

Họ và tên: Bùi Văn Anh

MSSV: 20184026

Học phần: Thực hành kiến trúc máy tính

Mã lớp: 122032

BÁO CÁO LAB2

I. Assignment 1:

- Chưa chạy: \$s0 = 0x00000000; pc = 0x00400000
Sau lệnh 1: \$s0 = 0x00003007; pc = 0x00400004
Sau lệnh 2: \$s0 = 0x00000000; pc = 0x00400008
- addi \$s0, \$zero, 0x3007
\$s0 = \$16 : 10000
\$zero = \$0: 00000
0x3007: 110000000000111

addi| \$s0 |\$zero | 0x3007

Mã máy: 0x20103007: 100000|00010|00000|110000000000111

op | rs | rt | offset

=> đúng tập lệnh

- Sửa lại lệnh

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x3c012110	lui \$1,0x00002110	3: addi \$s0, \$zero, 0x2110003d
<input type="checkbox"/>	0x00400004	0x3421003d	ori \$1,\$1,0x0000003d	
<input type="checkbox"/>	0x00400008	0x00018020	add \$16,\$0,\$1	
<input type="checkbox"/>	0x0040000c	0x00008020	add \$16,\$0,\$0	4: add \$s0, \$zero, \$0 # \$s0 = 0 + 0 = 0 ;R-type

Lệnh addi \$s0, \$zero, 0x2110003d thực chất thực hiện 3 lệnh (như hình)

lui: ghi hằng số 16 bit 0x00002110 vào 2 byte cao của \$1 => \$1 = 0x21100000

ori: ghi giá trị \$1 OR 0x0000003d vào \$1

add: cộng \$1 và \$0 ghi giá trị vào \$16

=> kết quả \$16=\$s0=0x2110003d

II. Assignment 2:

- Chưa chạy: \$s0 = 0x00000000; \$pc = 0x00400000
Sau lệnh 1: \$s0 = 0x21100000; \$pc = 0x00400004
Sau lệnh 2: \$s0 = 0x2110003d; \$pc = 0x00400008

- Các byte đầu tiên của vùng lệnh trùng với cột code trong cửa sổ Text Segment

Text Segment			Data Segment		
Bkpt	Address	Code	Address	Value (+0)	Value (+4)
<input type="checkbox"/>	0x00400000	0x3c102110	0x00400000	0x3c102110	0x3610003d
<input type="checkbox"/>	0x00400004	0x3610003d	0x00400004	0x00000000	0x00000000

III. Assignment 3:

- Điều bất thường: cùng là lệnh li nhưng lệnh 1 tách thành 2 lệnh mã máy nhưng lệnh 2 chỉ có 1 lệnh

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x3c012110	lui \$1,0x00002110	3: li \$s0,0x2110003d #pseudo instruction=2 basic instructions
<input type="checkbox"/>	0x00400004	0x3430003d	ori \$16,\$1,0x0000003d	
<input type="checkbox"/>	0x00400008	0x24110002	addiu \$17,\$0,0x00000002	4: li \$s1,0x2 #but if the immediate value is small, one ins

- Giải thích: Lệnh li thao tác với hằng số giới hạn 16 bit (Set \$s0 to 16 bits immediate) nên khi khởi tạo thanh ghi giá trị lớn hơn 16bit (32 bit) thì sẽ chạy 2 lệnh mã máy.

