VE270 Homework 5

Liu Yihao 515370910207

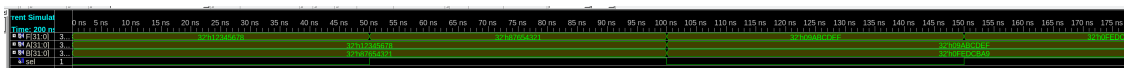
Problem 1.

Verilog Code:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    18:38:05 06/15/2017
// Design Name:
// Module Name:    Q1
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module Q1(F, A, B, sel);
    input [31:0] A, B;
    input sel;
    output [31:0] F;
    reg [31:0] F;

    always @(A, B, sel)
    begin
        if (sel == 0) F = A;
        else F = B;
    end
endmodule
```

Simulation Result:



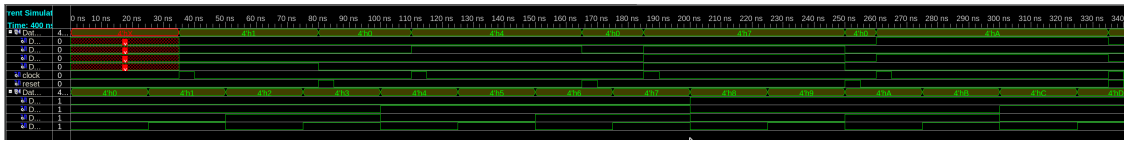
Problem 2.

Verilog Code:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    22:52:42 06/15/2017
// Design Name:
// Module Name:    Q2
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module Q2(Data_out, Data_in, clock, reset);
    input [3:0] Data_in;
    input clock, reset;
    output [3:0] Data_out;
    d_flip_flop d0(Data_out[0], Data_in[0], clock, reset);
    d_flip_flop d1(Data_out[1], Data_in[1], clock, reset);
    d_flip_flop d2(Data_out[2], Data_in[2], clock, reset);
    d_flip_flop d3(Data_out[3], Data_in[3], clock, reset);
endmodule

module d_flip_flop(Data_out, Data_in, clock, reset);
    input Data_in, clock, reset;
    output Data_out;
    reg Data_out;
    always @(posedge clock or posedge reset)
    begin
        if (reset == 1) Data_out = 0;
        else Data_out = Data_in;
    end
endmodule
```

Simulation Result:



Problem 3.

Verilog Code:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    16:36:27 06/16/2017
// Design Name:
// Module Name:    Q3
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module Q3(x, clock);
    parameter f0 = 1;
    input clock;
    output [3:0] x;
    reg [3:0] x = 4'h0;
    reg f = f0;
    always @(posedge clock)
    begin
        x[0] <= x[1];
        x[1] <= x[2];
        x[2] <= x[3];
        x[3] <= f;
        f <= x[3] ^ x[0];
    end
endmodule
```

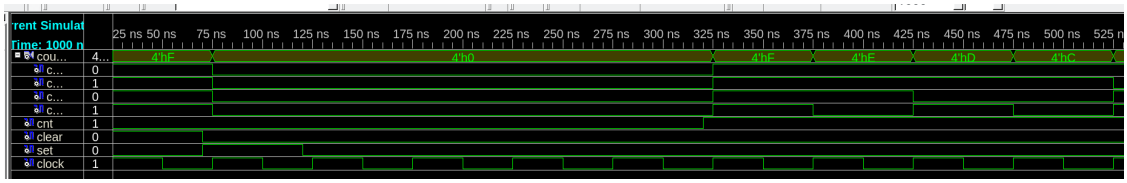
Simulation Result:

Problem 4.

Verilog Code:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    23:30:51 06/16/2017
// Design Name:
// Module Name:    Q4
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module Q4(counter, cnt, clear, set, clock);
    input cnt, clear, set, clock;
    output [3:0] counter;
    reg [3:0] counter = 4'b1111;
    always @(posedge clock)
    begin
        if (clear == 1) counter = 4'b1111;
        else if (set == 1) counter = 4'b0000;
        else if (cnt == 1) counter = counter - 1;
    end
endmodule
```

Simulation Result:



Problem 5.

Verilog Code:

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    21:52:47 06/21/2017
// Design Name:
// Module Name:    Q5
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////
module Q5(counter, upper, clock);
    output [3:0] counter;
    output upper;
    input clock;
    reg [3:0] counter = 4'h0;
    reg upper = 0;
    always @(posedge clock)
    begin
        counter = counter + 1;
        if (counter[3]) upper = 1;
        else upper = 0;
    end
endmodule
```

Simulation Result: