BÁO CÁO THỰC HÀNH KIẾN TRÚC MÁY TÍNH TUẦN 4

Họ và tên: Nguyễn Quý Đức MSSV: 20235682

Mã HP: IT3280 Mã lớp: 156788

1. Assignment 1
   a. HWA 1
      - Giải thích:

- Khởi tạo tạo t0 = 0, s1, s2
- Sau khi thực hiện phép cộng s3 = s1 + s2, chương trình kiểm tra dấu của s1 và s2 bằng phép XOR (t1 = s1 ⊕ s2).
  - Nếu t1 âm (do bit dấu của s1 và s2 khác nhau, nghĩa là s1 và s2 khác dấu), chắc chắn không có tràn số.
  - Nếu s1 âm, chương trình kiểm tra xem s3 có lớn hơn s1 không, nếu có thì tràn số. Ngược lại, nếu s1 không âm, kiểm tra s3 < s1, nếu có thì tràn số. Nếu có tràn số, đặt t0 = 1, ngược lại thoát chương trình.

TH1: s1, s2 âm, s3 tràn số

```
# Laboratory Exercise 4, Home Assignme
.text
        li s1, 0x80000000 #khởi tạo s1
        li s2, 0x80000000 #khởi tạo s2
        li t0, 0 # Mặc định không có t
        add s3, s1, s2 # s3 = s1 + s2
        xor t1, s1, s2 # Kiểm tra s1 v
        blt t1, zero, EXIT # Nếu t1 là
        blt s1, zero, NEGATIVE # Kiểm t
        bge s3, s1, EXIT # s1 không âm,
        # Nếu s3 >= s1, không tràn số
        j OVERFLOW
NEGATIVE:
        bge s1, s3, EXIT # s1 âm, kiểm
        # Nếu s1 >= s3, không tràn số

OVERFLOW:
        li t0, 1 # The result is overf
EXIT:
```

Registers | Floating Point | Control and Status

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000001
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x80000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x80000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400038

- Trong TH1, s1 = s2 = 0x80000000 = -2147483648, đáng lẽ ra s3 = s1 + s2 = -4294967296 nhưng lại trả về s3 = 0x00000000 = 0 > s1 => tràn số => t0 = 1

Trạng thái	S3	T0	T1	Pc
Ban đầu	0x00000000	0x00000000	0x00000000	0x00400000
Sau lệnh addi	0x00000000	-	-	0x00400018
Sau lệnh xor	-	-	0x00000000	0x0040001c
Sau lệnh blt kiểm tra s1 s2 cùng dấu	-	-	-	0x00400020
Sau lệnh blt kiểm tra s1 s2 âm/dương	-	-	-	0x0040002c
Sau lệnh bge (ở lable NEGATIVE)	-	-	-	0x00400030
Sau lệnh li check tràn số	-	0x00000001	-	0x00400038

TH2: s1, s2 dương, s3 không tràn số

```
# Laboratory Exercise 4, Home Assignment 1
.text
        li s1, 0x00000008 #khởi tạo s1 = 8
        li s2, 0x00000010 #khởi tạo s2 = 16
        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 c
        blt t1, zero, EXIT # Nếu t1 là số âm,
        blt s1, zero, NEGATIVE # Kiểm tra s1
        bge s3, s1, EXIT # s1 không âm, kiểm
        # Nếu s3 >= s1, không tràn số
        j OVERFLOW
NEGATIVE:
        bge s1, s3, EXIT # s1 âm, kiểm tra s
        # Nếu s1 >= s3, không tràn số

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

Registers | Floating Point | Control and Status

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000018
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000008
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000010
s3	19	0x00000018
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
t6	31	0x00000000
pc		0x00400030

- Trong TH2, s1 = 8, s2 = 0x10 = 16, s3 = s1 + s2 = 0x18 = 24 > s1 => không xảy ra tràn số => t0 = 0

Trạng thái	S3	T0	T1	Pc
Ban đầu	0x00000000	0x00000000	0x00000000	0x00400000
Sau lệnh addi	0x00000018	-	-	0x00400010
Sau lệnh xor	-	-	0x00000018	0x00400014
Sau lệnh blt kiểm tra s1 s2 cùng dấu	-	-	-	0x00400018
Sau lệnh blt kiểm tra s1 s2 âm/dương	-	-	-	0x0040001c
Sau lệnh bge	-	0x00000000	-	0x0040002c (sau đó nhảy đến lable EXIT)

b. HWA 2

