

Quick Reference

Integer Instruction Set

Name	Syntax	Space/Time
Add: _____	add Rd, Rs, Rt	1/1
Add Immediate: _____	addi Rs, Rt, Imm	1/1
Add Immediate Unsigned: _____	addiu Rs, Rt, Imm	1/1
Add Unsigned: _____	addu Rd, Rs, Rt	1/1
And: _____	and Rd, Rs, Rt	1/1
And Immediate: _____	andi Rt, Rs, Imm	1/1
Branch if Equal: _____	beq Rs, Rt, Label	1/1
Branch if Greater Than or Equal to Zero: _____	bgez Rs, Label	1/1
Branch if Greater Than or Equal to Zero and Link: _____	bgezal Rs, Label	1/1
Branch if Greater Than Zero: _____	bgtz Rs, Label	1/1
Branch if Less Than or Equal to Zero: _____	blez Rs, Label	1/1
Branch if Less Than Zero and Link: _____	bltzal Rs, Label	1/1
Branch if Less Than Zero: _____	bltz Rs, Label	1/1
Branch if Not Equal: _____	bne Rs, Rt, Label	1/1
Divide: _____	div Rs, Rt	1/38
Divide Unsigned: _____	divu Rs, Rt	1/38
Jump: _____	j Label	1/1
Jump and Link: _____	jal Label	1/1
Jump and Link Register: _____	jalr Rd, Rs	1/1
Jump Register: _____	jr Rs	1/1
Load Byte: _____	lb Rt, offset(Rs)	1/1
Load Byte Unsigned: _____	lbu Rt, offset(Rs)	1/1
Load Halfword: _____	lh Rt, offset(Rs)	1/1
Load Halfword Unsigned: _____	lhu Rt, offset(Rs)	1/1
Load Upper Immediate: _____	lui Rt, Imm	1/1
Load Word: _____	lw Rt, offset(Rs)	1/1
Load Word Left: _____	lwl Rt, offset(Rs)	1/1
Load Word Right: _____	lwr Rt, offset(Rs)	1/1
Move From High: _____	mfhi Rd	1/1
Move From Low: _____	mflo Rd	1/1
Move to High: _____	mthi Rs	1/1
Move to Low: _____	mtlo Rs	1/1
Multiply: _____	mult Rs, Rt	1/32
Multiply Unsigned: _____	multu Rs, Rt	1/32
NOR: _____	nor Rd, Rs, Rt	1/1
OR: _____	or Rd, Rs, Rt	1/1
OR Immediate: _____	ori Rt, Rs, Imm	1/1
Store Byte: _____	sb Rt, offset(Rs)	1/1
Store Halfword: _____	sh Rt, offset(Rs)	1/1
Shift Left Logical: _____	sll Rd, Rt, sa	1/1
Shift Left Logical Variable: _____	sllv Rd, Rt, Rs	1/1
Set on Less Than: _____	slt Rd, Rt, Rs	1/1
Set on Less Than Immediate: _____	slti Rt, Rs, Imm	1/1
Set on Less Than Immediate Unsigned: _____	sltiu Rt, Rs, Imm	1/1
Set on Less Than Unsigned: _____	sltu Rd, Rt, Rs	1/1
Shift Right Arithmetic: _____	sra Rd, Rt, sa	1/1
Shift Right Arithmetic Variable: _____	srav Rd, Rt, Rs	1/1
Shift Right Logical: _____	srl Rd, Rt, sa	1/1
Shift Right Logical Variable: _____	srlv Rd, Rt, Rs	1/1
Subtract: _____	sub Rd, Rs, Rt	1/1
Subtract Unsigned: _____	subu Rd, Rs, Rt	1/1

Store Word:_____	sw	Rt, offset(Rs)	1/1
Store Word Left:_____	swl	Rt, offset(Rs)	1/1
Store Right: _____	swr	Rt, offset(Rs)	1/1
System Call: _____	syscall		1/1
Exclusive OR:_____	xor	Rd, Rs, Rt	1/1
Exclusive OR Immediate:_____	xori	Rt, Rs, Imm	1/1

