CMPE-661 Interface & Digital Electronics

Exercise 1

Introduction to Xilinx Vivado and SDK

By submitting this report, I attest that its contents are wholly my individual writing about this exercise and that they reflect the submitted code. I further acknowledge that permitted collaboration for this exercise consists only of discussions of concepts with course staff and fellow students. Other than code provided by the instructor for this exercise, all code was developed by me.

Jacob LaPietra

Performed January 23, 2023

Submitted February 3, 2023

Lecture Section 01

Instructor:

Lukowiak

Kurdziel

TA:

Payton Burak

Matthew Krebs

Abstract:

The main purpose of this exercise was to learn how to use the Xilinx Vivado and SDK tools. Two tutorials were completed along with one hands on exercise. The tutorials showed the steps involved in creating a Vivado block diagram instantiating a Zynq processing block along with all the necessary GPIO with their connections. The first tutorial stepped through a simple “hello world” program while the second tutorial showed how to use the internal Zynq timers. The hands-on exercise ruired a LFSR to be created and tested.

Design Methodology:

The first steps in Tutorial 1 showed how to navigate the Vivado tool and setup a basic project. The Zybo 20 board was selected in the creation of the project in order to match the board being used in class. No HDL was written in Vivado and instead a block diagram was created in order to instantiate and connect all the necessary block of the design that were necessary to run code on the Zynq platform. The Zynq processing block, two GPIO blocks and a mailbox block were added to the diagram and the automated connection tool was used to connect all of the blocks together.

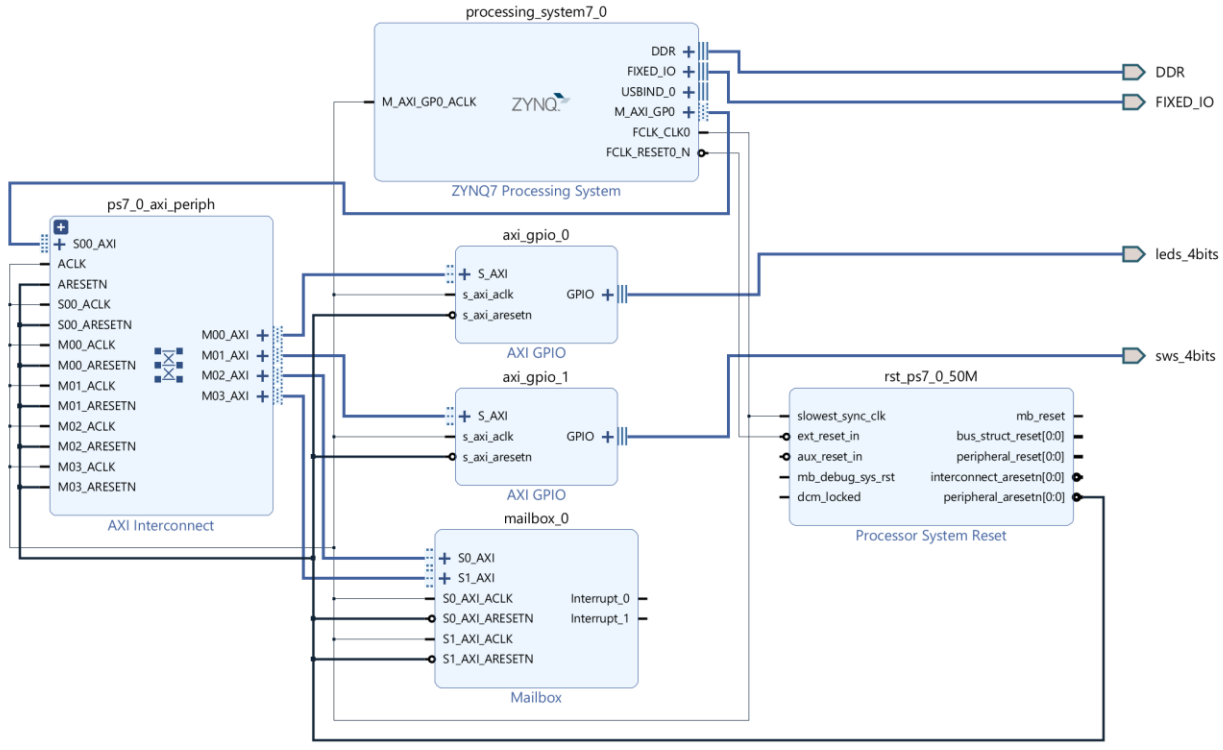

Figure 1: System Block Diagram

Figure 1 shows the block diagram that was used for all of the SDK platforms. The mailbox was not used in the first tutorial or hands on exercise but was used during the programs that used the private timers in Tutorial 2.

