UFMG

UNIVERSIDADE FEDERAL DE MINAS GERAIS

ENGENHARIA DE SISTEMAS

Projeto de Máquina de Vendas em Nível RTL

Bianca Lopes
Gabriel Veloso

Belo Horizonte
2025

Conteúdo

1 Introdução

A crescente demanda por sistemas automatizados tem impulsionado o desenvolvimento de soluções inteligentes e eficientes, como as máquinas de vendas. Este trabalho tem como objetivo a elaboração de um projeto em nível RTL para uma máquina de vendas, enfatizando a integração entre hardware e lógica de controle digital. Para tanto, a arquitetura proposta divide-se em duas partes principais: o caminho de dados (Datapath), responsável pelas operações aritméticas e armazenamento, e a máquina de estados finitos (FSM), que orquestra a sequência das operações conforme os sinais de entrada.

Figura 1: Máquina de Refrigerantes

A abordagem modular adotada permite que o sistema seja facilmente adaptado a novas funcionalidades, como a inclusão de diferentes moedas ou a alteração dos preços dos produtos, além de facilitar a manutenção e a verificação formal do projeto.

1.1 Objetivo

O presente projeto visa:

- Desenvolver uma arquitetura RTL que atenda aos requisitos funcionais de uma máquina de vendas;
- Implementar a separação entre a lógica aritmética e de controle, garantindo clareza e modularidade no design;
- Proporcionar uma base para futuras expansões, considerando aspectos de escalabilidade e reuso de módulos.

2 Modelagem do Sistema

Nesta seção, são apresentados os principais componentes do sistema, os quais se articulam para o funcionamento completo da máquina de vendas.

2.1 Periféricos (Entradas e Saídas)

O projeto considera a interação com diversos periféricos, fundamentais para a operação correta do sistema. As entradas e saídas definidas são:

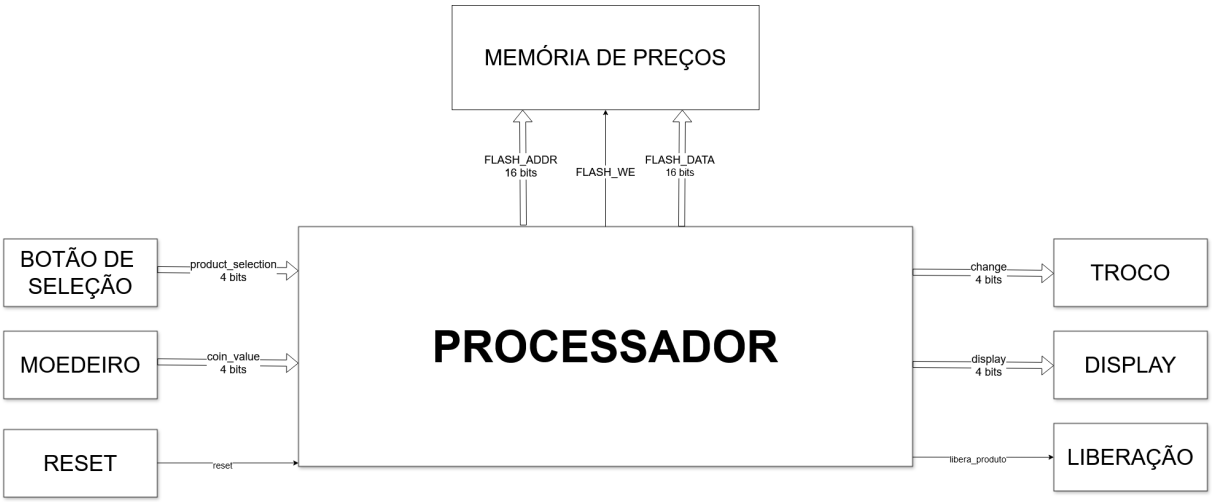


Figura 2: Diagrama de Blocos do Sistema

Entradas:

- `coin_value[3:0]`: Representa o valor da moeda inserida, codificado em 4 bits.
- `product_selection[3:0]`: Código do produto selecionado pelo usuário.
- `reset`: Sinal de reinicialização do sistema, utilizado para restaurar o estado inicial.

Saídas:

- `change[3:0]`: Indica o valor do troco a ser devolvido, representado em 4 bits.
- `display[3:0]`: Sinal destinado à comunicação de mensagens ou valores por meio de um display.
- `liberar_produto`: Sinal que ativa o mecanismo de entrega do produto.
- `liberar_troco`: Sinal responsável por iniciar a devolução do troco.

Tabela 1: Sinais e Registradores do Sistema

Nome	Bits	Tipo	Descrição resumida
`coin_value`	4	Entrada	Valor da moeda inserida.
`product_selection`	4	Entrada	Código do produto escolhido.
`reset`	1	Entrada	Reinicializa a FSM.
`FLASH_ADDR`	16	Saída	Endereço para a memória de preços.
`FLASH_DATA`	16	Entrada	Dados lidos da memória de preços.
`FLASH_WE`	1	Saída	Controle de escrita na memória.
`display`	4	Saída	Mensagem/estado no display.
`change`	4	Saída	Valor do troco a liberar.
`libera_produto`	1	Saída	Comanda a liberação do refrigerante.
`total_inserido`	8	Reg.	Acumula valor total inserido.
`troco`	8	Reg.	Armazena o valor de troco calculado.
`preco_produto`	8	Reg.	Armazena o preço do produto (LUT).

2.2 Arquitetura do Processador: FSM e Datapath

O núcleo do sistema é composto por dois módulos principais:

FSM (Finite State Machine): A FSM é responsável por gerenciar a sequência de operações, definindo os estados de operação, tais como:

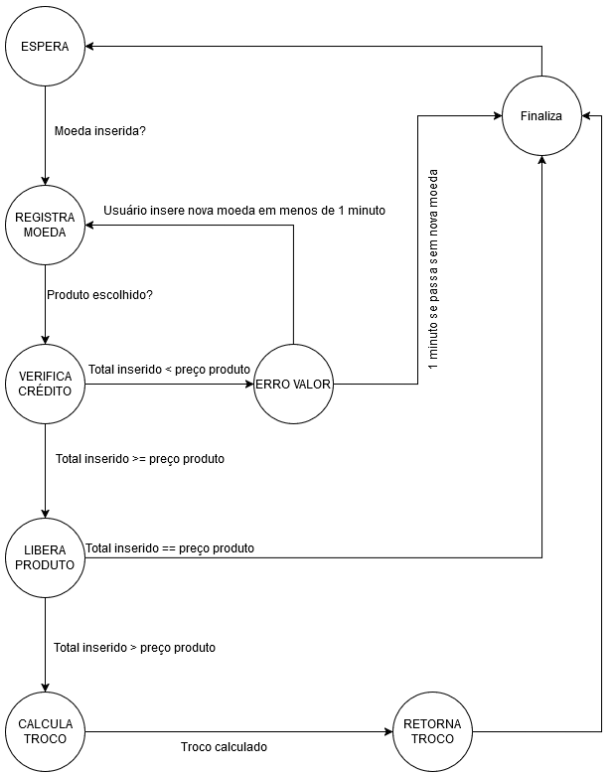


Figura 3: Diagrama de Estado de Alto Nível

- **ESPERA**: Estado inicial, no qual nenhuma operação aritmética é realizada. A FSM não ativa os sinais de `load_acumulador` nem do comparador. O registrador `total_inserido` permanece inalterado, salvo se `reset` for acionado, zerando-o. Se `coin_value` $\neq 0$, a FSM avança para REGISTRA MOEDA.
- **REGISTRA MOEDA**: O somador é ativado para adicionar `coin_value` ao `total_inserido`. O sinal de controle `load_acumulador` $= 1$ carrega o resultado da soma no registrador `total_inserido`. Após essa operação, a FSM segue para VERIFICA CRÉDITO.

- **VERIFICA CRÉDITO**: O comparador (`enable_comparador`) é ativado para comparar se `total_inserido` $\geq$ `preco_produto`. Se `total_inserido` $<$ `preco_produto`, a FSM vai para ERRO VALOR. Caso contrário, avança para LIBERA PRODUTO.

- **ERRO VALOR**: Não há operação aritmética. O `display` pode indicar "Falta X reais" dependendo da lógica de `select_display[1:0]`. O sistema aguarda nova moeda ou timeout. Se uma nova moeda for inserida, retorna para REGISTRA MOEDA. Se houver um limite de tempo ou `reset`, pode ir para FINALIZA.

- **LIBERA PRODUTO**: Como `total_inserido` $\geq$ `preco_produto` já foi verificado, o sinal `liberar_produto` = 1 sinaliza a liberação do refrigerante. Se `total_inserido` = `preco_produto`, a FSM segue para FINALIZA. Caso contrário, avança para CALCULA TROCO.

- **CALCULA TROCO**: O subtrator é ativado para calcular o troco: `troco` $\leftarrow$ `total_inserido - preco_produto`. O sinal `load_troco` = 1 carrega o valor no registrador `troco`. Após essa operação, a FSM segue para RETORNA TROCO.

- **RETORNA TROCO**: O sinal `liberar_troco` = 1 (ou `change[3:0]` recebe parte do valor de `troco`) indica a devolução das moedas. Normalmente, `troco` permanece até que a devolução seja finalizada. Após a devolução, a FSM segue para FINALIZA.

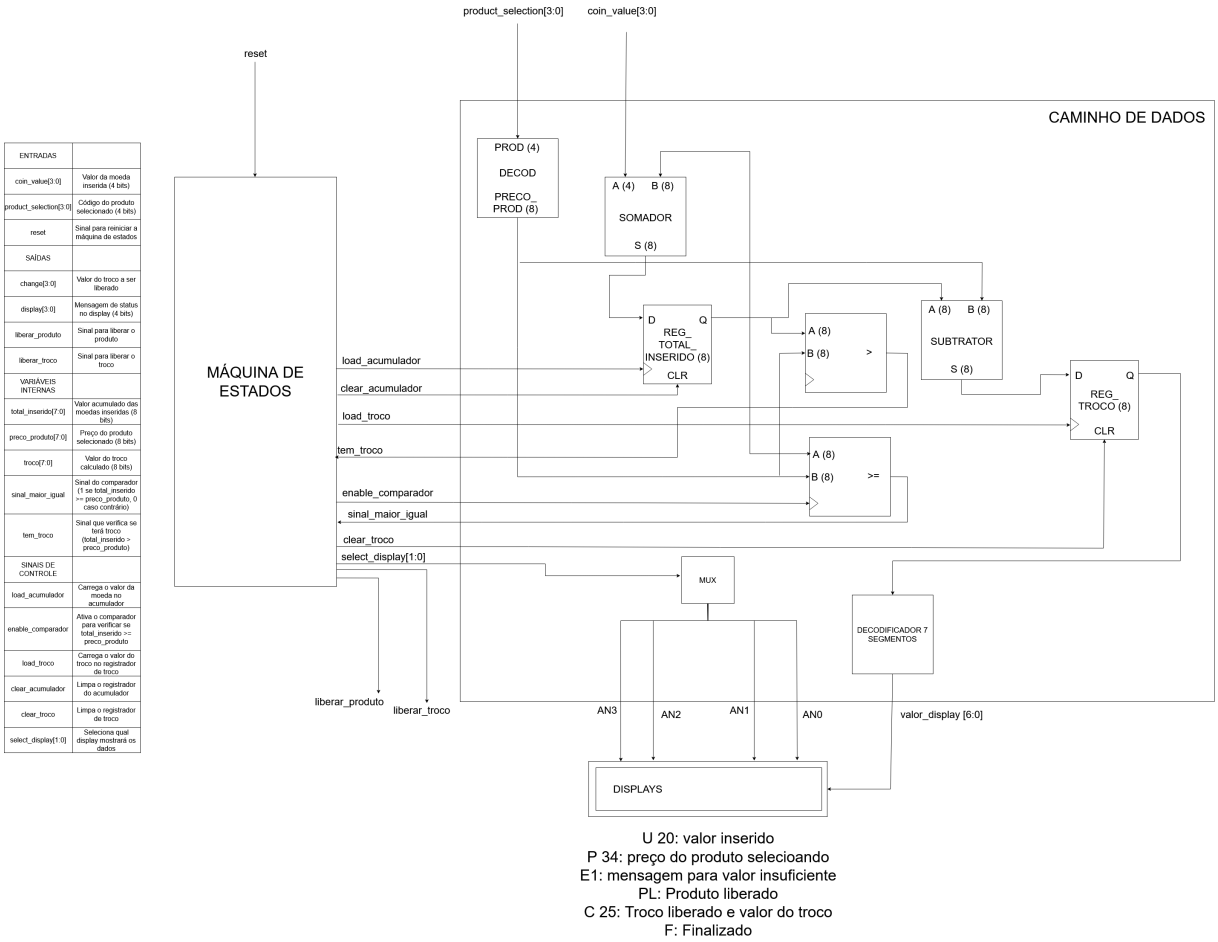
- **FINALIZA**: Os sinais `clear_acumulador` e `clear_troco` podem ser ativados, zerando `total_inserido` e `troco`. O `display` pode exibir uma mensagem de conclusão ou ser zerado. Após isso, retorna ao estado ESPERA, reiniciando o ciclo da FSM.

Datapath: O caminho de dados implementa as operações aritméticas e lógicas necessárias para o funcionamento do sistema. Entre seus componentes, destacam-se:

- **Registradores**: Armazenam informações como o `total_inserido`, o `troco` e o `preco_produto`.

- **Unidades Aritméticas**: Somador, subtrator e comparador, que processam os valores monetários conforme a operação requerida.

A integração entre a FSM e o Datapath é realizada por meio de sinais de controle, que determinam quando as operações devem ser iniciadas ou interrompidas, garantindo a sincronização e correta execução das funcionalidades do sistema.



ENTRADAS	
coin_value[3:0]	Valor da moeda inserida (4 bits)
product_selection[3:0]	Código do produto selecionado (4 bits)
reset	Sinal para reiniciar a máquina de estados
SAÍDAS	
change[3:0]	Valor do troco a ser liberado
display[3:0]	Mensagem de status no display (4 bits)
liberar_produto	Sinal para liberar o produto
liberar_troco	Sinal para liberar o troco
VARIÁVEIS INTERNAS	
total_inserido[7:0]	Valor acumulado das moedas inseridas (8 bits)
preco_produto[7:0]	Preço do produto selecionado (8 bits)
troco[7:0]	Valor do troco calculado (8 bits)
sinal_maior_igual	Sinal do comparador (1 se total_insendo >= preco_produto, 0 caso contrário)
tem_troco	Sinal que verifica se terá troco (total_insendo > preco_produto)
SINAIS DE CONTROLE	
load_acumulador	Carrega o valor da moeda no acumulador
enable_comparador	Ativa o comparador para verificar se total_inserido >= preco_produto
load_troco	Carrega o valor do troco no registrador de troco
clear_acumulador	Limpa o registrador do acumulador
clear_troco	Limpa o registrador do troco
select_display[1:0]	Seleciona qual display mostrará os dados

Figura 4: DataPath

3 Projeto do Bloco de Controle

O bloco de controle é implementado através de uma máquina de estados finitos (FSM) de baixo nível, cuja função é gerar os sinais de controle que coordenam as operações do Datapath. Entre os sinais gerados, destacam-se:

- `load_acumulador`: Sinal que habilita a acumulação do valor da moeda.
- `enable_comparador`: Ativa o comparador para verificar se o crédito inserido é suficiente.
- Outros sinais específicos para cada operação, como a liberação do produto e a devolução do troco.

3.1 Detalhamento da FSM

A seguir, apresenta-se um breve detalhamento dos estados que compõem a FSM:

- **S0 (ESPERA)**: Estado inicial em que o sistema aguarda a inserção de uma moeda ou a ativação do sinal `reset`.

- **S1 (REGISTRA MOEDA)**: Ao detectar uma moeda, o sistema ativa o sinal `load_acumulador` para somar o valor recebido ao `total_inserido`.
- **S2 (VERIFICA CRÉDITO)**: Com o valor acumulado, o sistema utiliza o comparador para verificar se o crédito é suficiente para a compra do produto selecionado.
- **S3 (LIBERA PRODUTO/TROCO)**: Dependendo do resultado da verificação, o sistema libera o produto ou calcula e devolve o troco, acionando os sinais `liberar_produto` e `liberar_troco`, respectivamente.

Esta estrutura possibilita um fluxo de operação claro e bem definido, facilitando tanto a implementação em hardware quanto a análise do comportamento do sistema.

4 Considerações Finais

O presente trabalho descreveu de forma detalhada o projeto de uma máquina de vendas em nível RTL, enfatizando a importância da separação entre a lógica aritmética (Datapath) e a lógica de controle (FSM). A abordagem modular adotada permite:

- **Facilidade de Manutenção:** A separação clara entre os módulos torna o diagnóstico e a correção de eventuais problemas mais diretos.
- **Escalabilidade:** A arquitetura possibilita a incorporação de novas funcionalidades ou a modificação dos parâmetros existentes (como valores de moedas e preços de produtos) sem a necessidade de uma reformulação completa do sistema.
- **Reutilização:** Componentes padronizados podem ser reutilizados em outros projetos, promovendo a eficiência no desenvolvimento de novos sistemas.

A implementação deste projeto contribui para a formação de uma base sólida em design digital e sistemas embarcados, áreas de crescente importância na engenharia moderna.