



## پروژه پایانی

### Ping Pong in VHDL

درس طراحی سیستم های دیجیتال

نگین حقیقی – 99521226

هلیا وفایی – 99522347

ستاره باباجانی – 99521109

آنیتا تلخابی – 99522284

استاد درس: مریم خدادادی

نیم سال دوم 1401-1402

## توضیح پروژه:

در این بازی دو پدل و یک توپ داشته و توپ با سرعت و جهتی از قبل تعریف شده، شروع به حرکت میکند. همچنین پدل ها نیز با سرعتی تعیین شده و قابل تغییر، به بالا و پایین حرکت میکنند. در صورت برخورد توپ به پدل ها، توپ بازتاب کرده و بازی ادامه میابد اما در صورت برخورد به دیواره های راست و چپ، بازی خاتمه پیدا میکند.

## توضیح کد:

در ابتدا یک موجودیت به نام Models تعریف میکنیم که شامل قسمت های زیر میباشد:

(1) Generic: در این بخش اندازه توپ و پدل، طول و عرض صفحه نمایش، سرعت حرکت

توپ و پدل ها و مختصات پدل ها در محور X تعریف میشوند.

(2) Port: در این بخش ورودی های clk و reset (برای شروع مجدد بازی) و start (به معنای

شروع حرکت توپ و بازی) و خروجی های مختصات Y پدل ها و مختصات X و Y توپ

تعریف میشود.

```
entity Models is
  generic (
    BALL_SIZE : positive := 8; -- Size of the ball
    PADDLE_SIZE : positive := 32; -- Size of the paddle
    SCREEN_WIDTH : positive := 240; -- Width of the screen
    SCREEN_HEIGHT : positive := 180; -- Height of the screen
    BALL_SPEED : positive := 5; -- Speed of the ball
    PADDLE_SPEED : integer := 3; -- Speed of the paddle
    PADDLE1_X : positive := 1;
    PADDLE2_X : positive := 239
  );
  port (
    clk : in std_logic; -- Clock signal
    reset : in std_logic; -- Reset signal
    start : in std_logic; -- Start the ball movement
    paddle1_y : out integer; -- Paddle1 Y position
    paddle2_y : out integer; -- Paddle2 Y position
    ball_x : out integer; -- Ball X position
    ball_y : out integer; -- Ball Y position
  );
end entity Models;
```

در architecture مربوط به Models، یک process ای تعریف کرده که حساس به clk و reset است. در این process، variableهایی برای مختصات توپ و پدل و جهت حرکت آنها تعریف میشود. که در محور x عدد 1 و 0 به معنای حرکت به سمت راست و چپ هستند. و در محور yها، عدد 1 و 0 به ترتیب به معنای حرکت به سمت بالا و پایین میباشند. همچنین جهت حرکت اولیه توپ به سمت شمال شرق و یکی از پدل ها به سمت بالا و دیگری به پایین شروع به حرکت میکنند.

```
architecture Behavioral of Models is
begin
    process(clk, reset)
    variable ball_x_pos : integer; -- Internal signal for ball X position
    variable ball_y_pos : integer; -- Internal signal for ball Y position
    variable paddle1_y_pos : integer; -- Internal signal for paddle1 Y position
    variable paddle2_y_pos : integer; -- Internal signal for paddle2 Y position
    variable ball_x_dir : std_logic := '1'; -- Direction of ball movement along the X-axis
    variable ball_y_dir : std_logic := '1'; -- Direction of ball movement along the Y-axis
    variable paddle1_y_dir : std_logic := '1'; -- Direction of paddle1 movement along the Y-axis
    variable paddle2_y_dir : std_logic := '0'; -- Direction of paddle2 movement along the Y-axis
    begin
```

در صورت reset شدن بازی، باید توپ و پدلها به موقعیت اولیه خود بازمی گردند.

```
    if (reset = '1') then -- Reset the ball position
        ball_x_pos := SCREEN_WIDTH/2;
        ball_y_pos := SCREEN_HEIGHT/2;
        paddle1_y_pos := SCREEN_WIDTH/2;
        paddle2_y_pos := SCREEN_WIDTH/2;
        ball_x_dir := '1';
        ball_y_dir := '1';
        paddle1_y_dir := '1';
        paddle2_y_dir := '0';
```

در صورت start شدن بازی، شروط زیر بررسی میشوند:

- 1) تغییر مختصات توپ: اگر جهت حرکت توپ به سمت بالا و یا راست بود، مختصات آن به علاوه سرعت توپ، و در غیر اینصورت منهای سرعت توپ میشود.
- 2) تغییر مختصات پدل ها: اگر جهت حرکت هریک از پدلها به سمت بالا بود، مختصات آن به اضافه سرعت پدل و در غیر اینصورت منهای آن میشود.

```

-- Update ball X position based on direction
if (ball_x_dir = '1') then
    ball_x_pos := ball_x_pos + BALL_SPEED;
else
    ball_x_pos := ball_x_pos - BALL_SPEED;
end if;

-- Update ball Y position based on direction
if (ball_y_dir = '1') then
    ball_y_pos := ball_y_pos + BALL_SPEED;
else
    ball_y_pos := ball_y_pos - BALL_SPEED;
end if;

-- Update paddle1 Y position based on direction
if (paddle1_y_dir = '1') then
    paddle1_y_pos := paddle1_y_pos + PADDLE_SPEED;
else
    paddle1_y_pos := paddle1_y_pos - PADDLE_SPEED;
end if;

-- Update paddle2 Y position based on direction
if (paddle2_y_dir = '1') then
    paddle2_y_pos := paddle2_y_pos + PADDLE_SPEED;
else
    paddle2_y_pos := paddle2_y_pos - PADDLE_SPEED;
end if;

```

(3) برخورد توپ با پدل: اگر توپ با پدلی برخورد کند، در خلاف جهت قبلی خود، به حرکت ادامه میدهد.

```

-- Check for collision with paddles (adjust direction if needed)
--paddle1
if (ball_y_pos >= paddle1_y_pos and ball_y_pos <= paddle1_y_pos + PADDLE_SIZE and ball_x_pos = 1) then
    ball_x_dir := '1'; -- Reverse direction
end if;
--paddle2
if (ball_y_pos >= paddle2_y_pos and ball_y_pos <= paddle2_y_pos + BALL_SIZE and ball_x_pos = SCREEN_WIDTH - 1) then
    ball_x_dir := '0'; -- Reverse direction
end if;

```

(4) برخورد با سقف یا کف صفحه نمایش: اگر توپ با سقف یا کف برخورد کند، در خلاف جهت قبلی خود، به حرکت ادامه میدهد.

(5) برخورد توپ با دیواره های راست یا چپ: اگر توپ به دیواره های راست یا چپ برخورد کند، بازی اتمام یافته و موقعیت توپ و پدل ها به حالت اولیه خود باز میگردد.

```

-- Check for collision with top or bottom screen boundary (reverse direction if needed)
if (ball_y_pos <= 0 or ball_y_pos >= SCREEN_HEIGHT - BALL_SIZE) then
    ball_y_dir := not ball_y_dir; -- Reverse direction
end if;

-- Check for collision with left and right screen boundary (reset position and direction)
if (ball_x_pos <= 0 or ball_x_pos >= SCREEN_WIDTH - 1) then
    ball_x_pos := SCREEN_WIDTH/2; -- Reset position
    ball_y_pos := SCREEN_HEIGHT/2; -- Reset position
    paddle1_y_pos := SCREEN_WIDTH/2;
    paddle2_y_pos := SCREEN_WIDTH/2;
    ball_x_dir := '1'; -- Reset direction
    ball_y_dir := '1'; -- Reset direction
    paddle1_y_dir := '1';
    paddle2_y_dir := '0';
end if;

```

6) برخورد پدل با سقف و کف صفحه نمایش: اگر پدلی به سقف یا کف برخورد کند، در خلاف جهت حرکت قبلی خود، به حرکتش ادامه میدهد.

```

-- Check for collision paddle1 with top and bottom screen boundary (reset position and direction)
if (paddle1_y_pos <= 0 or paddle1_y_pos >= SCREEN_HEIGHT - 1) then
    paddle1_y_dir := not paddle1_y_dir;
end if;

-- Check for collision paddle2 with top and bottom screen boundary (reset position and direction)
if (paddle2_y_pos <= 0 or paddle2_y_pos >= SCREEN_HEIGHT - 1) then
    paddle2_y_dir := not paddle2_y_dir;
end if;

```

پس از اجرای تست بنچ، خروجی کد به صورت زیر میباشد، که در آن دو حالت برخورد توپ به سقف و باخت بازی (برخورد توپ به دیواره راست) رخ داده است:

