EE352L Lab 3

Sang W. S. Do

09/25/2015

Lab 1

- Due date
  - *Due: In class, Friday, September 25th, 2015*
- Make a file ‘lab1.asm’ and do the followings:
  - 1. Initialize register t0, t1, t2, t3, t4, t5 as 0.
  - 2. t0 stores 1.
  - 3. t1 stores negative 1 in 2’s complement.
  - 4. t2 stores 1 if t1 is less than t0.
  - 5. t3 stores 2*t0.
  - 6. t4 stores the 1’s complement of t0.
  - 7. t5 stores the 2’s complement of t0.

Lab 1 (Contd.)

- *Instructions allowed:*
  - addi, add, sub, slt, sll, xori, addi
  - Refer to the help of MARS as necessary.

Lab 1 (Contd.)

```
•   .text
•       addi $t0,$0,0       #Initialize $t0 as 0
•       add $t1,$0,$0       #Initialize $t1 as 0
•       add $t2,$0,$0       #Initialize $t2 as 0
•       add $t3,$0,$0       #Initialize $t3 as 0
•       add $t4,$0,$0       #Initialize $t4 as 0
•       add $t5,$0,$0       #Initialize $t5 as 0
•       addi $t0,$0,1
•       sub $t1,$0,$t0      #subtraction
•       slt $t2,$t1,$t0     #set $t2 if $t1 is less than $t0
•       sll $t3,$t0,1       #multiplication *2
•       xori $t4,$t0,0xffffffff #1' complement
•       addi $t5,$t4,1      # 2's complement
```

Lab 3 Assignment

- Bubble sort & Stack
  - *Due: 11:00:00 AM. PST, October 2nd, 2015*
  - Bubble-sort positive numbers in ascending order.
  - Push on Stack.
  - In Blackboard
    - Download ‘lab3Template.asm’.
    - Rename as ‘FirstnameLastname_Lab3.asm’.
    - Submit ‘FirstnameLastname_Lab3.asm’.
    - *Do not submit at the last minute!*
      - The link may disappear.

MIPS Addressing Mode

- How to locate data.
  - In the load-store or register-register architecture.
- 2+2 modes
  - Immediate: ADDI R4, R4, #2
  - Displacement: LD R1, 30(R2)
  - Register Indirect: LD R1, 0(R2)
  - Absolute/Direct: LD R1, 30(R0)

MIPS Addressing Mode (Contd.)

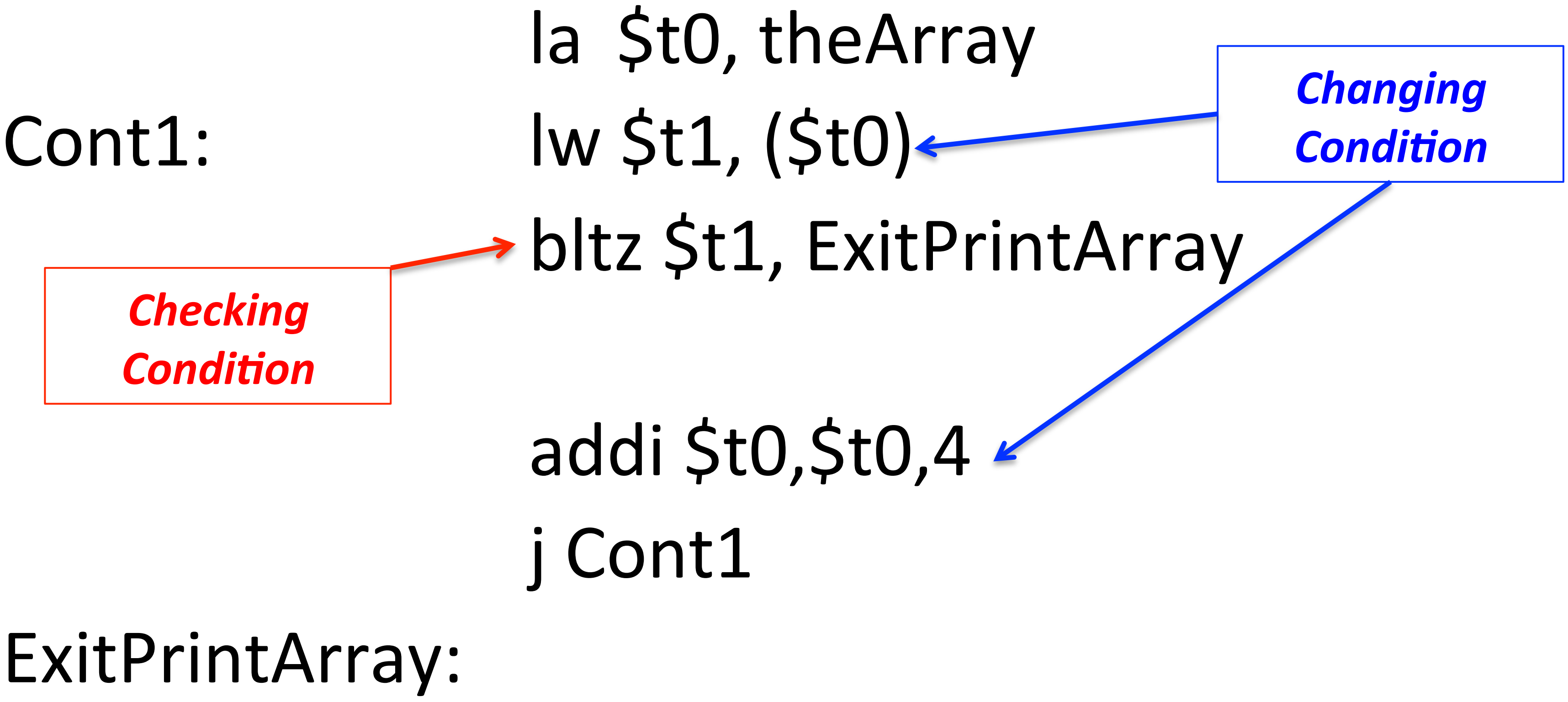
- Mars examples
  - la $t0, theArray
  - lw $t1, ($t0)
  - sw $t0,($t1)

Loop

- Loop structure
  - Checking condition
  - Changing condition
- MIPS
  - Branches
    - By changing PC (Program Counter)
    - Unconditional Branches: j Lable1
    - Conditional Branches: beq $t1, $t2, Lable2

Loop (Contd.)

- Example



```
                la  $t0, theArray
Cont1:          lw $t1, ($t0)
                bltz $t1, ExitPrintArray

                addi $t0,$t0,4
                j Cont1
ExitPrintArray:
```

Changing Condition

Checking Condition

Bubble Sort

- Swap two adjacent numbers if necessary.
  - Until no swap is done.
  - https://en.wikipedia.org/wiki/Bubble_sort
- Outer loop
  - Check whether a swap is done.
- Inner loop
  - Compare & swap from the first to the last number in the array.
  - Delimiter or stopper is -1.
    - Check out the two numbers being compared (< 0 ?)

Stack - basic

- LIFO
  - Last In First Out / First In Last Out
- Two pointers
  - Stack base and Stack pointer
  - Initially (Stack Base) = (Stack pointer)
- Two operations
  - Push
    - Put something and increment the stack pointer.
  - Pop
    - Decrement the stack pointer and get data.

lab3Template.asm

- Open the file in Mars.