```
# Laboratory Exercise 4, Home Assignment 2
.text
        li s0, 0x12345678 # Test value
        andi t0, s0, 0xff # Extract LSB of s0
        andi t1, s0, 0x0400 # Extract bit 10 of s0
        #0x0400 = 0b100 0000 0000
```

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000078
t1	6	0x00000400
t2	7	0x00000000
s0	8	0x12345678

- Giải thích:

s0 = 0x12345678 = 0b 0001 0010 0011 0100 0101 0110 0111 1000

0xff = 0b 1111 1111

0x0400 = 0b 0000 0100 0000 0000

t0 = (s0 and 0xff) = 0b 01111000 = 0x78

t1 = (s0 and 0x0400) = 0b 0000 0100 0000 0000 = 0x400

c. HWA 3

```
# Laboratory Exercise 3, Home Assignment 3
.text
        li s0, 1 # s0 = 1
        li t0, 3 # t0 = 3
        sll s1, s0, t0 # s1 = s0 * 2^t0
```

t0	5	0x00000003
t1	6	0x00000000
t2	7	0x00000000
s0	8	0x00000001
s1	9	0x00000008

2. Assignment 2

```
.text
        li s0, 0x87654321
        li t0, 0xff000000
        li t1, 0xffffff00
        and s1, s0, t0        #Trích xuất MSB của thanh ghi s0
        and s0, s0, t1        #Xóa LSB của thanh ghi s0
        ori s0, s0, 0x000000ff#Thiết lập LSB của thanh ghi s0 (bit 7 đến bit 0 được thiết lập bằng 1)
        and s0, s0, zero      #Xóa thanh ghi s0 bằng cách dùng các lệnh logic (s0 = 0)
```

Khởi tạo s0 = 0x87654321 và t0, t1 cần thiết

t0	5	0xff000000
t1	6	0xffffff00
t2	7	0x00000000
s0	8	0x87654321

Trích xuất MSB của s0 và lưu vào s1

s1	9	0x87000000

Xóa LSB của s0

s0	8	0x87654300

Thiết lập LSB của s0

s0	8	0x876543ff

Xóa thanh ghi s0

s0	8	0x00000000

3. Assignment 3
    a. neg s0, s1

    => sub s0, zero, s1

    b. mv s0, s1

    => addi s0, s1, 0

    c. not s0, s1

    => xori s0, s1, -1

    d. ble s1, s2, label

    => bge s2, s1, lbl

4. Assignment 4

```
.text
        li s1, 0x7FFFFFFF   # = INT_MAX
        li s2, 0x00000001
        li t0, 0

        add s3, s1, s2
        xor t1, s1, s2
        blt t1, zero, EXIT

        xor t1, s1, s3
        blt t1, zero, OVERFLOW
        j EXIT

OVERFLOW:
    li t0, 1

EXIT:
```

tp	4	0x00000000
t0	5	0x00000001
t1	6	0xffffffff
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x7fffffff
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
a7	17	0x00000000
s2	18	0x00000001
s3	19	0x80000000

Do s1 + s0 = 0x7FFFFFFF + 0x1 (= 2147483647 + 1) = 0x800000000 (= -2147483648 trái dấu s1) => tràn số => t0 = 1

5. Assignment 5

```
.text
    li a0, 6
    li a1, 7
    li a2, 0
    li t0, 1

LOOP:
    and t1, a1, t0
    beq  t1, zero, SKIP
    add a2, a2, a0

SKIP:
    sll a0, a0, t0
    srl a1, a1, t0
    bne a1, zero, LOOP

EXIT:
```

Registers | Floating Point

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffeffc
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000001
t1	6	0x00000001
t2	7	0x00000000
s0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000030
a1	11	0x00000000
a2	12	0x0000002a

- Giải thích:

- Kiểm tra bit thấp nhất của a1
  - Nếu là 1, cộng a0 vào a2
  - Nếu là 0, bỏ qua
- Dịch trái a0
- Dịch phải a1 (loại bỏ bit cuối cùng)
- Lặp lại đến khi a1 = 0 (xét hết bit)

Loop	A1	T1	A2	A0
0	0000 0111 (7)	0	0000 0000 (0)	0000 0110 (6)
1	0000 0111 (7)	1	0000 0110 (6)	0000 1100 (12)
2	0000 0011 (3)	1	0001 0010 (18)	0001 1000 (24)
3	0000 0001 (1)	1	0010 1010 (42)	0011 0000 (48)
4	0000 0000 (0)	0	-	-

Kiểm tra 6 * 7 = 42 = 0x2a => thỏa mãn