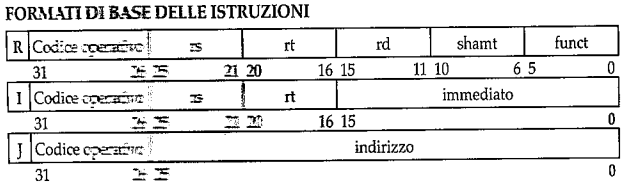
Scheda tecnica riassuntiva del MIPS

INSIEME DELLE ISTRUZIONI DI BASE

Nome		Formato	Operazione (in Verilog)		Codice Operativo/ Funz
Somma	add	R	R[rd] = R[rs] + R[rt]	(1)	$0/20_{esa}$
Somma immediata	addi	I	R[rt] = R[rs] + EstSegnoImm	(1,2)	8_{esa}
Somma immediata senza segno	addiu	I	R[rt] = R[rs] + EstSegnoImm	(2)	9_{esa}
Somma senza segno	addu	R	R[rd] = R[rs] + R[rt]		$0/21_{esa}$
And	and	R	R[rd] = R[rs] & R[rt]		$0/24_{esa}$
And immediato	andi	I	R[rt] = R[rs] & EstZeroImm	(3)	c_{esa}
Salto condizionato su uguaglianza	beq	I	if (R[rs] == R[rt]) PC = PC + 4 + + IndSaltoCond	(4)	4_{esa}
Salto condizionato su disuguaglianza	bne	I	if (R[rs] != R[rt]) PC = PC + 4 + + IndSaltoCond	(4)	5_{esa}
Salto incondizionato	j	J	PC = IndSaltoIncond	(5)	2_{esa}
Salta e unisci (*jump and link*)	jal	J	R[31] = PC + 4; PC = = IndSaltoIncond	(5)	3_{esa}
Salta a registro	jr	R	PC = R[rs]		$0/08_{esa}$
Carica un byte senza segno	lbu	I	R[rt] = {24'b0,M[R[rs]+ +EstSegnoImm](7:0)}	(2)	4_{esa}
Carica una mezza parola senza segno	lhu	I	R[rt] = {16'b0,M[R[rs]+ +EstSegnoImm](15:0)}	(2)	25_{esa}
Load linked	ll	I	R[rt] = M[R[rs]+EstSegnoImm]	(2,7)	30_{esa}
Carica la mezza parola superiore	lui	I	R[rt] = {imm, 16'b0}		f_{esa}
Carica una parola	lw	I	R[rt] = M[R[rs]+EstSegnoImm]	(2)	23_{esa}
Nor	nor	R	R[rd] = ~(R[rs] | R[rt])		$0/27_{esa}$
Or	or	R	R[rd] = R[rs] | R[rt]		$0/25_{esa}$
Or immediato	ori	I	R[rt] = R[rs] | EstZeroImm	(3)	d_{esa}
Imposta se minore	slt	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0		$0/2a_{esa}$
Imposta se minore immediato	slti	I	R[rd] = (R[rs] < EstSegnoImm) ? 1 : 0	(2)	a_{esa}
Imposta se minore immediato senza segno	sltiu	I	R[rd] = (R[rs] < EstSegnoImm) ? 1 : 0	(2, 6)	b_{esa}
Imposta se minore senza segno	sltu	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0	(6)	$0/2b_{esa}$
Scorrimento logico a sinistra	sll	R	R[rd] = R[rs] << shamt		$0/00_{esa}$
Scorrimento logico a destra	srl	R	R[rd] = R[rs] >> shamt		$0/02_{esa}$
Salva un byte	sb	I	M[R[rs]+EstSegnoImm](7:0) = = R[rt](7:0)	(2)	28_{esa}
Salva una mezza parola senza segno	lhu	I	M[R[rs]+EstSegnoImm] = R[rt]; R[rt] = (*atomica*) ? 1:0	(2,7)	38_{esa}
Salvataggio condizionato	ll	I	R[rt] = M[R[rs]+EstSegnoImm]	(2,7)	30_{esa}
Salvataggio di una mezza parola	lhw	I	R[rt] = M[R[rs]+EstSegnoImm] (15:0) = R[rt](15:0)	(2)	29_{esa}
Salvataggio di una parola	lw	I	R[rt] = M[R[rs]+EstSegnoImm] = = R[rt]	(2)	$2b_{esa}$
Sottrazione	sub	R	R[rd] = R[rs] - R[rt]	(1)	$0/22_{esa}$
Sottrazione senza segno	subu	R	R[rd] = R[rs] - R[rt]		$0/23_{esa}$

