University of Jordan

Computer Engineering Department

CPE439: Computer Design Lab

Experiment 7: Data Memory Module – Part II

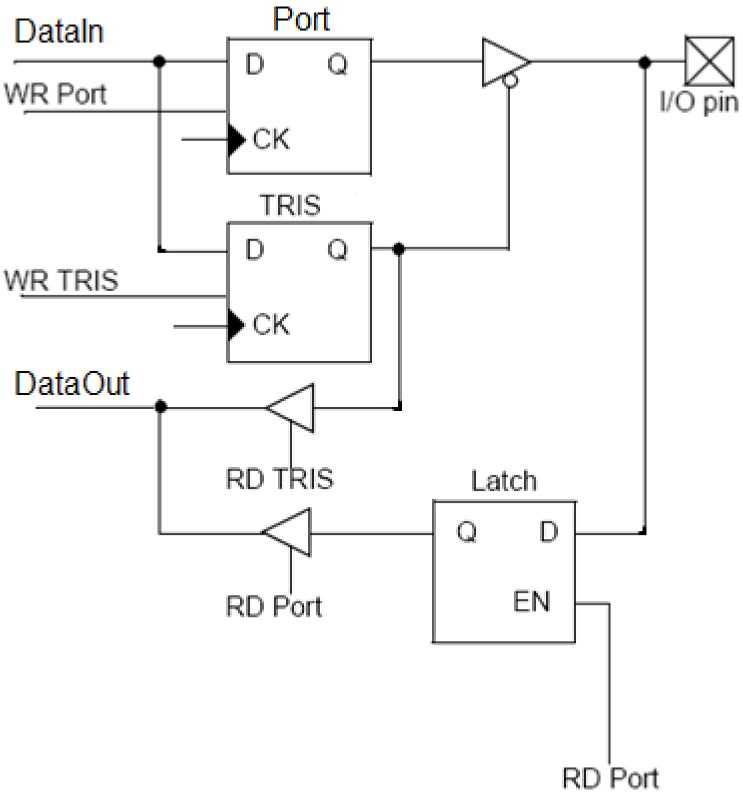
It is required to complete constructing and testing the Verilog module for a data memory suitable for incorporation in your PIC16F84A design. You need to complete the circuits of the special function registers: STATUS, PORTA, PORTB, TRISA, and TRISB.

Status Register

The Status register must be implemented using individual flip-flops so that each bit of the three status bits (C, DC, and Z) could be updated individually. This is in addition to the ability to read and write this register as an 8-bit register similar to the other data memory locations.

Parallel Input/Output

The following circuit shows how to build a configurable input/output port. Duplicate this circuit (with other needed logic) to implement PIC's Port A and Port B. You need to duplicate this circuit 5 times for implementing PORTA and TRISA, and 8 times to implement PORTB and TRISB.



Report

Your report should include detailed design, Verilog code for all modules including your test modules, and timing diagram that demonstrates the correct operation of your design.

To demonstrate the correct operation of you design, test your design using the input signals shown in the following two tables. You need to perform two test experiments.

1) Testing Ports A and B

Clock	Address	Data In	Data write	Reset
0 to 1 to 0	0	0	0	1
0 to 1 to 0	6	24	1	0
0 to 1 to 0	3	FF	1	0
0 to 1 to 0	6	F	1	0
0 to 1 to 0	5	F	1	0
0 to 1 to 0	3	00	1	0
0 to 1 to 0	6	00	0	0
0 to 1 to 0	4	00	1	0
0 to 1 to 0	0	00	1	0

You need to initialize the wires to Port A and B as demonstrated in the following code:

```
module PORTStest;
     wire [7:0]  PortB;
     wire [4:0] PortA;
     reg [7:0] RegB;
     reg [4:0] RegA;
…
     DataMemory m(PortA, PortB, …);
     assign PortA = RegA;
      assign PortB = RegB;
      initial begin
          RegA=8'hzz;
          RegB=5'bzzzzz;
     end
…
//Details are left to students as table 1 shows
endmodule
```

2) Testing the status register

Clock	Addr	Data In	Data write	C_en	C_in	Z_en	Z_in	DC_en	DC_in	Reset
0 to 1 to 0	3	0	0	0	0	0	0	0	0	1
0 to 1 to 0	3	0	1	0	0	0	0	0	0	0
0 to 1 to 0	3	0	0	1	1	0	0	0	0	0
0 to 1 to 0	3	0	0	0	0	1	1	0	0	0
0 to 1 to 0	3	0	0	0	0	0	0	1	1	0