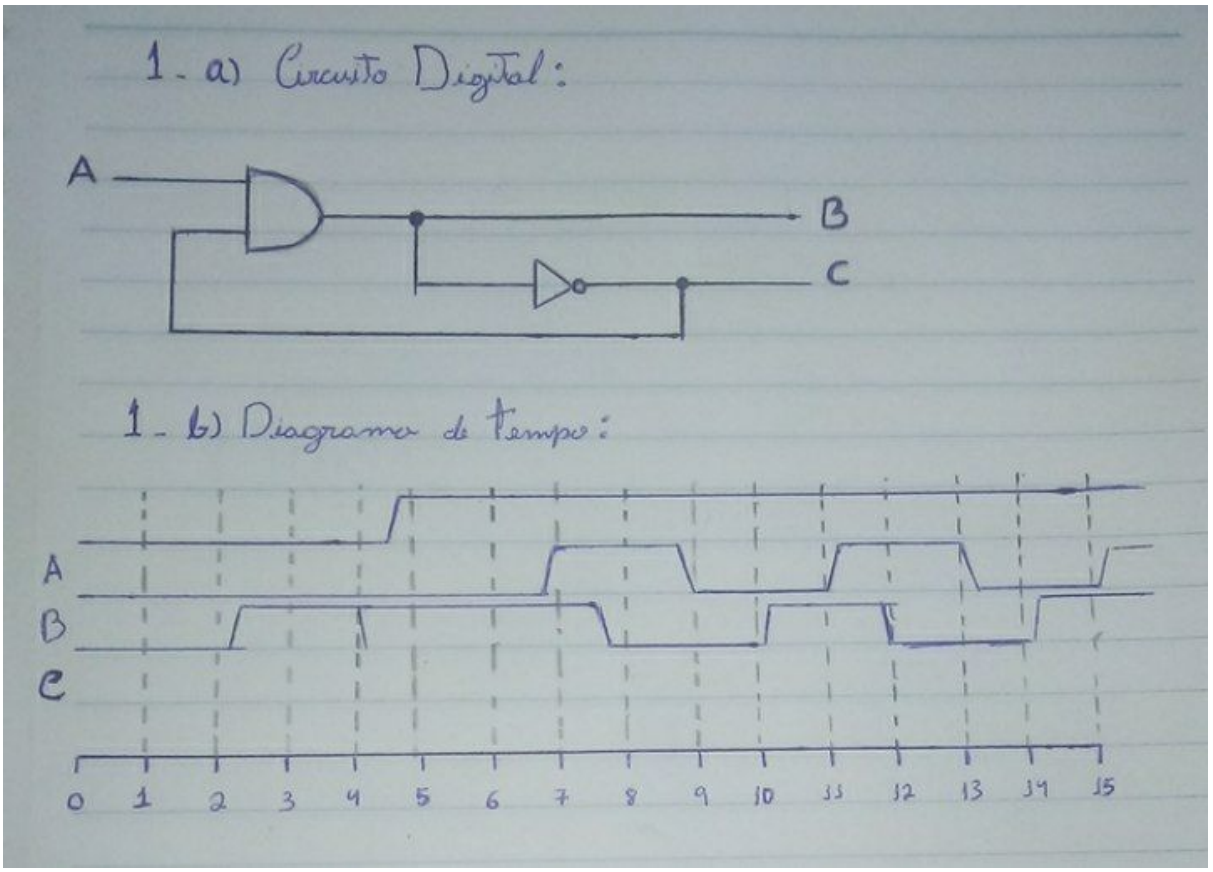
Projeto de Sistemas Digitais - P1: Primeiros passos em Verilog

Aluno: Vítor Aguiar Tavares (vat2)

Questão 1:



Questão 2:

```
module questao_2(input A, input B, input C, input D, output SAIDA);

   assign #5 AND = (A && B && C);
   assign #5 NOR = !(B || C);
   assign #5 OR = (AND || D);
   assign #5 NAND = !(A && NOR);
   assign #2 NOT = !NAND;
   assign #5 SAIDA = (OR ^ NOT);

endmodule
```

