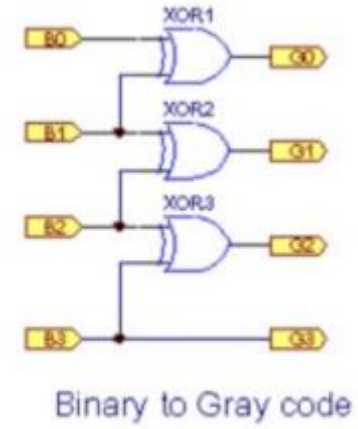
Илиян Кръстанов

Задача 1:



Binary to Gray code

Gate level:

```
module mod1 (input B0, B1, B2, B3,
                    output G);
wire G0, G1, G2, G3
xor XOR1 (G0, B0, B1);
xor XOR2 (G1, B1, B2);
xor XOR3 (G2, B2, B3);
endmodule
```

Dataflow:

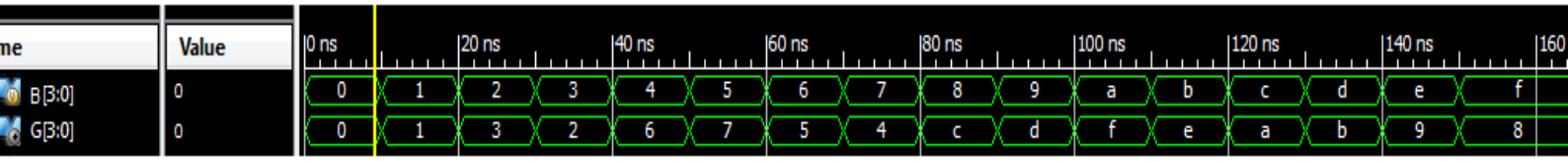
```
module mod1
      (input [3:0] B,
       output [3:0] G
      );
assign G[0] = B[1] ^ B[0];
assign G[1] = B[2] ^ B[1];
assign G[2] = B[3] ^ B[2];
assign G[3] = B[3];
endmodule
```

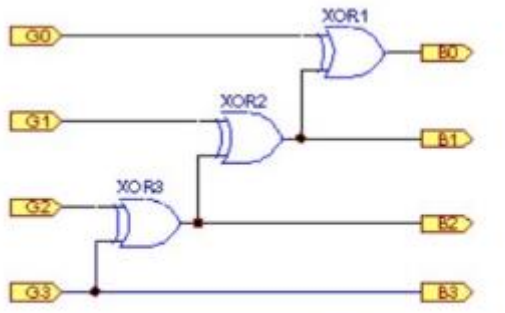
Behavioral:

```
module mod1 ( input B ,
                    output G );

wire [3:0] B ;
reg [3:0] G ;


always @ (B) begin
  if (B==0)
   G = 0;
  else if (B==1)
   G = 1;
  else if (B==2)
   G = 3;
  else if (B==3)
   G = 2;
  else if (B==4)
   G = 6;
  else if (B==5)
   G = 7;
  else if (B==6)
   G = 5;
  else if (B==7)
   G = 4;
  else if (B==8)
   G = 12;
  else if (B==9)
   G = 13;
  else if (B==10)
   G= 15;
  else if (B==11)
   G = 14;
  else if (B==12)
   G = 10;
  else if (B==13)
   G= 11;
  else if (B==14)
   G = 9;
  else
   G = 8;
 end
endmodule
```

TestBench:

```
module tb();
  reg [3:0] B;
  wire [3:0] G;
 mod1 uut1(.B(B),
           .G(G));
always
  begin
    B <= 0; #10;
      B <= 1;  #10;
      B <= 2;  #10;
      B <= 3;  #10;
      B <= 4;  #10;
      B <= 5;  #10;
      B <= 6;  #10;
      B <= 7;  #10;
      B <= 8;  #10;
      B <= 9;  #10;
      B <= 10;  #10;
      B <= 11;  #10;
      B <= 12;  #10;
      B <= 13;  #10;
      B <= 14;  #10;
      B <= 15;  #10;
      #100;
    $stop;
  end
endmodule
```





Gray code to Binary

Gate level:

```
module mod2 (input G0, G1, G2, G3,
                    output B);
wire B0, B1, B2, B3
xor XOR1 (B0, G0, B1);
xor XOR2 (B1, G1, B2);
xor XOR3 (B2, G2, G3);
endmodule
```

