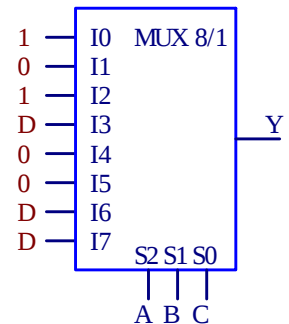


Név:

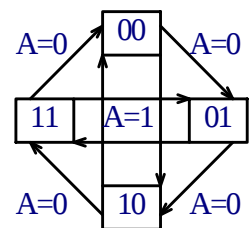
Index szám:

Pontszám:  
(max. 50)

1. Az  $Y = f(A, B, C, D)$  függvényt az ábrán megadott módon, 8/1-es multiplexerrel valósítottuk meg. Rajzolja meg a függvény igazságtáblázatát, majd végezze el a minimalizációt Karnaugh tábla segítségével! A minimalizált függvényt valósítsa meg tisztán NAND kapukkal! (10 pont)



1. D flip-flop-ok és tetszőlegesen megválasztott logikai kapuk segítségével tervezzen egy olyan Moore-féle hálózatot, amely a mellékelt állapotdiagramon megadott módon viselkedik, az A bemenet értékétől függően! A négyszögekben az állapotok kódját adtuk meg ( $Q1, Q0$ ). (15 pont)



2. Rajzolja meg a megadott Verilog HDL leírásnak megfelelő állapot-gráfot és magyarázza el a modul működését! (10 pont)

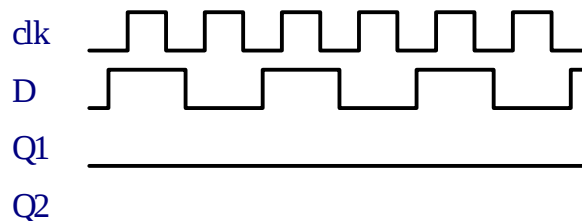
```
module seq0_detect (x,clk, y);
input x, clk;
output y;
reg [1:0] state;
parameter S0=2'b00, S1=2'b01, S2=2'b11,
S3=2'b10;
always @(posedge clk)
case (state)
S0: if (x) state <= S1;
else state <= S3;

```

```
S1: if (x) state <= S2;
else state <= S0;
S2: if (x) state <= S3;
else state <= S1;
S3: if (x) state <= S0;
else state <= S2;
endcase
assign y = (state == S0);
endmodule
```

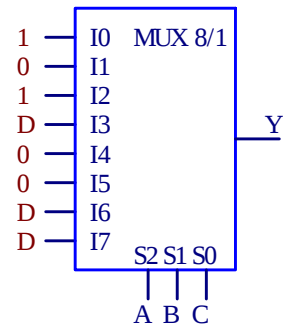
3. Rajzolja le a lenti Verilog HDL leírásból szintetizálható hardver rajzát és adja meg a  $Q1$  és  $Q2$  jelek idődiagramját, hogy a viselkedés megfeleljen az alábbi hardver-leírásnak! Tételezze fel, hogy a szimuláció kezdetén mindkét jel nulla értékű volt! (15 pont)

```
module exam (D, clk, Q1, Q2)
input D, clk;
output Q1, Q2;
reg Q1, Q2;
always @ (posedge clk)
begin
Q1<=D;
Q2<=Q1
end
endmodule
```

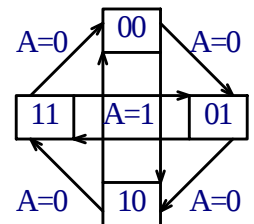


|      |               |                          |
|------|---------------|--------------------------|
| Ime: | Broj indeksa: | Broj poena:<br>(max. 50) |
|------|---------------|--------------------------|

1. Logičku funkciju  $Y = f(A, B, C, D)$  je ostvarena pomoću multipleksora 8/1, kao na slici. Nacrtati kombinacionu tablicu i obaviti minimizaciju funkcije primenom Karnaugh-ove tablice! Minimiziranu funkciju ostvariti čisto pomoću NAND kapija!  
(10 poena)



2. Pomoću D flip-flop-ova i proizvoljno odabranih logičkih kola projektovati Moore-ovu mrežu koja se ponaša prema datom dijagramu stanja, u zavisnosti od ulaza A! U pravougaonicima su dati kôdovi stanja ( $Q1, Q0$ ).  
(15 poena)



3. Nacrtati dijagram stanja koji odgovara datom Verilog HDL opisu i objasnite šta radi navedeni modul!  
(10 poena)

```
module seq0_detect (x,clk, y);
input x, clk;
output y;
reg [1:0] state;
parameter S0=2'b00, S1=2'b01, S2=2'b11,
S3=2'b10;
always @(posedge clk)
case (state)
S0: if (x) state <= S1;
else state <= S3;

```

```
S1: if (x) state <= S2;
else state <= S0;
S2: if (x) state <= S3;
else state <= S1;
S3: if (x) state <= S0;
else state <= S2;
endcase
assign y = (state == S0);
endmodule
```

4. Nacrtati hardver koji će se sintetizovati na bazi dole navedenog Verilog HDL opisa i dati vremenske dijagrame za signale Q1 i Q2 tako da ponašanje hardvera odgovara datom opisu! Pretpostaviti da su na početku simulacije oba signala imali nultu vrednost!  
(15 poena)

```
module exam (D, clk, Q1, Q2)
input D, clk;
output Q1, Q2;
reg Q1, Q2;
always @ (posedge clk)
begin
Q1<=D;
Q2<=Q1
end
endmodule
```

