Báo cáo thực hành KTMT tuần 4

Phạm Thành Lập – 20215076

Assignment 1

\$s1 âm, \$s2 âm và \$s3 tràn giá trị

```
0x00000000
 .text
                                             $at
$v0
                                                                             0x00000000
start:
         1 i
                   $s1, 0x9fffffff
                                                                             0x00000000
                  $s2, 0x9fffffff
         1 i
                                             $a3
$t0
                                                                             0x00000001
         1i
                  $t0, 0
                                                                             0x00000000
         addu $s3, $s1, $s2
                                                                             0x00000000
                 $t1, $s1, $s2
         xor
                                             $t5
$t6
                                                                             0x00000000
         bltz
                  $t1, EXIT
                  $t2, $s3, $s1
         slt
                  $s1, NEGATIVE
         bltz
                                             $s3
$s4
                                                                             0x00000000
                  $t2, $zero, EXIT
         beq
                                                                             0x00000000
                  OVERFLOW
         j
                                                                             0x00000000
NEGATIVE:
                                             $t9
$k0
                                                                   25
                                                                             0x00000000
                 $t2, $zero, EXIT
         bne
                                             $k1
$gp
OVERFLOW:
                                                                             0x00000000
                  $t0, 1
         li .
                                                                             0x00400038
EXIT:
```

- Khởi tao \$t0 = 0; \$s1 = 0x9fffffff; \$s2 = 0x9fffffff
- Lệnh addu giúp có thể lưu giá trị khi kết quả phép cộng nằm ngoài vùng giá trị của thanh ghi, sau lệnh addu ta thực hiện lệnh xor 2 thanh ghi \$s1 và \$s2 để biết 2 thanh ghi này cùng dấu hay không, nếu cùng dấu dương hoặc cùng âm thì sẽ trả về giá trị thanh ghi \$t1 là âm ngược lại sẽ trả về giá trị dương, nếu 2 thanh ghi này khác dấu thì lệnh btlz sẽ nhảy về thẻ EXIT kết thúc chương trình và trả về kết quả thanh ghi \$t0 là 0 (giá trị tổng không bị tràn) ngược lại chương trình sẽ so sánh giá trị thanh ghi \$s3 và \$1 hay không nếu \$s1 mang dấu âm thì lệnh bltz thứ 2 sẽ nhảy đến thé NEGATIVE và so sánh thẻ \$t2 với 0 nếu \$s3 tràn giá trị thì nó sẽ lớn hơn \$s1 nếu \$s3 bé hơn thì không tràn và nhảy đến thẻ EXIT còn nếu \$s1 dương mà \$t2 = 0 (\$s3 > \$s1) thì sẽ nhảy đến thẻ EXIT còn ngược lại thì sẽ nhảy đến thẻ OVERFLOW và thiết lập lại thanh ghi \$t0 = 1 từ đó kết luật được rằng tổng 2 toán hạng bị tràn giá trị.
- Sư thay đổi giá tri thanh ghi

Trạng thái	\$s3	\$t0	\$t1	\$t2	рс
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000	0x00400000
Sau lệnh	0x3ffffffe	-	-	-	0x00400018
addu					
Sau lệnh xor	-	-	-	-	0x0040001c
Sau lệnh slt	-	-	-	-	0x00400020
Sau lệnh btlz	-	-	-	-	0x0040002c
nhảy đến thẻ					(địa chỉ của
NEGATIVE					thẻ
					NEGATIVE)
Sau lệnh li	-	0x0000001	ı	ı	0x00400038

• \$s1 âm, \$s2 âm, \$s3 không tràn giá trị

				Name	Number	Value
.text				\$zero	0	0x0000000
start:				Şat	1	0x9fff000
Start:				\$ v 0	2	0x0000000
	1i	\$ 6.1	0x9fffffff	\$v1	3	0x0000000
	11	A2T,	ONSTITUTE	\$a0 \$a1	4	0x00000000
	1i	\$52	0xffffffff	\$a2	6	0x00000000
		452,	ONLLLLLLL	\$a3	7	0x00000000
	li	\$t0,	0	\$t0	8	0x0000000
				\$t1	9	0x60000000
	addu	\$s3,	\$s1, \$s2	\$t2	10	0x0000000
		•	•	\$t3	11	0x0000000
	xor	Şt1,	\$s1, \$s2	\$t4	12	0x00000000
			· · · · ·	\$t5	13	0x0000000
	bltz	Ştl,	EXIT	\$t6	14	0x00000000
	. .	0.0		\$t7 \$s0	15 16	0x00000000
	slt	Şt2,	\$s3, \$s1	Şs0 Ss1	17	0x9fffffff
	1-1	¢ - 1	NECAMINE	\$s2	18	0xffffffff
	bltz	SST,	NEGATIVE	Şs3	19	0x9ffffffe
	boa	¢+2	CZONO EVID	\$54	20	0x0000000
	beq	764,	\$zero, EXIT	\$s5	21	0x0000000
	i	OVER	EI OM	\$s6	22	0x0000000
	J	OVER	LLOW	\$s7	23	0x00000000
NEGATI	VE •			\$t8	24	0x0000000
IVEGATI	V L.			\$t9	25	0x0000000
	bne	St2	\$zero, EXIT	\$k0 \$k1	26	0x00000000
	2110	+01/	+2020, Biiii		27	0x00000000 0x10008000
OVERFL	OW:			\$gp \$sp	29	0x7fffeff
				\$fp	30	0x00000000
	li	\$t0,	1	\$ra	31	0x00000000
				pc		0x00400034
EXIT:				hi		0x0000000
				10		0x0000000

- Khởi tạo \$s1 = 0x9fffffff; \$s2 = 0xffffffff
- Có \$s1+\$s2 không tràn giá trị thanh ghi do \$s1 và \$s2 cùng dấu âm nên giá trị thanh ghi \$s3 sẽ nhỏ hơn \$s1 khiến cho giá trị thanh ghi \$t2 = 1 sau lệnh btlz thứ 2 chương trình lập tức nhảy đến thẻ NEGATIVE so sánh giá trị \$t2 với 0 và nhảy đến thẻ EXIT kết thúc chương trình và giả về giá trị \$t0 = 0 tức là không bị tràn giá trị thanh ghi
- Sự thay đổi giá trị thanh ghi

Trạng thái	\$s3	\$t0	\$t1	\$t2	рс
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000	0x00400000
Sau lệnh	0x9ffffffe	-	-	-	0x00400018
addu					
Sau lệnh xor	-	•	0x60000000	•	0x0040001c
Sau lệnh slt	-	-	-	0x0000001	0x00400020
Sau lệnh btlz	-	-	-	-	0x0040002c
nhảy đến thẻ					(địa chỉ của
NEGATIVE					thẻ
					NEGATIVE)
Sau lệnh	-	-	-	-	0x00400034
bne					

• \$s1 dương, \$s2 dương và \$s3 tràn giá trị

+		Name	Number	Value
.text		Şzero	0	0x00000000
start:		\$at	1	0x7fff0000
Start:		\$ v 0	2	0x00000000
li li	\$s1, 0x7fffffff	\$v1	3	0x00000000
11	PSI, UX/LLLLLL	\$a0	4	0x00000000
1i	\$s2, 0x456	\$a1	5	0x00000000
11	752, 0X430	\$a2	6	0x00000000
1i	\$t0, 0	\$a3	7	0x00000000
11	\$CO, O	\$t0	8	0x00000001
addu	\$s3, \$s1, \$s2	\$t1	9	0x7ffffba9 0x00000001
addu	755, 751, 752	\$t2 \$t3	10	
xor	\$t1, \$s1, \$s2	\$t3 \$t4	11	0x00000000 0x00000000
YOL	QCI, QSI, QSZ	St5	13	0x00000000
bltz	\$t1, EXIT	\$t6	14	0x00000000
DICZ	QCI, D211	\$t7	15	0x00000000
slt	\$t2, \$s3, \$s1	\$50	16	0x00000000
510	702, 700, 701	\$s1	17	0x7fffffff
bltz	\$s1, NEGATIVE	\$s2	18	0x00000456
	,,	\$s3	19	0x80000455
beq	\$t2, \$zero, EXIT	\$s4	20	0x00000000
•	+02, +2020, 2011	\$55	21	0x00000000
i	OVERFLOW	\$s6	22	0x00000000
,		\$s7	23	0x00000000
NEGATIVE:		\$t8	24	0x00000000
		\$t9	25	0x00000000
bne	\$t2, \$zero, EXIT	\$k0	26	0x00000000
		\$k1	27	0x00000000
OVERFLOW:		\$gp	28	0x10008000
		\$sp	29	0x7fffeffc
l li	\$t0, 1	\$fp	30	0x00000000
		\$ra	31	0x00000000
EXIT:		pc hi		0x00400034 0x00000000
		10		0x00000000
		10		0x0000000

- Khởi tạo \$s1 = 0x7ffffff; \$s2 = 0x456
- Có \$s1+\$s2 tràn giá trị thanh ghi \$s1 và \$s2 cùng dấu nên khi chương trình thực hiện lệnh xor sẽ trả về \$t1 giá trị dương và do \$s1 dương nên chương trình sẽ so sánh \$s3 và \$s1 do \$s3 bị tràn nên giá trị sẽ bé hơn \$s1 nên chương trình lập tức nhảy đến thẻ OVERFLOW để thiết lập giá trị \$t0 = 1 và kết luận tổng 2 toán hạng bị tràn giá trị
- Sự thay đổi giá trị thanh ghi

Trạng thái	\$s3	\$t0	\$t1	\$t2	рс
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000	0x00400000
Sau lệnh addu	0x80000455	-	-	-	0x00400018
Sau lệnh xor	-	0x7ffffba9	-	-	0x0040001c
Sau lệnh slt	-	-	-	0x0000001	0x00400020
Sau lệnh j	-	-	-	-	0x00400030
OVERFLOW					(địa chỉ của
					thẻ
					OVERFLOW)
Sau lệnh li	-	0x0000001	-	-	0x00400034

• \$s1 dương, \$s2 dương và \$s3 không tràn giá trị

.text				Name	Number	Value
start:				\$zero	0	0x0000000
Deare.			-	\$at	1	0x0000000
	li	C-1	0x123	\$v0	2	0x0000000
	11	751,	07123	\$v1	3	0x0000000
	li	5 = 2	0x456	\$a0	4	0x0000000
	11	432,	07430	Şa1 Şa2	5	0x0000000 0x0000000
	li	\$t0,	0	Şa2 Sa3	6	0x0000000
	11	ŞCO,	0	\$t0	8	0x0000000
	addu	602	\$s1, \$s2	St1	9	0x0000057
	audu	755,	49I, 49Z	St2	10	0x0000000
	xor	C+1	\$s1, \$s2	St3	11	0x0000000
	XOL	ąιI,	95I, 95Z	\$t4	12	0x0000000
	bltz	C+1	ESTE	\$t5	13	0x0000000
	DICZ	ąιI,	EXIT	\$t6	14	0x0000000
	-34	CHO	0-0 0-1	\$t7	15	0x0000000
	slt	Şt∠,	\$s3, \$s1	\$s0	16	0x0000000
	1-2	0-1	NE CARETTE	\$s1	17	0x0000012
	bltz	ŞSI,	NEGATIVE	\$s2	18	0x0000045
		0.0		\$s3	19	0x0000057
	beq	Şt2,	\$zero, EXIT	\$s4	20	0x0000000
				\$s5	21	0x0000000
	j	OVER:	FLOW	\$s6 \$s7	22	0x0000000 0x0000000
				\$57 St8	24	0x0000000
NEGATI	VE:			St9	25	0x0000000
		4		\$k0	26	0x00000000
	bne	Şt2,	\$zero, EXIT	Sk1	27	0x0000000
				Şgp	28	0x1000800
OVERFLO	οW :			\$sp	29	0x7fffeff
			-	\$fp	30	0x0000000
	li	\$t0,	1	Şra	31	0x0000000
				pc		0x0040003
EXIT:				hi		0x0000000
				10		0x0000000

