TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH

Hướng dẫn thực hành

Môn: Kiến Trúc Máy Tính - 504002

Bài thực hành số 3

TP. HCM 9/2014

Nội dung

Bảng tóm tắt các lệnh trong bài thực hành

Instruction	Meaning	Instruction Format					
add $s1, $s2, $s3	$s1 = $s2 + $s3	op = 0	rs = $s2	rt = $s3	rd = $s1	sa = 0	f = 0x20
addu $s1, $s2, $s3	$s1 = $s2 + $s3	op = 0	rs = $s2	rt = $s3	rd = $s1	sa = 0	f = 0x21
sub $s1, $s2, $s3	$s1 = $s2 – $s3	op = 0	rs = $s2	rt = $s3	rd = $s1	sa = 0	f = 0x22
subu $s1, $s2, $s3	$s1 = $s2 – $s3	op = 0	rs = $s2	rt = $s3	rd = $s1	sa = 0	f = 0x23
and $s1, $s2, $s3	$s1 = $s2 & $s3	op = 0	rs = $s2	rt = $s3	rd = $s1	sa = 0	f = 0x24
or $s1, $s2, $s3	$s1 = $s2 | $s3	op = 0	rs = $s2	rt = $s3	rd = $s1	sa = 0	f = 0x25
xor $s1, $s2, $s3	$s1 = $s2 ^ $s3	op = 0	rs = $s2	rt = $s3	rd = $s1	sa = 0	f = 0x26
nor $s1, $s2, $s3	$s1 = ~($s2|$s3)	op = 0	rs = $s2	rt = $s3	rd = $s1	sa = 0	f = 0x27
sll $s1,$s2,10	$s1 = $s2 << 10	op = 0	rs = 0	rt = $s2	rd = $s1	sa = 10	f = 0x00
srl $s1,$s2,10	$s1 = $s2>>>10	op = 0	rs = 0	rt = $s2	rd = $s1	sa = 10	f = 0x02
sra $s1, $s2, 10	$s1 = $s2 >> 10	op = 0	rs = 0	rt = $s2	rd = $s1	sa = 10	f = 0x03
mult $s1, $s2	(Hi, Lo) = $s1*$s2	op = 0	rs = $s1	rt = $s2	rd = 0	sa = 0	f = 0x18
multu $s1, $s2	(Hi, Lo) = $s1*$s2	op = 0	rs = $s1	rt = $s2	rd = 0	sa = 0	f = 0x19
div $s1, $s2	Hi = $s1/$s2 Lo = $s1%$s2	op = 0	rs = $s1	rt = $s2	rd = 0	sa = 0	f = 0x1A
divu $s1, $s2	Hi = $s1/$s2 Lo = $s1%$s2	op = 0	rs = $s1	rt = $s2	rd = 0	sa = 0	f = 0x1B
mfhi $s1	$s1 = Hi	op = 0	rs = 0	rt = 0	rd = $s1	sa = 0	f = 0x10
mflo $s1	$s1 = Lo	op = 0	rs = 0	rt = 0	rd = $s1	sa = 0	f = 0x12
Các lệnh số học/luận lý có hằng số							
addi $s1, $s2, 10	$s1 = $s2 + 10	op = 0x8	rs = $s2	rt = $s1	Imm^{16} = 10		
addiu $s1, $s2, 10	$s1 = $s2 + 10	op = 0x9	rs = $s2	rt = $s1	Imm^{16} = 10		
andi $s1, $s2, 10	$s1 = $s2 & 10	op = 0xc	rs = $s2	rt = $s1	Imm^{16} = 10		
ori $s1, $s2, 10	$s1 = $s2 | 10	op = 0xd	rs = $s2	rt = $s1	Imm^{16} = 10		
xori $s1, $s2, 10	$s1 = $s2 ^ 10	op = 0xe	rs = $s2	rt = $s1	Imm^{16} = 10		
lui $s1, 10	$s1 = 10 << 16	op = 0xf	0	rt = $s1	Imm^{16} = 10		
Các lệnh rẽ nhánh, nhảy							
j label	jump to label	op = 2	Imm^{26}				
beq $s1, $s2, label	branch if ($s1 == $s2)	op = 4	rs = $s1	rt = $s2	Imm^{16}		
bne $s1, $s2, label	branch if ($s1 != $s2)	op = 5	rs = $s1	rt = $s2	Imm^{16}		
blez $s1, label	branch if ($s1 <= 0)	op = 6	rs = $s1	0	Imm^{16}		
bgtz $s1, label	branch if ($s1 > 0)	op = 7	rs = $s1	0	Imm^{16}		
bltz $s1, label	branch if ($s1 < 0)	op = 1	rs = $s1	0	Imm^{16}		

