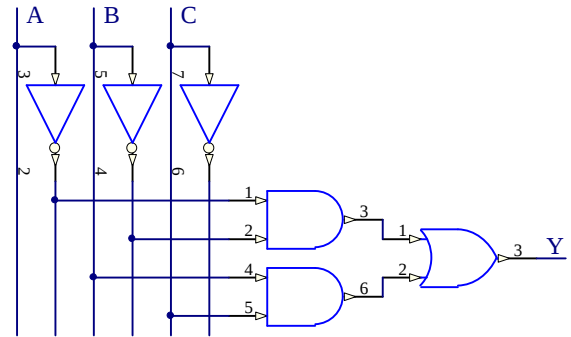


Név:	Index szám:	Pontszám: (max. 50)
------	-------------	------------------------

1. Írja fel a megadott logikai kapcsolásnak megfelelő $Y=f(A,B,C)$ függvényt, rajzolja meg a megfelelő Karnaugh táblát és az igazságtáblázatot! Írja fel az $\bar{Y}$ függvény minimalizált szorzatok összegeként, majd ez alapján alakítsa át az Y függvényt összegek szorzatára és rajzolja meg az ennek megfelelő minimális VAGY-ÉS hálózatot!

(10 pont)



2. D flip-flop-ok és tetszőlegesen megválasztott logikai kapuk segítségével tervezzen egy olyan Moore-féle hálózatot, amely $A=0$ vezérlőbemenet esetén Gray kódban előrefelé számlál, $A=1$ esetén viszont hátrafelé számlál, ugyancsak Gray kódban.

(15 pont)

3. A megadott Verilog HDL leírásban keresse meg és javítsa ki az összes hibát!
- ```

module Exam (sm_in, sm_clock, sm_out);
 parameter idle = 2'b00;
 parameter read = 2'b01;
 parameter write = 2'b11;
 parameter wait = 2'b1x;
 input sm_clock, reset, sm_in;
 output sm_out;
 reg [1:0] current_state, next_state;
 always @ (posedge sm_clock)
 begin
 if (reset = 1'b1)
 current_state <= 2'b00;
 else
 current_state <= next_state;
 end
 always @ (current_state or sm_in)
 begin
 sm_out = 1'b1;
 next_state = current_state;
 end
 end
 case (current_state)
 idle:
 sm_out = 1'b0;
 if (sm_in)
 next_state = 2'b11;
 write:
 sm_out = 1'b0;
 if (sm_in == 1'b0)
 next_state = 2'b10;
 read:
 if (sm_in == 1'b1)
 next_state = 2'b01;
 wait:
 if (sm_in == 1'b1)
 next_state = 2'b00;
 endcase
 end
 endmodule

```

(10 pont)

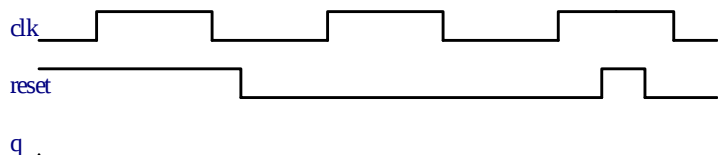
4. Rajzolja meg a  $q$  jel idődiagramját, hogy a viselkedés megfeleljen az alábbi hardver-leírásnak!

(15 pont)

```

module Exam (q, clk, reset);
 input clk, reset;
 output q;
 reg q;
 always @ (negedge clk)
 if (reset)
 q <= 1'b0;
 else
 q <= ~q;
 endmodule

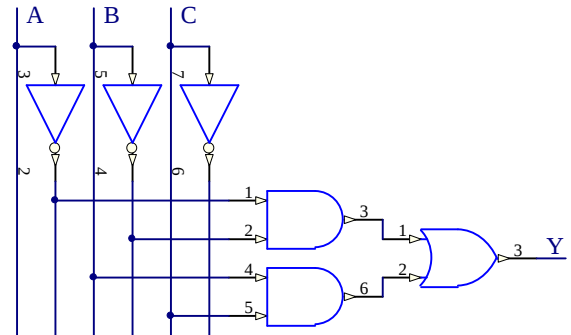
```



|      |               |                          |
|------|---------------|--------------------------|
| Ime: | Broj indeksa: | Broj poena:<br>(max. 50) |
|------|---------------|--------------------------|

1. Napisati logičku funkciju  $Y=f(A,B,C)$  koja odgovara datoj logičkoj šemi i nacrtati odgovarajuću Karnaugh-ovu tablicu i tablicu istinitosti! Napisati funkciju  $\bar{Y}$  u formi minimizirane sume proizvoda, zatim na bazi toga izvesti funkciju  $Y$  u formi proizvoda logičkih suma i nacrtati minimalnu ILI-I mrežu!

(10 poena)



2. Pomoću  $D$  flip-flop-ova i proizvoljno odabranih logičkih kola projektovati Moore-ovu mrežu koja pri kontrolnom ulazu  $A=0$  broji prema gore u Gray-ovom kodu, a pri ulazu  $A=1$  broji prema dole, isto u Gray-ovom kodu.

(15 poena)

3. U datom Verilog HDL opisu pronađite i korigujte sve greške!

(10 poena)

```

module Exam (sm_in, sm_clock, sm_out);
 parameter idle = 2'b00;
 parameter read = 2'b01;
 parameter write = 2'b11;
 parameter wait = 2'b1x;
 input sm_clock, reset, sm_in;
 output sm_out;
 reg [1:0] current_state, next_state;
 always @ (posedge sm_clock)
 begin
 if (reset = 1'b1)
 current_state <= 2'b00;
 else
 current_state <= next_state;
 end
 always @ (current_state or sm_in)
 begin
 sm_out = 1'b1;
 next_state = current_state;
 endcase
 idle:
 sm_out = 1'b0;
 if (sm_in)
 next_state = 2'b11;
 write:
 sm_out = 1'b0;
 if (sm_in == 1'b0)
 next_state = 2'b10;
 read:
 if (sm_in == 1'b1)
 next_state = 2'b01;
 wait:
 if (sm_in == 1'b1)
 next_state = 2'b00;
 endcase
end
endmodule

```

4. Docrtati vremenski dijagram za signal  $q$  tako da ponašanje hardvera odgovara datom opisu!

(15 poena)

```

module Exam (q, clk, reset);
 input clk, reset;
 output q;
 reg q;
 always @ (negedge clk)
 if (reset)
 q <= 1'b0;
 else
 q <= ~q;
 endmodule

```

