Computer Systems Engineering Technology

CST 345 – HW/SW Co-Design

Lab 03 – Introduction to Picoblaze
Winter 2015
Instructor: Troy Scevers
Possible Points: 20

Name______________________________
Due Date: Monday, February 2nd @ 5 pm

Objectives

- To learn the steps required to construct a PicoBlaze microcontroller in the Xilinx ISE environment
- To write an assembly language program to implement various bit-level manipulations
- To use the PicoBlaze assembler to assemble your source code and test the results in hardware

Programming Assignment

Write a PicoBlaze assembly language program that will:

- Read the status of the eight slide switches on the Digilent board.
- Write the status of the eight slide switches to the LEDS on the Digilent board.
- Depending on the status of BTN2 (refer to the Diligent reference manual for button logic levels), the following actions should occur on the Digilent board:
  - If BTN2 is not being pressed, display the number of switches that are in the on position by turning on the left-most LEDs and working towards the right.
  - If BTN2 is being pressed, swap the order of the bits (again, consider the eight slide switches to represent the input bits). For example, the status of SW1 will be displayed on LD8; the status of SW2 will be displayed on LD7 etc.

Switch Data	LEDs (1 = on)
BTN2 = 0 (not pressed)	10010110
BTN2 = 1 (pressed)	01101001

- Repeat the previous steps continuously.

Procedure

1. Create a new ICE project for your picoblaze deisgn using the kcpsm6_design_template as a base. This will be very similar to what we did for the in class demo.
2. Bring in the kcpsm6.v or kcpsm6.vhd file to your project depending on if you are using verilog or VHDL.
3. Writing the assembly code:
   a) Create a new assembly file called "*simple.psm*". This is the file where you'll write your PicoBlaze6 assembly language source code. Create this files by using a text editor or the Xilinx environment and do your editing there (the Xilinx editor isn't too bad!). Save the file when you're done editing.
4. Assembling the assembly language source code:
   a) Start the kcpsm6.exe program located in the picoblaze 6 directory
   b) When asked for your file location enter the path to your simple.psm file. Make sure you have a copy of the Verilog or VHDL ROM template in the same directory as your .psm file.
5. Testing the assembly code in hardware:
   a) Add your compiled verilog or vhdl rom file to your ISE project
   b) Synthesize your design using ISE (make sure to set your pin assignments)
   c) Download your design onto the Digilent board like you've been doing all quarter.

Deliverables

1. Demonstrate your working circuit to the lab instructor.
2. Provide a listing of your assembly source code with your lab write up. Make sure you take a look at the PicoBlaze assembly language style file before you submit your source code.
3. Answers to the questions below.

Questions

1. The structure of your program could be characterized as an endless loop (code that repeated itself over and over again). The system clock on the Digilent board runs at 100MHz. Using this value and your knowledge of the PicoBlaze architecture, calculate the time (in seconds) required to complete each of the two iterations of the code contained in the endless loop. State your assumptions and show your work for this calculation.

2. The endless loop mentioned in the previous problem is typical of embedded system applications. Using only the instructions available in the PicoBlaze instruction set, would it be possible to stop PicoBlaze from executing instructions at some point in the code? Explain your answer.

3. What was the main function provided by the Verilog (or VHDL) "wrapper" code for the PicoBlaze implementation?

4. There are actually two forms of the PicoBlaze INPUT and OUTPUT instructions. Briefly discuss the differences and describe a situation where form that uses parenthesis would be useful.