Họ và tên: Nguyễn Huỳnh Duy Hiếu

MSSV: 22520439

**Báo cáo Lab3**

**2. Thực hành**

**2.1:**

|  |  |  |
| --- | --- | --- |
| **Tên** | **Cú Pháp** | **Chức Năng** |
| **add** | add $t0, $t1, $t2 | Thực hiện cộng giá trị thanh ghi $t1 với giá trị thanh ghi $t2, tổng đưa vào thanh ghi $t0 |
| **addi** | addi $t0, $t1, 3 | Thực hiện cộng giá trị thanh ghi $t1 với 3 (số tức thời), tổng đưa vào thanh ghi $t0  Lưu ý: Phạm vi cho số tức thời trong lệnh này là 16 bits. Số tức thời trước khi cộng với thanh ghi $t1 phải được mở rộng có dấu thành (SignExtImm) thành số 32 bits. |
| **addu** | addu $t0, $t1, 0x1 | Addu có cú pháp và thực hiện chức năng giống add |
| **addiu** | addiu $t0, $t1, 1 | Addiu có cú pháp và thực hiện chức năng giống addi  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 |
| **sub** | sub $t0, $t1, $t2 | -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 |
| **subu** | subu $t0, $t1, $t2 | -Lệnh subu có cú pháp và chức năng giống như sub, nhưng  • subu không xét đến kết quả có bị overflow hay không • sub có xét đến kết quả có bị overflow hay không; nếu bị overflow, sẽ có thông báo |
| **and** | and $1,$2,$3 | Thao tác bit AND hai giá trị của thanh ghi $2 với $3, lưu kết quả vào thanh ghi $1 |
| **andi** | andi $1,$2,100 | Thao tác bit AND giá trị của thanh ghi $2 với một số(100) rồi lưu kết quả vào thanh ghi $1 |
| **or** | or $1,$2,$3 | Thao tác bit OR hai giá trị của thanh ghi $2 với $3, lưu kết quả vào thanh ghi $1 |
| or $1,$2,100 | Thao tác bit AND hai giá trị của thanh ghi $2 với một số(100) lưu kết quả vào thanh ghi $1 |
| **nor** | nor $1,$2,$3 | Thao tác bit NOR hai giá trị của thanh ghi $2 với $3, lưu kết quả vào thanh ghi $1 |
| **lw** | lw $1,0($2) | Chuyển dữ liệu(word) từ bộ nhớ vào thanh ghi $1 |
| **sw** | sw $1, 0($2) | Lệnh chuyển dữ liệu(word) từ thanh ghi $1 ra bộ nhớ |
| **slt** | slt $1,$2,$3 | Kiểm tra xem $2 có bé hơn $3 không nếu đúng, $1=1 ngược lại $1=0 |
| **slti** | slti $1,$2,100 | Kiểm tra xem $2 có bé hơn một số(100) không nếu đúng, $1=1 ngược lại $1=0 |
| **sltu** | sltu $1,$2,$3 | Kiểm tra xem $2 có bé hơn $3 không nếu đúng, $1=1 ngược lại $1=0($2, $3 chứa số không dấu) |
| **sltiu** | sltiu $1,$2,100 | Kiểm tra xem $2 có bé hơn một số(100) không nếu đúng, $1=1 ngược lại $1=0($2, $3 chứa số không dấu) |
| **syscall** | syscall | Dùng để yêu cầu dịch vụ từ kernel, mã dịch vụ được truyền từ thanh ghi $v0 |

**2.2**

|  |  |
| --- | --- |
| **Code** | **Giải tích** |
| .data  var1: .word 23  .text  \_\_start:  lw $t0, var1  li $t1, 5  sw $t1, var1 | Khai báo vùng nhớ data  Khai báo biến kiểu word: var1 = 23  Khai báo vùng nhớ text  t0 lưu giá trị var1  t1 = 5  var1 lưu giá trị t1 |

|  |  |
| --- | --- |
| **Code** | **Giải tích** |
| .data  array1: .space 12  .text  \_\_start:  la $t0,array1  li $t1,5  sw $t1, ($t0)  li $t1,13  sw $t1,4($t0)  li $t1,-7  sw $t1, 8($t0) | Cấp 12-byte bộ nhớ, chưa được khởi tạo  t0 = địa chỉ array1  t1 = 5  array1[0] = t1  t1 = 13  array[1] = t1  t1 = -7  array1[2] = t1 |

|  |  |
| --- | --- |
| **Code** | **Giải thích** |
| li $v0,5  syscall | Truyền tham số 5 vào thanh ghi v0  Thực hiện chức năng (đọc số nguyên) |

|  |  |
| --- | --- |
| **Code** | **Giải thích** |
| .data  string1: .asciiz “Print this.\n”  .text  main: li $v0,4    la $a0,string1  syscall | Khai báo mảng string1 = “Print this.\n”  Truyền tham số 4 vào v0(thực hiện chức năng in  chuỗi kí tự mà địa chỉ được lưu trong a0  a0 = đại chỉ string1  Thực hiện chức năng |

**3. Bài tập**

**a)**

.data

string1: .asciiz "Chao ban! Ban la sinh vien nam thu may?\n"

string2: .asciiz "Hihi, minh la sinh vien nam thu 1 ^-^"

.text

main:

li $v0,4

la $a0, string1

la $a0,string2

syscall

**b)**

01101111011000010110100001000011

01101110011000010110001000100000

01100001010000100010000000100001

01100001011011000010000001101110

01101110011010010111001100100000

01101001011101100010000001101000

01101110001000000110111001100101

01110100001000000110110101100001

01101101001000000111010101101000

00001010001111110111100101100001

01101000011010010100100000000000

01101101001000000010110001101001

00100000011010000110111001101001

01110011001000000110000101101100

00100000011010000110111001101001

01101110011001010110100101110110

01101101011000010110111000100000

01110101011010000111010000100000

01011110001000000011000100100000

00000000000010100101111000101101

**c)**

.data

string0: .asciiz "Nhap: "

string1: .space 20

string2: .asciiz "Xuat: "

.text

main: li $v0, 4

la $a0, string0

syscall

li $v0, 8

la $a0, string1

li $a1, 20

syscall

li $v0, 4

la $a0, string2

syscall

li $v0, 4

la $a0,string1

syscall

**d)**

.data

.text

main:

li $v0,5

syscall

move $t1,$v0

li $v0,5

syscall

move $t2,$v0

li $v0,1

add $a0,$t1,$t2

syscall