

**UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA**  
**DEPARTAMENTO DE TECNOLOGIA**  
**TEC498 PROJETO DE CIRCUITOS DIGITAIS**  
**DESENVOLVIMENTO DO CIRCUITO DIGITAL DE UMA MÁQUINA DE CAFÉ**

Douglas Oliveira de Jesus

**Tutor:** Armando Sanca Sanca

## **RESUMO**

*Este relatório técnico descreve os seguintes passos para o desenvolvimento do circuito digital de uma máquina de café: desenho do diagrama de transição de estados, bem como sua tabela e minimização dos estados; codificação; escolha dos elementos de memória; tabela de excitação; equações de excitação e de saída e desenho do circuito. A produção foi solicitada pela disciplina de M.I de Circuitos Digitais, do curso de Engenharia de Computação da Universidade Estadual de Feira de Santana, utilizando a metodologia PBL (Problem Based Learning).*

**Palavras-chaves:** Máquina de Estados, MEF, Circuitos Digitais.

## **1 INTRODUÇÃO**

A indústria do café é um dos setores mais importantes e lucrativos do mercado de bebidas. Com a crescente demanda por café em todo o mundo e seguindo o problema fictício proposto pela disciplina do Módulo Integrador de Circuitos Digitais com base na metodologia PBL, a empresa MANDACARU SA identificou uma oportunidade de negócio e decidiu investir no desenvolvimento de Vending Machines para atender aos amantes da bebida durante seus "coffee breaks" em ambientes de trabalho.

Nesse contexto, a MANDACARU SA já desenvolveu um protótipo de hardware para a Coffee Machine, que consiste em um compartimento de água, um compartimento para as cápsulas, um compartimento para copos, botões de seleção dos produtos, um dispensador, uma bomba de pressurização e um circuito elétrico de aquecimento por indução. No entanto, para automatizar o processo de vendas, é necessário implementar um circuito digital capaz de reconhecer e validar as moedas inseridas pelos usuários.

O objetivo deste relatório técnico é apresentar uma proposta de construção lógica dos circuitos internos da Coffee Machine, utilizando máquinas de estados, a fim de garantir o correto funcionamento do sistema de vendas. Serão abordadas as principais características do protótipo, como os sensores de nível mínimo no compartimento de água e os sensores de presença nos compartimentos de cápsulas e de recepção de copos. Além disso, serão discutidos os requisitos

específicos que o circuito digital deve atender, como a validação das moedas de acordo com os preços dos produtos, a exibição de mensagens de erro e a correta sequência de acionamento dos circuitos internos.

Para atingir esse objetivo, serão realizadas atividades como a análise detalhada dos requisitos do sistema, o projeto da lógica de funcionamento utilizando máquinas de estados, a seleção dos componentes eletrônicos adequados e a implementação do circuito digital. Serão consideradas também as diretrizes estabelecidas, como a exibição de informações relevantes no display da máquina, a utilização de LEDs para representar as saídas do sistema e a correta identificação e tratamento de situações de erro.

Ao final deste relatório, espera-se apresentar uma solução técnica eficiente e funcional para automatizar o processo de vendas da Coffee Machine, garantindo uma experiência satisfatória para os usuários e contribuindo para o sucesso do empreendimento da MANDACARU SA no mercado de café.

O projeto em questão requer a implementação de um circuito digital para a Coffee Machine, a fim de automatizar o processo de vendas através do reconhecimento de moedas. Dessa forma, será possível garantir a correta seleção e pagamento dos produtos disponíveis na máquina.

As funcionalidades essenciais que o circuito digital deve oferecer incluem a validação das moedas inseridas, verificando se correspondem aos valores dos produtos (café expresso, café com leite, chá de camomila e cappuccino), exibindo mensagens de erro caso seja detectada uma moeda inválida e devolvendo as cédulas ou moedas ao cliente. Além disso, o circuito deve monitorar os níveis mínimos de água, a presença de cápsulas e a disponibilidade de copos, exibindo mensagens de erro apropriadas caso algum desses elementos esteja em falta.

Para proporcionar uma interação clara e intuitiva com o usuário, a Coffee Machine deverá contar com um display que informe o produto escolhido, o valor total das cédulas inseridas e eventuais códigos de erro associados a ações incorretas. Os resultados das ações realizadas pelo circuito interno também serão representados por LEDs, que servirão como indicadores visuais das saídas do sistema.

É importante destacar que a máquina deve permanecer funcional mesmo em caso de situações adversas, evitando travamentos durante o processo de venda. Para isso, será necessário desenvolver uma lógica eficiente que permita a correta sequência de acionamento dos circuitos internos, ativando o aquecimento por indução 2 segundos após o início da pressurização de água, possibilitando a extração adequada do café e sua posterior disponibilização ao cliente pelo dispensador.

## **2 METODOLOGIA**

### **2.1 RECURSOS UTILIZADOS**

Em todo o processo de criação do protótipo, algumas ferramentas foram utilizadas como auxiliaadoras no processo.

Para desenvolvimento das tabelas, foi utilizado a plataforma do Google Sheets, disponível virtualmente. Para o desenho do circuito, o software Logisim foi utilizado como uma ferramenta poderosa de desenho e simulação. Na parte de registro, o Overleaf foi utilizado como uma ferramenta que permite escrever em LaTeX, utilizando o modelo da ABNT disponibilizado no site do Módulo Integrador de Circuitos Digitais da UEFS. O site foi escolhido por ter uma ótima estruturação de escrita e permitir o desenvolvimento em grupo dentro da plataforma. Além do Overleaf, também para registros, o Trello foi utilizado como uma tabela de registro do que foi tratado nas sessões armazenando as ideias, fatos, questões e metas propostos pela metodologia PBL. Os recursos bibliográficos estão apresentados na sessão de referência deste relatório.

## 2.2 ESCOLHA DO MODELO DE MÁQUINA

Máquinas de estados são modelos matemáticos amplamente utilizados na área de engenharia de sistemas digitais para descrever o comportamento sequencial de sistemas. Essas máquinas são compostas por um conjunto finito de estados, transições entre esses estados e ações associadas a essas transições. Elas desempenham um papel fundamental no projeto e implementação de circuitos lógicos que respondem a estímulos externos de forma controlada e previsível.

No contexto do projeto da Coffee Machine, a escolha do modelo de máquina de estados se baseia na necessidade de descrever o comportamento sequencial do sistema, que envolve desde a inserção e validação das moedas até a entrega do produto escolhido ao cliente. As máquinas de estados oferecem uma estrutura clara e organizada para especificar todas as etapas e condições necessárias para o funcionamento adequado do sistema.