Questão 3:

a)

```
module mux4to1_assign(input [3:0] I0, input [3:0] I1, input [3:0] I2,
input [3:0] I3, input [1:0] CD, output [3:0] F);

   assign #10 F = CD[1] ? (CD[0] ? I3 : I2) : (CD[0] ? I1 : I0);

endmodule
```

b)

```
module mux4to1_ifelse(input [3:0] I0, input [3:0] I1, input [3:0] I2,
input [3:0] I3, input [1:0] CD, output reg [3:0] F);

   always @(I0 or I1 or I2 or I3 or CD)
   begin
       if(CD == 2'b00)
       begin
           #10 F = I0;
       end
       else if(CD == 2'b01)
       begin
           #10 F = I1;
       end
       else if(CD == 2'b10)
       begin
           #10 F = I2;
       end
       else
       begin
           #10 F = I3;
       end
   end

endmodule
```

c)

```
module mux4to1_case(input [3:0] I0, input [3:0] I1, input [3:0] I2,
input [3:0] I3, input [1:0] CD, output reg [3:0] F);

   always @(I0 or I1 or I2 or I3 or CD)
   begin
       case(CD)
           2'b00 : #10 F <= I0;
           2'b01 : #10 F <= I1;
           2'b10 : #10 F <= I2;
           2'b11 : #10 F <= I3;
       endcase
   end

endmodule
```

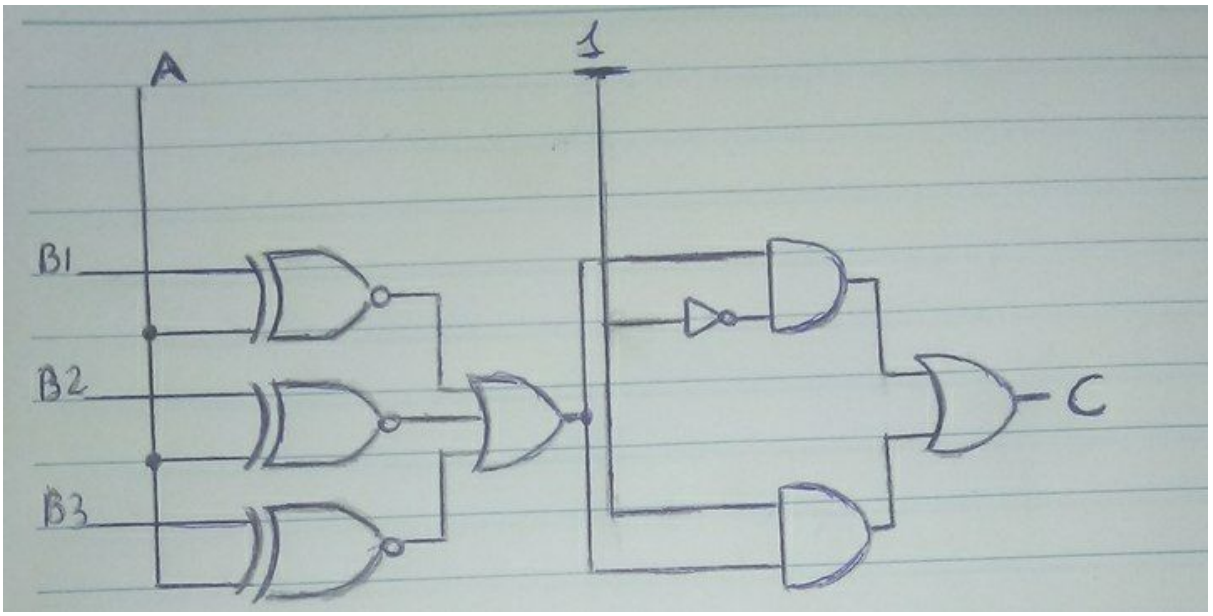
Questão 4:

a)

```
module questao_4_ifelse(input A, B1, B2, B3, output reg [1:0] C);

   always @(A, B1, B2, B3, C)
   begin
       if(A == B1)
       begin
           C = 2'b01;
       end
       else if(A == B2)
       begin
           C = 2'b10;
       end
       else if(A == B3)
       begin
           C = 2'b11;
       end
       else
       begin
           C = 2'b00;
       end
   end

endmodule
```