(1) Può causare l'eccezione di overflow
(2) EstSegnoImm = {16{immediato[15]}, immediato}
(3) EstZeroImm = {16{1b'0}, immediato}
(4) IndSaltoCond = {14{immediato[15]}, immediato, 2'b0}
(5) IndSaltoIncond = {PC+4[31:28], indirizzo, 2'b0}
(6) Operandi considerati come numeri senza segno, invece che numeri in complemento a 2
(7) Coppia di operazioni atomiche: test e impostazione di flag; R[rt] = 1 se la coppia di istruzioni è stata eseguita in modo atomica, 0 altrimenti.

FORMATI DI BASE DELLE ISTRUZIONI



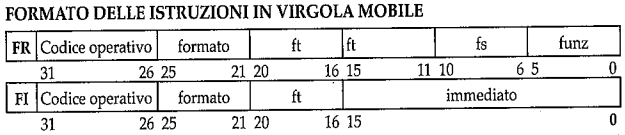
INSIEME DELLE ISTRUZIONI ARITMETICHE DI BASE

Nome		Formato	Operazione (in Verilog)		Codice Op/FMT/ FT/Funz
Salto cond. su uguaglianza su VM	bclt	FI	if(Fpcond) PC = PC+4+IndSalto	(4)	11/8/1 --
Salto cond. su disuguaglianza su VM	bclf	FI	if(!Fpcond) PC = PC+4+IndSalto	(4)	11/8/0 --
Divisione	div	R	Lo=R[rs]/R[rt]; Hi=R[rs]%R[rt]		0/--/--/1a
Divisione senza segno	divu	R	Lo=R[rs]/R[rt]; Hi=R[rs]%R[rt]		(6) 0/--/--/1b
Somma in in singola prec.	add.s	FR	F[fd]=F[fs]+F[ft]		11/10/--/0
Somma in VM in doppia prec.	add.d	FR	{F[fd],F[fd+1]}= {F[fs],F[fs+1]} +{F[ft],F[ft+1]}		11/11/--/0
Confronto in VM, in singola prec.	c.x.s*	FR	Fpcond = (F[fs] op F[ft]) ? 1 : 0		11/10/--/y
Confronto in VM, in doppia prec.	c.x.d*	FR	Fpcond = ({F[fs],F[fs+1]} op {F[ft],F[ft+1]}) ? 1 : 0		11/11/--/y

* (x è uguale agli operatori: eq, lt, o le) (op è uguale agli operatori: ==, < o <=) (y può assumere i valori: 32, 3c o 3e)

Divisione in VM in singola prec.	div.s	FR	F[fd] = F[fs] / F[ft]		11/10/--/3
Divisione in VM in doppia prec.	div.d	FR	{F[fd],F[fd+1]} = = {F[fs],F[fs+1]} / {F[ft],F[ft+1]}		11/11/--/3
Sottrazione in VM in singola prec.	sub.s	FR	F[fd] = F[fs] - F[ft]		11/10/--/2
Sottrazione in VM in doppia prec.	sub.d	FR	{F[fd],F[fd+1]} = {F[fs],F[fs+1]} - {F[ft],F[ft+1]}		11/11/--/2
Caricamento parola in VM in singola prec.	lwc1	I	F[rt]=M[R[rs]+EstSegnoImm]	(2)	31/--/--/--
Caricamento parola in VM in doppia prec.	ldc1	I	F[rt]=M[R[rs]+EstSegnoImm]; F[rt+1]=M[R[rs]+EstSegnoImm+4]	(2)	35/--/--/--
Copia da Hi	mfhi	R	R[rd] = Hi		0/--/--/10
Copia da Lo	mflo	R	R[rd] = Lo		0/--/--/12
Copia da Controllo	mfc0	R	R[rd] = CR[rs]		10/0/--/10
Moltiplicazione	mult	R	{Hi,Lo} = R[rs] * R[rt]		10/0/--/0
Moltiplicazione senza segno	multu	R	{Hi,Lo} = R[rs] * R[rt]		(6) 0/--/--/18
Scorrimento a destra aritmetico	sra	R	R[rd] = R[rt] >> shamt		0/--/--/3
Salvataggio in VM in singola prec.	swc1	I	M[R[rs]+EstSegnoImm]=F[rt]	(2)	39/--/--/--
Salvataggio in VM in doppia prec.	sdc1	I	M[R[rs]+EstSegnoImm]=F[rt]; M[R[rs]+EstSegnoImm+4]=F[rt+1]	(2)	3d/--/--/--

FORMATO DELLE ISTRUZIONI IN VIRGOLA MOBILE



INSIEME DELLE PSEUDOISTRUZIONI

NOME	NOME SIMBOLICO	OPERAZIONE
Salta se minore	blt	if (R[rs] < R[rt]) PC = Etichetta
Salta se maggiore	bgt	if (R[rs] > R[rt]) PC = Etichetta
Salta se minore uguale	ble	if (R[rs] <= R[rt]) PC = Etichetta
Salta se maggiore uguale	bge	if (R[rs] >= R[rt]) PC = Etichetta
Caricamento immediato	li	R[rd] = immediato
Copia	move	R[rd] = R[rs]

NOME DEI REGISTRI, NUMERO, UTILIZZO, CONVENZIONI DI CHIAMATA

Nome	Numero	Utilizzo	Conservato dalla chiamata?
$zero	0	Il valore costante 0	N.A.
$at	1	Temporaneo per l'assemblatore	No
$v0-$v1	2-3	Valore restituito da funzione o dalla valutazione di espressione	No
$a0-$a3	4-7	Argomenti	No
$t0-$t7	8-15	Registri per variabili temporanee	No
$s0-$s7	16-23	Registri di variabile	Sì
$t8-$t9	24-25	Registri per variabili temporanee	No
$k0-$k1	26-27	Riservati al kernel del SO	No
$gp	28	Global pointer	Sì
$sp	29	Stack pointer	Sì
$fp	30	Frame pointer	Sì
$ra	31	Indirizzo di ritorno	Sì

