



**UNIVERSIDADE FEDERAL DA BAHIA
ESCOLA POLITÉCNICA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO
ENGG52 - LABORATÓRIO INTEGRADO I-A**

RENILTON RIBEIRO ALMEIDA

AVALIAÇÃO SUBSTITUTIVA

Salvador - BA
2025

RENILTON RIBEIRO ALMEIDA

AVALIAÇÃO SUBSTITUTIVA

Trabalho apresentado como requisito para obtenção de nota na disciplina Laboratório Integrado I-A, no curso de Engenharia da Computação, na Universidade Federal da Bahia.

Orientador: Wagner Oliveira

Salvador - BA
2025

SUMÁRIO

| | |
|---|-----------|
| INTRODUÇÃO..... | 4 |
| DESCRIÇÃO DO PROJETO E DOS RESULTADOS OBTIDOS..... | 5 |
| Diagrama de Blocos..... | 5 |
| Diagrama de Estados..... | 6 |
| Estrutura do Projeto em Verilog..... | 7 |
| Simulação Funcional no ModelSim..... | 8 |
| Síntese no Quartus Prime..... | 9 |
| Resultados e Validação..... | 10 |
| CONCLUSÃO..... | 11 |
| REFERÊNCIAS BIBLIOGRÁFICAS..... | 12 |

INTRODUÇÃO

O presente trabalho tem como objetivo a implementação de uma máquina de estados finitos (FSM – Finite State Machine) utilizando ferramentas de desenvolvimento digital como o ModelSim e o Quartus Prime. A FSM projetada tem como função principal controlar a leitura de blocos de dados provenientes da memória, acumular os resultados parciais, armazenar as somas e sinalizar a conclusão do processo.

Durante o desenvolvimento do projeto, foram elaborados diagramas de estados e de blocos seguindo convenções formais, além da escrita do código em Verilog para descrever o comportamento sequencial da FSM. A simulação funcional foi realizada no ModelSim, validando o comportamento do sistema por meio da visualização de formas de onda. Em seguida, foi feita a síntese do projeto no Quartus, gerando o circuito correspondente ao código HDL.

O projeto proporciona uma aplicação prática dos conceitos de sistemas digitais sequenciais, FSMs, simulação de hardware e síntese lógica, reforçando a importância das ferramentas CAD no processo de desenvolvimento de sistemas embarcados.

DESCRIÇÃO DO PROJETO E DOS RESULTADOS OBTIDOS

O projeto desenvolvido consistiu na modelagem, implementação, simulação e síntese de um sistema digital sequencial baseado em uma máquina de estados finitos (FSM). A FSM teve como objetivo percorrer blocos de dados armazenados em uma memória externa, somar seus valores parciais com auxílio de um acumulador e, ao final do processo, gravar a soma total em um endereço específico. O desenvolvimento envolveu as etapas de codificação em Verilog, simulação funcional no ModelSim e síntese lógica no Quartus Prime.

Diagrama de Blocos

Para representar a estrutura geral do sistema, foi elaborado um diagrama de blocos contendo três módulos principais:

FSM (Finite State Machine) – Responsável por controlar a lógica sequencial e a coordenação do fluxo de dados, gerando sinais de controle como `acc_enable`, `write_en`, `addr`, `ready`, entre outros.

Acumulador – Recebe os dados da memória e realiza a soma cumulativa de cada bloco. É habilitado pelo sinal `acc_enable`.

Memória – Atua como componente externo ao sistema, onde estão armazenados os dados de entrada e os resultados parciais e finais. É acessada pela FSM para leitura e escrita.

As entradas globais do sistema são `clk` e `reset`, que se conectam à FSM (e desta aos demais módulos). Os sinais de controle e os dados trafegam entre os blocos via interconexões internas. O diagrama representa visualmente a comunicação entre os módulos e sua organização hierárquica.

