

Laboratorijska vježba – Upravljanje motorima i ultrazvučnim senzorom na FPGA

1 Dio I – Upravljanje Parallax servo motorima

1.1 Cilj

Naučiti generisati PWM signale za dva Parallax continuous-rotation servo motora pomoću FPGA. Implementirati mogućnost rotacije unaprijed i nazad pritiskom na tipke (ili izborom smjera kroz ulazne signale). Motori su suprotno orijentirani (lijevi motor ima obrnutu interpretaciju širine impulsa u odnosu na desni).

1.2 Teorija

Standardni RC/servo signal ima period od oko 20 ms, a korisni impuls između 1.0 ms i 2.0 ms. Za takt od 50 MHz vrijedi:

- 1.3 ms = 65,000 taktova
- 1.5 ms = 75,000 taktova (stop)
- 1.7 ms = 85,000 taktova
- 20 ms period = 1,000,000 taktova

Za Parallax continuous-rotation servoe koristimo konvenciju:

- **Lijevi motor:** naprijed = 1.3 ms, nazad = 1.7 ms
- **Desni motor:** naprijed = 1.7 ms, nazad = 1.3 ms

1.3 Spoj i napajanje

- Servo crvena = eksterno +5 V (ne iz FPGA pinova)
- Servo crna = GND (poveži sa FPGA GND)
- Servo signal = FPGA GPIO (izlaz PWM)

1.4 Verilog implementacija

```
// Dio I: PWM za dva Parallax continuous-rotation servoa
module servo(
    input clk,
    output reg left_motor,
    output reg right_motor
);

    reg [31:0] counter;

    localparam integer LEFT_PULSE    = 65_000;      // ~1.7ms      rotacija
    naprijed
    localparam integer RIGHT_PULSE   = 85_000;      // ~1.3ms      rotacija
    nazad
    localparam integer PERIOD        = 1_000_000;    // 20 ms

    always @(posedge clk) begin
        if (counter >= PERIOD - 1)
            counter <= 0;
        else
            counter <= counter + 1;

        left_motor  <= (counter < LEFT_PULSE);
        right_motor <= (counter < RIGHT_PULSE);
    end
endmodule
```

2 Dio II – Line follower sa 2 IR senzora (QTI)

2.1 Cilj

Implementirati jednostavan line-follower koji koristi dva IR senzora (lijevi i desni). Senzori detektuju crnu liniju (logička 1) kada je ispod senzora.

2.2 Ponašanje

- L=1, R=1: na liniji – voziti pravo
- L=1, R=0: desni izgubio liniju – skrenuti desno
- L=0, R=1: lijevi izgubio liniju – skrenuti lijevo
- L=0, R=0: izgubljena linija – stop

2.3 Spoj i napajanje

- QTI senzori daju digitalni izlaz (0/1) na FPGA ulaze
- Pazi na signalni napon i filtriranje šuma

2.4 Verilog implementacija

```
// Dio III: Line follower - 2 IR senzora
module servo_line_follower (
    input clk,
    input left_sensor,
    input right_sensor,
    output reg left_motor,
    output reg right_motor
);

reg [31:0] counter;

localparam integer PERIOD = 1_000_000;
localparam integer STOP_PULSE = 75_000;
localparam integer FWD_PULSE = 65_000;
localparam integer REV_PULSE = 85_000;

reg [31:0] left_pulse;
reg [31:0] right_pulse;

always @(posedge clk) begin
    if (counter >= PERIOD - 1)
        counter <= 0;
    else
        counter <= counter + 1;

    left_motor <= (counter < left_pulse);
    right_motor <= (counter < right_pulse);
end

always @(*) begin
```

```

case ({left_sensor, right_sensor})
2'b00: begin // bijelo - idi pravo
    left_pulse = FWD_PULSE;
    right_pulse = FWD_PULSE;
end
2'b10: begin // lijevi vidi crno - skreni lijevo
    left_pulse = STOP_PULSE;
    right_pulse = FWD_PULSE;
end
2'b01: begin // desni vidi crno - skreni desno
    left_pulse = FWD_PULSE;
    right_pulse = STOP_PULSE;
end
2'b11: begin // oba crna - stop
    left_pulse = STOP_PULSE;
    right_pulse = STOP_PULSE;
end
endcase
end
endmodule

```

3 Dio III – Ultrazvučni senzor HC-SR04

3.1 Cilj

Implementirati modul koji generiše TRIG impuls, mjeri trajanje ECHO impulsa i vraća udaljenost u centimetrima. Kada se robot približi prepreći na manje od 10 cm – motori se zaustavljaju.

3.2 Spoj

- TRIG pin: FPGA izlaz
- ECHO pin: FPGA ulaz (preko naponskog djelitelja ili level shiftera)
- Napajanje senzora: 5 V

3.3 Verilog implementacija

```
// Dio III: HC-SR04 ultrazvučni senzor - mjerjenje udaljenosti
module servo_ultrasonic (
    input clk,
    input echo,
    output reg trig,
    output reg left_motor,
    output reg right_motor
);

localparam integer PERIOD = 1_000_000;
localparam integer STOP_PULSE = 75_000;
localparam integer FWD_PULSE = 65_000;
localparam integer STOP_DISTANCE = 10;
localparam real CM_TO_TICKS = 58.8;

reg [31:0] counter_pwm;
reg [31:0] left_pulse;
reg [31:0] right_pulse;
reg [31:0] echo_time;
reg [31:0] distance_cm;
reg [31:0] trig_counter;
reg measuring;
reg echo_prev;

always @(posedge clk) begin
    if (trig_counter >= 3_000_000)
        trig_counter <= 0;
    else
        trig_counter <= trig_counter + 1;

    trig <= (trig_counter < 500) ? 1'b1 : 1'b0;
end

always @(posedge clk) begin
    echo_prev <= echo;

    if (echo && !echo_prev) begin
        measuring <= 1;
```

```

    echo_time <= 0;
end
else if (!echo && echo_prev) begin
    measuring <= 0;
    distance_cm <= echo_time / CM_TO_TICKS;
end
else if (measuring) begin
    echo_time <= echo_time + 1;
end
end
end

always @(posedge clk) begin
if (counter_pwm >= PERIOD - 1)
    counter_pwm <= 0;
else
    counter_pwm <= counter_pwm + 1;

left_motor <= (counter_pwm < left_pulse);
right_motor <= (counter_pwm < right_pulse);
end

always @(*) begin
if (distance_cm < STOP_DISTANCE && distance_cm > 0) begin
    left_pulse = STOP_PULSE;
    right_pulse = STOP_PULSE;
end else begin
    left_pulse = FWD_PULSE;
    right_pulse = FWD_PULSE;
end
end
endmodule

```

3.4 Pretvaranje u udaljenost

Udaljenost u centimetrima (približno):

$$d(\text{cm}) \approx \frac{t_{\mu s} \times 0.0343}{2}$$

gdje je $t_{\mu s}$ vrijeme trajanja ECHO impulsa u mikrosekundama.

3.5 Integracija s motorima

Ako se modul integrira s kontrolom motora:

- Ako $d < 10$ cm \rightarrow STOP (1.5 ms)
- Inače \rightarrow normalno kretanje (naprijed ili po liniji)