

# PCS3635 – Laboratório Digital I

## Relatório do Experimento 5

### Jogo do tempo de reação



Turma 3 – Bancada 3

Professores:

Paulo Sergio Cugnasca

Edson Midorikawa

Integrantes:

Arthur Pires da Fonseca – 10773096

Lucas Lopes de Paula Junior - 9344880

2 de fevereiro de 2020

## INTRODUÇÃO

Nesta experiência, será projetado um sistema digital que atua como um jogo de medida de tempo de reação de uma pessoa.

O objetivo do jogo é simples, o jogador deve acionar o sinal de início (“jogar”) e esperar pelo estímulo de saída para então poder acionar o sinal de resposta.

Após essa sequência de eventos, se o jogador não tiver acionado a resposta antes de o estímulo de saída ser ligado pelo circuito, a saída “tempo” deverá apresentar qual foi o tempo de reação do jogador, em milissegundos.

Para implementar esse projeto, foram usados uma placa FPGA (Altera Cyclone V 5CEBA4F23C7N), um gerador de sinais (Analog Discovery) e os *softwares* respectivos a cada um desses componentes, que são o Intel Quartus Prime e o Waveforms.

## 1. PARTE EXPERIMENTAL

### 1.1. Atividade 1 – Projeto do Jogo do Tempo de Reação

O primeiro passo para o desenvolvimento do jogo foi elaborar um pseudocódigo, o qual resume o funcionamento do circuito. Em complemento, gerou-se um diagrama de transição de estados e um diagrama de blocos para modelarem, respectivamente, a unidade de controle e o fluxo de dados do circuito. Os resultados estão a seguir.

