CS M152A Lab 1

Sequencer: An Example Project

In this lab, you will study an example project and do exercises based on the project.

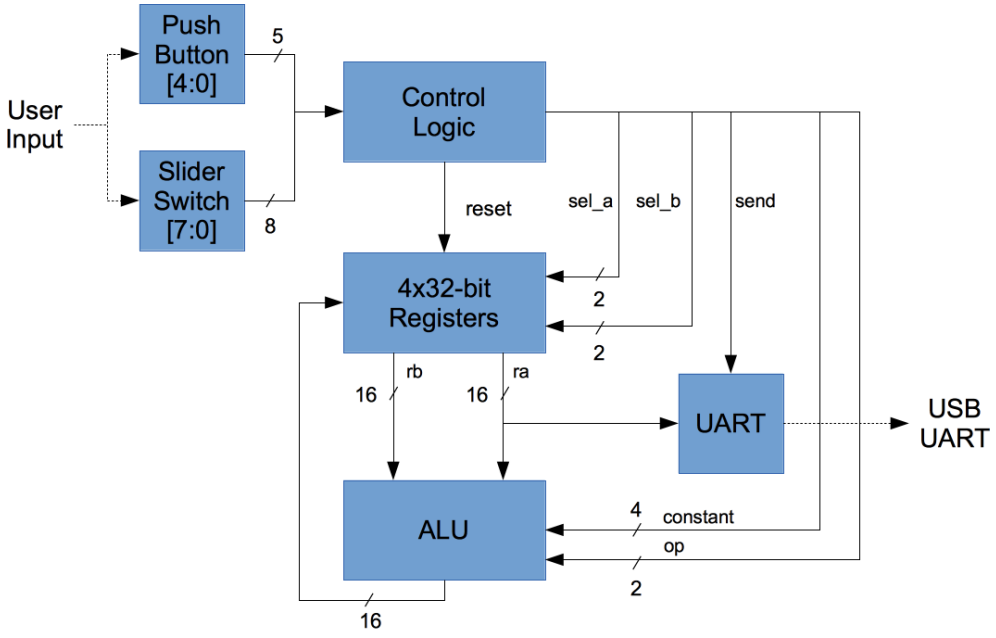
Introduction

In this lab, you are expected run an example FPGA project on FPGA and play with it. The lab consists of 2 parts. The first part of Lab 1 helps you get familiar with the FPGA design flow. You will get a general idea of how an FPGA design project is created and developed in Xilinx ISE. You will go through important steps in FPGA design flow including simulation, synthesis, place and route, and bit-stream generation. You will then interact with the programmed FPGA and finish a simple task based on it.

The second part of Lab 1 involves studying the Verilog module and test bench in the example project and answer related questions. The project is a good reference for future design projects, as you can learn the styles, formats, structures of code, etc. from the project. You can also learn useful techniques, like clock dividers and debouncers, from the project. There will be questions based on the code, and there will also be a few tasks that require modifying the original code.

Example Project Design

The example project is an adder/multiplier sequencer. It has four 16-bit general purpose registers. It can perform add or multiply instructions using registers as operands. The results are stored into register files.



