

**ILOC Reference**  
**CS 432 - Spring 2016**

Form	Op1	Op2	Op3	Comment
<b>Integer Arithmetic</b>				
add	op1, op2 =>	op3	reg reg reg	addition
sub	op1, op2 =>	op3	reg reg reg	subtraction
mult	op1, op2 =>	op3	reg reg reg	multiplication
div	op1, op2 =>	op3	reg reg reg	division
addI	op1, op2 =>	op3	reg imm reg	addition w/ constant
multI	op1, op2 =>	op3	reg imm reg	multiplication w/ constant
neg	op1 =>	op2	reg reg	arithmetic negation
<b>Boolean Arithmetic</b>				
and	op1, op2 =>	op3	reg reg reg	boolean AND
or	op1, op2 =>	op3	reg reg reg	boolean OR
not	op1 =>	op2	reg reg	boolean NOT
<b>Data Movement</b>				
i2i	op1 =>	op2	reg reg	register copy
loadI	op1 =>	op2	imm reg	load integer constant
loadS	&op1 =>	op2	sym reg	load symbol address
load	[op1] =>	op2	reg reg	load from address
loadAI	[op1+op2] =>	op3	reg imm reg	load from base + immediate offset
loadAO	[op1+op2] =>	op3	reg reg reg	load from base + offset
store	op1 =>	[op2]	reg reg	store to address
storeAI	op1 =>	[op2+op3]	reg reg imm	store to base + immediate offset
storeAO	op1 =>	[op2+op3]	reg reg reg	store to base + offset
<b>Comparison</b>				
cmp_LT	op1, op2 =>	op3	reg reg reg	less-than comparison
cmp_LE	op1, op2 =>	op3	reg reg reg	less-than-or-equal-to comparison
cmp_EQ	op1, op2 =>	op3	reg reg reg	equality comparison
cmp_GE	op1, op2 =>	op3	reg reg reg	greater-than-or-equal-to comparison
cmp_GT	op1, op2 =>	op3	reg reg reg	greater-than comparison
cmp_NE	op1, op2 =>	op3	reg reg reg	inequality comparison
<b>Control Flow</b>				
label	("op1:")		lbl	control flow label
jump	op1		lbl	unconditional branch
cbr	op1 =>	op2, op3	reg lbl lbl	conditional branch
param	op1		reg	pass parameter
call			fun	call function
return				return to caller
<b>Miscellaneous</b>				
print			imm/ reg/ str	print value to standard out
nop				no-op (do nothing)
phi			reg reg reg	$\phi$ -function (for SSA only)

Op	Meaning
reg	register (int or bool)
imm	immediate (int constant)
sym	symbol
lbl	jump label
fun	call label
str	string