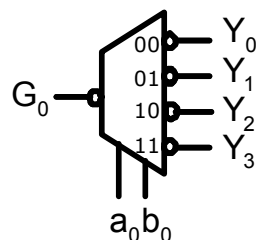


RAZVOJ DIGITALNIH SISTEMOV

Izpit 13. 06. 2012

- Realizirajte funkcijo $f^4 = V(1, 2, 4, 7)$ z redundantnimi mintermi pri $V_x(0, 3, 6)$ z enim TTL dekodermem 74139. Dekoder 74139 ima vhod za omogočenje elementa (G) in izhode Y_0, Y_1, Y_2, Y_3 v negativni logiki. Njegovo delovanje povzema spodnja tabela:

G_0	a_0	b_0	Y_0	Y_1	Y_2	Y_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

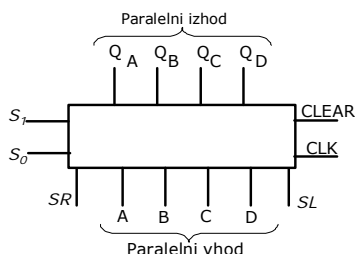


- Uporabite ROM vezje za realizacijo naslednjih funkcij:

$$g_1 = x_1 + x_2 \cdot \overline{x_3} \quad g_2 = \overline{x_1} \cdot x_3 + x_1 \cdot x_2$$

ROM vezje ima 3 naslovne spremenljivke in 2 bitno vsebino. Narišite shemo elementa in v shemi označite programirane povezave oz. 'varovalke' s piko (●).

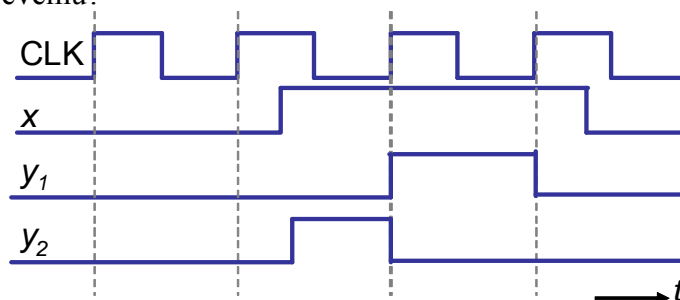
- Z uporabo D flip-flopov, ki so proženi na pozitivni rob signala ure CLK , prikažite sintezo univerzalnega 4-bitnega pomikalnega registra, ki ima dva funkcijska vhoda S_0 in S_1 in opravlja funkcije po spodnji tabeli: Register ima tudi zaporedna vhoda za pomik v levo (SL – serial left), pomik v desno (SR – serial right) in asinhroni vhod za brisanje CLEAR (aktiven nizek).



S_1	S_0	funkcija
0	0	drži stanje
0	1	pomik vsebine eno mesto desno
1	0	pomik vsebine eno mesto levo
1	1	nalaga vsebino z vhodov ABCD

- Z uporabo D flip-flopov, ki so proženi na sprednji rob signala ure (CLK) načrtajte Mooreov ali Mealyev avtomat končnih stanj, katerega izhod y postane '1' ko se vhod x spremeni iz logične '0' na '1'.

Delovanje avtomata povzema spodnja slika. Narisani sta dve realizaciji izhoda avtomata y_1 in y_2 : Kateri izhod (y_1 ali y_2) pripada Moore-ovemu tip avtomata in kateri Mealy-evemu?



Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

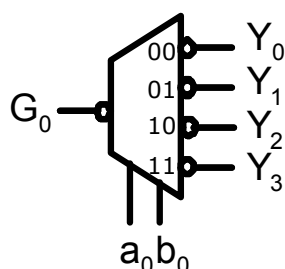
Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VŠŠ, UNI).

Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>

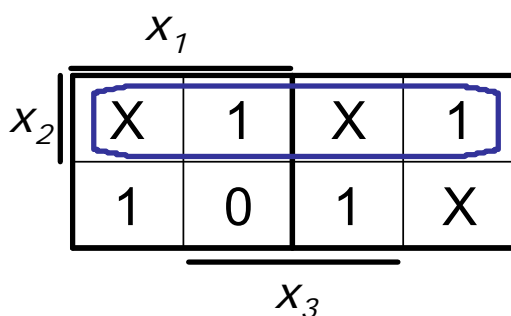
Rešitev 1. naloge:

Delovanje dekoderja 74139¹ povzema spodnja tabela:

G_0	a_0	b_0	Y_0	Y_1	Y_2	Y_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

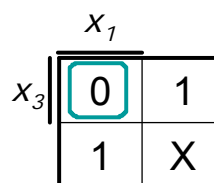


Funkcijo f narišemo v Veitch–ev diagram, da si jo lažje predstavljamo:

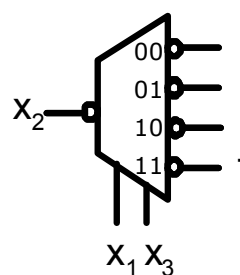


Čim imamo na voljo dekodirnik z ENABLE vhomom v negativni logiki preverimo ali obstaja spremenljivka v osnovni ali negirani obliki, pri kateri so vsa polja enaka '1' vključno z redundancami X.

V zgornjem Veitch–evem diagramu je to spremenljivka x_2 : Namreč, če je $x_2=1$, potem lahko vse redundance izberemo tako, da bo $f=1$ za vse vrednosti $x_2=1$. Ko določimo spremenljivko za omogočenje elementa (G), opazujemo samo preostali del Veitch–evega diagrama. Spodnje 4 vrednosti diagrama narišemo v novem Veitch–evem diagramu 2 spremenljivk.



Dekoder ima aktivno nizke izhode, zato iz nastalega diagrama realiziramo *negacijo* funkcije f zato v Veitch–evem diagramu združujemo ničle, kar nastopa samo v primeru ko sta $x_1=1$ in $x_3=1$.



¹<http://www.alldatasheet.com/view.jsp?Searchword=74HC139>

Rešitev 2. naloge:

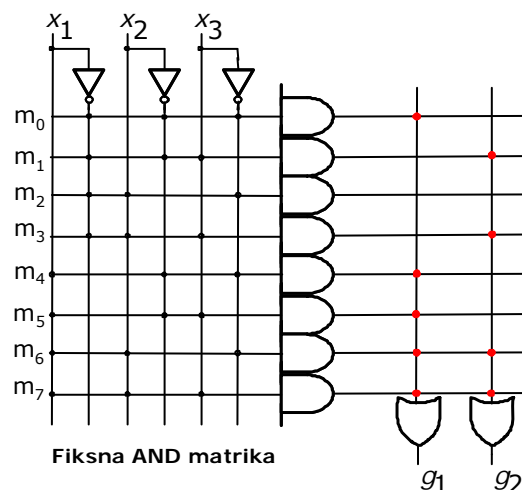
Če se funkcije ne nahajajo v popolni disjunktivni normalni obliki (PDNO), jih prevedemo v to obliko z uporabo pravil Boole-ove algebre. Funkcijo lahko tudi izpišemo v Veitch-ev diagram in izpišemo številke mintermov, kjer je funkcija enaka '1'.

$$\begin{aligned} g_1(x_1, x_2, x_3) &= x_1 + \overline{x_2} \cdot \overline{x_3} = x_1 \cdot (\overline{x_2} \cdot \overline{x_3} + x_2 \cdot \overline{x_3} + \overline{x_2} \cdot x_3 + x_2 \cdot x_3) + (\overline{x_1} + x_1) \cdot \overline{x_2} \cdot \overline{x_3} \\ g_1(x_1, x_2, x_3) &= x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot \overline{x_3} \\ g_1(x_1, x_2, x_3) &= x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot x_3 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \\ g_1(x_1, x_2, x_3) &= V(4, 6, 5, 7, 0) \end{aligned}$$

Podobno storimo še za preostale funkcije:

$$\begin{aligned} g_2 &= \overline{x_1} \cdot x_3 + x_1 \cdot x_2 = \overline{x_1} \cdot (\overline{x_2} + x_2) \cdot x_3 + x_1 \cdot x_2 \cdot (\overline{x_3} + x_3) \\ g_2(x_1, x_2, x_3) &= \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3 \\ g_2(x_1, x_2, x_3) &= V(1, 3, 6, 7) \end{aligned}$$

PDNO je najprimernejša oblika za realizacijo z ROM, ker je matrika AND fiksna. Programirane vrednosti AND matrike predstavljajo vse minterme funkcije treh spremenljivk ($x_1 \ x_2 \ x_3$) od m_0 do m_7 . Številka minterma določa naslov lokacije ROM pomnilnika.



