Problem 1

Design a priority encoder that implements the following truth table.

Inputs				Outputs		
a3	a2	a1	a0	y1	y0	Valid
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

```
module encoder (output logic [1:0] y, output logic valid,
                input logic [3:0] a);
```

Write a testbench that tests the module with all 16 possible input combinations. In addition, synthesize your design in Quartus. Your solution should include your code (both the module and testbench), the simulation waveform, and the RTL netlist from Quartus.

Problem 2

Design an 8-bit shift register with an asynchronous active low reset and the functionality as described in the following table.

s	**Operation**
00	Parallel load from **a** input
01	Shift right (use **shiftIn** bit as bit n-1)
10	Shift left (use **shiftIn** bit as bit 0)
11	Hold

```
module shiftReg (output logic [7:0] q,
                 input logic [7:0] a, input logic [1:0] s
                 input logic shiftIn, clk, reset_n);
```

Write a testbench that tests the module in all operating modes. In addition, synthesize your design in Quartus. Your solution should include your code (both the module and testbench), the simulation waveform, and the RTL netlist from Quartus.