between <> for example SW3<N17> this means that N17 it the FPGA pin connected to this switch.

Later we will use other parts in this kit.
Xilinx

The ISE Design Suite is the central electronic design automation (EDA) product family sold by Xilinx.

The ISE Design Suite features include design entry and synthesis supporting Verilog or VHDL, place-and-route (PAR), completed verification and debug, and creation of the bit files that are used to configure the chip.

Simple half adder implementation

We will use half adder as our example for this lab.

Step 1: Open ISE Design Suite and create a new project as follows
Then Finish

**Step 2 : Add new VHDL Module to write code in**
Here you can specify entity ports or you can write in manually as we used to do in Active-HDL program.
When you click finish and remove comments :D you will find a source similar to this:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity halfadder is
  Port ( a : in STD_LOGIC;
         b : in STD_LOGIC;
         sum : out STD_LOGIC;
         carry_out : out STD_LOGIC);
end halfadder;

architecture Behavioral of halfadder is
begin
end Behavioral;
```

**Step 3: Implement your system entity and architecture**

Use any architecture description you like to implement our half adder

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>SUM</th>
<th>Carry</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Sum = a xor b
Carry = a and b
Step 4: Start design process

After you finish your code, start a new process to implement your system:

Right click on **Synthesize – XST** and choose **run**, if no errors you should see a green sign.
Double click on **View RTL Schematic** and click **ok**, you should see this window:

Expand all files, mark them and click add then click Create Schematic.

The result:
With same steps, **view technology schematic**

**Step 5 : Simulating our module**

Add a VHDL test bench file
Remove all clock-related stuff and comments.

Look for this process and inset your simulations here

```
-- Stimulus process
stim_proc: process
begin
  -- hold reset state for 100 ns.
  wait for 100 ns;

  -- insert stimulus here
  wait;
end process;
```

```
-- Stimulus process
stim_proc: process
begin
  -- hold reset state for 100 ns.
  wait for 100 ns;

  a<= '0'; b<= '0'; wait for 10ns;
  a<= '0'; b<= '1'; wait for 10ns;
  a<= '1'; b<= '0'; wait for 10ns;
  a<= '1'; b<= '1'; wait for 10ns;

  wait;
end process;
```
After you do so, make sure that you are in the simulation area and you are selecting the test-bench source file as shown then click behavioral check syntax.

And the result:
Now you can run the simulation
Step 6: downloading the design into the kit

First connect the kit to your PC and make sure that your PC can recognize it.

Step 7: Pin-mapping
you should the **PlanAhead** program started,, expand folder below and notice the **site** column

the idea now is to connect the sites to pins printed on the board
Save and exit PlanAhead
Form ISE, run the complete process

Step 8: generate programming file
When it finishes: run “configure target device”

**ISE iMPACT** program will show up, double click on **boundary scan**
Make sure your kit is running then right-click on the program and choose **Initialize chain**

Right click to Add Device or Initialize JTAG chain

The result will be

---

**Identify Succeeded**
Note that the green device is our FPGA so we navigate to our project and choose the .bit file.

Then click NO, Bypass, Bypass, OK.
Step 9 : Program the FPGA

Results :

Program Succeeded

And on the kit : the Done led Goes High again
VHDL Lab starts now *^\`

*NOUR*