A escolha do modelo de Mealy para a Coffee Machine tem como objetivo aproveitar a capacidade de considerar as entradas durante as transições de estados para otimizar o controle do sistema (TOCCI; WIDMER; MOSS, 2010). Isso é particularmente relevante para o processo de validação das moedas, onde a decisão de aceitar ou rejeitar uma moeda depende tanto do estado atual quanto do valor inserido.

## 2.3 DESENHO DO DIAGRAMA DE TRANSIÇÃO DE ESTADOS

No desenho do diagrama de transição de estados 1, o texto que descreve as entradas estão na cor preta e o texto que descreve as saídas estão na cor verde. As setas pretas seguem um fluxo a partir do estado inicial, enquanto as de outras cores voltam para o estado inicial, seja devido aos erros ou devido a finalização do processo.

As saídas estão representadas pelos códigos: E0, E1, E2, E3, S0, S1, S2, S3, P, AQ, PP. As saídas que irão acionar o display de led para demonstração de erros são: E0 (para o sensor SR), E1 (para o sensor SP), E2 (para o sensor SN) e E3 (valor não correspondente com bebida selecionada). Para o preparo da bebida: S0 (café expresso), S1 (café com leite), S2 (camomila) e S3 (cappuccino). Para representar o processo de preparo, seguem as seguintes saídas: P (pressurizador), AQ (aquecedor) e PP (pisca-pisca, que representa a liberação da bebida). Por fim, o M representa o modo de espera.

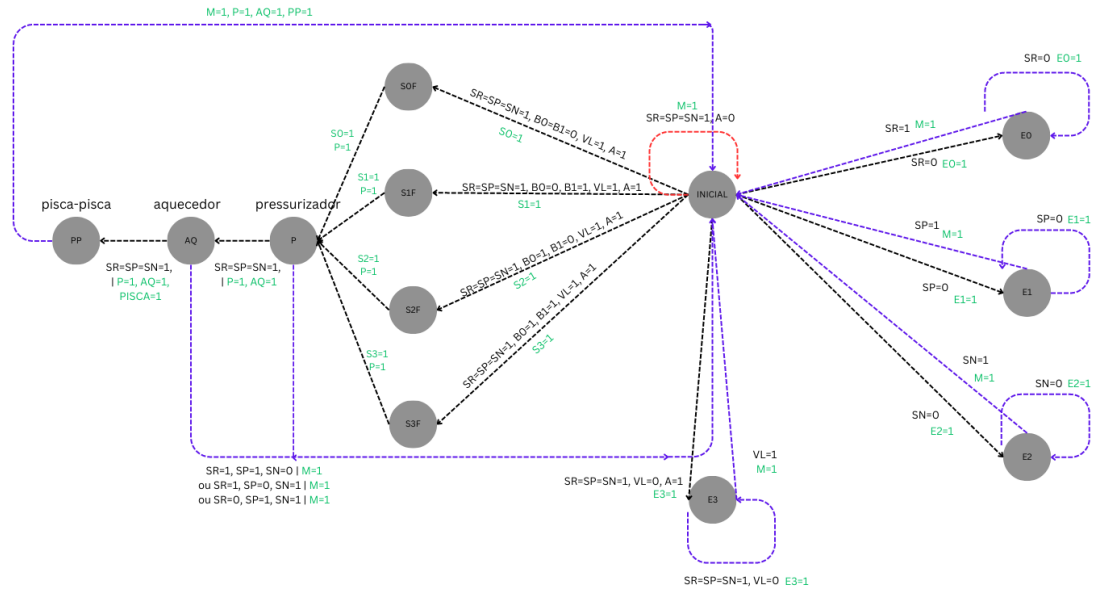


Figura 1 – Desenho do diagrama de transição de estados

As entradas estão representadas pelos códigos: SR, SP, SN, A, VL (V0 e V1), B0, B1. As entradas dos sensores são o SR (copos), SP (cápsulas) e SN (nível mínimo). Para acionamento do pedido, há a entrada A. O VL entrará como um validador do valor selecionado. Se o valor selecionado (V0 e V1) estiver de acordo com a bebida selecionada (B0 e B1), VL será nível lógico alto (1). Caso contrário, será nível lógico baixo (0). Para efetuar a validação, as entradas V0 e V1 entraram no sistema, mas não entrarão na MEF, responsabilizando a entrada VL da MEF. Por fim, B0 e B1 são as entradas que representam a seleção da bebida.

Os estados irão ter um atraso de transição de 2 segundos, para todas as exibições previstas a partir da saída serem efetuadas.

## 2.4 TABELA DE TRANSIÇÃO DE ESTADOS E REDUÇÃO DE ESTADOS

Devido a máquina obter 7 entradas, 128 possibilidades são registradas apenas a partir do estado inicial. Tendo a noção que nem todas as possibilidades precisam ser registradas, já que várias entradas podem ser utilizadas como sem importância para as equações de excitação, a tabela foi definida de acordo com todas as possibilidades possíveis abstraídas.

Os estados apresentados foram "INICIAL", que é o estado ideal de início da MEF, "EE0", "EE1", "EE2", "EE3", que representam os estados de erro dos sensores SR, SP, SN e da validação do valor e bebida, respectivamente, "ES0", "ES1", "ES2", "ES3", que representam a escolha da bebida com o valor inserido corretamente, "P" para pressurização, "AQ" para aquecimento e "PP" para saída final, que será o acionamento contínuo de um LED. A Figura 2 apresenta a tabela desenvolvida.

