



**UNIVERSIDADE FEDERAL
DE MATO GROSSO**

Discente:	RGA
Débora Mateus	202221901007
Marcos Antonio Borges	202121901056
Maria Vitória Sousa Rodrigues Palma	202221901035
Victor Daniel Mohr Steffens	202221901059

Relatório da aula prática do dia 01/08/2023

1. Objetivo

Fazer um contador em anel de 4 bits com clock manual.

2. Materiais e Métodos

Para a realização da atividade foram utilizados:

- **Software Altera**
- **Quartus II**
- **Placa OMDAZZ EP4CE6E22C8**
- **UBS Blaster**

3. Procedimento Experimental



**UNIVERSIDADE FEDERAL
DE MATO GROSSO**

Código comentado clkdiv

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.std_logic_unsigned.all;
4
5  entity clkdiv is
6  port(
7      mclk: in std_logic;      -- Sinal de clock mestre de entrada.
8      clr: in std_logic;      -- Sinal de reset assíncrono de entrada.
9      clk25: out std_logic;    -- Saída do clock de 25 MHz.
10     clk190: out std_logic;   -- Saída do clock de 190 kHz.
11     clk3: out std_logic      -- Saída do clock de 3 Hz.
12 );
13 end clkdiv;
14
15 architecture clkdiv of clkdiv is
16     signal q: std_logic_vector (23 downto 0); -- Registrador de contagem para divisão do clock.
17 begin
18     -- Processo para dividir o clock
19     process (mclk, clr)
20     begin
21         if (clr = '1') then
22             q <= X"0000000"; -- Caso clr esteja ativo, redefinimos o contador para 0.
23         else
24             if mclk'event and mclk = '1' then
25                 q <= q + 1;    -- Incrementa o contador a cada borda de subida do clock.
26             end if;
27         end if;
28     end process;
29
30     clk25 <= q(0);           -- Saída do clock de 25 MHz é o bit menos significativo.
31     clk190 <= q(17);        -- Saída do clock de 190 kHz é o bit correspondente ao índice 17.
32     clk3 <= q(23);          -- Saída do clock de 3 Hz é o bit mais significativo.
33 end clkdiv;
```



**UNIVERSIDADE FEDERAL
DE MATO GROSSO**

Código comentado ring4

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.std_logic_unsigned.all;
4
5  entity ring4 is
6  port (
7      clk: in std_logic;      -- Sinal de clock de entrada.
8      clr: in std_logic;      -- Sinal de reset assíncrono de entrada.
9      q:  out std_logic_vector (3 downto 0) -- Saída que representa o estado do contador.
10 );
11 end ring4;
12
13 architecture ring4 of ring4 is
14     signal qs: std_logic_vector (3 downto 0); -- Sinal interno para armazenar o estado do contador.
15 begin
16     process (clk, clr)
17     begin
18         if clr = '1' then
19             qs <= "0001"; -- Caso clr esteja ativo, redefinimos o contador para o valor inicial.
20         elsif clk'event and clk = '1' then
21             case qs is
22                 when "0001" => qs <= "0010"; -- Se o contador está em 0001, avança para 0010.
23                 when "0010" => qs <= "0100"; -- Se o contador está em 0010, avança para 0100.
24                 when "0100" => qs <= "1000"; -- Se o contador está em 0100, avança para 1000.
25                 when "1000" => qs <= "0001"; -- Se o contador está em 1000, volta para 0001.
26                 when others => qs <= "0001"; -- Para qualquer outro valor, volta para 0001.
27             end case;
28         end if;
29     end process;
30     q <= qs; -- A saída q assume o valor atual do contador.
31 end ring4;
32
```



**UNIVERSIDADE FEDERAL
DE MATO GROSSO**

Código comentado ring4_top

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3  use IEEE.std_logic_unsigned.all;
4
5  entity ring4_top is
6  port (
7      mclk: in std_logic;    -- Sinal de clock mestre de entrada.
8      ld : out std_logic_vector (3 downto 0) -- Saída que representa o estado do contador de anel.
9  );
10 end ring4_top;
11
12 architecture ring4_top of ring4_top is
13     component clkdiv
14     port (
15         mclk: in std_logic;
16         clr: in std_logic;
17         clk25: out std_logic; -- Saída do clock de 25 MHz.
18         clk190: out std_logic; -- Saída do clock de 190 kHz.
19         clk3: out std_logic;  -- Saída do clock de 3 Hz.
20     );
21     end component;
22
23     component ring4
24     port (
25         clk: in std_logic;
26         clr: in std_logic;
27         q: out std_logic_vector (3 downto 0) -- Saída que representa o estado do contador de anel.
28     );
29     end component;
30
31     signal sys_clock: std_logic; -- Sinal interno para o clock do sistema.
32 begin
33     C1: clkdiv port map (
34         mclk => mclk,
35         clr => '0',
36         clk3 => sys_clock
37     );
38
39     C2: ring4 port map (
40         clk => sys_clock,
41         clr => '0',
42         q => ld
43     );
44
45 end ring4_top;
```



**UNIVERSIDADE FEDERAL
DE MATO GROSSO**

Código comentado clock_pulse

```
1  library IEEE;
2  use IEEE.std_logic_1164.all;
3
4  entity clock_pulse is
5  port (
6      inp: in std_logic;      -- Entrada de dados
7      cclk: in std_logic;     -- Sinal de clock
8      clr: in std_logic;      -- Sinal de reset assíncrono
9      outp: out std_logic     -- Saída resultante
10 );
11 end clock_pulse;
12
13 architecture clock_pulse of clock_pulse is
14     signal delay1, delay2, delay3 : std_logic; -- Sinais de atraso
15 begin
16     process (cclk , clr)
17     begin
18         if (clr = '1') then -- Se o sinal de reset estiver ativo
19             delay1 <= '0';   -- Limpa os atrasos
20             delay2 <= '0';
21             delay3 <= '0';
22         elsif cclk'event and cclk = '1' then -- Apenas nas bordas de subida do sinal de clock
23             delay1 <= inp;    -- Captura o valor de entrada
24             delay2 <= delay1; -- Atrasa um ciclo de clock
25             delay3 <= delay2; -- Atrasa outro ciclo de clock
26         end if;
27     end process;
28     outp <= delay1 and delay2 and delay3 ; -- Resultado da operação lógica AND entre os atrasos
29 end clock_pulse;
30
```



UNIVERSIDADE FEDERAL DE MATO GROSSO

Código comentado clock_pulse_top

