



Centro Universitário FEI

Sistemas Digitais 2

Relatório Projeto 2

Filippe Franca Neves Filho 15.118.667-3

Vítor Dinis Watanabe 11.219.035-0

CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

1. Introdução e Descrição do Projeto

O projeto 2 de Sistemas Digitais 2 tem como objetivo a aplicação dos conhecimentos adquiridos nas aulas de teoria sobre metodologia de projetos de sistemas complexos, sobretudo, projeto no nível de transferência entre registradores (RTL). Com isso, trabalhamos as informações que fluem através de registradores, sendo que entre essas transferências os dados podem ser manipulados por blocos de lógica combinatória.

O projeto 2 consiste na elaboração de sistema de cobrança e medidor de velocidade em uma rodovia com pedágios que não é necessário parar para que a cobrança seja feita. A partir disso, é necessário criar a lógica para o controle das medidas, tais como, medição de velocidade, cobrança de pedágio, controle e verificação de multas, distância totais e parciais, além de sinalizadores e chaves para seleção de modo de operação e controle de crédito.

Para realizar o projeto, realizamos em passos com o acompanhamento da Professora Doutora Maria Claudia Ferrari em nossas aulas semanais. As etapas consistem em: Obter a máquina de estados de alto nível (FSMD) no qual descrevemos o funcionamento do sistema. Elaborar o bloco operacional utilizando os códigos que nos foram disponibilizados. Conectar o bloco operacional a uma unidade de controle. Obter a máquina de estados convertendo o bloco operacional a um bloco de controle (unidade de controle) definindo os sinais de interface.

Para usar o valores da tabela cada grupo usaria o valores de acordo com seu dígito de ra, o dígito que possuir o maior valor usar a linha do DA o de menor usar a linha do DB, então com nossos ra.s (15.118.667-3 e 11.219.035-0) usariamos a Linha do DA igual a 3 e o DB igual a 0.


DA	CRI (RS)	CRM (RS)	DB	AC (RS)	TRP (RS/KM)	LDM (KM)
0	2	15	0	5	1	10
1	3	20	1	4	1	11
2	4	25	2	3	2	12
3	5	30	3	2	2	13
4	6	35	4	5	3	14
5	7	40	5	4	3	15
6	8	45	6	3	4	16
7	9	50	7	2	4	17
8	10	55	8	4	5	18
9	11	60	9	6	5	19

CRI = Crédito Inicial, CRM = Crédito Máximo, AC = Valor do Ajuste de Crédito, TRP = Valor da Tarifa de Pedágio e LDM = Valor do Limite Máximo de Distância

CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

2. Máquina de Estados de Alto Nível do Sistema (FSMD)

Dados Utilizados na Máquina de Estados

DA	CRI	CRM	DB	AC	TRP	LDM
3	5	30	0,00	5	1	10 

Estados

ZER_RG = zera o registrador

CAR_CR = carrega o crédito inicial (5)

AJU_CR = ajusta o crédito

INC_CR = incrementa o crédito

VER_CRM = verifica o crédito máximo

MAX_CR = atribui o valor máximo de crédito (30)

DEC_CR = decrementa o crédito

MIN_CR = atribui o valor mínimo de crédito (0)

INC_DP = incrementa distância parcial

INC_DT = incrementa distância total

VER_DT = verifica distância total

SIN_LIM = sinaliza o limite de velocidade

VER_CR = verifica crédito

MUL_CR = multa por crédito

VF15 = espera o pulso de passagem pelo pórtico ou dispara timer

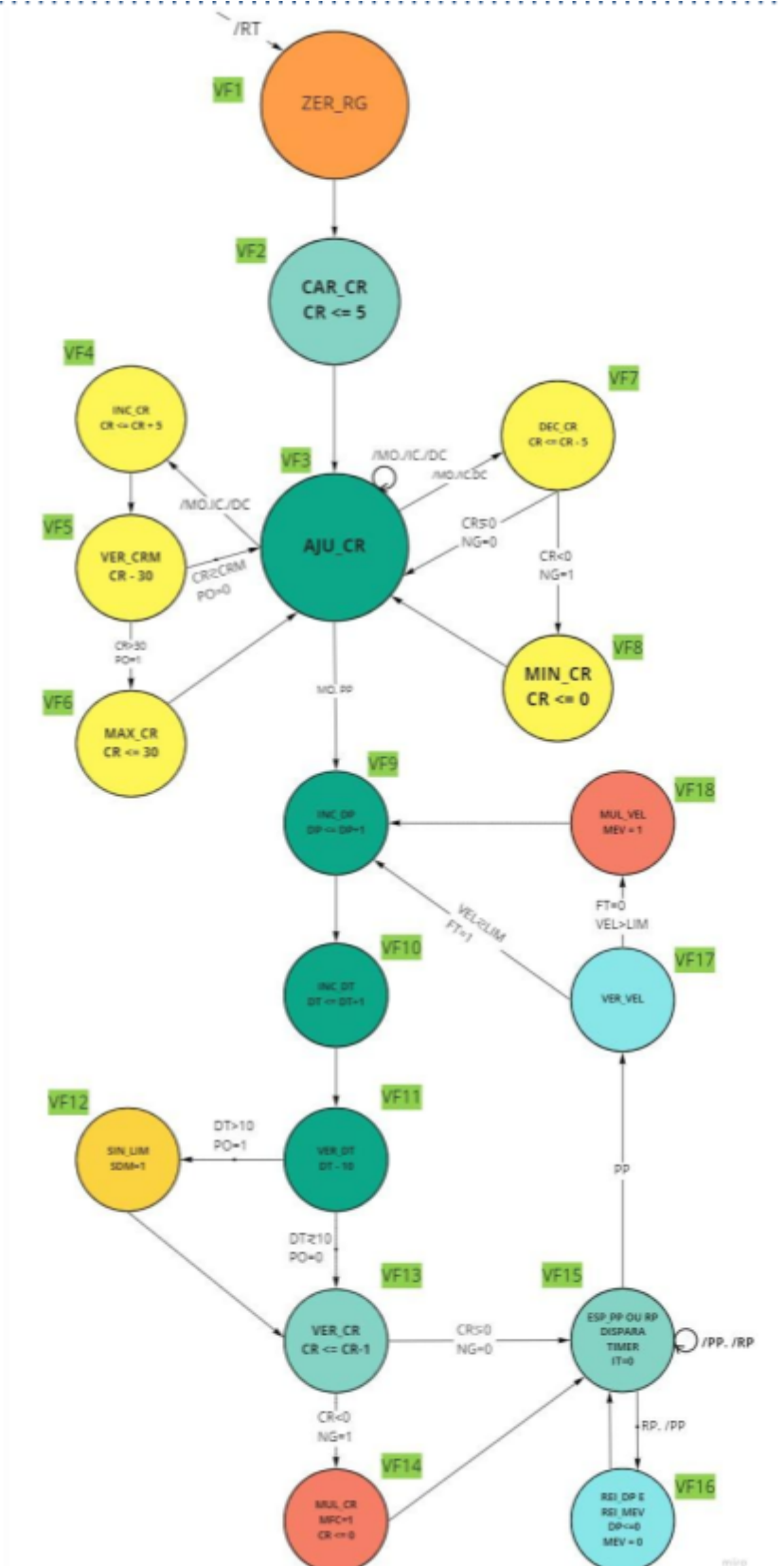
VF16 = reinicia a distância parcial e apaga o LED de multa por excesso de velocidade

