

Név:	Index szám:	Pontszám: (max. 50)
------	-------------	------------------------

1. 3/8-as dekóder és megfelelő VAGY kapuk segítségével tervezze meg az $X=\Sigma(0,3,4,6,7)$ és az $Y=\Sigma(0,1,2,5)$ logikai függvényeket megvalósító hálózatot! Rajzolja meg a hálózat logikai rajzát!

(10 pont)

2. D flip-flop-ok és tetszőlegesen megválasztott logikai kapuk segítségével tervezzen egy olyan Moore-féle hálózatot, amely négybites, tízes alapú (dekadikus) számlálóként működik. A számlálás Gray-féle kódban történik (0000,0001,0011,0010,0110, 0111,0101,0100,1100,1101).

(15 pont)

3. Rajzolja meg a megadott Verilog HDL leírásnak megfelelő logikai rajzot és magyarázza el a modul működését!

(10 pont)

```
module exam (in1, in2, out1, out2);  
input [7:0] in1;  
input [2:0] in2;  
output [2:0] out1;  
output out2;  
assign out1 = (in1 >> 2) | in2;  
assign out2 = in1[in2];  
endmodule
```

4. Rajzolja le a lenti Verilog HDL leírásnak megfelelő állapotdiagramot! Jelölje be az állapotokat, valamint a bemenet és a kimenet értékét!

(15 pont)

```
always @(*) begin  
nxtState = state;  
out = 0;  
case (state)  
A : if (in) nxtState = C;  
else nxtState = B;  
B : if (in) begin  
out = 1;  
nxtState = C;  
end
```

```
C : if (~in) begin  
out = 1;  
nxtState = B;  
end  
default : begin  
out = 1'bX;  
nxtState = 2'bX;  
end  
endcase  
end
```

Ime:	Broj indeksa:	Broj poena: (max. 50)
------	---------------	--------------------------

1. Projektovati logičku mrežu za ostvarivanje logičkih funkcija $X = \Sigma(0,3,4,6,7)$ i $Y = \Sigma(0,1,2,5)$ primenom dekodera 3/8 i odgovarajućim ILI kolima! Nacrtati logičku šemu proračunatog kola!

(10 poena)

2. Pomoću D flip-flop-ova i proizvoljno odabranih logičkih kola projektovati Moore-ovu mrežu koja će raditi kao četvorobitni dekadni brojač (modula 10)! Brojač treba da broji u Gray-evom kodu (0000,0001,0011,0010,0110, 0111,0101,0100,1100,1101)!

(15 poena)

3. Nacrtati logičku šemu koja odgovara datom Verilog HDL opisu i objasniti šta radi navedeni modul!

(10 poena)

```
module exam (in1, in2, out1, out2);  
  input [7:0] in1;  
  input [2:0] in2;  
  output [2:0] out1;  
  output out2;  
  assign out1 = (in1 >> 2) | in2;  
  assign out2 = in1[in2];  
endmodule
```

4. Nacrtati dijagram stanja koji odgovara dole navedenom Verilog HDL opisu! Naznačiti vrednosti stanja, ulaza i izlaza!

(15 poena)

```
always @(*) begin  
  nxtState = state;  
  out = 0;  
  case (state)  
    A : if (in) nxtState = C;  
    else nxtState = B;  
    B : if (in) begin  
      out = 1;  
      nxtState = C;  
    end  
  end
```

```
C : if (~in) begin  
  out = 1;  
  nxtState = B;  
end  
default : begin  
  out = 1'bX;  
  nxtState = 2'bX;  
end  
endcase  
end
```