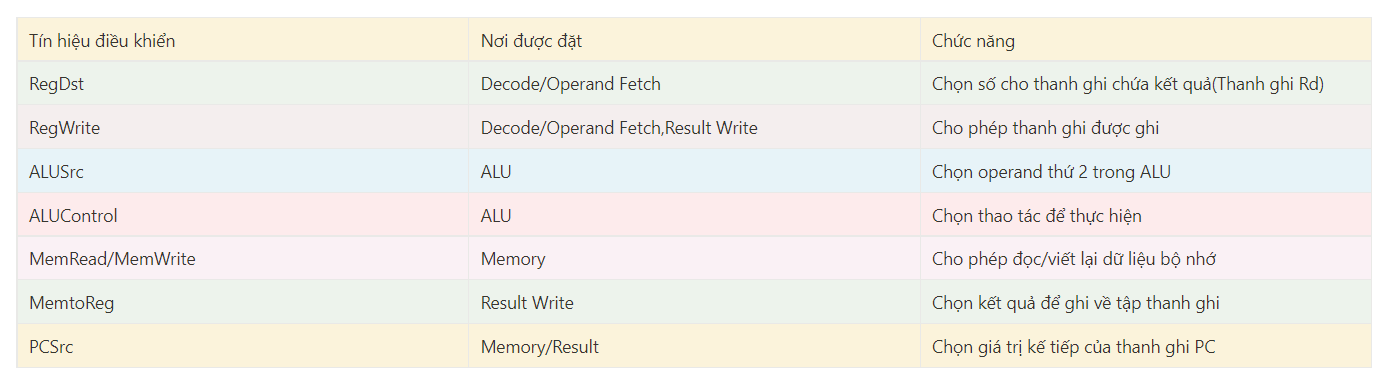
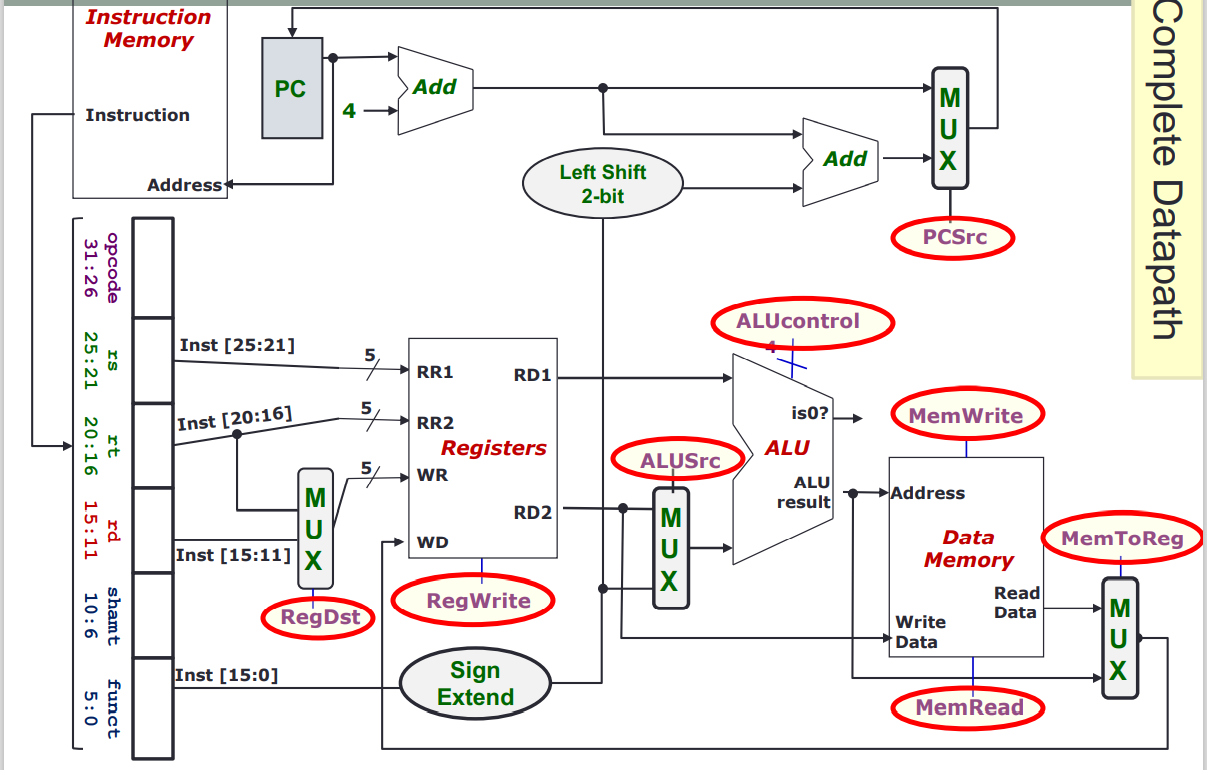
