Op Code	Op code representation	Function	Function codes
Arithmetic	00	Add	0000
		Add immediate	0001
		Multiply (unsigned)	0100
		Multiply (signed)	0101
		Comp	1000
		Complement Immediate	1001
Logic	00	AND	1100
		OR	1101
Shift	00	Shift left logical	0010
		Shift right logical	0011
		Shift left logical variable	0110
		Shift right logical variable	0111
		Shift right arithmetic	1110
		Shift right arithmetic variable	1111
Memory	01	Store Word	0000
	10	Load Word	0000

Branch	11	Unconditional branch	0000
		Branch Register	0001
		Branch on zero	0010
		Branch on not zero	0011
		Branch on Carry	0100
		Branch on No Carry	0101
		Branch on Sign	1000
		Branch on Not Sign	1001
		Branch on Overflow	1100
		Branch on No Overflow	1101
		Call	1110
		Return	1111

Instruction Format

OP	rs	rt	Func	Offset
2	5	5	4	16

Register Number	Conventional Name	Usage
$0	$zero	Hard-wired to 0
$1	$ra	Return Address
$2 - $3	$v0, $v1	Return values from functions
$4 - $7	$a0 - $a3	Arguments to functions - not preserved by subprograms
$8 - $15	$t0 - $t7	Temporary data, not preserved by subprograms
$16 - $23	$s0 - $s7	Saved registers, preserved by subprograms
$24 - $25	$t8 - $t9	More temporary registers, not preserved by subprograms
$26 - $27	$k0 - $k1	Reserved for kernel. Do not use.
$28	$gp	Global Area Pointer (base of global data segment)
$29	$sp	Stack Pointer
$30	$fp	Frame Pointer
$31	$at	Reserved for pseudo-instructions