Sub-conjunto de instruções do MIPS

Grupo	Sintaxe		Tipo	Op	Func	Comentário
Transferência de informação	lb	Rdest, Imm16(Rsrc)	I	0x20	-	load byte from memory
	lw	Rdest, Imm16(Rsrc)	I	0x23	-	load word from memory
	lbu	Rdest, Imm16(Rsrc)	I	0x24	-	load unsigned byte from memory
	sb	Rsrc2, Imm16(Rsrc1)	I	0x28	-	store byte to memory
	sw	Rsrc2, Imm16(Rsrc1)	I	0x2b	-	store word to memory
	lui	Rdest, Imm16	I	0x0f	-	load upper immediate
	mfhi	Rdest	R	00	0x10	move from hi
	mflo	Rdest	R	00	0x12	move from lo
	mthi	Rsrc	R	00	0x11	move to hi
	mtlo	Rsrc	R	00	0x13	move to lo
	li	Rdest, Imm16/32				load immediate
	la	Rdest, address16/32				load address
	move	Rdest, Rsrc				move
Operações aritméticas	add	Rdest, Rsrc1, Rsrc2	R	00	0x20	addition (with overflow)
	addi	Rdest, Rsrc1, Imm16	I	08	-	addition imm. (with ov.)
	addu	Rdest, Rsrc1, Rsrc2	R	00	0x21	addition (without overflow)
	addiu	Rdest, Rsrc1, Imm16	I	09	-	addition imm. (without ov.)
	sub	Rdest, Rsrc1, Rsrc2	R	00	0x22	subtract (with overflow)
	subu	Rdest, Rsrc1, Rsrc2	R	00	0x23	subtract (without overflow)
	mult	Rsrc1, Rsrc2	R	00	0x18	multiply
	multu	Rsrc1, Rsrc2	R	00	0x19	unsigned multiply
	div	Rsrc1, Rsrc2	R	00	0x1a	divide
	divu	Rsrc1, Rsrc2	R	00	0x1b	unsigned divide
	abs	Rdest, Rsrc				absolute value
	mul	Rdest, Rsrc1, Rsrc2				multiply
	div	Rdest, Rsrc1, Rsrc2				divide
	rem	Rdest, Rsrc1, Rsrc2				remainder
Operações lógicas e de comparação	and	Rdest, Rsrc1, Rsrc2	R	00	0x24	AND
	andi	Rdest, Rsrc1, Imm16	I	0x0c	-	AND immediate
	or	Rdest, Rsrc1, Rsrc2	R	00	0x25	OR
	ori	Rdest, Rsrc1, Imm16	I	0x0d	-	OR immediate
	xor	Rdest, Rsrc1, Rsrc2	R	00	0x26	XOR
	xori	Rdest, Rsrc1, Imm16	I	0x0e	-	XOR immediate
	nor	Rdest, Rsrc1, Rsrc2	R	00	0x27	NOR
	slt	Rdest, Rsrc1, Rsrc2	R	00	0x2a	set if less than
	slti	Rdest, Rsrc1, Imm16	I	0x0a	-	set if less than immediate
	sltu	Rdest, Rsrc1, Rsrc2	R	00	0x2b	set if less than unsigned
	sltiu	Rdest, Rsrc1, Imm16	I	0x0b	-	set if less than unsigned imm.
	not	Rdest, Rsrc				not
Operações de deslocamento de bits	sll	Rdest, Rsrc1, Shamt5	R	00	00	shift left logical
	srl	Rdest, Rsrc1, Shamt5	R	00	02	shift right logical
	sra	Rdest, Rsrc1, Shamt5	R	00	03	shift right arithmetic
	rol	Rdest, Rsrc1, Rsrc2				rotate left
	ror	Rdest, Rsrc1, Rsrc2				rotate right
Instruções de salto	j	address28	J	02	-	jump (absolute addr)
	jr	Rsrc	R	00	08	jump register
	beq	Rsrc1, Rsrc2, address18	I	04	-	branch on equal (relative addr)
	bne	Rsrc1, Rsrc2, address18	I	05	-	branch on not equal (relative addr)
	bgez	Rsrc, address18	I	01	01*	br. on greater than equal zero (")
	bgtz	Rsrc, address18	I	07	-	br. on greater than zero (")
	blez	Rsrc, address18	I	06	-	br. on less than equal zero (")
	bltz	Rsrc, address18	I	01	00*	br. on less than zero (")
	jal	address28	J	03	-	jump and link (absolute addr)
	jalr	Rsrc	R	00	09	jump and link register
	b	address18/32				branch inconditional (relative addr)
	b<cnd>	Rsrc1, Rsrc2, address18/32				br. on <cnd> = [gt, ge, lt, le] (")
	b<cnd>u	Rsrc1, Rsrc2, address18/32				br. on <cnd> = [gt, ge, lt, le] uns. (")
Excepção	rfe		R	0x10	0x20	return from exception
	syscall		R	00	0x0c	system call
	break	code20	R	00	0x0d	break

* especificado no campo rt

Formatos das instruções e exemplos

Instrução	Tipo	Op/6 bits	Rs/5 bits	Rt/5 bits	Rd/5 bits	Sa/5 bits	Func/6 bits
add $a0, $t0, $s0	R	0	8 ($t0)	0x10 ($s0)	4 ($a0)	0	0x20
addi $8, $9, 0x100	I	8	9	8	0x100		
sw $6, -4 ($7)	I	0x2b	7	6	0xfffc (-4)		
beq $4, $5, 0x60	I	4	4	5	0x18 (0x60/4)		
j 0x80000	J	2	0x20000 (0x80000/4)				

Convenção dos registos do MIPS

Nome	Número	Utilização	Preservado na chamada?
$zero	0	Constante 0	n.a.
$v0-$v1	2-3	Valores para resultados e avaliação de expressões	Não
$a0-$a3	4-7	Argumentos	Sim
$t0-$t7	8-15	Temporários	Não
$s0-$s7	16-23	Seguros	Sim
$t8-$t9	24-25	Mais temporários	Não
$gp	28	Apontador global	Sim
$sp	29	Apontador para pilha	Sim
$fp	30	Apontador para a *frame*	Sim
$ra	31	Endereço de retorno	Sim

O registo 1 ($at) está reservado para o *assemler*, os registos 26-27 ($k0-$k1) estão reservados para o sistema operativo