```
3 entity clock_pulse_top is
4   port (
5       mclk: in std_logic;           -- Sinal de clock mestre de entrada.
6       btn: in std_logic;           -- Sinal de botão de entrada.
7       ld: out std_logic_vector (3 downto 0) -- Saída que representa o estado do contador.
8   );
9 end clock_pulse_top;
10 architecture clock_pulse_top of clock_pulse_top is
11   component clkdiv is
12   port (
13       mclk: in std_logic;           -- Sinal de clock mestre de entrada.
14       clr: in std_logic;           -- Sinal de reset assíncrono de entrada.
15       clk25: out std_logic;         -- Saída do clock de 25 MHz.
16       clk190: out std_logic;        -- Saída do clock de 190 kHz.
17       clk3: out std_logic           -- Saída do clock de 3 Hz.
18   );
19 end component;
20   component clock_pulse is
21   port (
22       inp: in std_logic;           -- Sinal de entrada do botão.
23       cclk: in std_logic;          -- Sinal de clock para o componente clock_pulse.
24       clr: in std_logic;           -- Sinal de reset assíncrono de entrada.
25       outp: out std_logic          -- Saída do componente clock_pulse.
26   );
27 end component;
28   component ring4 is
29   port (
30       clk: in std_logic;           -- Sinal de clock para o componente ring4.
31       clr: in std_logic;           -- Sinal de reset assíncrono de entrada.
32       q: out std_logic_vector (3 downto 0) -- Saída que representa o estado do contador de anel.
33   );
34 end component;
35   signal sys_clock: std_logic;      -- Sinal interno para o clock do sistema.
36   signal fclock: std_logic;         -- Sinal interno para o clock gerado pelo componente clock_pulse.
37   signal invq: std_logic_vector (3 downto 0); -- Sinal interno para o estado invertido do contador.
38 begin
39   C1: clkdiv port map (--Mapeamento das portas
40       mclk => mclk,
41       clr => '0',
42       clk190 => sys_clock
43   );
44   C2: clock_pulse port map (
45       inp => btn,
46       clr => '0',
47       cclk => sys_clock,
48       outp => fclock
49   );
50   C3: ring4 port map (
51       clk => fclock,
52       clr => '0',
53       q => invq
54   );
55   ld <= not invq; -- Saída ld é a inversão de invq.
56 end clock_pulse_top;
```

Resultado da Compilação do Código

Quartus II 64-Bit - C:/Users/debor/Downloads/aula org exp3 e 4/exp3/ring4_top - ring4_top

File Edit View Project Assignments Processing Tools Window Help

Search altera.com

Project Navigator

Entity

Cyclone IV E: EP4CE6E22C8

ring4_top

dkdiv:C1

ring4:C2

Table ...

Flow Summary

Flow Status: Successful - Mon Aug 07 13:37:31 2023

Quartus II 64-Bit Version: 13.1.0 Build 162 10/23/2013 5J Web Edition

Revision Name: ring4_top

Top-level Entity Name: ring4_top

Family: Cyclone IV E

Device: EP4CE6E22C8

Timing Models: Final

Total logic elements: 28 / 6,272 (< 1 %)

Total combinational functions: 28 / 6,272 (< 1 %)

Dedicated logic registers: 28 / 6,272 (< 1 %)

Total registers: 28

Total pins: 5 / 92 (5 %)

Total virtual pins: 0

Total memory bits: 0 / 276,480 (0 %)

Embedded Multiplier 9-bit elements: 0 / 30 (0 %)

Total PLLs: 0 / 2 (0 %)

Flow: Compilation

Task

Task

Tim

00:00:18

00:00:04

00:00:08

00:00:02

00:00:04

Program Device (Open Programmer)

100% 00:00:18

38°C Ensolarado 13:37 07/08/2023

Pin planner

Pin Planner - C:/Users/debor/Downloads/aula org exp3 e 4/exp3/ring4_top - ring4_top

File Edit View Processing Tools Window Help

Search altera.com

Report

Report not available

Gr... Re...

Tasks

Early P

Earh

Run

Exp

Pin Fin

Highlig

I/O

VRE

Edge

Cloc

Top View

Wire Bond, with Exposed Pad

Cyclone IV E

EP4CE6E22C8

Named: *

Edit: *

Filter: Pins: all

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standarc	Reserved	irrent Stren	Slew Rate	ifferential Pa
mck	Unknown	PIN_23	1	B1_N0	2.5 V ...fault)		8mA (...ault)		
ld[3]	Unknown	PIN_87	5	B5_N0	2.5 V ...fault)		8mA (...ault)		
ld[2]	Unknown	PIN_86	5	B5_N0	2.5 V ...fault)		8mA (...ault)		
ld[1]	Unknown	PIN_85	5	B5_N0	2.5 V ...fault)		8mA (...ault)		
ld[0]	Unknown	PIN_84	5	B5_N0	2.5 V ...fault)		8mA (...ault)		
<<new node>>									

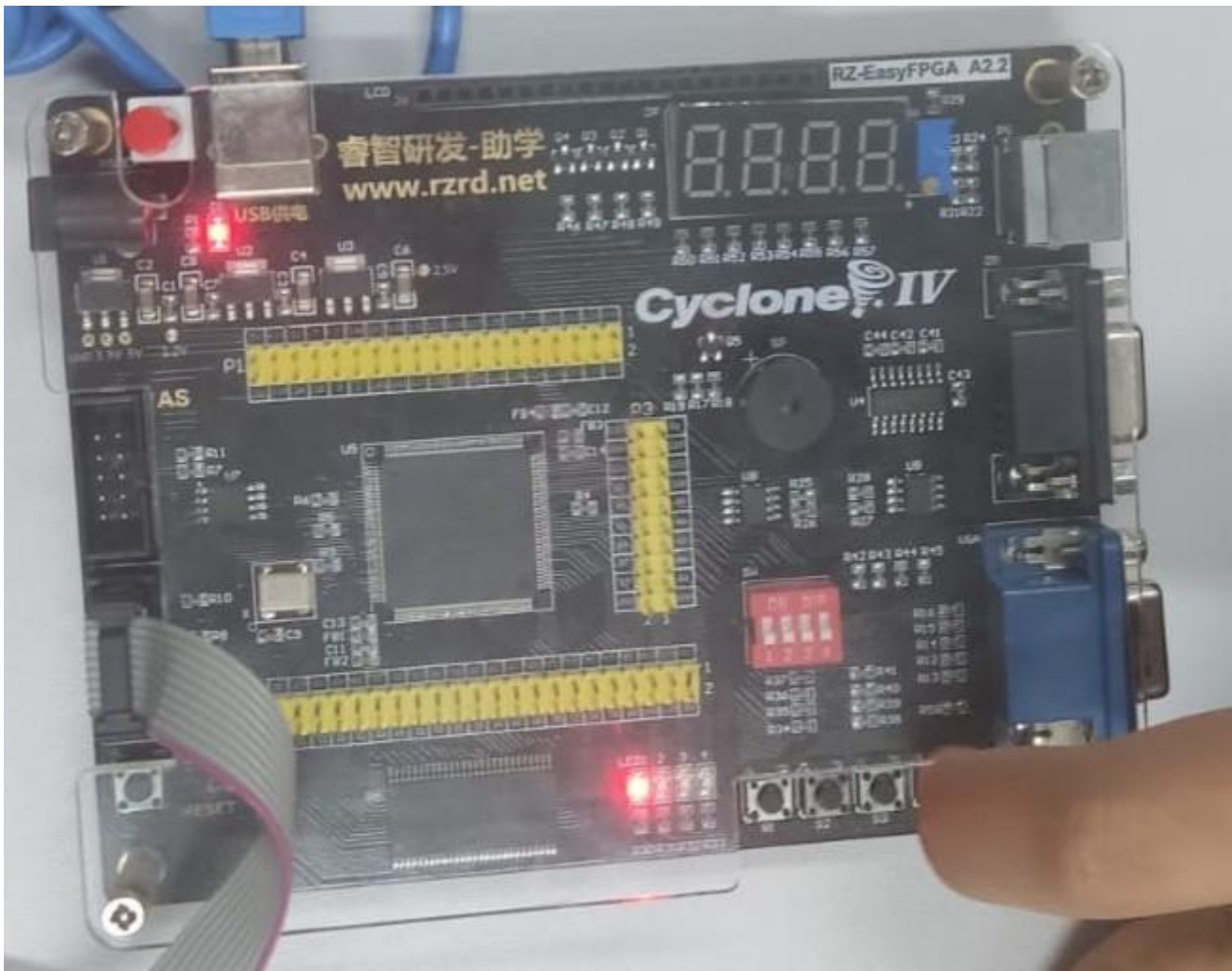
0% 00:00:00

38°C Ensolarado 13:35 07/08/2023



**UNIVERSIDADE FEDERAL
DE MATO GROSSO**

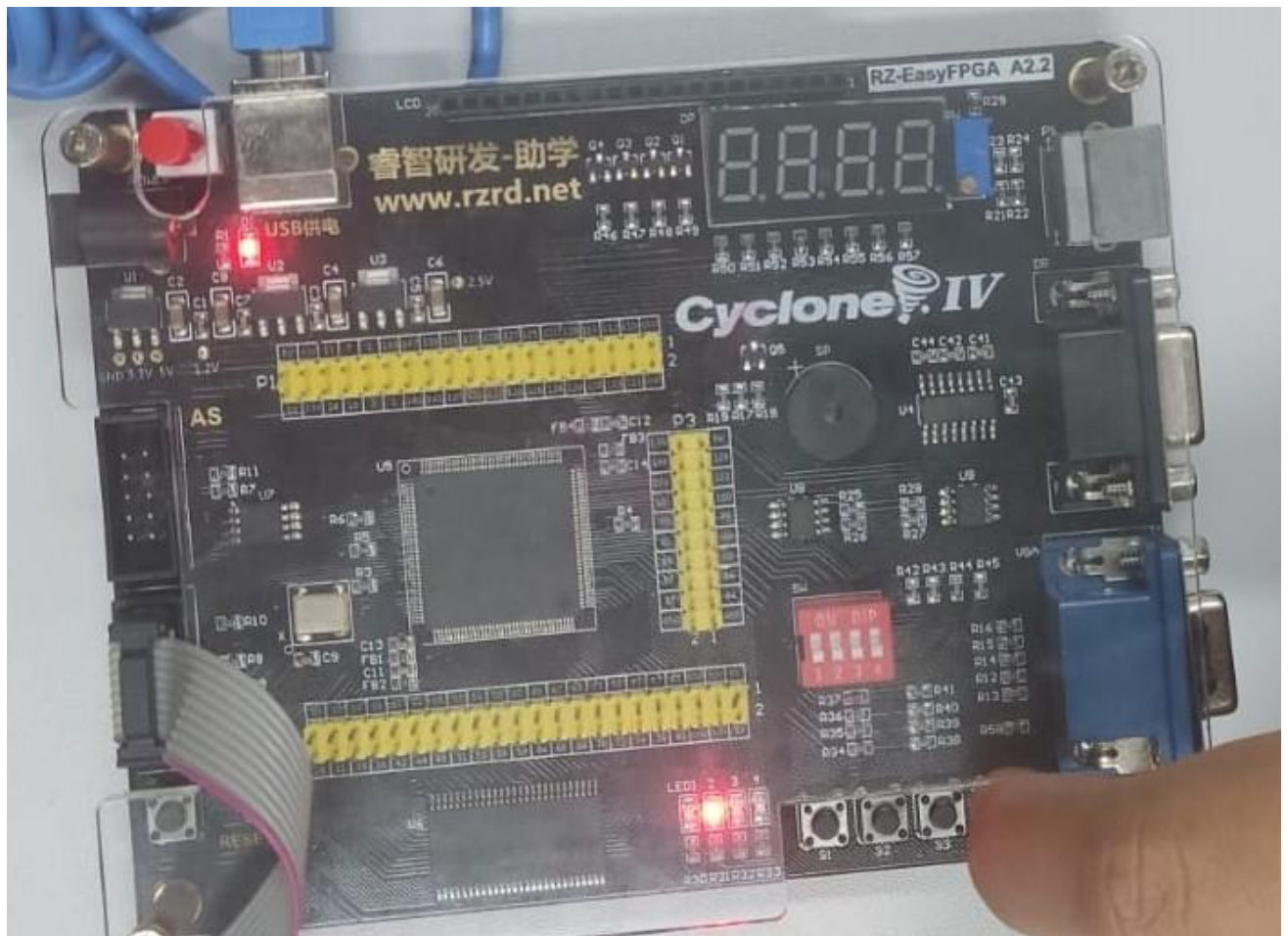
4. Resultados e Conclusão:



Primeiro Led Aceso



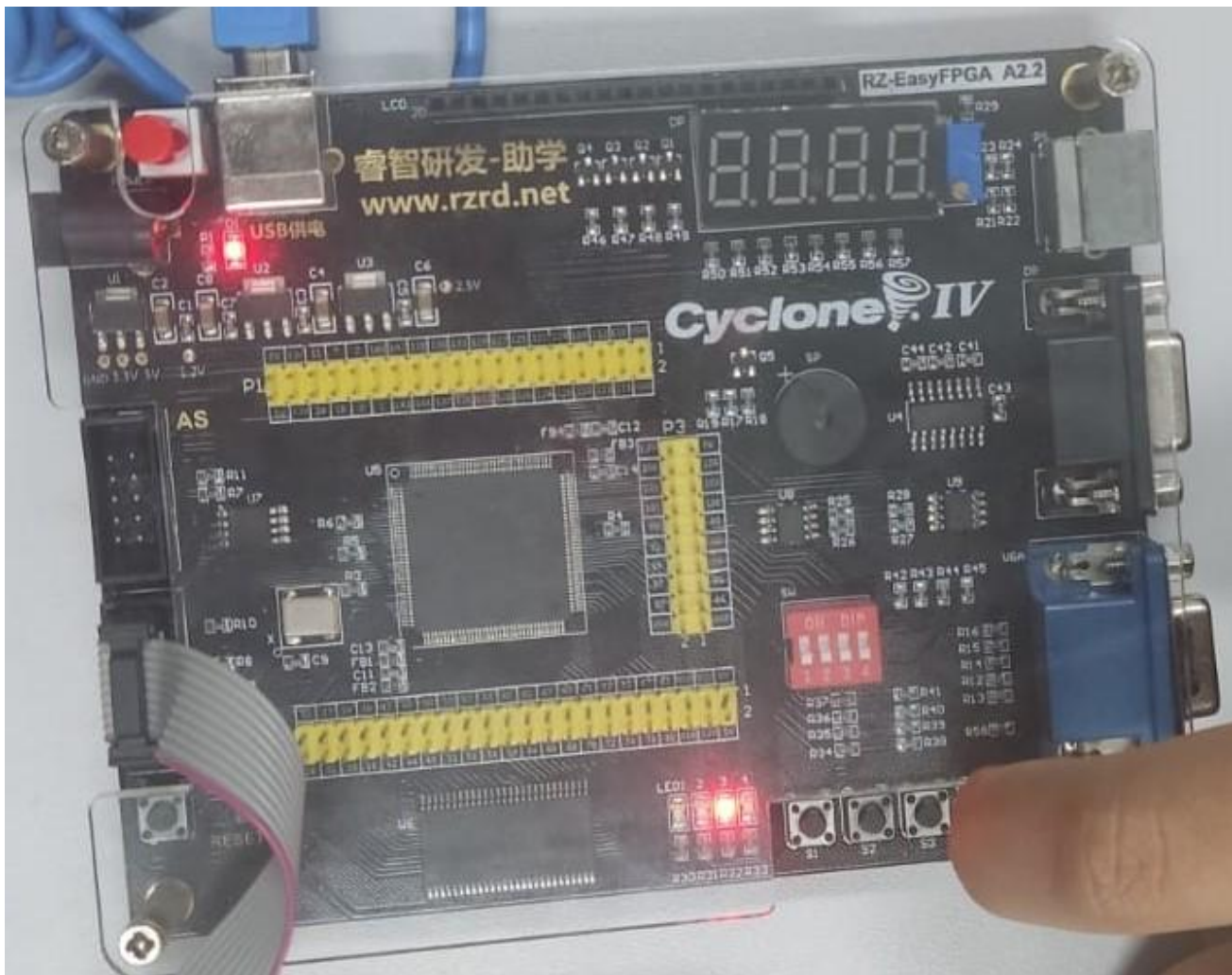
**UNIVERSIDADE FEDERAL
DE MATO GROSSO**



Segundo Led aceso



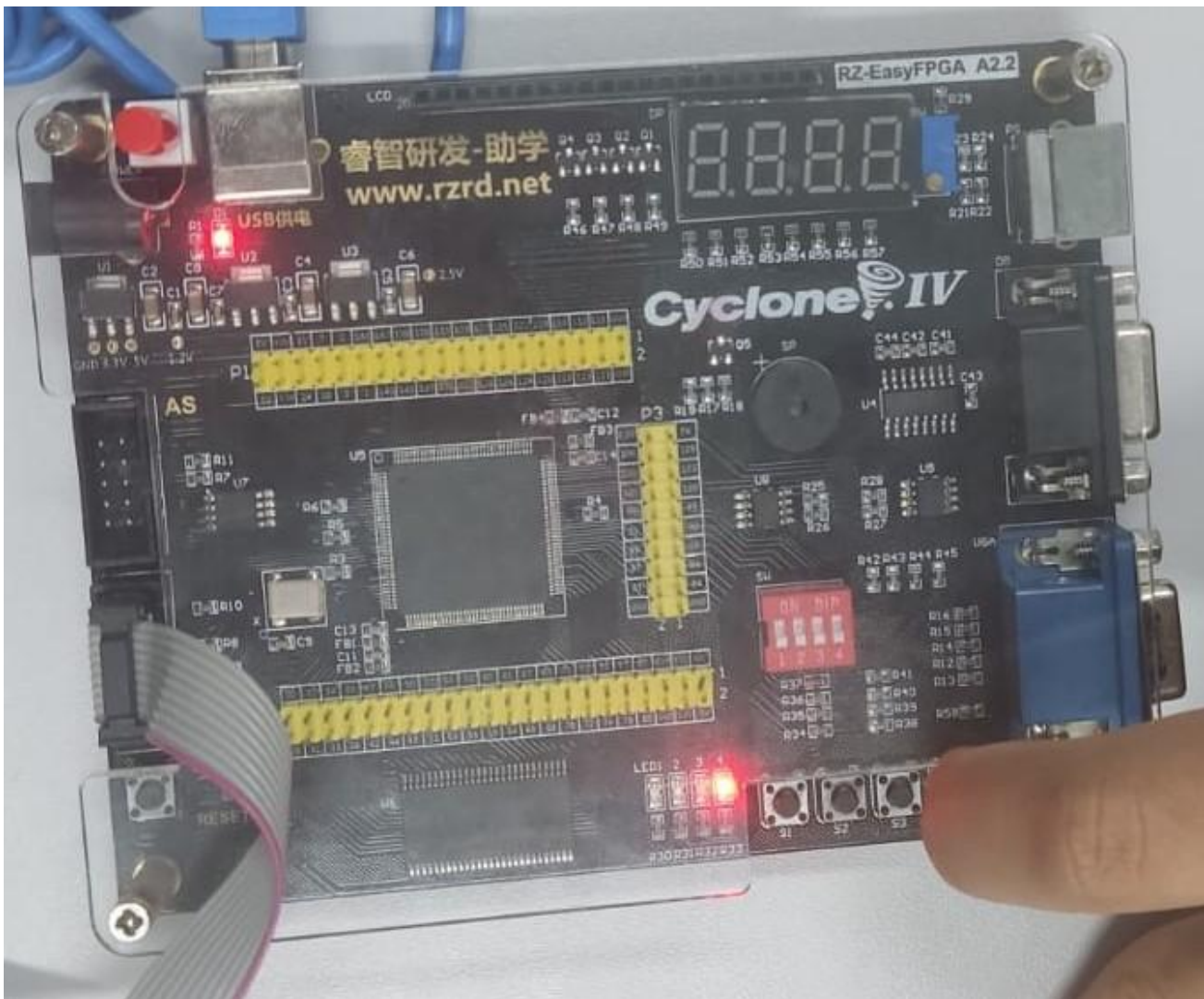
**UNIVERSIDADE FEDERAL
DE MATO GROSSO**



Terceiro Led aceso



**UNIVERSIDADE FEDERAL
DE MATO GROSSO**



Quarto led aceso

5. Bibliografia

BROWN, Stephen D. **Fundamentals of digital logic with VHDL design**. Boston: McGraw-Hill, 2000. 840 p. (004.22 B879f).

D'AMORE, Roberto. A Synthesis-Oriented VHDL Course. **ACM Transactions on Computing Education (TOCE)**, v. 10, n. 2, p. 1-24, 2010.

PEDRONI, Volnei A. **Eletrônica Digital Moderna e VHDL**. Elsevier Ltda. Editora, Rio de Janeiro, RJ: 2010. 619 p.