Macro instructions

Name	Syntax	Space/Time
Absolute Value:_____	abs Rd, Rs	3/3
Branch if Equal to Zero:_____	beqz Rs, Label	1/1
Branch if Greater Than or Equal :_____	bge Rs, Rt, Label	2/2
Branch if Greater Than or Equal Unsigned:_____	bgeu Rs, Rt, Label	2/2
Branch if Greater Than:_____	bgt Rs, Rt, Label	2/2
Branch if Greater Than Unsigned:_____	bgtu Rs, Rt, Label	2/2
Branch if Less Than or Equal:_____	ble Rs, Rt, Label	2/2
Branch if Less Than or Equal Unsigned:_____	bleu Rs, Rt, Label	2/2
Branch if Less Than:_____	blt Rs, Rt, Label	2/2
Branch if Less Than Unsigned:_____	bltu Rs, Rt, Label	2/2
Branch if Not Equal to Zero:_____	bnez Rs, Label	1/1
Branch Unconditional:_____	b Label	1/1
Divide:_____	div Rd, Rs, Rt	4/41
Divide Unsigned:_____	divu Rd, Rs, Rt	4/41
Load Address:_____	la Rd, Label	2/2
Load Immediate:_____	li Rd, value	2/2
Move:_____	move Rd, Rs	1/1
Multiply:_____	mul Rd, Rs, Rt	2/33
Multiply (with overflow exception):_____	mulo Rd, Rs, Rt	7/37
Multiply Unsigned (with overflow exception):_____	mulou Rd, Rs, Rt	5/35
Negate:_____	neg Rd, Rs	1/1
Negate Unsigned:_____	negu Rd, Rs	1/1
Nop:_____	nop	1/1
Not:_____	not Rd, Rs	1/1
Remainder Unsigned:_____	remu Rd, Rs, Rt	4/40
Rotate Left Variable:_____	rol Rd, Rs, Rt	4/4
Rotate Right Variable:_____	ror Rd, Rs, Rt	4/4
Remainder:_____	rem Rd, Rs, Rt	4/40
Rotate Left Constant:_____	rol Rd, Rs, sa	3/3
Rotate Right Constant:_____	ror Rd, Rs, sa	3/3
Set if Equal:_____	seq Rd, Rs, Rt	4/4
Set if Greater Than or Equal:_____	sge Rd, Rs, Rt	4/4
Set if Greater Than or Equal Unsigned:_____	sgeu Rd, Rs, Rt	4/4
Set if Greater Than:_____	sgt Rd, Rs, Rt	1/1
Set if Greater Than Unsigned:_____	sgtu Rd, Rs, Rt	1/1
Set if Less Than or Equal:_____	sle Rd, Rs, Rt	4/4
Set if Less Than or Equal Unsigned:_____	sleu Rd, Rs, Rt	4/4
Set if Not Equal:_____	sne Rd, Rs, Rt	4/4
Unaligned Load Halfword Unsigned:_____	ulh Rd, n(Rs)	4/4
Unaligned Load Halfword:_____	ulhu Rd, n(Rs)	4/4
Unaligned Load Word:_____	ulw Rd, n(Rs)	2/2
Unaligned Store Halfword:_____	ush Rd, n(Rs)	3/3
Unaligned Store Word:_____	usw Rd, n(Rs)	2/2

System I/O Services

Service	Code in \$v0	Arguments	Results
Print an Integer	1	\$a0 = Integer Value to be Printed	

Print Float	2		
Print Double	3		
Print a String	4	\$a0 = Address of String in Memory	
Read an Integer	5		Integer Returned in \$v0
Read Float	6		
Read Double	7		
Read a String	8	\$a0 = Address of Input Buffer in Memory \$a1 = Length of Buffer (n)	
Sbrk	9	\$a0 = amount	Address in \$v0
Exit	10		

The system call Read Integer reads an entire line of input from the keyboard up to and including the newline. Characters following the last digit in the decimal number are ignored. Read String has the same semantics as the Unix library routine fgets. It reads up to n – 1 characters into a buffer and terminates the string with a null byte. If fewer than n – 1 characters are on the current line, Read String reads up to and including the newline and again null-terminates the string. Print String will display on the terminal the string of characters found in memory starting with the location pointed to by the address stored in \$a0. Printing will stop when a null character is located in the string. Sbrk returns a pointer to a block of memory containing n additional bytes. Exit terminates the user program execution and returns control to the operating system.

ASCII Codes

dec	hex	Char	dec	hex	Char	dec	hex	Char	dec	hex	Char
0	00	null	32	20	sp	64	40	@	96	60	'
1	01	soh	33	21	!	65	41	A	97	61	a
2	02	stx	34	22	"	66	42	B	98	62	b
3	03	etx	35	23	#	67	43	C	99	63	c
4	04	eot	36	24	\$	68	44	D	100	64	d
5	05	enq	37	25	%	69	45	E	101	65	e
6	06	ack	38	26	&	70	46	F	102	66	f
7	07	bel	39	27	'	71	47	G	103	67	g
8	08	bs	40	28	(72	48	H	104	68	h
9	09	ht	41	29)	73	49	I	105	69	i
10	0a	nl	42	2a	*	74	4a	J	106	6a	j
11	0b	vt	43	2b	+	75	4b	K	107	6b	k
12	0c	np	44	2c	,	76	4c	L	108	6c	l
13	0d	cr	45	2d	-	77	4d	M	109	6d	m
14	0e	so	46	2e	.	78	4e	N	110	6e	n
15	0f	si	47	2f	/	79	4f	O	111	6f	o
16	10	dle	48	30	0	80	50	P	112	70	p
17	11	dc1	49	31	1	81	51	Q	113	71	q
18	12	dc2	50	32	2	82	52	R	114	72	r
19	13	dc3	51	33	3	83	53	S	115	73	s
20	14	dc4	52	34	4	84	54	T	116	74	t
21	15	nak	53	35	5	85	55	U	117	75	u
22	16	syn	54	36	6	86	56	V	118	76	v
23	17	etb	55	37	7	87	57	W	119	77	w
24	18	can	56	38	8	88	58	X	120	78	x
25	19	em	57	39	9	89	59	Y	121	79	y
26	1a	sub	58	3a	:	90	5a	Z	122	7a	z
27	1b	esc	59	3b	;	91	5b	[123	7b	{
28	1c	fs	60	3c	<	92	5c	\	124	7c	
29	1d	gs	61	3d	=	93	5d]	125	7d	}
30	1e	rs	62	3e	>	94	5e	^	126	7e	~
31	1f	us	63	3f	?	95	5f	_	127	7f	del