

Univerzitet u Tuzli
Fakultet elektrotehnike

Projekat 1- Dokumentacija

Projektovanje sistema na čipu

Dezinfekcija paketa pomoću robotske ruke

Studenti:

Indir Karabegović

Nahid Delić

Muharem Muharemović

Profesor:

dr.sc. Lejla Banjanović-Mehmedović, vanr. prof

Tuzla, januar, 2022.

Varijable:

Globalna varijabla za state mašinu:

reg [5:0] state

Ulazi u sistem:

- input wire clk // omogućavanje takt impulsa
- input wire echo // varijabla u koju dolazi impuls sa ultrasoničnog senzora

Izlazi iz sistema:

- output wire gripper // hvataljka na robotskoj ruci (u kodu br. 1)
- output wire upWrist //gornji zglob na robotskoj ruci (u kodu br. 2)
- output wire downWrist //donji zglob na robotskoj ruci (u kodu br. 3)
- output wire armPlatform //motor koji rotira robotsku ruku (u kodu br. 4)
- output wire sprayingPlatform //motor koji rotira platoformu za dezinfekciju (u kodu br 5)
- output wire sprayingServoOne //motor koji vrši dezinfekciju (u kodu br 6)
- output wire sprayingServoTwo //motor koji vrši dezinfekciju (u kodu br 7)
- output wire trig // varijabla preko koje šaljem impuls u ultrasonični senzor

Pomoćni registri za motore:

- reg [20:0] pf_reg
- reg [20:0] pf_reg2
- reg [20:0] pulse_duration
- reg [20:0], pulse duration2
- reg gripper_bit
- reg upWrist_bit
- reg downWrist_bit
- reg armPlatform_bit
- reg sprayingPlatform_bit
- reg sprayingServoOne_bit
- reg sprayingServoTwo_bit

Pomoćni counter-i (brojači) za Ultrasonični senzor (HC-SR 04)

- reg [21:0] counterUltrasonic
- reg [18:0] echo_counter
- reg [19:0] delay_counter

Pomoćni registri i varijable za task moveServo()

- reg [31:0] counter
- reg [31:0] counterLimit
- reg [20:0] ugao;
- integer i
- integer bool

Pomoćni registri za task delay()

- reg [31:0] counterDelay
- reg [31:0] counterDelayLimit

Pomoćni registri za task sprayTask()

- reg [31:0] counterSpray

Taskovi:

- task gripperMove (input reg [20:0] ugao);
- task upWristMove (input reg [20:0] ugao);
- task downWristMove (input reg [20:0] ugao);
- task armPlatformMove (input reg [20:0] ugao);
- task sprayingPlatformMove (input reg [20:0] ugao);
- task sprayingServoOneMove (input reg [20:0] ugao);
- task sprayingServoTwoMove (input reg [20:0] ugao);
- task moveServo (input reg [20:0] startUgao, input reg [20:0] endUgao, reg [8:0] motor);
- task delay();
- task sprayTask();
- task Ultrasonic();

Objašnjenje taskova:

- **task gripperMove (input reg [20:0] ugao);**

Ovaj task je pomoćni task, odnosno, poziva se po potrebi iz taska moveServo(). Ulazni parametar taska je ugao, te je njegova uloga da pomjeri gripper (hvataljku) u odgovarajući položaj.

```
task gripperMove();  
input reg [20:0] ugao;  
begin  
    pulse_duration=ugao;  
    gripper_bit <= (pf_reg < pulse_duration) ?1:0;  
    pf_reg<=(pf_reg < 1000000)?(pf_reg+1):0;  
end  
endtask
```

Ostali pomoćni taskovi imaju skoro identičan kod, samo su druge varijable u pitanju, tako da kod navednih taskova nećemo ovdje pisati.

- **task upWristMove (input reg [20:0] ugao);**

Ovaj task je pomoćni task, odnosno, poziva se po potrebi iz taska moveServo(). Ulazni parametar taska je ugao, te je njegova uloga da pomjeri upWrist (gornji zglob na robotskoj ruci) u odgovarajući položaj.

- **task downWristMove (input reg [20:0] ugao);**

Ovaj task je pomoćni task, odnosno, poziva se po potrebi iz taska moveServo(). Ulazni parametar taska je ugao, te je njegova uloga da pomjeri downWrist (donj zglob na robotskoj ruci) u odgovarajući položaj.

