module register8bit(

/\* Inputs \*/

input clk,

input reset,

input carry\_in,

input [7:0] data\_in,

/\* Outputs \*/

output reg [7:0] data\_out,

output reg carry\_out,

output reg over\_flow

);

/\* Initialization \*/

initial begin

#10 data\_out = 0;

#10 carry\_out = 0;

end

/\* Activate on rising clock edge or reset \*/

always @ (posedge clk | reset) begin

if (reset) begin // Reset outputs

data\_out = 8'b00000000;

carry\_out = 0;

end

else begin // Set outputs

data\_out = (data\_in \* 2) + carry\_in;

carry\_out = ~& data\_in[7|6];

end

end

/\* Handle Overflow \*/

assign over\_flow = data\_out[7];

endmodule