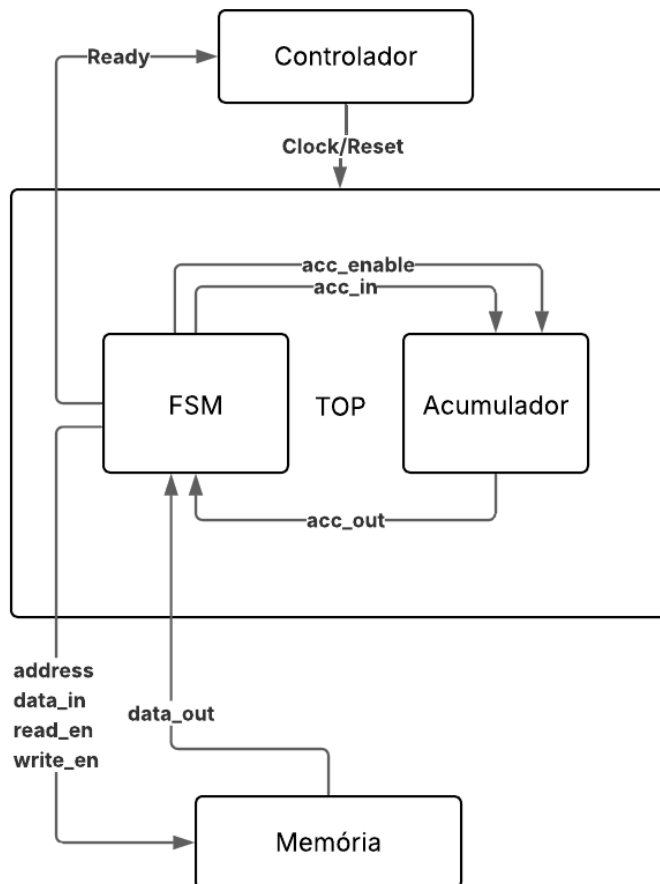


Figura 1 - Diagrama de blocos

Diagrama de Estados

O funcionamento da FSM foi modelado por meio de um diagrama de estados com 12 estados principais, cada um representando uma etapa do processamento:

IDLE – Estado inicial, aguardando o reset.

LOAD – Envia endereço para leitura da memória.

WAIT_READ – Espera 1 ciclo para a resposta da memória.

ACCUMULATE – Habilita o acumulador para somar o dado recebido.

NEXT_ADDR – Verifica se todos os dados do bloco foram lidos.

WRITE_RESULT – Grava o resultado parcial no endereço apropriado.

SUM_RESULTS – Soma o resultado parcial ao acumulado geral.

NEXT_BLOCK – Controla a mudança de bloco; se ainda houver blocos, retorna ao estado LOAD.

FINAL_WRITE – Grava a soma total acumulada no endereço 31.

SIGNAL_READY1 e SIGNAL_READY2 – Sinalizam a conclusão do processo com dois ciclos de ready = 1.

RESET_FSM – Restaura a FSM para o estado inicial.

Cada transição de estado é condicionada por sinais internos como count, bloco, ou tempo de resposta da memória. O diagrama foi elaborado seguindo as convenções de FSM do tipo Moore, com os sinais de controle associados aos estados.

