

Laboratório 4: Keypad with Debounce:

Professor: Felipe Calliari | Monitor: Cristiano Nascimento

Aluno: Pedro Gabriel Serodio Sales e Thiago Levis

Matrícula: 2211911 e 1812899

1 Introdução

O objetivo deste relatório é apresentar as soluções para o laboratório 4 da disciplina ENG1448 assim como apresentações do laboratório em vídeo.

2 Resolução

1.1 : Display de 7 Segmentos - Simulação



(1)

Figura 1: Simulação do Display de 7 Segmentos (hex0 e hex1)

Para fazer o teste do display de 7 segmentos criamos um testbench em que varia-se as entradas hex0 e hex1 do componente. A cada ciclo da saída an é exibido na saída sseg (7 segmentos) o dígito referente hex0, enquanto an tem valor lógico 0, e o dígito referente a hex1, quando an tem valor lógico 1.

1.2 Display de 7 Segmentos - Código

O display mostra um dígito de cada vez, logo, para visualizar os dois ao mesmo tempo, implementou-se um contador usado como base para alternar rapidamente qual dígito fica aceso. Além disso, há uma correspondência entre os inputs hexadecimais de 4 bits cada e a saída de 7 bits para cada dígito.

```
with hex
select sseg(6 downto 0) <=
"1111110" when "0000", -- 0
"0110000" when "0001", -- 1
"1101101" when "0010", -- 2
"1111001" when "0011", -- 3
"0110011" when "0100", -- 4
"1011011" when "0101", -- 5
"1011111" when "0110", -- 6
"1110000" when "0111", -- 7
"1111111" when "1000", -- 8
"1111011" when "1001", -- 9
"1110111" when "1010", -- A
"0011111" when "1011", -- b
"1001110" when "1100", -- C
"0111101" when "1101", -- d
"1001111" when "1110", -- E
"1000111" when "1111", -- F
"0110111" when others; -- H
-- decimal point
```

(2)

Figura 2: Inputs Hex e saídas 7 Bits pra cada dígito

2 Varredura do Teclado

A varredura atribui nível lógico 1 em uma coluna de cada vez, e verifica se alguma das linhas está em nível lógico 0 para detectar uma tecla pressionada. Esse processo foi implementado através de uma máquina de estado, que fica presa em um dos estados de leitura coluna enquanto uma tecla estiver pressionada.

3 Implementação na plaquinha

Criou-se um módulo principal que importa os componentes de teclado e display definidos anteriormente. Nele foi definida a lógica de colocar o dígito anterior à esquerda, baseada no sinal tick definido no componente do teclado. A forma como está montada a placa, está representada no vídeo no final do relatório. Definiu-se o arquivo de constraints e os componentes foram conectados nas seguintes portas:

```
1 NET "clk" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
2 NET "CLK" PERIOD = 20.0ns HIGH 40%;
3 NET "reset" LOC = "V16" | IOSTANDARD = LVTTTL | PULLDOWN ;
4
5 # Colunas
6 NET "cols<0>" LOC = "F11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
7 NET "cols<1>" LOC = "E12" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
8 NET "cols<2>" LOC = "A13" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
9 NET "cols<3>" LOC = "A14" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
10 # Linhas
11 NET "rows<0>" LOC = "E11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
12 NET "rows<1>" LOC = "F12" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
13 NET "rows<2>" LOC = "B13" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
14 NET "rows<3>" LOC = "B14" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
15 # LED's
16 NET "sseg<6>" LOC = "B4" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
17 NET "sseg<5>" LOC = "D5" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
18 NET "sseg<4>" LOC = "A6" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
19 NET "sseg<3>" LOC = "E7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
20 NET "sseg<2>" LOC = "D7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
21 NET "sseg<1>" LOC = "F8" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
22 NET "sseg<0>" LOC = "F9" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
23
24 NET "an" LOC = "D11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;|
```

(3)

Figura 3: Constraint File

4 Integração dos códigos e implementação na placa

O código funcionou como o esperado, começando em 00 e jogando cada dígito anterior para a esquerda. O reset foi definido como o botão do Rotary e está funcional. Se a mesma tecla é pressionada seguidamente também é detectada.

Implementação foi bem sucedida com vídeo de apresentação no final do relatório. O código final também será enviado junto ao relatório.

É importante verificar a posição dos encaixes no vídeo abaixo para reprodução igual pois se for colocado em entradas diferentes na placa do que foi colocado, haverá de mudar as terminações no constraint file.

3 Vídeo de Apresentação

O vídeo de apresentação segue ao lado: **Lab 4: Keypad with Debounce**
ou acessar pelo url: **<https://youtu.be/q7-MuYPTSPE>**