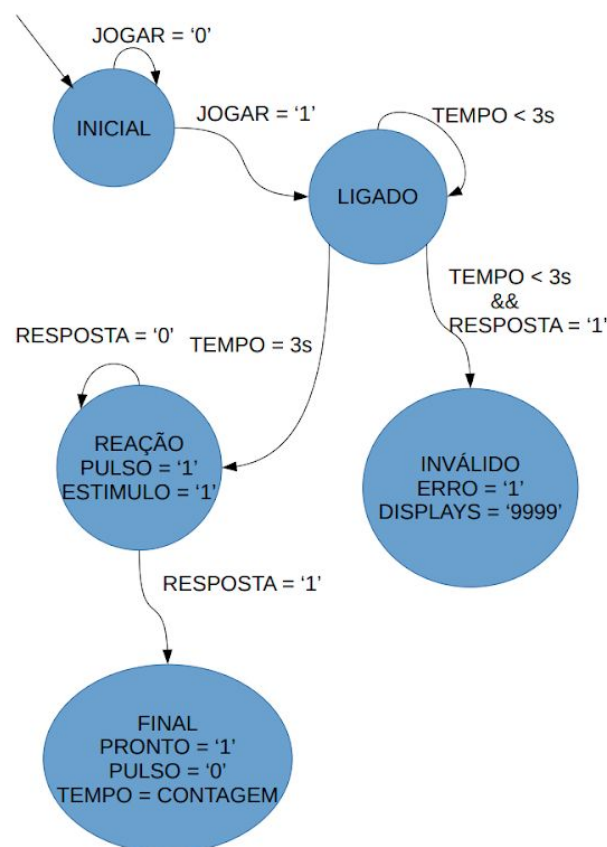


Figura 1 - Máquina de estados do Jogo do Tempo da Reação

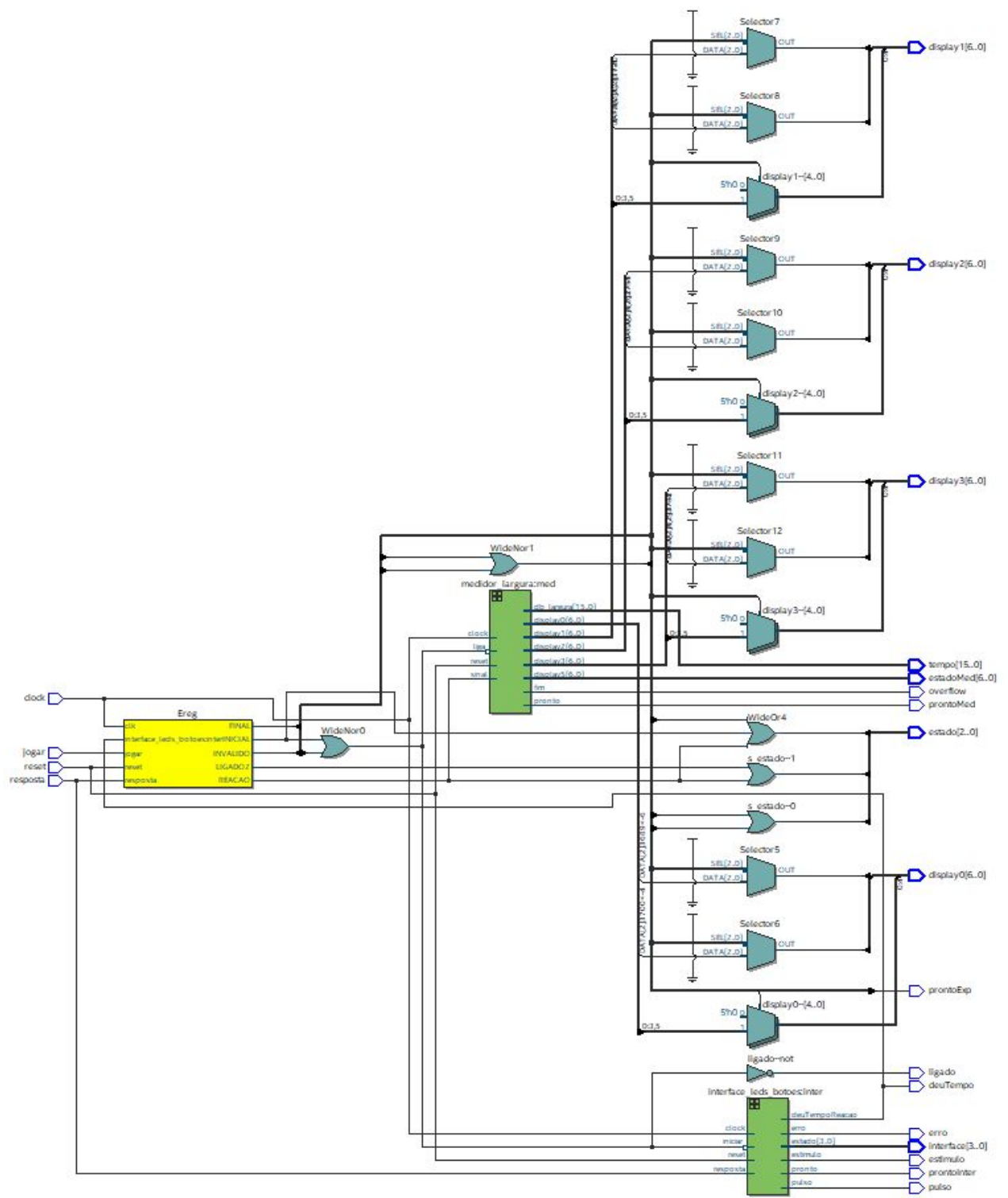


Figura 2 - Diagrama RTL do circuito gerado pelo Quartus

```

Algoritmo: jogo
entradas: jogar, resposta, reset, clock
saídas: ligado, pulso, estímulo, erro, pronto, tempo
{
    while (verdadeiro) {
        espera acionamento do sinal JOGAR
        LIGADO = 1
        while(tempo < 3s) {
            if (resposta = '1')
                ATIVAR ERRO
                AMOSTRAR 9999
        }
        PULSO = 1
        ESTIMULO = 1
        while (not resposta)
            conta
        TEMPO <= CONTAGEM
        PRONTO = 1
        PULSO = 0
    }
}

```

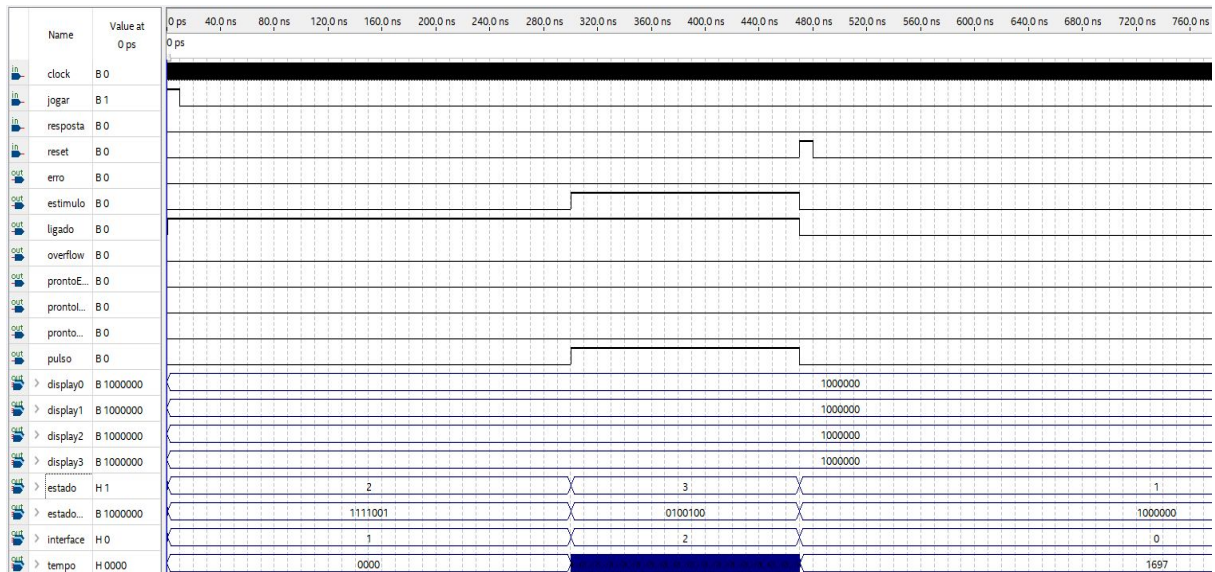
**Figura 3 - Pseudocódigo do Jogo do Tempo da Reação**

Foi criado em seguida um projeto no software Intel Quartus Prime, cujo nome é exp5-T3BB3.qar, onde todos os componentes do circuito foram descritos em VHDL e com sinais que convergem na entidade final do jogo, homônima ao nome do projeto.

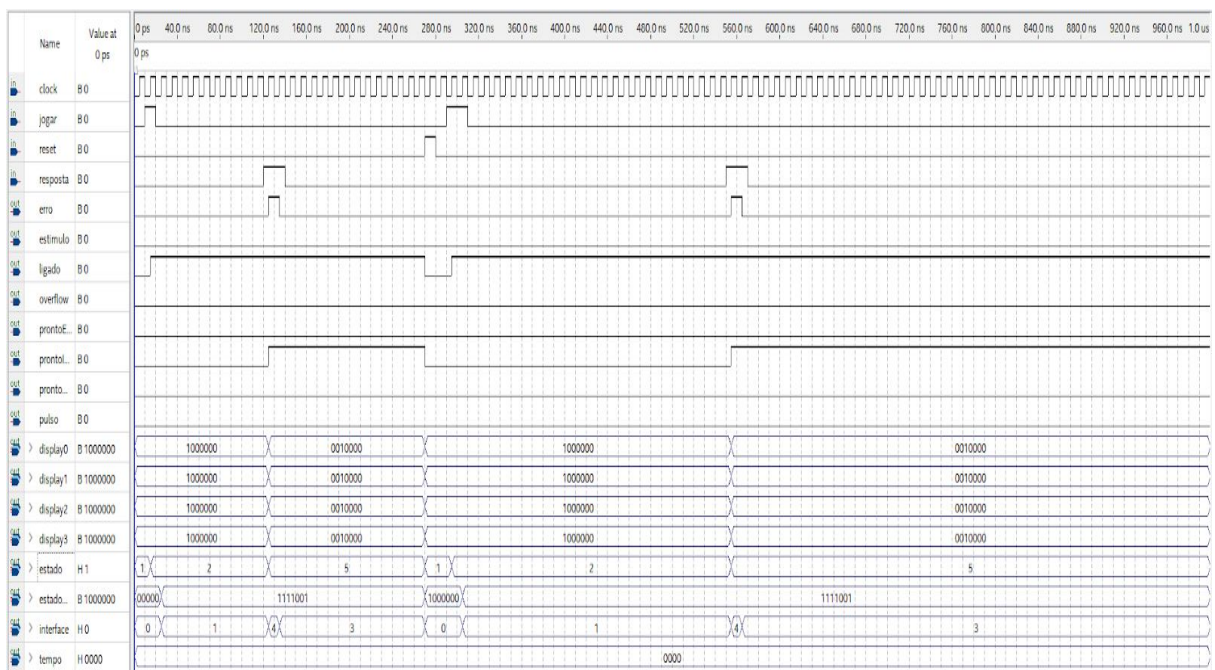
Um plano de testes foi elaborado para verificar o funcionamento do circuito, testando as duas situações básicas possíveis: uma jogada válida e uma inválida.

Para verificar uma jogada válida, basta acionar o sinal "jogar" e esperar por 3 segundos, visualizar o sinal de estímulo e então acionar a resposta. Uma jogada inválida precisa acionar essa resposta durante o tempo de espera de 3 segundos. No segundo caso, diferentemente do primeiro, deve-se observar o sinal de erro em alto na saída da entidade.

A entidade usada para o circuito do projeto foi testada no Quartus seguindo o plano de testes elaborado anteriormente como base. Os resultados foram os seguintes:



**Figura 4 - Teste de funcionamento normal do contador (jogada válida)**



**Figura 5 -Teste de jogada inválida, seguido de reset**

Após a descrição do *hardware* e os testes feitos no Quartus, foi feita a designação de quais entradas e saídas seriam ligadas a que elementos da placa FPGA e cabos do gerador de sinais; ela encontra-se a seguir

| Designação |             |           |             | Analog Discovery |            |       | Ligação adicional |
|------------|-------------|-----------|-------------|------------------|------------|-------|-------------------|
| sinal      | pino        | nome      | instrumento | função           | sinal      |       |                   |
| entradas   | clock       | GPIO_0_D0 | PIN_N16     | Patterns         | clock      | DIO0  | -                 |
|            | reset       | GPIO_0_D1 | PIN_B16     | Static I/O       | button 0/1 | DIO1  | -                 |
|            | jogar       | GPIO_0_D3 | PIN_C16     | Static I/O       | button 0/1 | DIO2  | -                 |
|            | resposta    | GPIO_0_D5 | PIN_K20     | Static I/O       | button 0/1 | DIO7  | -                 |
| saídas     | ligado      | led LEDR0 | PIN_AA2     |                  | -          | -     | -                 |
|            | pulso       | GPIO_0_31 | PIN_T20     | Scope            | -          | C1    | osciloscópio      |
|            | estímulo    | GPIO_0_33 | PIN_T18     | Static I/O       | LED        | DIO15 | -                 |
|            | erro        | GPIO_0_35 | PIN_T15     | Static I/O       | LED        | DIO14 | -                 |
|            | prontoExp   | led LEDR9 | PIN_L1      | -                | -          | -     | -                 |
|            | display0[0] | HEX00     | PIN_U21     | -                | -          | -     | -                 |
|            | display0[1] | HEX01     | PIN_V21     | -                | -          | -     | -                 |
|            | display0[2] | HEX02     | PIN_W22     | -                | -          | -     | -                 |
|            | display0[3] | HEX03     | PIN_W21     | -                | -          | -     | -                 |
|            | display0[4] | HEX04     | PIN_Y22     | -                | -          | -     | -                 |
|            | display0[5] | HEX05     | PIN_Y21     | -                | -          | -     | -                 |
|            | display0[6] | HEX06     | PIN_AA22    | -                | -          | -     | -                 |
|            | display1[0] | HEX10     | PIN_AA20    | -                | -          | -     | -                 |
|            | display1[1] | HEX11     | PIN_AB20    | -                | -          | -     | -                 |
|            | display1[2] | HEX12     | PIN_AA19    | -                | -          | -     | -                 |
|            | display1[3] | HEX13     | PIN_AA18    | -                | -          | -     | -                 |
|            | display1[4] | HEX14     | PIN_AB18    | -                | -          | -     | -                 |
|            | display1[5] | HEX15     | PIN_AA17    | -                | -          | -     | -                 |
|            | display1[6] | HEX16     | PIN_U22     | -                | -          | -     | -                 |
|            | display2[0] | HEX20     | PIN_Y19     | -                | -          | -     | -                 |
|            | display2[1] | HEX21     | PIN_AB17    | -                | -          | -     | -                 |

|             |           |          |            |     |       |   |
|-------------|-----------|----------|------------|-----|-------|---|
| display2[2] | HEX22     | PIN_AA10 | -          | -   | -     | - |
| display2[3] | HEX23     | PIN_Y14  | -          | -   | -     | - |
| display2[4] | HEX24     | PIN_V14  | -          | -   | -     | - |
| display2[5] | HEX25     | PIN_AB22 | -          | -   | -     | - |
| display2[6] | HEX26     | PIN_AB21 | -          | -   | -     | - |
| display3[0] | HEX30     | PIN_Y16  | -          | -   | -     | - |
| display3[1] | HEX31     | PIN_W16  | -          | -   | -     | - |
| display3[2] | HEX32     | PIN_Y17  | -          | -   | -     | - |
| display3[3] | HEX33     | PIN_V16  | -          | -   | -     | - |
| display3[4] | HEX34     | PIN_U17  | -          | -   | -     | - |
| display3[5] | HEX35     | PIN_V18  | -          | -   | -     | - |
| display3[6] | HEX36     | PIN_V19  | -          | -   | -     | - |
| overflow    | GPIO_0_D2 | PIN_M16  | Static I/O | LED | DIO13 |   |
| deuTempo    | GPIO_0_D4 | PIN_D17  | Static I/O | LED | DIO12 |   |

**Tabela 1 - Pinagem adotada para o projeto**



## 1.2. Atividade 2 - Implementação do Jogo do Tempo de Reação

Nesta seção, a placa FPGA foi programada no Quartus usando a opção *Pin Planner* e seguindo a designação planejada.

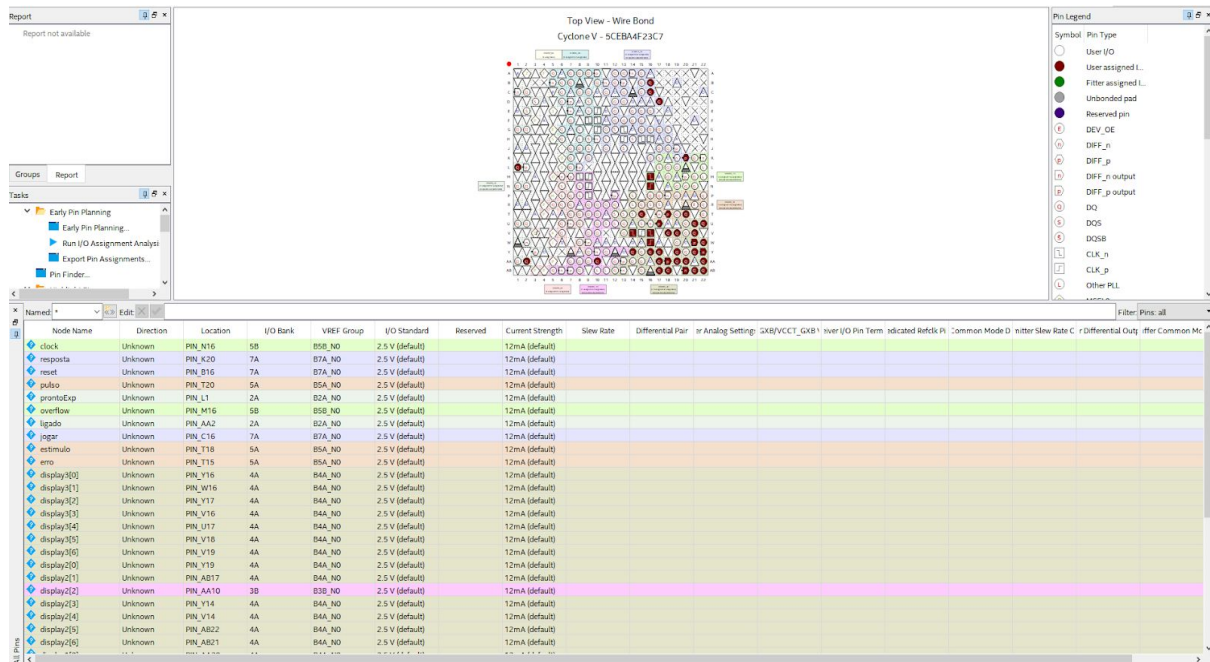


Figura 6 - Pinagem programada no Quartus

O Analog Discovery também foi usado nesse momento, sendo programado para gerar um sinal de *clock* de 1kHz e para comportar os sinais de entrada, os quais são acionados pelo jogador através dos botões presentes no Waveforms.

O plano de testes foi executado e produziu os seguintes resultados:

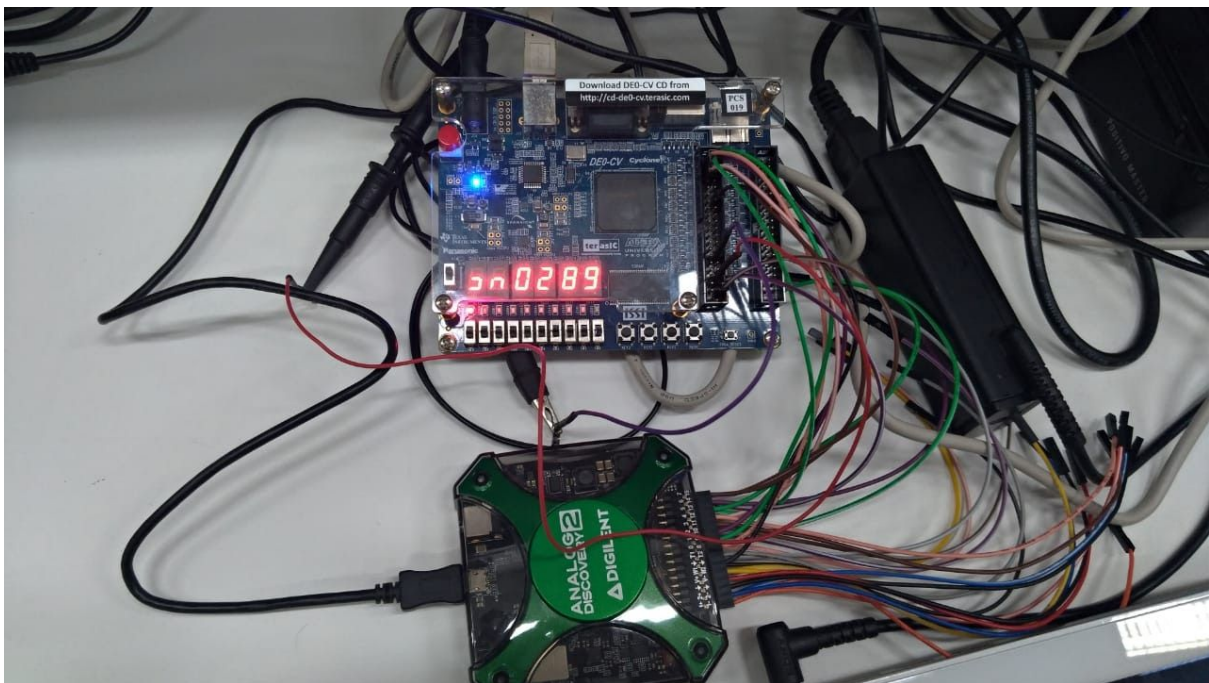
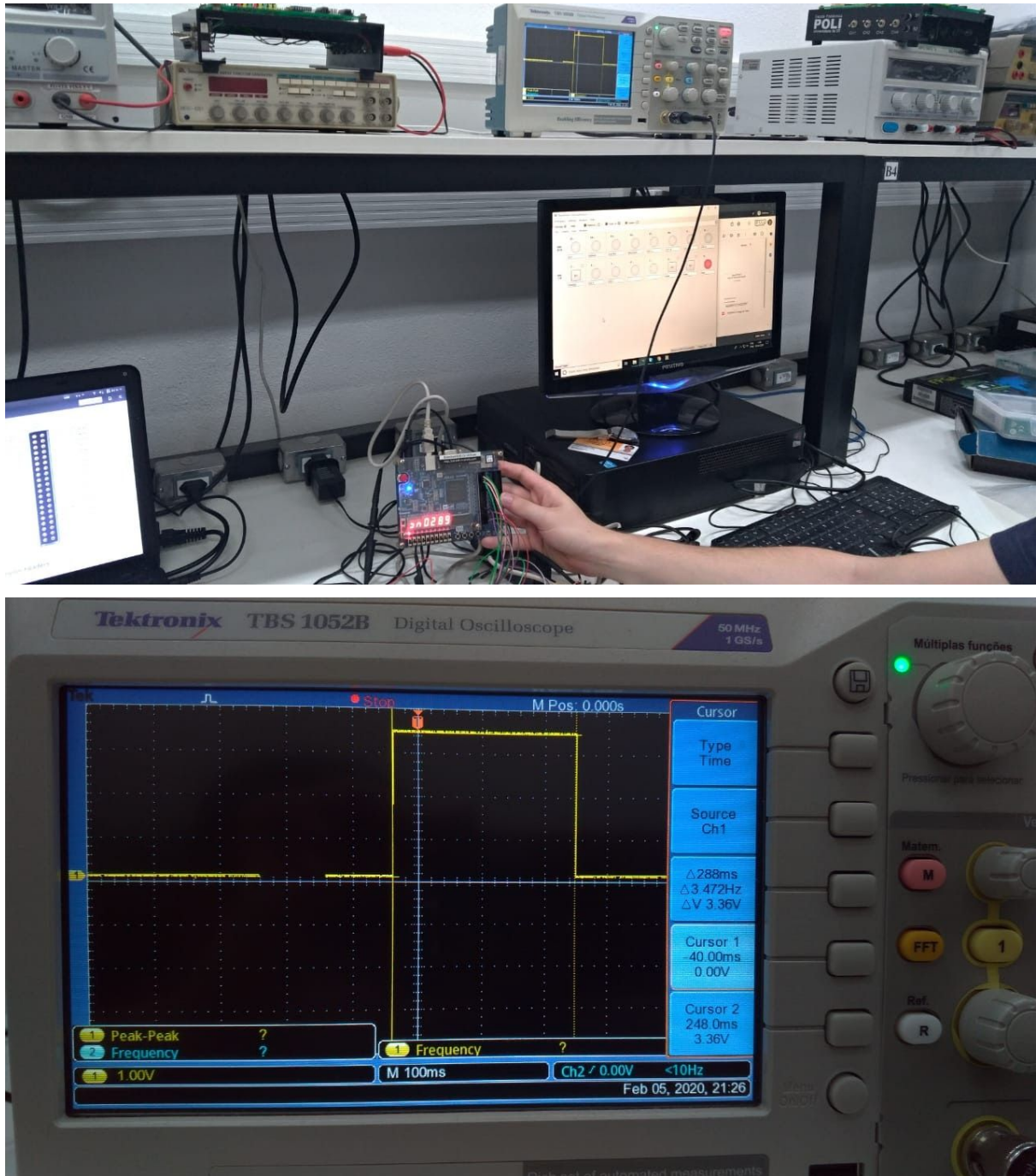


Figura 7 - Contagem feita pelo circuito após uma jogada válida



Para verificar a medida do tempo de reação, foi utilizado um osciloscópio digital presente na bancada da experiência. A imagem seguinte ilustra o procedimento adotado:



**Figura 8 - Medição do tempo de pulso no osciloscópio**

As medidas resultaram em uma diferença de 2 milissegundos entre o apresentado nos *displays* e o medido com o osciloscópio, o que estava dentro de uma margem de erro aceitável, visto que o contador presente no circuito conta até 9999 milissegundos, e portanto o sinal gerado pelo Analog Discovery foi satisfatório para o que se esperava.

## Perguntas

**1. O que acontece na saída se o valor da medida for superior a 9999?**

O contador volta a contagem para 0000.

**2. O que acontece no circuito do Jogo do Tempo de Reação se o jogador acionar o botão de resposta antes da ativação do estímulo?**

A saída “erro” é ativada, para sinalizar que a jogada foi inválida e os displays amostram 9999 na saída da entidade. O sinal de erro não pode ser visto, no entanto, quando a frequência de clock é ajustada para 1kHz, visto que o estado que aciona o erro só dura um período de clock, e o impulso torna-se imperceptível para um observador do circuito.

**3. Qual é a precisão das medidas do tempo de reação? Qual foi o erro médio (em porcentagem) obtido? Qual foi o maior erro absoluto (em microssegundos)?**

As medidas de tempo tiveram um erro médio de 2000 microssegundos. Isso corresponde a um erro de 0,02% em relação ao maior número representável no contador, o que consideramos bem preciso.

### 1.3. Atividade 3 - Demonstração Inicial do Jogo do Tempo de Reação

Não tivemos tempo para demonstrar o funcionamento do circuito em aula, devido ao atraso no desenvolvimento de uma entidade funcional para o projeto.

### 1.4. Atividade 4 - Desafio

Foi proposto que se usasse um buzzer para fazer o papel do estímulo no jogo da reação. Tivemos que desconectar algumas ligações entre o FPGA e o Analog Discovery para podermos acoplar o dispositivo do desafio.

Usamos a placa de circuito integrado presente na bancada do laboratório para materializar os botões do jogo e tivemos que usar um circuito de conversão de tensão entre eles e o FPGA, com o objetivo de não queimar os componentes da placa com tensões acima do nominal (3,3V).

A montagem foi feita da seguinte forma:

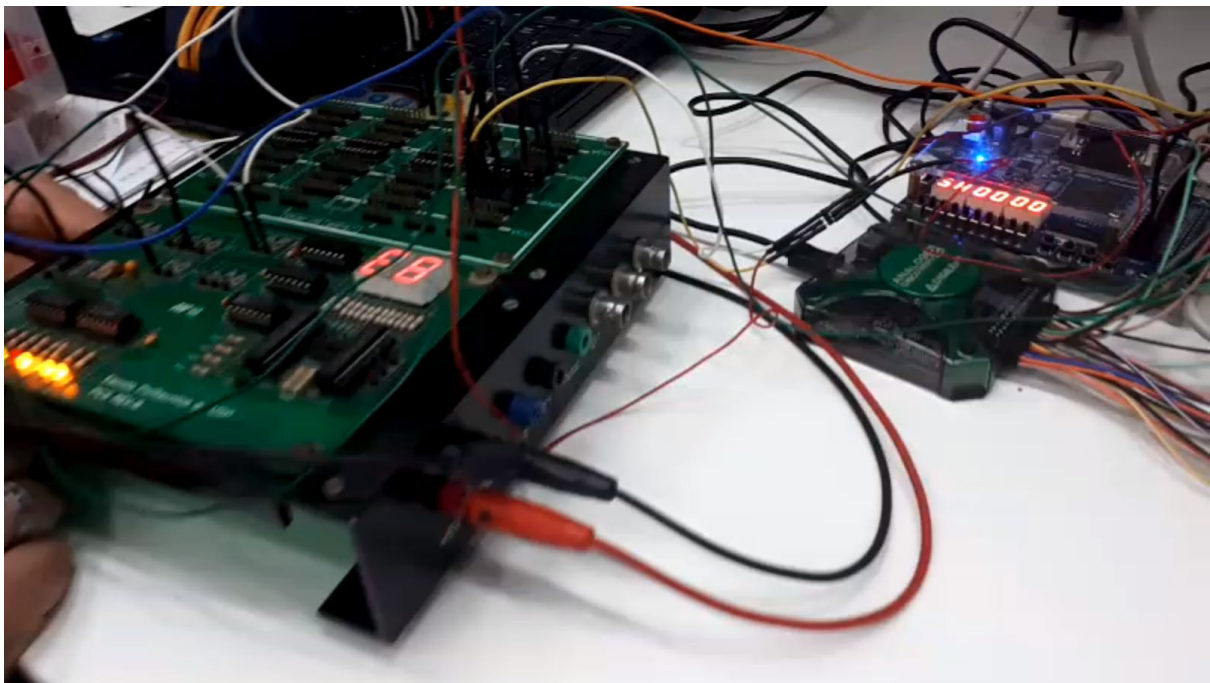


Figura 9 - FPGA integrado à placa de circuitos integrados



Figura 10 - Buzzer em cima da placa para circuitos integrados

## Resultados alcançados

Após resolvermos os problemas no circuito original projetado, pudemos testar a nossa entidade no Quartus e finalmente ter certeza de que ela funcionava. Fizemos o mesmo para o que foi proposto no desafio.

Com os VHDLs em mãos, sintetizamos o jogo e o desafio na placa FPGA durante o *open lab*, na quarta-feira logo após a experiência. O circuito funcionou como projetado nos dois casos propostos (válido e inválido).

## Pontos positivos:

- Conseguimos fazer a entidade funcionar a tempo.
- O circuito se comportou como planejado.

## Pontos negativos:

- Demoramos demais para fazer a nossa máquina de estados funcionar como desejado.
- Usamos todo o tempo de laboratório resolvendo o problema que não conseguimos solucionar durante o planejamento.
- Gastamos muito tempo no planejamento sem apresentar grandes resultados durante a segunda-feira.

## Lições aprendidas:

Conseguimos visualizar melhor como uma máquina de estados pode estar presente em um circuito de forma a interferir em outras máquinas de estados e atualizamos a nossa definição de “sinal assíncrono”, pois embora os estados do VHDL descrito nesta experiência transitem apenas em bordas de subida de *clock*, os sinais de entrada gerados pelo usuário jamais duram menos que um período completo de *clock*, ou seja, o acionamento torna-se assíncrono na prática, devido à inexistência dos botões do circuito.



## Apêndice

```
1  -- Arquivo : exp5-T3BB3.vhd
2  -- Projeto : Jogo do Tempo de Reação
3  -- Data : 01/02/2020
4
5  -- Descrição : circuito medidor de largura de pulso proposto
6  -- para o experimento 3.
7
8  --
9
10 -- T3BB3 :
11 -- Data : Alunos
12 -- 01/02/2020 Lucas Lopes de Paula Junior - 9344880
13 -- Arthur Pires da Fonseca - 10773096
14
15
16 library ieee;
17 use ieee.std_logic_1164.all;
18 use ieee.numeric_std.all;
19
20 entity exp5_T3BB3 is
21 port (
22     jogar : in std_logic;
23     resposta : in std_logic;
24     reset : in std_logic;
25     clock : in std_logic;
26     ligado : out std_logic;
27     pulso : out std_logic;
28     estimulo : out std_logic;
29     erro : out std_logic;
30     prontoExp, prontoInter, prontoMed : out std_logic;
31     tempo : out std_logic_vector(15 downto 0);
32     display0, display1, display2, display3 : out std_logic_vector(6 downto 0);
33     estado : out std_logic_vector(2 downto 0);
34     overflow : out std_logic;
35     interface : out std_logic_vector(3 downto 0);
36     deuTempo : out std_logic;
37     estadoMed : out std_logic_vector(6 downto 0)
38 );
39
40 end entity;
41
42 architecture arch of exp5_T3BB3 is
43
44     component interface_leds_botoes is
45     port (
46         clock, reset: in std_logic;
47         iniciar, resposta: in std_logic;
48         ligado, estimulo, pulso: out std_logic;
49         erro, pronto: out std_logic;
50         deuTempoReacao : out std_logic;
51         estado : out std_logic_vector(3 downto 0)
52         --contador : out std_logic_vector(3 downto 0)
53     );
54 end component;
55
56
57 component medidor_largura is --entidade medidor_largura
58 port (
59     clock, reset : in std_logic; --entradas clock e reset
60     liga, sinal : in std_logic; --entradas liga e sinal
61     display0 : out std_logic_vector(6 downto 0); --digito 0 (7 bits para display 7 seg)
62     display1 : out std_logic_vector(6 downto 0); --digito 1 (7 bits para display 7 seg)
63     display2 : out std_logic_vector(6 downto 0); --digito 2 (7 bits para display 7 seg)
64     display3 : out std_logic_vector(6 downto 0); --digito 3 (7 bits para display 7 seg)
65     display5 : out std_logic_vector(6 downto 0); --estado atual da máquina (7 bits para display 7 seg)
66     fim : out std_logic; --saída fim
67     pronto : out std_logic; --saída pronto
68     db_largura : out std_logic_vector(15 downto 0); --saída db_largura de 16 bits
69     db_clock : out std_logic; --saída de clock
70     db_zeraCont : out std_logic; --saída db_zeraCont
71     db_contaCont : out std_logic
72 );
73 end component;
74
75
76 signal s_ligado, s_pulso, s_erro, s_estimulo, s_contar, s_pulsoAux : std_logic;
77 type tipo_estado is (INICIAL, LIGADO2, REACAO, INVALIDO, FINAL);
78 signal Ereg, Eprox: tipo_estado;
79 signal displays : std_logic_vector(27 downto 0);
80 signal s_estado : std_logic_vector(2 downto 0);
81 signal deuTempoReacao : std_logic;
82 signal estadoInterface : std_logic_vector(3 downto 0);
83
84 begin
85
86     med : medidor_largura port map (
87         clock => clock,
88         reset => reset,
89         liga => s_ligado,
90         sinal => s_pulsoAux,
91         pronto => prontoMed,
92         fim => overflow,
93         db_largura => tempo,
94         display0 => displays(6 downto 0),
95         display1 => displays(13 downto 7),
96         display2 => displays(20 downto 14),
97         display3 => displays(27 downto 21),
98         display5 => estadoMed
99     );
100
101     estimulo <= s_estimulo;
102     ligado <= s_ligado;
103     interface <= estadoInterface;
104     deuTempo <= deuTempoReacao;
105     pulso <= s_pulso;
106
```

```

106
107 inter : interface_leds_botoes port map(
108     clock => clock,
109     reset => reset,
110     iniciar => s_ligado,
111     resposta => resposta,
112     ligado => open,
113     pulso => s_pulso,
114     erro => s_erro,
115     pronto => prontoInter,
116     deuTempoReacao => deuTempoReacao,
117     estimulo => s_estimulo,
118     estado => estadoInterface
119 );
120
121 main : process (clock, reset)
122 begin
123     if reset = '1' then --se reset = '1', então:
124         Ereg <= INICIAL;
125     elsif clock'event and clock = '1' then
126         Ereg <= Eprox; --sinal Ereg recebe sinal Eprox
127     end if; --fim do if
128 end process; --fim do processo
129
130 -- logica de proximo estado
131 transicao : process (jogar, resposta, deuTempoReacao, Ereg, reset)
132 begin
133     case Ereg is
134         when INICIAL => if jogar = '0' or reset = '1' then
135             Eprox <= INICIAL;
136         else
137             Eprox <= LIGADO2;
138         end if;
139
140         when LIGADO2 => if resposta = '0' and deuTempoReacao = '0' then
141             Eprox <= LIGADO2;
142         elsif resposta = '1' and deuTempoReacao = '0' then
143             Eprox <= INVALIDO;
144         elsif reset = '1' then
145             Eprox <= INICIAL;
146         else
147             Eprox <= REACAO;
148         end if;
149
150         when REACAO => if resposta = '0' then
151             Eprox <= REACAO;
152         elsif reset = '1' then
153             Eprox <= INICIAL;
154         else
155             Eprox <= FINAL;
156         end if;
157

```

```

157
158         when FINAL => if reset = '1' then
159             Eprox <= INICIAL;
160         else
161             Eprox <= FINAL;
162         end if;
163
164         when INVALIDO => if reset = '1' then
165             Eprox <= INICIAL;
166         else
167             Eprox <= INVALIDO;
168         end if;
169
170     when others => Eprox <= INICIAL;
171 end case;
172 end process;
173
174 with Ereg select
175     s_estado <= "001" when INICIAL,
176                "010" when LIGADO2,
177                "011" when REACAO,
178                "100" when FINAL,
179                "101" when INVALIDO,
180                "000" when others;
181
182 estado <= s_estado;
183 erro <= s_erro;
184
185
186 with Ereg select
187     s_ligado <= '0' when INICIAL | FINAL,
188              '1' when others;
189
190 with Ereg select
191     s_pulsoAux <= '1' when REACAO,
192                '0' when others;
193
194 with Ereg select
195     prontoExp <= '1' when FINAL,
196                '0' when others;
197
198

```



```

199
200     with Ereg select
201         display0 <= displays(6 downto 0) when FINAL,
202             "0010000" when INVALIDO,
203             "1000000" when others;
204
205     with Ereg select
206         display1 <= displays(13 downto 7) when FINAL,
207             "0010000" when INVALIDO,
208             "1000000" when others;
209
210     with Ereg select
211         display2 <= displays(20 downto 14) when FINAL,
212             "0010000" when INVALIDO,
213             "1000000" when others;
214
215     with Ereg select
216         display3 <= displays(27 downto 21) when FINAL,
217             "0010000" when INVALIDO,
218             "1000000" when others;
219
220 end architecture;
221

```

**exp5\_T3BB3.vhd**

## Bibliografia

[1][https://edisciplinas.usp.br/pluginfile.php/5046976/mod\\_resource/content/1/4-Interface-leds-botoes-2020.pdf](https://edisciplinas.usp.br/pluginfile.php/5046976/mod_resource/content/1/4-Interface-leds-botoes-2020.pdf)

[2][https://edisciplinas.usp.br/pluginfile.php/5046978/mod\\_resource/content/1/PCS3635-4-Interface%20com%20Leds%20e%20Botoes-v1.pdf](https://edisciplinas.usp.br/pluginfile.php/5046978/mod_resource/content/1/PCS3635-4-Interface%20com%20Leds%20e%20Botoes-v1.pdf)

[3]D'Amore, R. - VHDL Descrição e Síntese de Circuitos Digitais - 2a edição, LTC