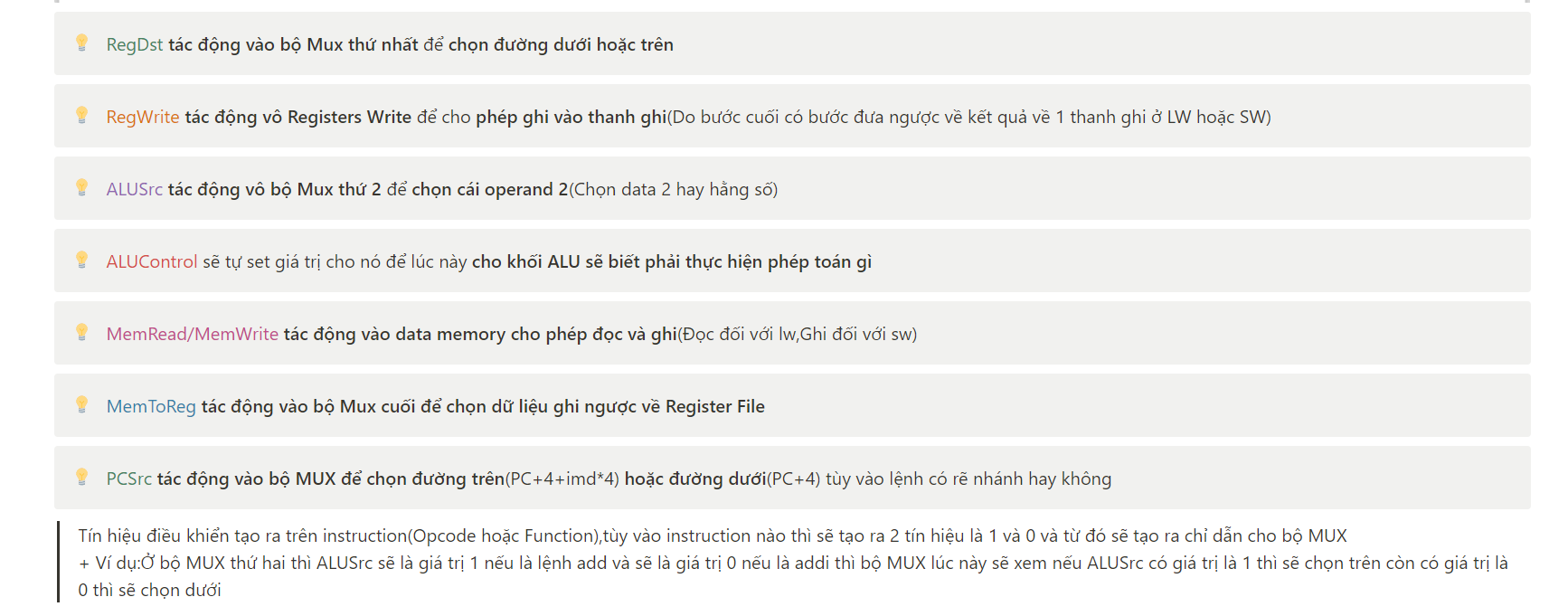
Control

1. *Các tín hiệu điều khiển của control*

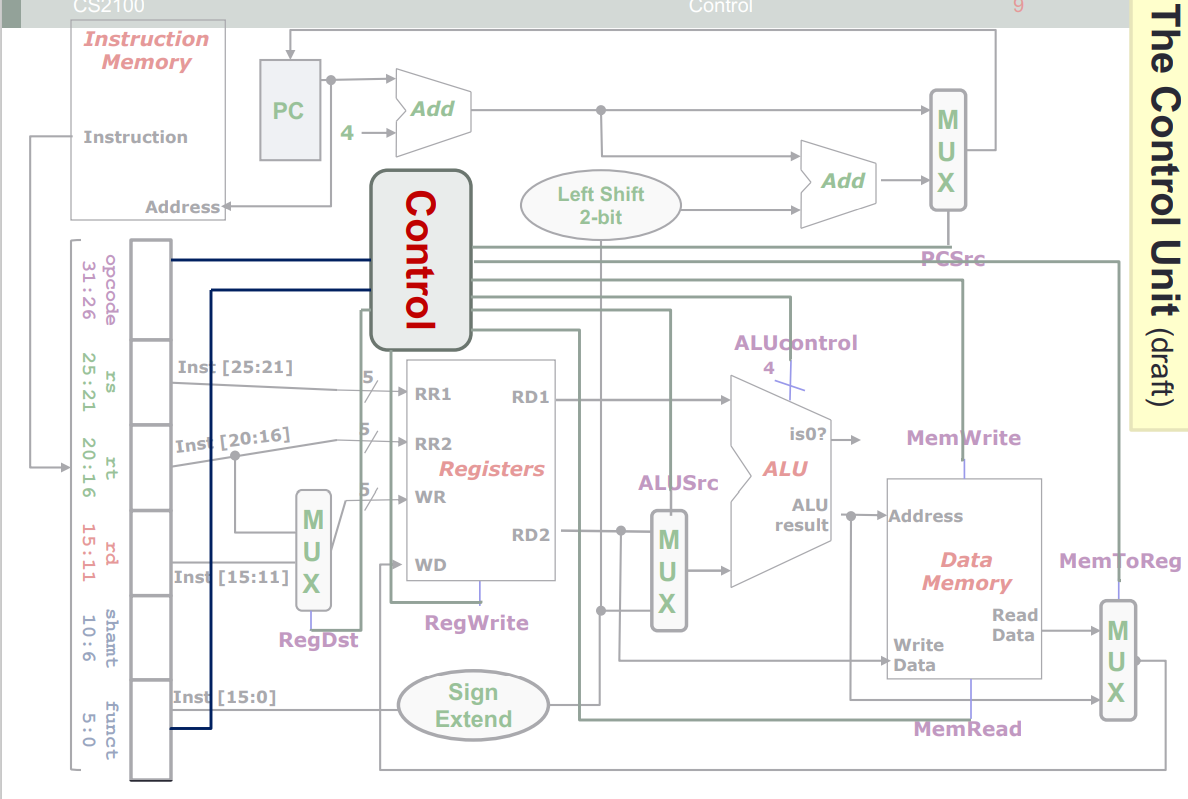


\*Các tín hiệu control trong hình Datapath





\* Mô hình sơ khai control trong Datapath



## *II.Tìm hiểu rõ các tín hiệu điều khiển(Bật cái hình tín hiệu control trong Datapath mà học)*

### *1.Register Destination(MUX đầu)*

* Khi mang tín hiệu là 0 thì Write Register tương ứng với chỉ thị 20-16⇒Chọn đường trên(add)
* Khi mang tín hiệu là 1 thì Write Register tương ứng với chỉ thị 15-11⇒Chọn đường dưới(addi)

### *2.Register Write*

* Khi mang tín hiệu là 0 thì Write Register sẽ không ghi
* Khi mang tín hiệu là 1 thì Write Register sẽ được ghi 1 giá trị mới

### *3.ALUSrc(MUX thứ hai)*

* Khi mang tín hiệu là 0 thì MUX sẽ chọn đường trên⇒Operand 2 = Data 2
* Khi mang tín hiệu là 1 MUX sẽ chọn đường dưới⇒Operand 2 tương ứng với chỉ thị 15-11

### *4.MemRead*

* Khi bằng 0 thì sẽ không cho thằng Data Memory đọc giá trị
* Khi bằng 1 thì sẽ cho phép Data Memory đọc giá trị là cái địa chỉ được tính ở bước ALU

### *5.MemWrite*

* Khi bằng 0 thì sẽ không cho thằng Data Memory ghi giá trị
* Khi bằng 1 thì sẽ cho phép Data Memory ghi giá trị là data2

### *6.Memory to register*

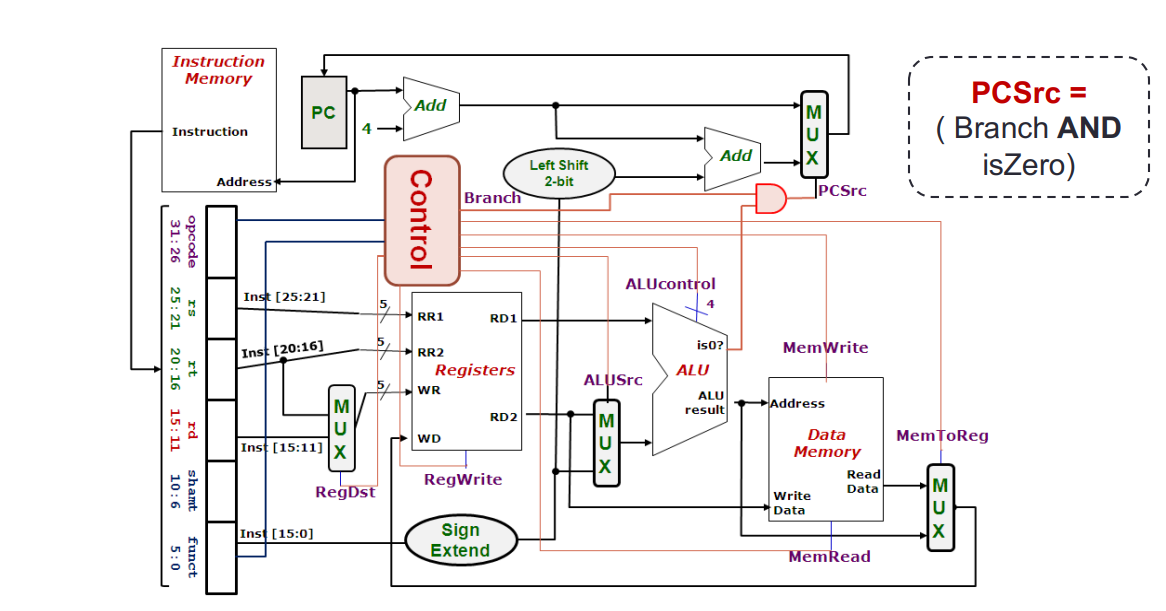
* Khi bằng 1 thì giá trị thanh ghi = Giá trị Read Data được xuất ra
* Khi bằng 0 thì giá trị thanh ghi = Giá trị được tính toán ở ALU

Đối chiếu với hình thì thằng Memory to register bị ngược lại khi mang giá trị 1 thì MUX sẽ chọn đường trên còn mang giá trị 0 thì MUX sẽ chọn đường dưới

