**Mips Terms**

4 bits= nibble

8 bits= byte

16/32 bits= word

Range of word= 256

Memory= large one dimensial 8 bit array

**MIPS IS LITTLE ENDIAN**

**Program Stack Uses**

1. Saving values of spilled registers
2. Storing local variables
3. Return values
4. Backup $ra

**Activation Frame**

1. Size of activation frame **IS FIXED**
2. Size determined by purpose
3. Location unknown till call is made

**Calling Conventions**

1. $a0-$a3- passes in values
2. $v0, $v1- returns values
3. $ra- return register
4. $t0-$t9- temp(do not expect save)(caller Responsibility)
5. $s0-$s7- saved(back up if use in function)(callee responsibility)

**FORMULA FOR ASSEMBLY TO MACHINE**

BEQ/ BNE = **PC = PC +4+(Sign-Extend)(imm)\*4)**

J TYPE = **PC = address <<2 or address \*4**

**Storing values to stack**

Add $sp, $sp, -4

Sw $s0, 0($sp)

(DO FUNCTION)

Lw $s0, 0($sp)

Addi $sp, $sp, 4

**Binary to Decimal Examples**

|  |  |  |  |
| --- | --- | --- | --- |
| Sign-Magnitude | | One’s | Two’s |
| −33 | 10100001 | 11011110 | 11011111 |
| −115 | 11110011 | 10001100 | 10001101 |

Conversion Examples

slt $t0, $s0, $a0 = R TYPE

op (6 bits): 0H or 0000002

• rs (5 bits): $s0 or 1610 or 1000002

• rt (5 bits): $a0 or 410 or 001002

• rd (5 bits): $t0 or 810 or 010002

• shamt (5 bits): 000002

• funct (6 bits): 2aH or 1010102

sb $s2, -12($sp) = I TYPE

• op (6 bits): 28H or 1010002

• rs (5 bits): $sp or 2910 or 111012

• rt (5 bits): $s2 or 2810 or 100102

• imm (16 bits): −12 or 11111111111101002

0020 | add $v0, $zero, $zero

0024 |loop: lb $t1, 0($t0)

0028 | beq $t1, $zero, done

0032 | addi $v0, $v0, 1

0036 | addi $t0, $t0, 1

0040 | j loop

0044 |done: jr $ra

Beq $t1, $zero, done

• op (6 bits): 4H or 0001002

• rs (5 bits): $t1 or 910 or 010012

• rt (5 bits): $zero or 010 or 000002

• Imm (16 bits): 3 (see why below)

PC = PC + 4 + (SignExt(Imm) ∗ 4)

44 = 28 + 4 + (SignExt(Imm) ∗ 4)

44 = 32 + (SignExt(Imm) ∗ 4)

12 = SignExt(Imm) ∗ 4

3 = SignExt(Imm)

3 = Imm

J loop

• op (6 bits): 2H or 0000102

• address (26 bits):

PC = address << 2

PC = address ∗ 4

24 = address ∗ 4

6 = address

lb $t2, 5($t6)? = I Type

lb = 20h in this scenaria

* op is 20H or 100000
* rs is $t6 or 01110
* rt is $t2 or 01010
* imm is 5 or 00....0101

81CA0005H

**Solving A PROBLEM SOLUTION IN CLASS**

Add $sp, $sp, -8

Sw $t0, 0($sp)

Sw $t1, 4($sp)

Sub $t1, $0, $t1

Lw $t0, 0($sp)

Lw $t1, 4($sp)

Addi $sp, $sp, 8

**AND** 1&0=0, **1&1=1**

**XOR** 0x0=0, **0x1=1** 1x1=0

**NOR** 0 nor 0=1 **1 nor0 =0** 1 nor 1=0

**OR** 0|0 =0 **1|0=2**

**Code Example**

.data

str: .asciiz "Computer Organization and Assembly Language\0"

.text

la $s0, str

loop: lb $a0, 0($s0) # Set the character to be printed

beq $a0, $zero, done

addi $v0, $zero, 11 # Syscall 11: print a character

syscall # Print the character in $a0

addi $s0, $s0, 2

j loop

done: addi $v0, $zero, 10 # Syscall 10: exit

syscall # Exit program**OUTPUT= Cmue raiainadAsml agae**

li   $s0, 0x123489ab **becomes**

lui  $at, 0x1234  
ori  $s0, $at, 0x89ab

|  |
| --- |
| According to the MARS program that we are using in class. If we store an integer 0x15a9b379 (a word) at the memory location 1210. What is the byte value stored at memory location 1210? |
| |  |  | | --- | --- | | Selected Answer: | 79 | |

36|         add $t0, $t1, $t2  
40|         j aLabel  
44|         sub $t0, $t1, $t2  
:  
76|         and $t0, $t1, $t2  
80| aLabel: or  $t0, $t1, $t2  
84|         nor $t0, $t1, $t2

ADRESS = 20