VER_VEL = verifica a velocidade

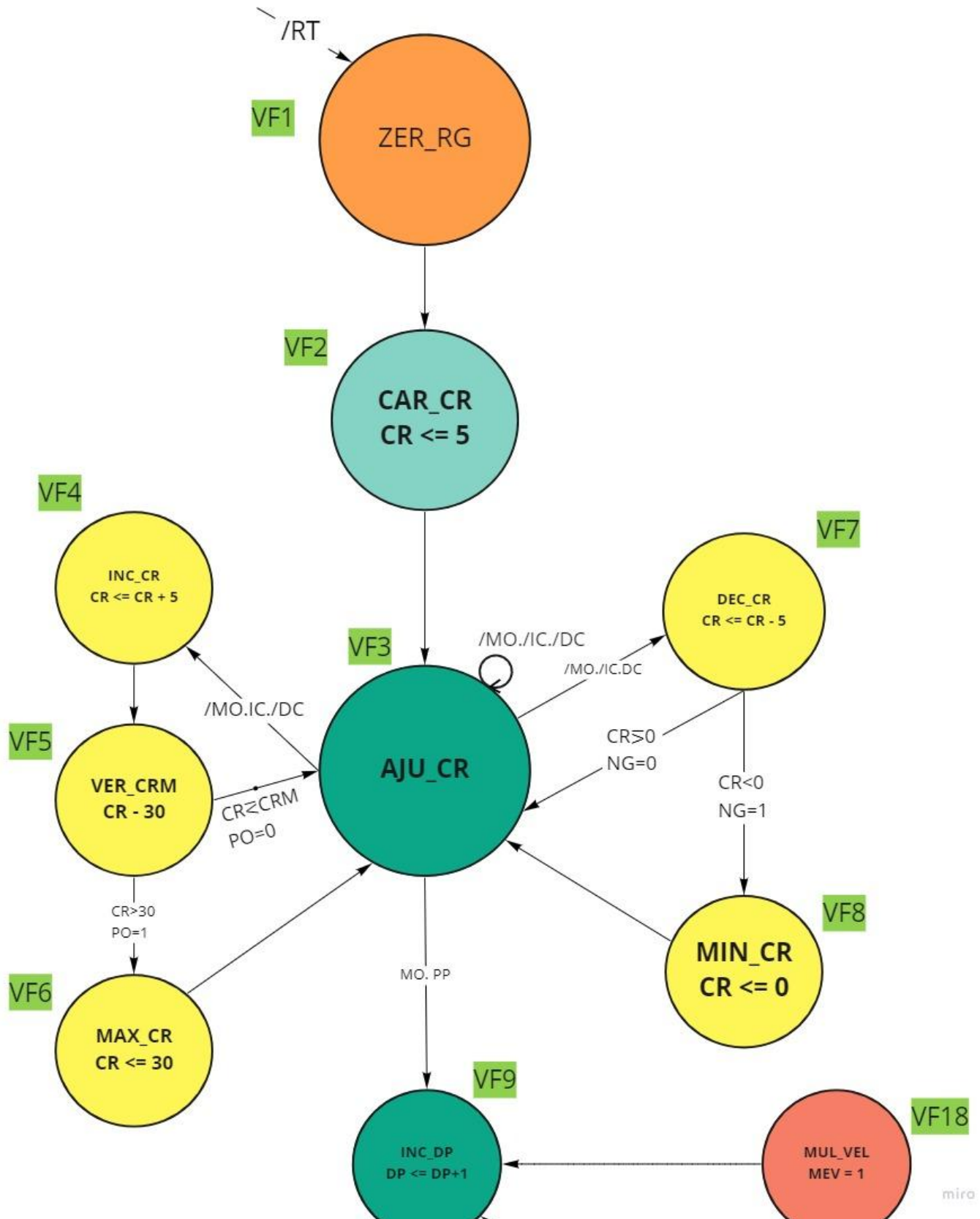
MUL_VEL = multa por velocidade

CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

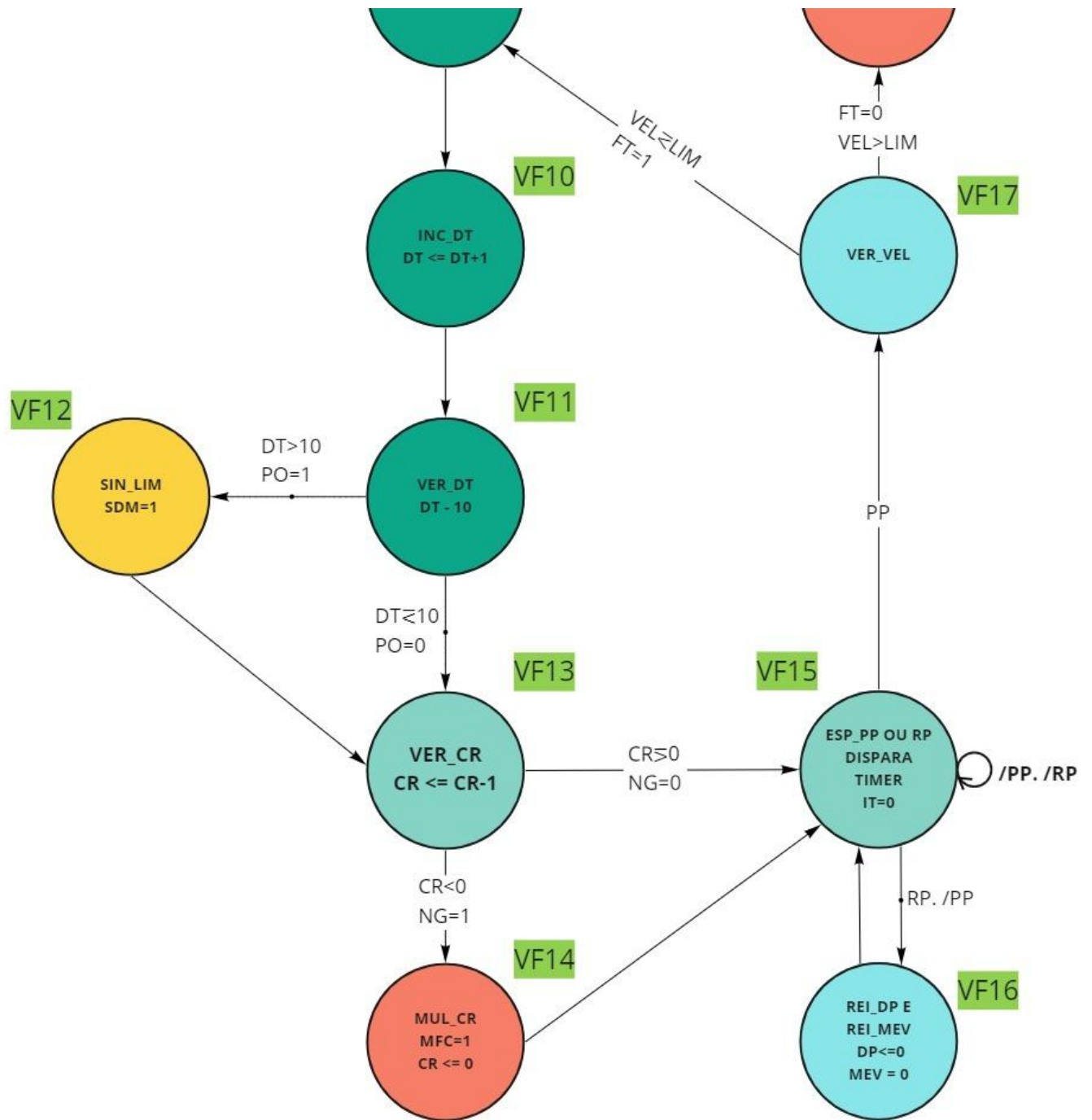
3. Diagrama de Estados da Unidade de Controle (FSM)



CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021



CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021



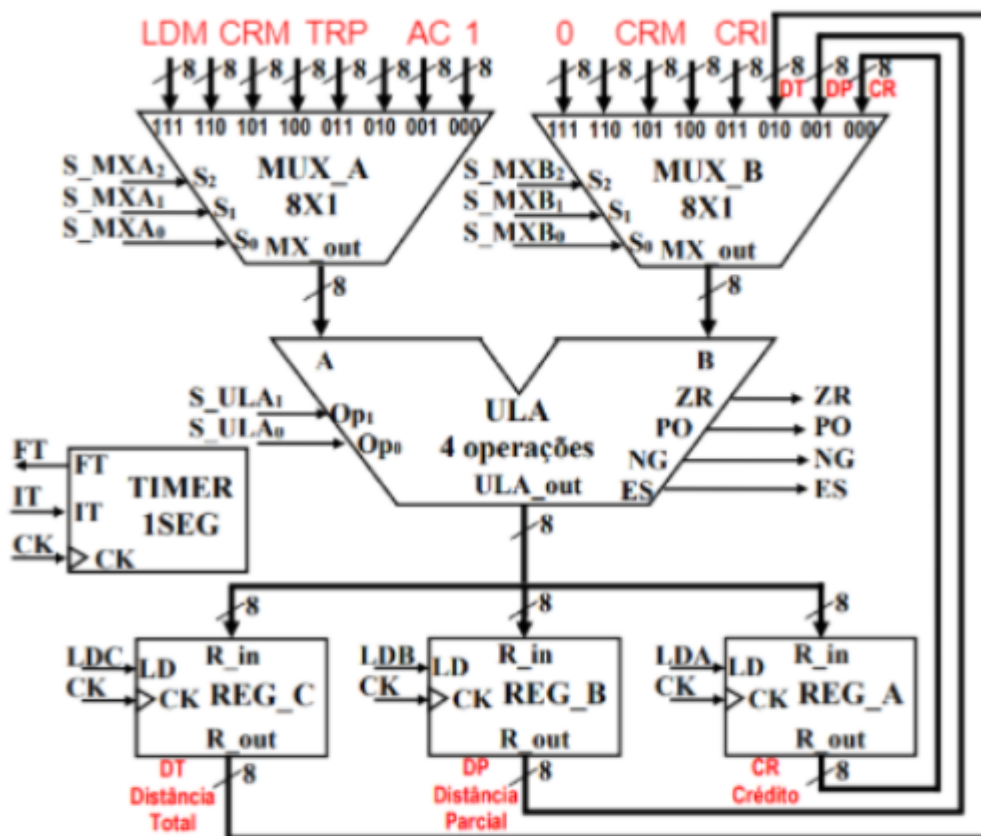
miro

CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

4. Tabela de Sinais de Controle do Fluxo de Dados

A Criação da Tabela de Sinais foi feita através das ações que realmente aconteciam no pedágio do projeto.

Como funciona ? Um exemplo. No estado que o usuário carrega o crédito a ação que ele toma é carregar o valor de 5 créditos, porém ao carregar 5 créditos a ação que o sistema toma é carregar o crédito inicial CRI no MUX_B que seria o valor em código binário de 011 e no MUX_A nenhuma ação, após isso a nossa ULA(que é basicamente uma unidade que faz operações matemáticas) precisa passar o valor do CRI para o Regulador A, a ação que ela toma em binário é 00 de passar valor de B, e para passar esse valor de B para o regulador A precisa ter o código binário de 001, assim ficaria REG_C igual a 0, REG_B igual A zero e REG_A igual a 1, é basicamente assim como funciona o FD.



CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

Tabela de Sinais							
Estado	SMX_A	SMX_B	SULA1-SU LA2	REG_C	REG_B	REG_A	TI
VF1_RG	XXX	111	00	1	1	1	1
VF2_CR	XXX	011	00	0	0	1	1
VF3_ACR	XXX	XXX	XX	0	0	0	1
VF4_ICR	001	000	10	0	0	1	1
VF5_CR	101	000	11	0	0	0	1
VF6_CR	XXX	101	00	0	0	1	1
VF7_CR	001	000	11	0	0	1	1
VF8_CR	XXX	111	00	0	0	1	1
VF9_DP	000	001	10	0	1	0	1
VF10_DT	000	010	10	1	0	0	1
VF11_DT	111	010	11	0	0	0	1
<u>VF12_SDM</u>	XXX	XXX	XX	0	0	0	1
VF13CR	011	000	11	0	0	1	1
V14_CR	XXX	111	00	0	0	1	1
V15_PP	XXX	XXX	XX	0	0	0	0
V16_DP	XXX	111	00	0	1	0	0
V17_VV	XXX	XXX	XX	0	0	0	0
V18_VV	XXX	101	XX	0	0	0	0

