CDA 3100

Fall 2013
Recitation – Week 3
MIPS Coding
Structure of a MIPS program

- **Text segment**
  - Contains all the code

- **Data segment**
  - Contains the data, like arrays, strings, etc.
  - Data segment begins with `.data`. Can be written before or after the text segment.
Memory Access

• Data is stored in words. Programmers tend to think in words.
• To move to the next logical memory location, we need to move 4 bytes.
• We can put data in an array, and move using a counter register, incrementing by 4.
LA, LI Pseudoinstructions

• Load Address

• LA d, label
  – Example: la $s0, A
  – Loads the address of the label into the register.

• Load Immediate

• LI d, immediate
  – Example: li $t1, 10
  – Loads the immediate value into the register.
Shifts

Shift instructions move all the bits in a word to the left or to the right.

– Shift left logical (sll) move all the bits to the left by the specified number of bits
  • sll $t2, $t1, 4

– Shift right logical (srl) move all the bits to the right
  • srl $t1, $t0, 3

– Fill the emptied bits with 0’s
### Branches and loops

- MIPS has conditional and non conditional branches and jumps to manage loops.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>beq</td>
<td>s,t,addr</td>
<td>branch if $s == t$</td>
</tr>
<tr>
<td>bne</td>
<td>s,t,addr</td>
<td>branch if $s !!= t$</td>
</tr>
<tr>
<td>j</td>
<td>target</td>
<td>after a delay of one machine cycle, PC $\leftarrow$ address of target</td>
</tr>
</tbody>
</table>
SLT, SLTI

- Set if Less Than
- `Slit t,s1,s2`
- Example: `slt $s1,$t1,$t2`
- Sets `s1` to 1 if `t1 < t2`. `S1` is 0 otherwise

- For `slti`, `s2` is an immediate operand (signed int). Logic is the same.
Program to find the largest number in an array

• Given an array of 10 numbers, write a MIPS program to find the largest
  – Assign a variable to be the largest number. Give it a small value (example 0).
  – Load the array elements one by one.
  – Check if the current element is larger than the variable, correct the value, and record the index.
Program to find the largest number in an array

# find the index of the maximum element in an array with 10 positive elements

.data
A: .word 34, 67, 10, 45, 90, 11, 3, 67, 19, 100

.text
.globl main
main:
la $s0, A       # starting address of A in $s0
li $s1, 10     # number of elements in $s1
li $s2, 0      # i in $s2
li $s3, 0      # max in $s3
li $s4, -1     # index in $s4
Program to find the largest number in an array (cont)

Loop:  
sll $t1, $s2, 2  # $t1 = i * 4;
add $t1, $t1, $s0  # $t1 = i * 4 + $s0
lw $t0, 0($t1)  # $t0 = A[i]
slt $t2, $t0, $s3  # $t2 = 1 if $t0 < $s3. $t2 = 0 if $t0 >= $s3.
bne $t2, $zero, L1  # if ($t2 != 0), s3 is still max, goto L1
ori $s3, $t0, 0  # update max value
ori $s4, $s2, 0  # update max index

L1:  
addi $s2, $s2, 1  # i = i + 1
bne $s2, $s1, Loop  # if (i != $s1), go back to loop

done:  
li $v0, 10
syscall