

Laboratório 5: Keyboard:

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 5 da disciplina ENG1448 assim como apresentações do laboratório em vídeo.

2 Resolução

1 : Funcionamento do teclado

Considerando as informações apresentadas no enunciado, incluindo as simplificações de paridade e rápida exibição do código de controle para cada uma das teclas, implementamos a leitura do teclado por meio de uma máquina de estados com três estados: idle, rx-kbd-data e send-key.

No estado idle a máquina de estados aguarda até que a entrada data assuma o valor lógico 0, indicando o recebimento de um novo bloco de dados. A partir disso, na transição positiva do clock associado à entrada PS/2 ocorre uma transição de estado para rx-kbd-data.

Para este estado foi implementado um contador para verificar quantos bits foram lidos. Como o teclado envia os dados começando pelo bit menos significativo, o bit de data vai sendo concatenado à esquerda de um sinal auxiliar key-reg para construir o valor da tecla pressionada. Este sinal é então ligado à saída do módulo de leitura do teclado.

Por fim, quando totaliza-se 8 bit lidos, ocorre uma transição para o estado send-key, que transiciona de volta para o estado idle, ignorando o bit de paridade e assumindo que o valor lógico da entrada data volta a ser 1.

Com a definição da máquina de estados, conecta-se a mesma ao componente contendo 2 displays de 7 segmentos, já implementado no laboratório anterior. Essa conexão se dá ligando os 4 bits mais significativos da saída key da máquina de estados à entrada hex1, enquanto que os outros 4 bits da saída são ligados à entrada hex0. Dessa forma, para cada tecla pressionada vê-se rapidamente um piscar no display, referente ao código de controle e em seguida estabiliza-se a exibição do código correspondente à tecla pressionada.

```

    signal state : fsm_t := idle;

begin
    key <= key_reg;

    sync_proc: process(clk_kbd, reset)
    begin
        if (reset = '1') then
            key_reg <= "00000000";

        elsif (rising_edge(clk_kbd)) then
            case state is
                when idle =>
                    if (data = '0') then
                        state <= rx_kbd_data;
                    end if;

                    when rx_kbd_data =>
                        if(counter_reg(3) /= '1') then
                            key_reg <= data & key_reg(7 downto 1);
                            counter_reg <= counter_reg + 1;
                        else
                            counter_reg <= "0000";
                            state <= send_key;
                        end if;

                    when send_key =>
                        state <= idle;

                    when others =>
                        state <= idle;
                    end case;
                end if;
            end process sync_proc;
        end Behavioral;

```

(1)

Figura 1: Funcionamento do Teclado

2 Implementação na placa

É enviado um caractere hexadecimal para cada dígito do display. Definiu-se o arquivo de constraints e os componentes foram conectados nas seguintes portas:

```
# Clock and reset
NET "CLK" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
NET "CLK" PERIOD = 20.0ns HIGH 50%;

NET "reset" LOC = "V16" | IOSTANDARD = LVTTTL | PULLDOWN ;

# Output LEDs
NET "sseg<6>" LOC = "B4" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "sseg<5>" LOC = "D5" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "sseg<4>" LOC = "A6" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "sseg<3>" LOC = "E7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "sseg<2>" LOC = "D7" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "sseg<1>" LOC = "F8" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;
NET "sseg<0>" LOC = "F9" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;

NET "an" LOC = "D11" | IOSTANDARD = LVCMOS33 | SLEW = FAST | DRIVE = 8 ;

# PS2 Keyboard
NET "CLK_KBD" LOC = "G14" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;
NET "DATA" LOC = "G13" | IOSTANDARD = LVCMOS33 | DRIVE = 8 | SLEW = SLOW ;
NET "CLK_KBD" CLOCK_DEDICATED_ROUTE = FALSE;
```

(2)

Figura 2: Constraint File

O código funcionou como o esperado, mostrando 00 no display quando é iniciado. Quando uma tecla é apertada, após mostrar rapidamente o código de controle, o display aponta o código em hexadecimal de cada tecla, como visto no vídeo na próxima seção. Este comportamento está de acordo com o descrito no user guide da placa, que ilustra a correspondência de cada código com as teclas:

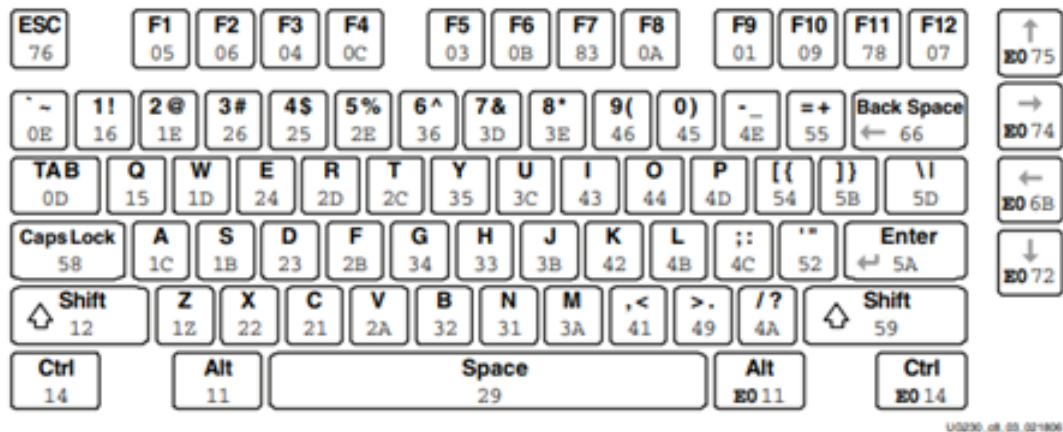


Figure 8-3: PS/2 Keyboard Scan Codes

(3)

Figura 3: PS/2 Keyboard Scan Codes

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 5: Keyboard**

ou acessar pelo url: <https://youtu.be/01Pjgqwuxw8>