**// This is a draft Verilog specification for register-to-register**

**// transfer with four 16-bit registers.**

**// Note: This code has not been tested for correctness.**

**// In the top-level module, “my\_reg” has been instantiated four times,**

**// and “my\_decoder” two times.**

**module my\_reg (ld, en, d\_in, d\_out, clk)**

**input ld, en, clk;**

**input [15:0] d\_in;**

**output [15:0] d\_out;**

**reg [15:0] d\_out, R;**

**always @(posedge clk)**

**begin**

**if (ld)**

**R <= d\_in;**

**else if (en)**

**d\_out <= R;**

**else**

**d\_out <= 16’bz;**

**end**

**endmodule**

**module my\_decoder (inp, outp) // A 2-to-4 decoder**

**input [1:0] inp;**

**output [3:0] outp;**

**always (inp)**

**case (inp)**

**2’b00: outp = 4’b0001;**

**2’b01: outp = 4’b0010;**

**2’b10: outp = 4’b0100;**

**2’b11: outp = 4’b1000;**

**endcase**

**endmodule**

**module top\_level (src, dest, MOVE, IN, clk )**

**input [1:0] src, dest;**

**input MOVE, IN;**

**wire [3:0] enable, load;**

**wire [15:0] data\_in, data\_out;**

**assign data\_in = data\_out;**

**my\_decoder D1 (enable, src);**

**my\_decoder D2 (load, dest);**

**my\_reg R0 (load[0], enable[0], data\_in, data\_out, clk);**

**my\_reg R1 (load[1], enable[1], data\_in, data\_out, clk);**

**my\_reg R2 (load[2], enable[2], data\_in, data\_out, clk);**

**my\_reg R3 (load[3], enable[3], data\_in, data\_out, clk);**

**endmodule**