**REPORT**

**Design and Implementation of a Simple Processor**

Ver 2.0

25/12/2021

|  |
| --- |
| **Abstract** |
| Bản báo cáo này thiết kế bộ vi xử lý trên tập lệnh cho sẵn .Đề xuất thiết kế mức RTL, mô hình hóa bằng VHDL, mô phỏng bằng ModelSim (QuestaSim) và thực thi trên FPGA một bộ vi xử lý 16-bit đơn giản. Ứng dụng bộ xử lý để xây dựng một hệ thống SoC ứng dụng trong điều khiển. |

|  |
| --- |
| **Keywords** |
| FSM, FSMD, datapath, controller, 16-bit microprocessor, VHDL |

Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Time** | **Revised by** | **Description** |
| V1.0 | 15/11/2021 | Nguyễn Kiêm Hùng | Original Version |
| V2.0 | 25/12/2021 | Nguyễn Quang Linh | Final project version |
|  |  |  |  |

Table of Contents

[Document History 2](#_Toc18486)

[Table of Contents 3](#_Toc16814)

[1. Introduction 4](#_Toc27818)

[2. Requirements 4](#_Toc2359)

[3. Architecture Design 6](#_Toc28672)

[3.1 FSMD 6](#_Toc1928)

[3.2 Datapath architecture 8](#_Toc6580)

[3.3 Controller 10](#_Toc25495)

[4. Simple System-on-Chip 12](#_Toc9384)

[5. Modeling 12](#_Toc25973)

[6. Simulation 12](#_Toc13948)

# Introduction

**Objective:** Vận dụng các kiến thức, kỹ năng đã được học đểthiết kế, mô phỏng và thực thi một bộ vi xử lý 16-bit đơn giản.

# Requirements

Bộ vi xử lý hỗ trợ các lệnh cơ bản như được liệt kê trong Bảng 1;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TT** | **Assembly Instruction** | **First Byte** | | **Second Byte** | | **Operation** |
| Opcode | Operand1 | Operand2 | |
| 1 | LD Rn, direct | 0000 | Rn | direct | | Rn = M(direct) |
| 2 | SD direct, Rn | 0001 | Rn | direct | | M(direct) = Rn |
| 3 | LR Rn, @Rm | 0010 | Rn | Rm |  | M(Rm) = Rn |
| 4 | SR Rn, @Rm | 0011 | Rn | Rm |  | Rn = M(Rm) |
| 5 | LI Rn, #immed | 0011 | Rn | immediate | | Rn = immediate |
| 6 | ADD Rn, Rm | 0100 | Rn | Rm |  | Rn = Rn + Rm |
| 7 | SUB Rn, Rm | 0101 | Rn | Rm |  | Rn = Rn - Rm |
| 8 | OR Rn, Rm | 0111 | Rn | Rm |  | Rn = Rn or Rm |
| 9 | AND Rn, Rm | 1000 | Rn | Rm |  | Rn = Rn and Rm |
| 10 | JPZ Rn, Addr | 1001 | Rn | Addr | | PC = Addr only if Rn = 0 |
| 11 | JMP Addr | 1010 | Rn | Adrr | | PC = Addr |
| 12 | HALT | 1011 |  |  | | Do nothing |

Bảng 1. Cấu trúc tập lệnh.

**Cấu trúc của bộ vi** xử lý bao gồm các khối chức năng như trong **Hình 1**, trong đó :

* + **Thanh ghi PC (Program Counter)**: 16-bit, dùng để chứa địa chỉ của lệnh tiếp theo mà bộ vi xử lý sẽ thực hiện.
  + **Thanh ghi IR (Instruction Register)**: 16-bit, dùng để chưa lệnh mà vi xử lý sẽ thực hiện.
  + **Tệp thanh ghi RF (Register File)**: 16×16 bit, dùng lưu dữ liệu trong quá trình tính toán của ALU.
  + **ALU (arithmetic and logic unit)**: hỗ trợ các phép tính trên dữ liệu 16-bit.
  + **Controller**: điều khiển tất cả cá tín hiệu ra vào của hệ thống.

**Bộ nhớ Memory** : 64K×16 bit, dùng để lưu chương trình và dữ liệu cho bộ vi xử lý.



Hình 1: Cấu trúc các khối chức năng cơ bản của bộ vi xử lý.

**Quá trình thực thi lệnh**

Chúng ta có chia quá trình thực hiện các lệnh của vi xử lý thành một số giai đoạn cơ bản như sau:

1. Tìm nạp lệnh (*Fetch Instruction*): có nhiệm vụ đọc lệnh tiếp theo từ bộ nhớ vào thanh ghi lệnh.

2. Giải mã lệnh (*Decode Instruction*): có nhiệm vụ xác định lệnh trong thanh ghi lệnh cần thực hiện thao tác nào (ví dụ: cộng, di chuyển dữ liệu, v.v.).

3. Tìm nạp các toán hạng (*Fetch Operands*): có nhiệm vụ di chuyển dữ liệu toán hạng của lệnh vào các thanh ghi thích hợp.

