**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**



**BÁO CÁO TIẾN ĐỘ ĐỒ ÁN LẦN 1**

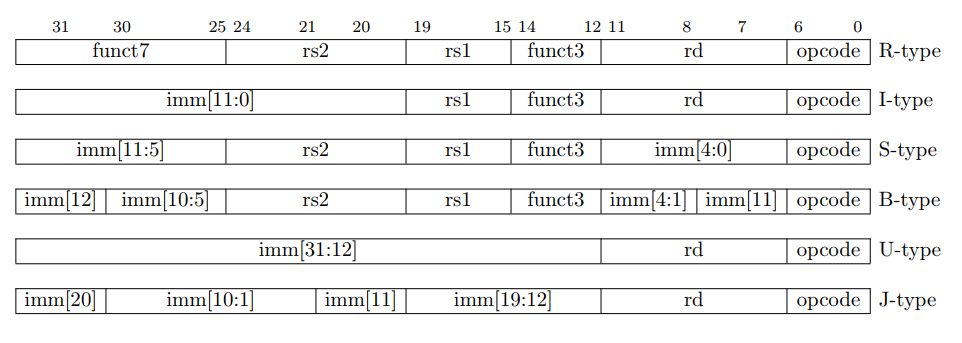
**MÔN THIẾT KẾ HỆ THỐNG SỐ VỚI HDL**

|  |  |  |
| --- | --- | --- |
| **THÀNH VIÊN NHÓM** | **NGUYỄN XUÂN TÙNG**    **NGUYỄN ĐẶNG ANH KIỆT** | **21521649**  **21520312** |
| **LỚP CE213.O11** |  | |

I. Tổng quan đề tài  
- Để thực hiện đề tài thiết kế một bộ xử lý pipeline theo kiến trúc RISC-V 32-bit, ta chia việc thiết kế thành 3 giai đoạn cụ thể như sau:

* Giai đoạn 1 (tiến độ lần 1): Xác định các thành phần phần cứng và vẽ datapath dựa vào tập lệnh và định dạng mã hóa lệnh. Thiết kế controller cho datapath đã có.
* Giai đoạn 2 (tiến độ lần 2): Ứng dụng cơ chế pipeline vào datapath đã có.
* Giai đoạn 3 (báo cáo cuối kì): Xử lý các xung đột và hoàn thiện bộ xử lý (có thể sẽ không làm được).

- Trong bản báo cáo tiến độ lần 1 này ta sẽ trình bày về giai đoạn 1 trong việc thiết kế, xác định lại tập lệnh và định dạng mã hóa lệnh:



Hình 1: Định dạng mã hóa lệnh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instruction** | **Type** | **op-code** | **function 3** | **function 7** |
| LB | I-Type | **0000011** | 000 |  |
| LH | I-Type | **0000011** | 001 |  |
| LW | I-Type | **0000011** | 010 |  |
| LBU | I-Type | **0000011** | 100 |  |
| LHU | I-Type | **0000011** | 101 |  |
| ADDI | I-Type | **0010011** | 000 |  |
| SLLI | I-Type | **0010011** | 001 |  |
| SLTI | I-Type | **0010011** | 010 |  |
| SLTIU | I-Type | **0010011** | 011 |  |
| XORI | I-Type | **0010011** | 100 |  |
| SRLI | I-Type | **0010011** | 101 |  |
| SRAI | I-Type | **1000000** | 101 |  |
| ORI | I-Type | **0010011** | 110 |  |
| ANDI | I-Type | **0010011** | 111 |  |
| AUIPC | U-type | **0010011** |  |  |
| SB | S-Type | **0100011** | 000 |  |
| SH | S-Type | **0100011** | 001 |  |
| SW | S-Type | **0100011** | 010 |  |
| SBU | S-Type | **0100011** | 100 |  |
| SHU | S-Type | **0100011** | 101 |  |
| ADD | R-Type | **0110011** | 000 | 0000000 |
| SUB | R-Type | **0110011** | 000 | 0100000 |
| SLL | R-Type | **0110011** | 001 | 0000000 |
| SLT | R-Type | **0110011** | 010 | 0000000 |
| SLTU | R-Type | **0110011** | 011 | 0000000 |
| XOR | R-Type | **0110011** | 100 | 0000000 |
| SRL | R-Type | **0110011** | 101 | 0000000 |
| SRA | R-Type | **0110011** | 101 | 0100000 |
| OR | R-Type | **0110011** | 110 | 0000000 |
| AND | R-Type | **0110011** | 111 | 0000000 |
| MUL | R-Type | **0110011** | 000 | 0111011 |
| DIV | R-Type | **0110011** | 100 | 0111011 |
| LUI | U-type | **0110111** |  |  |
| JALR | I-Type | **1100111** |  |  |
| JAL | J-Type | **1101111** |  |  |
| BEQ | B-Type | **1100011** | 000 |  |
| BNE | B-Type | **1100011** | 001 |  |
| BLT | B-Type | **1100011** | 100 |  |
| BGE | B-Type | **1100011** | 101 |  |
| BLTU | B-Type | **1100011** | 110 |  |
| BGEU | B-Type | **1100011** | 111 |  |

