Review Problem 48

- Implement the controlled register from lecture in Verilog.

<table>
<thead>
<tr>
<th>Reset</th>
<th>Load</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Q = old Q</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Q = 0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Q = D</td>
</tr>
</tbody>
</table>

```verilog
module reg4(out, reset, load, d, clk);
    output logic [3:0] out;
    input logic reset, load, clk;
    input logic [3:0] d;
    always @ (posedge clk or negedge reset)
        begin
            if(reset) begin
                out <= 4'b0000;
                end
            else if(load) begin
                out <= d;
                end
            else begin
                out <= out; // mandatory
            end
            end
        always @ (posedge clk) begin
            out <= ns;
        endmodule
```

Binary UP-Counter Imp.
Complex Binary Counter
Design a 3-bit count that goes through the sequence

000 → 010 → 100 → 101 → 110 → 111 → 001 → 011 → 000 → ...

Arbitrary Sequence Counters
module upcounter #(parameter WIDTH=8) (out, incr, reset, clk);

input logic [WIDTH-1:0] out;
input logic [WIDTH-1:0] incr;
input logic [WIDTH-1:0] reset;
input clk;

if (reset) out = {WIDTH-1:0} 0;
if (incr) out = {WIDTH-1:0} out + 1;
else if (out = {WIDTH-1:0} WIDTH-1) out = {WIDTH-1:0} 0;
end

Counter in Verilog
ROM: Read-only Memory
RAM: Random Access Memory, Read/Write

Memory

Computer programs, data, pictures, etc.
Need method for storing large amounts of data