- Khởi tạo \$s1 = 0x123; \$s2 = 0x456
- Có \$s1+\$s2 không tràn giá trị thanh ghi \$s1 và \$s2 cùng dấu nên khi chương trình thực hiện lệnh xor sẽ trả về \$t1 giá trị âm và do \$s1 dương nên chương trình sẽ so sánh \$s3 và \$s1 do \$s3 không bị tràn nên giá trị sẽ lớn hơn \$s1 nên chương trình lập tức nhảy đến thẻ EXIT do đó giá trị \$t0 = 0 và kết luận tổng 2 toán hạng không bị tràn giá trị
- Sự thay đổi giá trị thanh ghi

Trạng thái	\$s3	\$t0	\$t1	\$t2	рс
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000	0x00400000
Sau lệnh	0x00000579	-	-	-	0x00400018
addu					
Sau lệnh xor	•	0x00000575	-	•	0x0040001c
Sau lệnh slt	-	-	-	-	0x00400020
Sau lệnh	-	-	-	-	0x00400030
beq nhảy					
đến thể					
EXIT					

• \$s1 âm, \$s2 dương

.text				\$zero	0	0x00000000
. cext				Şat	1	0x9fff0000
start:				\$ v 0	2	0x00000000
Start.				\$v1	3	0x00000000
	1i	Se1	0x9fffffff	\$a0	4	0x00000000
	11	431,	ONSTITUTE	\$a1	5	0x00000000
	li	592	0x569	\$a2	6	0x00000000
	11	7.52,	02303	\$a3	7	0x00000000
	li	\$t0,	0	\$t0	8	0x00000000 0x9ffffa96
	11	400,	0	\$t1 \$t2	10	0x91111496
	addu	553	\$s1, \$s2	\$t3	11	0x00000000
	adda	455,	431, 432	\$t4	12	0x00000000
	xor	\$±1	\$s1, \$s2	\$t5	13	0x00000000
	AUL	YCI,	451, 45E	\$t6	14	0x00000000
	bltz	St1	EXIT	\$t7	15	0x00000000
	2202	+01/	21111	\$50	16	0x00000000
	slt	Št2.	\$s3, \$s1	\$s1	17	0x9fffffff
	510	+01/	455, 451	\$s2	18	0x00000569
	bltz	\$s1.	NEGATIVE	\$s3	19	0xa0000568
		+51/		\$s4	20	0x00000000
	beq	Št2.	\$zero, EXIT	\$s5	21	0x00000000
	200	+01/	42020, 2011	\$56	22	0x00000000
	i	OVERI	FLOW	\$s7	23	0x00000000
	-			\$t8	24	0x00000000
NEGATI	VE:			\$t9	25	0x00000000
				\$k0	26	0x00000000
	bne	St2.	\$zero, EXIT	\$k1	27	0x00000000
		,,	,,	\$gp	28	0x10008000
OVERFIC	ο <i>W</i> :			Şsp	30	0x7fffeffc
				\$fp \$ra	31	0x00000000 0x00000000
	li	\$t0,	1	pc	31	0x00400034
		/		hi		0x000000000
EXIT:				10		0x00000000

- Khởi tạo \$s1 = 0x9ffffff; \$s2 = 0x569.
- Có \$s1+\$s2 không tràn giá trị thanh ghi mà \$s1 và \$s2 trái dấu nên khi chương trình thực hiện lệnh xor sẽ trả về giá trị âm lên lập tưc nhảy đến thẻ EXIT kết thúc chương trình và giả về giá trị \$t0 = 0 tức là không bị tràn giá trị thanh ghi
- Sự thay đổi giá trị thanh ghi