ESTADO ATUAL	ENTRADAS							PRÓXIMO ESTADO	SAÍDAS											
	SR	SP	SN	A	VL	B0	B1		E0	E1	E2	E3	S0	S1	S2	S3	P	AQ	PP	M
INICIAL	0	X	X	X	X	X	X	EE0	1	0	0	0	0	0	0	0	0	0	0	0
INICIAL	1	0	X	X	X	X	X	EE1	0	1	0	0	0	0	0	0	0	0	0	0
INICIAL	1	1	0	X	X	X	X	EE2	0	0	1	0	0	0	0	0	0	0	0	0
INICIAL	1	1	1	0	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
INICIAL	1	1	1	1	0	X	X	EE3	0	0	0	1	0	0	0	0	0	0	0	0
INICIAL	1	1	1	1	1	0	0	ES0	0	0	0	0	1	0	0	0	0	0	0	0
INICIAL	1	1	1	1	1	0	1	ES1	0	0	0	0	0	1	0	0	0	0	0	0
INICIAL	1	1	1	1	1	1	0	ES2	0	0	0	0	0	0	1	0	0	0	0	0
INICIAL	1	1	1	1	1	1	1	ES3	0	0	0	0	0	0	0	1	0	0	0	0
EE0	0	X	X	X	X	X	X	EE0	1	0	0	0	0	0	0	0	0	0	0	0
EE0	1	X	X	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
EE1	X	0	X	X	X	X	X	EE1	0	1	0	0	0	0	0	0	0	0	0	0
EE1	X	1	X	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
EE2	X	X	0	X	X	X	X	EE2	0	0	1	0	0	0	0	0	0	0	0	0
EE2	X	X	1	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
EE3	X	X	X	X	0	X	X	EE3	0	0	0	1	0	0	0	0	0	0	0	0
EE3	X	X	X	X	1	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
ES0	X	X	X	X	X	X	X	P	0	0	0	0	1	0	0	0	1	0	0	0
ES1	X	X	X	X	X	X	X	P	0	0	0	0	0	1	0	0	1	0	0	0
ES2	X	X	X	X	X	X	X	P	0	0	0	0	0	0	1	0	1	0	0	0
ES3	X	X	X	X	X	X	X	P	0	0	0	0	0	0	0	1	1	0	0	0
P	1	1	1	X	X	X	X	AQ	0	0	0	0	0	0	0	0	1	1	0	0
P	1	1	0	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
P	1	0	1	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
P	0	1	1	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
AQ	1	1	1	X	X	X	X	PP	0	0	0	0	0	0	0	0	1	1	1	0
AQ	1	1	0	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
AQ	1	0	1	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
AQ	0	1	1	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	0	0	0	1
PP	X	X	X	X	X	X	X	INICIAL	0	0	0	0	0	0	0	0	1	1	1	1

Figura 2 – Tabela da transição de estados reduzida

## 2.5 CODIFICAÇÃO DE ESTADOS E MODIFICAÇÃO DA TABELA DE TRANSIÇÃO DE ESTADOS

Como são 12 estados, serão necessários 4 flip-flops para construir a MEF, precisando de uma codificação de 4 bits. Foi decidido a seguinte representação:

- INICIAL = 0000
- EE0 = 0001
- EE1 = 0010
- EE2 = 0011
- EE3 = 0100
- ES0 = 0101

- ES1 = 0110
- ES2 = 0111
- ES3 = 1000
- P = 1001
- AQ = 1010
- PP = 1011

A figura 3 apresenta a codificação aplicada na tabela.

ESTADO ATUAL	ENTRADAS							PRÓXIMO ESTADO	SAÍDAS											
	SR	SP	SN	A	VL	B0	B1		E0	E1	E2	E3	S0	S1	S2	S3	P	AQ	PP	M
0000	0	X	X	X	X	X	X	0001	1	0	0	0	0	0	0	0	0	0	0	0
0000	1	0	X	X	X	X	X	0010	0	1	0	0	0	0	0	0	0	0	0	0
0000	1	1	0	X	X	X	X	0011	0	0	1	0	0	0	0	0	0	0	0	0
0000	1	1	1	0	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0000	1	1	1	1	0	X	X	0100	0	0	0	1	0	0	0	0	0	0	0	0
0000	1	1	1	1	1	0	0	0101	0	0	0	0	1	0	0	0	0	0	0	0
0000	1	1	1	1	1	0	1	0110	0	0	0	0	0	1	0	0	0	0	0	0
0000	1	1	1	1	1	1	0	0111	0	0	0	0	0	0	1	0	0	0	0	0
0000	1	1	1	1	1	1	1	1000	0	0	0	0	0	0	0	1	0	0	0	0
0001	0	X	X	X	X	X	X	0001	1	0	0	0	0	0	0	0	0	0	0	0
0001	1	X	X	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0010	X	0	X	X	X	X	X	0010	0	1	0	0	0	0	0	0	0	0	0	0
0010	X	1	X	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0011	X	X	0	X	X	X	X	0011	0	0	1	0	0	0	0	0	0	0	0	0
0011	X	X	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0100	X	X	X	X	0	X	X	0100	0	0	0	1	0	0	0	0	0	0	0	0
0100	X	X	X	X	1	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0101	X	X	X	X	X	X	X	1001	0	0	0	0	1	0	0	0	1	0	0	0
0110	X	X	X	X	X	X	X	1001	0	0	0	0	0	1	0	0	1	0	0	0
0111	X	X	X	X	X	X	X	1001	0	0	0	0	0	0	1	0	1	0	0	0
1000	X	X	X	X	X	X	X	1001	0	0	0	0	0	0	0	1	1	0	0	0
1001	1	1	1	X	X	X	X	1010	0	0	0	0	0	0	0	0	1	1	0	0
1001	1	1	0	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1001	1	0	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1001	0	1	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1010	1	1	1	X	X	X	X	1011	0	0	0	0	0	0	0	0	1	1	1	0
1010	1	1	0	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1010	1	0	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1010	0	1	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1011	X	X	X	X	X	X	X	0000	0	0	0	0	0	0	0	0	1	1	1	1

Figura 3 – Tabela da transição de estados codificada

## 2.6 ESCOLHA DOS ELEMENTOS DE MEMÓRIA

Para armazenar e representar os estados da máquina de estados, o flip-flop do tipo D é utilizado. Esse tipo de flip-flop possui uma entrada de dados (D) que determina o valor a ser

armazenado e uma entrada de clock (CLK) que controla o momento de atualização do valor armazenado.

A escolha do flip-flop do tipo D se justifica pela sua simplicidade e facilidade de utilização, além de ser amplamente disponível e de baixo custo. Ele oferece a capacidade de armazenar e atualizar os estados da máquina de forma confiável e sincronizada com o clock do sistema (FLOYD, 2007).

Dessa forma, a combinação do modelo de máquina de estados de Mealy com o uso do flip-flop do tipo D proporciona uma abordagem adequada para o projeto da Coffee Machine. Essa escolha permite a especificação e implementação eficiente do controle do sistema, levando em consideração tanto o estado atual quanto as entradas relevantes para a geração de saída adequada.

## 2.7 CONSTRUÇÃO DA TABELA E OBTENÇÃO DAS EQUAÇÕES DE EXCITAÇÃO

Para gerar a tabela de excitação a partir de uma tabela codificada, é necessário tomar cada bit da quantidade de bits da codificação dos estados como a saída de um flip-flop. Como são 4 bits, serão 4 flip-flops. A expressão de saída pode ser construída através das situações onde a saída do flip-flop é 1. Para isso, basta registrar o estado atual de cada variável (saídas dos flip-flops atuais mais as entradas) através de min-terms. Uma forma de simplificar é utilizando o Mapa de Karnaugh, entretanto o mesmo não é recomendável para a quantidade de variáveis que são necessárias para cada termo da expressão da saída do flip-flop. Por isso, a simplificação será através da álgebra booleana.

De acordo com a análise da Figura 4 é possível verificar quais são as expressões obtidas, utilizando a estratégia de min-terms.

O flip-flop 3 (D3) tem a seguinte expressão de saída:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.B0.B1 + \tilde{w}3.w2.\tilde{w}1.w0 + \tilde{w}3.w2.w1.\tilde{w}0 + \tilde{w}3.w2.w1.w0 + w3.\tilde{w}2.\tilde{w}1.\tilde{w}0 + w3.\tilde{w}2.\tilde{w}1.w0.SR.SP.SN.SP + w3.\tilde{w}2.w1.\tilde{w}0.SR.SP.SN.SP$

O flip-flop 2 (D2) tem a seguinte expressão de saída:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.\tilde{V}L + \tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.\tilde{B}0.\tilde{B}1 + \tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.\tilde{B}0.B1 + \tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SP.SN.A.VL.B0.\tilde{B}1 + \tilde{w}3.w2.\tilde{w}1.\tilde{w}0.\tilde{V}L$

O flip-flop 1 (D1) tem a seguinte expressão de saída:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.\tilde{S}P + \tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.\tilde{S}N + \tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.\tilde{B}0.B1 + \tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.B0.\tilde{B}1 + \tilde{w}3.\tilde{w}2.w1.\tilde{w}0.\tilde{S}P + \tilde{w}3.\tilde{w}2.w1.w0.\tilde{S}N + w3.\tilde{w}2.\tilde{w}1.w0.SR.SP.SN + w3.\tilde{w}2.w1.\tilde{w}0.SR.SP.SN$

O flip-flop 0 (D0) tem a seguinte expressão de saída:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.\tilde{S}R + \tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.\tilde{S}N + \tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.\tilde{B}0.\tilde{B}1 + \tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.B0.\tilde{B}1 + \tilde{w}3.\tilde{w}2.\tilde{w}1.w0.\tilde{S}R + \tilde{w}3.\tilde{w}2.w1.w0.\tilde{S}N + w3.\tilde{w}2.w1.\tilde{w}0.SR.SP.SN + \tilde{w}3.w2.\tilde{w}1.w0 + \tilde{w}3.w2.w1.\tilde{w}0 + w3.\tilde{w}2.\tilde{w}1.\tilde{w}0$

Os bits "w3", "w2", "w1" e "w0" representam as saídas dos flip-flops D3, D2, D1 e D0, respectivamente, também conhecidas comumente como "Q". O símbolo "." representa uma porta AND e o símbolo "+" representa uma porta OR.



ESTADO ATUAL	ENTRADAS							PRÓXIMO ESTADO	SAÍDAS											
w3w2w1w0	SR	SP	SN	A	VL	B0	B1	d3d2d1d0	E0	E1	E2	E3	S0	S1	S2	S3	P	AQ	PP	M
0000	0	X	X	X	X	X	X	0001	1	0	0	0	0	0	0	0	0	0	0	0
0000	1	0	X	X	X	X	X	0010	0	1	0	0	0	0	0	0	0	0	0	0
0000	1	1	0	X	X	X	X	0011	0	0	1	0	0	0	0	0	0	0	0	0
0000	1	1	1	0	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0000	1	1	1	1	0	X	X	0100	0	0	0	1	0	0	0	0	0	0	0	0
0000	1	1	1	1	1	0	0	0101	0	0	0	0	1	0	0	0	0	0	0	0
0000	1	1	1	1	1	0	1	0110	0	0	0	0	0	1	0	0	0	0	0	0
0000	1	1	1	1	1	1	0	0111	0	0	0	0	0	0	1	0	0	0	0	0
0000	1	1	1	1	1	1	1	1000	0	0	0	0	0	0	0	1	0	0	0	0
0001	0	X	X	X	X	X	X	0001	1	0	0	0	0	0	0	0	0	0	0	0
0001	1	X	X	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0010	X	0	X	X	X	X	X	0010	0	1	0	0	0	0	0	0	0	0	0	0
0010	X	1	X	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0011	X	X	0	X	X	X	X	0011	0	0	1	0	0	0	0	0	0	0	0	0
0011	X	X	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0100	X	X	X	X	0	X	X	0100	0	0	0	1	0	0	0	0	0	0	0	0
0100	X	X	X	X	1	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
0101	X	X	X	X	X	X	X	1001	0	0	0	0	1	0	0	0	1	0	0	0
0110	X	X	X	X	X	X	X	1001	0	0	0	0	0	1	0	0	1	0	0	0
0111	X	X	X	X	X	X	X	1001	0	0	0	0	0	0	1	0	1	0	0	0
1000	X	X	X	X	X	X	X	1001	0	0	0	0	0	0	0	1	1	0	0	0
1001	1	1	1	X	X	X	X	1010	0	0	0	0	0	0	0	0	1	1	0	0
1001	1	1	0	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1001	1	0	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1001	0	1	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1010	1	1	1	X	X	X	X	1011	0	0	0	0	0	0	0	0	1	1	1	0
1010	1	1	0	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1010	1	0	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1010	0	1	1	X	X	X	X	0000	0	0	0	0	0	0	0	0	0	0	0	1
1011	X	X	X	X	X	X	X	0000	0	0	0	0	0	0	0	0	1	1	1	1

Figura 4 – Tabela das equações de excitação

Contudo, as expressões acima não estão simplificadas. Com a simplificação, e a fatoração de alguns termos, através da álgebra de boole, as equações ficam da seguinte forma:

O flip-flop 3 (D3):  $\tilde{w}2.\tilde{w}0(SR.SP.SN.(w3 + \tilde{w}1.A.VL.B0.B1) + \tilde{w}2.\tilde{w}0) + \tilde{w}3.w2.(w1 + w0)$

O flip-flop 2 (D2):  $\tilde{w}3.\tilde{w}1.\tilde{w}0.(\tilde{w}2.SR.SP.SN.A.(\tilde{V}L + \tilde{B}0 + \tilde{B}1) + w2.\tilde{V}L)$