Dataflow:

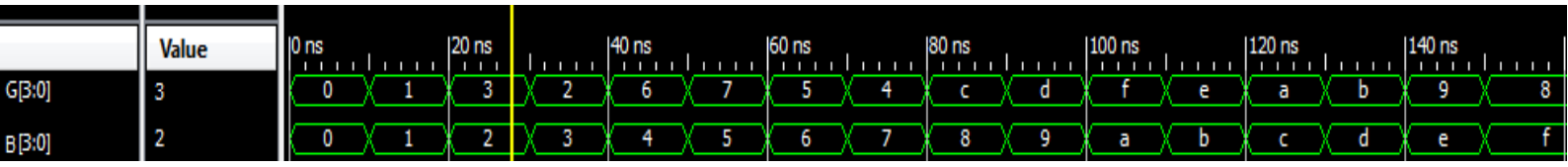
```
module mod2
      (input [3:0] G,
       output [3:0] B
      );
assign B[0] = G[3] ^ G[2] ^ G[1] ^ G[0];
assign B[1] = G[3] ^ G[2] ^ G[1];
assign B[2] = G[3] ^ G[2];
assign B[3] = G[3];
endmodule
```

Behavioral:

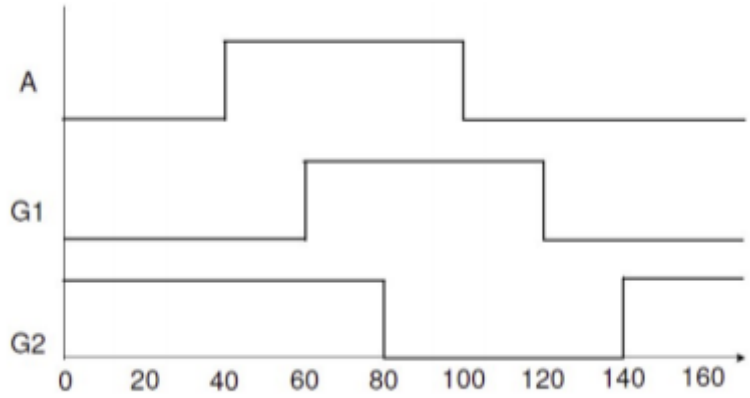
```
module mod2 ( input G ,
                    output B );
wire [3:0] G ;
reg [3:0] B ;

always @ (G) begin
  if (G==0)
   B = 0;
  else if (G==1)
   B = 1;
  else if (G==3)
   B = 2;
  else if (G==2)
   B = 3;
  else if (G==6)
   B = 4;
  else if (G==7)
   B = 5;
  else if (G==5)
   B = 6;
  else if (G==4)
   B = 7;
  else if (G==12)
   B = 8;
  else if (G==13)
   B = 9;
  else if (G==15)
   B= 10;
  else if (G==14)
   B = 11;
  else if (G==10)
   B = 12;
  else if (G==11)
   B = 13;
  else if (G==9)
   B = 14;
  else
   B = 15;
 end
endmodule
```

TestBench:

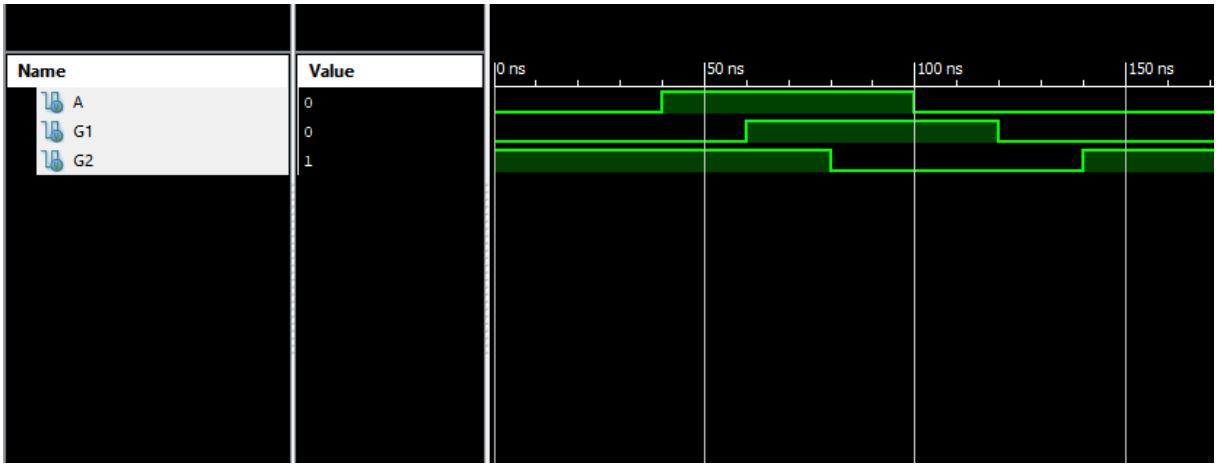
```
module tb();
   reg [3:0] G;
   wire [3:0] B;
  mod2 uut2(.G(G),
                    .B(B));
always
   begin
       G <= 0; #10;
       G <= 1;   #10;
        G <= 3;   #10;
        G <= 2;   #10;
        G <= 6;   #10;
        G <= 7;   #10;
        G <= 5;   #10;
        G <= 4;   #10;
        G <= 12;   #10;
        G <= 13;   #10;
        G <= 15;  #10;
        G <= 14;  #10;
        G <= 10;  #10;
        G <= 11;  #10;
        G <=  9; #10;
        #100;
      $stop;
   end
endmodule
```



Задача 2:




```
1   `timescale 1ns / 1ps
2
3   module task2_TB();
4
5     reg A;
6     reg G1;
7     reg G2;
8
9     test task2_test(.A(A),
10                    .G1(G1),
11                    .G2(G2));
12   initial
13     begin
14       A = 0;
15       G1 = 0;
16       G2 = 1;
17       #40 A = 1;
18       #20 G1 = 1;
19       #20 G2 = 0;
20       #20 A = 0;
21       #20 G1 = 0;
22       #20 G2 = 1;
23       #200
24       $finish;
25     end
26  endmodule
```



Задача 3:

1. Таблица на истинност:

A	M	C	f
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2. Аналитичен анализ:

f (2,3,5,6,7) = (~A.M.~C) + (~A.M.C) + (A.~M.C) + (A.M.~C) + (A.M.C)

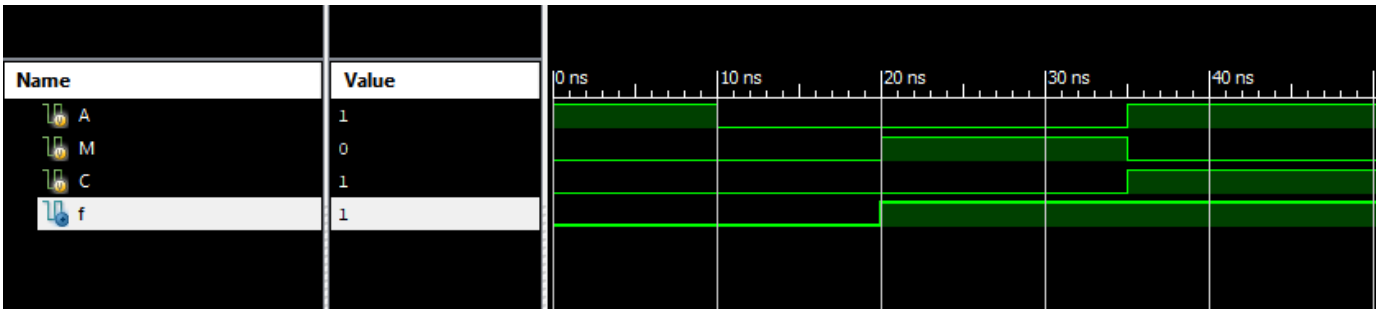
3. Карта на Карно:

	~C	C
~A.~M	0	0
~A.M	1	1
A.M	1	1
A.~M	0	1

f= M + AC

4.

```
module lock(input  A, M, C,
        output f);
                assign
                f = M | (A & C);
endmodule
```



TB:

```
 `timescale 1ns / 1ps
module lock_TB();
reg A;
reg M;
reg C;
wire f;
lock dut1(.A(A),
              .M(M),
               .C(C),
               .f(f) );
             initial
            begin
            A=1;
            M=0;
            C=0;
            #10 A=0;
            #10 M=1;
            #10 C=0;
            #5 A=1;
               M=0;
                C=1;
            #200;
  $finish;
end
endmodule
```

Задача 4: