



دانشگاه صنعتی نوشیروانی بابل

درس: طراحی سیستم‌های دیجیتال fpga / asic

استاد درس: دکتر بالغی

Keypad

مریم لیاقت

کد:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity KeypadDisplay is
    Port(
        clk      : in  STD_LOGIC;
        col : in  STD_LOGIC_VECTOR (3 downto 0);
        rows : out STD_LOGIC_VECTOR (3 downto 0);
        seg: out STD_LOGIC_VECTOR (6 downto 0)
    );
end KeypadDisplay;

architecture Logic of KeypadDisplay is
    signal active_row : INTEGER range 0 to 3 := 0;
    signal scan_counter : INTEGER := 0;
    signal key : INTEGER range 0 to 15 := 0;
begin
    process(clk)
    begin
        if rising_edge(clk) then
            scan_counter <= scan_counter + 1;
            if scan_counter = 50000 then
                scan_counter <= 0;
                active_row <= (active_row + 1) mod 4;
            end if;
        end if;
    end process;
```

```
    process(active_row)
    begin
        case active_row is
            when 0 => rows <= "1110";
            when 1 => rows <= "1101";
            when 2 => rows <= "1011";
            when 3 => rows <= "0111";
            when others => rows <= "1111";
        end case;
    end process;

    process(clk)
    begin
        if rising_edge(clk) then
            case active_row is
                when 0 =>
                    if col = "1110" then key <= 0;
                    elsif col = "1101" then key <= 1;
                    elsif col = "1011" then key <= 2;
                    elsif col = "0111" then key <= 3;
                    end if;
                when 1 =>
                    if col = "1110" then key <= 4;
                    elsif col = "1101" then key <= 5;
                    elsif col = "1011" then key <= 6;
                    elsif col = "0111" then key <= 7;
                    end if;
                when 2 =>
                    if col = "1110" then key <= 8;
                    elsif col = "1101" then key <= 9;
                    elsif col = "1011" then key <= 10;
                    elsif col = "0111" then key <= 11;
                    end if;
                when 3 =>
                    if col = "1110" then key <= 12;
                    elsif col = "1101" then key <= 13;
                    elsif col = "1011" then key <= 14;
                    elsif col = "0111" then key <= 15;
                    end if;
                when others => "0000";
            end case;
        end if;
    end process;
```

```

process(key)
begin
    case key is
        when 0 => seg <= "1111110";
        when 1 => seg <= "0110000";
        when 2 => seg <= "1101101";
        when 3 => seg <= "1111001";
        when 4 => seg <= "0110011";
        when 5 => seg <= "1011011";
        when 6 => seg <= "1011111";
        when 7 => seg <= "1110000";
        when 8 => seg <= "1111111";
        when 9 => seg <= "1111011";
        when 10 => seg <= "1110111";
        when 11 => seg <= "0011111";
        when 12 => seg <= "1001110";
        when 13 => seg <= "0111101";
        when 14 => seg <= "1001111";
        when 15 => seg <= "1000111";
        when others => seg <= "0000000";
    end case;
end process;

```

شرح بخش‌های مختلف کد:

۱. تعریف

library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

entity KeypadDisplay is

Port(

clk : in STD_LOGIC;

col : in STD_LOGIC_VECTOR (3 downto 0);

rows : out STD_LOGIC_VECTOR (3 downto 0);

seg : out STD_LOGIC_VECTOR (6 downto 0)

);

end KeypadDisplay;

architecture Logic of KeypadDisplay is

signal active_row : INTEGER range 0 to 3 := 0;

signal scan_counter : INTEGER := 0;

signal key : INTEGER range 0 to 15 := 0;

clk: سیگنال کلاک ورودی.

col: ورودی ۴ بیتی از ستون‌های کیپد.

rows: خروجی ۴ بیتی برای تحریک سطرهای کیپد.

seg: خروجی ۷ بیتی برای کنترل سون‌سگمنت.

active_row: سطر از کیپد که در حال حاضر اسکن می‌شود.

scan_counter: شمارنده‌ای برای کند کردن سرعت اسکن سطرها.

key: عدد خوانده‌شده از کیپد (مقدار کلید فشرده شده به صورت عددی).

۲. اسکن سطرهای کیپد

```
process(clk)
begin
    if rising_edge(clk) then
        scan_counter <= scan_counter + 1;
        if scan_counter = 50000 then
            scan_counter <= 0;
            active_row <= (active_row + 1) mod 4;
        end if;
    end if;
end process;
```

این فرآیند با هر پالس کلاک اجرا می‌شود.

وقتی شمارنده به ۵۰۰۰۰ رسید، مقدار آن صفر شده و سطر بعدی از کیپد برای اسکن انتخاب می‌شود.

```

process(active_row)

begin

    case active_row is

        when 0 => rows <= "1110";

        when 1 => rows <= "1101";

        when 2 => rows <= "1011";

        when 3 => rows <= "0111";

        when others => rows <= "1111";

    end case;

end process;

```

بسته به اینکه کدام سطر فعال است، تنها یکی از بیت‌های rows صفر می‌شود (فعال منطقی).

۳. تشخیص کلید فشرده‌شده

```

process(clk)

begin

    if rising_edge(clk) then

        case active_row is

            when 0 =>

                if col = "1110" then key <= 0;

                elsif col = "1101" then key <= 1;

                ...

```

این فرآیند بررسی می‌کند که در زمان فعال بودن هر سطر، آیا یکی از ستون‌ها نیز صفر شده‌اند یا نه.

ترکیب سطر فعال و ستون صفر، نشان‌دهنده کلیدی است که فشرده شده.

```
process(key)

begin

  case key is

    when 0 => seg <= "1111110"; -- نمایش ۰

    when 1 => seg <= "0110000"; -- نمایش ۱

    ...

    when 15 => seg <= "1000111"; -- نمایش F

    when others => seg <= "0000000";

  end case;

end process;
```

بسته به مقدار متغیر key ، الگوی مناسب برای سون سگمنت تولید می شود.

مقدار seg یک بردار ۷ بیتی است که هر بیت آن یک قسمت از سون سگمنت را کنترل می کند.

```

NET "clk" LOC = P80 ;

NET "col[0]" LOC = P27 | PULLUP ;
NET "col[1]" LOC = P28 | PULLUP ;
NET "col[2]" LOC = P29 | PULLUP ;
NET "col[3]" LOC = P31 | PULLUP ;

NET "row[0]" LOC = P26 ;
NET "row[1]" LOC = P24 ;
NET "row[2]" LOC = P22 ;
NET "row[3]" LOC = P21 ;

NET "seg[0]" LOC = P130 ;
NET "seg[1]" LOC = P131 ;
NET "seg[2]" LOC = P139 ;
NET "seg[3]" LOC = P138 ;
NET "seg[4]" LOC = P137 ;
NET "seg[5]" LOC = P133 ;
NET "seg[6]" LOC = P132 ;

```