

# Laboratorijska vježba – Upravljanje motorima i ultrazvučnim senzorom na FPGA

## 1 Osnovni pojmovi

### PWM i upravljanje motorima

PWM (Pulse Width Modulation) je tehnika koja omogućava kontrolu snage isporučene električnom uređaju putem promjene širine impulsa u periodičnom signalu. Osnovna ideja je da se signal stalno prebacuje između stanja logičke jedinice (1) i nule (0), pri čemu je omjer vremena kada je signal u stanju 1 u odnosu na ukupni period poznat kao **duty cycle**.

Duty cycle se izražava u procentima:

$$D = \frac{t_{on}}{T} \times 100\% \quad (1)$$

Gdje je  $t_{on}$  vrijeme kada je signal u stanju logičke jedinice, a  $T$  je ukupan period signala.

U kontekstu upravljanja motorima, PWM se koristi za kontrolu brzine vrtnje (kod DC motora) ili pozicije (kod servo motora). Promjenom širine impulsa, mijenja se efektivna snaga koju motor prima.

Za standardne RC/servo motore, tipični period je oko 20 ms, dok širina korisnog impulsa varira između približno 1.0 ms i 2.0 ms. Manja širina impulsa uzrokuje okretanje u jedan smjer, a veća u suprotan.

## 2 Dio I – Upravljanje servo motorom pomoću FPGA

### Zadatak

Koristeći Verilog implementirati signal upravljanja servo motorom na jednom pinu FPGA ploče. Pritisom jedne od dvije tipke moguće je postići sljedeće:

- TIPKA 1: Okretanje u jednom smjeru (impuls širine 1.3 ms)
- TIPKA 2: Okretanje u suprotnom smjeru (impuls širine 1.7 ms)

Takt FPGA razvojne ploče je 50 MHz i koristi se kao referenca za generisanje vremena.

## Teorija i proračun

Period takta 50 MHz odgovara periodu od 20 ns. Da bi se generisala širina impulsa od npr. 1.3 ms potrebno je prebrojati odgovarajući broj taktova:

$$1.3 \text{ ms} = 1.3 \times 10^{-3} \text{ s} \rightarrow \frac{1.3 \times 10^{-3}}{20 \times 10^{-9}} = 65\,000 \text{ taktova} \quad (2)$$

Slično za 1.5 ms i 1.7 ms:

$$1.5 \text{ ms} \rightarrow 75\,000 \text{ taktova}, \quad 1.7 \text{ ms} \rightarrow 85\,000 \text{ taktova} \quad (3)$$

Cijeli period PWM-a (20 ms) uključuje dodatnu pauzu, pa se brojač resetuje nakon isteka perioda.

## Spajanje Parallax servo motora na FPGA

Parallax standard servo motor koristi tri žice:

- **Crvena** – napajanje (+5V)
- **Crna ili smeđa** – masa (GND)
- **Bijela ili narandžasta** – signal (PWM ulaz)

Da bi servo pravilno radio, mora imati zajedničku masu sa FPGA pločom (npr. Cyclone IV, DE0-Nano ili DE10-Lite). To znači da se GND serva i GND FPGA-a moraju spojiti zajedno.

Servo žica	Spoj na FPGA
Crvena (+5V)	Vanjski izvor 5V (ne sa FPGA!)
Crna (GND)	GND pin FPGA ploče
Bijela (Signal)	GPIO pin koji izlazi iz modula motor

Primjer povezivanja:

- **motor** izlaz iz modula povezan na GPIO\_PIN[12]
- **BTN1** i **BTN2** povezani na tastere na ploči (npr. KEY[0] i KEY[1])
- Servo napajan sa eksternog 5V izvora
- GND između FPGA i servo napajanja mora biti zajednički

## Verilog kod

```
// Upravljanje servo motorom
module servo (
    input wire clk,
    input wire BTN1,
    input wire BTN2,
    output wire motor
```

```

);  

reg [25:0] brojac = 0;  

reg [25:0] impuls = 0;  

reg izlaz = 0;  

always @ (posedge clk) begin  

    if (BTN1 == 1'b1)  

        impuls <= 25'd65000; // 1.3 ms  

    else if (BTN2 == 1'b1)  

        impuls <= 25'd85000; // 1.7 ms  

    if (brojac < impuls) begin  

        izlaz <= 1'b1;  

        brojac <= brojac + 1;  

    end  

    else if (brojac >= impuls && brojac < 25'd1000000) begin  

        izlaz <= 1'b0;  

        brojac <= brojac + 1;  

    end  

    else begin  

        brojac <= 0;  

    end  

end  

assign motor = izlaz;  

endmodule

```

## Detaljno objašnjenje koda

U modulu **servo** implementiran je osnovni PWM generator koji se koristi za pozicioniranje servo motora. Kod se izvršava sinhrono sa taktom FPGA ploče (50 MHz), što znači da se svaki **always** blok pokreće na pozitivan brid takta (**posedge clk**).

- **Registrar** `brojac` služi za brojanje taktova od početka periodičnog PWM ciklusa. Kada dostigne vrijednost koja odgovara trajanju kompletног perioda (20 ms 1 000 000 taktova), brojač se resetuje na nulu.
- **Registrar** `impuls` čuva ciljnu širinu impulsa, tj. broj taktova za koje će izlaz `motor` biti postavljen na logičku jedinicu. Vrijednost `impuls` se mijenja zavisno od pritisnute tipke:
  - `BTN1` postavlja širinu impulsa na 65 000 taktova (1.3 ms)
  - `BTN2` postavlja širinu impulsa na 85 000 taktova (1.7 ms)
- **Signal** `izlaz` određuje trenutno stanje PWM izlaza. Dokle god je `brojac < impuls`, izlaz je 1. Kada `brojac` pređe vrijednost `impuls`, izlaz se postavlja na 0 i traje do kraja perioda (pauza od 20 ms).
- Na kraju svakog ciklusa, brojač se resetuje, čime se započinje novi PWM period.

Ovaj princip je dovoljno precizan da omogući stabilno pozicioniranje većine RC serva (uključujući Parallax standard servo) bez dodatnog hardverskog PWM modula.

### 3 Dio II – Ultrazvučni senzor (HC-SR04)

Ultrazvučni senzor HC-SR04 koristi se za mjerjenje udaljenosti pomoću zvučnih talasa visoke frekvencije.

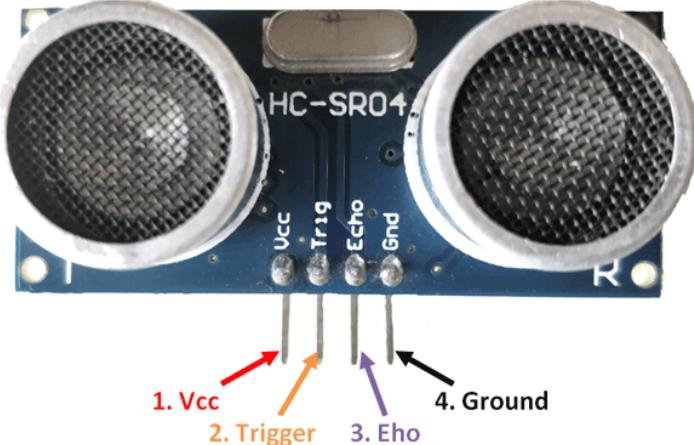


Figure 1: HC-SR04

Sastoji se od dva dijela:

- **Trig** – pin koji šalje kratki impuls od 10  $\mu$ s kako bi senzor emitovao ultrazvučni talas.
- **Echo** – pin koji vraća impuls čije trajanje odgovara vremenu potrebnom da talas dođe do prepreke i vrati se nazad.

Udaljenost se računa prema formuli:

$$d = \frac{t_{echo} \times v_{zvuka}}{2} \quad (4)$$

gdje je  $v_{zvuka} \approx 343$  m/s, a  $t_{echo}$  vrijeme trajanja povratnog impulsa.

## Verilog kod za HC-SR04

```
// Mjerenje udaljenosti pomocu HC-SR04
module ultrasonic (
    input wire clk,           // 50 MHz
    input wire echo,          // povratni signal
    output reg trig,          // impuls prema senzoru
    output reg [31:0] dist    // izracunata udaljenost (u taktnim
                             // ciklusima)
);
reg [31:0] brojac = 0;
reg [2:0] stanje = 0;

always @ (posedge clk) begin
    case(stanje)
        0: begin
            trig <= 1'b1;
            brojac <= brojac + 1;
            if (brojac >= 500) begin // 10 us na 50 MHz
                trig <= 1'b0;
                brojac <= 0;
                stanje <= 1;
            end
        end

        1: if (echo == 1'b1) begin
            stanje <= 2;
            brojac <= 0;
        end

        2: if (echo == 1'b0) begin
            dist <= brojac;
            stanje <= 0;
        end else
            brojac <= brojac + 1;
    endcase
end
endmodule
```

### Objašnjenje koda:

- **Ulazi i izlazi:** Modul ima ulazni signal `clk` (sistemska takt 50 MHz) i `echo` signal sa senzora, dok su izlazi `trig` (koji pokreće mjerjenje) i `dist` (rezultat u taktnim ciklusima).
- **Signalni i stanja:** Registri `brojac` i `stanje` služe za upravljanje radnim fazama:
  - `brojac` broji taktove (svaki takt = 20 ns).
  - `stanje` označava u kojoj je fazi sistem (generisanje impulsa, čekanje na eho, mjerjenje trajanja eha).
- **Stanje 0 – Generisanje TRIG impulsa:** Kada je `stanje = 0`, linija `trig` se postavlja na logičku jedinicu i ostaje visoka 500 taktičkih ciklusa. Pošto je period takta  $T = 1/50 \text{ MHz} = 20 \text{ ns}$ , impuls traje:

$$500 \times 20 \text{ ns} = 10 \mu\text{s}$$

što je minimalno vrijeme koje HC-SR04 zahtijeva za aktivaciju.

- **Stanje 1 – Čekanje početka ECHO impulsa:** Nakon što se TRIG signal spusti, sistem prelazi u stanje 1 i čeka da echo postane visok (1). To znači da je senzor poslao ultrazvuk i sada mjeri trajanje eha.
- **Stanje 2 – Mjerenje trajanja ECHO impulsa:** Dok je echo = 1, brojac se inkrementira svaku pozitivnu ivicu takta. Kada echo padne na 0, vrijeme trajanja se upisuje u dist. Vrijeme mjerenja može se pretvoriti u udaljenost:

$$d = \frac{t \cdot v_{zvuka}}{2}$$

gdje je  $t = \frac{\text{dist}}{50 \times 10^6}$ , a  $v_{zvuka} \approx 340 \text{ m/s}$ .

- **Ciklus se ponavlja:** Nakon završetka mjerenja, stanje se vraća na 0 i proces se ponavlja.

Vrijednost dist predstavlja vrijeme trajanja ECHO impulsa u taktnim ciklusima (na 50 MHz). Za pretvaranje u udaljenost (u cm), koristi se formula:

$$d \text{ (cm)} = \frac{dist \times 20 \times 10^{-9} \times 34300}{2} \quad (5)$$