



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO  
DCA0212.0 – CIRCUITOS DIGITAIS – TEORIA

## **PROJETO MÁQUINA DE SORVETE**

DISCENTES:  
EFRAIN MARCELO PULGAR PANTALEON  
FERNANDO LUCAS SOUSA SILVA  
MATHEUS GOMES DINIZ ANDRADE  
TEÓFILO VITOR DE CARVALHO CLEMENTE

DOCENTES:  
TIAGO BARROS  
SAMUEL XAVIER

NATAL/RN  
2022

## 1. Introdução

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. Como nosso projeto final da disciplina de Circuitos Digitais DCA0212.0, iremos projetar uma máquina de sorvete expresso descrevendo seus funcionamentos e a lógica de programação dos circuitos presentes em nosso sistema.

## 2. Processo de Design RTL

O processo de desenvolvimento do design de projeto RTL, consiste em 2 etapas, mas a segunda etapa é subdividida em 3 etapas, todas as etapas desenvolvidas estão mostradas abaixo para o projeto da máquina de sorvete.

### 2.1. Máquina de Estados de Alto Nível (HLASM)

Essa representação, apresentada na figura 1, foi desenvolvida como a primeira etapa do projeto e consiste na etapa de captura do comportamento do bloco de controle, a partir dela é possível observar os instantes no qual cada processo ocorre e as condições necessárias para a passagem de uma etapa para outra, e com isso podemos entender seu funcionamento para aplicação da lógica, a seguir descreveremos como seria a passagem pelos estados.

Na nossa máquina temos os estados necessários para a obtenção do sorvete, primeiramente o estado inicial (init), após iniciar a máquina estará no wait\_t, neste estado ela estará com refrigerador ligado, visto que, para que o sorvete obtenha a cremosidade necessária a temperatura tem que ser de  $-6^{\circ}\text{C}$ , tendo esse quesito atendido ele irá para o wait\_b, neste estado teremos o led ligado, sinalizando que o sorvete está pronto para ser retirado, e ao pressionar o botão B iremos para o próximo estado, o disp, neste o consumidor deverá permanecer pressionando o botão até atingir a quantidade por ele desejada, após ele soltar voltaremos para o estado wait\_b e nele também temos o count que em um período definido de tempo irá consultar a temperatura e acaso ela esteja acima de  $-6^{\circ}\text{C}$  ele irá retornar para o wait\_t até que a temperatura esteja novamente estabilizada e possamos retomar para o estado em que o sorvete está apropriado para consumo.

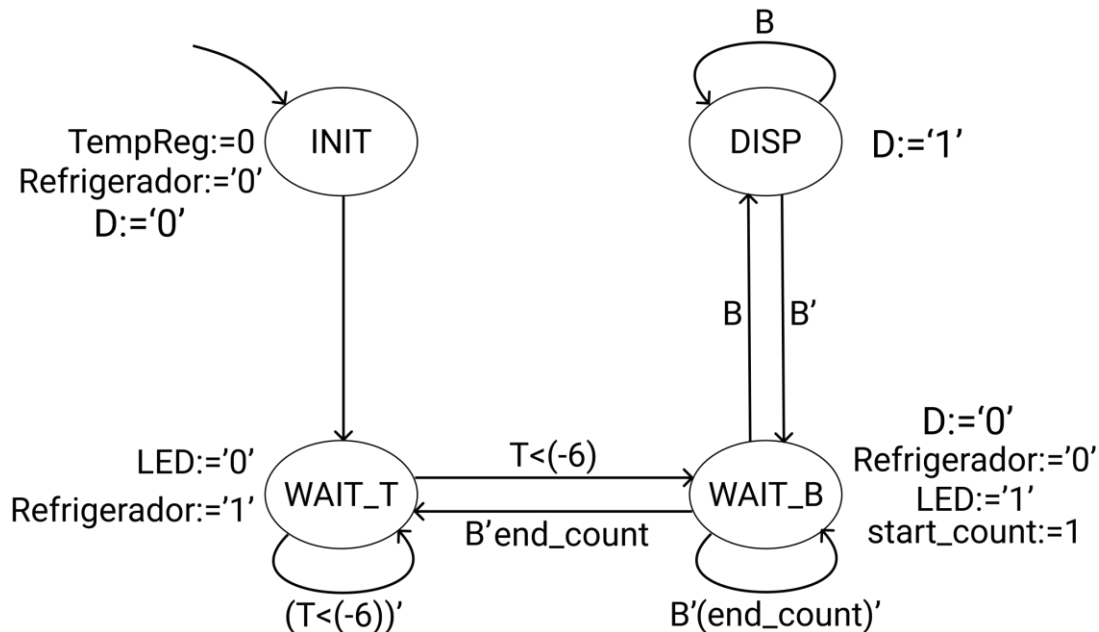


Figura 1 - Máquina de estados de alto nível (HLASM).

## 2.2. Datapath

O Datapath é uma arquitetura constituída de todos os componentes responsáveis pela execução das operações elementares sobre os dados (transformações nos dados), ela é configurada a partir da análise da HLSM, sabendo os processos que terão que ser realizados alocaremos os componentes digitais necessários, sejam eles registradores, comparadores, somadores, mux e outros, de modo que o conjunto deles e suas ligações possa formar o caminho de dados para a execução do processo pedido.

No nosso caso, utilizamos de registradores para o registro das temperaturas, comparadores para comparar se a temperatura atual é adequada baseada na temperatura programada, somadores, mux e portas lógicas, alguns desses estão alocados no temporizador, que foi criado de modo a realizar a contagem do tempo necessário para realizar a checagem. Com isso, definimos o tempo de 300 segundos (5 minutos), para a realização da checagem de temperatura, caso a temperatura esteja menor ou igual a  $-6^{\circ}\text{C}$ , será enviado um sinal ao controlador, ligando o LED indicador de que o sorvete está pronto para ser consumido.

Para o comparador, foi necessário analisar como iríamos fazer a comparação com a temperatura negativa, sendo assim, solucionamos esse problema colocando um bit específico dos bits que representam a temperatura, para indicar se é um valor positivo ou negativo. Assim, nosso comparador irá responder com sinal alto, indicando a temperatura ideal quando o valor em bits for maior ou igual a 6 e tiver o bit de positivo ou negativo igual a 1, demonstrando que é um número negativo.

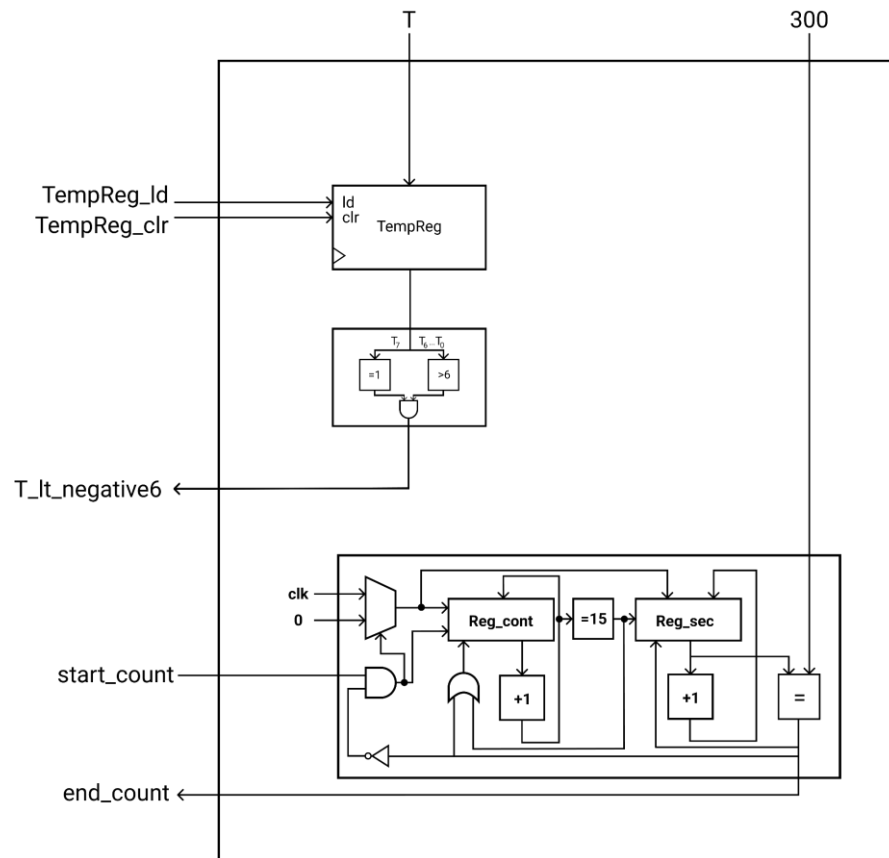


Figura 2 - Datapath com temporizador para a máquina de sorvete.

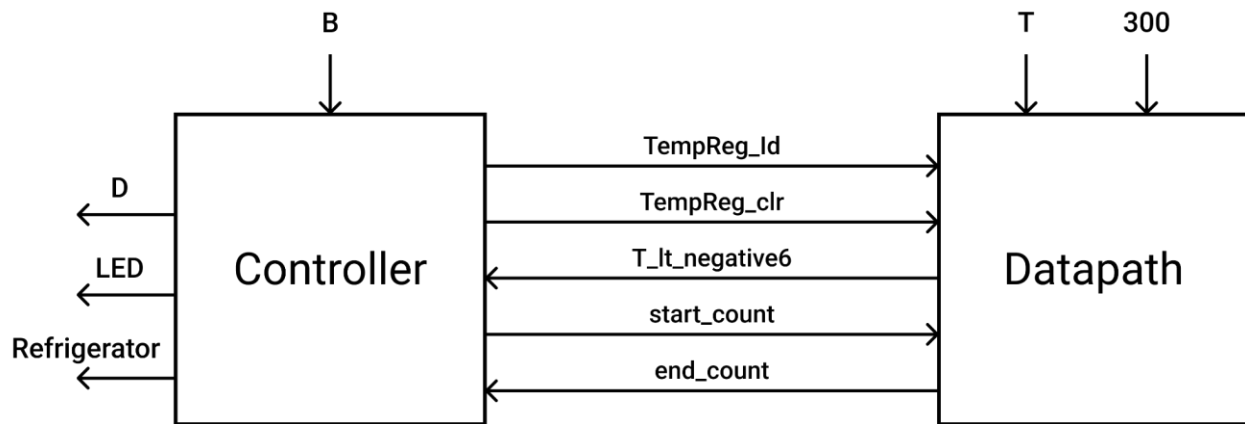


Figura 3 - Conexões internas e externas entre o Bloco de controle e Datapath.

### 2.3. Projeto da Máquina em Baixo Nível (FSM)

A máquina de estados finitos foi feita a partir da conversão do HLSM, aqui substituímos as operações que envolvem dados por sinais de controle do tipo booleano. Assim, após a obtenção da FSM é possível criar a lógica combinacional que de acordo com equações da obtidas da tabela verdade que veremos posteriormente, e com isso o bloco de controle que irá controlar o funcionamento da máquina de sorvete.

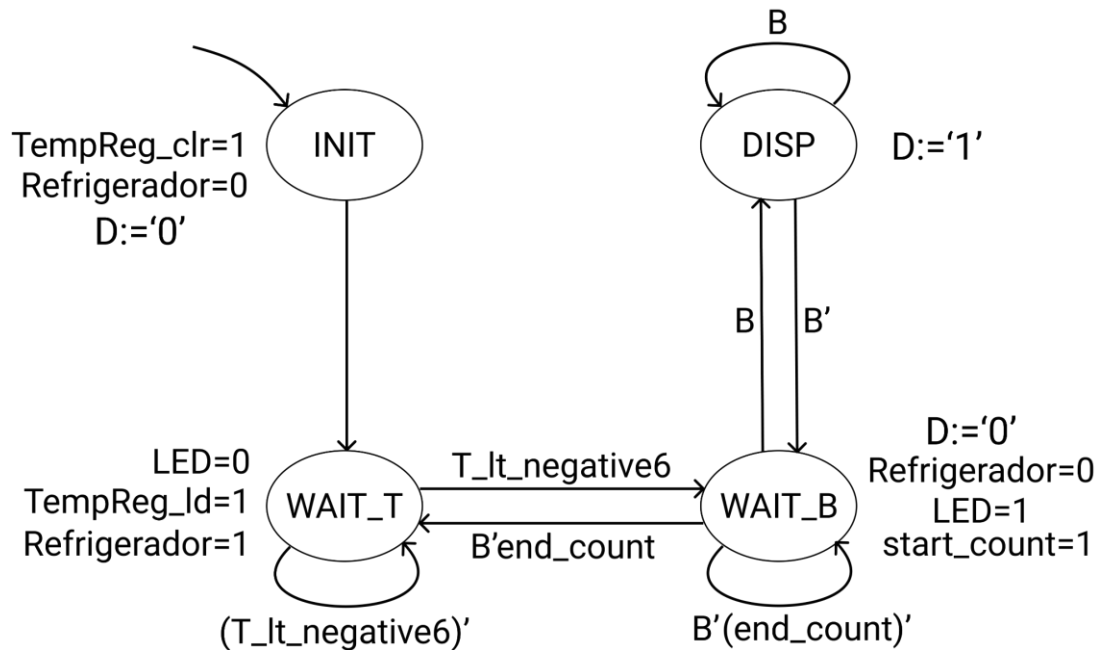


Figura 4 - Máquina de estados (FSM).

### 2.4. Tabela Verdade

Baseado nas informações anteriores, foi construída uma tabela verdade para a máquina de estados finitos de alto nível, feito isso obtemos uma tabela de 32 ( $2^5$ ) linhas, porém foi notado os casos que não dependiam de todas as variáveis, assim foi realizada a simplificação e obtida uma tabela verdade de melhor compreensão. Com isso, analisando os sinais de entrada e saída, percebe-se que para alguns casos os sinais:  $T\_lt\_negative6$ ,  $end\_count$  e  $B$ , não fazem diferença para o valor resultante das saídas:  $n0$  e  $n1$ , já para outros casos um único deles ou um par é responsável pela mudança de  $n0$  e  $n1$ . Desse modo, a tabela resultante é mostrada na figura 5.

	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

Figura 5 – Tabela verdade dos estados simplificada.

## 2.5. Equações Lógicas

Mediante a criação da tabela verdade podemos extrair as equações lógicas de cada estado. Para isso, são utilizados os minitermos que permitem retirar informações da tabela verdade e representar em forma de equações.

$$n0 = (t\_et\_n6 \cdot s0' \cdot s1) + (end\_count' \cdot B' \cdot s0 \cdot s1') + (B \cdot s0 \cdot s1') + (B' \cdot s0 \cdot s1) + (B \cdot s0 \cdot s1)$$

$$n1 = (s0' \cdot s1') + (t\_lt\_n6' \cdot s0' \cdot s1') + (B \cdot s0 \cdot s1') + (end\_count \cdot B' \cdot s0 \cdot s1') + (B \cdot s0 \cdot s1)$$

$$TempReg\_clr = s0' \cdot s1$$

$$TempReg\_ld = (t\_lt\_n6' \cdot s0' \cdot s1) + (t\_lt\_n6 \cdot s0' \cdot s1)$$

$$start\_count = (end\_count' \cdot B' \cdot s0 \cdot s1') + (B \cdot s0 \cdot s1') + (end\_count \cdot B' \cdot s0 \cdot s1')$$

$$LED = (end\_count' \cdot B' \cdot s0 \cdot s1') + (B \cdot s0 \cdot s1') + (end\_count \cdot B' \cdot s0 \cdot s1') + (B' \cdot s0 \cdot s1) + (B \cdot s0 \cdot s1)$$

$$Refrigerador = (t\_lt\_n6' \cdot s0' \cdot s1) + (t\_lt\_n6 \cdot s0' \cdot s1)$$

$$D = s0 \cdot s1$$

Após isso, foi constatado que as equações eram passíveis de simplificações e assim foi feito e com isso obtemos o seguinte resultado:

$$n0 = s1 \cdot s0' \cdot t\_et\_n6 + s0 \cdot (end\_count' + B + s1)$$

$$n1 = s0' \cdot (s1' + t\_lt\_n6') + s0 \cdot (B + s1'end\_count)$$

$$TempReg\_clr = s0' \cdot s1$$

$$TempReg\_ld = s0' \cdot s1'$$

$$start\_count = s0 \cdot s1'$$

$$LED = s0$$

$$Refrigerador = s0' \cdot s1$$

$$D = s0 \cdot s1$$

A estrutura da lógica combinacional e o projeto do controlador em VHDL, com os respectivos códigos serão apresentados no relatório da matéria de laboratório, visto que, ficaria muito extenso apresentar neste.

## 3. Dificuldades e Soluções

Dentre as dificuldades e dúvidas encontradas durante o desenvolvimento do projeto, a mais notória foi a implementação do temporizador, visto que, necessitou de muita pesquisa para a construção de um modelo ideal. Aliado a isso, a implementação do resto dos componentes que comporiam o Datapath baseado nos caminhos fornecidos pela interpretação da HLSM, esta última foi modificada várias vezes, pois no decorrer da implementação notou-se casos em que se notava algum mau comportamento que posteriormente foi consertado.

Posterior a isso ao construir a tabela verdade obtemos uma tabela com linhas em excesso como no projeto anterior, com isso realizamos as simplificações necessárias para a obtenção de sua versão simplificada e assim obtermos as equações que guiaram a construção da lógica combinacional e posteriormente a implementação em VHDL do controlador na parte referente ao laboratório da matéria.

## 4. Conclusões

A realização do projeto da máquina de sorvete permitiu a aplicação da teoria vista em sala de aula em um modelo real e utilizado no cenário comercial em larga escala, para tal foi de grande importância a compreensão dos elementos que compõem a estrutura da máquina e como seria implementada a HLSM no seu contexto. Além disso, também utilizamos da

simplificação na tabela verdade e manipulação de equações lógicas através de minitermos para reduzi-las a um tamanho menor. Outrossim, o uso do temporizador no circuito se mostrou de bastante relevância, visto que, sua integração com o resto do bloco de controle e seus elementos permite a obtenção de um produto final (sorvete) nas condições ideais, assim podemos concluir que o projeto consegue controlar de forma satisfatória a máquina de sorvete.

## **5. Referências bibliográficas**

Vahid, Frank. Digital Design with RTL Design, VHDL, and Verilog Solution Manual. 2º Edição.2010.