### *7.PCSrc(Tác động lên MUX trên cùng)*

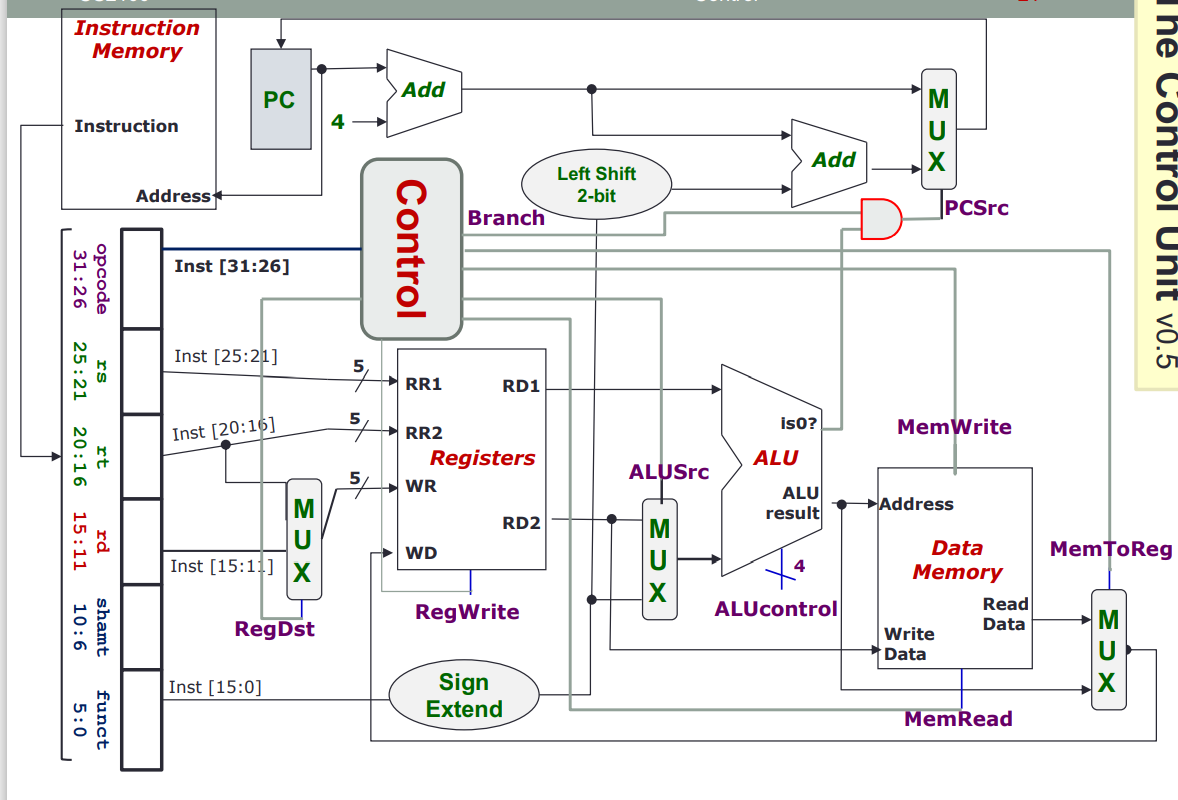
* IsZero cho biết lệnh sẽ có rẽ nhánh hoặc không rẽ nhánh

\*Hình của control IsZero



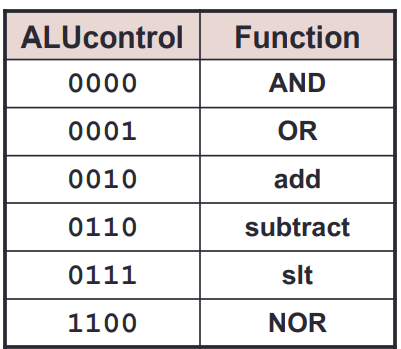
* Nếu PCSrc=0 ⇒Giá trị thanh ghi kế tiếp thanh ghi PC = PC+4
* Nếu PCSrc=1 ⇒Giá trị thanh ghi kế tiếp thanh ghi PC = SignExt(Inst[15:0]) << 2 + (PC + 4)

\*Phần lớn bộ control chỉ tạo dựa tên Opcode(Không cần funct) nên người ta có thể rút lại sơ đồ như sau:

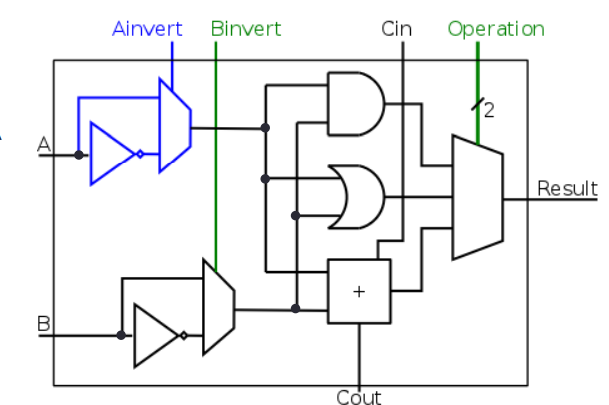


1. *Tìm hiểu về ALU Control*

\*Chức năng: Làm được 6 phép toán số học/logic ở bảng dưới dựa vào tín hiệu điều khiển 4 bit



