Table of ASCII Characters This table lists the ASCII characters and their decimal, octal and hexadecimal numbers. Characters which appear as names in parentheses (e.g., (nl)) are non-printing characters. A table of the common non-printing characters appears after this table.

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(nul)	0	0000	0x00	(sp)	32	0040	0x20	@	64	0100	0x40	‘	96	0140	0x60
(soh)	1	0001	0x01	!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
(stx)	2	0002	0x02	"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
(etx)	3	0003	0x03	#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
(eot)	4	0004	0x04	$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
(enq)	5	0005	0x05	%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
(ack)	6	0006	0x06	&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
(bel)	7	0007	0x07	’	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(bs)	8	0010	0x08	(	40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
(ht)	9	0011	0x09	)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
(nl)	10	0012	0x0a	*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
(vt)	11	0013	0x0b	+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
(np)	12	0014	0x0c	,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
(cr)	13	0015	0x0d	-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
(so)	14	0016	0x0e	.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
(si)	15	0017	0x0f	/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
(dle)	16	0020	0x10	0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
(dc1)	17	0021	0x11	1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
(dc2)	18	0022	0x12	2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
(dc3)	19	0023	0x13	3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
(dc4)	20	0024	0x14	4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
(nak)	21	0025	0x15	5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
(syn)	22	0026	0x16	6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
(etb)	23	0027	0x17	7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
(can)	24	0030	0x18	8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
(em)	25	0031	0x19	9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
(sub)	26	0032	0x1a	:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
(esc)	27	0033	0x1b	;	59	0073	0x3b	[	91	0133	0x5b	{	123	0173	0x7b
(fs)	28	0034	0x1c	<	60	0074	0x3c	\	92	0134	0x5c	|	124	0174	0x7c
(gs)	29	0035	0x1d	=	61	0075	0x3d	]	93	0135	0x5d	}	125	0175	0x7d
(rs)	30	0036	0x1e	>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
(us)	31	0037	0x1f	?	63	0077	0x3f	_	95	0137	0x5f	(del)	127	0177	0x7f

ASCII Name	Description	C Escape Sequence
nul	null byte	\0
bel	bell character	\a
bs	backspace	\b
ht	horizontal tab	\t
np	formfeed	\f
nl	newline	\n
cr	carriage return	\r
vt	vertical tab	
esc	escape	
sp	space	

Register Name	**Number**	**Usage**
zero	$0	Constant 0
$at	$1	Reserved for assembler
$v0–$v1	$2–$3	Expression evaluation and results of a function
$a0–$a3	$4–$7	Argument 1–4
$t0–$t7	$8–$15	Temporary (not preserved across call)
$s0–$s7	$16–$23	Saved temporary (preserved across call)
$t8–$t9	$24–$25	Temporary (not preserved across call)
$k0–$k1	$26–$27	Reserved for OS kernel
$gp	$28	Pointer to global area
$sp	$29	Stack pointer
$fp	$30	Frame pointer
$ra	$31	Return address (used by function call)

MIPS registers and the convention governing their use.

