

UFERN



Projeto Máquina de Sorvete

Circuitos Digitais

Efrain Marcelo Pulgar Pantaleon

Fernando Lucas Sousa Silva

Matheus Gomes Diniz Andrade

Teóphilo Vítor de Carvalho Clemente

DCA

Máquina de Sorvete

A máquina produtora de sorvete consiste em uma importante ferramenta utilizada para produzir comercialmente sorvetes de massa, picolés, gelatos e entre outros tipos. Em seu interior, os materiais serão ao mesmo tempo batidos e congelados, o que lhes dará a consistência adequada para serem vendidos.



Design RTL

CAPTURA (HLSM)

Criação da HLSM que descreve o comportamento da máquina de estados de alto nível.

DATAPATH

Executa as operações da máquina de estados de alto nível.

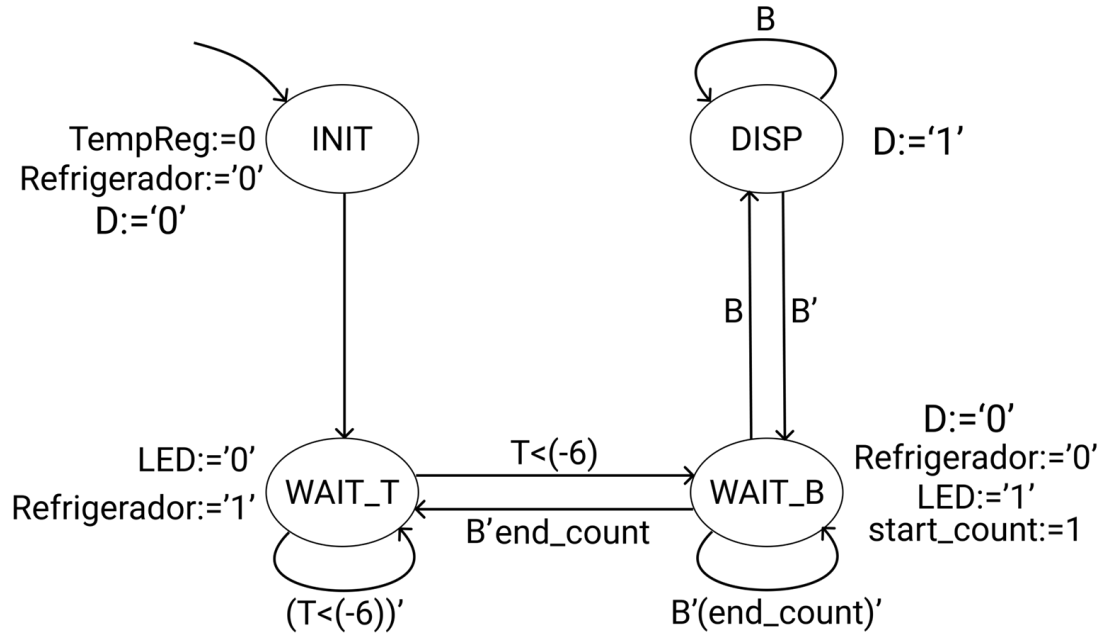
CONEXÕES

Promove as conexões internas e externas entre os sinais e os blocos.

FSM

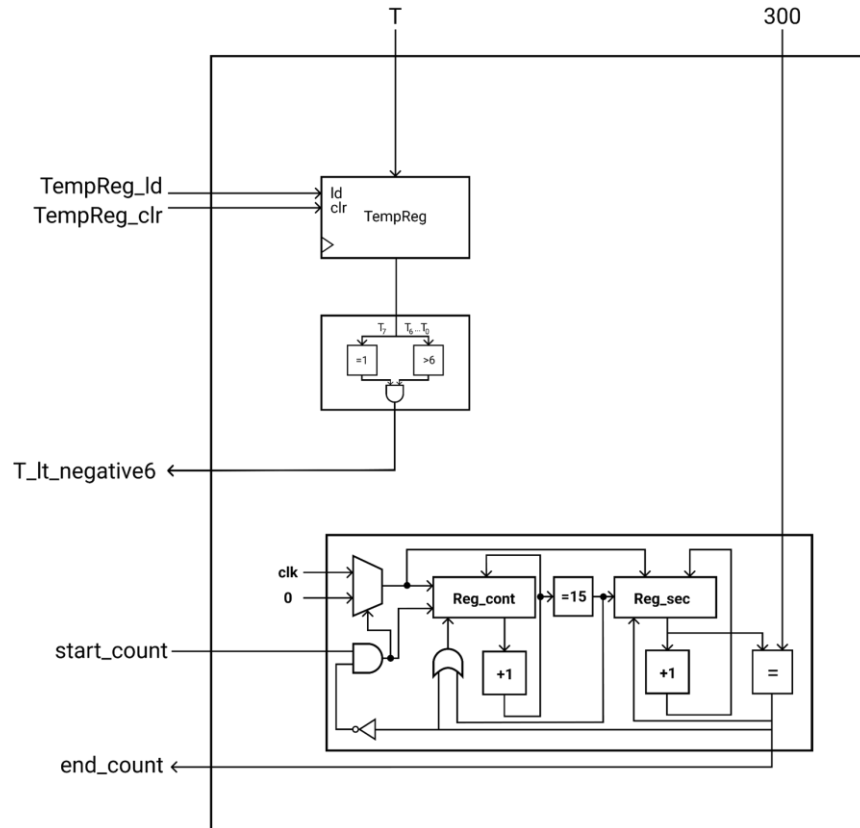
Converte a HLSM em uma máquina de estados finitos.