- **task armPlatformMove (input reg [20:0] ugao);**

Ovaj task je pomoćni task, odnosno, poziva se po potrebi iz taska moveServo(). Ulazni parametar taska je ugao, te je njegova uloga da pomjeri armPlaform (motor koji rotira robotsku ruku) u odgovarajući položaj.

- **task sprayingPlatformMove (input reg [20:0] ugao);**

Ovaj task je pomoćni task, odnosno, poziva se po potrebi iz taska moveServo(). Ulazni parametar taska je ugao, te je njegova uloga da pomjeri sprayingPlaform (motor koji rotira platformu za dezinfekciju) u odgovarajući položaj.

- **task sprayingServoOneMove (input reg [20:0] ugao);**

Ovaj task je pomoćni task, odnosno, poziva se iz taska sprayTask() ili po potrebi iz taska moveServo(). Ulazni parametar taska je ugao, te je njegova uloga da pomjeri sprayingServoOne (motor koji vrši dezinfekciju) u odgovarajući položaj.

- **task sprayingServoTwoMove (input reg [20:0] ugao);**

Ovaj task je pomoćni task, odnosno, poziva se iz taska sprayTask() ili po potrebi iz taska moveServo(). Ulazni parametar taska je ugao, te je njegova uloga da pomjeri sprayingServoTwo (motor koji vrši dezinfekciju) u odgovarajući položaj.

- **task moveServo (input reg [20:0] startUgao, input reg [20:0] endUgao, reg [8:0] motor);**

Ovaj task je predviđen za pozivanje iz always bloka. Naime, uloga ovog taska jeste da kao ulazne parametre dobije polazni ugao, krajnji ugao te motor koji će se pomjerati. Pored toga, u ovom tasku smo uspjeli redukovati brzinu pomjeranja motora. Kod ovog taska je možda i najkompleksniji u ovom projektu, te ćemo ga prikazati i objasniti u narednih nekoliko rečenica:

```
task moveServo();
input reg [20:0] startUgao;
input reg [20:0] endUgao;
input reg [8:0] motor;
```

```

begin
//Kreiramo registre startPulse i endPulse
//25000 je referentna vrijednost, odnosno vrijednost za 0 stepeni
// startPulse registar nam predstavlja početni ugao ali u
pulsevima
// endPulse registar nam predstavlja kranji ugao ali također u
pulsevima
// Razlog zašto smo množili startni i kranji ugao sa 555 jeste sto
je to pulsna vrijednost za jedan ugao
// Odnosno, offset je [25000, 125000], tj.
// 25000 = 0 stepeni, 125000 = 180 stepeni
//  $x = (125000 - 25000) / 180 = (\text{približno}) 555$ 
reg [20:0] startPulse = 25000 + (555*startUgao);
reg [20:0] endPulse = 25000 + (555*endUgao);
// ugao je pomoćna varijabla za task, te je navedeni bool-an
korišten zbog toga što ugao samo u prvoj iteraciji mora biti
startPulse
if(bool==0)
begin
ugao=startPulse;
bool=1;
end
//Povećavamo counter, razlog zašto imamo counter i counterLimit
jeste brzina kretanja servo motora
// CounterLimit je 1 000 000 odakle slijedi
// Clock = 50 Mhz, odnosno za 1 sec 50 miliona takt impulsa, kada
uradimo 1000000 takst impulsa to je vrijeme od 20 ms (milisekundi)
counter=counter+1;
if(counter>counterLimit)
begin
//U zavisnosti koji motor smo proslijedili u task poziva se
odgovarajući pomoćni move task
case(motor)
1:gripperMove(ugao);

```

```

2:upWristMove(ugao);
3:downWristMove(ugao);
4:armPlatformMove(ugao);
5:sparyingPlatformMove(ugao);
6:sparyingServoOneMove(ugao);
7:sparyingServoTwoMove(ugao);
endcase
end
// 40 milisec
if(counter>32'd2000000)
begin
counter = 0;
//Smanjuj ugao za jedan stepen (stepen=555 pulseva)
if(startUgao>endUgao) /
begin
//ako je pocetni ugao veci od kranjnjeg
ugao=ugao-555;
end
else
begin
// ako je pocetni ugao manji od krajnjeg
ugao=ugao+555;
end
//Ako je puls manji od 25000 znači da je ugao 0, limit je
dostignut i ograniči kretanje
if(startUgao>endUgao)
begin
if(ugao<=endPulse)
begin
ugao=endPulse;
state = state+1;
bool=0;

