

Projeto de Sistemas Digitais

Etapa 2: Especificação da HLFSM e Datapath

Grupo 2

Junho de 2025

Sumário

1	Especificação da Máquina de Estados de Alto Nível (HLFSM)	1
1.1	Descrição Detalhada dos Estados e Transições	1
1.2	Representação Gráfica da HLFSM	1
2	Especificação dos Componentes do Caminho de Dados (Datapath)	2

Lista de Figuras

1	Diagrama de estados completo da HLFSM, ilustrando os estados <i>Wait</i> , <i>Selecao_C</i> , <i>Selecao_R</i> e <i>Entrega</i> , junto com as condições e ações de transição.	2
2	Detalhamento das ações de Moore (ações executadas dentro dos estados) para os estados de <i>Selecao_C</i> e <i>Entrega</i>	2

Lista de Tabelas

1	Lista de Componentes do Caminho de Dados	3
---	--	---

1 Especificação da Máquina de Estados de Alto Nível (HLFSM)

A Máquina de Estados Finitos de Alto Nível (HLFSM) descreve o comportamento do sistema, incluindo as condições para transições de estado e as ações executadas. A especificação baseia-se no diagrama de estados previamente definido.

1.1 Descrição Detalhada dos Estados e Transições

- **Estado Wait:** Estado inicial de espera por interação.
 - **Transição para Selecao_C:** Ocorre quando a entrada **COMPRA** é ativada ('1').
 - **Transição para Selecao_R:** Ocorre quando a entrada **REP** é ativada ('1').
- **Estado Selecao_C (Compra):** Gerencia a seleção do produto e o pagamento.
 - **Ações internas:** Mostra preço e quantidade do item selecionado e acumula o dinheiro inserido (**money_in**).
 - **Transição para Entrega:** Condicionada ao acionamento de **PAG** e à validação de que o produto está disponível (**QTD='1'**) e o pagamento é suficiente.
 - **Transição para Wait:** Acionada por **ESC** para cancelar a operação. A ação associada é zerar o valor acumulado.
- **Estado Selecao_R (Reposição):** Permite ao operador reabastecer o estoque.
 - **Ação de transição:** A confirmação com **REP** atualiza o estoque com o valor de **replenish_quantity**.
 - **Transição para Wait:** Acionada por **ESQ** para sair do modo de reposição.
- **Estado Entrega:** Controla a liberação física do produto.
 - **Ações de transição:** Ativa o sinal **motor_enable** por um ciclo de clock, decrementa o estoque do produto e zera o valor acumulado.
 - **Transição para Wait:** Ocorre de forma incondicional após a conclusão da entrega para preparar o sistema para uma nova venda.

1.2 Representação Gráfica da HLFSM

As figuras a seguir ilustram o diagrama da HLFSM e detalham as ações internas aos estados.

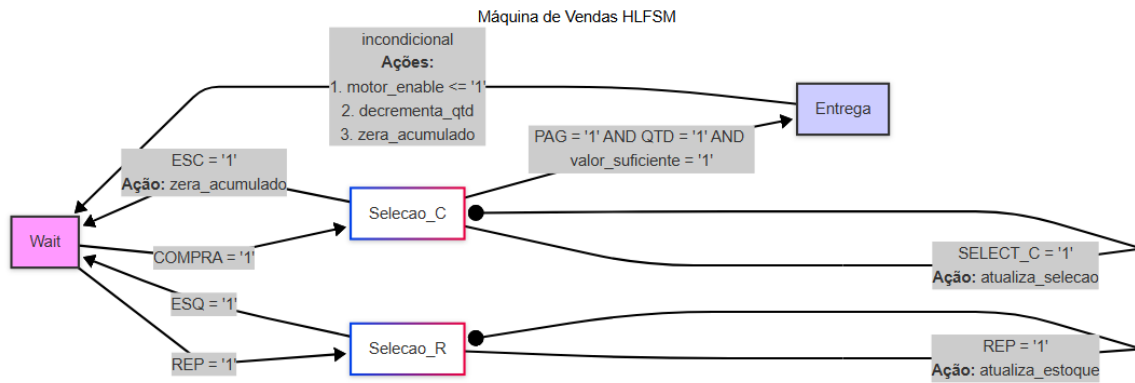


Figura 1: Diagrama de estados completo da HLFSM, ilustrando os estados *Wait*, *Selecao_C*, *Selecao_R* e *Entrega*, junto com as condições e ações de transição.

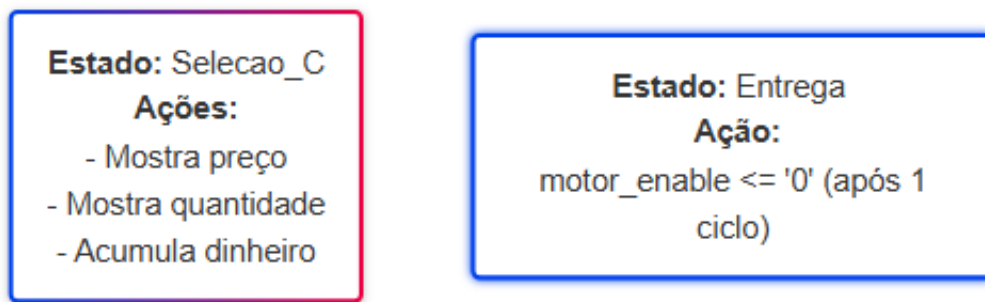


Figura 2: Detalhamento das ações de Moore (ações executadas dentro dos estados) para os estados de *Selecao_C* e *Entrega*.

2 Especificação dos Componentes do Caminho de Dados (Datapath)

O caminho de dados (datapath) consiste nos blocos funcionais que executam as operações de armazenamento e manipulação de dados, controlados pela FSM. Os componentes são definidos com base nos registradores internos especificados.

Tabela 1: Lista de Componentes do Caminho de Dados

Componente	Descrição Funcional	Entradas (Nome, Tipo, Tamanho)	Saídas (Nome, Tipo, Tamanho)
Preco_ROM	Memória ROM para armazenar os preços dos 4 produtos.	‘endereco’ (std_logic_vector, 2 bits)	‘preco_out’ (integer, 8 bits)
Estoque_RAM	Memória RAM para armazenar e atualizar a quantidade de cada produto.	‘clk’ (std_logic, 1 bit) ‘write_enable’ (std_logic, 1 bit) ‘decrementa’ (std_logic, 1 bit) ‘endereco’ (std_logic_vector, 2 bits) ‘data_in’ (integer, 8 bits)	‘data_out’ (integer, 8 bits)
Acumulador_Valor	Registrador que soma e armazena o valor monetário inserido pelo cliente.	‘clk’ (std_logic, 1 bit) ‘reset’ (std_logic, 1 bit) ‘add_enable’ (std_logic, 1 bit) ‘valor_in’ (integer, 8 bits)	‘valor_total’ (integer, 8 bits)
Reg_Selecao_Compra	Registrador para o índice do produto selecionado pelo cliente.	‘clk’ (std_logic, 1 bit) ‘load_enable’ (std_logic, 1 bit) ‘sel_in’ (std_logic_vector, 2 bits)	‘sel_out’ (std_logic_vector, 2 bits)
Comparador_Pagamento	Lógica combinacional que compara o valor acumulado com o preço do produto.	‘valor_acumulado’ (integer, 8 bits) ‘preco_produto’ (integer, 8 bits)	‘valor_suficiente’ (std_logic, 1 bit)