Trạng thái	\$s3	\$t0	\$t1	\$t2	рс
Ban đầu	0x00000000	0x00000000	0x00000000	0x00000000	0x00400000
Sau lệnh	0xa0000568	-	-	-	0x00400018
addu					
Sau lệnh xor	-	-	0x9ffffa96	-	0x0040001c
Sau lệnh slt	-	•	ı	•	0x00400020
Sau lệnh	-	-	-	-	0x00400034
beq nhảy					(địa chỉ của
đến thẻ EXIT					the EXIT)

Assignment 2

• Extract MSB of \$s0

.text

li \$s0, 0x12345678 andi \$t0, \$s0, 0xff000000

\$t0	8	0x12000000
\$t1	9	0x00000000
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x12345678

Clear LSB of \$s0

.text

li \$s0, 0x12345678 andi \$t1, \$s0, 0xffffff00

\$t1	9	0x12345600
\$t2	10	0x00000000
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x12345678

• Set LSB of \$s0 (bits 7 to 0 are set to 1)

.text

li \$s0, 0x12345678 ori \$t2, \$s0, 0x000000ff

\$t2	10	0x123456ff
\$t3	11	0x00000000
\$t4	12	0x00000000
\$t5	13	0x00000000
\$t6	14	0x00000000
\$t7	15	0x00000000
\$s0	16	0x12345678

• Clear \$s0 (\$s0 = 0, must use logical instructions)

