A logo with blue and green squares

Description automatically generatedUniversidade de Brasília – UNB

Curso: Engenharia de Redes de Comunicação  
Disciplina: Laboratório de Sistemas Digitais   
Turma: 08

Relatório da Disciplina Laboratório de Sistemas Digitais  
Tema: Experimento 07 – Máquinas de Estado de Moore

Aluno: Pedro Henrique Dias Avelar  
Matrícula: 241037112  
Professor: Eduardo Paiva

[Figura 1:Diagrama de Transição de Estados para a máquina de refrigerante proposta – Imagem do autor feita com a ferramenta online draw.io 4](#_Toc189321817)

[Figura 2:Resultado da simulação para a máquina de refrigerante 10](#_Toc189321818)

[Figura 3:Resultado do primeiro cenário de teste 10](#_Toc189321819)

[Figura 4:Resultado do segundo cenário de teste 11](#_Toc189321820)

[Figura 5:Resultado do terceiro cenário de teste 11](#_Toc189321821)

[Figura 6:Resultado do quarto cenário de teste 12](#_Toc189321822)

[Figura 7:Resultado do quinto cenário de teste 12](#_Toc189321823)

[Figura 8:Resultado do sexto cenário de teste 12](#_Toc189321824)

[Figura 9:Resultado do sétimo cenário de teste 13](#_Toc189321825)

[Figura 10:Resultado do oitavo cenário de teste 13](#_Toc189321826)

[Figura 11:Resultado do nono cenário de teste 14](#_Toc189321827)

[Figura 12:Resultado do décimo cenário de teste 14](#_Toc189321828)

[Figura 13:Resultado do décimo primeiro cenário de teste 15](#_Toc189321829)

[Figura 14:Resultado do décimo segundo cenário de teste 15](#_Toc189321830)

[Figura 15:Resultado do décimo terceiro cenário de teste 15](#_Toc189321831)

[Figura 16:Código do décimo quarto cenário de teste realizado 15](#_Toc189321832)

[Figura 17:Resultado do décimo quarto cenário de teste 16](#_Toc189321833)

[Tabela 1: Tabela de Transição de estados para a máquina de refrigerantes proposta 5](#_Toc189321834)

[Código 1:Modelagem da máquina de refrigerante 7](#_Toc189321835)

[Código 2: Definição do tipo customizado ESTADO 7](#_Toc189321836)

[Código 3:Testbench para a máquina de refrigerante 10](#_Toc189321837)

[Código 4:Código do primeiro cenário de teste realizado 10](#_Toc189321838)

[Código 5:Código do segundo cenário de teste realizado 11](#_Toc189321839)

[Código 6:Código do terceiro cenário de teste realizado 11](#_Toc189321840)

[Código 7:Código do quarto, quinto e sexto cenário de teste realizado 12](#_Toc189321841)

[Código 8:Código do sétimo, oitavo, nono e décimo cenário de teste realizado 13](#_Toc189321842)

[Código 9:Código do décimo primeiro, décimo segundo e décimo terceiro cenário de teste realizado 14](#_Toc189321843)

# Introdução

O presente experimento tem o seguinte objetivo:

• Implementar uma máquina de estados síncrona do tipo Moore em VHDL e a simular no ModelSim

# Atividade

Implemente em VHDL e simule no ModelSim uma máquina de estados síncrona do tipo Moore para controlar uma máquina de vendas que aceita moedas de 25 centavos e 50 centavos. A cada transição do clock, a máquina deve contar o dinheiro inserido e liberar o produto e o troco assim que a soma totalizar ou exceder 1 real. A máquina deve aceitar qualquer combinação de moedas de 25 centavos e 50 centavos, em qualquer ordem. O usuário pode cancelar a compra a qualquer momento, desde que a soma das moedas ainda esteja abaixo de 1 real.

Considere que a máquina vende apenas um produto e que ele é liberado automaticamente quando a soma das moedas inseridas atingir ou exceder 1 real, com ou sem troco. Considere também que o usuário faz, no máximo, uma ação a cada período do clock. Isso impede, por exemplo, a possibilidade de o usuário inserir R$ 1,50.

