

Laboratory Exercise 4

Bài 1

Ta nhận 2 số từ bàn phím (2 số 32 bit) ta sẽ xét xem 2 số này có tràn hay không, và ta sẽ dùng thanh ghi t0 để kiểm tra, nếu t0 = 1 thì là tràn còn nếu t0 = 0 thì không bị tràn

2 số được lưu trong thanh ghi s1, s2

Tổng 2 số nằm trên thanh ghi s3

Ta có chương trình

Trường hợp 1: 2 số dương không tràn ($s1+s2 < 2^{31}$)

Code

```

1  .text
2  addi $s1, $0, 20200440
3  addi $s2, $0, 2127283207
4  start:
5  li $t0,0 #Thiết lập trạng thái mặc định
6  addu $s3,$s1,$s2 # s3 = s1 + s2
7  xor $t1,$s1,$s2 #kiểm tra xem $s1 và $s2 có cùng dấu với nhau hay không
8  bltz $t1,EXIT # nếu không thì nhảy tới exit
9  slt $t2,$s3,$s1
10 bltz $s1,NEGATIVE #nếu như s1 mà <0 thì nhảy tới NEGATIVE
11 beq $t2,$zero,EXIT #nếu như t2 mà = 0 thì sẽ nhảy tới exit
12     #nếu s3 mà lớn hơn s1 thì kết quả ko bị tràn
13     #nhảy tới OVERFLOW
14 j OVERFLOW
15
16 NEGATIVE:
17 bne $t2,$zero,EXIT # $s1 và $s2 cùng là số âm
18     # nếu s3<s1 thì kết quả sẽ không bị tràn
19
20 OVERFLOW:
21 li $t0,1 #kết quả là tràn
22
23 EXIT:

```

Run

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x3c010134	lui \$1,0x00000134	2: addi \$s1, \$0, 20200440
<input type="checkbox"/>	0x00400004	0x34213bf8	ori \$1,\$1,0x00003bf8	
<input type="checkbox"/>	0x00400008	0x00018820	add \$17,\$0,\$1	
<input type="checkbox"/>	0x0040000c	0x3c017ecb	lui \$1,0x00007ecb	3: addi \$s2, \$0, 2127283207
<input type="checkbox"/>	0x00400010	0x3421c407	ori \$1,\$1,0x0000c407	
<input type="checkbox"/>	0x00400014	0x00019020	add \$18,\$0,\$1	
<input type="checkbox"/>	0x00400018	0x24080000	addiu \$8,\$0,0x00000...	5: li \$t0,0 #Thiết lập trạng thái mặc định
<input type="checkbox"/>	0x0040001c	0x02329821	addu \$19,\$17,\$18	6: addu \$s3,\$s1,\$s2 # s3 = s1 + s2
<input type="checkbox"/>	0x00400020	0x02324826	xor \$9,\$17,\$18	7: xor \$t1,\$s1,\$s2 #kiểm tra xem \$s1 và \$s2 có cùng dấu với nhau hay không
<input type="checkbox"/>	0x00400024	0x05200006	bltz \$9,0x00000006	8: bltz \$t1,EXIT # nếu không thì nhảy tới exit
<input type="checkbox"/>	0x00400028	0x0271502a	slt \$10,\$19,\$17	9: slt \$t2,\$s3,\$s1
<input type="checkbox"/>	0x0040002c	0x06200002	bltz \$17,0x00000002	10: bltz \$s1,NEGATIVE #nếu như s1 mà <0 thì nhảy tới NEGATIVE
<input type="checkbox"/>	0x00400030	0x11400003	beq \$10,\$0,0x00000003	11: beq \$t2,\$zero,EXIT #nếu như t2 mà = 0 thì sẽ nhảy tới exit
<input type="checkbox"/>	0x00400034	0x0810000f	j 0x0040003c	14: j OVERFLOW
<input type="checkbox"/>	0x00400038	0x15400001	bne \$10,\$0,0x00000001	17: bne \$t2,\$zero,EXIT # \$s1 và \$s2 cùng là số âm
<input type="checkbox"/>	0x0040003c	0x24080001	addiu \$8,\$0,0x00000...	21: li \$t0,1 #kết quả là tràn

Khi mà 2 số ko tràn

\$t0	8	0
------	---	---

Khi 2 số mà bị tràn

\$t0	8	1
------	---	---

Ta có 2 số 20200440 và 2127283207 là 2 số mà tổng của chúng $< 2^{31}$

Khi chạy chương trình ta nhận được giá trị của $t0 = 0$

Dòng 2,3 ta lưu 2 số vào thanh ghi $s1, s2$

Dòng số 5 ta thiết lập trạng thái ban đầu (giả sử như 2 số đó ko bị tràn)

Dòng số 6, ta thực hiện lệnh cộng 2 số trên

Sử dụng lệnh xor để check xem 2 số này có cùng dấu với nhau hay không nếu không thì nhảy tới exit còn nếu có thì vẫn chạy như bình thường

Câu lệnh dòng số 9 ta mang so sánh nếu như $s3 < s1$ thì $t2$ sẽ bằng 1 (và đây là 2 số dương) nên khi $s3 < s1$ thì sẽ là tràn số vậy nếu như $t2 = 1$ thì kết quả đang bị tràn

Ta mang đi so sánh với số 0 dòng lệnh beq

Nếu như $t2 = 0$ tức là kết quả ko bị tràn thì ta kết thúc luôn chương trình

Còn nếu như kết quả bị tràn thì ta sẽ nhảy tới overflow

Lệnh dòng số 10 ta xét xem $s1$ có < 0 hay là không

Nếu như nó nhỏ hơn 0 vậy tức là 2 số đều < 0 (vì là điều kiện trên là xét 2 số cùng dấu với nhau)

Ta thực hiện câu lệnh bne để so sánh $t2$ và số 0 nếu như mà $t2$ khác 0 thì nhảy tới exit còn nếu bằng 0 thì chạy tiếp

Và khi mà $t2 = 0$ ở trường hợp 2 số âm thì kết quả nó đã bị tràn bởi vì ở trên đã xét điều kiện $s3 < s1$ nếu là số âm thì kết quả đã bị tràn và đã sai nên thanh ghi $t0$ chuyển về số 1

Kết thúc chương trình

Bài 2 :

```

1  .text
2  addi $s0, $0, 0x20200440
3
4  start:
5  srl $s1,$s0,24
6  # thực hiện lệnh dịch sang bên phải, các bit sẽ được lưu tại s1
7
8  andi $s3, $s0, 0xffffffff00
9  # xóa tất cả các bit khác f thành 0, các bit cùng f thì giữ nguyên
10
11 xori $s4,$s3, 0x00000011
12 #chuyển LSB thành 11
13
14 andi $s0 , $s0, 0x00000000
15 #xóa s0 về bằng 0

```

Run

Text Segment				
Bkpt	Address	Code	Basic	S
<input type="checkbox"/>	0x00400000	0x3c012020	lui \$1,0x00002020	2: addi \$s0, \$0, 0x20200440
<input type="checkbox"/>	0x00400004	0x34210440	ori \$1,\$1,0x00000440	
<input type="checkbox"/>	0x00400008	0x00018020	add \$16,\$0,\$1	
<input type="checkbox"/>	0x0040000c	0x00108e02	srl \$17,\$16,0x0000...	5: srl \$s1,\$s0,24
<input type="checkbox"/>	0x00400010	0x3c01ffff	lui \$1,0xffffffff	8: andi \$s3, \$s0, 0xffffffff00
<input type="checkbox"/>	0x00400014	0x3421ff00	ori \$1,\$1,0x0000ff00	
<input type="checkbox"/>	0x00400018	0x02019824	and \$19,\$16,\$1	
<input type="checkbox"/>	0x0040001c	0x3a740011	xori \$20,\$19,0x0000...	11: xori \$s4,\$s3, 0x00000011
<input type="checkbox"/>	0x00400020	0x32100000	andi \$16,\$16,0x0000...	14: andi \$s0 , \$s0, 0x00000000

ở bài 2

ta thực hiện lưu mssv vào s0

ta thực hiện dịch phải 24 bit tức là chúng ta dịch 6 số thì số 200440 sẽ bị mất và còn số 20 ở đầu chính là 2 số MSB

ta thực hiện so sánh từng bit với nhau nếu là f thì ta giữ nguyên các bit s0 còn nếu như bằng 0 thì ta đổi toàn bộ các bit đó về 0 và lưu vào s3

tiếp theo ta thực hiện chuyển LSB thành số 1

tức là nó đang ở số 0 ta thực hiện lệnh XOR nếu là 0 thì giữ nguyên các bit còn nếu là 1 thì ta thực hiện chuyển từ 0 thành 1 bit

xóa s0 về 0 thì ta có kcas nhiều cách như có thể dùng add, nhưng do em dùng andi và đang thích dùng andi thì em sai luôn andi để dùng andi xử lí bài trên :3 nên em cho so sánh với số 0 và tất cả chuyển về số 0

Bài 3

Ý A

Ta cần phân tích đầu bài

Ta thực hiện gán $s0 = \text{abs}(s1)$

Vậy có 2 trường hợp có thể xảy ra

Nếu như $s1 < 0$ thì ta thực hiện lệnh trừ với số 0, $0-s1$ và lưu vào $s0$

Trường hợp còn lại $s1 > 0$ ta thực hiện lệnh gán luôn giá trị

Ta có chương trình sau

Code

```
.text
addi $s1, $0, -440
start:
# đề bài đặt ra yêu cầu lưu s0 = abs(s1)
# yêu cầu đặt ra ta sẽ xét xem s0 nó lớn hơn hay nhỏ hơn 0
# nếu <0 ta dùng lệnh sub đổi dấu, còn nếu >0 ta thực hiện phép gán luôn

slt $s2, $s1, $0
bne $s2, $0, code
j code2
code:
sub $s1, $0, $s1
code2:

add $s0, $s1, $0
```

Run

Text Segment				
Bkpt	Address	Code	Basic	
<input type="checkbox"/>	0x00400000	0x2011fe48	addi \$17,\$0,-440	2: addi \$s1, \$0, -440
<input type="checkbox"/>	0x00400004	0x0220902a	slt \$18,\$17,\$0	8: slt \$s2, \$s1, \$0
<input type="checkbox"/>	0x00400008	0x16400001	bne \$18,\$0,1	9: bne \$s2, \$0, code
<input type="checkbox"/>	0x0040000c	0x08100005	j 0x00400014	10: j code2
<input type="checkbox"/>	0x00400010	0x00118822	sub \$17,\$0,\$17	12: sub \$s1,\$0,\$s1
<input type="checkbox"/>	0x00400014	0x02208020	add \$16,\$17,\$0	15: add \$s0, \$s1, \$0

Tương tự như đã phân tích ta dùng lệnh `slt` và `bne` để rẽ nhánh nếu như $s1 < 0$ thì nhảy tới để dùng lệnh `sub` còn nếu $s1 > 0$ thì ta thực hiện nhảy tới code 2 để thực hiện lưu ngay lập tức

Bài 3 ý B

Ta thực hiện lệnh addu cộng s1 với \$0 thì ta sẽ được giá trị của lệnh gán

```
.text
addi $s1, $0, 440
start:
addu $s0, $s1, $0
```

Run

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x201101b8	addi \$17,\$0,440	2: addi \$s1, \$0, 440
<input type="checkbox"/>	0x00400004	0x02208021	addu \$16,\$17,\$0	4: addu \$s0, \$s1, \$0

Bài 3 ý C

```
B0yB.asm  B0yC.asm
1  .text
2  addu $s1, $0, 7
3  start:
4  nor $s0, $s1, $0
```

Ta thực hiện lệnh gán vào s1 và thực hiện nor với s1 để trở thành s0

Thì sẽ chuyển số s1 thành số nhị phân

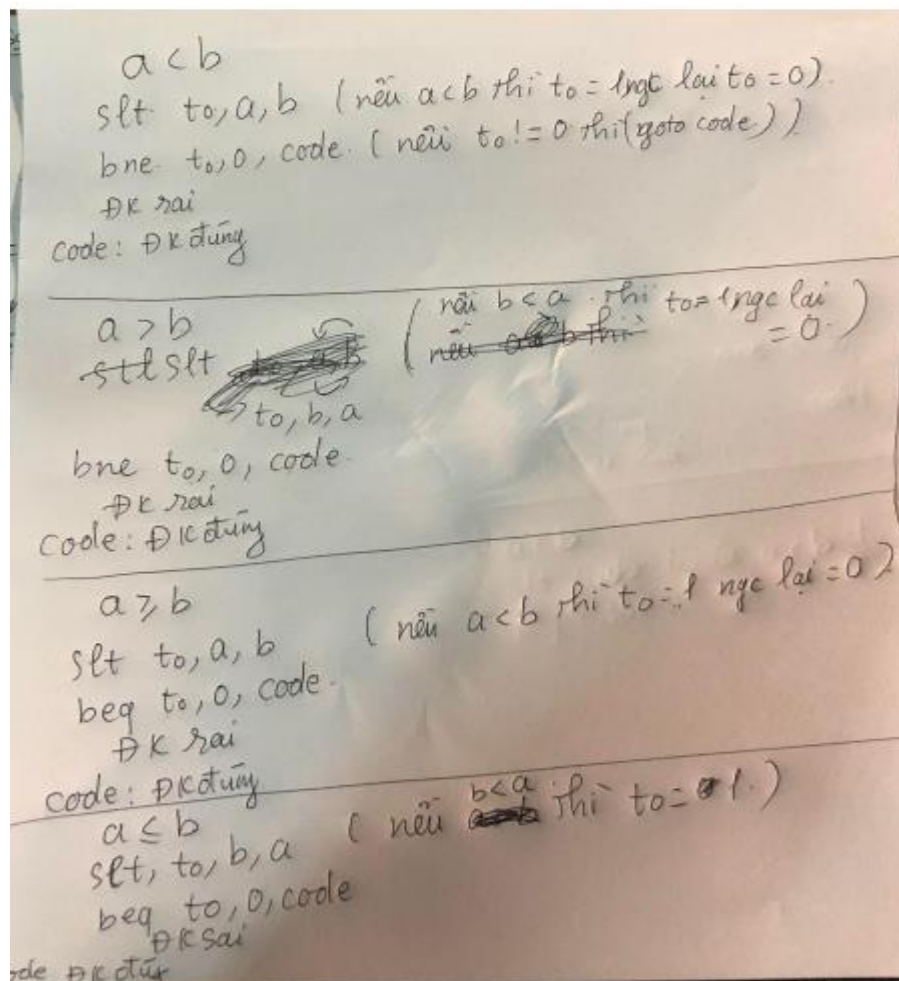
Thực hiện đảo số 1 thành 0, 0 thành 1

Và lưu lại trong s0

Bài 3 ý D

Trước khi tính toán

Phân tích và chia các trường hợp



```
B3yD.asm
1  .text
2  addi $s1, $0, 12
3  addi $s2, $0, 23
4  label:
5  slt $t1, $s2, $s1
6  beq $t1, $0, code
7  j next
8  code:
9  j label
10 next:
11
```

Đầu tiên ta lấy giá trị của 2 số gán vào $s1, s2$

Ta thực hiện lệnh so sánh `slt` và `beq` để rẽ nhánh

Nếu như $s1 > s2$ thì sẽ kết thúc chương trình

Còn nếu $s1 \leq s2$ thì sẽ quay trở lại chương trình if

Nếu như $s1 \leq s2$ thì vòng lặp mãi mãi

Bài 4

```

1  .text
2  addi $s6, $0, 20200440
3  addi $s7, $0, 2127283214
4  start:
5  li $t0,0 #Thiết lập trạng thái mặc định
6  addu $s3,$s1,$s2 # s3 = s1 + s2
7  slt $s1, $s6, $0      # so sánh a và 0
8  slt $s2, $s7, $0      # so sánh b và 0
9  beq $s1, $s2, next
10 li $t0, 1
11 j end
12
13 next:
14 add $s5, $s6, $s7 # s5 = a+b
15 slt $s3, $s5, $0
16 bne $s3, $s1, endl
17 li $t0, 0
18 j end
19 endl:
20 li $t0, 1
21 end:

```

Run

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	4194304	0x3c010134	lui \$1,308	2: addi \$s6, \$0, 20200440
<input type="checkbox"/>	4194308	0x34213bf8	ori \$1,\$1,15352	
<input type="checkbox"/>	4194312	0x0001b020	add \$22,\$0,\$1	
<input type="checkbox"/>	4194316	0x3c017ecb	lui \$1,32459	3: addi \$s7, \$0, 2127283214
<input type="checkbox"/>	4194320	0x3421c40e	ori \$1,\$1,50190	
<input type="checkbox"/>	4194324	0x0001b820	add \$23,\$0,\$1	
<input type="checkbox"/>	4194328	0x24080000	addiu \$8,\$0,0	5: li \$t0,0 #Thiết lập trạng thái mặc định
<input type="checkbox"/>	4194332	0x02329821	addu \$19,\$17,\$18	6: addu \$s3,\$s1,\$s2 # s3 = s1 + s2
<input type="checkbox"/>	4194336	0x02c0882a	slt \$17,\$22,\$0	7: slt \$s1, \$s6, \$0 # so sánh a và 0
<input type="checkbox"/>	4194340	0x02e0902a	slt \$18,\$23,\$0	8: slt \$s2, \$s7, \$0 # so sánh b và 0
<input type="checkbox"/>	4194344	0x12320002	beq \$17,\$18,2	9: beq \$s1, \$s2, next
<input type="checkbox"/>	4194348	0x24080001	addiu \$8,\$0,1	10: li \$t0, 1
<input type="checkbox"/>	4194352	0x08100013	j 4194380	11: j end
<input type="checkbox"/>	4194356	0x02d7a820	add \$21,\$22,\$23	14: add \$s5, \$s6, \$s7 # s5 = a+b
<input type="checkbox"/>	4194360	0x02a0982a	slt \$19,\$21,\$0	15: slt \$s3, \$s5, \$0
<input type="checkbox"/>	4194364	0x16710002	bne \$19,\$17,2	16: bne \$s3, \$s1, endl
<input type="checkbox"/>	4194368	0x24080000	addiu \$8,\$0,0	17: li \$t0, 0
<input type="checkbox"/>	4194372	0x08100013	j 4194380	18: j end
<input type="checkbox"/>	4194376	0x24080001	addiu \$8,\$0,1	20: li \$t0, 1

Đầu tiên ta cần cho điều kiện 2 số nếu <0 để t1, t1 = 1 ngược lại = 0

Sau đó ta xác định xem 2 số cùng dấu hay ko nếu ko cùng dấu thì ta nhảy tới next để kết thúc

Còn nếu ko thì ta tổng lại và so sánh với 1 trong 2 số xem có cùng dấu hay không

(vì 2 số đang cùng dấu nên ta chỉ cần so sánh 1 số)

Vậy ta xét nếu cùng dấu thì giá trị bằng 0 còn nếu khác dấu thì sẽ là 1

Và kết luận =0 là cùng dấu và =1 thì khác dấu

Bài 5

Chương trình

Code

```

1  #bài toán sử dụng nhân với 1 số lũy thừa nhỏ của 2
2  #vậy ta cần phân tích bài toán
3  .text
4      li $s1,5 #giá trị cần tính lũy thừa 2
5      li $s2,1 #giá trị khởi chạy, bước nhảy
6      add $s3, $s1, $0 #lưu giá trị s3 = s1
7  loop:
8      add $s1, $s1, $s3 # thực hiện mỗi vòng lặp
9      addi $s2, $s2, 1 #mỗi vòng lặp thực hiện tăng biến đếm
10     bne $s2, $s3, loop
11     j next
12 next:

```

Run

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	4194304	0x24110005	addiu \$17,\$0,5	4: li \$s1,5 #giá trị cần tính lũy thừa 2
<input type="checkbox"/>	4194308	0x24120001	addiu \$18,\$0,1	5: li \$s2,1 #giá trị khởi chạy, bước nhảy
<input type="checkbox"/>	4194312	0x02209820	add \$19,\$17,\$0	6: add \$s3, \$s1, \$0 #lưu giá trị s3 = s1
<input type="checkbox"/>	4194316	0x02338820	add \$17,\$17,\$19	8: add \$s1, \$s1, \$s3 # thực hiện mỗi vòng lặp
<input type="checkbox"/>	4194320	0x22520001	addi \$18,\$18,1	9: addi \$s2, \$s2, 1 #mỗi vòng lặp thực hiện tăng biến đếm
<input type="checkbox"/>	4194324	0x1653ffff	bne \$18,\$19,-3	10: bne \$s2, \$s3, loop
<input type="checkbox"/>	4194328	0x08100007	j 4194332	11: j next

Đầu tiên ta cần phân tích chương trình

Ta khởi tạo giá trị cần tính lưu vào s1

Ta khởi tạo step để bước nhảy

Giá thuật toán sẽ là

$$2^2 = 2+2 \mid 6^2 = 6+6+6+6+6+6$$

Vậy ta cần 1 bước nhảy đếm từ 1 đến n

Và mỗi vòng lặp ta thực hiện cộng số đó với số nguyên thủy của nó tức là n

$$\text{Vậy } n^2 = n + \dots + n \text{ (n số n)}$$

Ta cần 1 câu lệnh để kiểm soát vòng lặp, nếu như $i \neq n$ thì ta thực hiện quay lại còn nếu $=$ thì ta thực hiện thoát chương trình