University of Tehran
College of Engineering
School of Electrical & Computer Engineering

Experiment 4
Sessions 12

Digital Modulation completion

Digital Logic Laboratory
ECE 045
Laboratory Manual

Fall 1403

Contents

Contents 1

1 Introduction 2

2 FIFO 2

3 Writing and reading in FIFO 2

4 Observation 2

5 The Total design 4

Acknowledgment 5

Prepared and developed by Katayoon Basharkhah under supervision of Professor Z. Navabi

1 Introduction

This lab is designed to expand and improve upon some limitations found in Lab 3. The goals of this extension are as follows:

1. Sending various messages
2. Understanding the concept of FIFO (First In, First Out)
3. Capturing received data using an oscilloscope's trigger

In Lab 3, you sent one message using 5 switches on the FPGA board. Now, we want to use a FIFO to handle various messages, store them, and send them to the message unit from Lab 3. We need to write data to the FIFO and read it using 2 keys, ensuring both writing and reading are completed. Additionally, capturing data with an oscilloscope can be challenging, so you will learn how to use the oscilloscope to capture messages with defined trigger.

2 FIFO

You can use LPM (Library of Parameterized Modules) in Quartus to build a parameterized FIFO. Set its width to 5 bits and its depth to 8 bits. This setting means you have 8 data entries with a width of 5 bits each, which need to be written to and read from the FIFO. Its input comes from 5 switches, SW [4:0], and the output connects to the message processing unit for further processing. Instance FIFO IP according to the path below:

1. Click on *File* ▷ *New Project Wizard*.
2. Click on *File* ▷ *MegaWizard Plug-In Manager*.
3. Complete its setting

3 Writing and reading in FIFO

As mentioned, the FIFO has a *wrreq* input which receives data in 1 clock cycle when this pin is active. Similarly, the *rdreq* input performs the same task. To apply data, we use two pushbutton,**KEY1** for *rdreq* and **KEY2** for *wrreq*. It's important to note that we need just one clock cycle while the key is active, so using two *OnePulsers* is necessary.

Morse code **0101**

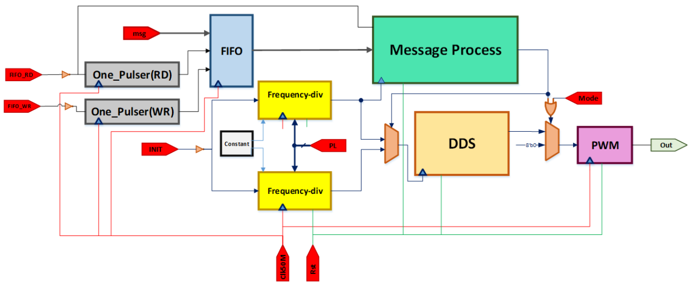
4 Observation

The oscilloscope captures data during modulation. To capture each message by pressing the *rdreq* of the FIFO, which is connected to the send pin of the message unit, you should define a new valid output pin in the message process. This signal will be active only in the modulation state and will come out from the FPGA's GPIO to be used for channel 2 of the oscilloscope. This pin helps in setting the trigger of the oscilloscope for capturing each message by pressing *rdreq*. Please ensure you set the following setup in your oscilloscope.

Prepared and developed by Katayoon Basharkhah under supervision of Professor Z. Navabi

1. Open Menu TRIGGER button
2. Set Source to CH2
3. Set Coupling to AC
4. Use Single button in each message

Figure 1: Block diagram of total design



5 The Total design

add FIFO, two Onepulser and their connection according to figure 1, you must integrate them into a pervious schematic design in LAB3.

Quartus II has this capability to generate symbols for any HDL code with the corresponding inputs and outputs.

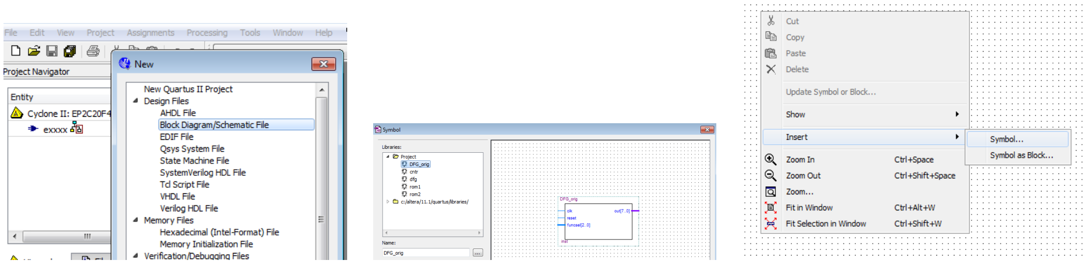
1. Click on *File* ▷ *New Project Wizard*.
2. Create a good directory for your project and complete the form as in the previous experiments.
3. In the *ADD/Remove Files in Project* page add all the Verilog code you have written.
4. By right-clicking on a Verilog code, you can create a symbol for it. If the symbol generation is done successfully, a file with the suffix `.bdf` will be generated.

You should use bus or line tool from the top toolbar menu ⌝ ⌝ ⌝ ╲ ╲ ╲ □ and make all the interconnects between components.

Double-click on the blank space in the Graphic Editor window, or click on the icon in the toolbar that looks like an `AND` gate. A pop-up box will appear.

Expand the hierarchy in the Libraries box. First, expand libraries, then expand the library primitives, followed by expanding the library logic comprising the logic gates. Use the gates if any is needed. When the schematic design is completed, compile it as before. Program the design on FPGA and record all the results.

Figure 2: Quartus II



Name: DFG_orig

Acknowledgment

This lab manual was prepared and developed by Katayoon Basharkhah, Ph.D. student of Digital Systems at the University of Tehran, under the supervision of Professor Zain Navabi.

This manual has been revised and edited by Zahra Jahanpeima, Zahra Mahdavi Ph.D. student of Digital Systems at the University of Tehran, Moeen Farhadi and Mohammad Yahyapour, and undergraduate students of Electrical Engineering at the University of Tehran.