bgez	$s1, label	branch if ($s1 >= 0)	op = 1	rs = $s1	1	Imm16	
slt	$t0,$s1,$s2	$t0=($s1<$s2?1:0)	op = 0	rs = $s1	rt = $s2	rd = $t0	0	f = 0x2a
sltu	$t0,$s1,$s2	$t0=($s1<$s2?1:0)	op = 0	rs = $s1	rt = $s2	rd = $t0	0	f = 0x2b
slti	$t0,$s1,10	$t0=($s1<10?1:0)	op = 0xa	rs = $s1	rt = $t0	Imm16 = 10	
sltiu	$t0,$s1,10	$t0=($s1<10?1:0)	op = 0xb	rs = $s1	rt = $t0	Imm16 = 10	
Các lệnh truy xuất bộ nhớ load và store							
lb	$s1, imm16($s0)	s1 = MEM[s0+imm16]	op = 0x20	rs = $s0	rt = $s1	Imm16	
lh	$s1, imm16($s0)	s1 = MEM[s0+imm16]	op = 0x21	rs = $s0	rt = $s1	Imm16	
lw	$s1, imm16($s0)	s1 = MEM[s0+imm16]	op = 0x23	rs = $s0	rt = $s1	Imm16	
lbu	$s1, imm16($s0)	s1 = MEM[s0+imm16]	op = 0x24	rs = $s0	rt = $s1	Imm16	
lhu	$s1, imm16($s0)	s1 = MEM[s0+imm16]	op = 0x25	rs = $s0	rt = $s1	Imm16	
sb	$s1, imm16($s0)	MEM[s0+imm16] = s1	op = 0x28	rs = $s0	rt = $s1	Imm16	
sh	$s1, imm16($s0)	MEM[s0+imm16] = s1	op = 0x29	rs = $s0	rt = $s1	Imm16	
sw	$s1, imm16($s0)	MEM[s0+imm16] = s1	op = 0x2b	rs = $s0	rt = $s1	Imm16	

Bảng tóm tắt các lệnh trong bài thực hành

Bài thực hành này nhằm mục đích nắm được cách dùng các lệnh store(ghi), load(đọc), cách khai báo các array (BYTE, HALF, WORD, FLOAT, DOUBLE, ASCII, ASCIIZ, SPACE ...) cùng với việc xử lý trên các array đó. Ở mỗi câu thực hành lưa lại thành file và được đặt tên theo format mssv_lab_bai.[asm,txt]. Ví dụ: 5130xxxx_lab4_bai1a.asm

Bài 1. Lệnh load/store (đọc/ghi) dữ liệu.

Cho biểu thức f như bên dưới (bài 2 lab 2).

$f = ax^3 - bx^2 - cx + d$

với a = 4, b = 3, c = 2, d = 1

Trong đó a, b, c, d x là những số nguyên và được lưa sẵn ở vùng data. Tính giá trị của biểu thức f rồi ghi giá trị đó vào vùng dữ liệu có nhãn là **kq** được mô tả sẵn bên dưới.

```
.data
x:    .word    0
a:    .word    1
b:    .word    2
c:    .word    3
d:    .word    4
kq:   .word    0   #quan sát giá trị "kq" ở cửa sổ "data"
.text
main:
   la   $s0, x    #$s0 chứa địa chỉ của biến x
   #địa chỉ biến (a, b, c, d, kq) lần lược là $s0 +
   #(4, 8, 12, 16, 20)

   #thêm code tính toán biểu thức f
```

Dùng lệnh load (lw –load word) để đọc, store (sw –store word) để ghi.

Bài 2. Sắp xếp mảng

Cho mảng các phần tử 10 sau:

1, 6, 3, 23, 3, 7, 8, 34, 24, 50

Viết chương trình bằng hợp ngữ sắp xếp lại mảng trên theo chiều tăng dần.

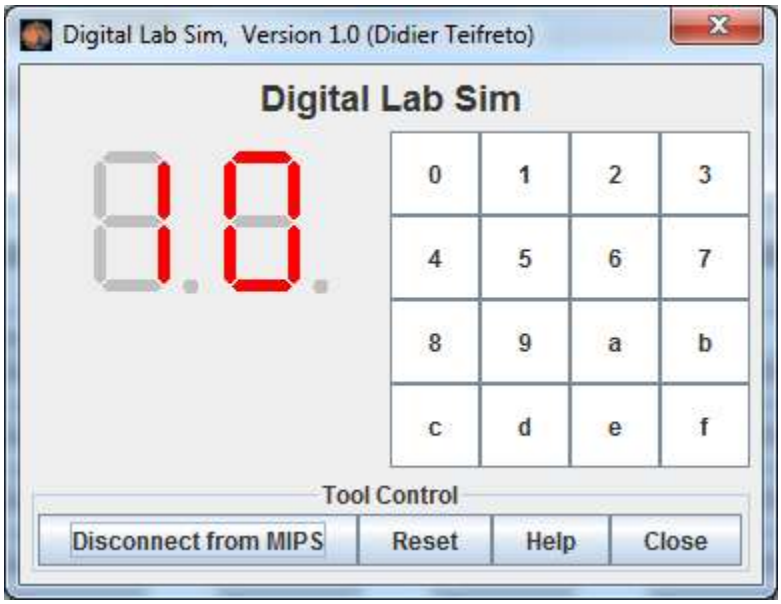
Bài 3. Xử lý chuỗi

Cho chuỗi ký tự sau:

"Kien Truc May Tinh CS13"

Sinh viên viết đoạn chương trình để sinh ra chuỗi mới từ chuỗi ký tự trên. Những ký tự nào viết hoa thì sẽ chuyển thành viết thường và ngược lại. Chỉ áp dụng cho ký tự chữ.

Bài 4. Xuất giá trị ra LED 7 đoạn



Sử dụng công cụ "Digital Lab Sim" (trong menu "Tool"), viết chương trình hiển thị số có hai chữ số bất kỳ được nhập từ người dùng.

Đoạn chương trình mẫu dưới đây hiển thị số "10" . Tham khảo phần "Help" của công cụ này để hiểu rõ chương trình.

```
# Display LED's Value using Digital Lab Sim,
# please read "help"
      .data
LEDL: .byte  1          # Left Digit value
LEDR: .byte  0          # Right Digit value
LEDFONT1: .byte 0x3f, 0x06, 0x5b, 0x4f
      .text
      la   $t0, LEDL  #load address of Left Digit value byte
      la   $t1, LEDR  # load address of Right Digit value byte
      la   $t5, LEDFONT1
      li   $t2, 0xFFFF0011   # load address of Left Digit
      li   $t3, 0xFFFF0010   # load address of Right Digit
      lbu  $t0, 0($t0)       # load value of Left Digit
```

```
        lbu   $t1, 0($t1)   #load value of Right Digit

        add   $t0,$t0,$t5   #get the LEDFONT1[LEDL]
        lb    $t0,0($t0)
        sb    $t0, 0($t2)   #push the LEDFONT1[LEDL] to Left LED

        add   $t1,$t1,$t5   # get the LEDFONT1[LEDR]
        lb    $t1,0($t1)
        sb    $t1, 0($t3)   # push the LEDFONT1[LEDR] to Right LED

        li    $v0, 10       # system call for exit
        syscall             # we are out of here.
```