

Relatório da Atividade 3.2 - Algoritmos Sequenciais

Universidade Federal do Rio Grande do Norte
Departamento de Informática e Matemática Aplicada

Janeiro de 2025

Conteúdo

1	Introdução	2
2	Especificação do Problema	2
3	Projeto e Modelagem	2
4	Introdução	2
5	Descrição do Diagrama de Blocos	2
5.1	Componentes Principais	2
5.2	Entradas e Saídas	3
5.2.1	Banco de Registradores	3
5.2.2	FSM (Máquina de Estados Finitos)	3
5.3	Detalhamento dos Estados da FSM	4
5.4	Diagrama de Blocos Completo	4
5.5	FSM (Máquina de Estados Finitos)	5
6	Simulação	5
6.1	Plano de Simulação	5
6.2	Resultados das Simulações	6

1 Introdução

Este relatório apresenta o desenvolvimento da Atividade 3.1 do componente curricular DIM0128 - Circuitos Lógicos, referente ao projeto, implementação, simulação e prototipagem de circuitos sequenciais que executam algoritmos específicos em VHDL, utilizando a placa DE2-115. O objetivo é implementar um banco de registradores de 8 registradores de 16 bits cada.

2 Especificação do Problema

O banco de registradores deve permitir duas operações:

1. Leitura:

- Precisa do endereço do registrador que deverá ser lido.
- Retorna o valor do dado armazenado nesse registrador.

2. Escrita:

- Precisa do endereço do registrador no qual será feita a escrita.
- Precisa do dado que será escrito nesse registrador.

3 Projeto e Modelagem

4 Introdução

O objetivo deste documento é especificar o diagrama de blocos do banco de registradores controlado por uma máquina de estados finitos (FSM), com base no código VHDL fornecido. O diagrama de blocos descreve as interações entre as entradas e saídas do banco de registradores e da FSM.

5 Descrição do Diagrama de Blocos

5.1 Componentes Principais

O sistema consiste de dois blocos principais:

- **Banco de Registradores:** Composto por um vetor de registradores e controlado pela FSM.
- **FSM (Máquina de Estados Finitos):** Controla as operações de leitura e escrita no banco de registradores.

5.2 Entradas e Saídas

5.2.1 Banco de Registradores

Entradas:

- **clock:** Sinal de relógio, utilizado para sincronizar o processo de leitura e escrita.
- **reset:** Sinal de reset, utilizado para limpar os registradores.
- **reg_addr:** Endereço do registrador selecionado (controlado pela FSM).
- **startReading:** Sinal de controle que indica que a operação de leitura deve ser realizada.
- **startWriting:** Sinal de controle que indica que a operação de escrita deve ser realizada.
- **data_in:** Dados que serão gravados em um registrador específico.

Saídas:

- **data_out:** Dados lidos do registrador selecionado. Se estiver em operação de leitura, o valor desse sinal é o conteúdo do registrador indicado por **reg_addr**.
- **current_state_out:** Estado atual da FSM, com 2 bits que indicam o estado da FSM. Os valores possíveis são:
 - 00: **pickingReg**
 - 01: **writing**
 - 10: **reading**

5.2.2 FSM (Máquina de Estados Finitos)

Entradas:

- **clock:** Usado para realizar transições de estado sincronizadas com o relógio.
- **reset:** Sinal de reset, utilizado para reiniciar a FSM.
- **startReading:** Indica que a operação de leitura foi iniciada.
- **startWriting:** Indica que a operação de escrita foi iniciada.
- **reg_addr:** O endereço do registrador selecionado.
- **data_in:** Dados a serem gravados no banco de registradores, caso a operação seja de escrita.

Saídas:

- **state:** Controla a FSM com 3 estados principais:
 - **pickingReg:** Estado inicial, onde o banco de registradores seleciona o registrador.
 - **writing:** Estado de escrita, onde os dados de **data_in** são gravados no registrador selecionado.
 - **reading:** Estado de leitura, onde os dados de um registrador são enviados para **data_out**.

5.3 Detalhamento dos Estados da FSM

- **pickingReg**: Estado inicial, onde a FSM seleciona o registrador a ser lido ou escrito.
- **writing**: Estado onde os dados são gravados no registrador selecionado.
- **reading**: Estado onde os dados são lidos do registrador selecionado e enviados para `data_out`.

5.4 Diagrama de Blocos Completo

O diagrama de blocos completo inclui as interações entre as entradas e saídas, com a FSM controlando o fluxo de dados entre o banco de registradores e as operações de leitura e escrita. A FSM depende das entradas `startReading` e `startWriting` para determinar o estado atual e, conseqüentemente, controlar as operações do banco de registradores.

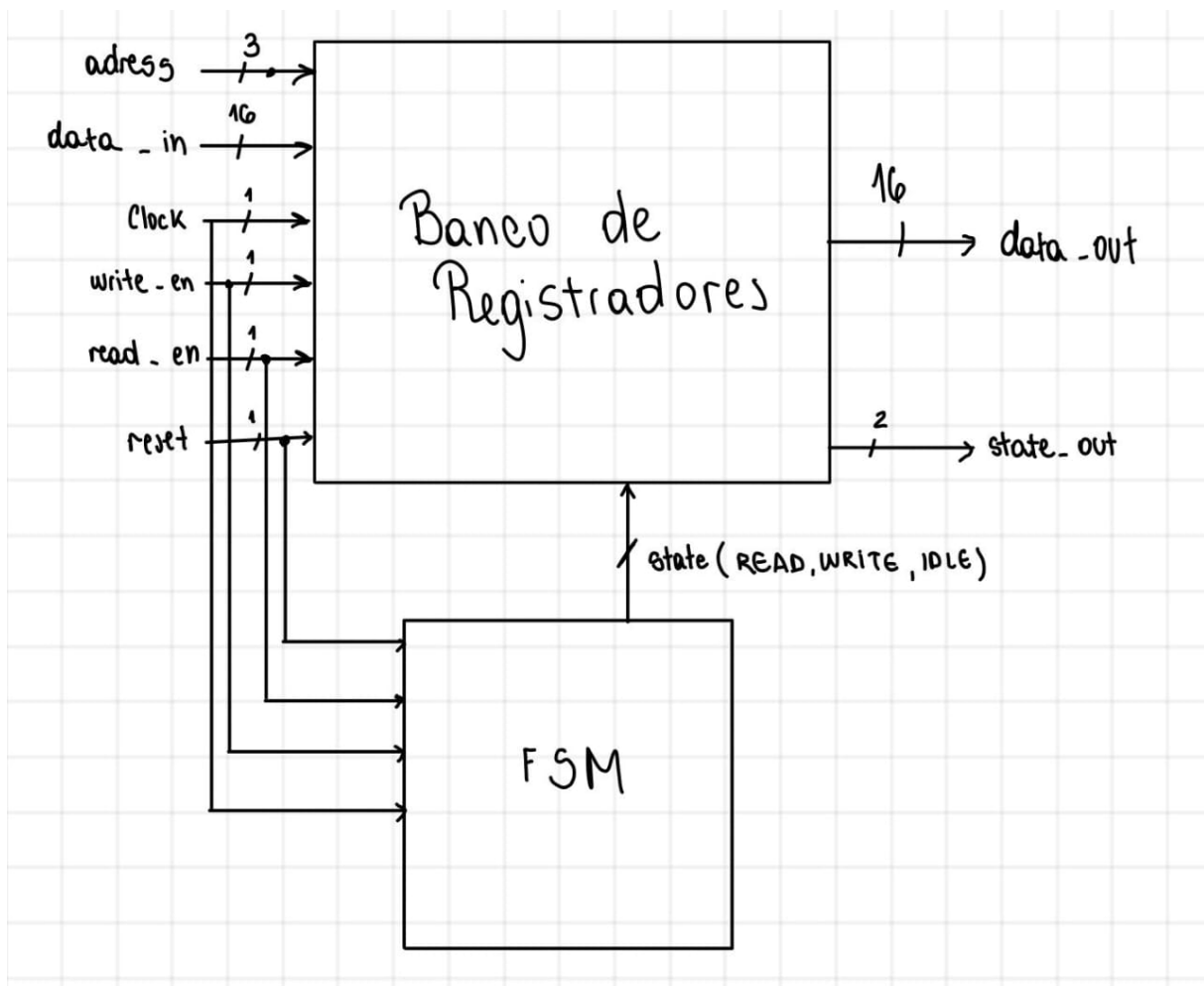
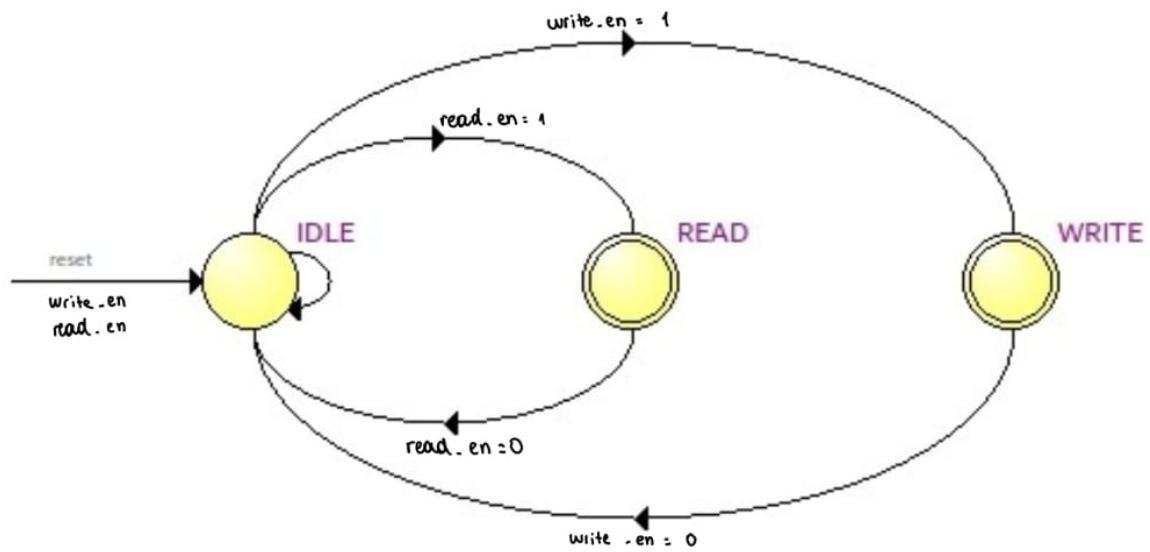


Figura 1: Diagrama de blocos do sistema.

5.5 FSM (Máquina de Estados Finitos)



Estado Atual	Reset	WE	RE	Próximo Estado
IDLE	1	X	X	IDLE
IDLE	0	1	0	WRITE
IDLE	0	0	1	READ
IDLE	0	0	0	IDLE
READ	1	X	X	IDLE
READ	0	1	0	WRITE
READ	0	0	1	READ
READ	0	0	0	READ
WRITE	1	X	X	IDLE
WRITE	0	1	0	WRITE
WRITE	0	0	1	READ
WRITE	0	0	0	WRITE

Figura 2: Diagrama FSM.

6 Simulação

6.1 Plano de Simulação

A simulação realizada cobre os seguintes cenários:

- Leitura do registrador.

- Escrita do registrador.
- Mudança de estados da FSM.
- Ciclos do clock.
- Sobrescrita do registrador.

6.2 Resultados das Simulações

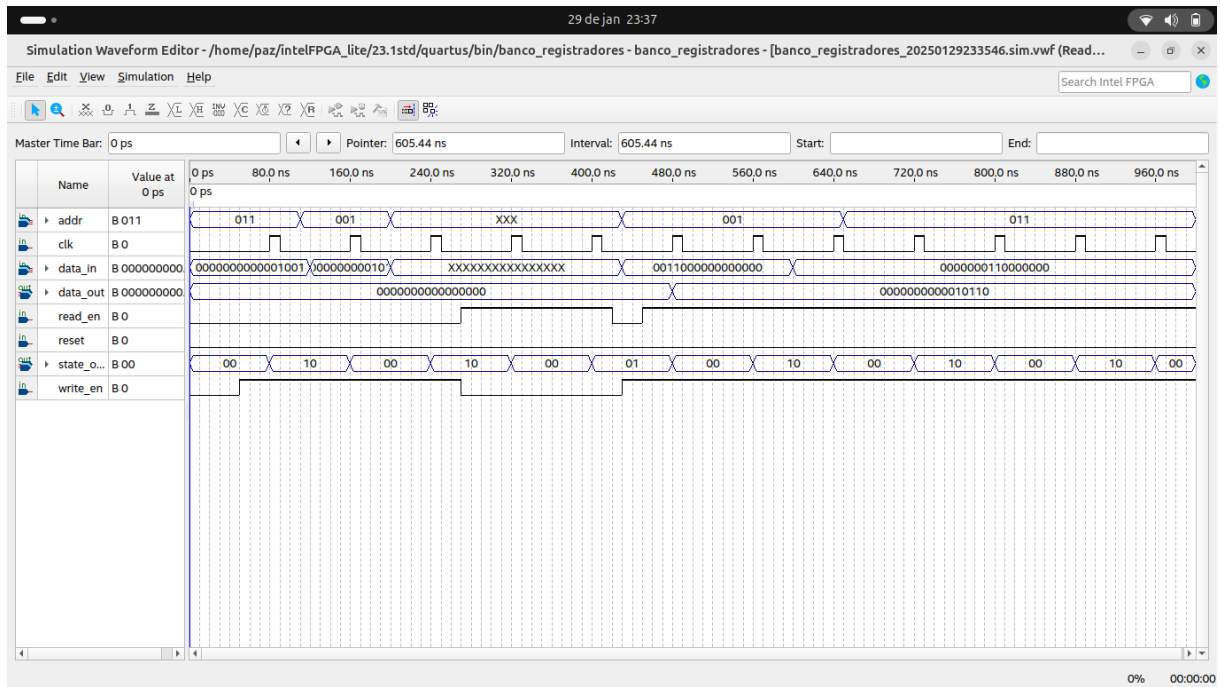


Figura 3: Simulação do banco de registradores com pulso de clock.

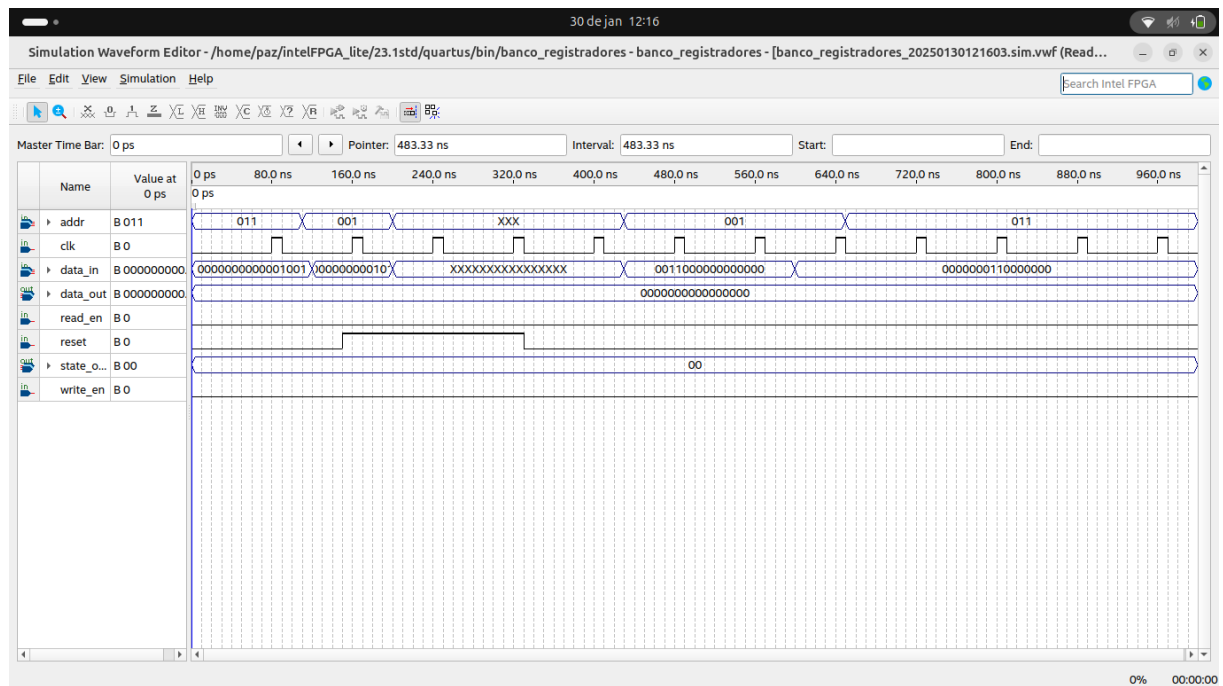


Figura 4: Simulação do banco de registradores com reset.