Problem Set #8  Fall 2003

1.b) Module: RDI-INST (CLK, RESET, RDI, MUX1, MUX2, INC-PC, LOAD-ADDR, LOAD-RDI)

Input  
- CLK;
- RESET;
- RDI;

Output  
- MUX1[1:0];
- MUX2[1:0];
- INC-PC;
- LOAD-ADDR;
- LOAD-RDI;

Reg  
- [3:0] cs, ns;

Parameter  
- state0 = 2'b00;
- state1 = 2'b10;
- state2 = 2'b11;

always @ (posedge CLK) begin
  if (RESET) cs <= 0; else cs <= ns;
end

always @ (posedge CLK) begin
  case(cs)
    state0 : begin
      if (RDI) ns = state1; else ns = state0;
      end
    state1 : begin
      INC-PC = 1'b1;
      LOAD-ADDR = 1'b1;
      LOAD-RDI = 1'b0;
      MUX1 = 2'b00; // no reg sel[3] in MUX1
      MUX2 = 2'b00; // no reg sel[1] in MUX2
      ns = state2;
    end
    state2 : begin
      INC-PC = 1'b0;
      LOAD-ADDR = 1'b0;
      LOAD-RDI = 1'b1;
      MUX1 = 2'b01; // 01 regsel + memout to MUX1
      MUX2 = 2'b00; // ns restart;
    end
  endcase
end

1.c) Block schematic for PC

new fetch cycle

PC <= ADD-R, PC + 1 <= PC

LOAD-ADDR, INC-PC

MUX = MEM[OUT], LOAD-RDI
3. We need to ensure that all unused states get sent back to a defined state. Consider a next state decoder:

Case (present_state):

10'b0000000000:

10'b1000000000:

default: next_state: 10'b0000000001;

The default logic needs to look for any combination with more than 1 bit on or all bits zero. Xilinx likes functions of 4, 5, or 6 inputs, which will fit conveniently inside a CLB LUT. One hot encoding can lead to using a lot of LUTs. Also, adding one more state can add a lot more look-up table logic. While and gates can use a lot of LUT logic, leading to reduced operation speed.