\*Sơ đồ ALU 1bit đơn giản:



### \*Bảng sự thật cho sơ đồ ALU trên

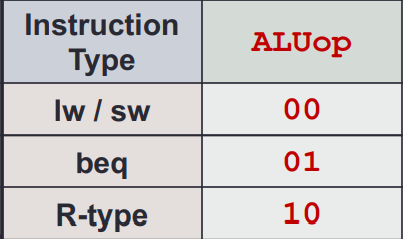
### :image (7)

### *IV.Thiết kế ALU Control nâng cao*

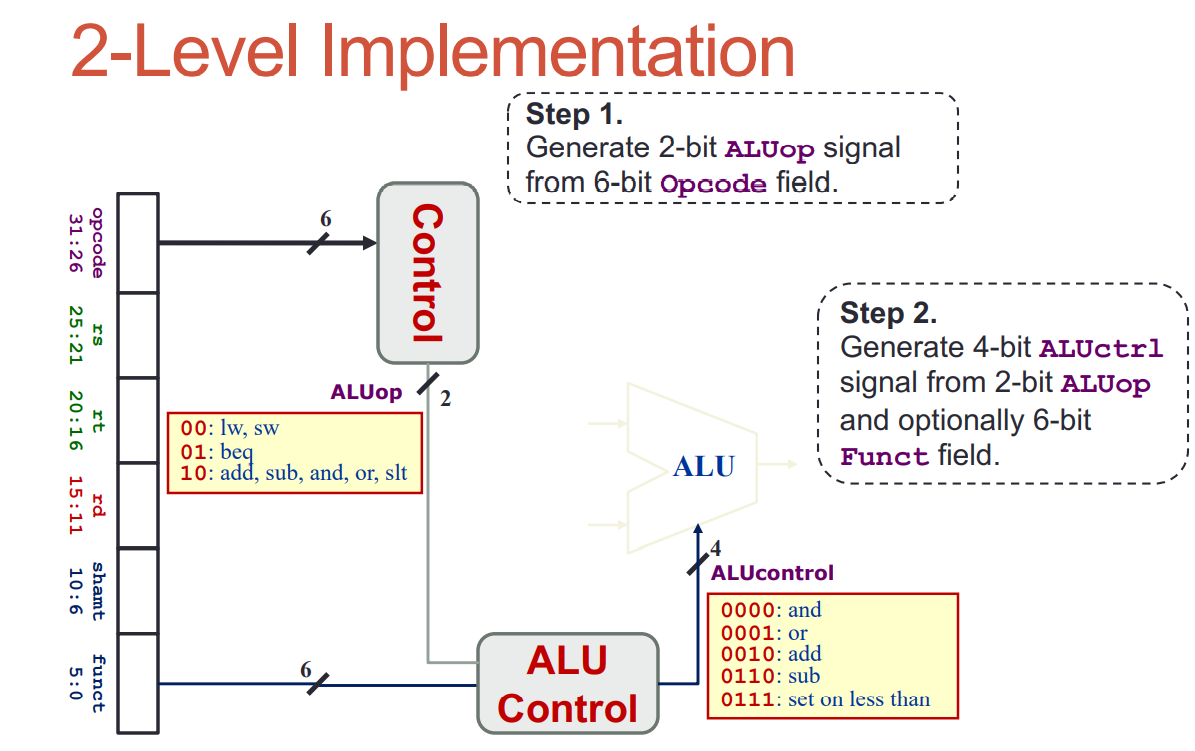
* Không thể dùng Opcode với Function trực tiếp vì tới 12 biến ⇒Xây dựng bảng trạng thái rất mệt

⇒ **Dùng loại thiết kế Multilevel Decoding để đơn giản hóa thiết kế,giảm độ lớn của của vi mạch chính**

**\***Ý tưởng dùng Opcode có số bit ban đầu là 6 tạo ra 2 bit được gọi là ALUOp Signal như dưới:

****

**\***Sơ đồ mạch hoàn thiện đầu vào ALU Control:



*⇒Tổng kết: ALU sẽ đươc tác động bởi ALU Control và ALU Control sẽ được tác động bởi Opcode 6 bit và Funct 6 bit*