Once the block diagram was generated the bitstream was compiled for the FPGA. Once the bitstream compilation was finished, the Vivado SDK was opened in order to load code onto the board. The program that was created was a basic hello world program with a slight modification to the message to include the developer's name.

After the hello world program was created a program was then written that could take the input values from the switches on the board and display their current value onto the LEDs on the board.

The last part of Tutorial 1 consisted of creating two programs that ran on each of the CPU cores on the board. The two CPU cores utilized the mailbox and printed which CPU they were.

Tutorial 2 involved using the timers in the Zynq system. This was achieved by setting up the block diagram as stated before in Tutorial 1 then launching the Vivado SDK. In the SDK a program was provided to demonstrate the global timer functionality on the Zynq system. The code consisted of a loop that took timer counter values at periodic times after computing large computations to show the difference in counts between the different points in the loop.

Another program was provided to show the use of the private timer. The code provided achieved the same functionality as the code provided in the first part of the tutorial.

Lastly the Hands-On exercise was completed. The exercise consisted of implementing a Linear Feedback Shift Register that was described in the documentation. A program was written in order to determine the period of the register and to show all the cycles with the provided input.

Results:

The programs in the first tutorial printed information to the terminal over a UART port. For the first program, the output was expected to be a hello world message along with a location.

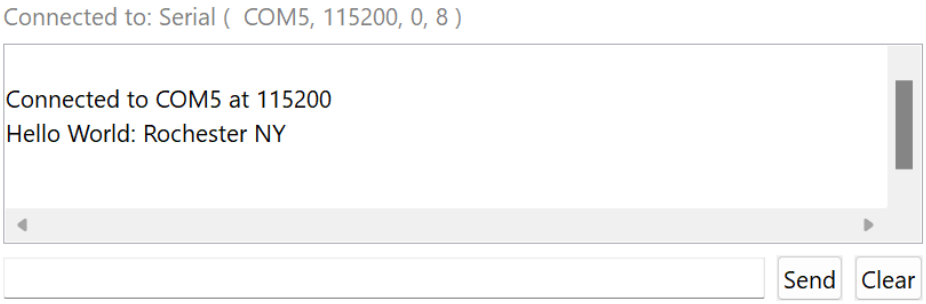

Figure 2: Tutorial 1 Part 1 Output

Figure 2 shows that the message was properly displayed to the terminal.

For part 2 of Tutorial 1 the expected out was a value showing the current state of the switches.

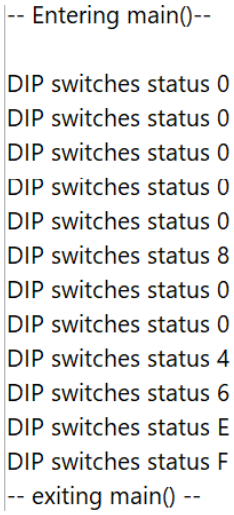

Figure 3: Tutorial 1 Part 2 Output

Figure 3 shows that the value in the printed value changed as the value of the switches changed.

For the last part of Tutorial 1 the expected output was a message showing the mailbox values each core received along with which core received them.