O flip-flop 1 (D1):  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.(\tilde{S}P + \tilde{S}N + SN.A.VL.(\tilde{B}0.B1 + B0.\tilde{B}1)) + \tilde{w}3.\tilde{w}2.w1(\tilde{w}0.\tilde{S}P + w0.\tilde{S}N) + w3.\tilde{w}2.SR.SP.SN(\tilde{w}1.w0 + w1.\tilde{w}0)$

O flip-flop 0 (D0):  $\tilde{w}2.\tilde{w}0(\tilde{w}3.\tilde{w}1(\tilde{S}R + SP.(\tilde{S}N + A.VL.\tilde{B}1)) + w3.w1.SR.SP.SN) + \tilde{w}3.\tilde{w}2.w0.(\tilde{w}1.\tilde{S}R + w1.\tilde{S}N)$

## 2.8 OBTENÇÃO DAS EQUAÇÕES DE SAÍDA

No caso específico deste projeto, o modelo de máquina de estados escolhido é o modelo de Mealy. Nesse modelo, as saídas do sistema dependem tanto do estado atual quanto das entradas



no momento da transição entre os estados. Isso significa que a geração de saída é determinada pelo estado em que a máquina se encontra e pelas entradas recebidas nesse momento, permitindo uma maior flexibilidade e adaptabilidade do sistema. Portanto, as equações de saída terão tanto os estados quanto as entradas.

A saída E0 tem a seguinte expressão:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.\tilde{S}R + \tilde{w}3.\tilde{w}2.\tilde{w}1.w0.\tilde{S}R$

A saída E1 tem a seguinte expressão:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.\tilde{S}P + \tilde{w}3.\tilde{w}2.w1.\tilde{w}0.\tilde{S}P$

A saída E2 tem a seguinte expressão:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.\tilde{S}N + \tilde{w}3.\tilde{w}2.w1.w0.\tilde{S}N$

A saída E3 tem a seguinte expressão:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.\tilde{V}L + \tilde{w}3.w2.\tilde{w}1.\tilde{w}0.\tilde{V}L$

A saída S0 tem a seguinte expressão:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.\tilde{B}0.\tilde{B}1 + \tilde{w}3.w2.\tilde{w}1.w0$

A saída S1 tem a seguinte expressão:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.\tilde{B}0.B1 + \tilde{w}3.w2.w1.\tilde{w}0$

A saída S2 tem a seguinte expressão:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.B0.\tilde{B}1 + \tilde{w}3.w2.w1.w0$  A saída S3 tem a seguinte expressão:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.B0.B1 + w3.\tilde{w}2.\tilde{w}1.\tilde{w}0$

A saída P tem a seguinte expressão:  $\tilde{w}3.w2.\tilde{w}1.w0 + \tilde{w}3.w2.w1.\tilde{w}0 + \tilde{w}3.w2.w1.w0 + w3.\tilde{w}2.\tilde{w}1.\tilde{w}0 + w3.\tilde{w}2.\tilde{w}1.w0.SR.SP.SN + w3.\tilde{w}2.w1.\tilde{w}0.SR.SP.SN + w3.\tilde{w}2.w1.w0$

A saída AQ tem a seguinte expressão:  $w3.\tilde{w}2.\tilde{w}1.w0.SR.SP.SN + w3.\tilde{w}2.w1.w0 + w3.\tilde{w}2.w1.\tilde{w}0.SR.SP.SN$

A saída PP tem a seguinte expressão:  $w3.\tilde{w}2.w1.\tilde{w}0.SR.SP.SN + w3.\tilde{w}2.w1.w0$

A saída M tem a seguinte expressão:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.\tilde{A} + \tilde{w}3.\tilde{w}2.\tilde{w}1.w0.SR + \tilde{w}3.\tilde{w}2.w1.\tilde{w}0.SP + \tilde{w}3.\tilde{w}2.w1.w0.SN + \tilde{w}3.w2.\tilde{w}1.\tilde{w}0.VL + w3.\tilde{w}2.\tilde{w}1.w0.SR.SP.\tilde{S}N + w3.\tilde{w}2.\tilde{w}1.w0.SR.\tilde{S}P.SN + w3.\tilde{w}2.\tilde{w}1.w0.\tilde{S}R.SP.SN + w3.\tilde{w}2.w1.\tilde{w}0.SR.SP.\tilde{S}N + w3.\tilde{w}2.w1.\tilde{w}0.SR.\tilde{S}P.SN + w3.\tilde{w}2.w1.\tilde{w}0.\tilde{S}R.SP.SN + w3.\tilde{w}2.w1.w0$

Após a simplificação e fatoração de alguns termos utilizando a álgebra de boole, as equações das saídas ficaram da seguinte forma:

Saída E0:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{S}R$

Saída E1:  $\tilde{w}3.\tilde{w}2.\tilde{w}0.\tilde{S}P(SR + w1)$

Saída E2:  $\tilde{w}3.\tilde{w}2.\tilde{S}N(\tilde{w}1.\tilde{w}0.SR.SP + w1.w0)$

Saída E3:  $\tilde{w}3.\tilde{w}1.\tilde{w}0.\tilde{V}L(SR.SP.SN.A + w2)$

Saída S0:  $\tilde{w}3.\tilde{w}1(\tilde{w}2.\tilde{w}0.SR.SP.SN.A.VL.\tilde{B}0.\tilde{B}1 + w2.w0)$

Saída S1:  $\tilde{w}3.\tilde{w}0(\tilde{w}2.\tilde{w}1.SR.SP.SN.A.VL.\tilde{B}0.B1 + w2.w1)$

Saída S2:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.\tilde{w}0.SR.SP.SN.A.VL.B0.\tilde{B}1 + \tilde{w}3.w2.w1.w0$

Saída S3:  $\tilde{w}2.\tilde{w}1.\tilde{w}0(SR.SP.SN.A.VL.B0.B1 + w3)$

Saída P:  $\tilde{w}3.w2.(\tilde{w}1.w0 + w1) + w3.\tilde{w}2.(SR.SP.SN.(\tilde{w}1.w0. + w1.\tilde{w}0) + w1.w0 + \tilde{w}1.\tilde{w}0)$

Saída AQ:  $w3.\tilde{w}2.SR.SP.SN.(\tilde{w}1.w0. + w1.\tilde{w}0) + w3.\tilde{w}2.w1.w0$

