**BÁO CÁO ĐỒ ÁN CUỐI KÌ**

**INSTRUCTION OF MIPS**

**GVHD**: Hồ Ngọc Diễm

Họ tên: Trần Triều Trung

MSSV: 21522727

**MỤC LỤC**

1. **Giới thiệu về đề tài**
2. **Các thanh ghi**
   * 1. Program Counter (PC)
     2. Instruction Memory (IM)
     3. Register File (RF)
     4. ALU
     5. Data Memory
     6. Control Unit
     7. Jump Address, PCNext
3. **Run**

**Bắt đầu**

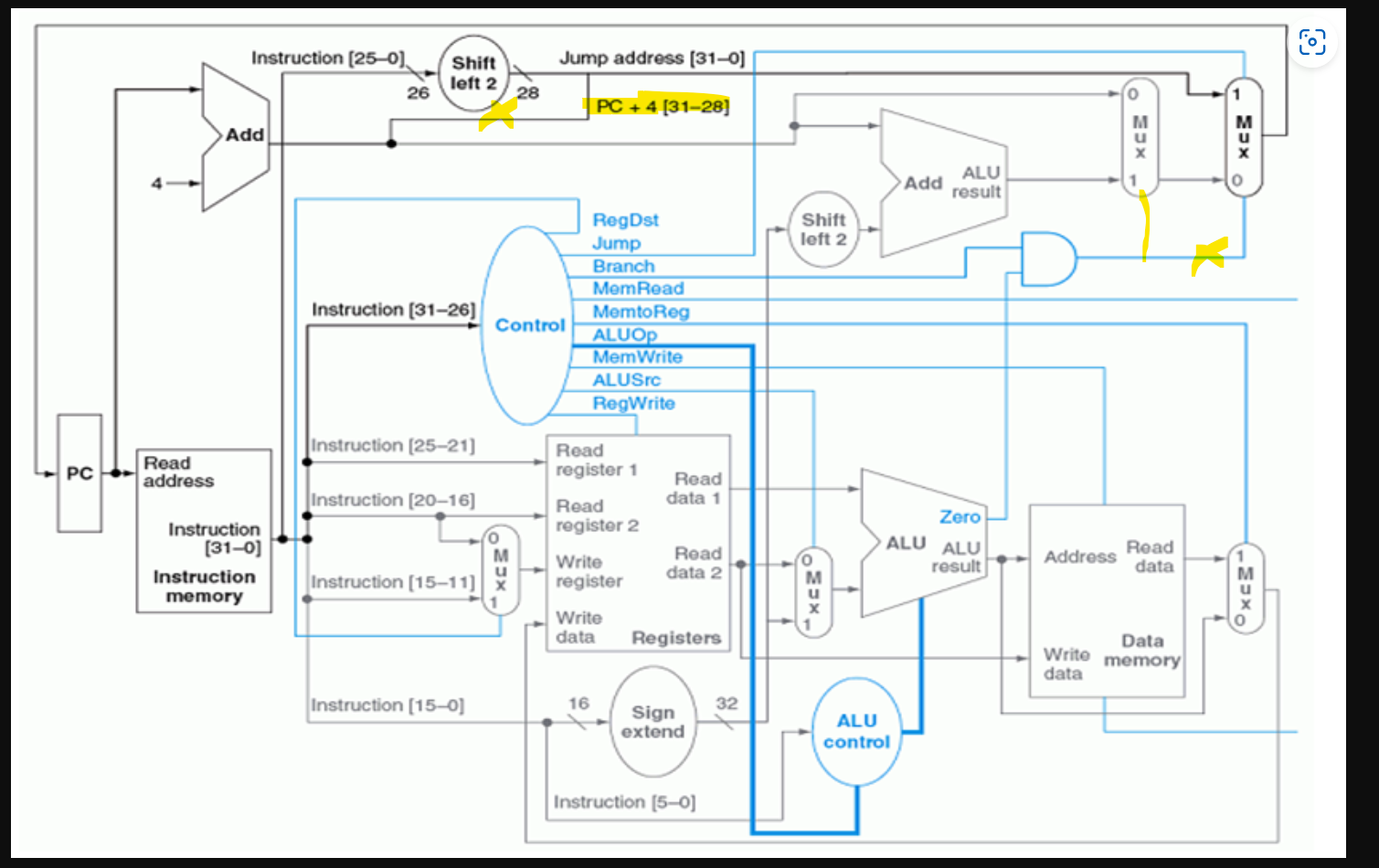
1. **Giới thiệu về đề tài**

Đề tài được sử dụng theo kiến trúc MIPS và tập lệnh của MIPS.

Sử dụng một số lệnh như sau:

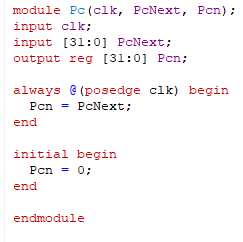
|  |  |  |  |
| --- | --- | --- | --- |
| **Add** | **Addi** | **Sub** | **Sll** |
| **And** | **Lw** | **Beq** | **Srl** |
| **Or** | **Sw** | **Slt** | **J** |

Kiến trúc được mô tả như hình sau:



1. **Các thanh ghi**
2. **Program Counter (PC)**

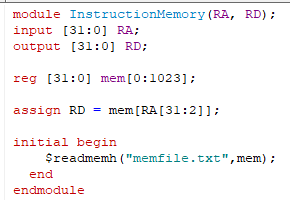
Sẽ là 1 reg để lưu địa chỉ hiện tại của chương trình như sau:



Với khởi tạo ban đầu cho chương trình biết ở địa chỉ 0x00000000

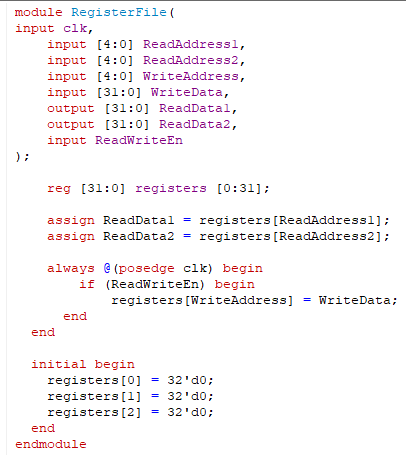
1. **Instruction Memory (IM)**

Ở đây là nơi lưu địa chỉ lệnh nhập từ file txt ở bên ngoài đã chuyển sang mã máy là hex.



1. **Register File (RF)**

Thanh ghi này là nơi lưu giá trị của thanh ghi gồm 32 thanh ghi có độ dài 1 word.



1. **ALU**

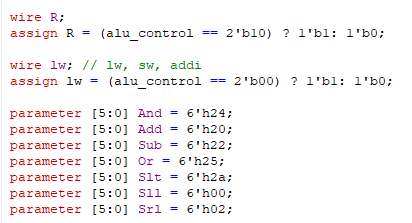
Tại đây gồm 2 giai đoạn:

Decode ALUOp từ Control Unit và Inst[5:0] để ra được ALUcontrol

Sử dụng ALUcontrol để giải quyết

Giai đoạn 1:

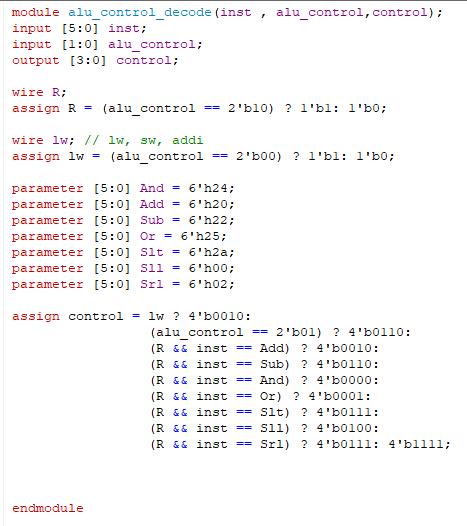
Định nghĩa các opcode của các lệnh dùng ở I