CODICI OPERATIVI, BASE, CONVERSIONE, SIMBOLI ASCII

Codice Op MIPS (31:26)	Campo Funz MIPS (1) (5:0)	Campo Funz MIPS (2) (5:0)	Binario	Decimale	Esadec.	Carattere ASCII	Decimale	Esadec.	Carattere ASCII
(1)	sll	add.*f*	00 0000	0	0	NULL	64	40	@
		sub.*f*	00 0001	1	1	SOH	65	41	A
j	srl	mul.*f*	00 0010	2	2	STX	66	42	B
jal	sra	div.*f*	00 0011	3	3	ETX	67	43	C
beq	sllv	sqrt.*f*	00 0100	4	4	EOT	68	44	D
bne		abs.*f*	00 1001	5	5	ENQ	69	45	E
blez	srlv	mov.*f*	00 0110	6	6	ACK	70	46	F
bgtz	srav	neg.*f*	00 0111	7	7	BEL	71	47	G
addi	jr		00 1000	8	8	BS	72	48	H
addiu	jalr		00 1001	9	9	HT	73	49	I
slti	movz		00 1010	10	a	LF	74	4a	J
sltiu	movn		00 1011	11	b	VT	75	4b	K
andi	syscall	round.w.*f*	00 1100	12	c	FF	76	4c	L
ori	break	trunc.w.*f*	00 1101	13	d	CR	77	4d	M
xori		ceil.w.*f*	00 1110	14	e	SO	78	4e	N
lui	sync	floor.w.*f*	00 1111	15	f	SI	79	4f	O
	mfhi		01 0000	16	10	DLE	80	50	P
(2)	mthi		01 0001	17	11	DC1	81	51	Q
	mflo	movz.*f*	01 0010	18	12	DC2	82	52	R
	mtlo	movn.*f*	01 0011	19	13	DC3	83	53	S
			01 0100	20	14	DC4	84	54	T
			01 0101	21	15	NAK	85	55	U
			01 0110	22	16	SYN	86	56	V
			01 0111	23	17	ETB	87	57	W
	mult		01 1000	24	18	CAN	88	58	X
	multu		01 1001	25	19	EM	89	59	Y
	div		01 1010	26	1a	SUB	90	5a	Z
	divu		01 1011	27	1b	ESC	91	5b	[
			01 1100	28	1c	FS	92	5c	\
			01 1101	29	1d	GS	93	5d	]
			01 1110	30	1e	RS	94	5e	^
			01 1111	31	1f	US	95	5f	_
lb	add	cvt.s.*f*	10 0000	32	20	Spazio	96	60	`
lh	addu	cvt.d.*f*	10 0001	33	21	!	97	61	a
lwl	sub		10 0010	34	22	"	98	62	b
lw	subu		10 0011	35	23	#	99	63	c
lbu	and	cvt.s.*f*	10 0100	36	24	$	100	64	d
lhu	or	cvt.*d.f*	10 0101	37	25	%	101	65	e
lwr	xor		10 0110	38	26	&	102	66	f
	nor		10 0111	39	27	'	103	67	g
sb			10 1000	40	28	(	104	68	h
sh			10 1001	41	29	)	105	69	i
swl	slt		10 1010	42	2a	*	106	6a	j
sw	sltu		10 1011	43	2b	+	107	6b	k
			10 1100	44	2c	,	108	6c	l
			10 1101	45	2d	-	109	6d	m
swr			10 1110	46	2e	.	110	6e	n
cache			10 1111	47	2f	/	111	6f	o
ll	tge	c.*f*.*f*	11 0000	48	30	0	112	70	p
lwc1	tgeu	c.un.*f*	11 0001	49	31	1	113	71	q
lwc2	tlt	c.eq.*f*	11 0010	50	32	2	114	72	r
pref	tltu	c.ueq.*f*	11 0011	51	33	3	115	73	s
	teq	c.olt.*f*	11 0100	52	34	0	116	74	t
ldc1		c.ult.*f*	11 0101	53	35	1	117	75	u
ldc2	tne	c.ole.*f*	11 0110	54	36	2	118	76	v
		c.ule.*f*	11 0111	55	37	3	119	77	w
sc		c.sf.*f*	11 1000	56	38	4	120	78	x
swc1		c.ngle.*f*	11 1001	57	39	5	121	79	y
swc2		c.seq.*f*	11 1010	58	3a	6	122	7a	z
		c.ngl.*f*	11 1011	59	3b	7	123	7b	{
		c.lt.*f*	11 1100	60	3c	8	124	7c	|
sdc1		c.nge.*f*	11 1101	61	3d	9	125	7d	}
sdc2		c.le.*f*	11 1110	62	3e	:	126	7e	~
		c.ngt.*f*	11 1111	63	3f	;	127	7f	DEL

(1) codice operativo(31:26) == 0
(2) codice operativo(31:26) == $17_{dec}(11_{esa})$; if formato(25:21) == $16_{dec}(10_{esa})$ *f* = s (singola); if formato(25:21) == $17_{dec}(11_{esa})$ *f* = d (doppia);

STANDARD IEEE754 DEI NUMERI IN VIRGOLA MOBILE

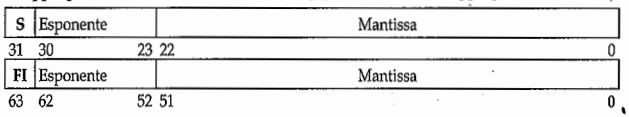
$(-1)^S \times (1 + \text{mantissa}) \times 2^{(\text{Esponente} - \text{Polarizzazione})}$
Dove la polarizzazione in singola precisione = 127,
in doppia precisione = 1023.

Simboli in IEEE754

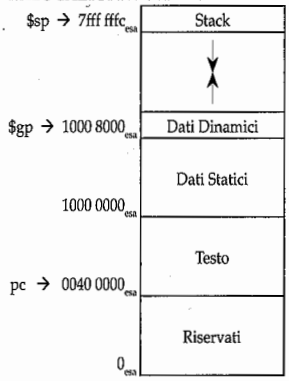
Esponente	Mantissa	Significato
0	0	± 0
0	≠ 0	± denormalizzato
Da 1 a MAX-1	qualsiasi	± numero in VM
MAX	0	± ∞
MAX	≠ 0	NaN

Sing. prec. MAX = 255; Doppia prec. MAX = 2047

Formati IEEE in singola e doppia precisione

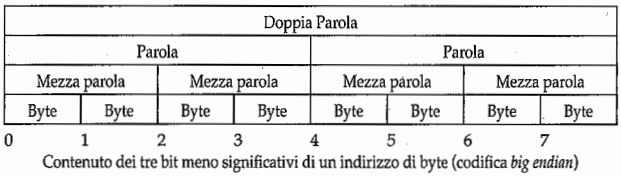


ALLOCAZIONE DELLA MEMORIA



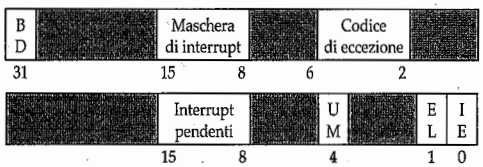
FRAME DI STACK

ALLINEAMENTO DEI DATI



Contenuto dei tre bit meno significativi di un indirizzo di byte (codifica *big endian*)

REGISTRI DI GESTIONE DELLE ECCEZIONI: CAUSA E STATO



BD = Branch Delay; UM = Modalità Utente, EL = Livello Eccezione, IE = Interrupt Enable.

CODICI DELLE ECCEZIONI

Numero	Nome	Causa dell'eccezione	Numero	Nome	Causa dell'eccezione
0	Int	Interrupt (hardware)	9	Bp	Eccezione di breakpoint
4	AdEL	Eccezione di errore nell'indirizzo (caricamento dati o fetch istruzione)	10	RI	Eccezione di istruzione riservata
5	AdES	Eccezione di errore nell'indirizzo (memorizzazione)	11	CpU	Coprocessore non implementato
6	IBE	Errore sul bus nel fetch di un'istruzione	12	Ov	Eccezione di overflow aritmetico
7	DBE	Errore sul bus in una load o store	13	Tr	Trap
8	Sys	Eccezione di Syscall	15	FPE	Eccezione Floating Point (VM)

PREFISSI DELLE DIMENSIONI

Dimensione	Prefisso	Simbolo	Dimensione	Prefisso	Simbolo	Dimensione	Prefisso	Simbolo	Dimensione	Prefisso	Simbolo
10^3	Kilo	K	2^{10}	Kibi	Ki	10^{15}	Peta	P	2^{50}	Pebi	Pi
10^6	Mega	M	2^{20}	Mebi	Mi	10^{18}	Exa	E	2^{60}	Exbi	Ei
10^9	Giga	G	2^{30}	Gibi	Gi	10^{21}	Zetta	Z	2^{70}	Zebi	Zi
10^{12}	Tera	T	2^{40}	Tebi	Ti	10^{24}	Yotta	Y	2^{80}	Yobi	Yi