Physics 623

FPGA I Construction of a Synchronous Counter

Aug. 4, 2008

1 The Goal of This Experiment

You will design a small digital circuit, download the design to a Field Programmable Gate Array (FPGA) IC, and verify that the circuit operates as predicted.

When using an FPGA, you build the circuit by creating circuit components with a hardware description language (VHDL) or use precompiled circuit components. The circuit component symbols are then entered into a schematic circuit diagram and wired together. Programs are then run that synthesize the circuit by creating a netlist and then performing translate, map, and place/route operations that describe the entry of the circuit into the FPGA. The final step is to run a program that creats a bitmap that can be loaded into the FPGA by use of a special loading program. This technique is valuable, in research or any enterprise, since you can quickly make moderately complex circuits and yet quickly make changes as the needs change.

2 Xilinx FPGA Introduction

A Field Programmable Gate Array, commonly known as an FPGA, is a general purpose programmable IC which is mainly used in the digital design environment that requires flexibility in design and fast turn around time.

In contrast to design using commercial ICs, FPGA can be configured to the designer's specific circuits in a single chip. In addition, in the case of design errors, FPGA can be redesigned and reconfigured easily in software, where as in the commercial IC design, it is not so easy to modify the already connected circuitry. In comparison to EPLD (Electrically Programmable Logic Device), FPGAS have more levels of registers and a more general routing topology. This renders the EPLD less useful in more general systems, especially in pipelined systems. Because of its size, ease of use, and reconfigurability, FPGA represents an attractive alternative in digital design. It is particularly suitable for experimental and rapid prototyping environments. Different types of FPGA from different vendors are available. The one we will be using is the Xilinx FPGA. Therefore, throughout this document, the term FPGA is referring to the Xilinx FPGA.

In general, there are two kinds of blocks in a Xilinx FPGA: Configurable Logic Blocks (CLB) and Input/Output Blocks (IOB). Each CLB is capable of supporting generic combinational logic functions and is composed of Look-Up tables, Carry and Control Logic and D-type flip-flop storage elements. Each FPGA contains a good number of CLBs arranged in a 2-dimensional square array to implement and interconnect the desired logic functions of the system. In the Spartan II XC2S50 containing 50,000 logic gates, there are 384 CLBs arranged in a 16 by 24 array.

IOBs are used to support the interface, including TTL voltage level, higher driving current, 3-state buffers, etc..., to the external environments. In addition, one flip-flop is also available in each IOB to provide latched input/output. Depending on the size of the chip, different size of FPGA contains different number of IOBs placed on the perimeter of the CLB array. In the case of XC2S50, there are 92 IOBs available for external connections.

Configurable routing wires are presented between CLBs and between CLBs and IOBs for interconnecting blocks. The routing wires are arranged in the form of a configurable matrix such that connections from any direction to any other direction are possible.

The Xilinx FPGA design process consists of the following procedures:

2.1 FPGA Design Flow

- Required circuit components such as counters and 7-segment LED Display decoders are built using a hardware description language (VHDL) or may already exist as prebuilt circuit components.
- The Schematic Capture tool is used for the entry of the circuit components into a circuit diagram.
- The Design Implementation tools perform the translation, map, place, route and bitstream generation phases of the design flow.
- The binary image of the circuit design is downloaded to the FPGA test board using a Download tool.

3 Experimental Procedures

This experiment is to acquaint you with the design process of Xilinx FPGA. You will enter the design of a simple circuit for the FPGA, download it to the FPGA, and verify that it works in hardware.

The tools for drawing the circuit schematic and creating circuit components with VHDL are integral to the Xilinx Integrated Software Environment (ISE 8.2) software which we run on a Windows platform.

The FPGA chip you will be using is the Spartan II XC2S50. It contains 384 CLBs arranged in a 16 by 24 array. In addition, there are 92 user I/Os available for external connection.

The first circuit with which we will implement is a four bit synchronous binary count-up counter with a 7-segment display as readout. A 100 MHz external clock is brought into the circuit and stepped down to about 3 Hz using the MSB of a 25 bit counter. The four output bits of the counter are decoded by a 7 Segment Display Decoder and the output seven bits are sent to the on-board 7 Segment Display. Check the logic of the circuit and understand how it works.

4 Equipment

The experiment uses the following equipment.

- A test board (XSA-50) which contains the XC2S50 Sparten II Xilinx FPGA, readout devices, and input/output cable connectors.
- A motherboard (XStend) for the test board that contains more readout devices, switches, analog I/O, a keboard and VGA interface, and a prototyping area in which additional circuit components can be added.
- A Windows XP PC with the Xilinx ISE 8.2 development software.

- Stabilized voltage supply of +9.0 Volts that plugs into the test board.

5 Design Process

It is assumed that you already know how to draw and edit a schematic using CAD schematic capture tools.

- Log on to the PC. The instructor will supply the login password if needed.
- Under the Xilinx program Group, start the Project Navigator. The Project Navigator controls all aspects of the design flow. You can access all of the design entry and design implementation tools as well as the files and documents associated with your project. Create a new project by selecting File → New Project Item.
- Under Project Location select a folder for your project files. Make a folder for yoursef in the Student directory. (C:\623Students\YourFolder) Set the project name in the Project Name field and select Schematic as the Top-Level Module.
- Choose the appropriate family, part, and speed grade. This operation will load the appropriate part libraries. Use the Spartan2 xc2s50 with package tq144 and speed grade -5. Keep clicking Next until Finish shows up and then click on it.
- Right click on the xc2s50-stq144 object and select New Source. Select Schematic.
- Click on the Symbols button to bring up the Parts library. Click on the desired part, move the mouse over to the schematic window and click again. The part will then be entered into the schematic window and will clone itself. Right mouse click to terminate the entry for that part. Enter all the desired parts.
- You will have to build the count-down counter and the 7 Segment Display Decoder in VHDL. At this point we assume you have worked through enough of the WebPack tutorial to make these components. After building the circuit component in VHDL make a circuit symbol by highlighting the component and selecting *Create Schematic Symbol* in the *Processes Window* When you return to the schematic diagram you will find a circuit symbol for both new parts.
- You can now wire up the complete circuit. Unlike **Electronics Workbench**, wires or buses must be added by using the **wire** tool.
- After the circuit is complete I/O markers must be added to designate inputs and outputs and later to assign them to real pin numbers. Click on the I/O markers button and select *add an output marker* or *add an input marker* in the Options tab. Then click on the free end of the wire segment to place the marker. Note the names assigned to the I/O pads. You will later have to assign them to real IC pin numbers. Save the schematic and return to the Project Navigator.
- In reality, all input/output signals are bonded to IC pins. Normally, It is the designer's job to designate the IC pin for each signal. In the Xilinx sotware this information is entered into a .ucf constraint file. Open a new source and select *Implementation Constraints File*.

The **PACE** window will open and pin numbers are entered with a **P** in front of the number. The selection of pins will depend on which board components are used or which pins are used for external I/O. Save the file and return to the Project navigator. In some versions of the ISE sofware you would first synthesize and map the circuit and then assign the pins by opening *Translate* and selecting *Assign Package Pins Post-Translate*. After the pin assignments are made in the **PACE** editor, rerun *Implement Design*.

- Now the circuit must be synthesized and mapped to the chip. Double click on *Synthesize*, let it finish, and then double click on *Implement Design* to execute Translate, Map, and Place and Route. The final step is to generate the bitstream.

- Right click on *Generate Programming File* choose *Properties* and then *StartupOptions*. Select the JTAG Clock if you are using the USB cable or the CCLK Clock if you are using the parallel port cable. Then double click on *Generate Programming File*. A .bit file with the project name will be generated and stored in the project directory.

6 Downloading the Circuit

The binary configuration bit file is now ready to be downloaded to the actual FPGA chip. Follow the following steps for downloading.

1. Make sure the circuit boards are connected to a +9V plug-in power supply.

2. Make sure the circuit board is connected to the appropriate PC port using either the parallel port or USB serial cable.

3. You will then use one of several available **GXSTOOLS** depending on which function you are trying to implement.

   - **GXSTEST**: This utility lets the user test an XS Board for proper functioning.
   - **GXSLOAD**: This utility lets the user download FPGA configuration files to the FPGA. Choose the appropriate download port (either the Parallel or USB0 port) depending on which download cable you are using.
   - **GXSPORT**: This utility lets the user send logic inouts to an XS Board by toggling the data pins of the parallel port.
   - **GXSSETCLK**: This utility allows you to set an integer divisor for the on-board 100 MHz clock.

