

Projeto Grupo: Máquina de Vendas
Grupo 2

Arthur Henrique Liberato Lima
Levi Mariano Batista
William Dhomini Miranda Martins

Julho 2025

Sumário

1 - Funcionamento geral do sistema

2 - Especificação do projeto

3 - FSM

4 - Lista de componentes

5 - Caminho de Dados

1 - Funcionamento geral do sistema

A máquina de vendas opera em um ciclo contínuo. O cliente seleciona o produto desejado, e a máquina exibe o preço e a disponibilidade em estoque para verificação. Em seguida, ao inserir uma moeda, o sistema detecta qual o valor da moeda e o acumula

A transação só é concluída se as condições necessárias forem atendidas. Se o valor inserido for igual ou superior ao preço e o produto estiver disponível, a máquina libera o item para o cliente. Caso contrário, se o crédito for insuficiente ou o produto estiver fora de estoque, a compra não é efetuada.

Além do modo de operação padrão, o sistema possui o modo de reposição, em que um operador pode reabastecer o estoque de produtos. O sistema também inclui funções de reset e escape, que permitem ao operador gerenciar e reiniciar o fluxo das operações conforme necessário.

2 - Especificação do projeto

Entradas

Nome	Tipo	Tamanho / Faixa	Descrição
clk	std_logic	1 bit	Clock de 1 Hz
reset	std_logic	1 bit	Reset síncrono
COMPRA	std_logic	1 bit	Inicia processo de compra
REP	std_logic	1 bit	Inicia/confirma reposição
SELECT_C	std_logic	1 bit	Confirma seleção de produto (cliente)
PAG	std_logic	1 bit	Confirma pagamento
ESC	std_logic	1 bit	Cancela compra
ESQ	std_logic	1 bit	Sai do processo atual
product_select_buy	std_logic_vector(1 downto 0)	2 bits	Seleção de produto para compra

product_select_replenish	std_logic_vector(1 downto 0)	2 bits	Seleção de produto para reposição
money_in	integer range 0 to 255	8 bits (equiv.)	Valor inserido
replenish_quantity	integer range 0 to 255	8 bits (equiv.)	Quantidade a ser adicionada ao estoque

