

Thực hành kiến trúc máy tính

Báo cáo thực hành

Bài 4. Các lệnh số học và logic

Họ Tên	Lê Thành An
MSSV	20235631

ASSIGNMENT 1

ĐOẠN MÃ :

TH1: Không bị tràn số

```
# Laboratory Exercise 4, Home Assignment 1
.text
# TODO: Thiết lập giá trị cho s1 và s2 với trường hợp khác nhau
li s1, 18
li s2, 26
# Thuật toán xác định tràn số
li t0, 0                # Mặc định không có tràn số
add s3, s1, s2          # s3 = s1 + s2
xor t1, s1, s2          # Kiểm tra s1 với s2 có cùng dấu
blt t1, zero, EXIT      # Nếu t1 là số âm, s1 và s2 khác dấu
blt s1, zero, NEGATIVE  # Kiểm tra s1 và s2 là số âm hay không âm
bge s3, s1, EXIT        # s1 không âm, kiểm tra s3 nhỏ hơn s1 không
# Nếu s3 >= s1, không tràn số
j OVERFLOW
NEGATIVE:
bge s1, s3, EXIT        # s1 âm, kiểm tra s3 có lớn hơn s1 không
# Nếu s1 >= s3, không tràn số
OVERFLOW:
li t0, 1                # The result is overflow
EXIT:
```

Giá trị các thanh ghi:

s1	9	0x00000012
s2	18	0x0000001a
t0	5	0x00000000
s3	19	0x0000002c
t1	6	0x00000008

TH2: Bị tràn số (âm)

```
# Laboratory Exercise 4, Home Assignment 1
.text
# TODO: Thiết lập giá trị cho s1 và s2 với trường hợp khác nhau
li s1, -1999999999
li s2, -2147483647
# Thuật toán xác định tràn số
li t0, 0                # Mặc định không có tràn số
add s3, s1, s2          # s3 = s1 + s2
xor t1, s1, s2          # Kiểm tra s1 với s2 có cùng dấu
blt t1, zero, EXIT      # Nếu t1 là số âm, s1 và s2 khác dấu
blt s1, zero, NEGATIVE  # Kiểm tra s1 và s2 là số âm hay không âm
bge s3, s1, EXIT        # s1 không âm, kiểm tra s3 nhỏ hơn s1 không
# Nếu s3 >= s1, không tràn số
j OVERFLOW
NEGATIVE:
bge s1, s3, EXIT        # s1 âm, kiểm tra s3 có lớn hơn s1 không
# Nếu s1 >= s3, không tràn số
OVERFLOW:
li t0, 1                # The result is overflow
EXIT:
Giá trị các thanh ghi:
```

s1	9	0x88ca7000
s1	9	0x88ca6c01
s2	18	0x80000000
s2	18	0x80000001
t0	5	0x00000000
s3	19	0x08ca6c02
t1	6	0x08ca6c00
t0	5	0x00000001

TH3: Bị tràn số (dương)

```
# Laboratory Exercise 4, Home Assignment 1
.text
# TODO: Thiết lập giá trị cho s1 và s2 với trường hợp khác nhau
li s1, 1
li s2, 2147483647
# Thuật toán xác định tràn số
li t0, 0                # Mặc định không có tràn số
add s3, s1, s2          # s3 = s1 + s2
xor t1, s1, s2          # Kiểm tra s1 với s2 có cùng dấu
blt t1, zero, EXIT      # Nếu t1 là số âm, s1 và s2 khác dấu
blt s1, zero, NEGATIVE  # Kiểm tra s1 và s2 là số âm hay không âm
bge s3, s1, EXIT        # s1 không âm, kiểm tra s3 nhỏ hơn s1 không
# Nếu s3 >= s1, không tràn số
j OVERFLOW
NEGATIVE:
bge s1, s3, EXIT        # s1 âm, kiểm tra s3 có lớn hơn s1 không
# Nếu s1 >= s3, không tràn số
OVERFLOW:
```

```
li t0, 1          # The result is overflow
EXIT:
```

Giá trị của các thanh ghi:

s1	9	0x00000001
s2	18	0x80000000
s2	18	0x7fffffff
t0	5	0x00000000
s3	19	0x80000000
t1	6	0x7fffffff
t0	5	0x00000001

ASSIGNMENT 2 :

TH1:

CODE :

```
.text
li s0, 1234567899
srli s1, s0, 24      #MSB của s0
andi s2, s0, 0xffffffff00    #xóa LSB
ori s3, s0, 0x00000000ff     #thiết lập LSB
andi s4, s0, 0x00000000     #xóa S0
```

Giá trị của thanh ghi:

s0	8	0x49960000
----	---	------------

s0	8	0x499602db
----	---	------------

s1	9	0x00000049
----	---	------------

s2	18	0x49960200
----	----	------------

s4	20	0x00000000
----	----	------------

s3	19	0x499602ff
----	----	------------

s4	20	0x00000000
----	----	------------

ASSIGNMENT 3 :

```
.text
addi s0, s0, 1234
sub s1, zero, s0      #s1 =-s0
```

```
add s2, x0, s0      #s2=s0
xori s3, s0, -1     #s3=not(s0)
bge s2, s1, LABEL   #thay the lenh ble bang bge
```

`LABEL:`

Giá trị các thanh ghi:

<code>s0</code>	8	0x000004d2
<code>s1</code>	9	0xfffffb2e
<code>s2</code>	18	0x000004d2
<code>s3</code>	19	0xfffffb2d

ASSIGNMENT 4 :

```
.text
li s1, -9999999999
li s2, -1999999999
addi t0, t0, 0
add s3, s1, s2  #s3 = s1 + s2
xor t1, s3, s1  #so sanh dau cua s3 voi s1
xor t2, s3, s2  #so sanh dau cua s3 voi s2
blt t1, zero, OVERFLOW
blt t2, zero, OVERFLOW
j END
OVERFLOW:
addi t0, t0, 1
END:
```

Giá trị các thanh ghi:

s1	9	0xabf42000
s1	9	0xabf41c01
s2	18	0x88ca7000
s2	18	0x88ca6c01
t0	5	0x00000000
s3	19	0x34be8802
t1	6	0x9f4a9403
t2	7	0xbc74e403
t0	5	0x00000001

ASSIGNMENT 5 :

```
.text

    addi s0, s0, 5
    addi s1, s1, 4
    addi t0, t0, 0    #i=0
    addi t1, t1, 1    #step=1
    addi s3, s3, 2    #s3=2

loop:
    blt s1, s3, continue
    div s1, s1, s3
    add t1, t1, t2
    j loop

continue:
    sll s3, s0, t1 #s3=s1*s0
```

Giá trị của các thanh ghi:

s0	8	0x00000005
s1	9	0x00000004
t0	5	0x00000000
t1	6	0x00000001
s3	19	0x00000002
s1	9	0x00000002
t1	6	0x00000001
s1	9	0x00000001
t1	6	0x00000001
s3	19	0x0000000a