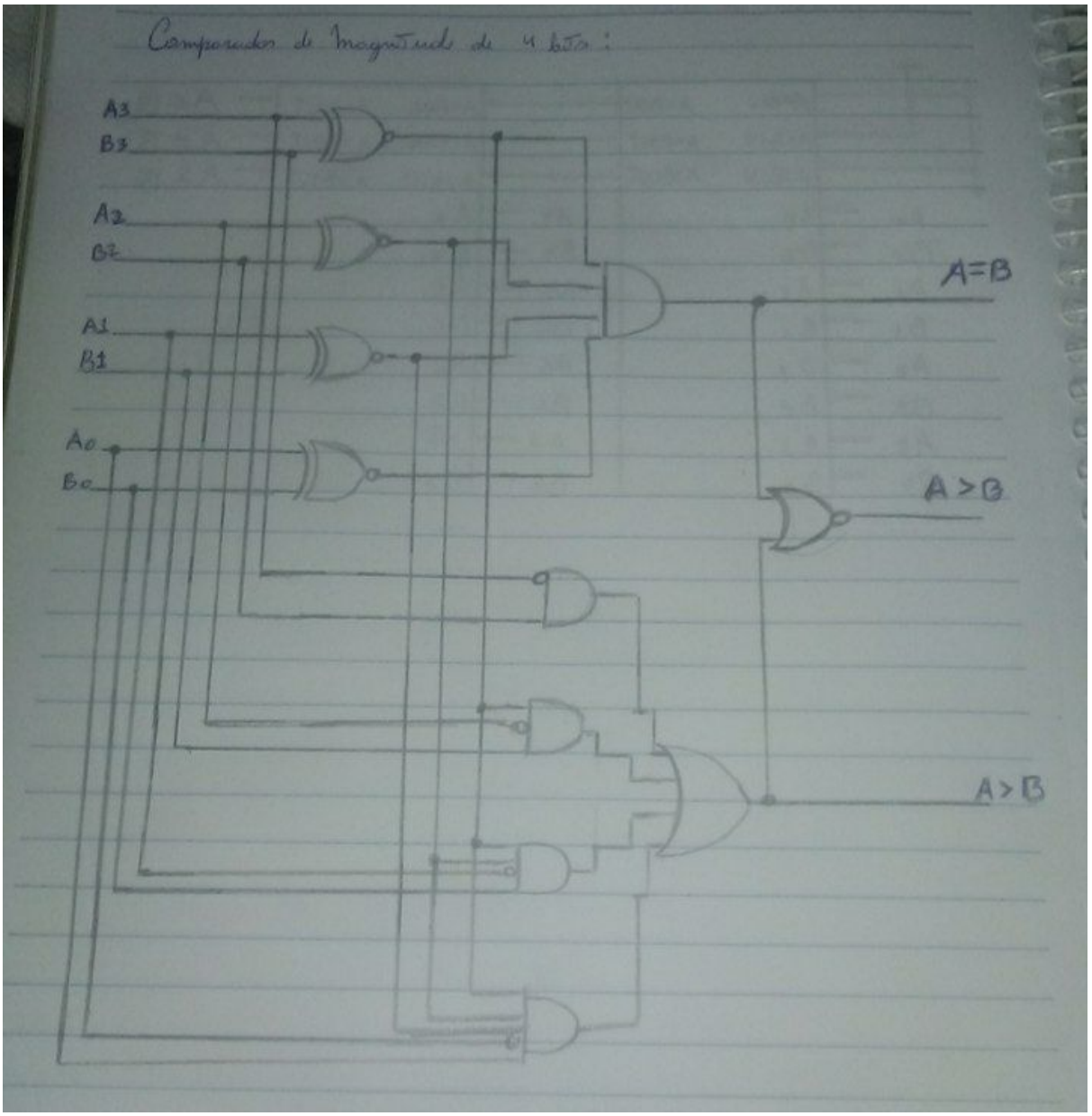
```
module questao_4_case(input A, B1, B2, B3, output reg [1:0] C);

   always @(A, B1, B2, B3, C)
   begin
       case(A)
           B1 : C = 2'b01;
           B2 : C = 2'b10;
           B3 : C = 2'b11;
           default : C = 2'b00;
       endcase
   end

endmodule
```

b)

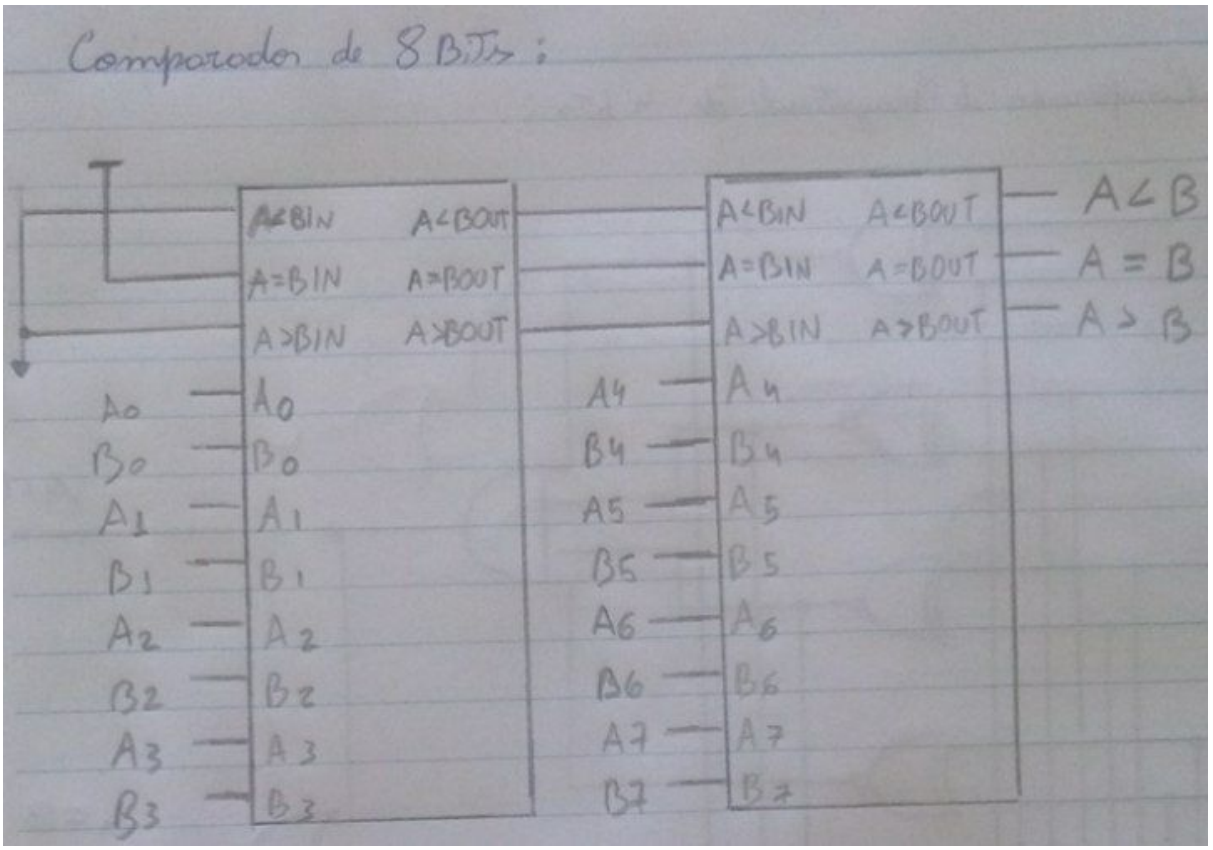


Questão 5:

a)


Comparador de Magnitude de 4 bits:
A3
B3
A2
B2
A1
B1
A0
B0
A=B
A>B
A>B

b)



c)

```
module comparador_4bits(output wire G, L, E, input wire [3:0] A, B);

   assign G = (A > B);
   assign L = (A < B);
   assign E = (A == B);

endmodule
```

ou

```
module Comparador_4bits (output wire G, L, E, input G_IN, input L_IN,
input E_IN, input wire [3:0] A, B);

   assign G = (A > B) || ((A == B) && G_IN);
   assign L = (A < B) || ((A == B) && L_IN);
   assign E = ((A == B) && E_IN);

endmodule
```

d)

```
module Comparador_4bits (output wire G, L, E, input G_IN, input L_IN,
input E_IN, input wire [3:0] A, B);

   assign G = (A > B) || ((A == B) && G_IN);
   assign L = (A < B) || ((A == B) && L_IN);
   assign E = ((A == B) && E_IN);

endmodule

module Comparador_8bits (output wire G, L, E, input wire [7:0] A, B);

   wire G_old, L_old, E_old;
   Comparador_4bits c1(G_old, L_old, E_old, 1'b0, 1'b0, 1'b1, A[3:0],
B[3:0]);
   Comparador_4bits c2(G, L, E, G_old, L_old, E_old, A[7:4], B[7:4]);

endmodule
```

e)

```
module stimulus;
   // Inputs
   reg [7:0] A;
   reg [7:0] B;

   // Outputs
   wire G;
   wire L;
   wire E;

   // Instantiate the Unit Under Test (UUT)
   Comparador_8bits uut (
       .A(A),
       .B(B),
       .G(G),
       .L(L),
       .E(E)
   );

   initial begin
       $dumpfile ("waves.vcd");
       $dumpvars (0, stimulus);
```

```
        // Initialize Inputs
        A = 8'b10000000;
        B = 8'b10000000;

        #20 A = 8'b11111111;
        #20 B = 8'b11111111;
        #20 A = 8'b10000000;
        #20 B = 8'b10000000;
        #20 A = 8'b11111111;
        #20 B = 8'b11111111;
        #20 A = 8'b10000000;
        #20 B = 8'b10000000;
        #20 A = 8'b11111111;
        #20 B = 8'b11111111;
        #20 A = 8'b10000000;
        #20 B = 8'b10000000;
        #20 A = 8'b11111111;
        #20 B = 8'b11111111;
        #20 A = 8'b10000000;
        #20 B = 8'b10000000;

        $finish;
    end
    initial begin
        $monitor ("T=%3d A=%d,B=%d,G=%d L=%d E=%d \n", $time, A, B, G,
L, E);
    end
endmodule
```

Imagem das formas de onda obtidas utilizando a bancada de testes descrita acima: