UNIVERZITET U TUZLI

Fakultet elektrotehnike
Odsjek: Automatika i robotika (AR)
Godina studija: II
Predmet: **Projektovanje logičkih sistema**

Projektni zadatak 2

Grupa 3

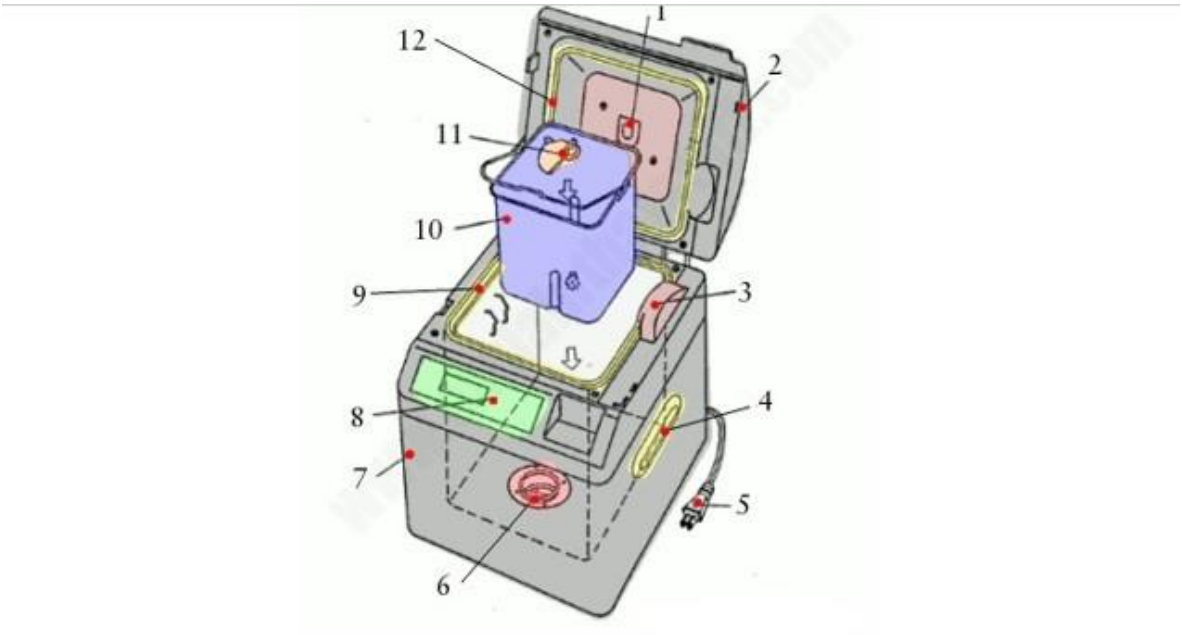
Profesor:
Vanr. prof. dr. Lejla Banjanović-Mehmedović

Studenti:
Irvana Hrustić
Emina Šabačkić
Damir Mijatović

Sadržaj

Tekst zadatka

Ideja našeg projekta je realizacija Breadmaking machine-a (mašina konačnog stanja za proizvodnju hljeba ili tijesta - pekač).
Naša mašina funkcioniše na sljedeći način: dugme B pokreće rad našeg automata. Nakon paljenja, korisnik unosi željene sastojke. Mašina neće početi s radom dok korisnik ne unese količinu sastojaka određene mase, to jeste dok je prazna. Nakon toga naš uređaj provjerava da li je poklopac zatvoren, jer ne može funkcionisati s otvorenim poklopcem (npr. radnja pečenja). Nakon ispunjenih uslova i pokretanja mašine, ista kreće s procesima miješanja i gnječenja. Ovi procesi su kontrolirani od strane timer-a, dakle izvršavat će se sve dok timer ne signalizira da su ovi procesi završeni. Zatim slijedi proces nadolaženja tijesta. Ovaj proces će biti kontrolisan senzorom za mjerenje visine tijesta. Dakle, kada tijesto dosegne određenu visinu, značit će da je spremno za upotrebu ili pečenje. Sada se korisniku obezbjeđuju dvije opcije. Jedna je ta da proces nastavi ići dalje do pečenja, koje je također kontrolirano timer-om, a druga opcija je ta da korisnik pritisne dugme B, i time vrati našu mašinu u početno stanje, preskakajući proces pečenja. (ova opcija je za slučaj da korisnik želi iskoristiti tijesto u druge svrhe). Nakon pečenja, odnosno nakon što timer signalizira kraj stanja, mašina se gasi i time je proces završen.
Napomena: ponovno stiskanje dugmeta B će u bilo kom trenutku zaustaviti rad mašine i vratiti na početno stanje (ukoliko je radnja određenog stanja izvršena).



1. Baking tin cover integrated into folding lid.
2. Folding, hinged lid contains steam exhaust vent in the center.
3. Exhaust port allows steam to escape from baking tin and flow out through lid.
4. Air vent allows air into the dough to help it rise.
5. Power supply lead.
6. Hole where baking tin screws securely into place containing, at its center, the motorized axle that turns the kneading paddle.
7. Plastic outer case insulates the oven and makes the breadmaker safe to touch during operation.
8. Simple LCD display and wipe-clean touch-control panel.
9. Recessed groove where the bottom of the lid locks in.
10. Removable baking tin will bake a single loaf at a time.
11. Detachable kneading paddle clicks onto an axle slotted through the center of the tin in a waterproof seal.
12. Tongue around edge of lid fits securely into groove (9) in bottom part of machine to ensure heat is retained.

Slika 1: Arhitektura pekača.

Slika 2: Vanjski izgled pekača

Dakle, razmatramo sljedeća moguća stanja:

1.) START (000)
2.) WEIGHT (001)
3.) LID (010)
4.) MIXING (011)
5.) RISING (100)
6.) BAKING (101)

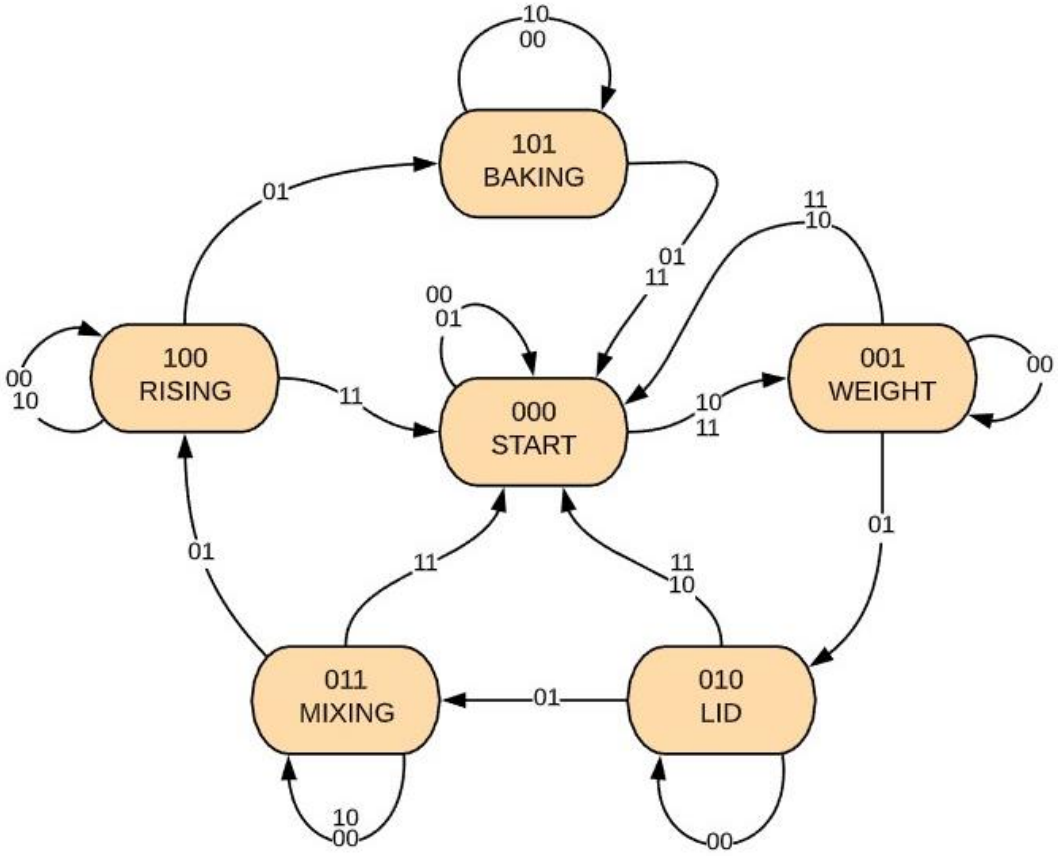
Za realizaciju našeg projekta, koristit ćemo sljedeće ulaze: B (button, dugme za paljenje/gašenje), W(senzor za provjeru mase sastojaka), LC(senzor koji provjerava da li je poklopac otvoren/zatvoren), T (Timer koji nam govori da li je određeni korak izvršen) i HM (senzor za mjerenje visine tijesta).
S ciljem da lakše reailiziramo našu mašinu, planirano je korištenje 4-to-1 multiplexer-a za svođenje ulaza W, LC, T, HM na jedan izlaz, M.

Sada, vidimo da naša mašina konačnog stanja ima sljedeće specifikacije:

Ulazi_FSM: B, M
Izlazi_FSM: q1, q2, q3
Izlazi_7seg: a, b, c, d, e, f, g
Ulaz clock-a: clk_50MHz

Obzirom da predstavljamo sedam mogućih stanja, potrebno nam je najviše 8 različitih kombinacija, te ćemo za implementaciju koristiti tri flip-flopa. Konkretno, koristit ćemo jedan D i dva T flip flopa.

Dijagram stanja



Slika 3: Dijagram stanja naše FSM

Početno stanje naše mašine je stanje START. Nakon stiskanja BUTTON-a (B – što je jedini ulaz od kojeg ovo stanje ovisi, drugi ulazi nas ne zanimaju) mašina prelazi u stanje WEIGHT. U ovom stanju se provjerava da li je pekač prazan, jer mašina ne može započeti naredne procese ukoliko je prazna. Ovo stanje ovisi od senzora WEIGHT (W) koji je u stanju logičke jedinice kada je masa u pekaču veća od 0 (nije prazan), dakle u ovom stanju se na izlaz multipleksera prosljeđuje W. Ukoliko je W u stanju jedinice, i nije ponovo stisnuto dugme B, mašina prelazi u naredno stanje. Ukoliko je pak stisnut gumb B, mašina se gasi tj. vraća u početno stanje. Naredno stanje je stanje LID, gdje se provjerava da li je poklopac pekača zatvoren, jer pekač ne može praviti i peći tijesto ukoliko je otvoren. Sada na izlazu multipleksera gledamo senzor LC koji je u stanju jedinice kada je poklopac uspješno zatvoren. Opet, kao i u prethodnom stanju, ukoliko je nanovo stisnut gumb B, mašina se vraća u početno stanje. Sada prelazimo u stanje MIXING, odnosno miješanje sastojaka i stvaranje tijesta. U ovom stanju izlaz našeg multiplekera odgovara timer-u T, odnosno miješanje će trajati sve dok timer ne signalizira da je došlo do kraja (T=1). U ovom slučaju ignorišemo dugme B, sve dok timer ne bude u stanju logičke jedinice, jer je zabranjeno ugasiti uređaj dok je radnja miksanja u tijeku. Kada timer bude u stanju jedinice, i ako je stisnut gumb, tek tada će se mašina moći vratiti u početno stanje. Kada mašina završi s miksanjem, prelazimo u stanje RISING, odnosno stanje nadolaženja tijesta, gdje obraćamo pažnju na

ulaz HM, odnosno senzor za mjerenje visine tijesta koji je u ovom stanju jednak izlazu multipleksera, M. HM će biti u stanju 0 sve dok tijesto ne nadođe do određene visine, nakon čega prelazi u stanje 1. Tada, ukoliko nije stisnut gumb, mašina prelazi u posljednju fazu pečenja. No ako je sisnut gumb nakon što je tijesto nadošlo, mašina se vraća u početno stanje i korisnik može iskoristiti tijesto po vlastitom izboru. I posljednje stanje je BAKING, tj. Pečenje. Ovo stanje, kao i stanje MIXING, ovisi o timer-u, što znači da će pečenje trajati sve dok timer ne signalizira da je pečenju došao kraj, tj kada je u stanju 1. Kao i u prethodna dva stanja, nije moguće prekinuti stanje pečenja dugmetom B, jer timer ima tačno određeno vrijeme potrebno da se tijesto pravilno ispeče. Kada T bude jednako jedinici, to znači da je mašina završila s pečenjem, i uređaj se gasi.

Tabela istinitosti

Trenutno stanje			Ulazi		Sljedeće stanje			Flip flopovi		
Q2	Q1	Q0	B	M	q2	q1	q0	D2	T1	T0
000 (START)			0	0	0	0	0	0	0	0
			0	1	0	0	0	0	0	0
			1	0	0	0	1	0	0	1
			1	1	0	0	1	0	0	1
001 (WEIGHT)			0	0	0	0	1	0	0	0
			0	1	0	1	0	0	1	1
			1	0	0	0	0	0	0	1
			1	1	0	0	0	0	0	1
010 (LID)			0	0	0	1	0	0	0	0
			0	1	0	1	1	0	0	1
			1	0	0	0	0	0	1	0
			1	1	0	0	0	0	1	0
011 (MIXING)			0	0	0	1	1	0	0	0
			0	1	1	0	0	1	1	1
			1	0	0	1	1	0	0	0
			1	1	0	0	0	0	1	1
100 (RISING)			0	0	1	0	0	1	0	0
			0	1	1	0	1	1	0	1
			1	0	1	0	0	1	0	0
			1	1	0	0	0	0	0	0
101 (BAKING)			0	0	1	0	1	1	0	0
			0	1	0	0	0	0	0	1
			1	0	1	0	1	1	0	0
			1	1	0	0	0	0	0	1

Minimizacija Karnaugh-ovim mapama

Q2=0

D2 Flip flop

BM / Q1Q0	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	1	0	0
10	0	0	0	0

Q2=1

BM / Q1Q0	00	01	11	10
00	1	1	0	1
01	1	0	0	1
11	x	x	x	x
10	x	x	x	x

$$D2 = Q_2\overline{M} + Q_2\overline{Q_0B} + Q_1Q_0\overline{B}M$$

T1 Flip flop

Q2=0

Q1Q0 \ BM	00	01	11	10
00	0	0	0	0
01	0	1	0	0
11	0	1	1	0
10	0	0	1	1

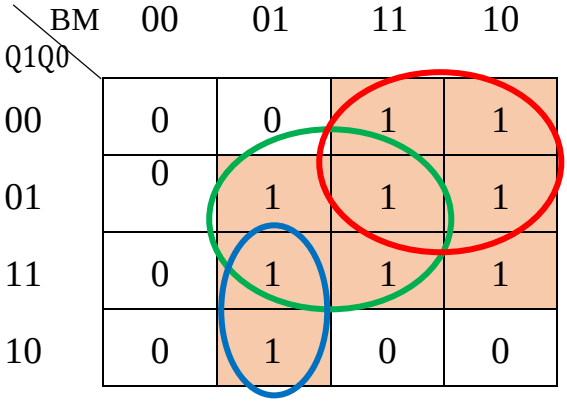
Q2=1

Q1Q0 \ BM	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	x	x	x	x
10	x	x	x	x

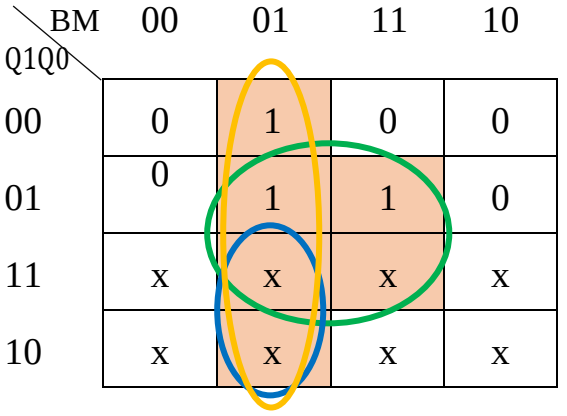
$$T1 = Q_1\overline{Q_0}B + Q_1BM + \overline{Q_2}Q_0\overline{B}M$$

T0 Flip flop

Q2=0



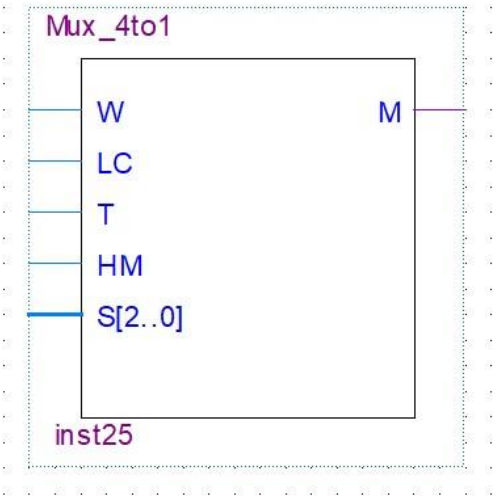
Q2=1



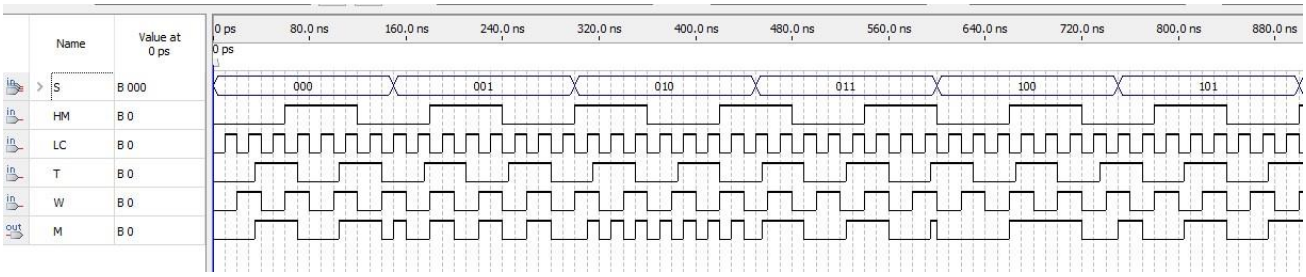
$$T0 = Q_0M + \overline{Q_2Q_1}B + Q_1\overline{B}M + Q_2\overline{B}M$$

Verilog modul za 4-to-1 multiplexer

```
module Mux_4to1(W,LC,T,HM,M,S);
input W,LC,T,HM;
input[2:0]S;
output reg M;
always @ (*)
if(S==3'b001)
M=W;
else if(S==3'b010)
M=LC;
else if(S==3'b011 || S==3'b101)
M=T;
else if(S==3'b100)
M=HM;
else
M=T;
endmodule
```



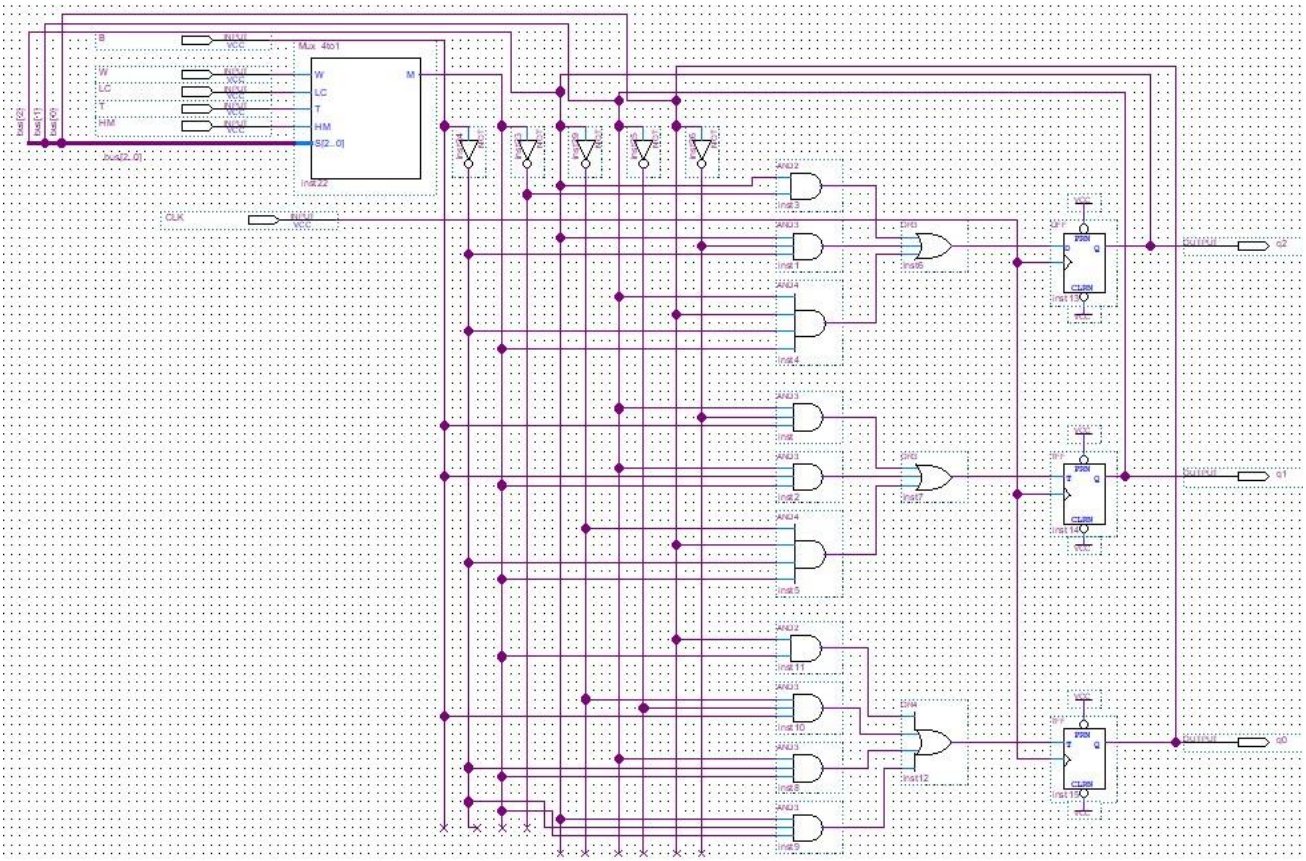
Slika 4: Simbol našeg 4-to-1 multiplexer-a – Quartus



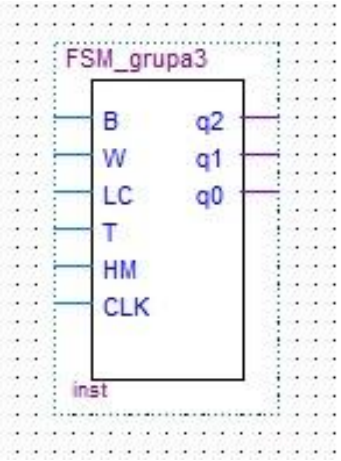
Slika 5: Quartus Waveform za naš Mux4to1

Obzirom da imamo više različitih ulaza koji utiču na različita stanja naše mašine, odlučili smo pojednostaviti implementaciju korištenjem multiplexer-a. Ulaze W, LC, T i HM smo sveli na jedan izlaz M 4-to-1 multiplexer-a, gdje određeni ulazi, ovisno o trenutnom stanju, korespondiraju korespondiraju izlazu M. Tako primjerice u stanju 001 vidimo da izlaz M odgovara ulazu W (M=W), što u tekstu zadatka i jeste slučaj. Jednako tako važi i za sva druga stanja koja su definisana u našoj mašini, što je i vidljivo s Waveform dijagrama. Pošto nam stanje 000 ne ovisi niti od jednog ulaza multipleksera, već od ulaza B, proizvoljno smo mu dodijelili vrijednost timera T, da bismo lakše očitavali vrijednosti s dijagrama. Istu stvar smo uradili za stanja 110 i 111 (koja nam nisu definisana).

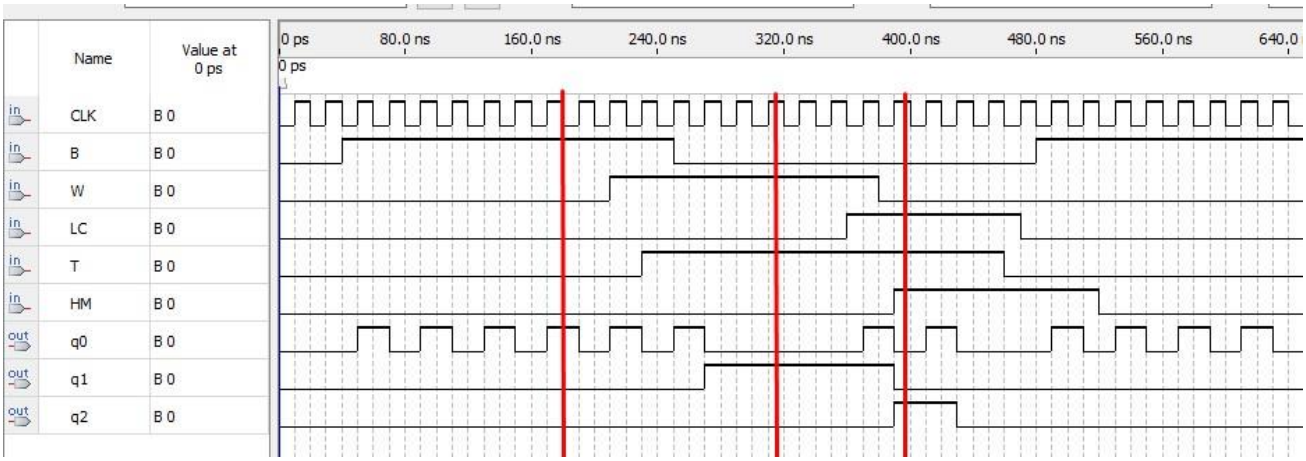
FSM shema i Waveform vremenski dijagram



Slika 6: Quartus shema naše FSM



Slika 7: Simbol naše FSM

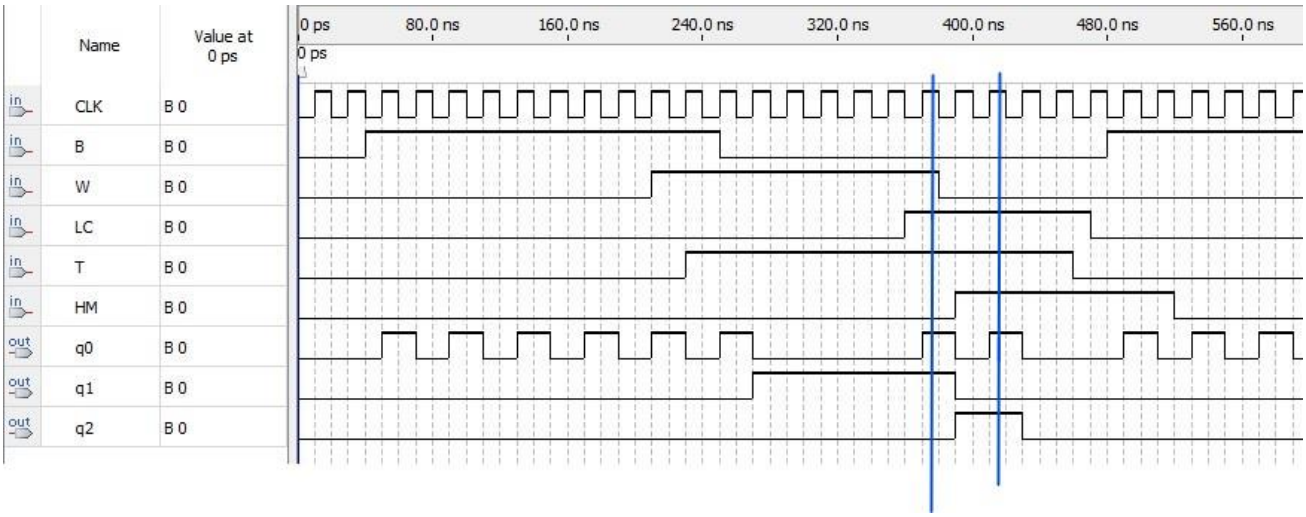


Slika 8: Waveform naše FSM (1)

Na slici 8 posmatrajmo prvu crvenu liniju. Jasno vidimo da je B=1, dok su ostali ulazi u stanjima logičke nule. Prvom crvenom linijom smo kao što vidimo, označili stanje 001 (q2q1q0). Jedini dovoljan uslov za prelazak u stanje 001, iz stanja 000 je pritisnut gumb B, što se i vidi sa slike. Ostali ulazi nisu važni za ovo stanje.

Drugom crvenom linijom je označeno stanje 010. Za dolazak u ovo stanje iz prethodnog, obraćamo pažnju na ulaz W. Kao što vidimo, isti je u stanju jedinice tako da se i ovaj slučaj poklapa s našom tabelom. Jednako tako, gumb B nije stisnut, tako da je dolazak u ovo stanje neometan.

Treća crvena linija predstavlja stanje 100. Dolazak u ovo stanje iz stanja 011 je uslovljen timer-om T. Dakle, čim timer signalizira kraj procesa, ide se u naredno stanje. Kao što vidimo na dijagramu, T je u stanju logičke jedinice dok nas drugi ulazi ne zanimaju. (izuzev gumba B koji je i ovdje u stanju nule – nije stisnut)



Slika 9: Waveform naše FSM (2)

Prvom plavom linijom smo označili stanje 011. Za dolazak u ovo stanje je potrebno da je poklopac zatvoren, tj da je senzor LC u stanju jedinice. Kao što na slici vidimo, to i jeste slučaj. Gumb B opet nije stisnut, tako da je prelazak u naredno stanje neometan. Ostali senzori za ovo stanje ne igraju ulogu.

Drugom plavom linijom je označeno posljednje stanje 101. Da bismo došli u ovo stanje, odnosno stanje pečenja, potrebno je da je tijesto nadošlo, to jeste da je senzor HM u stanju logičke jedinice. Na dijagramu vidimo da to jeste slučaj, kao i da gumb B prethodno nije pritisnut, tj. u stanju je nule. Ovim smo potvrdili funkcionisanje pekača, kako je definisano i u tabeli istinitosti.

Verilog modul za dekoder i opis algoritma rada display-a

U kodu ispod ćemo definisati ispis na 7-segmentnom display-u u određenim stanjima naše FSM.

```
module DEC_grupa3(q2, q1, q0, a, b, c, d, e, f, g);

input q2, q1, q0;

output reg a, b, c, d, e, f, g;

always@*

case({q2, q1, q0})

3'b000:

                begin

                        a=1;

                        b=1;

                        c=1;

                        d=1;

                        e=1;

                        f=1;

                        g=0;//(nula)

                end

3'b001:

                begin

                        a=1;
```

```
                b=0;
                c=1;
                d=1;
                e=0;
                f=1;
                g=1; //(petica)
            end
3'b010:
            begin
                a=0;
                b=1;
                c=1;
                d=0;
                e=0;
                f=1;
                g=1; //(cetvorka)
            end
3'b011:
            begin
                a=1;
                b=1;
                c=1;
                d=1;
                e=0;
                f=0;
                g=1;//(trica)
            end
 3'b100:
            begin
```

```
                    a=1;
                    b=1;
                    c=0;
                    d=1;
                    e=1;
                    f=0;
                    g=1; //(dvica)
                end
3'b101:
                begin
                    a=0;
                    b=1;
                    c=1;
                    d=0;
                    e=0;
                    f=0;
                    g=0; //(jedinica)
                end
        endcase
endmodule
```

Dakle, u kodu iznad smo odredili kako će izgledati naš 7-segmentni display za određena stanja u kojima se pekač nalazi u datom trenutku.

Kada se pekač nalazi u stanju 000, na 7seg display-u će biti ispisano „0“ , jer je to „nulto“ stanje naše mašine

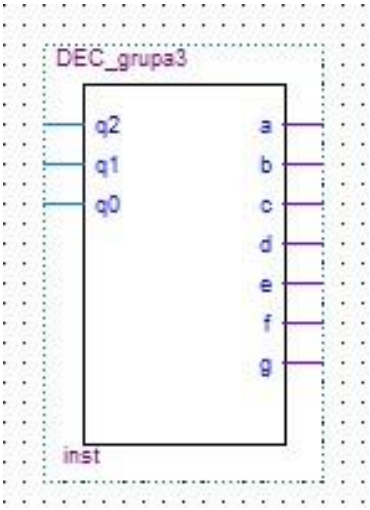
Stanje 001 našeg pekača će biti predstavljeno s „5“ na našem 7seg display-u, što znači da je do kraja ostalo još 5 koraka/etapa/stanja.

Jednako tako i za stanje 010, ispis na 7seg display-u će biti „4“ što znači da je do kraja ostalo još 4 koraka.

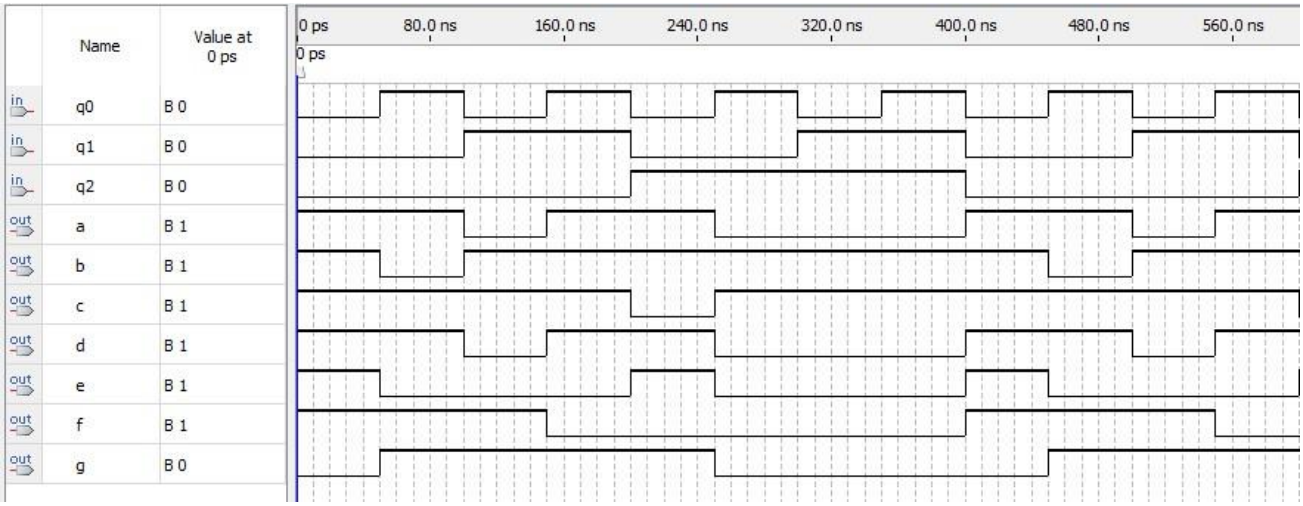
Stanje 011 će biti predstavljeno brojem „3“ na 7-seg display-u, dakle još 3 koraka do kraja.

Stanje 100 je popraćeno ispisom broja „2“ na display-u (2 koraka do kraja).

I na kraju, stanje 101 je predstavljeno ispisom broja „1“ na našem display-u što znači da je to posljednje stanje u kojem se naš pekač nalazi.



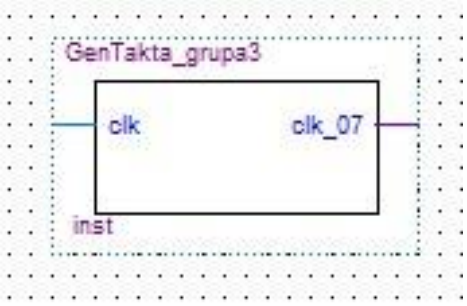
Slika 10: Simbol našeg dekodera



Slika 11: Waveform dijagram našeg dekodera

Verilog modul za generator takta

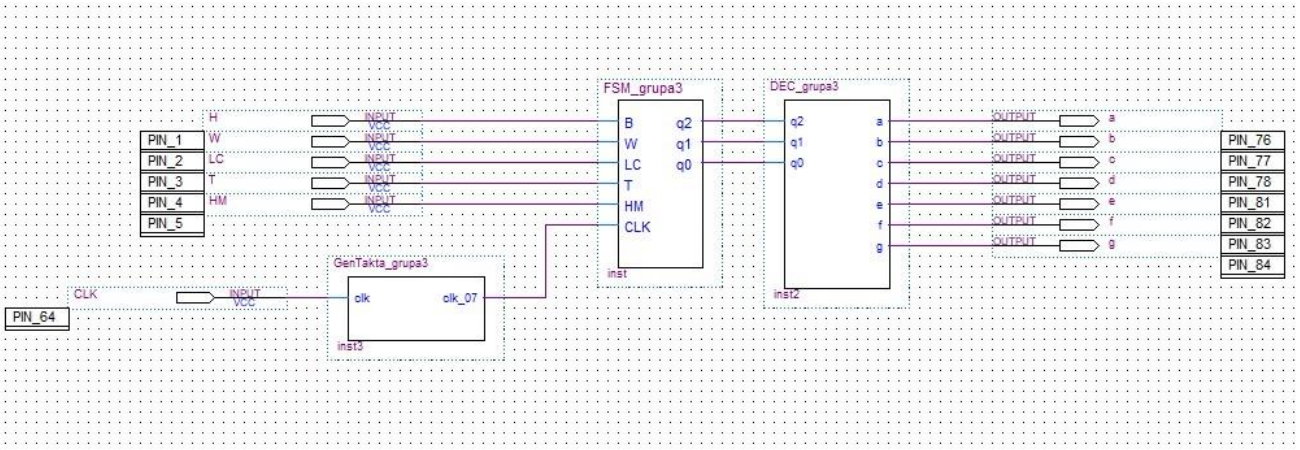
```
module GenTakta_grupa3(clk, clk_07);
input clk;
output reg clk_07;
reg [25:0] counter;

always@ (posedge clk)

begin

counter <= counter+1;

if (counter >= 26'b10000101100000111011000000)

 clk_07<=0;

else if(counter >= 26'b100001011000001110110000000)

 counter<=0;

else

 clk_07<=1;

 end
endmodule
```



Slika 12: Simbol našeg generatora takta

Generator takta koristimo da bismo interni clock reducirali na frekvecniju koja je vidljiva našem oku, tj. da se može primijetiti prelazak iz stanja u stanje. Mi ćemo ovaj generator takta spojiti na clock naše FSM, tako da smanjimo frekvenciju.

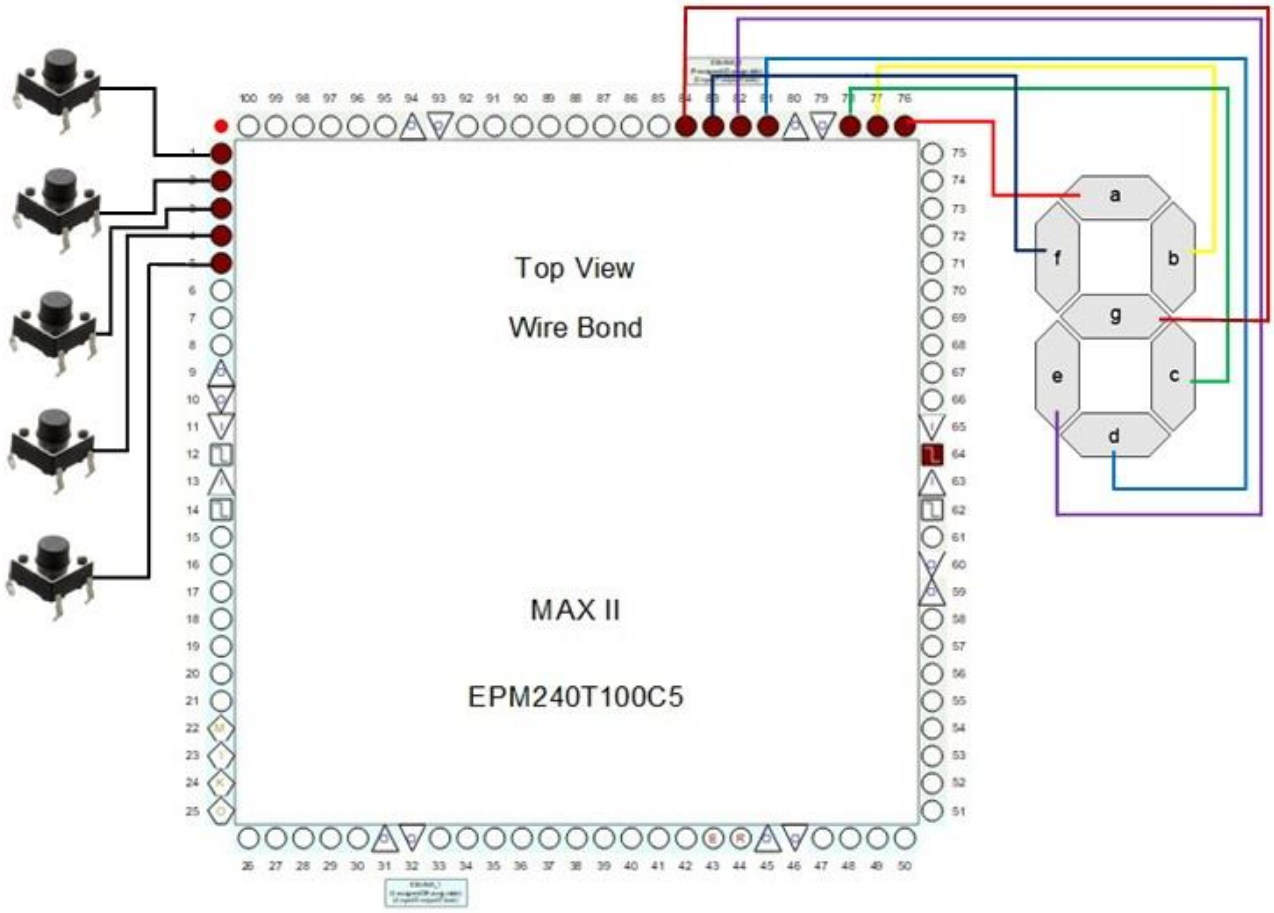
Blok dijagram



Slika 13: Blok dijagram / glavni schematic file

Sada smo od sva tri prethodno navedena file-a kreirali simbole koje smo implementirali u blok dijagram prikazan na slici. Kao što možemo vidjeti, ulaze B, W, LC, T, HM smo doveli na ulaz naše FSM. Na clock FSM smo doveli naš generator takta. Izlazi FSM predstavljaju ulaze u dekoder, iz kojeg zatim imamo 7 izlaza a, b, c, d, e, f, g za ispis na 7-segmentnom display-u. Napomena: zbog problema s kompajliranjem, na ovoj shemi i pin planner-u, promijenili smo ime ulaza B u H, jer već na shemi postoji pin s imenom b.

Shema povezivanja



Slika 13: Vizuelna reprezentacija implementacije

Node Name	Direction	Location	I/O Bank	Fitter Location	I/O Standard	Reserved	Current Strength
a	Output	PIN_76	2	PIN_15	3.3-V LV...default)		16mA (default)
b	Output	PIN_77	2	PIN_6	3.3-V LV...default)		16mA (default)
c	Output	PIN_78	2	PIN_12	3.3-V LV...default)		16mA (default)
CLK	Input	PIN_64	2		3.3-V LV...default)		16mA (default)
d	Output	PIN_81	2	PIN_17	3.3-V LV...default)		16mA (default)
e	Output	PIN_82	2	PIN_7	3.3-V LV...default)		16mA (default)
f	Output	PIN_83	2	PIN_95	3.3-V LV...default)		16mA (default)
g	Output	PIN_84	2	PIN_8	3.3-V LV...default)		16mA (default)
H	Input	PIN_1	2	PIN_97	3.3-V LV...default)		16mA (default)
HM	Input	PIN_5	1	PIN_90	3.3-V LV...default)		16mA (default)
LC	Input	PIN_3	1	PIN_36	3.3-V LV...default)		16mA (default)
T	Input	PIN_4	1	PIN_96	3.3-V LV...default)		16mA (default)
W	Input	PIN_2	1		3.3-V LV...default)		16mA (default)

Slika 14: Tabela raspodjele pinova na pin planner-u