A entidade VHDL deve ter duas entradas: um clock (de um bit) e um vetor 𝐴 de dois bits. Se 𝐴 = 01, foi inserida uma moeda de 25 centavos, se 𝐴 = 10, foi inserida uma moeda de 50 centavos, se 𝐴 = 11, o usuário solicitou o cancelamento da compra e, se 𝐴 = 00, não houve ação do usuário. As saídas serão três, todas de um bit: uma para indicar se a máquina liberou o produto, outra para indicar se a máquina devolveu uma moeda de 25 centavos e outra para indicar se a máquina devolveu uma moeda de 50 centavos. Após o fim da venda (pela liberação do produto ou cancelamento), a máquina deve voltar ao estado inicial. Uma vez que a soma das moedas atinja ou exceda 1 real ou que o usuário cancele a compra, a máquina só aceitará novas moedas após voltar ao estado inicial.

**ATENÇÃO! Considere os seguintes pontos:**

* Antes de começar a escrever o código, desenhe o diagrama de transição de estados, você deve incluí-lo no relatório.
* Lembre-se que a variável 𝐴 tem quatro valores possíveis, logo, para cada estado, pode haver até quatro transições possíveis.
* Em alguns casos, não serão possíveis todos os valores de 𝐴 . Por exemplo, se a máquina já acumulou 1 real ou mais, não é mais possível inserir moedas nem cancelar a compra, logo, o único valor de entrada possível neste estado é 𝐴=00 (seu código deve simplesmente ignorar outros valores de entrada, eles seriam fisicamente impossíveis em uma implementação real).

Uma máquina de estados de Moore é um tipo de máquina de estados finita onde as saídas são determinadas apenas pelo estado atual da máquina. Em outras palavras, para cada estado, há uma saída específica associada a ele. No caso de uma máquina de Moore síncrona, as transições de estado irão ocorrer no evento de sincronia, sendo este geralmente a borda de subida do clock.

O processo de arquitetura de uma máquina de estados geralmente envolve as seguintes etapas:

* Preparo do diagrama de transição de estados
* Preparo da tabela de transição de estados
* Atribuição de códigos aos estados da tabela de transição
* Construção das equações do próximo estado a partir da tabela de transição de estados
* Construção das equações das saídas a partir das equações de próximo estado
* Montagem do circuito de forma a atender as equações de próximo estado e das saídas

Por meio da linguagem VHDL podemos modelar e simular uma máquina que atenda os requisitos propostos sem necessariamente descrever a arquitetura física do circuito – no caso, pensando no conceito de caixa preta, precisamos que a máquina modelada apenas atenda os requisitos de entrada e saída. Embora dessa forma o modelo não seja possível de ser implantado fisicamente, ainda assim ele terá utilidade como *“golden module”* para a realização de testes.

Mesmo para este modelo ideal, ainda assim se faz necessário preparar o diagrama de transição e a tabela de transição de estados. Estes dois componentes irão definir qual o comportamento esperado para a máquina.

A diagram of a diagram

AI-generated content may be incorrect.

Figura :Diagrama de Transição de Estados para a máquina de refrigerante proposta – Imagem do autor feita com a ferramenta online draw.io

Para a máquina proposta foram definidos 9 estados: INIT; 0,25; 0,50; 0,75; 1,00; 1,25; Devolve 0,75; Devolve 0,50; Devolve 0,25. De acordo com o comportamento definido no enunciado, os estados INIT, 0,25, 0,50 e 0,75 devem aceitar todas as combinações possíveis de entrada. A partir do momento em que a máquina atinge 1 real, situação representada pelos estados 1,00 e 1,25, ou que seja solicitada a devolução, situação representada pelos estados Devolve 0,75, Devolve 0,50 e Devolve 0,25, a máquina deve então impedir a inserção de novas moedas e voltar ao estado inicial INIT, momento em que voltará a aceitar novas moedas.



Tabela : Tabela de Transição de estados para a máquina de refrigerantes proposta

Em uma máquina real, o estado da máquina é registrado com o uso de flip-flops. Para n flip-flops, é possível registrar 2n estados. Como foi necessário criar 9 estados, em tese com 3 flip-flops seria possível representar apenas 23=8 estados, sendo assim um número insuficiente de flip-flops. No entanto, com 4 flip-flops temos então 24=16 estados possíveis; como apenas 9 estados foram designados, temos 7 estados não utilizados.

Há duas abordagens para lidar com esses estados, a abordagem de custo mínimo e a abordagem de risco mínimo. No custo mínimo, designamos *don’t cares* para os estados não utilizados. Esta abordagem, embora mais barata, apresenta o risco de, por alguma situação inesperada, a máquina acabar transitando para um desses estados e, a partir daí, tendo então um comportamento não definido. A outra alternativa, risco mínimo, envolve planejar para os estados não utilizados a transição para um determinado estado, de forma a assegurar o conhecimento do comportamento da máquina independente do estado atual e da próxima transição. Para esta máquina, por exemplo, uma boa alternativa seria regredir dos estados não utilizado para o estado INIT. No entanto, como o experimento está sendo modelado como *“golden module”*, iremos então assumir que a máquina é perfeita e incapaz de transitar para um dos estados indefinidos, e assim então será modelada sob a abordagem de custo mínimo. A máquina foi modelada através do código a seguir:

0 -- Experimento 07

1 -- Aluno: Pedro Henrique Dias Avelar 241037112

2 -- Turma 08

3 -- Data: 25/01/2025

4

5 -- Maquina de Estado Síncrona de Moore - Máquina de Refrigerante

6

7 **LIBRARY** IEEE**;**

8 **USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

9 **USE** WORK**.**type\_estado**.ALL;**

10

11

12 **ENTITY** MAQ\_MOORE\_REFRI **is**

13 **PORT** **(**

14 CLK **:** **IN** STD\_LOGIC**;**

15 A **:** **IN** STD\_LOGIC\_VECTOR **(**1 **DOWNTO** 0**);**

16 L**,** D50**,** D25**:** **OUT** STD\_LOGIC**;**

17 CURRENT\_STATE**:** **OUT** ESTADO

18 **);**

19 **END** MAQ\_MOORE\_REFRI**;**

20

21 **ARCHITECTURE** ARC\_MAQ\_MOORE\_REFRI **OF** MAQ\_MOORE\_REFRI **IS**

22

23 **SIGNAL** ESTADO\_ATUAL**,** PROXIMO\_ESTADO**:** ESTADO**;**

24

25 **BEGIN**

26 CURRENT\_STATE **<=** ESTADO\_ATUAL**;**

27 PROCESSO\_SINCRONO**:** **PROCESS** **(**CLK**)**

28 **BEGIN**

29 **IF** **RISING\_EDGE(**CLK**)** **THEN**

30 ESTADO\_ATUAL **<=** PROXIMO\_ESTADO**;**

31 **END** **IF;**

32 **END** **PROCESS** PROCESSO\_SINCRONO**;**

33

34 PROCESSO\_COMBINACIONAL**:** **PROCESS** **(**ESTADO\_ATUAL**,** A**)**

35 **BEGIN**

36 L **<=** '0'**;**

37 D50 **<=** '0'**;**

38 D25 **<=** '0'**;**

39 **CASE** ESTADO\_ATUAL **IS**

40 **WHEN** INIT **=>**

41 **CASE** A **IS**

42 **WHEN** "00" **=>** PROXIMO\_ESTADO **<=** INIT**;**

43 **WHEN** "01" **=>** PROXIMO\_ESTADO **<=** VLR\_25**;**

44 **WHEN** "10" **=>** PROXIMO\_ESTADO **<=** VLR\_50**;**

45 **WHEN** "11" **=>** PROXIMO\_ESTADO **<=** INIT**;**

46 **WHEN** **OTHERS** **=>** PROXIMO\_ESTADO **<=** INIT**;**

47 **END** **CASE;**

48 **WHEN** VLR\_25 **=>**

49 **CASE** A **IS**

50 **WHEN** "00" **=>** PROXIMO\_ESTADO **<=** VLR\_25**;**

51 **WHEN** "01" **=>** PROXIMO\_ESTADO **<=** VLR\_50**;**

52 **WHEN** "10" **=>** PROXIMO\_ESTADO **<=** VLR\_75**;**

53 **WHEN** "11" **=>** PROXIMO\_ESTADO **<=** DEVOLVE\_25**;**

54 -- EM CASO DE ALGUM DEFEITO, A MÁQUINA IRÁ DEVOLVER OS 25 CENTAVOS

55 **WHEN** **OTHERS** **=>** PROXIMO\_ESTADO **<=** DEVOLVE\_25**;**

56 **END** **CASE;**

57 **WHEN** VLR\_50 **=>**

58 **CASE** A **IS**

59 **WHEN** "00" **=>** PROXIMO\_ESTADO **<=** VLR\_50**;**

60 **WHEN** "01" **=>** PROXIMO\_ESTADO **<=** VLR\_75**;**

61 **WHEN** "10" **=>** PROXIMO\_ESTADO **<=** VLR\_100**;**

62 **WHEN** "11" **=>** PROXIMO\_ESTADO **<=** DEVOLVE\_50**;**

63 -- EM CASO DE ALGUM DEFEITO, A MÁQUINA IRÁ DEVOLVER OS 50 CENTAVOS

64 **WHEN** **OTHERS** **=>** PROXIMO\_ESTADO **<=** DEVOLVE\_50**;**

65 **END** **CASE;** --

66 **WHEN** VLR\_75 **=>**

67 **CASE** A **is**

68 **WHEN** "00" **=>** PROXIMO\_ESTADO **<=** VLR\_75**;**

69 **WHEN** "01" **=>** PROXIMO\_ESTADO **<=** VLR\_100**;**

70 **WHEN** "10" **=>** PROXIMO\_ESTADO **<=** VLR\_125**;**

71 **WHEN** "11" **=>** PROXIMO\_ESTADO **<=** DEVOLVE\_75**;**

72 -- EM CASO DE ALGUM DEFEITO, A MÁQUINA IRÁ DEVOLVER OS 75 CENTAVOS

73 **WHEN** **OTHERS** **=>** PROXIMO\_ESTADO **<=** DEVOLVE\_75**;**

74 **END** **CASE;**

75 --PARA TODOS OS DEMAIS CASOS, O PROXIMO ESTADO SEMPRE SERA

76 --INIT. ASSIM QUE A MAQUINA CONSIGA 1 REAL OU SEJA SOLICITADO

77 --A DEVOLUCAO, SÓ SERÁ ACEITO NOVOS DEPÓSITOS APÓS A MÁQUINA VOLTAR

78 --AO ESTADO INICIAL

79 **WHEN** VLR\_100 **=>**

80 PROXIMO\_ESTADO **<=** INIT**;**

81 L **<=** '1'**;** -- LIBERA O REFRI

82 **WHEN** VLR\_125 **=>**

83 PROXIMO\_ESTADO **<=** INIT**;**

84 L **<=** '1'**;** -- LIBERA O REFRI

85 D25 **<=** '1'**;** -- DEVOLVE 25 CENTAVOS

86 **WHEN** DEVOLVE\_75 **=>**

87 PROXIMO\_ESTADO **<=** INIT**;**

88 D50 **<=** '1'**;** -- DEVOLVE 50 CENTAVOS

89 D25 **<=** '1'**;** -- DEVOLVE 25 CENTAVOS

90 **WHEN** DEVOLVE\_50 **=>**

91 PROXIMO\_ESTADO **<=** INIT**;**

92 D50 **<=** '1'**;** -- DEVOLVE 50 CENTAVOS

93 **WHEN** DEVOLVE\_25 **=>**

94 PROXIMO\_ESTADO **<=** INIT**;**

95 D25 **<=** '1'**;** -- DEVOLVE 25 CENTAVOS

96 **END** **CASE;**

97 **END** **PROCESS** PROCESSO\_COMBINACIONAL**;**

98 **END** ARC\_MAQ\_MOORE\_REFRI**;**

Código :Modelagem da máquina de refrigerante

A arquitetura da máquina de refrigerantes é divida em dois processos. No processo síncrono (linha 27) temos a atualização do estado da máquina a cada borda de subida do clock. No processo combinacional, temos a definição do próximo estado da máquina de acordo com a entrada A e o estado atual da máquina, replicando o comportamento definido na tabela 1. Além disso, temos a ativação das saídas nos respectivos estados informados na tabela 1. Na linha 13 do código há a instrução USE WORK.type\_estado.ALL; o tipo estado foi definido no arquivo type\_estado.vhd, apresentado a seguir:

1 **library** IEEE**;**

2 **use** IEEE**.**STD\_LOGIC\_1164**.ALL;**

3

4 **package** type\_estado **is**

5 **TYPE** ESTADO **IS** **(**INIT**,** VLR\_25**,** VLR\_50**,** VLR\_75**,** VLR\_100**,** VLR\_125**,** DEVOLVE\_75**,** DEVOLVE\_50**,** DEVOLVE\_25**);**

6 **end** **package** type\_estado**;**

Código : Definição do tipo customizado ESTADO

O tipo de dado estado é uma simples lista enumerada. Poderíamos utilizar a representação binária definida na tabela 1 para representar os estados, mas o uso da lista enumerada auxilia na escrita e na visualização do código. Além disso, na entidade MAQ\_MOORE\_REFRI definida na linha 12 do código 1, foi adicionado uma saída do tipo estado. Embora não possua significado físico, mais a frente será possível ver que, com essa saída, podemos representar o estado da máquina na simulação do testbench, facilitando muito na visualização dos resultados. Segue o código do testbench:

0 -- Experimento 07 - TESTBENCH

1 -- Aluno: Pedro Henrique Dias Avelar 241037112

2 -- Turma 08

3 -- Data: 25/01/2025

4

5 -- Testbench - Tempo de simulação: ?

6

7 **LIBRARY** IEEE**;**

8 **USE** IEEE**.**STD\_LOGIC\_1164**.ALL;**

9 **USE** IEEE**.**STD\_LOGIC\_UNSIGNED**.ALL;**

10 **USE** IEEE**.**NUMERIC\_STD**.ALL;**

11 **USE** WORK**.**type\_estado**.ALL;**

12

13 **ENTITY** TESTBENCH\_E07\_MAQ\_REFRI **IS**

14 **END** TESTBENCH\_E07\_MAQ\_REFRI**;**

15

16 **ARCHITECTURE** ARC\_TESTBENCH\_E07\_MAQ\_REFRI **OF** TESTBENCH\_E07\_MAQ\_REFRI **IS**

17 **COMPONENT** MAQ\_MOORE\_REFRI **IS**

18 **PORT** **(**

19 CLK **:** **IN** STD\_LOGIC**;**

20 A **:** **IN** STD\_LOGIC\_VECTOR **(**1 **DOWNTO** 0**);**

21 L**,** D50**,** D25**:** **OUT** STD\_LOGIC**;**

22 CURRENT\_STATE**:** **OUT** ESTADO

23 **);**

24 **END** **COMPONENT;**

25

26 **SIGNAL** CLK\_TB**:** STD\_LOGIC **:=** '0'**;**

27 **SIGNAL** A\_TB**:** STD\_LOGIC\_VECTOR**(**1 **DOWNTO** 0**);**

28 **SIGNAL** L\_TB**,** D50\_TB**,** D25\_TB**:** STD\_LOGIC**;**

29 **SIGNAL** CURRENT\_STATE\_TB**:** ESTADO**;**

30

31 **BEGIN**

32 DUT**:** MAQ\_MOORE\_REFRI **PORT** **MAP** **(**CLK\_TB**,** A\_TB**,** L\_TB**,** D50\_TB**,** D25\_TB**,**CURRENT\_STATE\_TB**);**

33 CLK\_TB **<=** **NOT** CLK\_TB **AFTER** 5 NS**;**

34 **PROCESS**

35 **BEGIN**

36 **REPORT** "INICIANDO TESTE..." **SEVERITY** NOTE**;**

37 -- CENARIO 1: INIT -> VLR\_25 -> DEVOLVE\_25 -> INIT

38 --A: 01 11 XX

39 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

40 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

41 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

42 -- CENARIO 2: INIT -> VLR\_25 -> VLR\_50 -> DEVOLVE\_50 -> INIT

43 --A: 01 01 11 XX

44 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

45 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

46 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

47 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

48 -- CENARIO 3: INIT -> VLR\_50 -> DEVOLVE\_50 -> INIT

49 --A: 10 11 XX

50 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

51 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

52 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

53 -- CENARIO 4: INIT -> VLR\_25 -> VLR\_50 -> VLR\_75 -> DEVOLVE\_75 -> INIT

54 --A: 01 01 01 11 XX

55 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

56 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

57 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

58 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

59 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

60 -- CENARIO 5: INIT -> VLR\_25 -> VLR\_75 -> DEVOLVE\_75 -> INIT

61 --A: 01 10 11 XX

62 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

63 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

64 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

65 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

66 -- CENARIO 6: INIT -> VLR\_50 -> VLR\_75 -> DEVOLVE\_75 -> INIT

67 --A: 10 01 11 XX

68 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

69 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

70 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

71 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

72 -- CENARIO 7: INIT -> VLR\_25 -> VLR\_50 -> VLR\_75 -> VLR\_100 -> INIT

73 --A: 01 01 01 01 XX

74 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

75 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

76 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

77 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

78 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

79 -- CENARIO 8: INIT -> VLR\_25 -> VLR\_50 -> VLR\_100 -> INIT

80 --A: 01 01 10 XX

81 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

82 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

83 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

84 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

85 -- CENARIO 9: INIT -> VLR\_25 -> VLR\_75 -> VLR\_100 -> INIT

86 --A: 01 10 01 XX

87 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

88 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

89 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

90 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

91 -- CENARIO 10: INIT -> VLR\_50 -> VLR\_100 -> INIT

92 --A: 10 10 XX

93 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

94 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

95 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

96 -- CENARIO 11: INIT -> VLR\_25 -> VLR\_50 -> VLR\_75 -> VLR\_125 -> INIT

97 --A: 01 01 01 10 XX

98 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

99 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

100 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

101 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

102 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

103 -- CENARIO 12: INIT -> VLR\_25 -> VLR\_75 -> VLR\_125 -> INIT

104 --A: 01 10 10 XX

105 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

106 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

107 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

108 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

109 -- CENARIO 13: INIT -> VLR\_50 -> VLR\_75 -> VLR\_125 -> INIT

110 --A: 10 01 10 XX

111 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

112 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

113 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

114 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

115 -- CENARIO 14: INIT -> INIT -> VLR\_25 -> VLR\_25 -> VLR\_50 ->

116 -- 00 01 00 01 00

117 -- VLR\_50 -> VLR\_75 -> VLR\_75 -> VLR\_100 -> INIT

118 -- 01 00 01 00

119 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

120 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

121 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

122 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

123 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

124 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

125 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

126 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

127 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

128 **REPORT** "TESTE FINALIZADO!" **SEVERITY** NOTE**;**

129 **WAIT;**

130 **END** **PROCESS;**

131 **END** ARC\_TESTBENCH\_E07\_MAQ\_REFRI**;**

Código :Testbench para a máquina de refrigerante

O testbench consistiu na realização de 14 diferentes cenários buscando passar pelas possíveis transições de estado esperadas no funcionamento da máquina. A simulação teve uma duração de 610 ns e apresentou o seguinte resultado:

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado da simulação para a máquina de refrigerante

Como mencionado anteriormente, o uso da variável de saída CURRENT\_STATE, presente na linha 22 do código 3, junto com sua atribuição ao sinal CURRENT\_STATE\_TB na linha 29 do mesmo código nos permite facilmente visualizar em qual estado a máquina se encontra no diagrama de ondas do Modelsim. Visualizando agora cenário a cenário, temos os seguintes resultados:

37 -- CENARIO 1: INIT -> VLR\_25 -> DEVOLVE\_25 -> INIT

38 --A: 01 11 XX

39 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

40 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

41 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

Código :Código do primeiro cenário de teste realizado

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do primeiro cenário de teste

Na primeira borda de subida de clock com a entrada 01 (moeda de 25 centavos) a máquina sai do estado INIT para o estado VLR\_25; em seguida na próxima borda de subida de clock com a entrada 11 (cancelamento da compra) a máquina passa para o estado DEVOLVE\_25 e ativa a saída D25. Por fim, a máquina retorna ao estado INIT.

42 -- CENARIO 2: INIT -> VLR\_25 -> VLR\_50 -> DEVOLVE\_50 -> INIT

43 --A: 01 01 11 XX

44 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

45 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

46 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

47 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

Código :Código do segundo cenário de teste realizado

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do segundo cenário de teste

A máquina passa por duas bordas de subida de clock com a entrada 01 (moeda de 25 centavos), transitando pelos estados VLR\_25 e VLR\_50. Em seguida é solicitada a devolução, o que faz a máquina ir para o estado DEVOLVE\_50, o que simultaneamente ativa a saída D50. Por fim a máquina volta para o estado INIT.

48 -- CENARIO 3: INIT -> VLR\_50 -> DEVOLVE\_50 -> INIT

49 --A: 10 11 XX

50 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

51 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

52 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

Código :Código do terceiro cenário de teste realizado

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do terceiro cenário de teste

Após a inserção de uma moeda de 50 centavos (entrada 10), é solicitada a devolução, fazendo com que a máquina transite pelos estados VLR\_50 e DEVOLVE\_50, antes de voltar a INIT. No estado DEVOLVE\_50 a saída D50 é ativada.

53 -- CENARIO 4: INIT -> VLR\_25 -> VLR\_50 -> VLR\_75 -> DEVOLVE\_75 -> INIT

54 --A: 01 01 01 11 XX

55 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

56 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

57 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

58 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

59 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

60 -- CENARIO 5: INIT -> VLR\_25 -> VLR\_75 -> DEVOLVE\_75 -> INIT

61 --A: 01 10 11 XX

62 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

63 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

64 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

65 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

66 -- CENARIO 6: INIT -> VLR\_50 -> VLR\_75 -> DEVOLVE\_75 -> INIT

67 --A: 10 01 11 XX

68 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

69 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

70 A\_TB **<=** "11"**;** **WAIT** **FOR** 10 NS**;**

71 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

Código :Código do quarto, quinto e sexto cenário de teste realizado

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do quarto cenário de teste

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do quinto cenário de teste

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do sexto cenário de teste

O quarto, quinto e sexto cenário de teste envolve as possíveis permutações para que a máquina chegue ao estado VLR\_75, seguido de um pedido de devolução. Ao chegar ao estado DEVOLVE\_75, as saídas D50 e D25 são ativadas.

72 -- CENARIO 7: INIT -> VLR\_25 -> VLR\_50 -> VLR\_75 -> VLR\_100 -> INIT

73 --A: 01 01 01 01 XX

74 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

75 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

76 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

77 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

78 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

79 -- CENARIO 8: INIT -> VLR\_25 -> VLR\_50 -> VLR\_100 -> INIT

80 --A: 01 01 10 XX

81 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

82 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

83 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

84 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

85 -- CENARIO 9: INIT -> VLR\_25 -> VLR\_75 -> VLR\_100 -> INIT

86 --A: 01 10 01 XX

87 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

88 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

89 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

90 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

91 -- CENARIO 10: INIT -> VLR\_50 -> VLR\_100 -> INIT

92 --A: 10 10 XX

93 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

94 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

95 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

Código :Código do sétimo, oitavo, nono e décimo cenário de teste realizado

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do sétimo cenário de teste

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do oitavo cenário de teste

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do nono cenário de teste

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do décimo cenário de teste

Os cenários 7 a 10 representam 4 possíveis maneiras de chegar ao estado VLR\_100. Evidencia-se que, ao chegar ao estado VLR\_100, apenas a saída L é ativada.

96 -- CENARIO 11: INIT -> VLR\_25 -> VLR\_50 -> VLR\_75 -> VLR\_125 -> INIT

97 --A: 01 01 01 10 XX

98 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

99 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

100 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

101 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

102 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

103 -- CENARIO 12: INIT -> VLR\_25 -> VLR\_75 -> VLR\_125 -> INIT

104 --A: 01 10 10 XX

105 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

106 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

107 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

108 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

109 -- CENARIO 13: INIT -> VLR\_50 -> VLR\_75 -> VLR\_125 -> INIT

110 --A: 10 01 10 XX

111 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

112 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

113 A\_TB **<=** "10"**;** **WAIT** **FOR** 10 NS**;**

114 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

Código :Código do décimo primeiro, décimo segundo e décimo terceiro cenário de teste realizado

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do décimo primeiro cenário de teste

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do décimo segundo cenário de teste

A screenshot of a computer

AI-generated content may be incorrect.

Figura :Resultado do décimo terceiro cenário de teste

Os cenários 11,12 e 13 envolvem diferentes percursos para chegar ao estado VLR\_125. Neste estado, as saídas L e D25 são ativadas.

115 -- CENARIO 14: INIT -> INIT -> VLR\_25 -> VLR\_25 -> VLR\_50 ->

116 -- 00 01 00 01 00

117 -- VLR\_50 -> VLR\_75 -> VLR\_75 -> VLR\_100 -> INIT

118 -- 01 00 01 00

119 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

120 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

121 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

122 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

123 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

124 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

125 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

126 A\_TB **<=** "01"**;** **WAIT** **FOR** 10 NS**;**

127 A\_TB **<=** "00"**;** **WAIT** **FOR** 10 NS**;**

Figura :Código do décimo quarto cenário de teste realizado

A screenshot of a video game

AI-generated content may be incorrect.

Figura :Resultado do décimo quarto cenário de teste

O último teste realizado visou avaliar o comportamento da máquina nos casos de inatividade do usuário (entrada 00). Podemos ver que nestes casos, o aparentemente estado se manteve; na prática o que ocorre é uma transição para o mesmo estado.