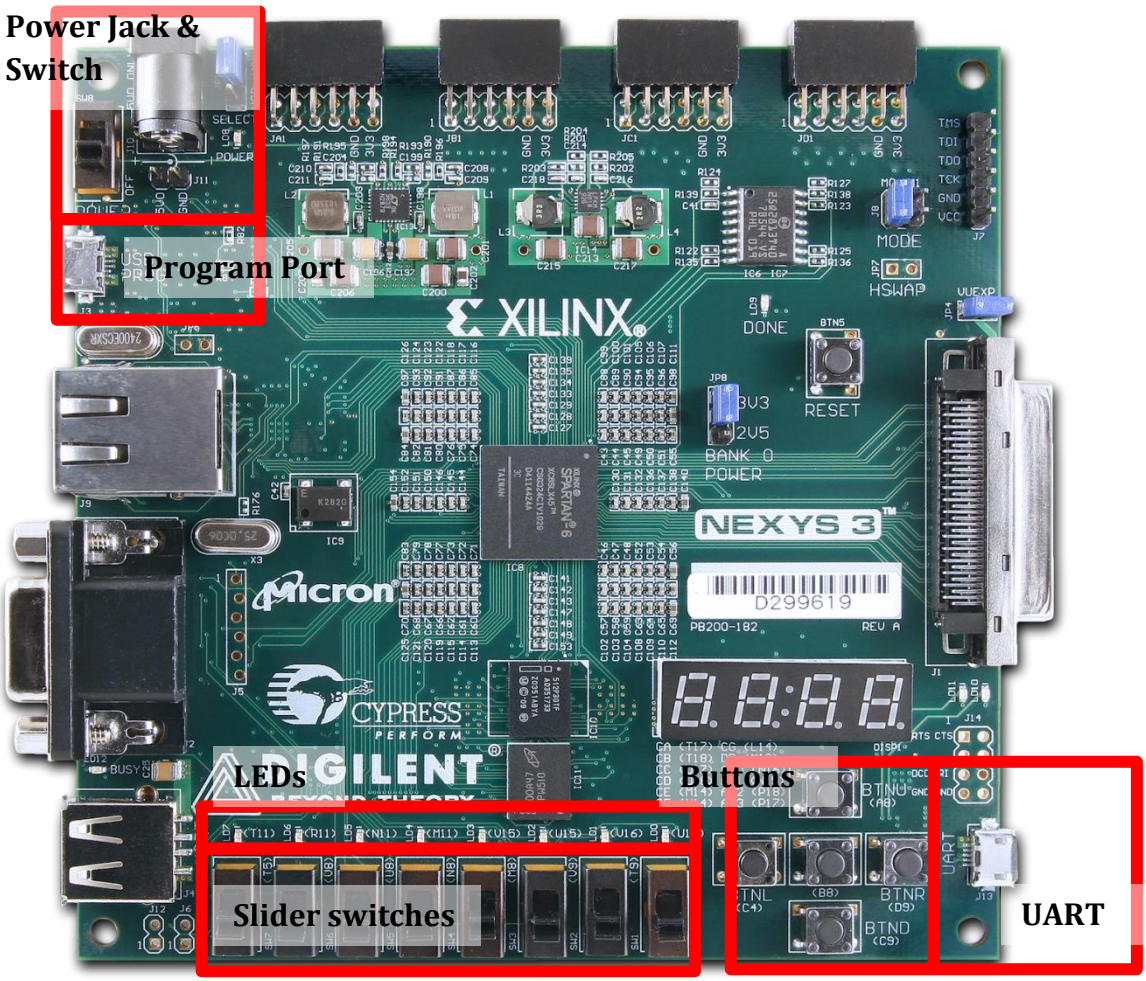
Project Diagram

Sequencer Operation

To operate the sequencer, the user enters instructions using the switches and push buttons:

- 8 switch positions represent a single 8-bit instruction (up: 1, down: 0)
- Push center button to enter and execute the instruction
- The push button BTNR (right button) resets values of all registers to 0

In addition, the 8 LED indicators shows the number of instructions executed since the last reset, in binary.

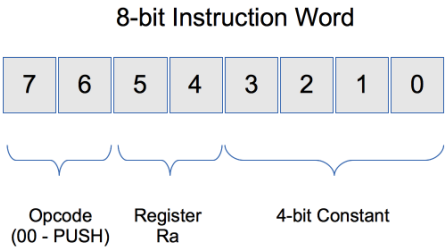


Nexys 3 Board

Sequencer Instructions

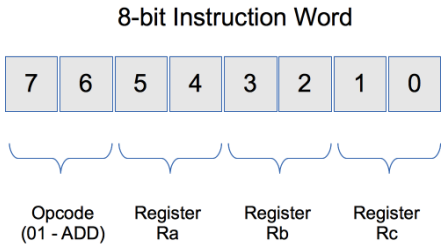
PUSH (00) instruction left-shifts the target register by 4 bits and "pushes" the new constant in. Example:

- R0 starts with value 0x55AA
- PUSH R0 0xB
- Now R0 is 0x5AAB



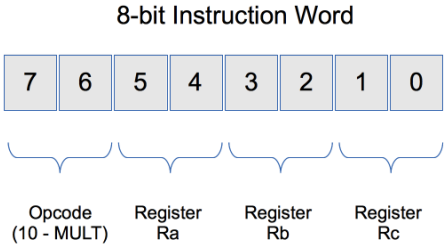
ADD (01) instruction adds Ra and Rb and stores the result to Rc. Addition is performed in unsigned integer arithmetic. Example:

- ADD R0 R1 R3
- R0 + R1 R3



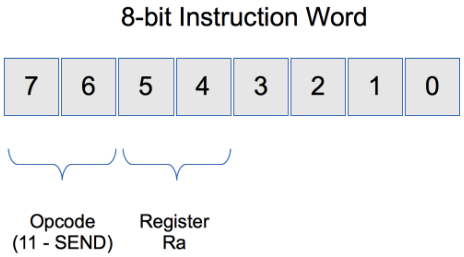
MULT (10) instruction multiplies Ra and Rb and stores the result to Rc. Inputs are assumed to be unsigned integers. Only the lower 16-bit results are retained. Example:

- MULT R3 R0 R2
- R3 * R0 R2



SEND (11) instruction sends the content of Rt to the UART for display. The 16-bit value is converted to ASCIIHEX and appended with a newline character. Example:

- R1 has a value of 0x1234
- SEND R1 causes the UART to send "1234\n"



Exercises

1. Read *FPGA Design and Implementation Fundamentals* and implement the project according to the guide attached in it. Please explain in your report how the steps in the FPGA design workflow correspond to the steps in the actual implementation using the Xilinx ISE and simulation tools.
   For example: The technology used for *Bitstream Generation* is the *BitGen* tool in Xilinx ISE. It generates a bit file contains all the information we need to configure the FPGA, and then downloads it to the FPGA board.
2. Translate the following "program" into binary instructions. Demo the UART console output to your TA after you finish.

- PUSH R0 0x4
- PUSH R0 0x0
- PUSH R1 0x3
- MULT R0 R1 R2
- ADD R2 R0 R3
- SEND R0
- SEND R1
- SEND R2
- SEND R3

3. Write a “program” (instruction) that outputs first 10 numbers of the Fibonacci series. Demo the simulation results to your TA after you finish.
4. Finish workshop 1. Include answers in the report.
5. Finish workshop 2. Include answers in the report.

Report

The report format described in the syllabus does not apply to Lab 1. The report for Lab 1 should include only the exercises from the section above.