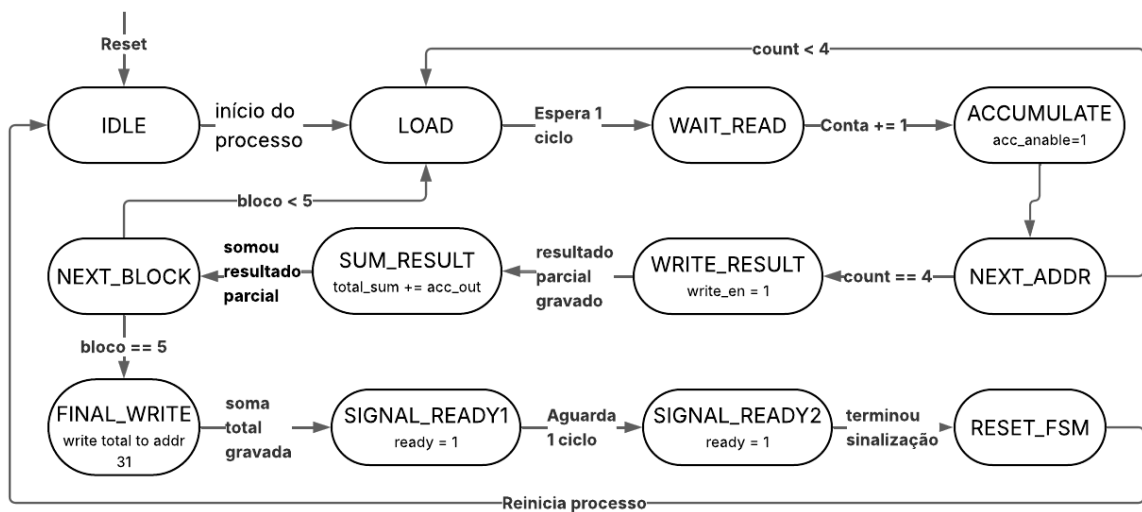


Figura 2 - Diagrama de estados

Estrutura do Projeto em Verilog

O sistema foi implementado modularmente, utilizando quatro arquivos principais em Verilog:

fsm.v

Contém a implementação da máquina de estados finitos responsável por controlar todo o processo. Os estados foram codificados com base em um diagrama de estados previamente definido. A FSM gera os sinais de controle como acc_enable, write_en e ready, além de gerenciar os contadores de dados e blocos.

acumulador.v

Módulo que realiza a soma dos dados recebidos, acumulando valores sucessivos sempre que o sinal acc_enable é ativado. Também possui lógica de reset e de leitura do valor atual do acumulador.

memoria.v

Simula uma memória externa de leitura e escrita. Este módulo foi utilizado principalmente no testbench para verificar a comunicação e os acessos realizados pela FSM.

top.v

Módulo de integração que conecta a FSM, o acumulador e a memória. Atua como ponto de entrada do sistema, instanciando os demais módulos e repassando sinais de entrada como clock e reset.

Simulação Funcional no ModelSim

Para validar o comportamento do sistema, foi desenvolvido um testbench que simula os sinais de entrada e observa as respostas da FSM. A simulação foi realizada no ambiente ModelSim - Intel FPGA Edition.

Principais pontos testados:

A transição correta entre os 12 estados da FSM.

O funcionamento dos sinais de controle (acc_enable, write_en, ready) em seus respectivos ciclos.

A contagem de dados dentro de cada bloco (4 dados por bloco).

A mudança de blocos após o término da contagem.

A gravação da soma total no endereço 31 da memória.

A geração do sinal ready durante dois ciclos consecutivos ao fim da operação.

O comportamento da FSM após o reset.

As formas de onda geradas mostraram o funcionamento correto de todo o processo, com destaque para o controle sincronizado entre FSM, acumulador e memória. Os dados escritos nos endereços esperados da memória validaram a operação de escrita e acumulação.

As imagens das simulações foram salvas e incluídas neste relatório como evidência da correta execução.

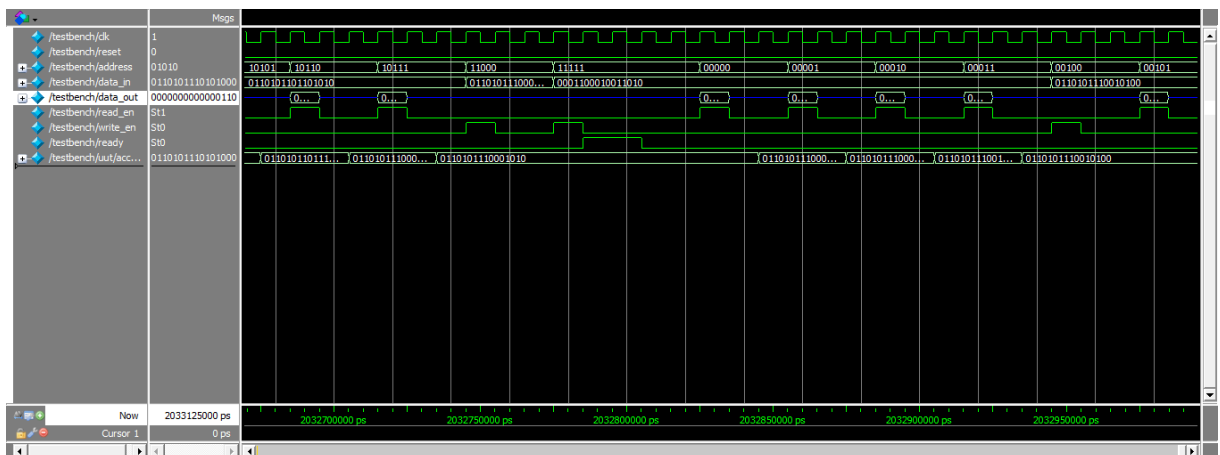


Figura 3 - Forma de onda da simulação

Síntese no Quartus Prime

Após a validação da lógica funcional no ModelSim, o projeto foi sintetizado no Quartus Prime. A etapa de síntese consistiu em:

Criação de um novo projeto no Quartus com o módulo `top.v` como entidade principal.