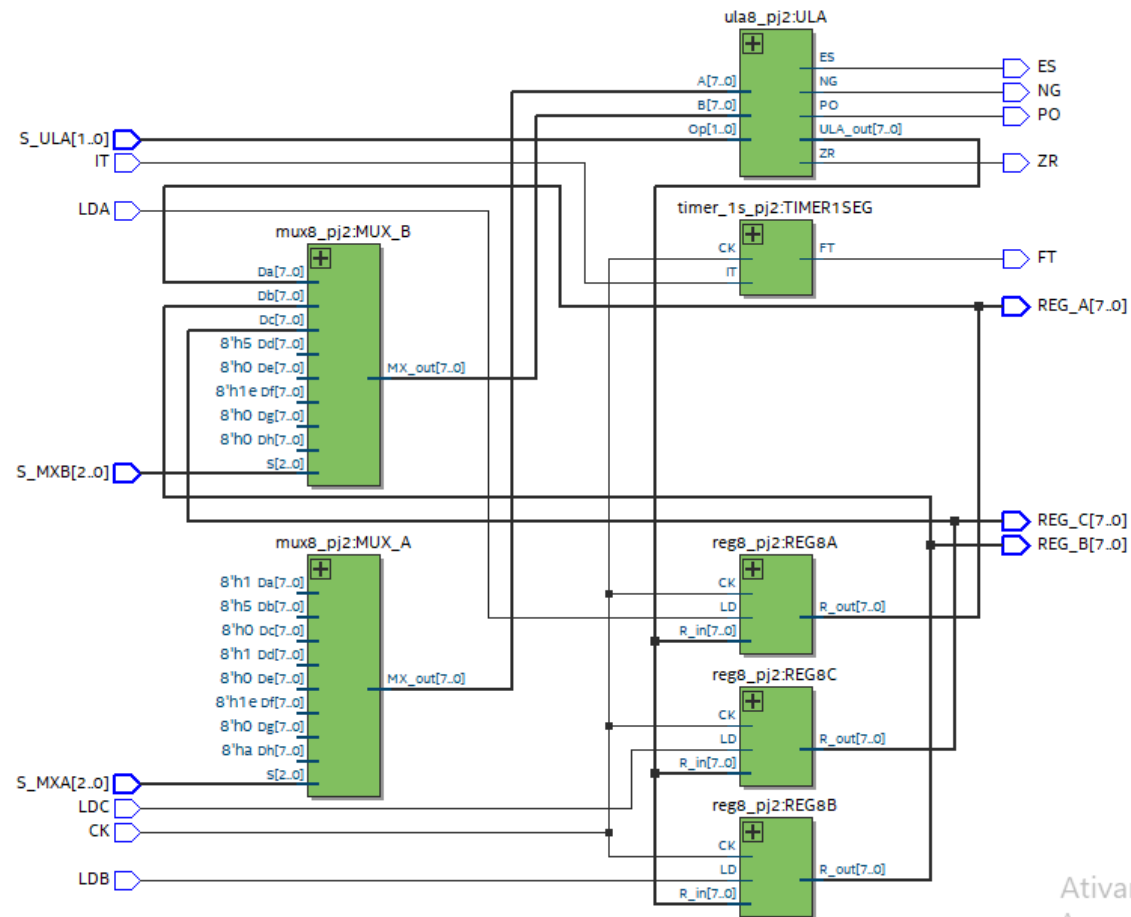
CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

5. Diagrama de Interconexão do Fluxo de Dados

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  ENTITY FD IS PORT(
5      S_MXA, S_MXB: IN STD_LOGIC_VECTOR (2 DOWNTO 0);
6      S_ULA: IN STD_LOGIC_VECTOR (1 DOWNTO 0);
7      LDA, LDB, LDC: IN STD_LOGIC;
8      IT: IN STD_LOGIC;
9      CK: IN STD_LOGIC;
10     FT: OUT STD_LOGIC;
11     ZR, PO, NG, ES: OUT STD_LOGIC;
12     REG_A, REG_B, REG_C: OUT STD_LOGIC_VECTOR(7 downto 0));
13 END FD;
14
15 ARCHITECTURE BEHAVIOR OF FD IS
16
17     SIGNAL UA, UB: std_logic_vector(7 downto 0);
18     SIGNAL RIN: std_logic_vector(7 downto 0);
19     SIGNAL SDT, SDP, SCR: std_logic_vector(7 downto 0);
20
21     component mux8_pj2 is
22     port(Da : in std_logic_vector(7 downto 0);
23          Db : in std_logic_vector(7 downto 0);
24          Dc : in std_logic_vector(7 downto 0);
25          Dd : in std_logic_vector(7 downto 0);
26          De : in std_logic_vector(7 downto 0);
27          Df : in std_logic_vector(7 downto 0);
28          Dg : in std_logic_vector(7 downto 0);
29          Dh : in std_logic_vector(7 downto 0);
30          S : in std_logic_vector(2 downto 0);
31          MX_out : out std_logic_vector(7 downto 0)
32     );
33 end component mux8_pj2;
34
35     component ula8_pj2 is
36     port
37     (
38         A, B : in std_logic_vector(7 downto 0); -- vetores de entrada
39         Op : in std_logic_vector(1 downto 0); -- código de operação
40         ULA_out : out std_logic_vector(7 downto 0); -- vetor de saída
41         ZR : out std_logic; -- resultado igual a zero
42         PO : out std_logic; -- resultado positivo
43         NG : out std_logic; -- resultado negativo
44         ES : out std_logic; -- resultado maior que 255
45     );
46 end component ula8_pj2;
47
48     component reg8_pj2 is
49     port (CK : in std_logic;
50          LD : in std_logic; -- habilita carga
51          R_in : in std_logic_vector(7 downto 0); -- dados de entrada
52          R_out : out std_logic_vector(7 downto 0); -- dados de saída
53     );
54 end component reg8_pj2;
55
56     component timer_is_pj2 IS PORT(
57         CK : IN std_logic; -- Clock de 50MHz
58         IT : IN std_logic; -- Reseta e inicia temporização
59         FT : OUT std_logic; -- Final de temporização
60     );
61 END component timer_is_pj2;
62
63 BEGIN
64
65     MUX_A: mux8_pj2 PORT MAP (S=>S_MXA, Da=>"00000001", Db=>"00000101", Dc=>"00000000", Dd=>"00000001",
66                               De=>"00000000", Df=>"00011110", Dg=>"00000000", Dh=>"00001010", MX_out=>UA);
67
68     MUX_B: mux8_pj2 PORT MAP (S=>S_MXB, Da=>SCR, Db=>SDP, Dc=>SDT, Dd=>"00000101", De=>"00000000",
69                               Df=>"00011110", Dg=>"00000000", Dh=>"00000000", MX_out=>UB);
70
71     ULA: ula8_pj2 PORT MAP (A=>UA, B=>UB, Op=>S_ULA, ZR=>ZR, PO=>PO, NG=>NG, ES=>ES, ULA_out=>RIN);
72
73     REG8A: reg8_pj2 PORT MAP (R_in=>RIN, LD=>LDA, CK=>CK, R_out=>SCR);
74     REG8B: reg8_pj2 PORT MAP (R_in=>RIN, LD=>LDB, CK=>CK, R_out=>SDP);
75     REG8C: reg8_pj2 PORT MAP (R_in=>RIN, LD=>LDC, CK=>CK, R_out=>SDT);
76
77     TIMER1SEG: timer_is_pj2 PORT MAP (FT=>FT, IT=>IT, CK=>CK);
78
79     REG_A<=>SCR;
80     REG_B<=>SDP;
81     REG_C<=>SDT;
82
83 END BEHAVIOR;
```

CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

5. RTL do Fluxo De Dados



Ativar o Windows
Acesse Configurações para ativar o V

CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

6. Código da UC

```
1
2
15 library ieee;
16 use ieee.std_logic_1164.all;
17 use ieee.std_logic_arith.all;
18 use ieee.std_logic_unsigned.all;
19
20 entity UC is
21 port( CK : in std_logic;           -- clock de 50MHz
22       RT : in std_logic;           -- reinício total -> ativo em zero
23       -- Entradas Externas:
24       IC : in std_logic;           -- incrementa credito -> ativo em um
25       DC : in std_logic;           -- decrementa credito -> ativo em um
26       RP : in std_logic;           -- reinício parcial -> ativo em um
27       PP : in std_logic;           -- passagem pelo portico -> ativo em um
28       MO : in std_logic;           -- modo de operação -> 1= Normal, 0= Ajusta valor
29       -- Sinais de Estado da ULA:
30       ZR : in std_logic;           -- resultado zero na operação da ULA
31       PO : in std_logic;           -- resultado positivo na operação da ULA
32       NG : in std_logic;           -- resultado negativo na operação da ULA
33       ES : in std_logic;           -- resultado da operação da ULA maior que 255
34       -- Sinais de Estado do TIMER:
35       FT : in std_logic;           -- fim da temporização de 1 segundo
36       -- Sinais de Saida para MUX:
37       Sel_mxa : out std_logic_vector(2 downto 0); -- seleciona entrada de MUX_A
38       Sel_mxb : out std_logic_vector(2 downto 0); -- seleciona entrada de MUX_B
39       -- Sinais de Saida para ULA:
40       Sel_ula : out std_logic_vector(1 downto 0); -- seleciona operação da ULA
41       -- Sinais de Saida para Registradores:
42       Lda : out std_logic;         -- carrega RA
43       Ldb : out std_logic;         -- carrega RB
44       Ldc : out std_logic;         -- carrega RC
45       -- Sinais de Saida para TIMER:
46       IT : out std_logic;          -- inicia temporização
47       -- Sinais de Saida Externos:
48       MEV : out std_logic;         -- sinaliza multa por excesso de velocidade
49       MFC : out std_logic;         -- sinaliza multa por falta de crédito
50       SDM : out std_logic;         -- sinaliza limite de distância total percorrida
51 );
52 end UC;
53
54 architecture FSM of UC is
55 type ESTADOS_ME is (
56     ZER_RG -- zera registrador
57     ,CAR_CR -- carrega credito
58     ,AJU_CR -- ajusta credito
59     ,INC_CR -- incrementa credito
60     ,VER_CRM -- verifica crm
61     ,MAX_CR -- max valor de cr
62     ,DEC_CR -- decrementa cr
63     ,MIN_CR -- minimo valor de cr
64     ,INC_DP -- incrementa dp
65     ,INC_DT -- incrementa dt
66     ,VER_DT -- verifica dt
67     ,VER_CR -- verifica cr
68     ,SIN_LIM -- sinaliza limite
69     ,MUL_CR -- multa credito
70     ,VF15 -- espera pp ou dispara timer
71     ,VF16 -- reinicia dp e reinicia mev
72     ,REI_DP -- reinicia dp
73     ,REI_MEV -- reinicia mev
74     ,VER_VEL -- verifica velocidade
75     ,MUL_VEL -- multa velocidade
76 );
77
78 signal E: ESTADOS_ME;
79 begin
80 process(CK, RT)
81 begin
82 if RT='0' then E <= ZER_RG; -- zera registros
83 MFC <= '0'; MEV <= '0'; SDM <= '0'; -- zera multas e sinalização
84 elsif (CK'event and CK='1') then
85 case E is
86 when ZER_RG =>
87 E <= CAR_CR; -- carrega CR com credito inicial
88 when CAR_CR =>
89 E <= AJU_CR; --- espera IC e DC com MO=0 para ajustar CR
90 when AJU_CR =>
91 if MO = '1' and PP='1' then
92 E <= INC_DP; -- incrementa distancia parcial
93 elsif MO='0' and IC='1' and DC='0' then
94 E <= INC_CR;
95 elsif MO='0' and IC='0' and DC='1' then
96 E <= DEC_CR;
97 elsif MO='0' and IC='0' and DC='0' then
98 E <= AJU_CR;
99 end if;
100 when INC_CR =>
101 E <= VER_CRM;
102 when VER_CRM =>
103 if PO='1' then E <= MAX_CR;
104 elsif PO='0' then E <= AJU_CR;
105 end if;
106 when MAX_CR => E <= AJU_CR;
107 when DEC_CR =>
108 if NG='0' then E <= AJU_CR;
109 elsif NG='1' then E <= MIN_CR;
110 end if;
111
```

CURSO DE ENGENHARIA ELÉTRICA

ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II

1º SEMESTRE DE 2021

```

111 when MIN_CR =>
112     E <= AJU_CR;
113 when INC_DP =>
114     E <= INC_DT;
115 when INC_DT =>
116     E <= VER_DT;
117 when VER_DT =>
118     if PO='1' then E <= SIN_LIM; SDM<='1';
119     elsif PO='0' then E <= VER_CR;
120     end if;
121 when SIN_LIM =>
122     E <= VER_CR;
123 when VER_CR =>
124     if NG='1' then E <= MUL_CR; MFC<='1';
125     elsif NG='0' then E <= VF15;
126     end if;
127 when MUL_CR =>
128     E <= VF15;
129 when VF15 =>
130     if PP='0' and RP='0' then E <= VF15;
131     elsif RP='1' and PP='0' then E <= VF16; MEV<='0';|
132     elsif PP='1' then E <= VER_VEL;
133     end if;
134 when VF16 =>
135     E <= VF15;
136 when VER_VEL =>
137     if FT='0' then E <= MUL_VEL; MEV<='1';
138     elsif FT='1' then E <= INC_DP;
139     end if;
140 when MUL_VEL => E <= INC_DP;
141
142 when others => Null;
143 end case;
144 end if;
145 end process;
146
147 -- Atualização das Saida para Fluxo de Dados (Multiplexadores, Registradores, ULA)
148 process(E)
149 begin
150     case E is
151     when ZER_RG => -- zera registros
152         Sel_mxa <= "xxx"; Sel_mxb <= "111"; Sel_ula <= "00";
153         Ldc <= '1'; Ldb <= '1'; Lda <= '1'; IT <= '1';
154     when CAR_CR => -- carrega CR com Credito Inicial
155         Sel_mxa <= "xxx"; Sel_mxb <= "011"; Sel_ula <= "00";
156         Ldc <= '0'; Ldb <= '0'; Lda <= '1'; IT <= '1';
157     when AJU_CR => -- espera IC e DC para ajustar CR
158         Sel_mxa <= "xxx"; Sel_mxb <= "xxx"; Sel_ula <= "xx";
159         Ldc <= '0'; Ldb <= '0'; Lda <= '0'; IT <= '1';
160
161     when INC_CR => -- incrementa CR
162         Sel_mxa <= "001"; Sel_mxb <= "000"; Sel_ula <= "10";
163         Ldc <= '0'; Ldb <= '0'; Lda <= '1'; IT <= '1';
164     when VER_CRM => -- verifica CRM
165         Sel_mxa <= "101"; Sel_mxb <= "000"; Sel_ula <= "11";
166         Ldc <= '0'; Ldb <= '0'; Lda <= '0'; IT <= '1';
167     when MAX_CR => -- maximo valor de CR
168         Sel_mxa <= "xxx"; Sel_mxb <= "101"; Sel_ula <= "00";
169         Ldc <= '0'; Ldb <= '0'; Lda <= '1'; IT <= '1';
170     when DEC_CR => -- decrementa CR
171         Sel_mxa <= "001"; Sel_mxb <= "000"; Sel_ula <= "11";
172         Ldc <= '0'; Ldb <= '0'; Lda <= '1'; IT <= '1';
173     when MIN_CR => -- minimo valor CR
174         Sel_mxa <= "xxx"; Sel_mxb <= "111"; Sel_ula <= "00";
175         Ldc <= '0'; Ldb <= '0'; Lda <= '1'; IT <= '1';
176     when INC_DP => -- incrementa DP
177         Sel_mxa <= "000"; Sel_mxb <= "001"; Sel_ula <= "10";
178         Ldc <= '0'; Ldb <= '1'; Lda <= '0'; IT <= '1';
179     when INC_DT => -- incrementa DP
180         Sel_mxa <= "000"; Sel_mxb <= "010"; Sel_ula <= "10";
181         Ldc <= '1'; Ldb <= '0'; Lda <= '0'; IT <= '1';
182     when VER_DT => -- verifica DT
183         Sel_mxa <= "111"; Sel_mxb <= "010"; Sel_ula <= "11";
184         Ldc <= '0'; Ldb <= '0'; Lda <= '0'; IT <= '1';
185     when SIN_LIM => -- sinaliza limite
186         Sel_mxa <= "xxx"; Sel_mxb <= "xxx"; Sel_ula <= "xx";
187         Ldc <= '0'; Ldb <= '0'; Lda <= '0'; IT <= '1';
188     when VER_CR => -- verifica CR
189         Sel_mxa <= "011"; Sel_mxb <= "000"; Sel_ula <= "11";
190         Ldc <= '0'; Ldb <= '0'; Lda <= '1'; IT <= '1';
191     when MUL_CR => -- multa CR
192         Sel_mxa <= "xxx"; Sel_mxb <= "111"; Sel_ula <= "00";
193         Ldc <= '0'; Ldb <= '0'; Lda <= '1'; IT <= '1';
194     when VF15 => -- espera PP ou RP e dispara timer
195         Sel_mxa <= "xxx"; Sel_mxb <= "xxx"; Sel_ula <= "xx";
196         Ldc <= '0'; Ldb <= '0'; Lda <= '0'; IT <= '0';
197     when VF16 => -- reinicia DP e MEV
198         Sel_mxa <= "xxx"; Sel_mxb <= "111"; Sel_ula <= "00";
199         Ldc <= '0'; Ldb <= '1'; Lda <= '0'; IT <= '0';
200     when VER_VEL => -- verifica velocidade
201         Sel_mxa <= "xxx"; Sel_mxb <= "xxx"; Sel_ula <= "xx";
202         Ldc <= '0'; Ldb <= '0'; Lda <= '0'; IT <= '0';
203     when MUL_VEL => -- incrementa DP
204         Sel_mxa <= "xxx"; Sel_mxb <= "xxx"; Sel_ula <= "xx";
205         Ldc <= '0'; Ldb <= '0'; Lda <= '0'; IT <= '0';
206
207     when others => Null;
208     end case;
209 end process;
210 end FSM;

```

CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

7. Código da Junção da UC com FD

```

1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  ENTITY juncao_2FV IS PORT(
5      PPP, ICC, DCC, CKK, RTT, RPP, MOO: IN std_logic;
6      LD1, LD2, LD3: OUT std_logic;
7      REGA, REGB, REGC: OUT std_logic_vector(7 downto 0);
8      ITT, FTT: buffer std_logic);
9
10 END juncao_2FV;
11
12 ARCHITECTURE BEHAVIOR OF juncao_2FV IS
13
14     SIGNAL Z, P, N, E: std_logic;
15     SIGNAL SA, SB: std_logic_vector(2 DOWNTO 0);
16     SIGNAL SU: std_logic_vector(1 DOWNTO 0);
17     SIGNAL LA, LB, LC: std_logic;
18     SIGNAL CR, DT, DP: std_logic_vector(7 downto 0);
19
20     component FD IS PORT(
21         S_MXA, S_MXB: IN std_logic_vector(2 DOWNTO 0);
22         S_ULA: IN std_logic_vector(1 DOWNTO 0);
23         LDA, LDB, LDC: IN std_logic;
24         IT: IN std_logic;
25         CK: IN std_logic;
26         FT: OUT std_logic;
27         ZR, PO, NG, ES: OUT std_logic;
28         REG_A, REG_B, REG_C: OUT std_logic_vector(7 downto 0));
29     end component FD;
30
31     component UC is
32     port(
33         CK : in std_logic;           -- clock de 50MHz
34         RT : in std_logic;           -- reinicio total -> ativo em zero
35
36         -- Entradas Externas:
37         IC : in std_logic;           -- incrementa credito -> ativo em um
38         DC : in std_logic;           -- decrementa credito -> ativo em um
39         RP : in std_logic;           -- reinicio parcial -> ativo em um
40         PP : in std_logic;           -- passagem pelo portico -> ativo em um
41         MO : in std_logic;           -- modo de operação -> 1= Normal, 0= Ajusta valor
42
43         -- Sinais de Estado da ULA:
44         ZR : in std_logic;           -- resultado zero na operação da ULA
45         PO : in std_logic;           -- resultado positivo na operação da ULA
46         NG : in std_logic;           -- resultado negativo na operação da ULA
47         ES : in std_logic;           -- resultado da operação da ULA maior que 255
48
49         -- Sinais de Estado do TIMER:
50         FT : in std_logic;           -- fim da temporização de 1 segundo
51
52         -- Sinais de Saída para MUX:
53         Sel_mxa : out std_logic_vector(2 downto 0); -- seleciona entrada de MUX_A
54         Sel_mxb : out std_logic_vector(2 downto 0); -- seleciona entrada de MUX_B
55
56         -- Sinais de Saída para ULA:
57         Sel_ula : out std_logic_vector(1 downto 0); -- seleciona operação da ULA
58
59         -- Sinais de Saída para Registradores:
60         Lda : out std_logic;         -- carrega RA
61         Ldb : out std_logic;         -- carrega RB
62         Ldc : out std_logic;         -- carrega RC
63
64         -- Sinais de Saída para TIMER:
65         IT : out std_logic;          -- inicia temporização
66
67         -- Sinais de Saída Externos:
68         MEV : out std_logic;         -- sinaliza multa por excesso de velocidade
69         MFC : out std_logic;         -- sinaliza multa por falta de crédito
70         SDM : out std_logic;         -- sinaliza limite de distância total percorrida
71     );
72 end component UC;
73
74 begin
75
76     FD1 : FD PORT MAP (S_MXA=>SA, S_MXB=>SB, S_ULA=>SU, CK=>CKK, LDA=>LA, LDB=>LB, LDC=>LC, IT=> ITT, FT=> FTT,
77         ZR=>Z, PO=>P, NG=>N, ES=>E, REG_A=>REGA, REG_B=> REGB, REG_C=>REGC);
78
79     UC1 : UC PORT MAP (PP=>PPP, IC=>ICC, DC=>DCC, CK=>CKK, RT=>RTT, RP=>RPP, MO=>MOO, ZR=>Z, PO=>P, NG=>N, ES=>E,
80         FT=>FTT, IT=>ITT, Sel_mxa=>SA, Sel_mxb=>SB, Sel_ula=>SU, Lda=>LA, Ldb=>LB, Ldc=>LC, MEV=>LD1, MFC=>LD2, SDM=>LD3);
81
82 END BEHAVIOR;

```

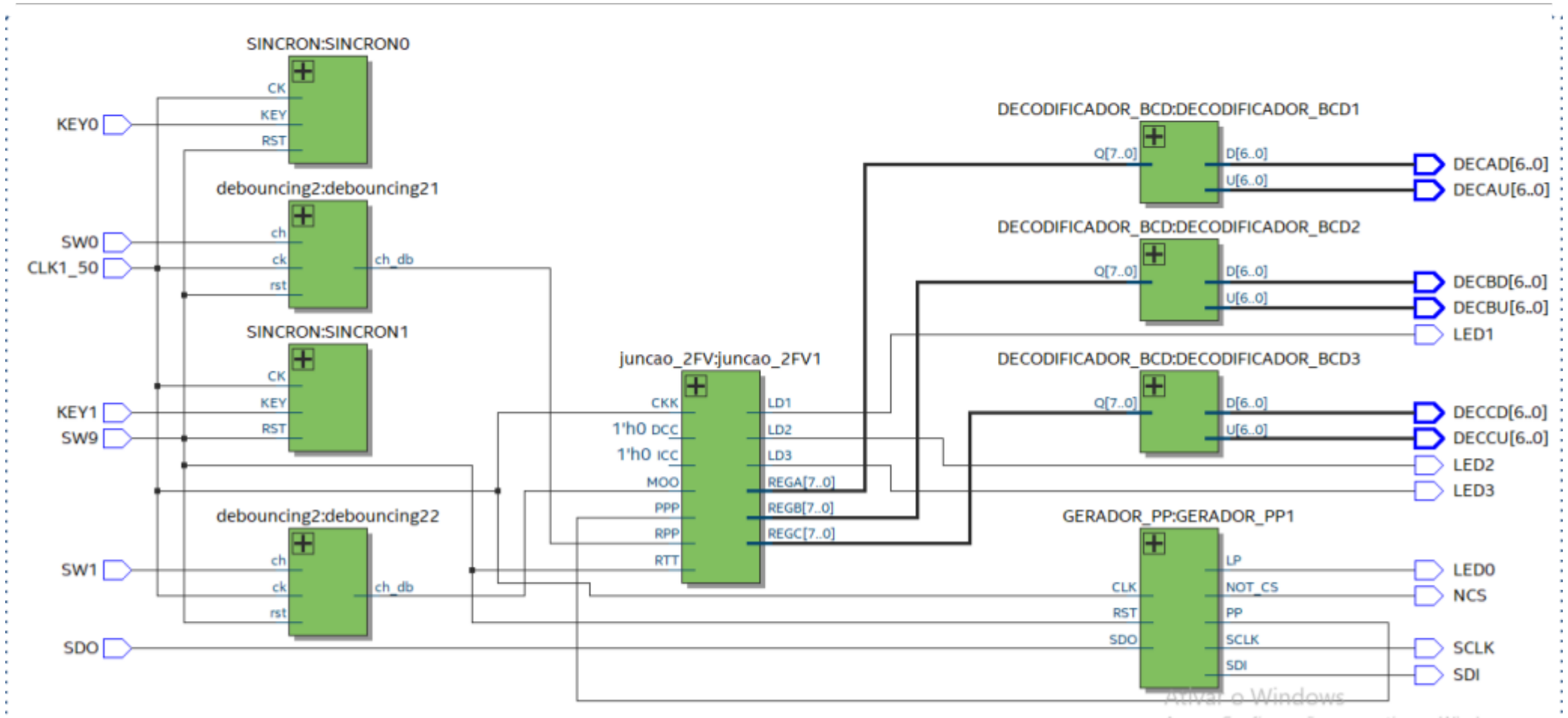

CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

8. Código VHDL do Projeto Completo

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3
4  ENTITY projeto2final IS PORT(
5      SDO, KEY0, KEY1, CLK1_50, SW9, SW0, SW1 : IN STD_LOGIC;
6      NCS, SDI, SCLK : OUT STD_LOGIC;
7      LED0, LED1, LED2, LED3 : OUT STD_LOGIC;
8      DECAD, DECB0, DECCD, DECAU, DECBU, DECCU : OUT STD_LOGIC_VECTOR(6 DOWNTO 0) );
9
10 END projeto2final;
11 ARCHITECTURE BEHAVIOR OF projeto2final IS
12     SIGNAL P, IC, DC, RT, RP, MO, MEV, MFC, SDM, SI, DB : STD_LOGIC;
13     SIGNAL AD, BD, CD : std_logic_vector(7 downto 0);
14
15
16
17     COMPONENT juncao_2FV IS PORT(
18         PPP, ICC, DCC, CKK, RTT, RPP, MOO: IN std_logic;
19         LD1, LD2, LD3: OUT STD_LOGIC;
20         REGA, REGB, REGC: OUT std_logic_vector(7 downto 0));
21
22     END COMPONENT juncao_2FV;
23
24     COMPONENT SINCRON IS
25     PORT ( CK      : IN STD_LOGIC;    -- clock de 50MHz
26           RST      : IN STD_LOGIC;    -- reset (ativo em zero)
27           KEY       : IN STD_LOGIC;   -- botao de entrada (ativo em zero)
28           SINC      : OUT STD_LOGIC ); -- pulso de sinal de saída (ativo em um)
29     END COMPONENT SINCRON;
30
31     COMPONENT DECODIFICADOR_BCD IS -- declaracao da entidade DECODIFICADOR_BCD
32     PORT
33     ( Q : IN STD_LOGIC_VECTOR(7 DOWNTO 0); -- vetor de 8 bits de entrada
34       D, U : OUT STD_LOGIC_VECTOR(6 DOWNTO 0)); -- vetores dezena e unidade
35     END COMPONENT DECODIFICADOR_BCD;
36
37     COMPONENT debouncing2 is
38     port
39     ( ck      : in std_logic;          -- periodo de referencia
40       rst      : in std_logic;        -- iniciacao da contagem
41       ch       : in std_logic;        -- sinal com oscilacao
42       ch_db    : out std_logic);      -- sinal sem oscilacao
43     END COMPONENT debouncing2;
44
45     COMPONENT GERADOR_PP IS          -- declaracao da entidade GEADORE_PP
46     PORT
47     ( CLK, RST: IN STD_LOGIC;        -- sinais de controle
48       PP : OUT STD_LOGIC;            -- puse de pórtoico
49       LP : BUFFER STD_LOGIC;         -- led de sinalização de frequência
50       SDO: IN STD_LOGIC;             -- entrada de dados do sensor ADXL345
51       SCLK,SDI,NOT_CS: OUT STD_LOGIC);--sinais de comunicação com sensor ADXL345
52     END COMPONENT GERADOR_PP;
53 BEGIN
54
55     GERADOR_PP1 : GERADOR_PP PORT MAP ( SDO=>SDO, CLK=>clk1_50, RST=>SW9, NOT_CS=>NCS, SDI=>SDI, SCLK=>SCLK, LP=>LED0, PP=>P);
56     SINCRON0 : SINCRON PORT MAP ( KEY=>KEY0, CK=>clk1_50, RST=>SW9, SINC=>IC);
57     SINCRON1 : SINCRON PORT MAP ( KEY=>KEY1, CK=>clk1_50, RST=>SW9, SINC=>DC);
58
59     debouncing21 : debouncing2 PORT MAP ( ch=>SW0, ck=>clk1_50, rst=>SW9, ch_db=>RP);
60     debouncing22 : debouncing2 PORT MAP ( ch=>SW1, ck=>clk1_50, rst=>SW9, ch_db=>MO);
61
62     juncao_2FV1 : juncao_2FV PORT MAP ( PPP=>P, ICC=>SI, DCC=>SI, CKK=>clk1_50, RTT=>SW9, RPP=>RP,
63     MOO=>MO, LD1=>LED1, LD2=>LED2, LD3=>LED3, REGA=>AD, REGB=>BD, REGC=>CD);
64
65     DECODIFICADOR_BCD1 : DECODIFICADOR_BCD PORT MAP (Q=>AD, D=>DECAD, U=>DECAU);
66     DECODIFICADOR_BCD2 : DECODIFICADOR_BCD PORT MAP (Q=>BD, D=>DECB0, U=>DECBU);
67     DECODIFICADOR_BCD3 : DECODIFICADOR_BCD PORT MAP (Q=>CD, D=>DECCD, U=>DECCU);
68 END BEHAVIOR;
```


CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

9. Diagrama RTL do Projeto Completo



CURSO DE ENGENHARIA ELÉTRICA

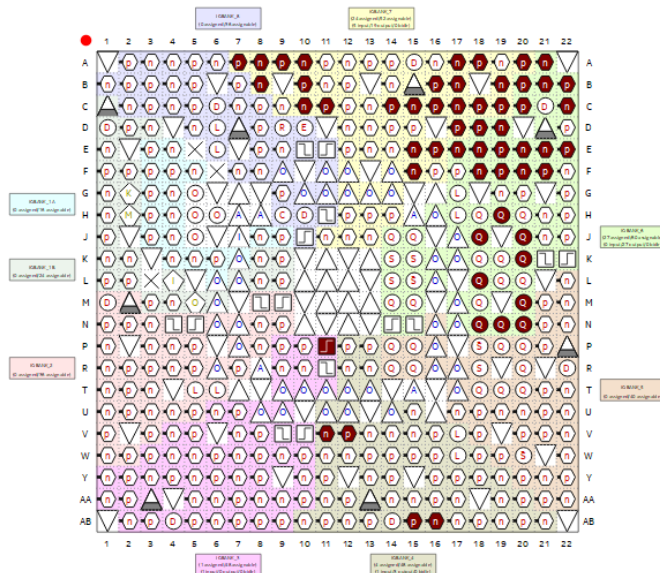
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II

1º SEMESTRE DE 2021

11. Pinagem do Projeto(Pin Planner)

Top View - Wire Bond

MAX 10 - 10M50DAF484C7G



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
in CLK1_50	Input	PIN_P11	3	B3_N0	PIN_P11	2.5 V		12mA (default)			
out DECAU[6]	Output	PIN_B17	7	B7_N0	PIN_B17	2.5 V		12mA (default)	2 (default)		
out DECAU[5]	Output	PIN_A18	7	B7_N0	PIN_A18	2.5 V		12mA (default)	2 (default)		
out DECAU[4]	Output	PIN_A17	7	B7_N0	PIN_A17	2.5 V		12mA (default)	2 (default)		
out DECAU[3]	Output	PIN_B16	7	B7_N0	PIN_B16	2.5 V		12mA (default)	2 (default)		
out DECAU[2]	Output	PIN_E18	6	B6_N0	PIN_E18	2.5 V		12mA (default)	2 (default)		
out DECAU[1]	Output	PIN_D18	6	B6_N0	PIN_D18	2.5 V		12mA (default)	2 (default)		
out DECAU[0]	Output	PIN_C18	7	B7_N0	PIN_C18	2.5 V		12mA (default)	2 (default)		
out DECAU[6]	Output	PIN_C17	7	B7_N0	PIN_C17	2.5 V		12mA (default)	2 (default)		
out DECAU[5]	Output	PIN_D17	7	B7_N0	PIN_D17	2.5 V		12mA (default)	2 (default)		
out DECAU[4]	Output	PIN_E16	7	B7_N0	PIN_E16	2.5 V		12mA (default)	2 (default)		
out DECAU[3]	Output	PIN_C16	7	B7_N0	PIN_C16	2.5 V		12mA (default)	2 (default)		
out DECAU[2]	Output	PIN_C15	7	B7_N0	PIN_C15	2.5 V		12mA (default)	2 (default)		
out DECAU[1]	Output	PIN_E15	7	B7_N0	PIN_E15	2.5 V		12mA (default)	2 (default)		
out DECAU[0]	Output	PIN_C14	7	B7_N0	PIN_C14	2.5 V		12mA (default)	2 (default)		
out DECB[6]	Output	PIN_E17	6	B6_N0	PIN_E17	2.5 V		12mA (default)	2 (default)		
out DECB[5]	Output	PIN_D19	6	B6_N0	PIN_D19	2.5 V		12mA (default)	2 (default)		
out DECB[4]	Output	PIN_C20	6	B6_N0	PIN_C20	2.5 V		12mA (default)	2 (default)		
out DECB[3]	Output	PIN_C19	7	B7_N0	PIN_C19	2.5 V		12mA (default)	2 (default)		
out DECB[2]	Output	PIN_E21	6	B6_N0	PIN_E21	2.5 V		12mA (default)	2 (default)		

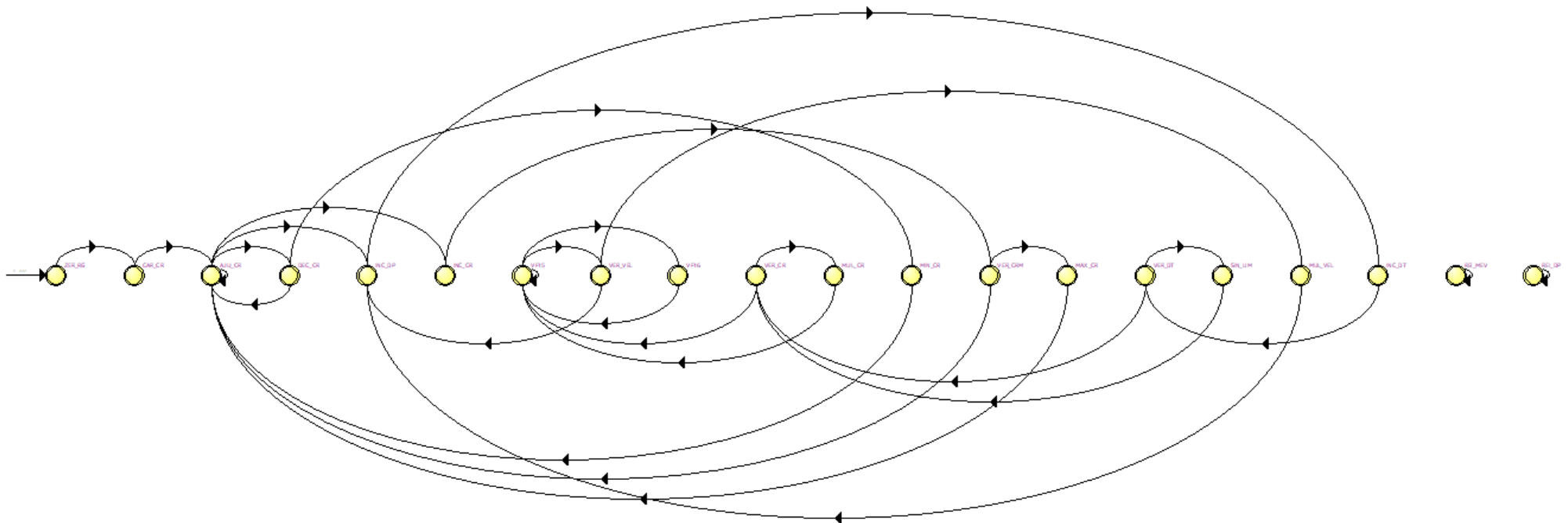
CURSO DE ENGENHARIA ELÉTRICA

ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II

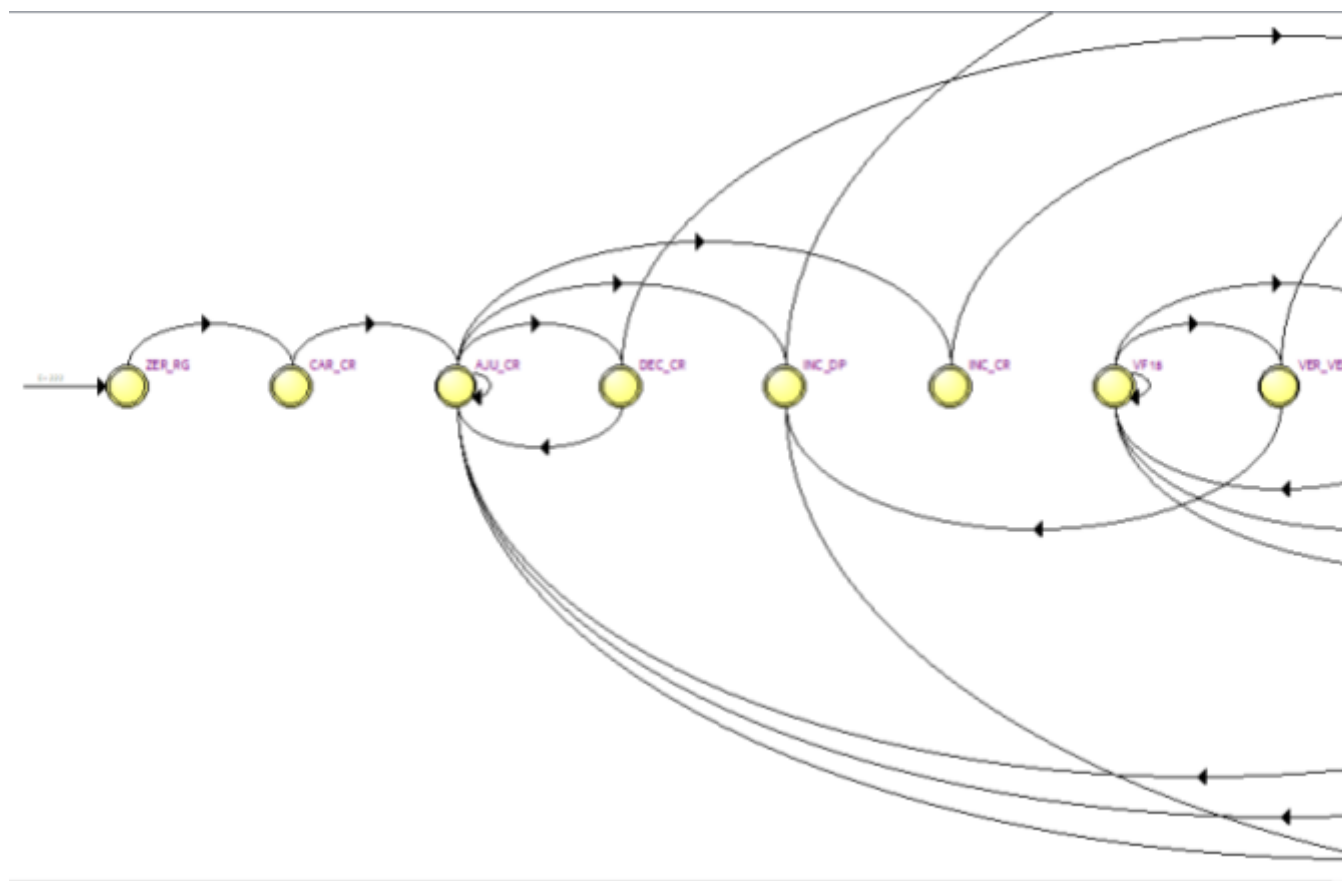
1º SEMESTRE DE 2021

out	DECB0[1]	Output	PIN_E22	6	B6_NO	PIN_E22	2.5 V		12mA (default)	2 (default)		
out	DECB0[0]	Output	PIN_F21	6	B6_NO	PIN_F21	2.5 V		12mA (default)	2 (default)		
out	DECBU[6]	Output	PIN_B22	6	B6_NO	PIN_B22	2.5 V		12mA (default)	2 (default)		
out	DECBU[5]	Output	PIN_C22	6	B6_NO	PIN_C22	2.5 V		12mA (default)	2 (default)		
out	DECBU[4]	Output	PIN_B21	6	B6_NO	PIN_B21	2.5 V		12mA (default)	2 (default)		
out	DECBU[3]	Output	PIN_A21	6	B6_NO	PIN_A21	2.5 V		12mA (default)	2 (default)		
out	DECBU[2]	Output	PIN_B19	7	B7_NO	PIN_B19	2.5 V		12mA (default)	2 (default)		
out	DECBU[1]	Output	PIN_A20	7	B7_NO	PIN_A20	2.5 V		12mA (default)	2 (default)		
out	DECBU[0]	Output	PIN_B20	6	B6_NO	PIN_B20	2.5 V		12mA (default)	2 (default)		
out	DECCD[6]	Output	PIN_N20	6	B6_NO	PIN_N20	2.5 V		12mA (default)	2 (default)		
out	DECCD[5]	Output	PIN_N19	6	B6_NO	PIN_N19	2.5 V		12mA (default)	2 (default)		
out	DECCD[4]	Output	PIN_M20	6	B6_NO	PIN_M20	2.5 V		12mA (default)	2 (default)		
out	DECCD[3]	Output	PIN_N18	6	B6_NO	PIN_N18	2.5 V		12mA (default)	2 (default)		
out	DECCD[2]	Output	PIN_L18	6	B6_NO	PIN_L18	2.5 V		12mA (default)	2 (default)		
out	DECCD[1]	Output	PIN_K20	6	B6_NO	PIN_K20	2.5 V		12mA (default)	2 (default)		
out	DECCD[0]	Output	PIN_J20	6	B6_NO	PIN_J20	2.5 V		12mA (default)	2 (default)		
out	DECCU[6]	Output	PIN_F20	6	B6_NO	PIN_F20	2.5 V		12mA (default)	2 (default)		
out	DECCU[5]	Output	PIN_F19	6	B6_NO	PIN_F19	2.5 V		12mA (default)	2 (default)		
out	DECCU[4]	Output	PIN_H19	6	B6_NO	PIN_H19	2.5 V		12mA (default)	2 (default)		
out	DECCU[3]	Output	PIN_J18	6	B6_NO	PIN_J18	2.5 V		12mA (default)	2 (default)		
out	DECCU[2]	Output	PIN_E19	6	B6_NO	PIN_E19	2.5 V		12mA (default)	2 (default)		
out	DECCU[1]	Output	PIN_E20	6	B6_NO	PIN_E20	2.5 V		12mA (default)	2 (default)		
out	DECCU[0]	Output	PIN_F18	6	B6_NO	PIN_F18	2.5 V		12mA (default)	2 (default)		
in	KEY0	Input	PIN_B8	7	B7_NO	PIN_B8	2.5 V		12mA (default)			
in	KEY1	Input	PIN_A7	7	B7_NO	PIN_A7	2.5 V		12mA (default)			
out	LED0	Output	PIN_A8	7	B7_NO	PIN_A8	2.5 V		12mA (default)	2 (default)		
out	LED1	Output	PIN_A9	7	B7_NO	PIN_A9	2.5 V		12mA (default)	2 (default)		
out	LED2	Output	PIN_A10	7	B7_NO	PIN_A10	2.5 V		12mA (default)	2 (default)		
out	LED3	Output	PIN_B10	7	B7_NO	PIN_B10	2.5 V		12mA (default)	2 (default)		
out	NCS	Output	PIN_AB16	4	B4_NO	PIN_AB16	2.5 V		12mA (default)	2 (default)		
out	SCLK	Output	PIN_AB15	4	B4_NO	PIN_AB15	2.5 V		12mA (default)	2 (default)		
out	SDI	Output	PIN_V11	4	B4_NO	PIN_V11	2.5 V		12mA (default)	2 (default)		
in	SDO	Input	PIN_V12	4	B4_NO	PIN_V12	2.5 V		12mA (default)			
in	SW0	Input	PIN_C10	7	B7_NO	PIN_C10	2.5 V		12mA (default)			
in	SW1	Input	PIN_C11	7	B7_NO	PIN_C11	2.5 V		12mA (default)			
in	SW9	Input	PIN_F15	7	B7_NO	PIN_F15	2.5 V		12mA (default)			
<<new node>>												

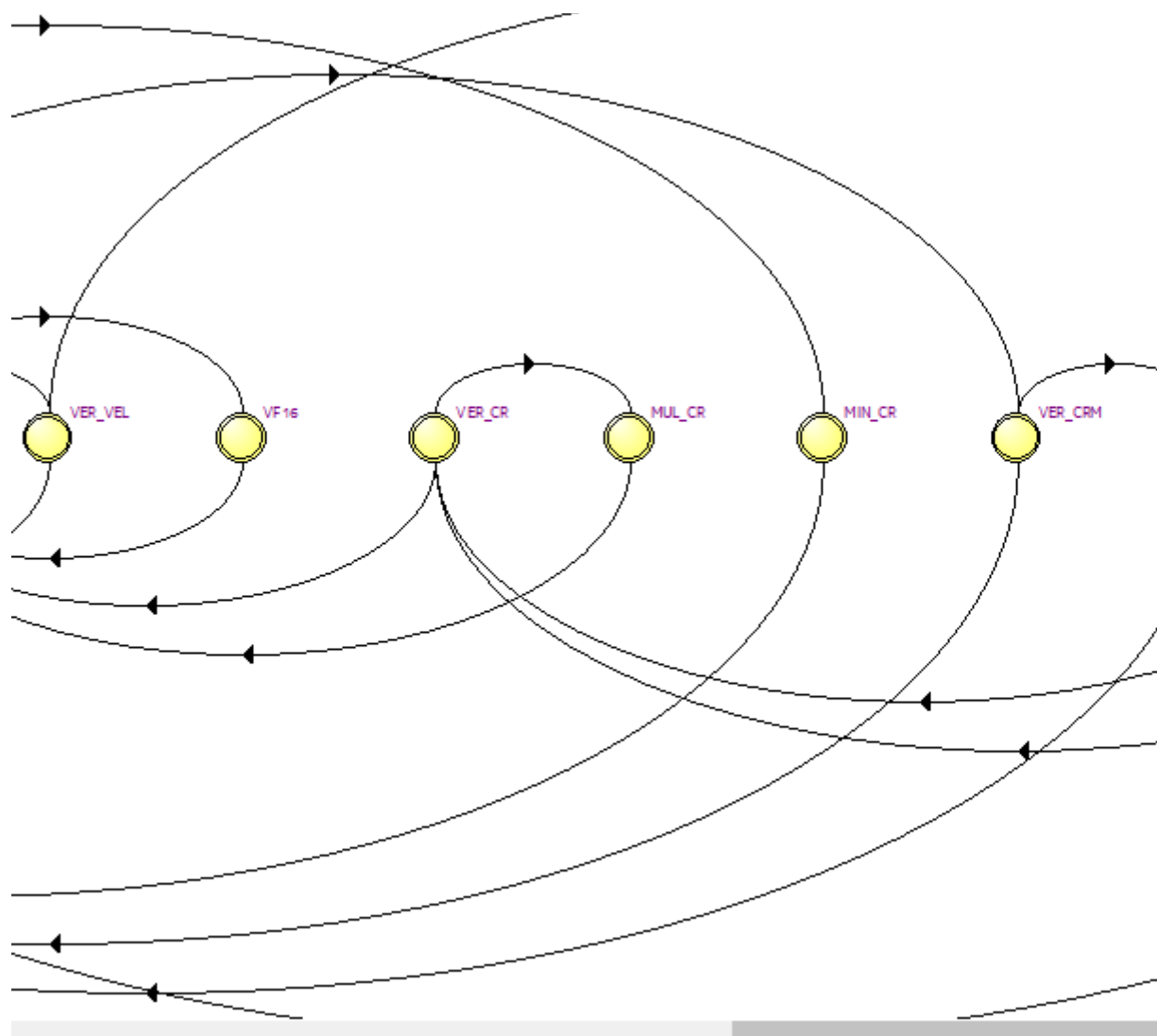
12. Diagrama de Estados da UC gerado pelo Quartus Prime



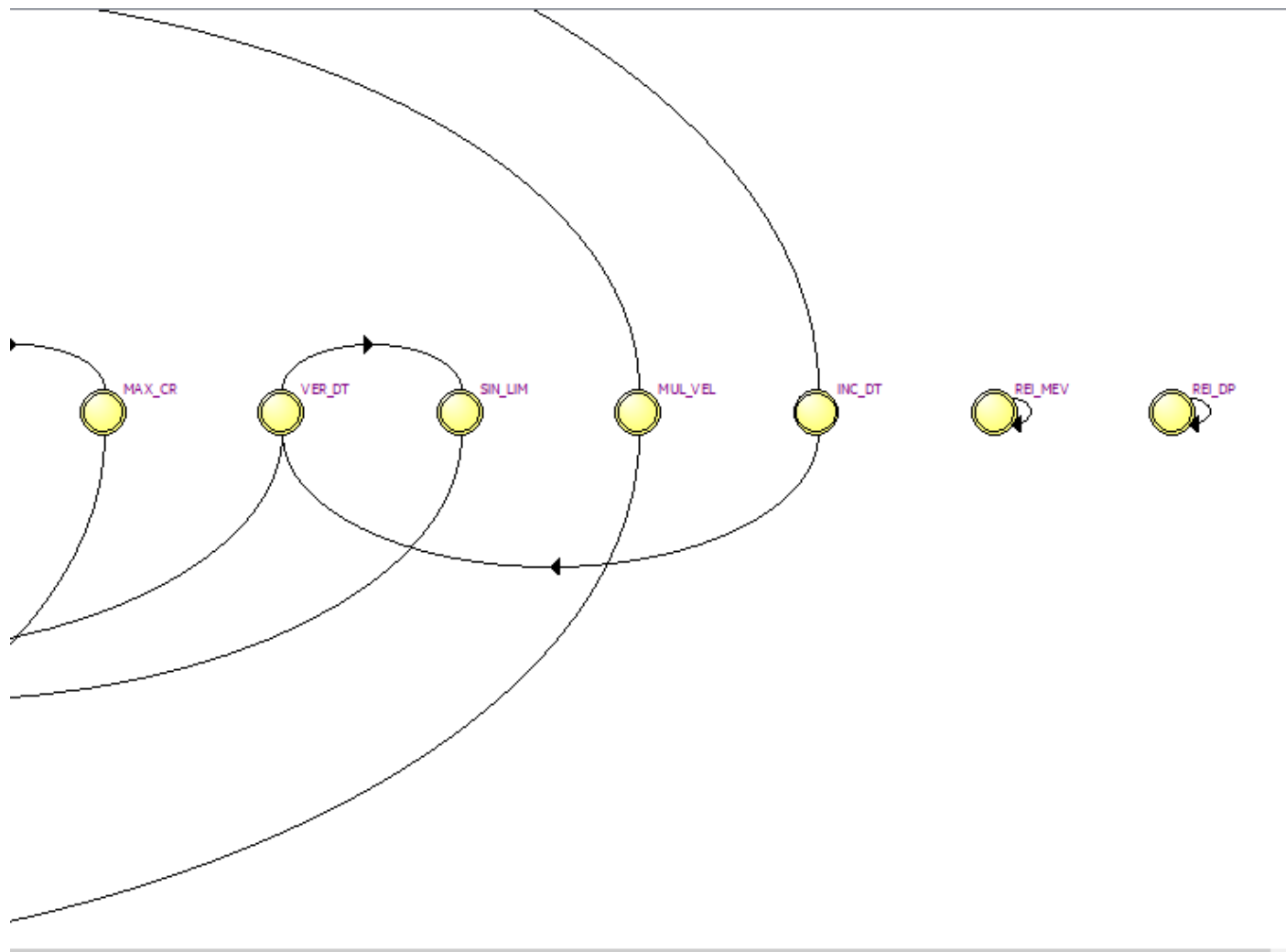
CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021



CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

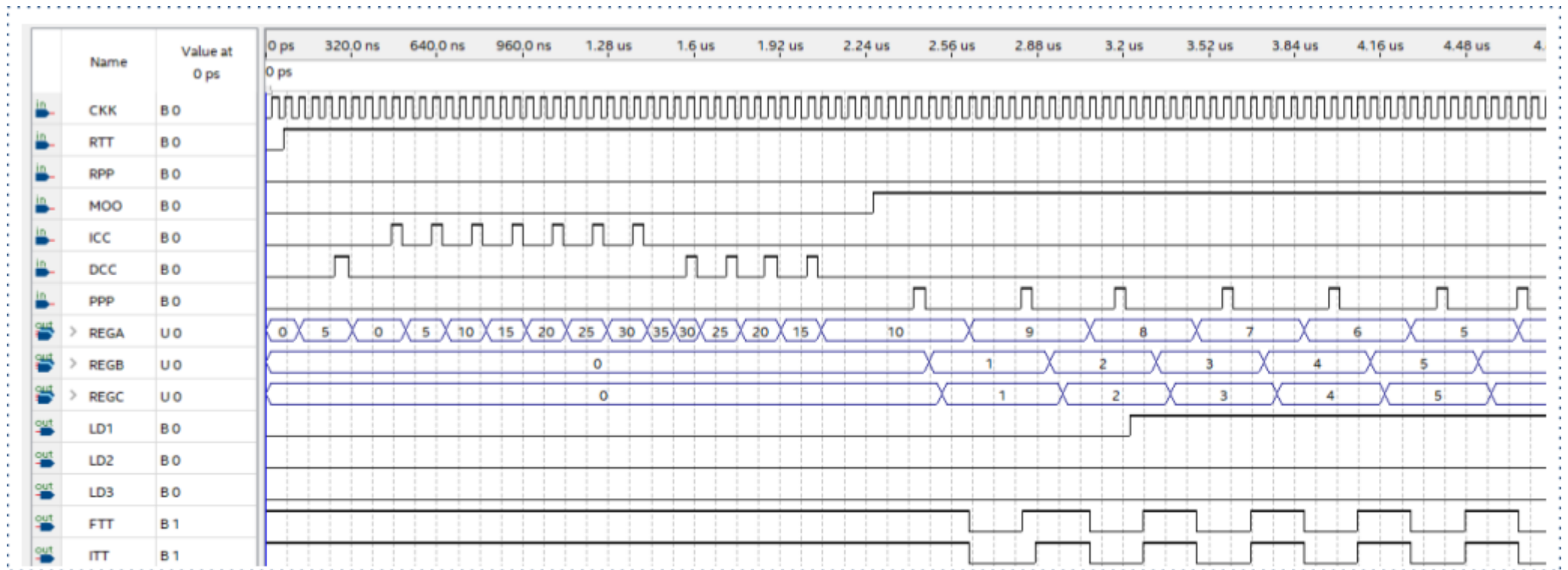


CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

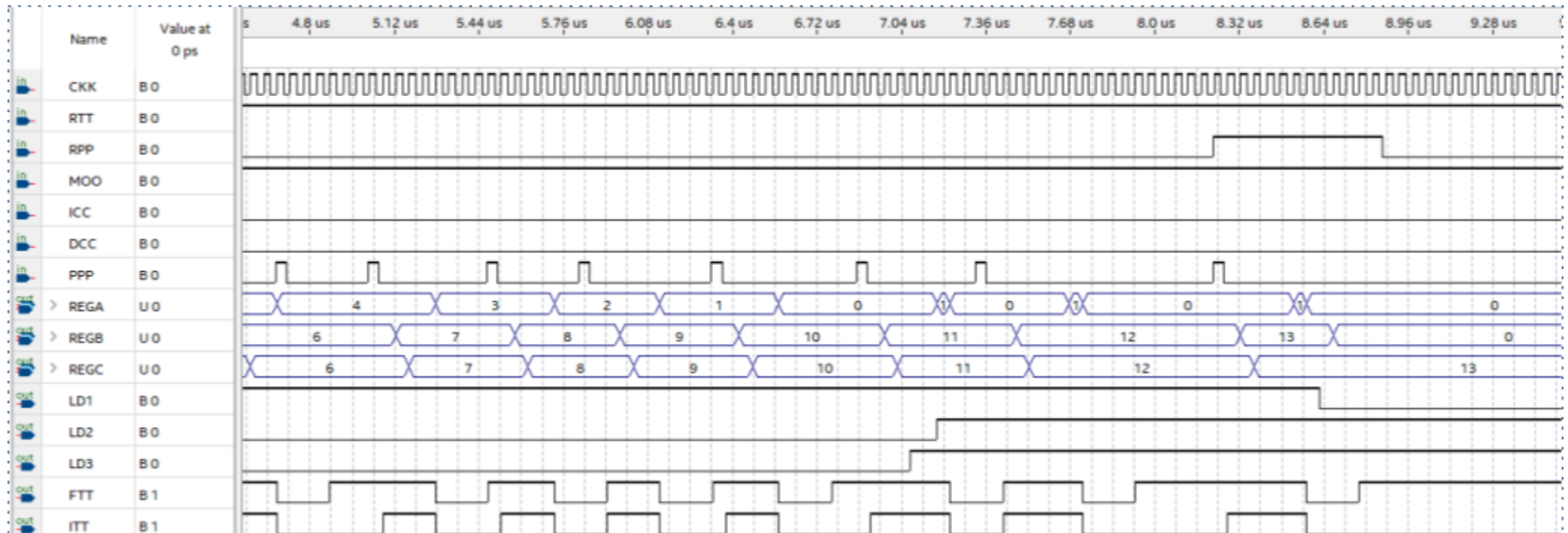


CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

12.Simulação Funcional da UC + FD



CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021



CURSO DE ENGENHARIA ELÉTRICA
ELD020 – LABORATÓRIO DE SISTEMAS DIGITAIS II
1º SEMESTRE DE 2021

13. Conclusão

Após a execução do projeto conseguimos afirmar que colocamos em prática o conhecimento que é passado na teoria, desde o planejamento do projeto com a utilização de ferramentas e métodos que auxiliam no desenvolvimento e na simulação da sua funcionalidade a partir da descrição de hardware com a linguagem VHDL.

Sentimos dificuldades na elaboração da máquina de estados, na criação do código VHDL para que este funcionasse exatamente como o esperado e na simulação, entretanto, a partir da simulação conseguimos identificar aonde estava os erros na máquina de estados e no VHDL, junto com as orientações da professora. Exercitamos conceitos importantes para a criação de projetos de sistemas digitais, tais como, fluxo de dados, unidade de controle, banco de registradores e metodologia de projetos RTL.

Se fossemos realizar o projeto novamente leríamos o roteiro com mais atenção e levaríamos mais dúvidas às aulas. Para futuros projetos nos manteríamos esse processo da realização em parte, isso faz com que os erros sejam resolvidos aos poucos fazendo com que no final do projeto os erros sejam bem menores.

Gostaríamos de agradecer o professor de teoria Valter Avelino e a professora de laboratório Maria Claudia por ensinar a gente a matéria de Sistemas Digitais 2 e por ter paciência e dedicação aos alunos.