```

```
end
```

```
end
```

```
else
```

```
begin
```

```
if(ugao>=endPulse)
```

```
begin
```

```
ugao=endPulse;
```

```
//kada smo došli do kranje pozicije, mijenjamo state i izlazimo iz  
taska
```

```
state = state+1;
```

```
bool=0;
```

```
end
```

```
end
```

```
end
```

```
end
```

```
endtask
```

- **task delay();**

Ovaj task je predviđen za pozivanje iz always bloka. Uloga ovog taska je jednostavna, naime, potrebno je samo da napravi vremensku zadržku od dvije sekunde. Navedena logika je implementirana pomoću counter-a. Task ne prihvata nikakve parametre.

```
reg [31:0] counterDelay = 0;
```

```
reg [31:0] counterDelayLimit = 100000000; //2sec
```

```
task delay();
```

```
begin
```

```
counterDelay = counterDelay + 1;
```

```
if(counterDelay>=counterDelayLimit)
```

```
begin
```

```
state = state + 1; //prelazimo u naredni state
```

```
counterDelay=0;
```

```
end
```

```
end
endtask
```

- **task sprayTask();**

Ovaj task je predviđen za pozivanje iz always bloka. Uloga ovog taska je da istovremeno pomjeri dva motora za dezinfekciju, odnosno, da pokrene prskalicu. Također, ni ovaj task ne prima nikakve parametre.

```
reg [31:0] counterSpray = 0;
task sprayTask();
begin
counterSpray = counterSpray + 1;

//Kada prođe 500ms (0.5s) uđi u ovaj if uslov, u suprotnom uđi u
else
//Odnosno, prvih 500ms kod ulazi u else statement I forsiranje
pomjeranje motora tako da se zateže opruga I vrši se prskane
//Narednih 500ms kod ulazi u prvi if I forsira vraćanje motora u
nulti položaj
//Nakon jedne sekunde završi dezinfekciju
if(counterSpray>25000000 && counterSpray<=50000000)
begin
sparyingServoOneMove(75000);
sparyingServoTwoMove(75000);
end
else if(counterSpray>50000000)
begin
state= state+1;
counterSpray=0;
end
else
begin
sparyingServoOneMove(25000);
sparyingServoTwoMove(125000);
```



```
end
end
endtask
```

task Ultrasonic();

Ovaj task je predviđen za pozivanje iz always bloka. Uloga ovog taska je da detektuje da li se objekat za dezinfekciju nalazi na početnoj poziciji, ukoliko se nalazi, state se promjeni i sistem počinje da radi, odnosno robotska ruka počinje sa postupkom dezinfekcije.

```
task ultraSonic();
begin
counterUltraSonic<=counterUltraSonic+1'b1;
// prvih 500 taktova se šalje trig signal, tačnije, da bismo
poslali signal moramo ga slati određeni broj takt impulsa, a ne
samo u jednom trenutku
if(counterUltraSonic<=500)
begin
trig_value <=1'b1;
end
else
begin
//Prestajemo slati trig signal
trig_value <= 1'b0;
//Očekujemo echo signal, te kada stigne, prelazimo u naredni state
if(echo && counterUltraSonic<700000)
begin
echo_counter<=echo_counter+1'b1;
if(echo_counter > 19'd294118 && echo_counter < 19'd300000)
//led_t <= 3'b111;
state = state+1;
//else if(echo_counter >= 19'd147059)
//led_t <= 3'b110;
end
else
```

```

led_t <= 3'b000;
end
else
if(echo_counter !== 19'd0)
begin
delay_counter <= delay_counter+1;
if(delay_counter >= 20'd100000000)
begin
counterUltraSonic <= 22'd0;
echo_counter <= 19'd0;
delay_counter <= 20'd0;
end
end
end
end
endtask

```