Adição de todos os arquivos `.v` do projeto.

Seleção do dispositivo FPGA compatível (EP4CE115F29C7 ou equivalente compatível com o ambiente).

Compilação e análise do projeto.

A ferramenta realizou a análise e a síntese lógica, gerando o circuito correspondente em linguagem de máquina. O mapeamento foi completado com sucesso, indicando que o projeto é viável para implementação física em FPGA.

O Technology Map Viewer exibiu os blocos lógicos correspondentes à FSM, acumulador e interconexões, representando a estrutura gerada pela síntese.

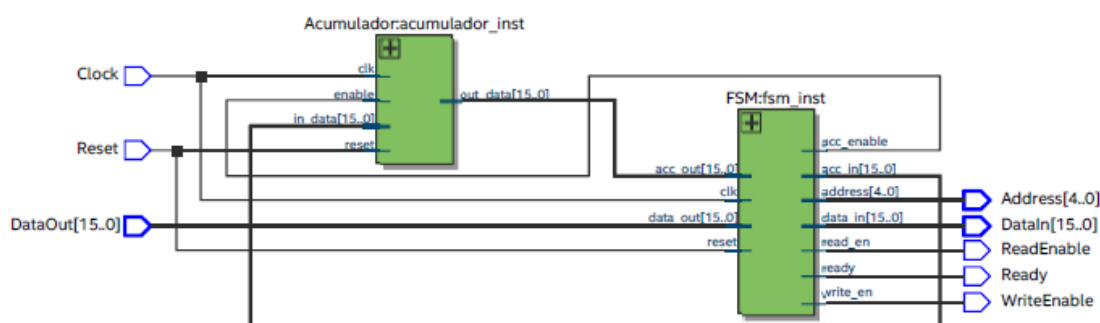


Figura 4 - Visão lógica do circuito sintetizado

Resultados e Validação

A FSM comportou-se conforme especificado, executando todas as etapas da operação com precisão.

O acumulador funcionou corretamente, somando os valores de cada bloco.

A simulação demonstrou que a FSM respeita os tempos de leitura e gravação da memória.

A síntese confirmou que o projeto está apto para implementação física, com todos os sinais e módulos reconhecidos corretamente.

O sinal ready, como saída final do sistema, foi gerado de forma visível e precisa ao fim da execução, sinalizando o término do processo.

CONCLUSÃO

O desenvolvimento da máquina de estados finitos permitiu consolidar conhecimentos fundamentais sobre projeto digital, modelagem de controle sequencial e uso de linguagens de descrição de hardware. Através da FSM implementada, foi possível automatizar o processo de leitura e acumulação de blocos de dados, aplicando técnicas de controle sincronizado via sinais de clock, reset, enable e flags internas.

A simulação no ModelSim garantiu a verificação funcional do projeto, enquanto a síntese no Quartus confirmou sua viabilidade em termos de implementação física em FPGA. Com isso, o trabalho atingiu seus objetivos, mostrando o potencial do uso conjunto de simulação e síntese como ferramentas pedagógicas e profissionais no contexto da engenharia digital.

REFERÊNCIAS BIBLIOGRÁFICAS

1. BROWN, S.; Vranesic, Z. **Fundamentals of Digital Logic with Verilog Design**. 3ª ed. McGraw-Hill, 2013.
2. WAKERLY, J. F. **Digital Design: Principles and Practices**. 4ª ed. Pearson Education, 2006.
3. INTEL. **ModelSim - Intel® FPGA Edition Software User Guide**.* Disponível em: <https://www.intel.com>
4. INTEL. **Quartus® Prime Software User Guide**. Disponível em: <https://www.intel.com>
5. GONÇALVES, A. C. **Sistemas Digitais: Fundamentos e Aplicações**. LTC, 2011.