Arithmetic and Logical Instructions

Description	op-code	operands
Absolute Value	abs†	Rdest, Rsrc
Addition (with overflow)	add	Rdest, Rsrc1, Src2
Addition Immediate (with overflow)	addi	Rdest, Rsrc1, Imm
Addition (without overflow)	addu	Rdest, Rsrc1, Src2
Addition Immediate (without overflow)	addiu	Rdest, Rsrc1, Imm
AND	and	Rdest, Rsrc1, Src2
AND Immediate	andi	Rdest, Rsrc1, Imm
Divide (signed)	div	Rsrc1, Rsrc2
Divide (unsigned)	divu	Rsrc1, Rsrc2
Divide (signed, with overflow)	div†	Rdest, Rsrc1, Src2
Divide (unsigned, without overflow)	divu†	Rdest, Rsrc1, Src2
Multiply (without overflow)	mul†	Rdest, Rsrc1, Src2
Multiply (with overflow)	mulo†	Rdest, Rsrc1, Src2
Unsigned Multiply (with overflow)	mulou†	Rdest, Rsrc1, Src2
Multiply	mult	Rsrc1, Rsrc2
Unsigned Multiply	multu	Rsrc1, Rsrc2
Negate Value (with overflow)	neg†	Rdest, Rsrc
Negate Value (without overflow)	negu†	Rdest, Rsrc
NOR	nor	Rdest, Rsrc1, Src2
NOT	not†	Rdest, Rsrc
OR	or	Rdest, Rsrc1, Src2
OR Immediate	ori	Rdest, Rsrc1, Imm
Remainder	rem†	Rdest, Rsrc1, Src2
Unsigned Remainder	remu†	Rdest, Rsrc1, Src2
Rotate Left	rol†	Rdest, Rsrc1, Src2
Rotate Right	ror†	Rdest, Rsrc1, Src2
Shift Left Logical	sll	Rdest, Rsrc1, Src2
Shift Left Logical Variable	sllv	Rdest, Rsrc1, Rsrc2
Shift Right Arithmetic	sra	Rdest, Rsrc1, Src2
Shift Right Arithmetic Variable	srav	Rdest, Rsrc1, Rsrc2
Shift Right Logical	srl	Rdest, Rsrc1, Src2
Shift Right Logical Variable	srlv	Rdest, Rsrc1, Rsrc2
Subtract (with overflow)	sub	Rdest, Rsrc1, Src2
Subtract (without overflow)	subu	Rdest, Rsrc1, Src2
XOR	xor	Rdest, Rsrc1, Src2
XOR Immediate	xori	Rdest, Rsrc1, Imm

Branch and Jump Instructions

Description	op-code	operands
Branch instruction	b	label
Branch Coprocessor *z* True	bc*z*t	label
Branch Coprocessor *z* False	bc*z*f	label
Branch on Equal	beq	Rsrc1, Src2, label
Branch on Equal Zero	beqz†	Rsrc, label
Branch on Greater Than Equal	bge†	Rsrc1, Src2, label
Branch on GTE Unsigned	bgeu†	Rsrc1, Src2, label
Branch on Greater Than Equal Zero	bgez	Rsrc, label
Branch on Greater Than Equal Zero And Link	bgezal	Rsrc, label
Branch on Greater Than	bgt†	Rsrc1, Src2, label
Branch on Greater Than Unsigned	bgtu†	Rsrc1, Src2, label
Branch on Greater Than Zero	bgtz	Rsrc, label
Branch on Less Than Equal	ble†	Rsrc1, Src2, label
Branch on LTE Unsigned	bleu†	Rsrc1, Src2, label
Branch on Less Than Equal Zero	blez	Rsrc, label
Branch on Greater Than Equal Zero And Link	bgezal	Rsrc, label
Branch on Less Than And Link	bltzal	Rsrc, label
Branch on Less Than	blt†	Rsrc1, Src2, label
Branch on Less Than Unsigned	bltu†	Rsrc1, Src2, label
Branch on Less Than Zero	bltz	Rsrc, label
Branch on Not Equal	bne	Rsrc1, Src2, label
Branch on Not Equal Zero	bnez†	Rsrc, label
Jump	j	label
Jump and Link	jal	label
Jump and Link Register	jalr	Rsrc
Jump Register	jr	Rsrc

Data Movement Instructions

Description	op-code	operands
Move	move†	Rdest, Rsrc
Move From hi	mfhi	Rdest
Move From lo	mflo	Rdest
Move To hi	mthi	Rdest
Move To lo	mtlo	Rdest
Move From Coprocessor *z*	mfc*z*	Rdest, CPsrc
Move Double From Coprocessor 1	mfc1.d†	Rdest, FRsrc1
Move To Coprocessor *z*	mtc*z*	Rsrc, CPdest

Comparison Instructions

Description	op-code	operands
Set Equal	seq†	Rdest, Rsrc1, Src2
Set Greater Than Equal	sge†	Rdest, Rsrc1, Src2
Set Greater Than Equal Unsigned	sgeu†	Rdest, Rsrc1, Src2
Set Greater Than	sgt†	Rdest, Rsrc1, Src2
Set Greater Than Unsigned	sgtu†	Rdest, Rsrc1, Src2
Set Less Than Equal	sle†	Rdest, Rsrc1, Src2
Set Less Than Equal Unsigned	sleu†	Rdest, Rsrc1, Src2
Set Less Than	slt	Rdest, Rsrc1, Src2
Set Less Than Immediate	slti	Rdest, Rsrc1, Imm
Set Less Than Unsigned	sltu	Rdest, Rsrc1, Src2
Set Less Than Unsigned Immediate	sltiu	Rdest, Rsrc1, Imm
Set Not Equal	sne†	Rdest, Rsrc1, Src2