Captura (HLSM)



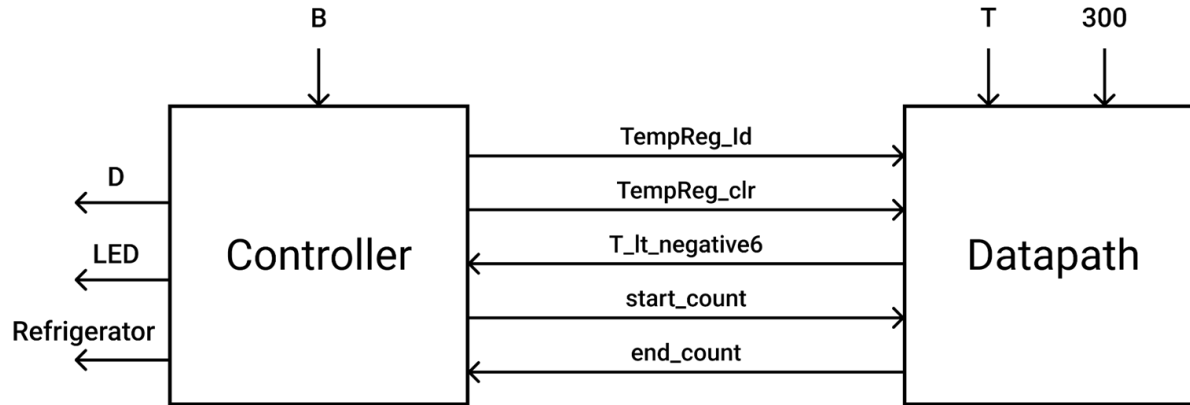
- Máquina de estados de alto nível
- Instantes dos processos

Datapath



- Caminho da dados
- Temporizador

Conexões



Inputs:

- TempReg_Id
- TempReg_clr
- start_count
- end_count
- T_lt_negative6
- B = Botão

Outputs:

- LED = Luz para indicar a permissão de uso;
- Refrigerador = Refrigera o sorvete;
- D = Dispensador.

Captura (FSM)

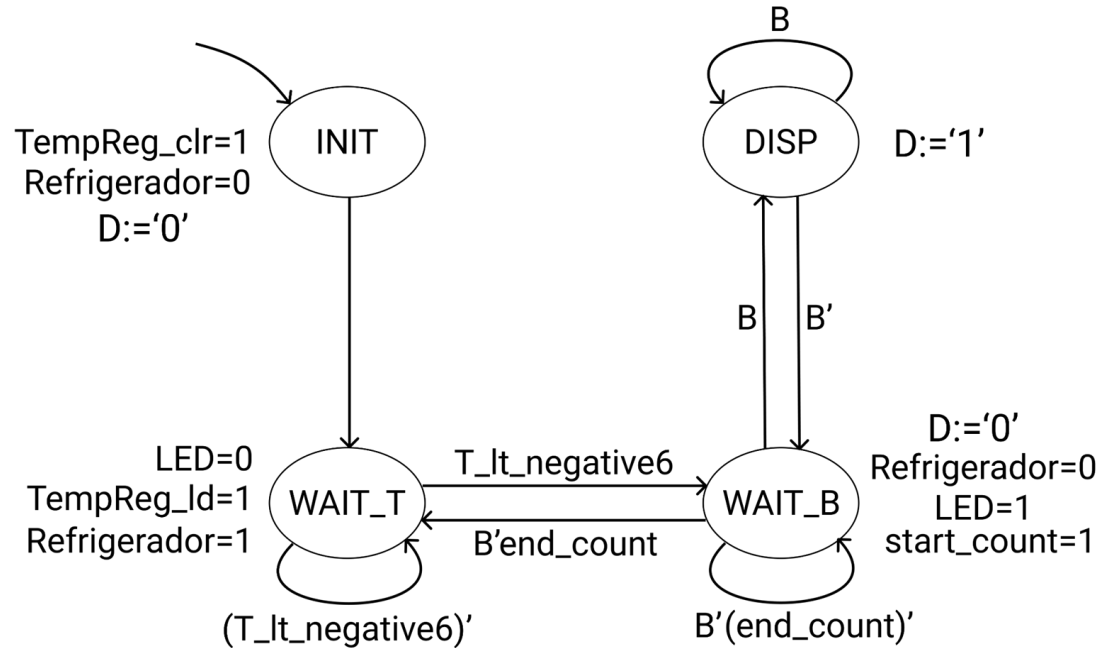


Tabela Verdade

	Inputs					Outputs								
	T_lt_negative6	end_count	B	s0	s1	n0	n1	TempReg_clr	TempReg_ld	start_count	LED	Refrigerador	D	
INIT	0	0	0	0	0	0	1	1	0	0	0	0	0	
	0	0	1	0	0	0	1	1	0	0	0	0	0	
	0	1	0	0	0	0	1	1	0	0	0	0	0	
	0	1	1	0	0	0	1	1	0	0	0	0	0	
	1	0	0	0	0	0	1	1	0	0	0	0	0	
	1	0	1	0	0	0	1	1	0	0	0	0	0	
	1	1	0	0	0	0	1	1	0	0	0	0	0	
	1	1	1	0	0	0	1	1	0	0	0	0	0	
WAIT_T	0	0	0	0	1	0	1	0	1	0	0	1	0	
	0	0	1	0	1	0	1	0	1	0	0	1	0	
	0	1	0	0	1	0	1	0	1	0	0	1	0	
	0	1	1	0	1	0	1	0	1	0	0	1	0	
	1	0	0	0	1	1	0	0	1	0	0	1	0	
	1	0	1	0	1	1	0	0	1	0	0	1	0	
	1	1	0	0	1	1	0	0	1	0	0	1	0	
	1	1	1	0	1	1	0	0	1	0	0	1	0	
WAIT_B	0	0	0	1	0	1	0	0	0	1	1	0	0	
	0	0	1	1	0	1	1	0	0	1	1	0	0	
	0	1	0	1	0	0	1	0	0	1	1	0	0	
	0	1	1	1	0	1	1	0	0	1	1	0	0	
	1	0	0	1	0	1	0	0	0	1	1	0	0	
	1	0	1	1	0	1	1	0	0	1	1	0	0	
	1	1	0	1	0	0	1	0	0	1	1	0	0	
	1	1	1	1	0	1	1	0	0	1	1	0	0	
DISP	0	0	0	1	1	1	0	0	0	0	1	0	1	
	0	0	1	1	1	1	1	0	0	0	1	0	1	
	0	1	0	1	1	1	0	0	0	0	1	0	1	
	0	1	1	1	1	1	1	0	0	0	1	0	1	
	1	0	0	1	1	1	0	0	0	0	1	0	1	
	1	0	1	1	1	1	1	0	0	0	1	0	1	
	1	1	0	1	1	1	0	0	0	0	1	0	1	
	1	1	1	1	1	1	1	0	0	1	0	1		

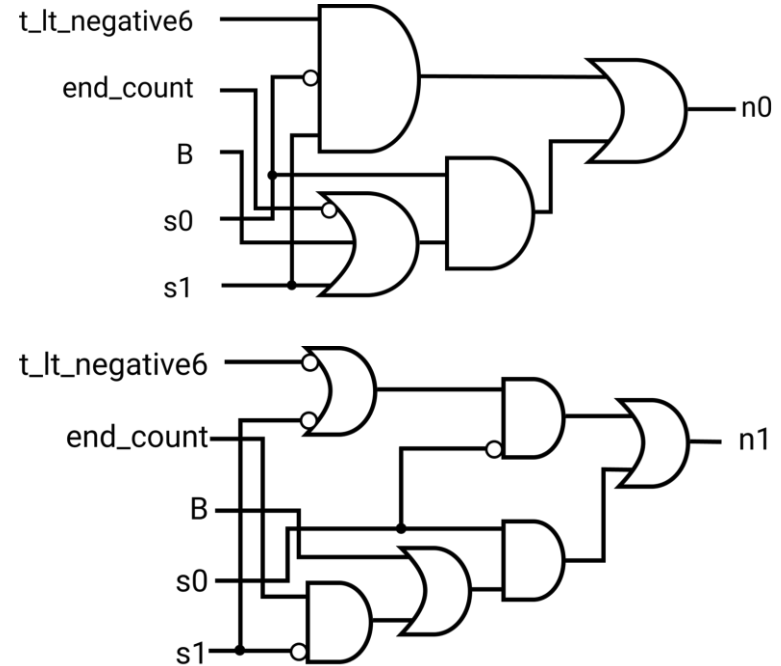
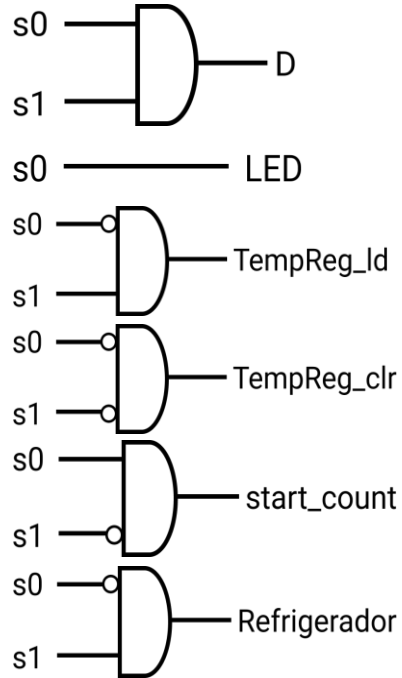
Tabela Verdade Simplificada

	Inputs					Outputs							
	T_lt_negative6	end_count	B	s0	s1	n0	n1	TempReg_clr	TempReg_Id	start_count	LED	Refrigerador	D
INIT	x	x	x	0	0	0	1	1	0	0	0	0	0
WAIT_T	0	x	x	0	1	0	1	0	1	0	0	1	0
	1	x	x	0	1	1	0	0	1	0	0	1	0
WAIT_B	x	0	0	1	0	1	0	0	0	1	1	0	0
	x	x	1	1	0	1	1	0	0	1	1	0	0
	x	1	0	1	0	0	1	0	0	1	1	0	0
DISP	x	x	0	1	1	1	0	0	0	0	1	0	1
	x	x	1	1	1	1	1	0	0	0	1	0	1

Equações Simplificadas

- $n0 = s1 \cdot s0' \cdot t_lt_negative6 + s0 \cdot (end_count' + B + s1)$
- $n1 = s0' \cdot (s1' + t_lt_negative6') + s0 \cdot (B + s1'end_count)$
- $TempReg_clr = s0' \cdot s1$
- $TempReg_ld = s0' \cdot s1'$
- $start_count = s0 \cdot s1'$
- $LED = s0$
- $Refrigerador = s0' \cdot s1$
- $D = s0 \cdot s1$

Lógica Combinacional



Controlador em VHDL: Saídas

- Refrigerador

```
1  entity Refrigerador is
2  port(
3      s0_Refrigerador: in bit;
4      s1_Refrigerador: in bit;
5      Refrigerador: out bit
6  );
7  end;
8
9  architecture behav of Refrigerador is
10 begin
11     Refrigerador <= not(s0_Refrigerador) and s1_Refrigerador;
12 end;
```

- LED

```
1  entity LED is
2  port(
3      s0_LED: in bit;
4      LED: out bit
5  );
6  end;
7
8  architecture behav of LED is
9  begin
10     LED <= s0_LED;
11 end;
```

Controlador em VHDL: Saídas

- D

```
1  entity D is
2  port(
3      s0_D, s1_D: in bit;
4      saida_D: out bit
5  );
6  end;
7
8  architecture behav of D is
9  begin
10     saida_D <= s0_D and s1_D;
11 end;
```

Controlador em VHDL: Saída -> Datapath

- Clear do registrador de temperatura

```
1 entity TempReg_clr is
2   port(
3       s0_TempReg_clr: in bit;
4       s1_TempReg_clr: in bit;
5       TempReg_clr: out bit
6   );
7 end;
8
9 architecture behav of TempReg_clr is
10 begin
11     tempReg_clr <= not(s0_TempReg_clr) and not(s1_TempReg_clr);
12 end;
```

- Load do registrador de temperatura

```
1 entity TempReg_ld is
2   port(
3       s0_TempReg_ld: in bit;
4       s1_TempReg_ld: in bit;
5       TempReg_ld: out bit
6   );
7 end;
8
9 architecture behav of TempReg_ld is
10 begin
11     tempReg_ld <= not(s0_TempReg_ld) and s1_TempReg_ld;
12 end;
```

Controlador em VHDL: Saída -> Datapath

- Início da contagem do temporizador

```
1  entity start_count is
2  port(
3      s0_start_count: in bit;
4      s1_start_count: in bit;
5      start_count: out bit
6  );
7  end;
8
9  architecture behav of start_count is
10 begin
11     start_count <= s0_start_count and not(s1_start_count);
12 end;
```

Controlador em VHDL: Registrador

- Flip-flops

```
1  entity flipflop is
2  port(
3      d_ff: in bit;
4      clk_ff: in bit;
5      q_ff: out bit
6  );
7  end;
8
9  architecture behav of flipflop is
10 begin
11     process(d_ff, clk_ff)
12     begin
13         if(clk_ff ' event and clk_ff = '1' and d_ff = '0') then
14             q_ff <= '0';
15         elsif(clk_ff ' event and clk_ff = '1' and d_ff = '1') then
16             q_ff <= '1';
17         end if;
18     end process;
19 end;
```


Controlador em VHDL: Registrador

- Registrador

```
1  entity registrador is
2  port(
3      clk_reg: in bit;
4      in_n0, in_n1: in bit;
5      out_s0, out_s1: out bit
6  );
7  end;
8
9  architecture behav of registrador is
10 component flipflop is
11 port(
12     d_ff: in bit;
13     clk_ff: in bit;
14     q_ff: out bit
15 );
16 end component;
17
18 begin
19     u1: flipflop port map(d_ff => in_n0, clk_ff => clk_reg, q_ff => out_s0);
20     u2: flipflop port map(d_ff => in_n1, clk_ff => clk_reg, q_ff => out_s1);
21 end;
```

Controlador em VHDL

• Parte 1

```
1 entity Controlador is
2   port(
3     t_lt_negative6: in bit;
4     end_count: in bit;
5     B: in bit;
6     clk: in bit;
7
8     TempReg_clr_c: out bit;
9     TempReg_ld_c: out bit;
10    start_count_c: out bit;
11    LED_c: out bit;
12    Refrigerador_c: out bit
13  );
14 end;
15
16 architecture behav of Controlador is
17   signal s0_c: bit;
18   signal s1_c: bit;
19
20   signal n0_c: bit;
21   signal n1_c: bit;
22
23
24   component registrador is
25     port(
26       clk_reg: in bit;
27       in_n0, in_n1: in bit;
28       out_s0, out_s1: out bit
29     );
30   end component;
```

• Parte 2

```
31
32 component n0 is
33   port(
34     t_lt_negative6_n0: in bit;
35     end_count_n0: in bit;
36     B_n0: in bit;
37     s0_n0: in bit;
38     s1_n0: in bit;
39     saida_n0: out bit
40   );
41 end component;
42
43 component n1 is
44   port(
45     t_lt_negative6_n1: in bit;
46     end_count_n1: in bit;
47     B_n1: in bit;
48     s0_n1: in bit;
49     s1_n1: in bit;
50     saida_n1: out bit
51   );
52 end component;
53
54 component LED is
55   port(
56     s0_LED: in bit;
57     saida_LED: out bit
58   );
59 end component;
```

• Parte 3

```
61
62 component Refrigerador is
63   port(
64     s0_Refrigerador: in bit;
65     s1_Refrigerador: in bit;
66     saida_Refrigerador: out bit
67   );
68 end component;
69
70 component start_count is
71   port(
72     s0_start_count: in bit;
73     s1_start_count: in bit;
74     saida_start_count: out bit
75   );
76 end component;
77
78 component TempReg_clr is
79   port(
80     s0_TempReg_clr: in bit;
81     s1_TempReg_clr: in bit;
82     saida_TempReg_clr: out bit
83   );
84 end component;
85
86 component TempReg_ld is
87   port(
88     s0_TempReg_ld: in bit;
89     s1_TempReg_ld: in bit;
90     saida_TempReg_ld: out bit
91   );
92 end component;
93
94 component D is
95   port(
96     s0_D, s1_D: in bit;
97     saida_D: out bit
98   );
99 end component;
```

Controlador em VHDL

• Parte 4

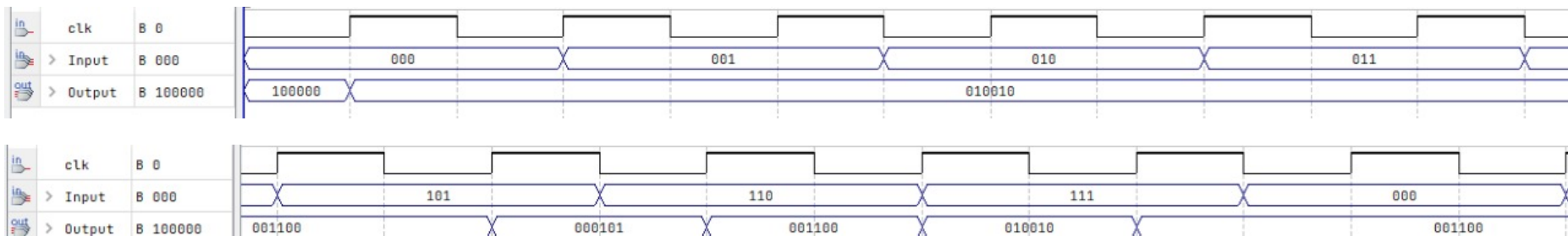
```
93 begin
94
95 u1: n0 port map(
96     t_lt_negative6_n0 => t_lt_negative6,
97     end_count_n0 => end_count,
98     B_n0 => B,
99     s0_n0 => s0_c,
100     s1_n0 => s1_c,
101     saida_n0 => n0_c
102 );
103 u2: n1 port map(
104     t_lt_negative6_n1 => t_lt_negative6,
105     end_count_n1 => end_count,
106     B_n1 => B,
107     s0_n1 => s0_c,
108     s1_n1 => s1_c,
109     saida_n1 => n1_c
110 );
111
112 u3: registrador port map(
113     clk_reg => clk,
114     in_n0 => n0_c,
115     in_n1 => n1_c,
116     out_s0 => s0_c,
117     out_s1 => s1_c
118 );
```

• Parte 5

```
120 u4: LED port map(
121     s0_LED => s0_c,
122     saida_LED => LED_c
123 );
124 u5: Refrigerador port map(
125     s0_Refrigerador => s0_c,
126     s1_Refrigerador => s1_c,
127     saida_Refrigerador => Refrigerador_c
128 );
129 u6: start_count port map(
130     s0_start_count => s0_c,
131     s1_start_count => s1_c,
132     saida_start_count => start_count_c
133 );
134 u7: TempReg_clr port map(
135     s0_TempReg_clr => s0_c,
136     s1_TempReg_clr => s1_c,
137     saida_TempReg_clr => TempReg_clr_c
138 );
139 u8: TempReg_ld port map(
140     s0_TempReg_ld => s0_c,
141     s1_TempReg_ld => s1_c,
142     saida_TempReg_ld => TempReg_ld_c
143 );
144
145 u9: D port map(
146     s0_D => s0_c,
147     s1_D => s1_c,
148     saida_D => D_c
149 );
150
151
152
153
154
155
156
157 end;
```

Controlador em VHDL: Simulação

- Simulação do código apresentado anteriormente:



	Inputs					Outputs							
	T_lt_negative6	end_count	B	s0	s1	n0	n1	TempReg_clr	TempReg_ld	start_count	LED	Refrigerador	D
INIT	x	x	x	0	0	0	1	1	0	0	0	0	0
WAIT_T	0	x	x	0	1	0	1	0	1	0	0	1	0
	1	x	x	0	1	1	0	0	1	0	0	1	0
WAIT_B	x	0	0	1	0	1	0	0	0	1	1	0	0
	x	x	1	1	0	1	1	0	0	1	1	0	0
	x	1	0	1	0	0	1	0	0	1	1	0	0
DISP	x	x	0	1	1	1	0	0	0	0	1	0	1
	x	x	1	1	1	1	1	0	0	0	1	0	1

**OBIGADO PELA
ATENÇÃO!**