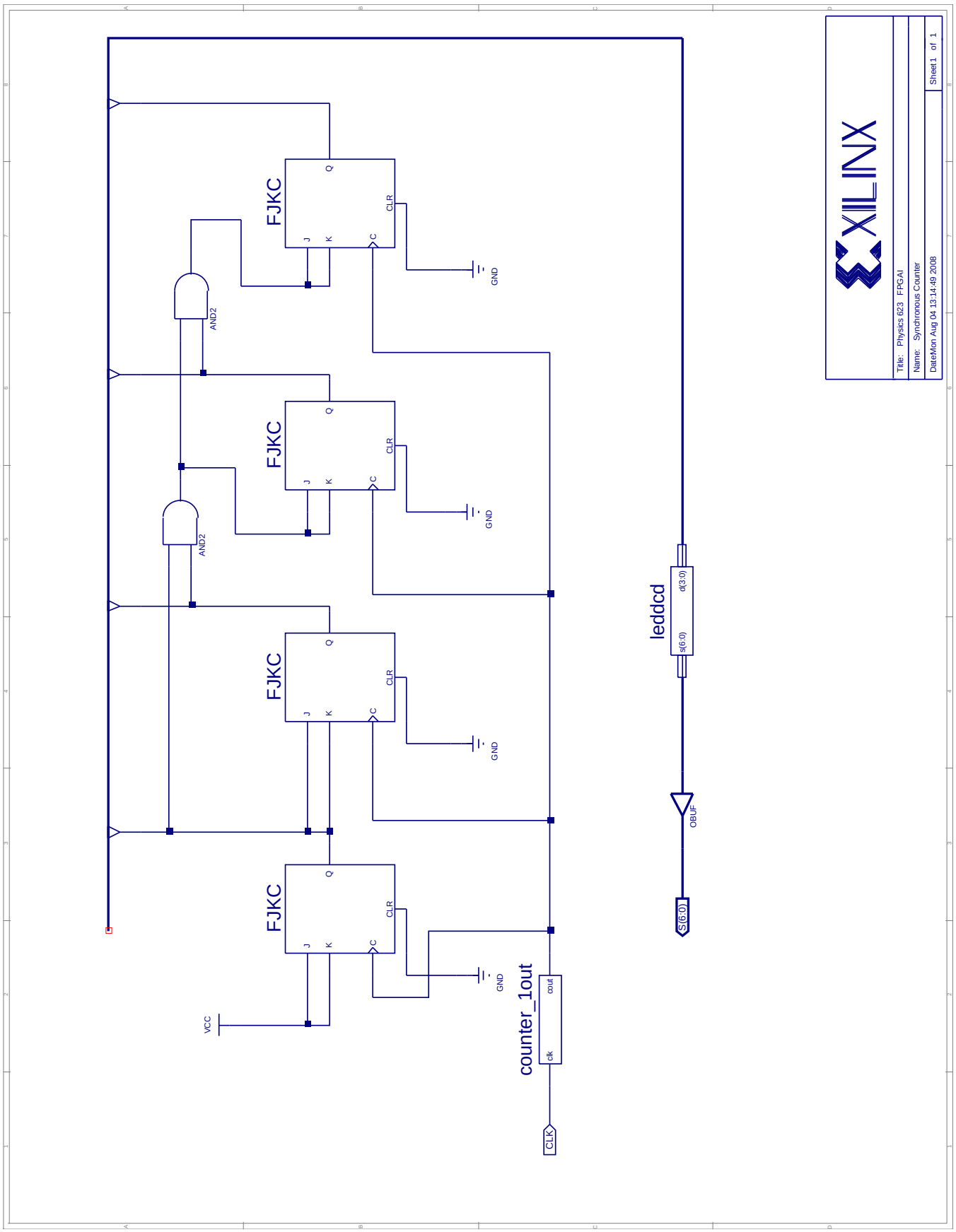
7 The Counter Circuit

7.1 Procedure

1. Create the design for the 4 bit synchronous counter and download the design into the Xilinx XC2S50 Spartan II chip. You can clock the counter with the the on-board external 100 MHz oscillator. The external oscillator will have to be stepped down to about 1 Hz so you can see the display increment. Use one of the seven segment displays on the circuit boards.

2. Record all observations in your notebook and include an explanation of how the circuit works.

7.2 Circuit Diagram



7.3 7-Segment LED Display Pin Assignments

LED Decoder Output	FPGA Pin	7-Segment LED
S0	P67	XSA Board
S1	P39	
S2	P62	
S3	P60	
S4	P46	
S5	P57	
S6	P49	
S0	P47	Rightmost XStend
S1	P40	
S2	P28	
S3	P29	
S4	P27	
S5	P42	
S6	P48	
S0	P64	Leftmost XStend
S1	P65	
S2	P76	
S3	P50	
S4	P51	
S5	P54	
S6	P56	
CLK	P88	XSA Board Clock