Exploring FPGA For Design Of Digital Hardware

Presentation on Mini Project

Sayak Haldar (111205038)
Shuvam Bosana (111205042)
Subham Banerjee (111205014)

Indian Institute Of Engineering,Science and Technology,Shibpur,
Howrah-03

April 21,2014

Presentation Outline

Introduction

Objectives:

- ⇒ Realizing different logic circuits using an FPGA with VHDL
- ⇒ Extending the realization to construct experiments for the Hardware Lab using FPGA

Advantages Of The Proposed Hardware Lab Over Existing Lab:

$\Rightarrow$ Easier to debug

- ⋆ A single FPGA board serves as a substitute to all tools and ICs used in an experiment
- ⋆ Relief from entangled wire
- ⋆ Logic probes to debug complex circuits will no longer be necessary
- ⋆ Takes less amount of time in debugging purpose

Advantages Of The Proposed Hardware Lab Over Existing Lab(contd.):

⇒ Flexible work schedule

- ⋆ Interface softwares are open source
- ⋆ Code can be developed at home and can be carried to the lab using pendrives
- ⋆ Code is machine independent therefore a "platform independent" environment is offered

Advantages Of The Proposed Hardware Lab Over Existing Lab(contd.):

$\Rightarrow$ Work can be saved from time to time

- $\star$ No need to concern over the time period of the lab
- $\star$ Previous day's work can be utilised

Advantages Of The Proposed Hardware Lab Over Existing Lab(contd.):

$\Rightarrow$ More number of circuits with various levels of complexity can be achieved

- $\star$ More number of circuits can be designed in a day
- $\star$ Problem solving approach will get better
- $\star$ Concepts will be more clear

What is an FPGA?

- ⇒ Field Programmable Gate Array
- ⇒ An integrated circuit with large no. of logic gates(ten thousands to several millions)
- ⇒ Initially unprogrammed. Users can program it with the help of Hardware Description Language at their workfield. Hence, "Field Programmable"
- ⇒ Our work is based on Xilinx Spartan-3 family XC3S200 FPGA.

Xilinx ISE

A software tool produced by Xilinx

- $\Rightarrow$ For synthesis and analysis of HDL designs
- $\Rightarrow$ For enabling the developer to synthesize their designs
- $\Rightarrow$ To perform timing analysis
- $\Rightarrow$ To examine RTL diagrams,
- $\Rightarrow$ To simulate a design's reaction to different stimuli
- $\Rightarrow$ To configure the target device with the programmer.

The version we used for our work is Xilinx ISE 10.1

First step: Creating an HDL source File:

We used VHDL for creating the HDL source. Now a VDHL code consists of several things like:-

- ⇒ Entity declarations
- ⇒ Architecture description
  - ⋆ Dataflow model
  - ⋆ Behavioral model
  - ⋆ Structural model
- ⇒ Timing Model/Delay Model
  - ⋆ Inertial delay(by default added)
  - ⋆ Transport delay
  - ⋆ Delta Delay

First step: Creating an HDL source File(contd.) :

An example of VHDL code(code for the full adder):

Entity declaration:

```
entity MYFA is
Port ( X : in STD_LOGIC;
       Y : in STD_LOGIC;
       C_IN : in STD_LOGIC;
       SUM : out STD_LOGIC;
       C_OUT : out STD_LOGIC);
end MYFA;
```

First step: Creating an HDL source File(contd.) :

An example of VHDL code(code for the full adder)(contd.):

Architecture Description:

architecture Dataflow of MYFA is
–Defining a dataflow model
begin
SUM<=transport(X xor Y xor C_IN) after 2ns;
–Now this is how transport delay is added using the keyword "transport"
–and mentioning the delay like after <time amount>
C_OUT<=transport((X and Y) or (Y and C_IN) or (C_IN and X)) after 3ns ;
end Dataflow;

Step 2:Synthesize the HDL source File:

- ⇒ To check if there is any syntactic error in the code
- ⇒ To check if the code is synthesizable or not(not all syntactic error free code are synthesizable)

Step 3: Adding an Implementation Constraints File:

⇒ Here pin assignments are done for the input and output pins

⇒ Comes with an extenstion of .ucf(User Constraints File)

Step 4:Generating .bit File:

For generating the .bit file the following steps should be done:-

- ⇒ Save the .ucf file
- ⇒ Select the vhdl file from the tab
- ⇒ Click on the Synthesize-XST option
- ⇒ Click on the Implement Design option
- ⇒ Click on the Generate Programming File option

Step 5:Download The .bit File To The FPGA Via JTAG:

To do so, the following steps should be done:-

- ⇒ Make sure that the FPGA device is correctly connected to the computer via JTAG(Joint Test Action Group) port
- ⇒ Click on the Configure Target Device Option
- ⇒ Download the .bit file to the FPGA via Impact software

Now, these 5 steps are need to be done for implementing a hardware design with an FPGA

Things We Provide For The Lab Course....

Now, here's a list of the things we are going to provide for the lab course:

- ⇒ Documentation about PLDs
- ⇒ Easy guide to realize circuits using FPGA with VHDL
- ⇒ Datasheets containing Experiment Details along with question modules
- ⇒ Solution set of the question modules
- ⇒ An introductory guide to VHDL

Future Advantages Of Learning FPGA Based Hardware Design:

Nowadays FPGA is used in many fields.

- ⇒ Aerospace and Defense
- ⇒ ASIC Prototyping Audio
- ⇒ Automotive
- ⇒ High Performance Computing
- ⇒ Industrial
- ⇒ Medical
- ⇒ Security
- ⇒ Video and Image Processing
- ⇒ Wired and Wireless Connection and many more

No Doubt,FPGA is the future

References:

- ⇒ Xilinx ISE 10.1 quick start tutorial
- ⇒ Xilinx ISE 10.1 in depth tutorial
- ⇒ VHDL primer by Jan Van der Spiegel. University of Pennsylvania
- ⇒ VHDL Programming by Example by Doughlas Perry

A Special Thanks To:

- ⇒ Professor Amit Kumar Das, Department of Computer Science and Technology,IIEST,Shibpur
- ⇒ Professor Ramanath Dutta, Department of Electronics & Communication Engineering,STCET,Khidirpur

Thank You!