Constant-Manipulating Instructions

Description	op-code	operands
Load Immediate	li†	Rdest, imm
Load Upper Immediate	lui	Rdest, imm

Load Instructions

Description	op-code	operands
Load Address	la†	Rdest, address
Load Byte	lb	Rdest, address
Load Unsigned Byte	lbu	Rdest, address
Load Double-Word	ld†	Rdest, address
Load Halfword	lh	Rdest, address
Load Unsigned Halfword	lhu	Rdest, address
Load Word	lw	Rdest, address
Load Word Coprocessor	lwc$z$	Rdest, address
Load Word Left	lwl	Rdest, address
Load Word Right	lwr	Rdest, address
Unaligned Load Halfword	ulh†	Rdest, address
Unaligned Load Halfword Unsigned	ulhu†	Rdest, address
Unaligned Load Word	ulw†	Rdest, address

Store Instructions

Description	op-code	operands
Store Byte	sb	Rsrc, address
Store Double-Word	sd†	Rsrc, address
Store Halfword	sh	Rsrc, address
Store Word	sw	Rsrc, address
Store Word Coprocessor	swc*z*	Rsrc, address
Store Word Left	swl	Rsrc, address
Store Word Right	swr	Rsrc, address
Unaligned Store Halfword	ush†	Rsrc, address
Unaligned Store Word	usw†	Rsrc, address

Floating Point Instructions

Description	op-code	operands
Floating Point Absolute Value Double	abs.d	FRdest, FRsrc
Floating Point Absolute Value Single	abs.s	FRdest, FRsrc
Floating Point Addition Double	add.d	FRdest, FRsrc1, FRsrc2
Floating Point Addition Single	add.s	FRdest, FRsrc1, FRsrc2
Compare Equal Double	c.eq.d	FRsrc1, FRsrc2
Compare Equal Single	c.eq.s	FRsrc1, FRsrc2
Compare Less Than Equal Double	c.le.d	FRsrc1, FRsrc2
Compare Less Than Equal Single	c.le.s	FRsrc1, FRsrc2
Compare Less Than Double	c.lt.d	FRsrc1, FRsrc2
Compare Less Than Single	c.lt.s	FRsrc1, FRsrc2
Convert Single to Double	cvt.d.s	FRdest, FRsrc
Convert Integer to Double	cvt.d.w	FRdest, FRsrc
Convert Double to Single	cvt.s.d	FRdest, FRsrc
Convert Integer to Single	cvt.s.w	FRdest, FRsrc
Convert Double to Integer	cvt.w.d	FRdest, FRsrc
Convert Single to Integer	cvt.w.s	FRdest, FRsrc
Floating Point Divide Double	div.d	FRdest, FRsrc1, FRsrc2
Floating Point Divide Single	div.s	FRdest, FRsrc1, FRsrc2
Load Floating Point Double	l.d†	FRdest, address
Load Floating Point Single	l.s†	FRdest, address
Move Floating Point Double	mov.d	FRdest, FRsrc
Move Floating Point Single	mov.s	FRdest, FRsrc
Floating Point Multiply Double	mul.d	FRdest, FRsrc1, FRsrc2
Floating Point Multiply Single	mul.s	FRdest, FRsrc1, FRsrc2
Negate Double	neg.d	FRdest, FRsrc
Negate Single	neg.s	FRdest, FRsrc
Store Floating Point Double	s.d†	FRdest, address
Store Floating Point Single	s.s†	FRdest, address
Floating Point Subtract Double	sub.d	FRdest, FRsrc1, FRsrc2
Floating Point Subtract Single	sub.s	FRdest, FRsrc1, FRsrc2

Exception and Trap Instructions

Description	op-code	operands
Return From Exception	rfe	
System Call	syscall	
Break	break	n
No operation	nop	