|  |  |  |
| --- | --- | --- |
| Decimal | 8-bit binary | 2-digit hexadecimal |
| 2210 | 000101102 | 16H |
| 10010 | 011001002 | 64H |

|  |  |  |  |
| --- | --- | --- | --- |
| Decimal | 8-bit Binary Sign-magnitude | 8-bit Binary One’s complement | 8-bit Binary Two’s complement |
| -3310 | 101000012 | 110111102 | 110111112 |
| -11510 | 111100112 | 100011002 | 100011012 |

From decimal to one’s complement:

Take the absolute value and convert it to normal binary

Flip the bits

From decimal to two’s complement:

Convert to one’s complement

Add 1

slt $t0, $s0, $a0 to binary: 000000 100000 00100 01000 00000 101010

op rs rt rd shamt funct

**shamt –** only used for instructions like sll or srl

sb $s2, -12($sp) to binary: 101000 11101 10010 1111111111110100

Find the immediate of a beq:

PC = PC + 4 + ( SignExt(immediate) << 2 )

**Calling Conventions**

$a0 - $a3 = passing arguments to a function

$v0 - $v1 = return values from a function

$s0 - $s7 = “saved registers”; a function should save

The values of these registers to the stack

if it will overwrite their values, and restore

them before returning to the caller

$t0 - $t9 = “temporary registers”; a caller should save

The values of these registers to the stack before

calling a function if they will still be needed

after the function returns, at which point it

should restore them

**Calling another function**

Before calling, save the registers:

addi $sp, $sp, -12

sw $s0, 0($sp)

sw $s1, 4($sp)

sw $s2, 8($sp)

When the function returns:

lw $s2, 8($sp)

lw $s1, 4($sp)

lw $s0, 0($sp)

addi $sp, $sp, 12

**Jumping to a function –** jal func\_name

**Returning from a function –** jr $ra