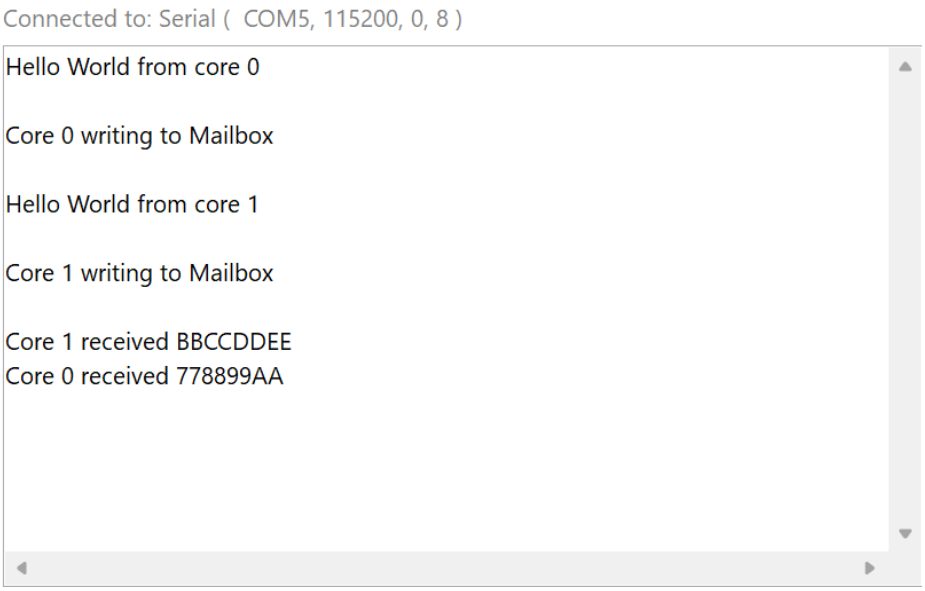

Figure 4: Tutorial 1 Part 3 Output

Figure 4 shows how the program produced the expected results.

For the first part of Tutorial 2 the expected output was different timer count values from the global timer that were taken at different parts of the loop.

```
--------------------------------------------------

Cycles loop 1 -      53736

DIP switches status F

Cycles loop 2 -    1671154
Cycles loop 3 -    2539202
--------------------------------------------------

Profiling
Start        - cycles1 = 0. 12
Start        - cycles2 = 0.391
Start        - cycles3 = 0.595
--------------------------------------------------
```

Figure 5: Tutorial 2 Part 1 Output

Figure 5 shows that the global timer works and properly shows how it can be used to determine the length of time it takes for given section of code to run on the processor.

Part two of Tutorial 2 has an expected output showing similar results to the first part, but instead used the private timer of the Zynq system.

```
--------------------------------------------------

Cycles loop 1 -      53753

DIP switches status D

Cycles loop 2 -    1636822
Cycles loop 3 -    2509052
--------------------------------------------------

Profiling
Start        - cycles1 = 0. 12
Start        - cycles2 = 0.389
Start        - cycles3 = 0.597
--------------------------------------------------
```

Figure 6: Tutorial 2 Part 2 Output

Figure 6 shows the expected output with timing values similar to Figure 5 which is correct since the same functional code was ran.

The Hands-On exercise had an expected output showing the LFSR values at each cycle in the period.

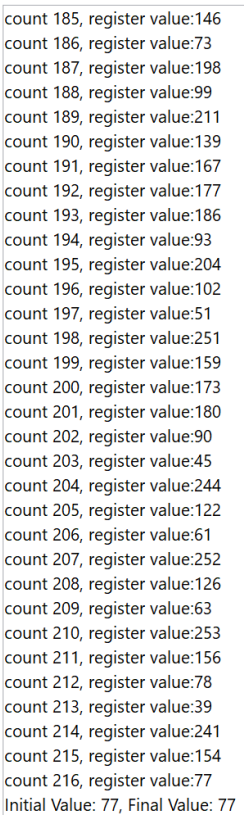

Figure 7: Hand-On Output

Figure 7 shows how the LFSR had a period of 216 and that the last value in the register matched the input value, which was expected.

Conclusion:

The exercise was successful since it demonstrated all parts of the tutorials and hands on assignments competing as expected. A process for creating and using a project in the Xilinx tools was shown how to do and will be used in further labs. An introduction to LFSR was also shown which will be an important tool that can be used in future crypto applications.

Hands-on 1: Introduction to Xilinx Vivado & SDK

Student's Name: Jacob LaPietra Section: 1

Demo		Point Value	Points Earned	Date
Part 1: Zynq HW & SW	Hello World	15	15	PB 1/25/2023
Part 2: Swiches/LEDs	Switches and LEDs Demo	15	15	PB 1/25/2023
Part 3: Zynq Dual-Core	Mailbox	15	15	PB 1/25/2023
Part 4: Zynq Timers	Global Timer	10	10	PB 2/1/2023
	Private Timer	10	10	PB 2/1/2023
Part 5: C Code	LFSR Implementation	15	15	PB 2/1/2023

To receive any grading credit students must earn points for both the demonstration and the report.

Report		Point Value	Points Earned	Comments
Discussion		4		
Part 1: Zynq Hw & SW	Terminal Output	2		
Part 2: Swiches/LEDs	Block Diagram	2		
	Terminal Output	2		
Part 3: Zynq Dual-Core	Block Diagram	2		
	Terminal Output	2		
Part 4: Zynq Timers	Terminal Output Global Timer	2		
	Terminal Output Private Timer	2		
	Questions	2		
Total for demo and report		100		