

# Decoder

## Generics

- yfg6yugiukj	<type> integer : hola 2	<value>
- U	<type> integer : hola	<value> 32
- N	<type> integer : prueba	<value>
- T	<type> integer : prueba	<value>
- Gad	<type> std_logic_vector : preuba3	<value> N-1

## Ports

- clk	<type> input : alfa	<value> std_logic
- rst	<type> input : alfa	<value> std_logic
- yhmm	<type> input : alfa	<value> std_logic
- Rowerererrerererere	<type> input : dfhdfg	<value> std_logic_vector <bits> (N-1) downto (N-2)
- Col	<type> output : dfhdfg1	<value> std_logic_vector <bits> 3 downto 0
- DecodeOut	<type> output : marcafa	<value> std_logic_vector <bits> 3 downto 0

## Source

```

-----
-- Company: Digilent Inc 2011
-- Engineer: Michelle Yu
-- Create Date: 17:18:24 08/23/2011
--
-- Module Name: Decoder - Behavioral
-- Project Name: PmodKYPD
-- Target Devices: Nexys3
-- Tool versions: Xilinx ISE 13.2
-- Description:
-- This file defines a component Decoder for the demo project PmodKYPD.
-- The Decoder scans each column by asserting a low to the pin corresponding to the column
-- at 1KHz. After a column is asserted low, each row pin is checked.
-- When a row pin is detected to be low, the key that was pressed could be determined.
--
-- Revision:
-- Revision 0.01 - File Created
-----

```

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

---dgfgfsgedg
--fghfghfcgh

entity Decoder is --dfhdfg

    generic (
        yfg6yugiukj : integer;  -- hola 2
        U : integer := 32 --hola
        -- comentario
        N, T : integer;  -- prueba
        Gad : std_logic_vector := N-1  -- preuba3
    )
    Port (
        clk, rst, yhmm : in  STD_LOGIC := 32;  --alfa
        --34234523
        Rowerererrerererere : in  STD_LOGIC_VECTOR ((N-1) downto (N-2)) := 324; --dfhdfg
        Col : out  STD_LOGIC_VECTOR (3 downto 0);--dfhdfg1
        DecodeOut : out  STD_LOGIC_VECTOR (3 downto 0);--marcafa
    end;

    architecture Behavioral of Decoder is

        signal sclk :STD_LOGIC_VECTOR(19 downto 0);
    begin
        process(clk)
        begin
            if clk'event and clk = '1' then
                -- 1ms
                if sclk = "00011000011010100000" then
                    --C1
                    Col<= "0111";
                    sclk <= sclk+1;
                    -- check row pins
                elsif sclk = "00011000011010101000" then
                    --R1
                    if Row = "0111" then
                        DecodeOut <= "0001";  --1
                    --R2
                    elsif Row = "1011" then
                        DecodeOut <= "0100"; --4
                    --R3
                    elsif Row = "1101" then
                        DecodeOut <= "0111"; --7
                    --R4

```

```

    elsif Row = "1110" then
        DecodeOut <= "0000"; --0
    end if;
    sclk <= sclk+1;
-- 2ms
elsif sclk = "00110000110101000000" then
    --C2
    Col<= "1011";
    sclk <= sclk+1;
-- check row pins
elsif sclk = "00110000110101001000" then
    --R1
    if Row = "0111" then
        DecodeOut <= "0010"; --2
    --R2
    elsif Row = "1011" then
        DecodeOut <= "0101"; --5
    --R3
    elsif Row = "1101" then
        DecodeOut <= "1000"; --8
    --R4
    elsif Row = "1110" then
        DecodeOut <= "1111"; --F
    end if;
    sclk <= sclk+1;
--3ms
elsif sclk = "01001001001111100000" then
    --C3
    Col<= "1101";
    sclk <= sclk+1;
-- check row pins
elsif sclk = "01001001001111101000" then
    --R1
    if Row = "0111" then
        DecodeOut <= "0011"; --3
    --R2
    elsif Row = "1011" then
        DecodeOut <= "0110"; --6
    --R3
    elsif Row = "1101" then
        DecodeOut <= "1001"; --9
    --R4
    elsif Row = "1110" then
        DecodeOut <= "1110"; --E
    end if;
    sclk <= sclk+1;
--4ms
elsif sclk = "01100001101010000000" then
    --C4

```

```
Col<= "1110";
sclk <= sclk+1;
-- check row pins
elsif sclk = "01100001101010001000" then
  --R1
  if Row = "0111" then
    DecodeOut <= "1010"; --A
  --R2
  elsif Row = "1011" then
    DecodeOut <= "1011"; --B
  --R3
  elsif Row = "1101" then
    DecodeOut <= "1100"; --C
  --R4
  elsif Row = "1110" then
    DecodeOut <= "1101"; --D
  end if;
  sclk <= "00000000000000000000";
else
  sclk <= sclk+1;
end if;
end if;
end process;
```

```
end Behavioral;
```