4. Thực thi thao tác (*Execute operation*): nhiệm vụ thực hiện thao tác trên các thanh ghi thích hợp bằng ALU và ghi kết quả vào một thanh ghi thích hợp.

5. Lưu trữ kết quả (*Store Resutls*): có nhiệm vụ lưu trữ nội dung một thanh ghi vào bộ nhớ.

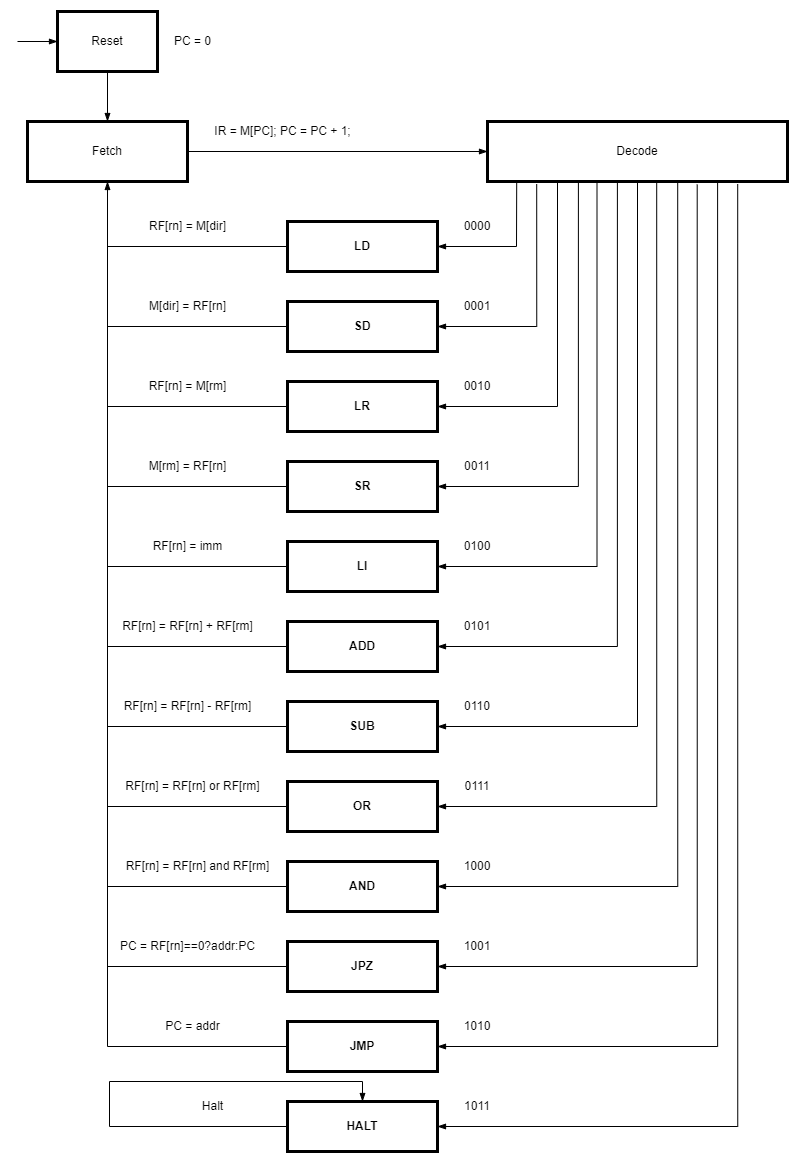
Nếu mỗi giai đoạn mất một chu kỳ xung nhịp, thì chúng ta có thể thấy rằng một lệnh đơn có thể mất vài chu kỳ để hoàn thành.

# Architecture Design



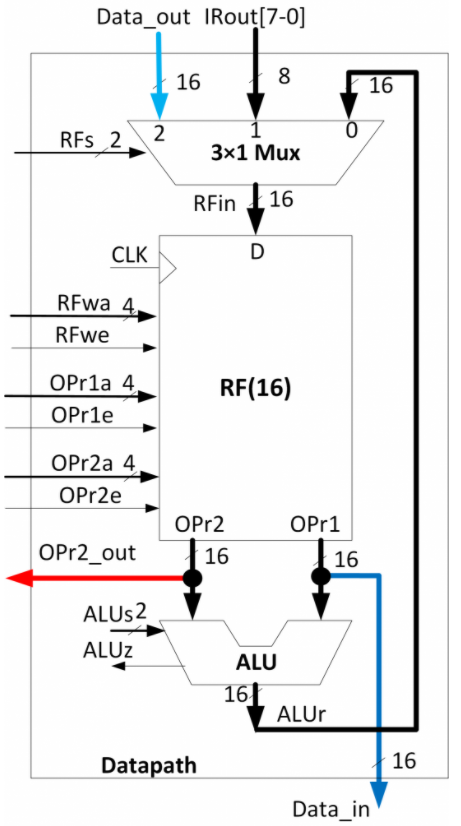
**Hình 2. Cấu trúc mức RTL hoàn thiện của bộ xử lý**