Bảng 1: Tập lệnh

# II. Datapath & Controller

A diagram of a computer

Description automatically generated

Hình 2: Datapath & Controller

# III. Chi tiết các khối thành phần

## 1. Datapath

### 1.1. Program Counter (PC)

A diagram of a computer component

Description automatically generated

Hình 3: Khối Program Counter (PC)

- Input: CLK, Reset, pc\_next[31:0]

- Output: pc[31:0], pc\_4[31:0]

- Chức năng: lưu địa chỉ của lệnh hiện tại và sau mỗi lệnh được thực thi tăng giá trị PC lên để trỏ đến địa chỉ của lệnh tiếp theo trong bộ nhớ lệnh.

### 1.2. Instruction Memory

A diagram of instruction and instruction

Description automatically generated

Hình 4: Khối Instruction Memory

- Input: CLK, pc[31:0]

- Output: instruction[31:0]

- Chức năng: đọc lệnh tại địa chỉ pc trỏ tới.

### 1.3. Register File

A diagram of a computer

Description automatically generated

Hình 5: Khối Register File

- Input: Instruction[19:15], Instruction[24:20], Instruction[11:7], data\_in[31:0], clk, Reset, w\_reg

- Output: data1[31:0], data2[31:0].

- Chức năng: Đọc một lần hai thanh ghi 32 bit và cho phép ghi vào 1 thanh ghi dựa vào tín hiệu w\_reg (write register).

### 1.4. Sign-extend

A close-up of a sign

Description automatically generated

Hình 6: Khối Sign-extend

- Input: Instruction[31:7], sign\_ext[2:0]

- Output: extend\_imm[31:0]

- Chức năng: dùng để sắp xếp trường imm và mở rộng chúng thành 32 bit dựa vào tín hiệu điều khiển sign\_ext (đối với các lệnh loại I,S,B,U,J).

**A screenshot of a computer

Description automatically generated**

### 1.5. ALU

A diagram of a computer code

Description automatically generated with medium confidence

Hình 7: Khối ALU

- Input: data1[31:0], operand\_b[31:0], alu\_op[4:0]

- Output: alu\_result[31:0], isbranch

- Chức năng: Tính toán giá trị.

|  |  |  |
| --- | --- | --- |
| ALUOP[4:0] | RESULT[31:0] | BRANCH |
| 00000 | ADD | 0 |
| 00001 | SLL | 0 |
| 00010 | SLT | 0 |
| 00011 | SLTU | 0 |
| 00100 | XOR | 0 |
| 00101 | SRL | 0 |
| 00110 | OR | 0 |
| 00111 | AND | 0 |
| 01000 | 0 | BEQ |
| 01001 | 0 | BNE |
| 01010 | 0 | 0 |
| 01011 | 0 | 0 |
| 01100 | 0 | BLT |
| 01101 | 0 | BGE |
| 01110 | 0 | BLTU |
| 01111 | 0 | BGEU |
| 10000 | SUB | 0 |
| 10001 | ADD | 1 |
| 10010 | 0 | 0 |
| 10011 | 0 | 0 |
| 10100 | 0 | 0 |
| 10101 | SRA | 0 |
| 10110 | 0 | 0 |
| 10111 | 0 | 0 |
| 11000 | MUL | 0 |
| 11100 | DIV | 0 |
| 11101 | REM | 0 |
| 11110 | FWD | 0 |
| 11111 | REMU | 0 |

### 1.6. Data Memory

A diagram of a computer

Description automatically generated

Hình 8: Khối Data Memory

- Input: alu\_result[31:0], data2[31:0], r\_dm[2:0], w\_dm[1:0].

- Output: memory\_out[31:0]

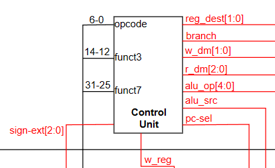
- Chức năng: Đọc dữ liệu bộ nhớ và ghi giá trị vào bộ nhớ dựa vào tín hiệu r\_dm(read data memory) và w\_dm(write data memory).

### 1.7. Others Module

- Mux

- Adder

## 2. Controller



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **Type** | **r\_dm** | **w\_dm** | **w\_reg** | **alu\_src** | **Branch** | **reg\_dest** | **sign\_ext** | **pc\_sel** |
| LB | I-Type | **1** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| LH | I-Type | **2** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| LW | I-Type | **3** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| LBU | I-Type | **4** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| LHU | I-Type | **5** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| ADDI | I-Type | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| SLTI | I-Type | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| SLTIU | I-Type | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| XORI | I-Type | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| ORI | I-Type | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| ANDI | I-Type | **0** | **0** | **1** | **1** | **0** | **1** | **0** | **x** |
| SLLI | I-Type | **0** | **0** | **1** | **1** | **0** | **1** | **1** | **x** |
| SRLI | I-Type | **0** | **0** | **1** | **1** | **0** | **1** | **1** | **x** |
| SRAI | I-Type | **0** | **0** | **1** | **1** | **0** | **1** | **1** | **x** |
| SB | S-Type | **0** | **1** | **0** | **1** | **0** | **x** | **2** | **x** |
| SH | S-Type | **0** | **2** | **0** | **1** | **0** | **x** | **2** | **x** |
| SW | S-Type | **0** | **3** | **0** | **1** | **0** | **x** | **2** | **x** |
| ADD | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| SUB | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| SLL | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| SLT | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| SLTU | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| XOR | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| SRL | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| SRA | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| OR | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| AND | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| MUL | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| DIV | R-Type | **0** | **0** | **1** | **0** | **0** | **0** | **x** | **x** |
| LUI | U-type | **0** | **0** | **1** | **1** | **0** | **0** | **4** | **x** |
| AUIPC | U-type | **0** | **0** | **1** | **x** | **0** | **0** | **4** | **x** |
| JALR | I-Type | **0** | **0** | **1** | **1** | **1** | **3** | **0** | **1** |
| JAL | J-Type | **0** | **0** | **1** | **x** | **1** | **3** | **5** | **0** |
| BEQ | B-Type | **0** | **0** | **0** | **0** | **1** | **x** | **3** | **0** |
| BNE | B-Type | **0** | **0** | **0** | **0** | **1** | **x** | **3** | **0** |
| BLT | B-Type | **0** | **0** | **0** | **0** | **1** | **x** | **3** | **0** |
| BGE | B-Type | **0** | **0** | **0** | **0** | **1** | **x** | **3** | **0** |
| BLTU | B-Type | **0** | **0** | **0** | **0** | **1** | **x** | **3** | **0** |
| BGEU | B-Type | **0** | **0** | **0** | **0** | **1** | **x** | **3** | **0** |
|  |  |  |  |  |  |  |  |  |  |

# IV. Kết quả kiểm thử

Phần mềm sử dụng: ModelSim.

Instructions được kiểm tra:

addi $0, $1, 8 //$0 = 8, pc = 0

sw $0, 1($2) // pc = 4

lw $1, 1($2) //$1 = 8, pc = 8

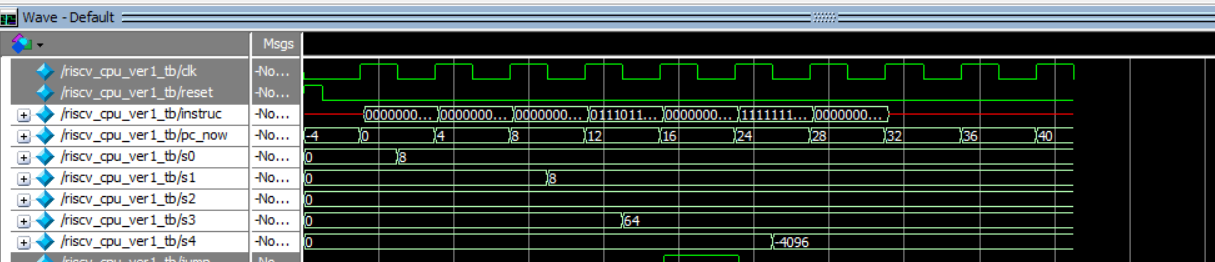
mul $3, $0, $1 //$3 = 64, pc = 12

beq $0, $1, End //will make a branch, pc = 16

add $3, $0, $1 //$3 = 16, pc = 20

End: lui $4, 0xFFFFF //$4 = -4096, pc = 24

nop // pc = 28



instruction

pc now

$0

$1

$2

$3

$4