Bảng sự thật:

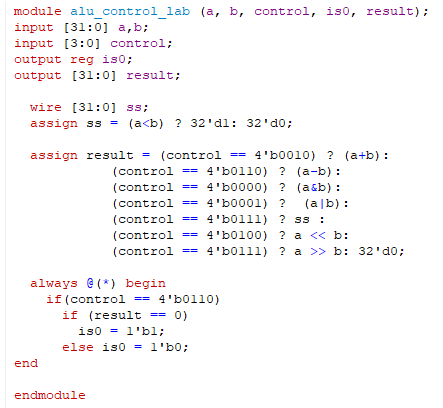
|  |  |
| --- | --- |
| add | 0010 |
| sub | 0110 |
| lw | 0010 |
| sw | 0010 |
| beq | 0110 |
| and | 0000 |
| or | 0001 |
| slt | 0111 |
| sll | 0100 |
| srl | 0111 |

Code:



Giai đoạn 2:

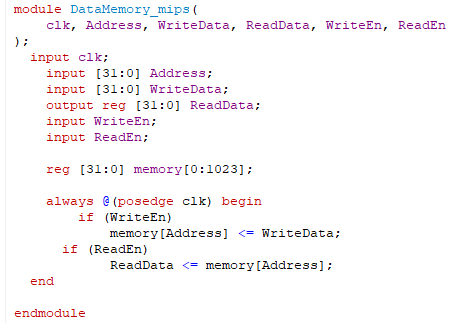
Xử lí:



1. **Data Memory**

Nơi lưu trữ gồm 1024 thanh ghi 32 bits lưu trữ

MemRead, MemWrite cho phép đọc, ghi



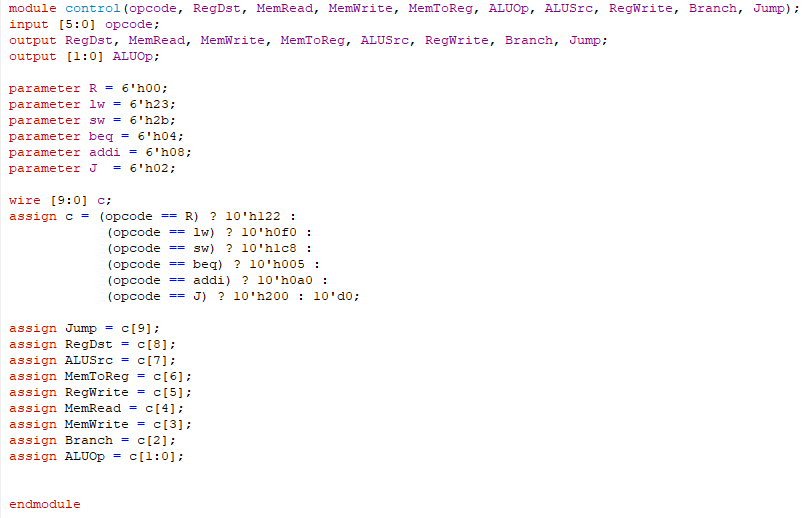
1. **Control Unit**

Đây là cần thiết cấp đầu ra cho các thành phần trong Datapath

**Bảng sự thật:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **R-type** | **Lw** | **Sw** | **Beq** | **Addi** | **J** |
| **Jump** | 0 | 0 | 0 | 0 | 0 | 1 |
| **RegDst** | 1 | 1 | 0 | 0 | 0 | 0 |
| **ALUSrc** | 0 | 1 | 1 | 0 | 1 | 0 |
| **MemToReg** | 0 | 1 | 1 | 0 | 0 | 0 |
| **RegWrite** | 1 | 1 | 1 | 0 | 1 | 0 |
| **MemRead** | 0 | 0 | 1 | 0 | 0 | 0 |
| **MemWrite** | 0 | 0 | 0 | 0 | 0 | 0 |
| **Branch** | 0 | 0 | 0 | 1 | 0 | 0 |
| **ALUOp** | 10 | 0 | 0 | 1 | 0 | 0 |

Code:

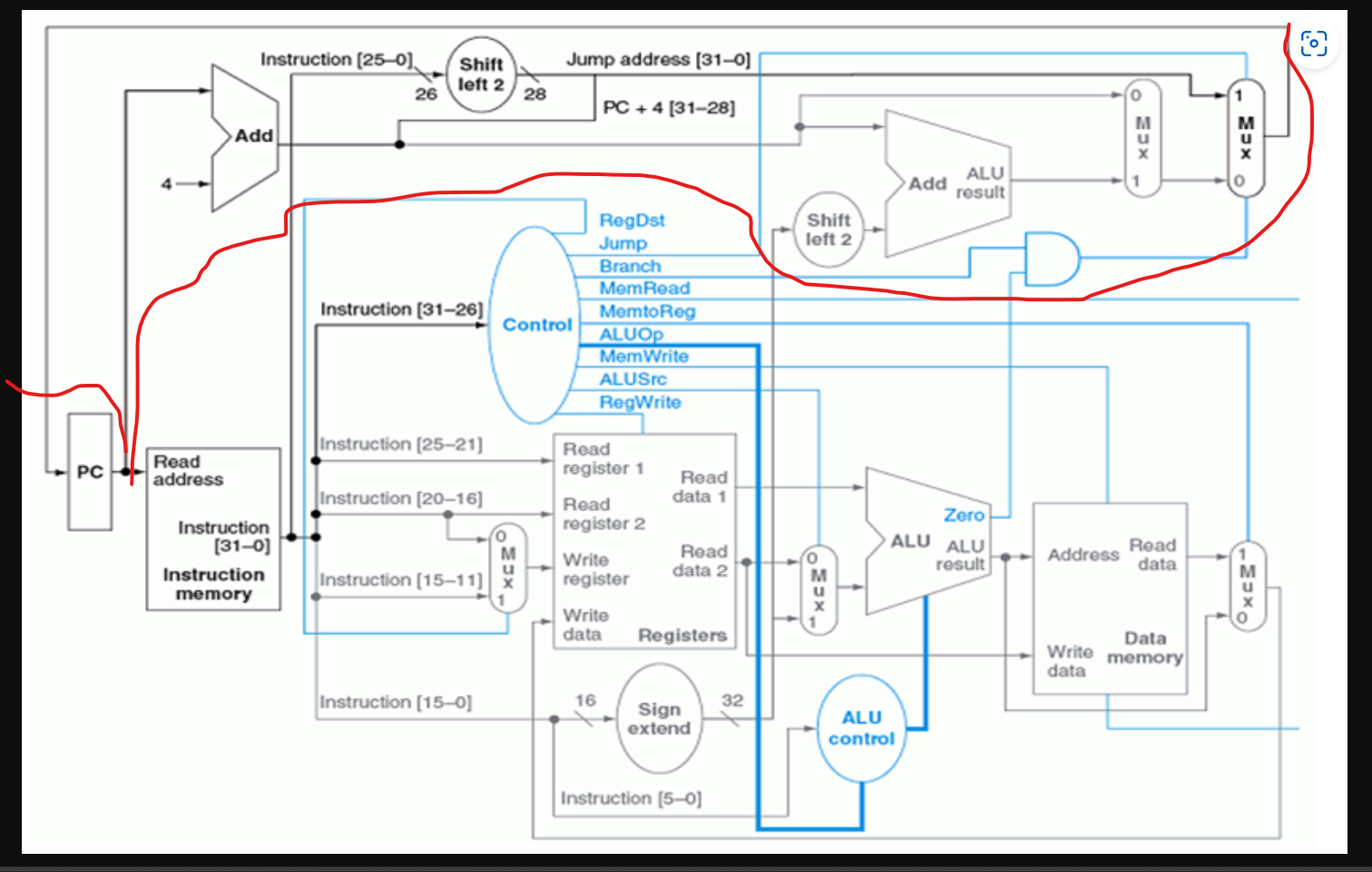


1. **Jump Address, PCNext**

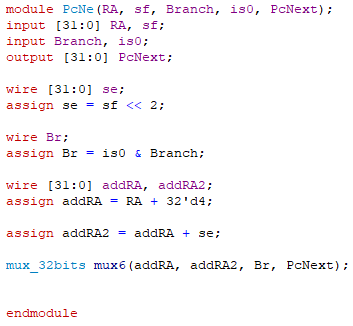
Ở lệnh Jump trong MIPS sẽ cần 26 bit [25:0] để đẩy lên địa chỉ PC



PCNext là nơi ta xử lí phần khoanh đỏ:







**III. Run**

Đầu tiên cần viết mã code bằng assembly

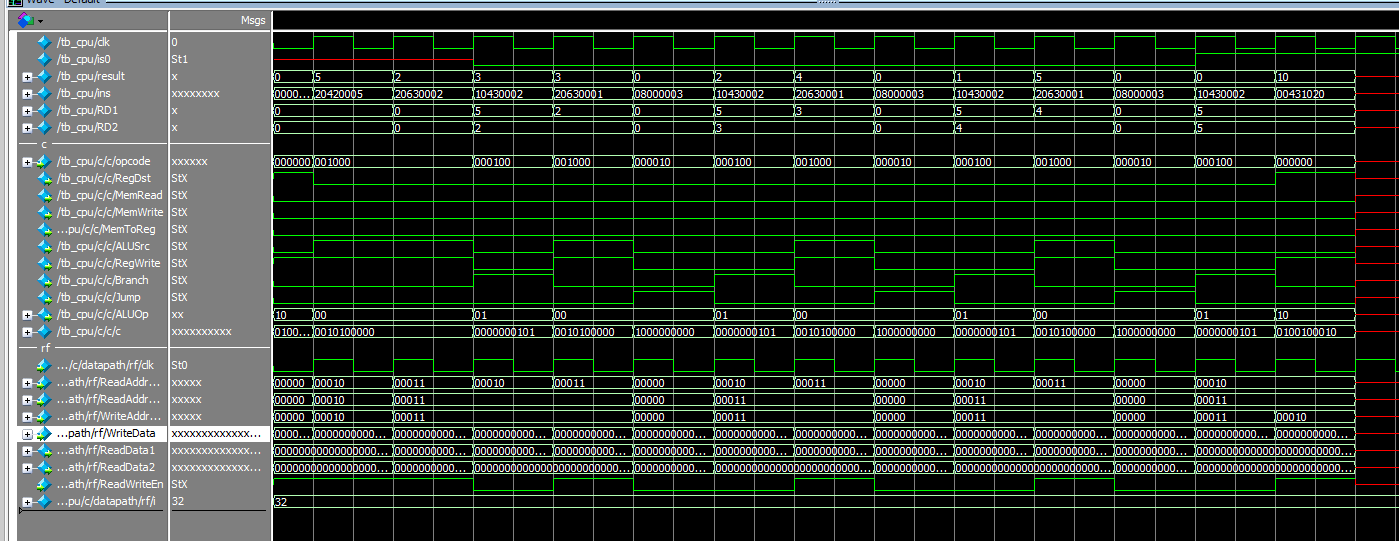
**Ví dụ:**

|  |
| --- |
| addi $2, $2, 5  addi $3, $3, 2  M:  beq $2, $3, X  addi $3, $3, 1  j M  X:  add $2, $2, $3 |

**Sau đó chuyển qua mã máy bằng MARS**

|  |
| --- |
| 00000000  20420005  20630002  10430002  20630001  08000003 (Lưu ý khác MARS)  00431020 |

**Waveform:**



----------HẾT----------