**3.1 FSMD**

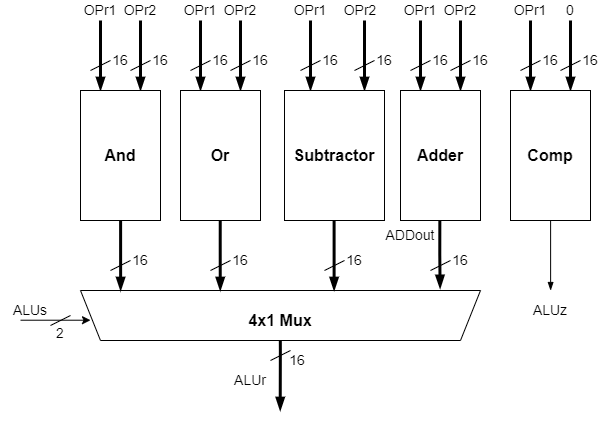


Hình 3. Máy trạng thái FSMD.

## 3.2 Datapath architecture

****

**Hình 4. Cấu trúc Datapath.**



Hình 5. Cấu trúc ALU.

|  |  |
| --- | --- |
| **ALUs** | **ALUr** |
| 00 | OPr1 + OPr2 |
| 01 | OPr1 – Opr2 |
| 10 | OPr1 or OPr2 |
| 11 | OPr1 and OPr2 |

**Bảng 2. Các phép tính của ALU**

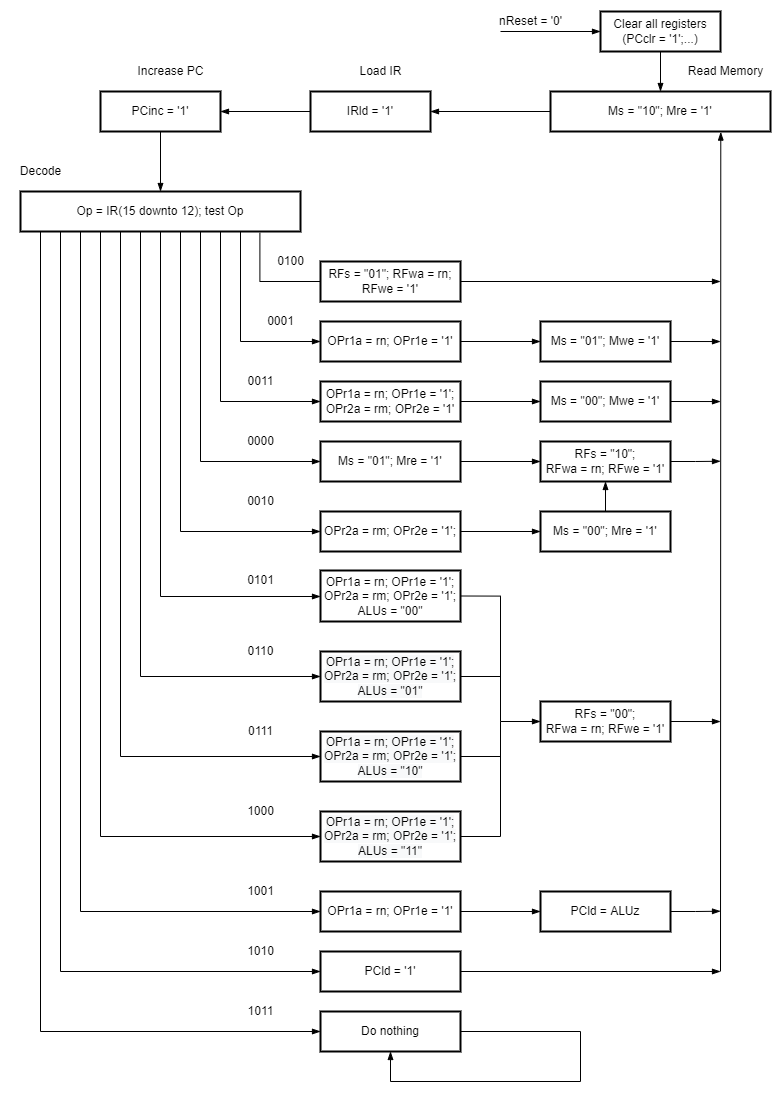


Hình 6: Tệp thanh ghi RF.

## 3.3 Controller



Hình 7. Giao diện ghép nối vào/ra của controller.



Hình 8. Mô hình máy trạng thái FSM của bộ điều khiển.

# Simple System-on-Chip



Hình 9. Thiết kế một hệ thống SoC đơn giản sử dụng CPU phần trên.

# Modeling



Hình 10. Tổ chức của các tệp VHDL.

# Simulation

Kết quả mô phỏng được thể hiện thông qua việc chạy file **cpu\_tb\_opt** trong source code. Chương trình được thực thi là tính tổng array: i = n; while(i-- > n) sum += \*(A++); được cài đặt trong file **dpmem.hdl**. Hoặc tham khảo trong file **Examples.xlsx**.