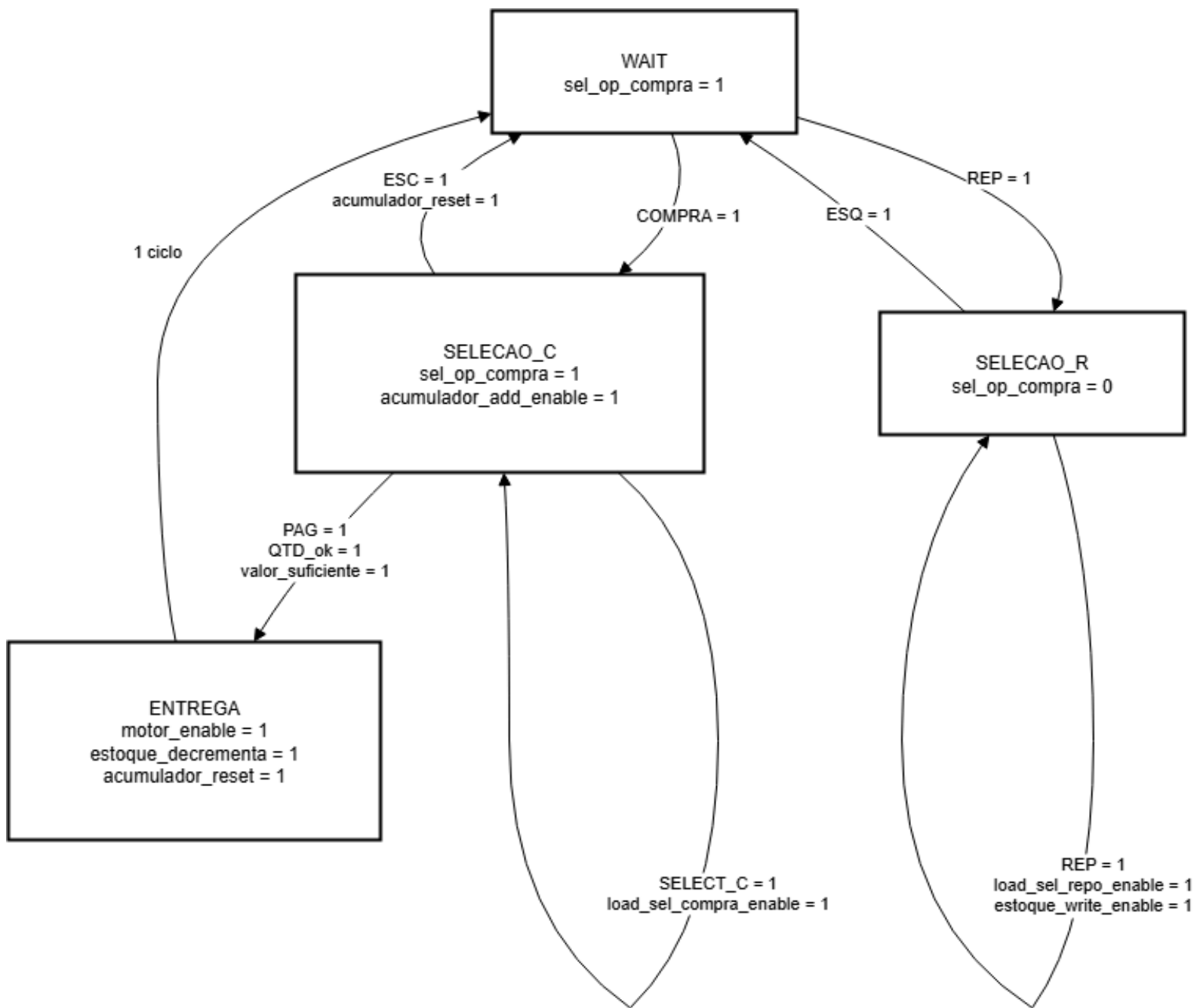
Saídas

Nome	Tipo	Tamanho / Faixa	Descrição
motor_enable	std_logic	1 bit	Ativa motor para entrega
price_display	integer range 0 to 255	8 bits (equiv.)	Mostra o preço do produto
quantity_display	integer range 0 to 255	8 bits (equiv.)	Mostra a quantidade em estoque

Registradores

Nome	Tipo	Tamanho Faixa	Descrição
current_state	state_type (enum)	variável	Estado atual da FSM principal
next_state	state_type	variável	Próximo estado da FSM
selected_product	std_logic_vector(1 downto 0)	2 bits	Produto selecionado
money_received	integer range 0 to 255	8 bits	Valor de pagamento acumulado
product_quantity	integer	8 bits (por produto)	Estoque atual de cada produto
product_price	integer	8 bits (por produto)	Preço de cada produto

3 - FSM



4 - Lista de componentes

Nome do Componente	Entradas	Saídas	Tipos	Tamanhos	Função
moedeiro	habilita_moedeiro, sinal_moeda_25c, sinal_moeda_50c, sinal_moeda_1real	valor_moeda	std_logic, integer	1 bit (sinais), 8 bits (inteiro 0–255)	Detecta moeda inserida e gera valor correspondente
reg_2b	clk, rst, enable, in_D	out_Q	std_logic, std_logic_vector	1 bit (sinais), 2 bits (vetor)	Armazena o código do produto selecionado
memoria_ precos	endereco	dado_out	std_logic_vector, integer	2 bits (endereço), 8 bits (preço)	Retorna o preço do produto com base no código
memoria_ estoque	clk, rst, wr, decrementa, endereco, dado_in	dado_out	std_logic, std_logic_vector, integer	1 bit (sinais), 2 bits (endereço), 8 bits (dados)	Lê e atualiza a quantidade de produtos no estoque
somador	in_A, in_B	out_S	integer	8 bits (0–255)	Soma o valor inserido ao saldo acumulado
comparador	in_A, in_B	out_S	integer, std_logic	8 bits (valores), 1 bit (saída lógica)	Compara saldo com o preço para habilitar compra
mux_2x1_2b	sel, in_A, in_B	out_S	std_logic, std_logic_vector	1 bit (seletor), 2 bits (vetores)	Seleciona entre dois códigos de produto diferentes

subtrator	in_A, in_B	out_S	integer	8 bits (0–255)	Calcula novo valor de estoque após venda (estoque - 1)
-----------	------------	-------	---------	-------------------	--

5 - Caminho de Dados

