University of Jordan
Computer Engineering Department
CPE439: Computer Design Lab

Experiment 9: Full Processor Simulation

First, it is required to construct a Verilog module for the entire PIC16F84A processor. This module should include the four modules that you have constructed in the previous experiments, namely: datapath module, program memory module, data memory module, and control module.

This module should have Verilog code similar to the following code:

```
module Processor(PortA, PortB, Clock, Reset);
  inout [4:0] PortA;
  inout [7:0] PortB;
  input Clock, Reset;
  // implementation details are left to the student
  …
endmodule
```

Then, it is required to test your design for the entire processor using the instruction sequence shown in the following table. This instruction sequence reads a number from Port A, outputs it to Port B, and keeps decrementing this number until it reaches zero. Finally, it outputs 8 ones to Port B.

Location	Instruction	Instruction Code
00	call 05	10 0000 0000 0101
01	compf portb,1 ;set Port B	
02	nop	
03	nop	
04	nop	
05	bsf status,5	
06	movlw B'00011111'	
07	movwf trisa ;all input	
08	clrf trisb ;all output	
09	bcf status,5	
0a	clrf portb ;clear Port B	
0b	movf porta,0 ;get number	
0c	movwf portb ;output number to Port B	
0d	decfsz portb	
0e	goto 0d	
0f	return	
10	nop	

To carry out this test you need to perform the following steps:

1) Instantiate your processor module in a test module.
2) The test module should provide to the processor initial reset signal and square wave clock of period suitable for your design.
3) The control memory must be initialized with all the needed control signals as specified in Experiment 8.
4) Complete the instruction codes in the table above and use these codes to initialize the program memory.

5) Port A should be processor input port and Port B should be processor output port. Your test module should set Port A to value B’00011’ and monitor Port B.
6) You need to calculate the number of cycles needed to execute the above code sequence and stop your simulation as soon as the processor finishes executing the complement instruction.

Report

Your report should include detailed design and Verilog code for all modules, critical path analysis, and timing diagrams that demonstrate the correct operation of your design.