Narišemo celotno vezje ROM strukture in vstavimo pike (•) v OR matriki tam, kjer želimo programirati določeno spremenljivko v členu PDNO.

Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VŠŠ, UNI).

Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>

Rešitev 3. naloge:

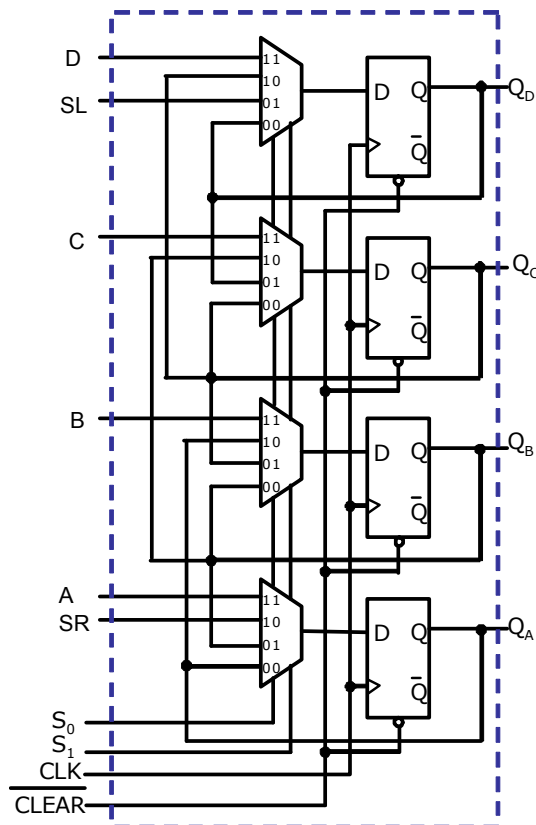
Vsako od operacij izpišemo v pravilnostno tabelo v kateri združimo funkcijska bita S_1 in S_0 in trenutno stanje na i -tem mestu registra $Q_i(t)$. Mesta registra od leve proti desni so $Q_i = (Q_A, Q_B, Q_C, Q_D)$. Realizacija z D flip-flopi nam analizo močno poenostavi, zaradi enačbe D flip-flopa: $D = Q(t+1)$.

Tabela 1: Prehajanje stanj univerzalnega registra.

S_1	S_0	$Q_i(t+1)$	funkcija
0	0	$Q_i(t)$	HOLD
0	1	$Q_{i+1}(t)$	SHR
1	0	$Q_{i-1}(t)$	SHL
1	1	x_i	LOAD

Iz poenostavljene tabele prehajanja stanj univerzalnega registra sestavimo realizacijo, ki bo vključevala izbiralnike MUX 4/1 in D-FF.

Na naslovna vhoda vseh MUX 4/1 vodimo funkcijska signala S_1 in S_0 . Potem na vsakem podatkovnem vhodu realiziramo ustrezno funkcijo.

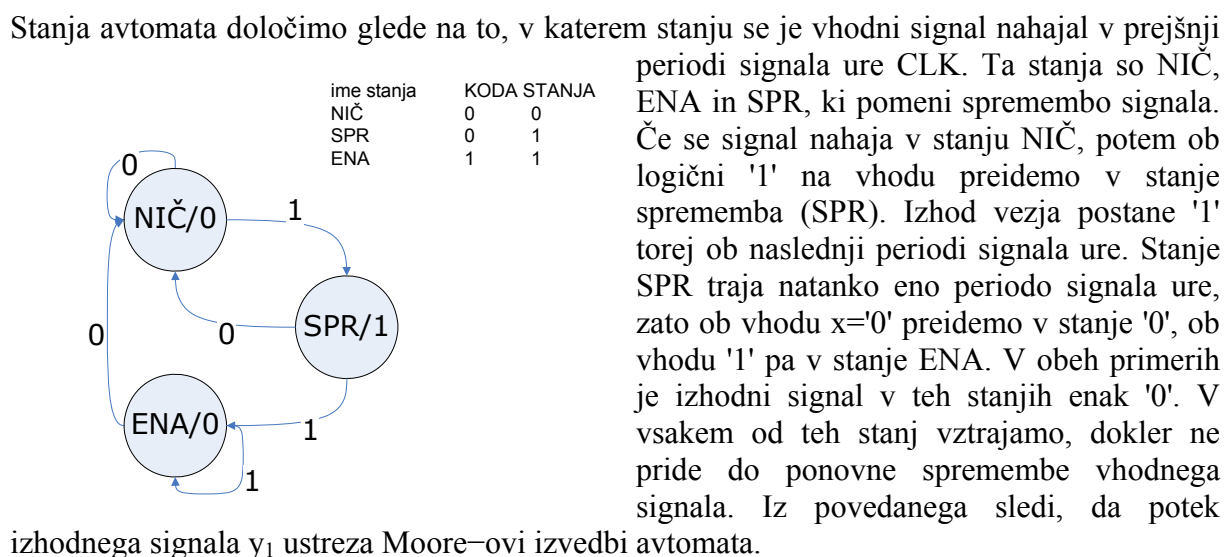


Stanje $S_1S_0 = "00"$ pomeni držanje stanja (*HOLD*), torej bodo trenutne vrednosti D-FF ohranile vrednost $Q_i(t+1) = Q_i(t)$. Na sliki to realiziramo tako, da vodimo izhod D-FF nazaj na vhod pri podatkovnem vhodu 00. Stanje $S_1S_0 = "01"$ pomeni pomik desno (*SHR* – ang. shift right), torej bodo D-FF pomaknili vsebino eno mesto desno. Pomik desno pomeni, da na mesto skrajno levega bita vpišemo vrednost zaporednega vhoda SR, nato Q_A vodimo na vhod Q_B in tako do skrajno desnega bita. Stanje $S_1S_0 = "10"$ pomeni pomik levo (*SHL* – ang. shift left), torej bodo D-FF pomaknili vsebino eno mesto levo. Pomik levo pomeni, da na mesto skrajno desnega bita vpišemo vrednost zaporednega vhoda SL, nato Q_D vodimo na vhod Q_C in tako do skrajno levega bita. $S_1S_0 = "11"$ pomeni vzporedno nalaganje z vhodov (*LOAD*) $Q_D(t+1) = D$, $Q_C(t+1) = C$, $Q_B(t+1) = B$, $Q_A(t+1) = A$. Na tabeli smo i -ti vhod za vzporedno nalaganje označili kot $x_i = (A, B, C, D)$.

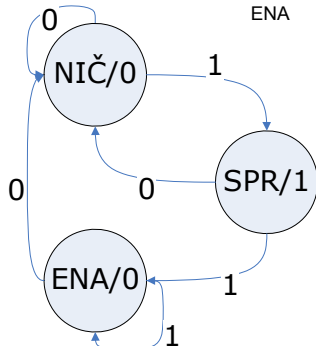
Rešitev 4. naloge:

Moore-ova realizacija avtomata končnih stanj.

Diagram stanj:



ime stanja	KODA STANJA
NIČ	0
SPR	1
ENA	1

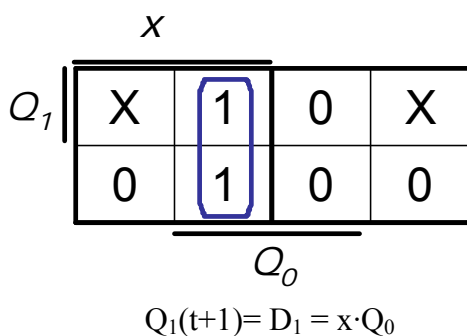


Narišemo tabelo prehajanja stanj

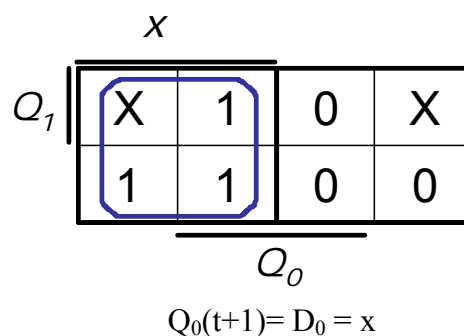
vhod in trenutno stanje			naslednje stanje		enačbe FF in izhoda		
x	Q ₁	Q ₀	Q ₁	Q ₀	D ₁	D ₀	y
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1
0	1	0	X	X	X	X	X
0	1	1	0	0	0	0	0
1	0	0	0	1	0	1	0
1	0	1	1	1	1	1	1
1	1	0	X	X	X	X	X
1	1	1	1	1	1	1	0

Iz tabele prehajanja stanj avtomata določimo enačbe D-FF:

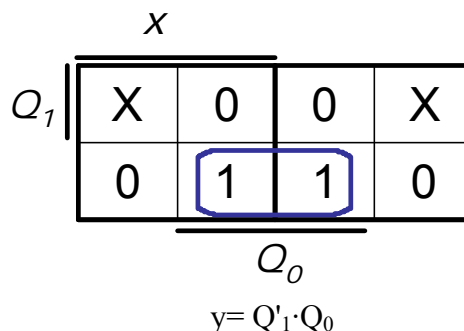
Za Q₁ narišemo Veitchev diagram



Podobno za Q₀ narišemo Veitchev diagram



In še za izhod y:



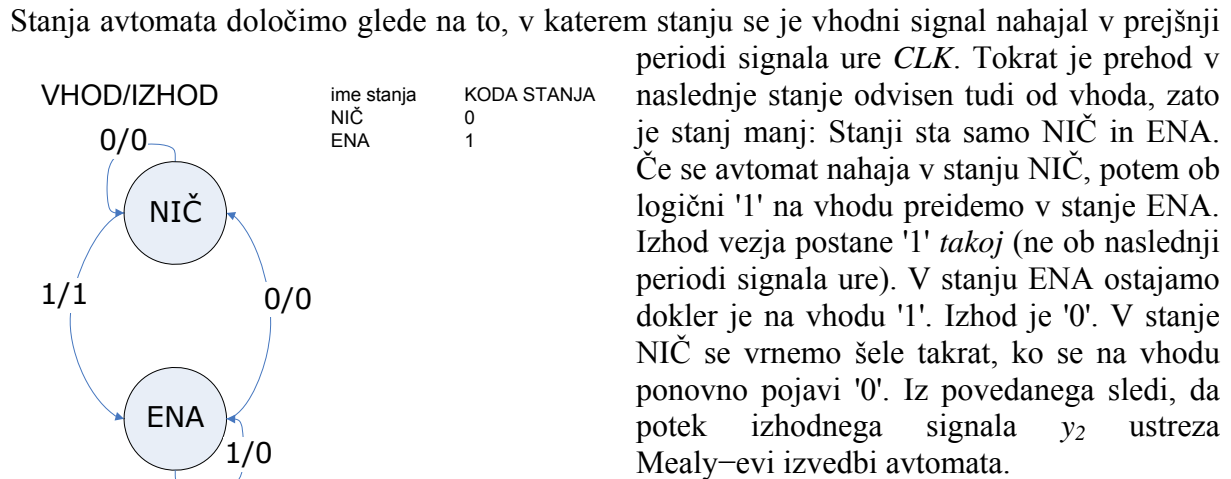
Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VSŠ, UNI).

Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>

Mealy–eva realizacija avtomata končnih stanj.

Diagram stanj:



Narišemo tabelo prehajanja stanj:

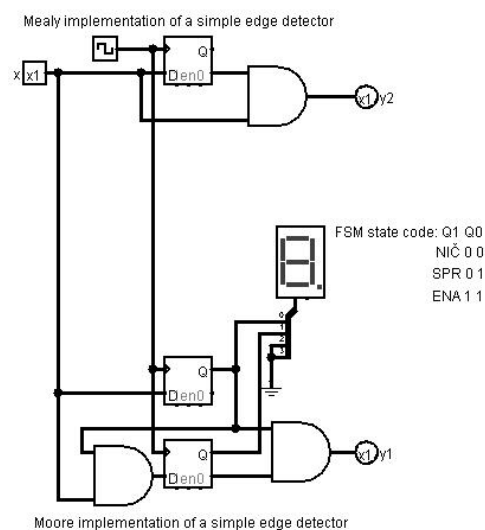
vhod in trenutno stanje		naslednje stanje	D–FF in izhod	
x	Q	Q	D	y
0	0	0	0	0
0	1	0	0	0
1	0	1	1	1
1	1	1	1	0

Veitchevih diagramov nam ni treba risati, saj enačbi D–FF in izhoda neposredno sledita iz tabele.

$$Q(t+1)=x \quad \text{in} \quad y=x \cdot Q'(t)$$

Vezje se nahaja v Logisim predlogah rešenih nalog na domači strani predmeta:

Logisim\fsm\front_edge_detector_mealy_moore.circ



Čas pisanja je 60 minut. Vsaka naloga je vredna 10 točk.

Na list z rešitvami se podpišite in napišite še vpisno številko ter kateri predmet pišete (VSŠ, UNI).

Rezultati bodo objavljeni na: <https://estudent.fri.uni-lj.si>