**BÁO CÁO BÀI THỰC HÀNH LAB3**

Môn: Tổ chức và Cấu Trúc máy tính – IT012.L14

Họ tên: Nguyễn Đại Kỳ

Mssv: 19521731

**2.1 Mô phỏng việc thực thi các lệnh và cho biết chức năng:**

add, addi, addu, addiu, sub, subu, and, andi, or, nor, lw, sw, slt, slti, sltu, sltiu, syscall

* *Lệnh add:*

Ý nghĩa: R[rd] = R[rs] + R[rt]

Thực hiện cộng giá trị thanh ghi rs với giá trị thanh ghi rt, tổng đưa vào thanh ghi rd

* *Lệnh addi:*

Ý nghĩa: R[rt] = R[rs] + SignExtImm

Thực hiện cộng giá trị thanh ghi rs với số tức thời, kết quả đưa vào thanh ghi rt.

* *Lệnh addu và addiu:*

Addiu có cú pháp và thực hiện chức năng giống addi. Addu có cú pháp và thực hiện chức năng giống add. Tuy nhiên, addiu và addu không xét kết quả có bị overflow hay không, trong khi đó addi và add sẽ báo khi overflow xuất hiện.

* *Lệnh sub và subu:*

Lệnh sub có cú pháp tương tư như lệnh add, nhưng add thực hiện phép toán cộng 2 thanh ghi, kết quả lưu vào thanh ghi thứ 3. Trong khi đó, sub thực hiện phép toán trừ 2 thanh ghi, kết quả lưu vào thanh ghi thứ 3.

Lệnh subu có cú pháp và chức năng giống như sub, nhưng subu không xét đến kết quả có bị overflow hay không, còn sub có xét đến kết quả có bị overflow hay không: nếu bị overflow, sẽ có thông báo.

* *Lệnh and:*

Ý nghĩa: R[rd] = R[rs] & R[rt]

Thực hiện and từng bit giá trị của thanh ghi rs và rt với nhau, kết quả lưu vào thanh ghi rd.

* *Lệnh andi:*

Ý nghĩa: R[rt] = R[rs] & ZeroExtImm

Lệnh này thực hiện and từng bit giá trị thanh ghi rs và một số tức thời. Số tức thời đang là số 16 bits, mở rộng thành số 32 bits theo kiểu ZeroExtImm, tức 16 bits nữa cao còn thiếu sẽ điền 0 vào. Sau đó thực hiện and từng bit giá trị của thanh ghi rs và số tức thời đã được mở rộng thành 32 bits với nhau, kết quả lưu vào thanh ghi rd.

* *Lệnh or và nor:*

Or và Nor cách viết tương tự như and, nhưng thay vì thực hiện phép toán and, 2 lệnh này sẽ thực hiện phép toán or hoặc nor cho từng bit trong 2 thanh ghi, kết quả lưu vào thanh ghi thứ 3.

* *Lệnh lw:*

Ý nghĩa: R[rt] = M[R[rs] + SingExtImm]

Lấy giá trị trong thanh ghi rs cộng với số tức thời đang lưu trong offset (số tức thời này này được mở rộng có dấu thanh 32 bits) ta được địa chỉ của từ nhớ cần lấy dữ liệu. Dữ liệu của từ nhớ này sẽ được lấy để lưu vào thanh ghi rt.

* *Lệnh sw:*

Ý nghĩa: M[R[rs] + SignExtImm] = R[rt]

Lưu giá trị thanh ghi rt vào từ nhớ có địa chỉ được tính bằng giá trị thanh ghi rs cộng với offset (offset được mở rộng có dấu thành số 32 bits trước khi cộng).

* *Lệnh slt và sltu:*

Ý nghĩa: slt: R[rd] = (R[rs] < R[rt])? 1 : 0

Kiểm tra xem giá trị trong thanh ghi rs có nhỏ hơn thanh ghi rt hay không, nếu nhỏ hơn thì thanh ghi rd nhận giá trị là 1; ngược lại thanh ghi rd sẽ nhận giá trị 0

sltu: Ý nghĩa thực hiện giống như slt. Nhưng việc kiểm tra giá trị thanh ghi rs có nhỏ hơn thanh ghi rt hay không trong lệnh slt thực hiện trên số có dấu, còn trong sltu thực hiện trên số không dấu

* *Lệnh slti và sltiu:*

slti/sltiu: R[rd] = (R[rs] < SignExtImm)? 1 : 0

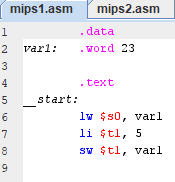
Ý nghĩa 2 lệnh này giống nhau là so sánh giá trị một thanh ghi với một số tức thời, nếu giá trị trong thanh ghi rs nhỏ hơn số tức thời thì thanh ghi rd nhận giá trị là 1; ngược lại thanh ghi rd sẽ nhận giá trị 0 Số tức thời cho phép trong lệnh này là số 16 bits. Trước khi so sánh với thanh ghi rs, số tức thời được mở rộng có dấu (SignExtImm) thành số 32 bits.

slti khác sltiu là slti so sánh 2 giá trị theo kiểu có dấu dạng bù 2, trong khi đó sltiu so sánh theo kiểu số không dấu.

- *Lệnh syscall:* Nhập hoặc xuất giá trị của thanh ghi.

**2.2 Mô phỏng và cho biết ý nghĩa chương trình:**

*- Ví dụ 1:*



Ý nghĩa:

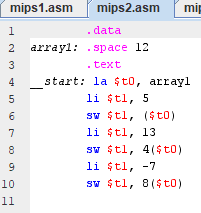
Khai báo biến var1 có kiểu dữ liệu word giá trị là 23.

Load giá trị biến var1 vào thanh ghi $s0

Load giá trị 5 vào thanh ghi $t1.

Save giá trị thanh ghi $t1 vào biến var1.

*-Ví dụ 2:*



Ý nghĩa:

Khai báo mảng array1 12 phần tử.

Load mảng array1 vào thanh ghi $t0

Load giá trị 5 vào thanh ghi $t1.

Store word địa chỉ tại thanh ghi $t0 = $t1

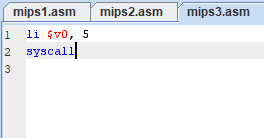
Load giá trị $t1 bằng 13.

Store giá trị $t1 vào $t0[value+4]

Load giá trị $t1 bằng -7.

Store giá trị $t1 vào $t0[value+8]

*-Ví dụ 3:*

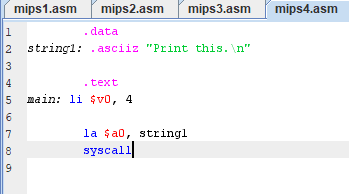


Ý nghĩa:

Load giá trị thanh ghi $v0 bằng 5.

syscall: có chức năng đưa giá trị vào thanh ghi $v0.

*-Ví dụ 4:*



Ý nghĩa:

Khai báo biến string1 có giá trị: “Print this.\n”

Load giá trị thanh ghi $v0 bằng 4.

Load string1 vào thanh ghi $a0

syscall: xuất giá trị $a0 ra màn hình.

**3.1 Nhập vào một chuỗi, xuất ra cửa sổ I/O của MARS theo từng yêu cầu sau:**

*a) Xuất ra lại đúng chuỗi đã nhập:*

Nhap: Truong Dai hoc Cong nghe Thong tin

Xuất: Truong Dai hoc Cong nghe Thong tin

