ENGR 304
Hardware Lab 2: Basic Clocked Registers

Objectives

After completing this lab you should:

- recall binary arithmetic operations,
- recall how to use Quartus design software,
- be able to write VHDL code for registers and for combinational logic,
- be able to program the Altera part on the Altera DE2 board and test it out.

Altera DE2 Board

In this lab, you will be doing a simple logic design for implementation on the Altera DE2 board. During this lab, you will focus on inputting data from switches and displaying information on the 7-segment displays.

Specifications of the Design

In the design for this lab, you shall input two 8-bit unsigned numbers (not ASCII codes this time) that will then be added together to produce a 9-bit result. Two ("input") registers will be created that will store input values from the switches and one ("operation result") register will store the result of the addition of the two inputs. Two sets of eight slide switches will be used for inputting the numbers and three pushbuttons will be used as follows: one that operates as the system reset, one that clocks the switch values into the input registers, and one that clocks in the result of the add operation into the operation result register (or the adder output register). Seven segment displays shall show the bit patterns stored in the input registers and the bit pattern stored in the result register, all in hexadecimal format.

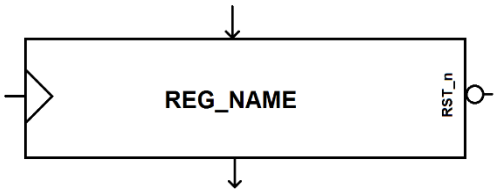
Specifically, the inputs and outputs of the system are defined as follows:

PIN Name	**Description**
SW(7 downto 0)	8-bit Input A with SW(7) as the most significant bit (unsigned format)
SW(15 downto 8)	8-bit Input B with SW(15) as the most significant bit (unsigned format)
KEY(3)	Input Clock. All <u>input</u> registers should operate on the rising edge of this signal (release of the button).
KEY(2)	Operation Clock. The <u>operation result</u> register should operate on the rising edge of this signal (release of the button).
KEY(1)	Unused, but may be easier to map the pins if KEY is considered a vector.
KEY(0)	System Reset. All registers should be reset to zero when this signal is asserted (Asserted = logic 0 or when the button is pressed).
HEXi(6 downto 0)	7 segment display signals for each of the 8 displays (HEX7 … HEX0). • HEX7&6 display the <u>registered</u> value from Input B in hex (RegB), • HEX5&4 display the <u>registered</u> value from Input A in hex (RegA), • HEX3 is always off (blank) • HEX2 is off unless there is a carry out from the 8-bit addition of RegB and RegA, and • HEX1&0 display the lower 8 bits of the RegB + RegA addition operation result. - HEX7-4 shall not change unless either KEY0 is pressed (resetting the inputs) or KEY3 is pressed and released (clocking in new values from the switches).

	- HEX2-0 shall not change unless either KEY0 is pressed (resetting the output register to 0) or KEY2 is pressed and released (clocking in a new adder result from the adder).

Steps:

1. Draw using paper and pencil a detailed block diagram of the system. You can use the block diagram template as a starting point. Add any blocks and label any internal signals that are needed. Below is a standard register symbol (multi-D-flip-flop) that can be used in your block diagram.



2. Estimate how many flip-flops should be needed in your final design, based on your block diagram. Be specific about how you counted up your flip-flops.
3. Create a new folder for this design and copy any important files needed for this design, renaming them as necessary (e.g. "HWLab2.vhd".)
4. Create a new project in Quartus in your new design directory. Again, specify the target device as the Cyclone II: EP2C35F672C6. Copy your HW01 files into the HW02 folder, then modify your previous VHDL file to implement the new requirements. You should build your VHDL code based on the blocks and signals that are in your block diagram. In other words, you should not add any VHDL code that does not have a corresponding signal or block in your block diagram. Screen-capture all Quartus compiler warnings and screen capture the compiler report showing the number of flip-flops in the compiled design.
5. Program the DE2 board with your design. Choose a small set of tests to run on your design. Identify why you chose each test and what the results were. Include this information in your writeup.

HAND IN:

1. Detailed and readable block diagram identifying the components of the system. Labels of signals and functional modules should match the labels found in your VHDL code.

2. Well-commented VHDL code (you do not have to print out the 7-segment code). You will lose points if the code is not printed according to the code printing standards introduces for assembly code labs (e.g. fixed width font, single spaced, few or no line-wraps).

3. Printout of a screen capture showing the number of registers in your design based on the compilation summary report from Quartus.

4. Printout of a screen capture showing all compilation warnings in Quartus.

5. A summary of the lab in a one-page writeup that:
   a. describes the main pieces of your block diagram,
   b. compares the number of actual and predicted registers, and
   c. includes a description of your DE2 testing plan and results.