IV. Assignment 4: phép tính $2x+y=?$

	\$t1 = \$9	\$t2 = \$10	\$s0 = \$16	pc
Chưa chạy	0x00000000	0x00000000	0x00000000	0x00400000
addi \$t1, \$zero, 5	0x00000005	0x00000000	0x00000000	0x00400004
addi \$t2, \$zero, -1	0x00000005	0xffffffff	0x00000000	0x00400008
add \$s0, \$t1, \$t1	0x00000005	0xffffffff	0x0000000a	0x0040000c
add \$s0, \$s0, \$t2	0x00000005	0xffffffff	0x00000009	0x00400010

- Kết quả: $x = \$t1 = 5$, $y = \$t2 = 0xffffffff = -1 \Rightarrow \$s0 = 0x00000009$
-> kết quả đúng

0x00400000	0x20090005	addi \$9,\$0,0x00000005	4: addi \$t1, \$zero, 5
0x00400004	0x200affff	addi \$10,\$0,0xffffffff	5: addi \$t2, \$zero, -1

- Lệnh addi \$t1, \$zero, 5
addi \$9,\$0,0x00000005
\$t1 = \$9: 01001
\$zero = \$0: 00000
5 = 101

Mã máy: 0x20090005 : 001000|00000|01001|00000000000000101

addi |\$zero| \$t1 | 5

=> đúng khuôn mẫu kiểu lệnh I

0x00400008	0x01298020	add \$16,\$9,\$9	7: add \$s0, \$t1, \$t1
0x0040000c	0x020a8020	add \$16,\$16,\$10	8: add \$s0, \$s0, \$t2

- add \$s0, \$t1, \$t1 ~ add \$16, \$9, \$9
\$s0 = \$16: 10000
\$t1 = \$s9: 1001

0 | \$t1 | \$t1 | \$s0 | | add

Mã máy: 0x01298020: 000000|01001|01001|10000|00000|100000

op | rs | rt | rd | shamt|funct

=> đúng khuôn mẫu của lệnh kiểu R

V. Assignment 5:

Code	Basic	Source
0x20090004	addi \$9,\$0,0x00000004	4: addi \$t1, \$zero, 4 # X = \$t1 = ?
0x200a0005	addi \$10,\$0,0x00000005	5: addi \$t2, \$zero, 5 # Y = \$t2 = ?
0x712a8002	mul \$16,\$9,\$10	7: mul \$s0, \$t1, \$t2 # HI-LO = \$t1 * \$t2 = X * Y ; \$s0 = LO
0x20010003	addi \$1,\$0,0x00000003	8: mul \$s0, \$s0, 3 # \$s0 = \$s0 * 3 = 3 * X * Y
0x72018002	mul \$16,\$16,\$1	
0x00008812	mflo \$17	10: mflo \$s1

- Điều bất thường: Dòng 7 và 8 đều là lệnh mul nhưng khi biên dịch ra mã máy thì dòng 7 có 1 lệnh mã máy còn dòng 8 tách thành 2 lệnh addi và mul (như hình)
Giải thích: lệnh mã máy (Basic instruction) mul cú pháp mul \$t1, \$t2, \$t3 -> Nhân value 2 thành ghi t2, t3 lưu vào thanh ghi t1
-> Số 3 cần được gán vào thanh ghi -> lệnh addi để gán 3 vào \$1
Sau đó mới thực hiện phép nhân \$16 và \$1.

	\$t1	\$t2	\$s0	\$s1	at	Lo	Hi
	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
addi \$t1, \$zero, 4	0x00000004	-	-	-	-	-	-
addi \$t2, \$zero, 5	0x00000004	0x00000005	-	-	-	-	-
mul \$s0, \$t1, \$t2	-	-	0x00000014	-	-	0x00000014	0x00000000
mul \$s0, \$s0, 3	-	-	0x00000014	-	0x00000003	0x00000014	0x00000000
	-	-	0x0000003c	-	0x00000003	0x0000003c	0x00000000
mflo \$s1	-	-	-	0x0000003c	-	-	-

- Kết quả: $x=\$t1=4$; $y=\$t2=5 \rightarrow z=3*xy = 60 = 0x0000003c = \$s0$
=> kết quả đúng

VI. Assignment 6:

Text Segment				
Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	0x00400000	0x3c011001	lui \$1,0x00001001	8: la \$t8, X # Get the address of X in Data Segment
<input type="checkbox"/>	0x00400004	0x34380000	ori \$24,\$1,0x00000000	
<input type="checkbox"/>	0x00400008	0x3c011001	lui \$1,0x00001001	9: la \$t9, Y # Get the address of Y in Data Segment
<input type="checkbox"/>	0x0040000c	0x34390004	ori \$25,\$1,0x00000004	
<input type="checkbox"/>	0x00400010	0x8f090000	lw \$9,0x00000000(\$24)	10: lw \$t1, 0(\$t8) # \$t1 = X
<input type="checkbox"/>	0x00400014	0x8f2a0000	lw \$10,0x00000000(\$25)	11: lw \$t2, 0(\$t9) # \$t2 = Y
<input type="checkbox"/>	0x00400018	0x01298020	add \$16,\$9,\$9	13: add \$s0, \$t1, \$t1 # \$s0 = \$t1 + \$t1 = X + X = 2X
<input type="checkbox"/>	0x0040001c	0x020a8020	add \$16,\$16,\$10	14: add \$s0, \$s0, \$t2 # \$s0 = \$s0 + \$t2 = 2X + Y
<input type="checkbox"/>	0x00400020	0x3c011001	lui \$1,0x00001001	16: la \$t7, Z # Get the address of Z in Data Segment
<input type="checkbox"/>	0x00400024	0x342f0008	ori \$15,\$1,0x00000008	
<input type="checkbox"/>	0x00400028	0xadf00000	sw \$16,0x00000000(\$15)	17: sw \$s0, 0(\$t7) # Z = \$s0 = 2X + Y

- Lệnh la được biên dịch thành 2 lệnh mã máy lui và ori
VD: la \$t8, X -> lấy địa chỉ của X gán vào \$t8

lui \$1, 0x00001001 -> gán 16 bit đầu của địa chỉ của X vào 16 bit cao của \$1, 16 bit còn lại bằng 0

ori \$24,\$1,0x00000000 -> gán vào \$24 kết quả của phép OR \$1 và 0x00000000

- Địa chỉ của X, Y, Z không đổi khi lệnh la thành mã máy

	\$t1	\$t2	\$t8	\$t9	\$s0	\$t7
la \$t8, X	0x00000000	0x00000000	0x10010000	0x00000000	0x00000000	0x00000000
la \$t9, Y	0x00000000	0x00000000	0x10010000	0x10010004	0x00000000	0x00000000
lw \$t1, 0(\$t8)	0x00000005	0x00000000	0x10010000	0x10010004	0x00000000	0x00000000
lw \$t2, 0(\$t9)	0x00000005	0xffffffff	0x10010000	0x10010004	0x00000000	0x00000000
add \$s0,\$t1,\$t1	0x00000005	0xffffffff	0x10010000	0x10010004	0x0000000a	0x00000000
add \$s0,\$s0,\$t2	0x00000005	0xffffffff	0x10010000	0x10010004	0x00000009	0x00000000
la \$t7,Z	0x00000005	0xffffffff	0x10010000	0x10010004	0x00000009	0x10010008
sw \$s0, 0(\$t7)	0x00000005	0xffffffff	0x10010000	0x10010004	0x00000009	0x10010008

- lw: nạp 1 từ (word) từ dữ liệu bộ nhớ vào thanh ghi (load word)
VD: lw \$t1, 0(\$t8) -> Nạp từ có vị trí (\$t8+0) vào \$t1 -> \$t1 = X = 5
sw: lưu dữ liệu 1 từ từ thanh ghi vào bộ nhớ
VD: sw \$s0, 0(\$t7) -> lưu giá trị của thanh ghi s0 vào vùng nhớ địa chỉ (\$t7+0)
- lb \$t1, (\$t2): nạp vào thanh ghi t1 giá trị 8 bit (1byte) có dấu ở vị trí (\$t2) của bộ nhớ
sb \$t1, (\$t2): lưu 8bit thấp của giá trị thanh ghi t1 vào vùng nhớ có địa chỉ (\$t2)