.text

li \$s0, 0x12345678 and \$s0, \$s0, \$zero

\$v1 3 0x000000000 \$a0 4 0x000000000 \$a1 5 0x00000000 \$a2 6 0x00000000 \$t0 8 0x00000000 \$t1 9 0x00000000 \$t2 10 0x00000000 \$t3 11 0x00000000 \$t4 12 0x00000000 \$t5 13 0x00000000 \$t6 14 0x00000000 \$t7 15 0x00000000 \$s0 16 0x00000000 \$s1 17 0x00000000 \$s2 18 0x00000000 \$s3 19 0x00000000 \$s4 20 0x00000000 \$s5 21 0x00000000 \$s6 22 0x00000000 \$s7 23 0x00000000 \$t8 24 0x00000000 \$k0 26 0x00000000 \$k1 27 0x00000000 \$sp			
\$v0 2 0x000000000 \$v1 3 0x00000000 \$a0 4 0x00000000 \$a1 5 0x00000000 \$a2 6 0x00000000 \$a3 7 0x00000000 \$t0 8 0x00000000 \$t1 9 0x00000000 \$t2 10 0x00000000 \$t4 12 0x00000000 \$t5 13 0x00000000 \$t6 14 0x00000000 \$t7 15 0x00000000 \$s1 17 0x00000000 \$s2 18 0x00000000 \$s3 19 0x00000000 \$s4 20 0x00000000 \$s4 20 0x00000000 \$s5 21 0x00000000 \$s6 22 0x00000000 \$s6 22 0x00000000 \$t8 24 0x00000000 \$k0 26 0x00000000 \$k1 <th>\$zero</th> <th>0</th> <th>0x00000000</th>	\$zero	0	0x00000000
\$v1 3 0x000000000 \$a0 4 0x000000000 \$a1 5 0x00000000 \$a2 6 0x00000000 \$t0 8 0x00000000 \$t1 9 0x00000000 \$t2 10 0x00000000 \$t3 11 0x00000000 \$t4 12 0x00000000 \$t5 13 0x00000000 \$t6 14 0x00000000 \$t7 15 0x00000000 \$s0 16 0x00000000 \$s1 17 0x00000000 \$s2 18 0x00000000 \$s3 19 0x00000000 \$s4 20 0x00000000 \$s5 21 0x00000000 \$s5 21 0x00000000 \$s6 22 0x00000000 \$t8 24 0x00000000 \$k0 26 0x00000000 \$k1 27 0x00000000 \$sp	\$at	1	0x12340000
\$a0	\$ v 0	2	0x00000000
\$a1	\$v1	3	0x00000000
\$a2 6 0x000000000 \$a3 7 0x0000000000000000000000000000000000	\$a0	4	0x00000000
\$a3	\$a1	5	0x00000000
\$t0 8 0x00000000 \$t1 9 0x00000000 \$t2 10 0x00000000 \$t3 11 0x00000000 \$t4 12 0x00000000 \$t5 13 0x00000000 \$t6 14 0x00000000 \$s0 16 0x00000000 \$s1 17 0x00000000 \$s2 18 0x00000000 \$s3 19 0x00000000 \$s4 20 0x00000000 \$s5 21 0x00000000 \$s6 22 0x00000000 \$t8 24 0x00000000 \$t9 25 0x00000000 \$k0 26 0x00000000 \$k1 27 0x00000000 \$pg 28 0x10008000 \$pg 28 0x10008000 \$pg 29 0x7fffeffc \$pg 30 0x00000000 \$ra 31 0x000000000	\$a2	6	0x00000000
\$t1 9 0x00000000 \$t2 10 0x00000000 \$t3 11 0x00000000 \$t4 12 0x00000000 \$t5 13 0x00000000 \$t6 14 0x00000000 \$s0 16 0x00000000 \$s1 17 0x00000000 \$s2 18 0x00000000 \$s3 19 0x00000000 \$s4 20 0x00000000 \$s5 21 0x00000000 \$s6 22 0x00000000 \$t8 24 0x00000000 \$t9 25 0x00000000 \$k0 26 0x00000000 \$k1 27 0x00000000 \$sp 28 0x10008000 \$sp 29 0x7fffeffc \$fp 30 0x00000000 \$ra 31 0x00000000 hi 0x000000000	\$a3	7	0x00000000
\$t2	\$t0	8	0x00000000
\$t4	\$t1	9	0x00000000
\$t4	\$t2	10	0x00000000
\$t5	\$t3	11	0x00000000
\$t6	\$t4	12	0x00000000
\$t7	\$t5	13	0x00000000
\$s0	\$t6	14	0x00000000
\$s1	\$t7	15	0x00000000
\$s2	\$s0	16	0x00000000
\$s3	\$s1	17	0x00000000
\$s4	\$s2	18	0x00000000
\$s5	\$s3	19	0x00000000
\$s6	\$s4	20	0x00000000
\$s7 23 0x00000000 \$t8 24 0x00000000 \$t9 25 0x00000000 \$k0 26 0x00000000 \$k1 27 0x00000000 \$gp 28 0x10008000 \$sp 29 0x7fffeffc \$fp 30 0x00000000 \$ra 31 0x00000000 pc 0x00000000000000000000000000000000000	\$s5	21	0x00000000
\$t8 24 0x00000000 \$t9 25 0x00000000 \$k0 26 0x00000000 \$k1 27 0x00000000 \$gp 28 0x10008000 \$sp 29 0x7fffeffc \$fp 30 0x00000000 \$ra 31 0x00000000 pc 0x0040000c hi 0x000000000	\$s6	22	0x00000000
\$t9 25 0x00000000 \$k0 26 0x00000000 \$k1 27 0x00000000 \$gp 28 0x10008000 \$sp 29 0x7fffeffc \$fp 30 0x00000000 \$ra 31 0x00000000 pc 0x0040000c hi 0x000000000	\$s7	23	0x00000000
\$k0 26 0x00000000 \$k1 27 0x00000000 \$gp 28 0x10008000 \$sp 29 0x7fffeffc \$fp 30 0x00000000 \$ra 31 0x00000000 pc 0x0040000c hi 0x000000000	\$t8	24	0x00000000
\$k1 27 0x00000000 \$gp 28 0x10008000 \$sp 29 0x7fffeffc \$fp 30 0x00000000 \$ra 31 0x00000000 pc 0x0040000c hi 0x000000000	\$t9	25	0x00000000
\$gp 28 0x10008000 \$sp 29 0x7fffeffc \$fp 30 0x00000000 \$ra 31 0x00000000 pc 0x0040000c hi 0x000000000	\$k0	26	0x00000000
\$sp 29 0x7fffeffc \$fp 30 0x00000000 \$ra 31 0x00000000 pc 0x0040000c hi 0x00000000	\$k1	27	0x00000000
\$sp 29 0x7fffeffc \$fp 30 0x00000000 \$ra 31 0x00000000 pc 0x00040000c hi 0x000000000	\$gp	28	0x10008000
\$fp 30 0x00000000 \$ra 31 0x00000000 pc 0x0040000c hi 0x000000000	\$sp	29	0x7fffeffc
pc 0x0040000c hi 0x00000000	\$fp	30	0x00000000
hi 0x00000000	\$ra	31	0x00000000
	pc		0x0040000c
lo 0x00000000	hi		0x00000000
	10		0x00000000

Assignment 3

a) abs \$s0, \$s1

.text

slt \$t0, \$s1, \$zero beqz \$t0, ABS

sub \$s1, \$zero, \$s1

ABS:

add \$s0, \$s1, \$zero

b) move \$s0, \$s1

.text

add \$s0, \$s1, \$zero

c) not \$s0, \$s1

.text

sub \$t0, \$zero, \$s1

li \$t1, 1

sub \$s0, \$t0, \$t1

d) ble \$s1,\$s2,label

.text

sle \$t0, \$s1, \$s2 bnez \$t0, label

label:

Assignment 4

#Result

Registers	Coproc 1	Coproc 0	
Name		Number	Value
\$zero		0	0x00000000
\$at		1	0x9fff0000
\$ v 0		2	0x00000000
\$v1		3	0x00000000
\$a0		4	0x00000000
\$a1		5	0x00000000
\$a2		6	0x00000000
\$a3		7	0x00000000
\$t0		8	0x0000001
\$t1		9	0xa0000001
\$t2		10	0x00000000
\$t3		11	0x00000000
\$t4		12	0x00000000
\$t5		13	0x00000000
\$t6		14	0x00000000
\$t7		15	0x00000000
\$s0		16	0x00000000
\$s1		17	0x9fffffff
\$s2		18	0x9fffffff
\$s3		19	0x3ffffffe
\$s4		20	0x00000000
\$s5		21	0x00000000
\$s6		22	0x00000000
\$s7		23	0x00000000
\$t8		24	0x00000000
\$t9		25	0x00000000
\$k0		26	0x00000000
\$k1		27	0x00000000
\$gp		28	0x10008000
\$sp		29	0x7fffeffc
\$fp		30	0x00000000
\$ra		31	0x00000000
pc			0x00400024
hi			0x00000000
10			0x00000000

.text

start:

li \$s1, 0x9fffffff

li \$s2, 0x9ffffff

li \$t0, 0

addu \$s3, \$s1, \$s2 #s3 = s1 + s2

xor \$t1, \$s1, \$s2 #kiem tra s1 va s2 cung hay khac dau

bltz \$t1, EXIT

xor \$t1, \$s3, \$s1 #kiem tra s1 va s3 cung hay khac dau

bltz \$t1, OVERFLOW

OVERFLOW:

li \$t0, 1

EXIT:

Assignment 5

#Result

\$s0	16	4
\$s1	17	3
\$s2	18	16
\$s3	19	48
\$s4	20	0
\$s0 \$s1 \$s2 \$s3 \$s4 \$s5 \$s6	21	0
\$s6	22	0

#Text assignment

.text

li \$s0, 1

li \$t0, 1

li \$s1, 3

li \$s2, 16

LOOP:

t1, t0, t0, t1 = t0 * s0 (t0 luon bang 1)

beq \$t1, \$s2, MUL # if t1 < s2 thuc hien lenh nhan

bgt \$t1, \$s2, EXIT #t1 > s2 tuc s2 khong phai luy thua cua 2 break

addi \$s0, \$s0, 1 # tang gia tri dich bit len 1

j LOOP

MUL:

sllv \$s3, \$s1, \$s0 #s3 = s1 * s2

EXIT: