-- Company:

-- Engineer:

--

-- Create Date: 14:49:31 10/16/2024

-- Design Name:

-- Module Name: keypad – Behavioral

-- Project Name:

-- Target Devices:

-- Tool versions:

-- Description:

--

-- Dependencies:

--

-- Revision:

-- Revision 0.01 – File Created

-- Additional Comments:

--

Library IEEE;

Use IEEE.STD\_LOGIC\_1164.ALL;

Use IEEE.STD\_LOGIC\_ARITH.ALL;

Use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

Entity keypad is

Port ( clk : in STD\_LOGIC;

Rst : in STD\_LOGIC;

Row : out STD\_LOGIC\_VECTOR (3 downto 0);

Col : in STD\_LOGIC\_VECTOR (3 downto 0);

Key\_code : out STD\_LOGIC\_VECTOR (3 downto 0);

Key\_valid : out STD\_LOGIC);

End keypad;

Architecture Behavioral of keypad is

Signal state : integer := 0;

Signal key\_detected : STD\_LOGIC\_VECTOR(3 downto 0) := (others => ‘0’);

Begin

Process(clk, rst)

Begin

If rst = ‘1’ then

Row <= “1111”; -- Disable all rows

Key\_code <= (others => ‘0’);

Key\_valid <= ‘0’;

State <= 0;

Elsif rising\_edge(clk) then

Case state is

When 0 =>

Row <= “1110”; -- Enable row 0

If col(0) = ‘0’ then key\_code <= “0000”; key\_valid <= ‘1’; state <= 1; end if;

If col(1) = ‘0’ then key\_code <= “0001”; key\_valid <= ‘1’; state <= 1; end if;

If col(2) = ‘0’ then key\_code <= “0010”; key\_valid <= ‘1’; state <= 1; end if;

If col(3) = ‘0’ then key\_code <= “0011”; key\_valid <= ‘1’; state <= 1; end if;

When 1 =>

Row <= “1101”; -- Enable row 1

If col(0) = ‘0’ then key\_code <= “0100”; key\_valid <= ‘1’; state <= 2; end if;

If col(1) = ‘0’ then key\_code <= “0101”; key\_valid <= ‘1’; state <= 2; end if;

If col(2) = ‘0’ then key\_code <= “0110”; key\_valid <= ‘1’; state <= 2; end if;

If col(3) = ‘0’ then key\_code <= “0111”; key\_valid <= ‘1’; state <= 2; end if;

When 2 =>

Row <= “1011”; -- Enable row 2

If col(0) = ‘0’ then key\_code <= “1000”; key\_valid <= ‘1’; state <= 3; end if;

If col(1) = ‘0’ then key\_code <= “1001”; key\_valid <= ‘1’; state <= 3; end if;

If col(2) = ‘0’ then key\_code <= “1010”; key\_valid <= ‘1’; state <= 3; end if;

If col(3) = ‘0’ then key\_code <= “1011”; key\_valid <= ‘1’; state <= 3; end if;

When 3 =>

Row <= “0111”; -- Enable row 3

If col(0) = ‘0’ then key\_code <= “1100”; key\_valid <= ‘1’; state <= 0; end if;

If col(1) = ‘0’ then key\_code <= “1101”; key\_valid <= ‘1’; state <= 0; end if;

If col(2) = ‘0’ then key\_code <= “1110”; key\_valid <= ‘1’; state <= 0; end if;

If col(3) = ‘0’ then key\_code <= “1111”; key\_valid <= ‘1’; state <= 0; end if;

When others =>

State <= 0; -- Reset state

End case;

End if;

End process;

End Behavioral;