Saída PP:  $w3.\tilde{w}2.w1.(SR.SP.SN + w0)$

Saída M:  $\tilde{w}3.\tilde{w}2.\tilde{w}1.SR(SP.SN.\tilde{A} + w0) + \tilde{w}3.\tilde{w}2.w1.(\tilde{w}0.SP + w0.SN) + \tilde{w}3.w2.\tilde{w}1.\tilde{w}0.VL$



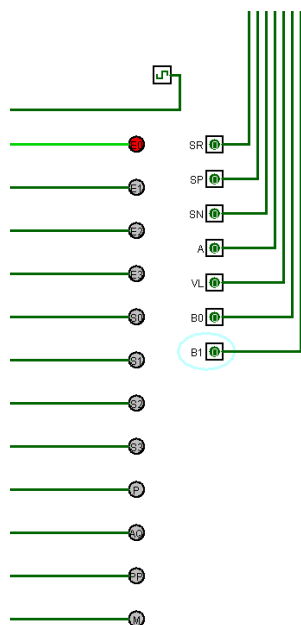


Figura 6 – Simulação com entradas 0000000 e saídas 1000000000000

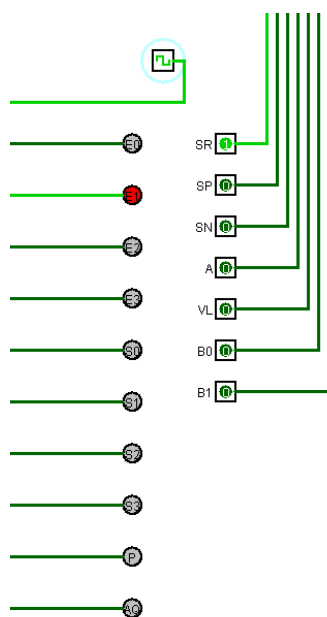


Figura 7 – Desenho do circuito

Como é possível verificar na simulação da Figura 6, todas as entradas possuem nível lógico baixo, com isso a saída E0 é apresentada, informando que o sensor SR está recebendo nível lógico baixo.

Na segunda simulação (Figura 7), o sensor SR envia nível lógico alto, mas as outras entradas continuam recebendo nível lógico baixo, o que possibilita que a saída E1 seja apresentada, informando que o sensor SP está enviando nível lógico baixo.

Com todos os sensores enviando nível lógico alto (Figura 8), a MEF entra em estado inicial. A saída apresentada é o modo de espera. Com isso o usuário pode modificar as entradas para acionar o preparo.

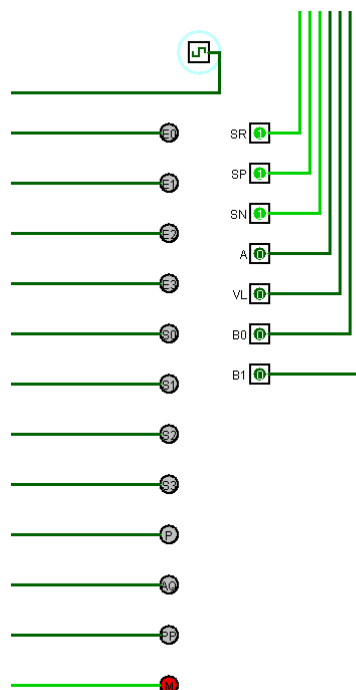


Figura 8 – do circuito lógico

Por fim, na Figura 9 é possível verificar, após acionamento com a entrada "A", com todos os sensores em nível lógico alto, junto com o VL, e a escolha de bebida B0=0 e B1=1, a saída S1 é apresentada, informando o tipo de bebida selecionado.

Outras simulações foram feitas, mas devido a quantidade de possibilidades de testes a serem feitos, não cabe a inserção neste relatório técnico.

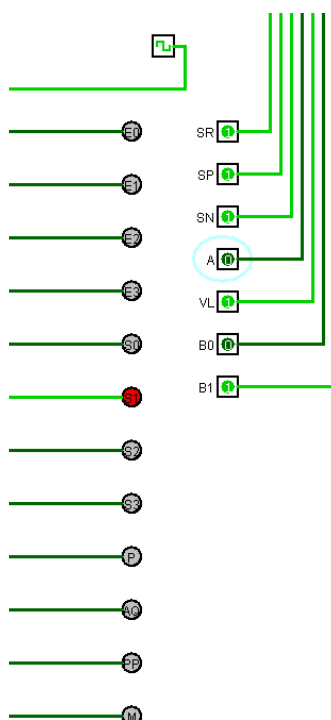


Figura 9 – Desenho d lógico

## 2.11 DESCRIÇÃO DO CÓDIGO

O código gerado contém a descrição do módulo chamado "MainCoffeeMachine". Esse módulo é parte principal da lógica da máquina de café e desempenha um papel crucial na coordenação das funcionalidades da máquina. O módulo possui várias entradas e saídas, incluindo sinais de controle como CLK, SR, SP, SN, A, V0, V1, B0 e B1, bem como sinais de saída para o display de sete segmentos (a, b, c, d, e, f, g, h, dig1, dig2, dig3, dig4) e para os LEDs (P, AQ, PP). O sinal de reticências ("...") na imagem representa continuação do código seguindo a mesma lógica do trecho posterior.

```
module MainCoffeeMachine(CLK, SR, SP, SN, A, V0, V1, B0, B1, P, AQ, PP, a, b, c, d, e, f, g, h, dig1,
dig2, dig3, dig4);

    ///////////////////////////////////////////////////HEAD
    input CLK;
    input SR, SP, SN, V0, V1, B0, B1; //CHAVE
    input A; //BOTÃO
    output a, b, c, d, e, f, g, h, dig1, dig2, dig3, dig4; //DISPLAY DE SETE SEGMENTOS
    output P, AQ; //LEDS
    output PP; //LED INDIVIDUAL
    wire VL, E0, E1, E2, E3, S0, S1, S2, S3, M;
    ///////////////////////////////////////////////////HEAD

    ///////////////////////////////////////////////////BODY
    //Determinação do VL: validar V0, V1 e B0 e B1 se são iguais.
    and (val1, ~V0, ~B0, ~V1, ~B1);
    and (val2, ~V0, ~B0, V1, B1);
    and (val3, V0, B0, ~V1, ~B1);
    and (val4, V0, B0, V1, B1);
    or (VL, val1, val2, val3, val4);

    MefCoffeeMachine INST(CLK, SR, SP, SN, A, VL, B0, B1, E0, E1, E2, E3, S0, S1, S2, S3, P, AQ, PP, M);

    Decoders INST2(CLK, S0, S1, S2, S3, SR, SP, SN, VL, M, a, b, c, d, e, f, g, h, dig1, dig2, dig3,
dig4);
    ///////////////////////////////////////////////////BODY

endmodule
```

Figura 10 – Módulo principal do código importando os módulos da MEF e dos decodificadores

Dentro do módulo, são declarados sinais de fio que serão utilizados posteriormente, como VL, E0, E1, E2, E3, S0, S1, S2, S3 e M. Esses sinais são usados para conectar os diferentes componentes do sistema e garantir o correto funcionamento da máquina de café.

Uma parte importante do código é a determinação do valor de VL. Isso é feito através do uso de portas lógicas, como AND e OR, que verificam as combinações das entradas V0, V1, B0 e B1. O objetivo é determinar se essas entradas são iguais ou diferentes, e o resultado é atribuído ao sinal VL.

Além disso, o módulo "MainCoffeeMachine" instancia dois outros módulos: "MefCoffeeMachine" e "Decoders". O primeiro é instanciado como "INST" e recebe diversas entradas e saídas, incluindo os sinais de controle, VL, B0, B1, e os sinais de saída P, AQ, PP e M. O segundo é instanciado como "INST2" e também recebe várias entradas e saídas, como os sinais de controle e os sinais de saída para o display de sete segmentos.

Esses módulos adicionais desempenham funções específicas dentro do sistema da máquina de café. O "MefCoffeeMachine" implementa a lógica de controle principal da máquina, enquanto o "Decoders" é responsável por converter os sinais de controle em sinais adequados para o display de sete segmentos, permitindo a exibição das informações através das interfaces.

O módulo "MefCoffeeMachine" é responsável por implementar a lógica de controle da máquina, permitindo que ela funcione corretamente. Dentro do módulo, são declarados sinais de fio que serão utilizados posteriormente, como Q3, Q2, Q1, Q0 e uma série de sinais w1 a w41 e s1 a s56. É utilizado quatro flip-flops (FFD) para armazenar o estado atual da máquina de café. Os sinais w1 a w41 são usados para codificar as equações de excitação dos estados e determinar o próximo estado com base nas entradas e no estado atual.

```

module MefCoffeeMachine(CLK, SR, SP, SN, A, VL, B0, B1, E0, E1, E2, E3, S0, S1, S2, S3, P, AQ, PP, M);
    ///////////////////////////////////////////////////HEAD
    input CLK, SR, SP, SN, A, VL, B0, B1;
    output E0, E1, E2, E3, S0, S1, S2, S3, P, AQ, PP, M;

    //Registrador de estado
    wire Q3, Q2, Q1, Q0;
    ...
    ///////////////////////////////////////////////////HEAD

    ///////////////////////////////////////////////////BODY
    //Codificação das equações de excitação dos estados. Como a máquina possui 12 estados, 4 flip-flops
    são utilizados.
    or (w1, Q1, Q0);
    and (w2, w1, Q2, ~Q3);
    and (w3, A, VL, B0, B1, ~Q1);
    or (w4, w3, Q3);
    and (w5, w4, SR, SP, SN);
    and (w6, w5, ~Q0, ~Q2);
    or (w7, w6, w2);
    //w7 = Q3
    ...
    //Estado próximo
    FFD D3(CLK, w7, Q3);
    FFD D2(CLK, w12, Q2);
    FFD D1(CLK, w27, Q1);
    FFD D0(CLK, w41, Q0);

    //Codificação das equações das saídas.
    and (E0, ~Q3, ~Q2, ~Q1, ~SR); //E0
    ...
    ///////////////////////////////////////////////////BODY
endmodule

```

Figura 11 – Módulo de implementação da Máquina de Estados Finitos

Em seguida, são codificadas as equações das saídas da máquina. Os sinais E0, E1, E2, E3, S0, S1, S2 e S3 representam as saídas específicas da máquina, que serão enviadas para os decodificadores como análise das interfaces do display de sete segmentos. Por fim, P, AQ e PP representam a pressurização, aquecimento e informação de bebida disponível para retirada, respectivamente. Esses sinais são enviados para o bloco principal.

O módulo MEF também instancia o módulo de delay, dando 2 segundos para as saídas do pressurizador e aquecedor serem disponibilizadas. Como, em sessão, foi permitido a utilização de um código comportamental, o Delay foi importado do módulo DelayedOutput. Como esse bloco

de código utiliza bastante recurso, foi apenas utilizado nas saídas do pressurizador e aquecedor, completando os requisitos solicitados no problema.

O módulo Decoders é responsável por enviar os sinais de acordo com a interface selecionada pela máquina de estados. O módulo possui uma série de decodificadores, que são comumente utilizados para exibir dígitos em displays de 7 segmentos, em que cada segmento pode ser ativado ou desativado para formar diferentes caracteres numéricos ou alfanuméricos. Os seguintes códigos alfanuméricos a serem exibidos no display de sete segmentos estão representados pelos seguintes sinais:

- "CE01": escolha do S0;
- "CL02": escolha do S1;
- "CC05": escolha do S2;
- "CP10": escolha do S3;
- "ERSR": erro do sensor SR;
- "ERSP": erro do sensor SP;
- "ERSN": erro do sensor SN;
- "ERDI": erro de validação do valor/dinheiro;
- "....": modo de espera.

O módulo "Decoders" possui diversas entradas e saídas que são utilizadas para controlar e exibir os dígitos no display. Para implementar a funcionalidade do decodificador de 7 segmentos, o módulo faz uso de outros módulos e interfaces. O módulo Contador é utilizado para gerar dois sinais de saída (saida1Contador e saida2Contador), que determinam o dígito atual a ser exibido no display. Esses sinais são utilizados como entrada para as interfaces correspondentes (InterfaceS0, InterfaceS1, InterfaceS2, InterfaceS3, InterfaceESR, InterfaceESP, InterfaceESN, InterfaceERDI), que recebem também os sinais de controle. Cada interface é responsável por gerar os sinais de saída correspondentes aos segmentos do display de 7 segmentos, dependendo das combinações específicas dos sinais de controle.

Além disso, o código implementa a lógica de seleção do dígito atual a ser exibido. Por meio de portas lógicas and, os sinais saida1Contador e saida2Contador são combinados com sinais complementares para ativar apenas um dígito por vez. Os sinais resultantes (dig1, dig2, dig3, dig4) são utilizados para selecionar o dígito apropriado no display.

Por fim, as portas nor são utilizadas para combinar os sinais dos segmentos individuais (A1 a A8, B1 a B8, etc.) em um único sinal para cada segmento do display (a, b, c, d, e, f, g). Esses sinais combinados representam os segmentos ativados ou desativados para exibir o dígito atual no display de 7 segmentos.

O sinal h é conectado diretamente à entrada M por meio de um buffer (buf) para exibir um ponto decimal, representando o modo de espera no display.



```

module Decoders(CLK, S0, S1, S2, S3, SR, SP, SN, VL, M, a, b, c, d, e, f, g, h, dig1, dig2, dig3, dig4);

    input CLK, S0, S1, S2, S3, SR, SP, SN, VL, M;
    output a, b, c, d, e, f, g, h, dig1, dig2, dig3, dig4;

    wire saida1Contador, saida2Contador;
    wire A1, A2, A3, A4, A5, A6, A7, A8,
         B1, B2, B3, B4, B5, B6, B7, B8,
        ...

    Contador(CLK, saida1Contador, saida2Contador);

    //Determinando as saídas do display de acordo com a interface escolhida
    InterfaceS0 (saida1Contador, saida2Contador, A1, B1, C1, D1, E1, F1, G1, S0, S1, S2, S3, SR, SP, SN, VL);
    InterfaceS1 (saida1Contador, saida2Contador, A2, B2, C2, D2, E2, F2, G2, S0, S1, S2, S3, SR, SP, SN, VL);
    InterfaceS2 (saida1Contador, saida2Contador, A3, B3, C3, D3, E3, F3, G3, S0, S1, S2, S3, SR, SP, SN, VL);
    InterfaceS3 (saida1Contador, saida2Contador, A4, B4, C4, D4, E4, F4, G4, S0, S1, S2, S3, SR, SP, SN, VL);
    InterfaceESR (saida1Contador, saida2Contador, A5, B5, C5, D5, E5, F5, G5, S0, S1, S2, S3, SR, SP, SN, VL);
    InterfaceESP (saida1Contador, saida2Contador, A6, B6, C6, D6, E6, F6, G6, S0, S1, S2, S3, SR, SP, SN, VL);
    InterfaceESN (saida1Contador, saida2Contador, A7, B7, C7, D7, E7, F7, G7, S0, S1, S2, S3, SR, SP, SN, VL);
    InterfaceERDI (saida1Contador, saida2Contador, A8, B8, C8, D8, E8, F8, G8, S0, S1, S2, S3, SR, SP, SN, VL);

    //Escolha do dígito do display
    and(dig1, ~saida1Contador, ~saida2Contador);
    and(dig2, ~saida1Contador, saida2Contador);
    and(dig3, saida1Contador, ~saida2Contador);
    and(dig4, saida1Contador, saida2Contador);

    //Portas nor porque os leds do display de 7 segmentos são acionados em nível lógico baixo
    nor (a, A1, A2, A3, A4, A5, A6, A7, A8);
    nor (b, B1, B2, B3, B4, B5, B6, B7, B8);
    nor (c, C1, C2, C3, C4, C5, C6, C7, C8);
    nor (d, D1, D2, D3, D4, D5, D6, D7, D8);
    nor (e, E1, E2, E3, E4, E5, E6, E7, E8);
    nor (f, F1, F2, F3, F4, F5, F6, F7, F8);
    nor (g, G1, G2, G3, G4, G5, G6, G7, G8);

    buf (h, M);

endmodule

```

Figura 12 – Módulo de implementação dos decodificadores das interfaces de saída

## 2.12 RECURSOS FÍSICOS

Sem o código comportamental do Delay, a implementação consumiu os seguintes recursos:

Entity	Logic Cells	LC Registers	UFM Blocks	Pins	Virtual Pins	LUT-Only LCs	Register-Only LCs	LUT/Register LCs	Carry Chain LCs	Packed LCs
MAX II: EPM240T100C5										
MainCoffeeMachine	93 (1)	6	0	24	0	87 (1)	0 (0)	6 (0)	0 (0)	0 (0)
MefCoffeeMachine:INST	47 (43)	4	0	0	0	43 (43)	0 (0)	4 (0)	0 (0)	0 (0)
FFD:D0	1 (1)	1	0	0	0	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)
FFD:D1	1 (1)	1	0	0	0	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)
FFD:D2	1 (1)	1	0	0	0	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)
FFD:D3	1 (1)	1	0	0	0	0 (0)	0 (0)	1 (1)	0 (0)	0 (0)
Decoders:INST2	45 (33)	2	0	0	0	43 (33)	0 (0)	2 (0)	0 (0)	0 (0)
Contador:comb_4	2 (0)	2	0	0	0	0 (0)	0 (0)	2 (0)	0 (0)	0 (0)
InterfaceS0:comb_5	1 (1)	0	0	0	0	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
InterfaceS1:comb_6	1 (1)	0	0	0	0	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
InterfaceS2:comb_7	1 (1)	0	0	0	0	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
InterfaceS3:comb_8	1 (1)	0	0	0	0	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
InterfaceESR:comb_9	0	0	0	0	0	0	0	0	0	0
InterfaceESP:comb_10	1 (1)	0	0	0	0	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
InterfaceESN:comb_11	1 (1)	0	0	0	0	1 (1)	0 (0)	0 (0)	0 (0)	0 (0)
InterfaceERDI:comb_12	4 (4)	0	0	0	0	4 (4)	0 (0)	0 (0)	0 (0)	0 (0)

Figura 13 – Tabela de consumo de recursos físicos do código

### **3 RESULTADOS**

Por fim, a MEF realizada apresentou resultados positivos em todos os testes realizados. Entretanto, isso não resume todo o projeto. No início houve dificuldade na escolha do tipo de máquina, em qual seria mais ideal para a implementação do problema proposto. Não obstante, com reforço no referencial teórico e na percepção da necessidade das saídas em lidarem com as entradas, foi definido o tipo de Mealy. Correções foram feitas em todo o processo e isso fez com que a construção tenha sido contínua, o que definiu um resultado mais embasado e fortificado em todos os seus processos. Com isso, o projeto está pronto para ser aplicado fisicamente com todas as adaptações necessárias.

## REFERÊNCIAS

FLOYD, T. *Sistemas Digitais: fundamentos e aplicações*. 7. ed. [S.l.]: Bookman Editora, 2007. ISBN 978-85-7780-107-7. Citado na página 7.

TOCCI, R. J.; WIDMER, N. S.; MOSS, G. L. *Sistemas digitais*. [S.l.]: Pearson Educación, 2010. Citado na página 3.