



Departamento de Engenharia Elétrica  
Universidade de Brasília  
Disciplina: Laboratório de Sistemas Digitais (Turma: 04)  
Professor: Guilherme Torres

Primeiro Semestre de 2024

## **Experimento 7**

**Datas da realização do experimento:** 29 de maio de 2024

**Aluno:** Felipe Lopes Gibin Duarte

**Matrícula:** 231025207

---

### **1. INTRODUÇÃO**

Neste experimento, vamos expandir sobre os conceitos aprendidos no experimento anterior ao implementar uma máquina de estados síncrona do tipo Moore, tudo isso usando a linguagem de descrição de hardware VHDL. Máquinas de estado usam flip flops para armazenar a informação de qual é o estado atual, além disso, ser do tipo Moore implica que a saída da máquina depende apenas do estado atual. Ademais, a estrutura process da linguagem de descrição será usada novamente neste experimento. Desse modo, todos os procedimentos serão desenvolvidos usando o software “ModelSim”. Para alcançar tais objetivos, o seguinte foi feito.

---

### **2. DESENVOLVIMENTO**

#### **QUESTÃO 01**

Implementar em VHDL e simular no ModelSim uma máquina de estado síncrona do tipo Moore para controlar uma máquina de refrigerantes que aceita moedas de R\$ 0,25 e R\$ 0,50. A cada transição do clock, a máquina deve contar o dinheiro inserido e liberar o refrigerante (e o troco) assim que a soma totalizar ou exceder R\$ 1,00. A máquina deve aceitar qualquer combinação de moedas de R\$ 0,25 e R\$ 0,50, independentemente da ordem em que as moedas foram inseridas. A qualquer momento (desde que a contagem ainda não tenha alcançado R\$ 1,00) o usuário poderá cancelar a compra e a máquina deve, também na transição do clock, devolver a quantia inserida. Considere que a máquina só dispõe de um sabor de refrigerante

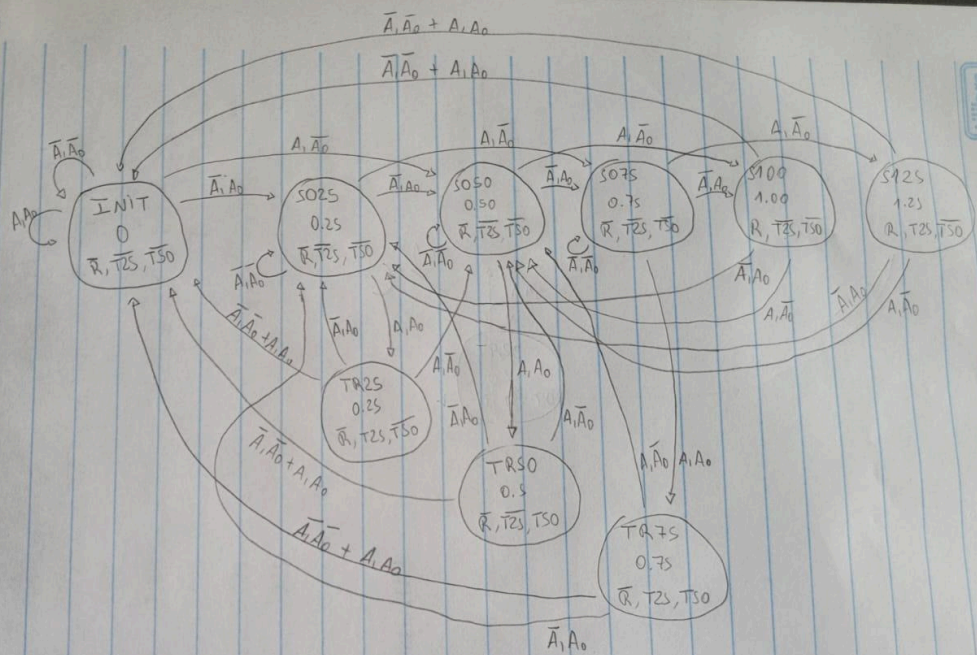
(ou que a escolha do refrigerante é feita antes da máquina de estados iniciar). Logo, o refrigerante é liberado automaticamente (mas na transição do clock) após a inserção do valor de R\$ 1,00 com ou sem troco, não sendo necessário pressionar nenhum botão após a inserção do montante para receber o refrigerante. Isto impede a possibilidade, por exemplo, da inserção do valor de R\$ 1,50. A entidade VHDL deverá ter como entrada um vetor A de 2 bits que indicará se foi inserida uma moeda de R\$ 0,25 (se  $A = 01$ ), se foi inserida uma moeda de R\$ 0,50 (se  $A = 10$ ), se foi solicitada o cancelamento da compra (se  $A = 11$ ) ou se não houve nenhuma ação por parte do usuário (se  $A = 00$ ). Deverá ter como entrada também um clock de 1 bit que fará com que a máquina leia as entradas e mude (ou não) o estado e as saídas. A leitura das entradas, assim como a mudança de estado e das saídas, deverá acontecer exclusivamente na borda de subida do clock. A entidade VHDL deverá ter três saídas de 1 bit cada que indicarão, respectivamente, se a máquina liberou (ou não) um refrigerante, devolveu (ou não) uma moeda de R\$ 0,25 e (ou) devolveu (ou não) uma moeda de R\$ 0,50. Na estrutura process que descreve a lógica dos registradores (memória da máquina de estados), inclua uma condição de “reset” (associada a uma entrada da entidade, ligada a um botão ou a uma chave do kit de desenvolvimento) que leve, de forma assíncrona (isto é, independentemente do sinal de clock) a máquina de estados de volta ao estado inicial. Isto facilitará o processo de teste da máquina implementada. Um exemplo de implementação de condição de “reset” é mostrado no documento tutorial fornecido em conjunto a este roteiro.

## **-Detalhes do projeto**

# Diagrama de Estados

A: 00 - Hold  
01 - 0.25  
Entradas: 10 - 0.50  
11 - comula

R - Refri  
Socilas: T25 - T100.25  
T50 - T100.50



## Tabela de transição de Estados

Estado	A <sub>1</sub> A <sub>0</sub>				R, T25, T50
	00	01	10	11	
0 INIT	INIT	S025	S050	INIT	000
1 S025	S025	S050	S075	TR025	000
2 TR025	INIT	S025	S050	INIT	010
3 S050	S050	S075	S100	TR050	000
4 TR050	INIT	S025	S050	INIT	001
5 S075	S075	S100	S125	TR075	000
6 TR075	INIT	S025	S050	INIT	011
7 S100	INIT	S025	S050	INIT	100
8 S125	INIT	S025	S050	INIT	110

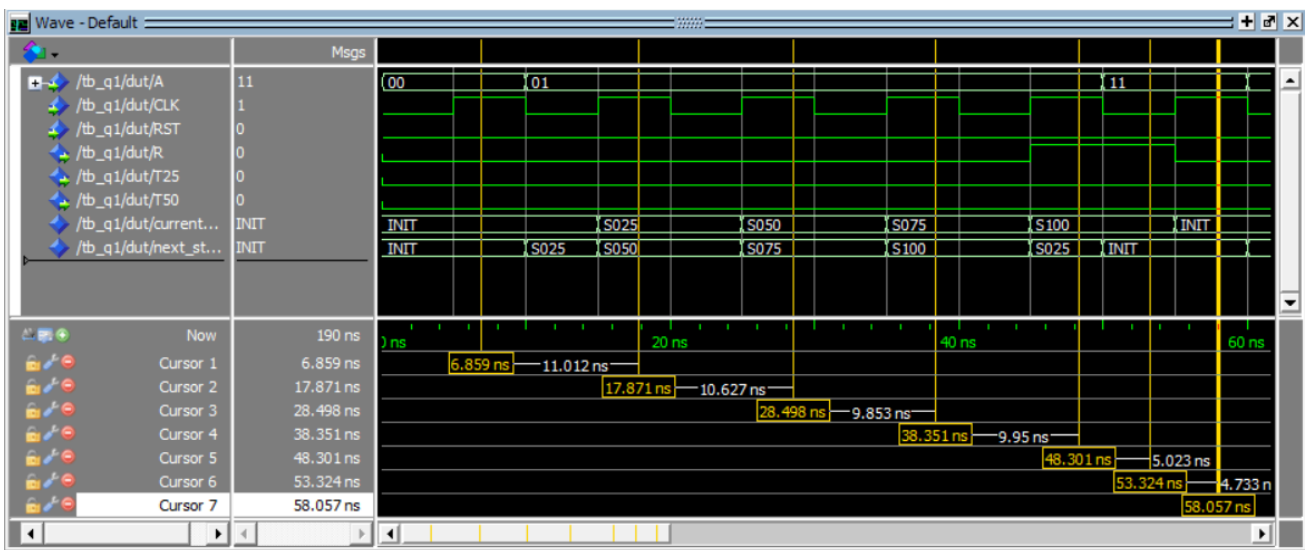
-O código documentado está no segundo arquivo

-Confirmação do código compilado:

Q1(7).vhd	✓	VHDL	0	05/30/2024 11:39:35 ...
tb_Q1(7).vhd	✓	VHDL	1	05/30/2024 11:42:59 ...

```
# Compile of Q1(7).vhd was successful.
# Compile of tb_Q1(7).vhd was successful with warnings.
# 2 compiles, 0 failed with no errors.
```

### -Simulação do código no ModelSim:



-Note que entre 2 cursores pode haver houve uma mudança de A, contudo o estado só muda na borda de subida do clock.

### -Análise:

Cursor 1) Situação inicial, estado INIT

Cursor 2) A = 01, RST = 0, R=0, T25 = 0, T50 = 0;

current\_state = S025, next\_state = S050

Cursor 3) A = 01, RST = 0, R=0, T25 = 0, T50 = 0;

current\_state = S050, next\_state = S075

Cursor 4) A = 01, RST = 0, R=0, T25 = 0, T50 = 0;

current\_state = S075, next\_state = S100

Cursor 5) A = 01, RST = 0, R=1, T25 = 0, T50 = 0;

current\_state = S100, next\_state = S025

Cursor 6) A = 11, RST = 0, R=1, T25 = 0, T50 = 0;

current\_state = S100, next\_state = INIT

Cursor 7) A = 11, RST = 0, R=1, T25 = 0, T50 = 0;

current\_state = INIT, next\_state = INIT



Cursor 8) A = 01, RST = 0, R=0, T25 = 0, T50 = 0;

current\_state = INIT, next\_state = S025

Cursor 9) A = 10, RST = 0, R=0, T25 = 0, T50 = 0;

current\_state = S025, next\_state = S075

Cursor 10) A = 10, RST = 0, R=0, T25 = 0, T50 = 0;

current\_state = S075, next\_state = S125

Cursor 11) A = 11, RST = 0, R=1, T25 = 1, T50 = 0;

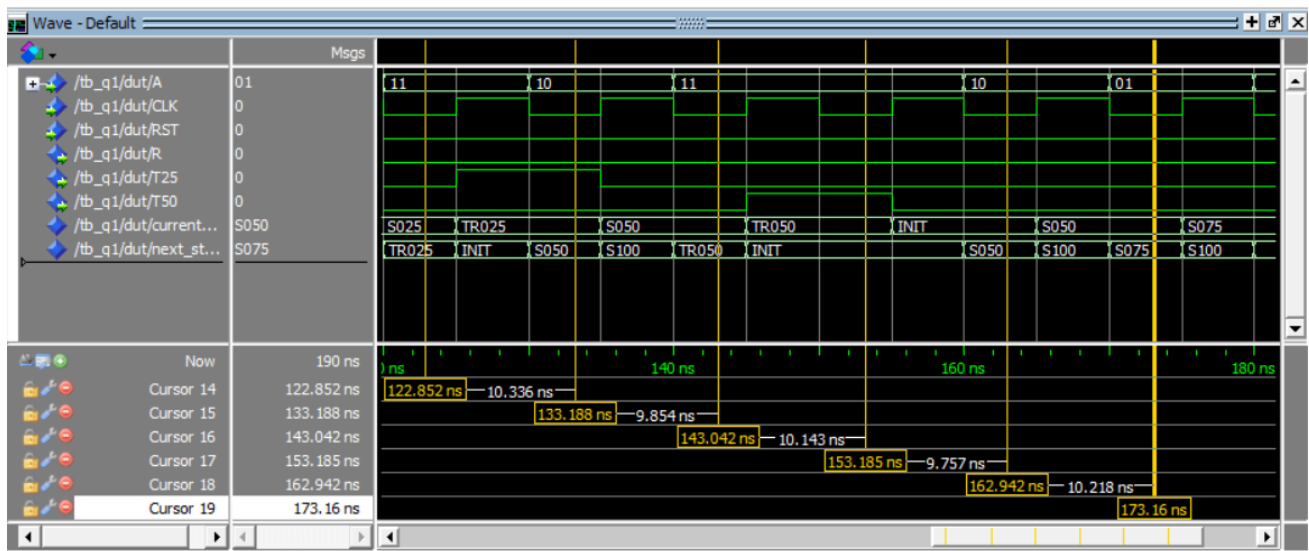
current\_state = S125, next\_state = INIT

Cursor 12) A = 11, RST = 0, R=0, T25 = 0, T50 = 0;

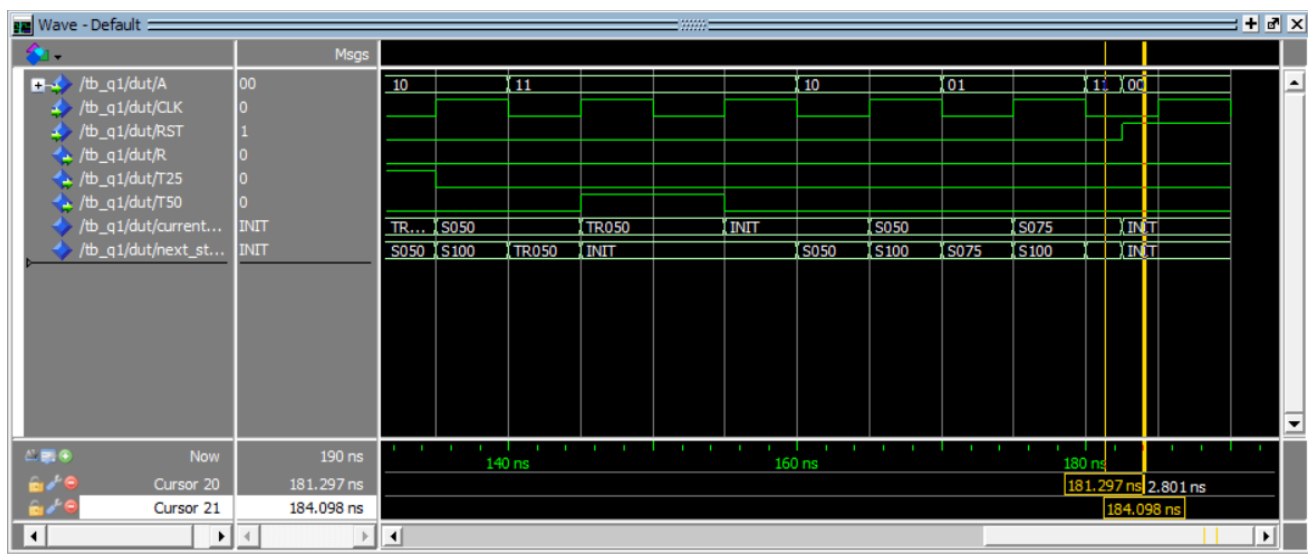
current\_state = INIT, next\_state = INIT

Cursor 13) A = 01, RST = 0, R=0, T25 = 0, T50 = 0;

current\_state = INIT, next\_state = S025



- Cursor 14) A = 11, RST = 0, R=0, T25 = 0, T50 = 0;  
current\_state = S025, next\_state = TR025
- Cursor 15) A = 10, RST = 0, R=0, T25 = 1, T50 = 0;  
current\_state = TR025, next\_state = S050
- Cursor 16) A = 11, RST = 0, R=0, T25 = 0, T50 = 0;  
current\_state = S050, next\_state = TR050
- Cursor 17) A = 11, RST = 0, R=0, T25 = 1, T50 = 1;  
current\_state = TR050, next\_state = INIT
- Cursor 18) A = 10, RST = 0, R=0, T25 = 0, T50 = 0;  
current\_state = INIT, next\_state = S050
- Cursor 19) A = 01, RST = 0, R=0, T25 = 0, T50 = 0;  
current\_state = S050, next\_state = S075



-Note que no cursor 20, o reset (assíncrono) está ativo, logo a máquina não espera o clock para mudar de estado

Cursor 20) A = 11, RST = 0, R=0, T25 = 0, T50 = 0;  
current\_state = S075, next\_state = TR075  
Cursor 21) A = 00, RST = 1, R=0, T25 = 1, T50 = 0;  
current\_state = INIT, next\_state = INIT

---

### 3. CONCLUSÃO

Após todos os procedimentos, houve uma compreensão mais sólida acerca dos conceitos básicos de design digital usando VHDL. Nesse experimento em particular, construímos uma máquina de estados do tipo Moore engatilhada na borda de subida do clock. Foi necessário realizar processos tanto síncronos como combinacionais, o que condiz com as diferentes seções de uma máquina de estado. O maior obstáculo encontrado no experimento foi em como mostrar o estado atual e o próximo estado no testbench, o que foi resolvido ao adicionar a forma de onda à instância do dut. Por fim, a experiência foi proveitosa ao continuar a exposição a conceitos do VHDL como o process e ao proporcionar um primeiro contato com máquinas de estado do tipo Moore.