

Guia de instrução rápida

1

CONJUNTO DE INSTRUÇÕES BÁSICO

NOME	MNEMÔNICO	FORMATO	OPERAÇÃO (EM VERILOG)	OPCODE/FUNCT (HEXA)
Add	add	R	$R[rd] = R[rs] + R[rt]$	(1) 0/20 _{hexa}
Add Immediate	addi	I	$R[rt] = R[rs] + \text{ImSinExt}$	(1)(2) 8 _{hexa}
Add Imm. Unsigned	addiu	I	$R[rt] = R[rs] + \text{ImSinExt}$	(2) 9 _{hexa}
Add Unsigned	addu	R	$R[rd] = R[rs] + R[rt]$	0/21 _{hexa}
And	and	R	$R[rd] = R[rs] \& R[rt]$	0/24 _{hexa}
And Immediate	andi	I	$R[rt] = R[rs] \& \text{ImZeroExt}$	(3) c _{hexa}
Branch On Equal	beq	I	if($R[rs] == R[rt]$) PC=PC+4+EndBranch	(4) 4 _{hexa}
Branch On Not Equal	bne	I	if($R[rs] != R[rt]$) PC=PC+4+EndBranch	(4) 5 _{hexa}
Jump	j	J	PC=EndJump	(5) 2 _{hexa}
Jump And Link	jal	J	$R[31] = \text{PC} + 4$; PC=EndJump	(5) 3 _{hexa}
Jump Register	jr	R	PC= $R[rs]$	0/08 _{hexa}
Load Byte Unsigned	lbu	I	$R[rt] = \{24'b0, M[R[rs] + \text{ImSinExt}](7:0)\}$	(2) 0/24 _{hexa}
Load Halfword Unsigned	lhu	I	$R[rt] = \{16'b0, M[R[rs] + \text{ImSinExt}](15:0)\}$	(2) 0/25 _{hexa}
Load Upper Imm.	lui	I	$R[rt] = \{\text{imm}, 16'b0\}$	f _{hexa}
Load Word	lw	I	$R[rt] = M[R[rs] + \text{ImSinExt}]$	(2) 0/23 _{hexa}
Nor	nor	R	$R[rd] = \sim(R[rs]) \& \sim(R[rt])$	0/27 _{hexa}
Or	or	R	$R[rd] = R[rs] R[rt]$	0/25 _{hexa}
Or Immediate	ori	I	$R[rt] = R[rs] \text{ImZeroExt}$	(3) d _{hexa}
Set Less Than	slt	R	$R[rd] = (R[rs] < R[rt]) ? 1 : 0$	0/2a _{hexa}
Set Less Than Imm.	slti	I	$R[rt] = (R[rs] < \text{ImSinExt}) ? 1 : 0$	(2) a _{hexa}
Set Less Than Imm. Unsigned	sltiu	I	$R[rt] = (R[rs] < \text{ImSinExt}) ? 1 : 0$	(2)(6) b _{hexa}
Set Less Than Unsigned	sltu	R	$R[rd] = (R[rs] < R[rt]) ? 1 : 0$	(6) 0/2b _{hexa}
Shift Left Logical	sll	R	$R[rd] = R[rs] \ll \text{shamt}$	0/00 _{hexa}
Shift Right Logical	srl	R	$R[rd] = R[rs] \gg \text{shamt}$	0/02 _{hexa}
Store Byte	sb	I	$M[R[rs] + \text{ImSinExt}](7:0) = R[rt](7:0)$	(2) 28 _{hexa}
Store Halfword	sh	I	$M[R[rs] + \text{ImSinExt}](15:0) = R[rt](15:0)$	(2) 29 _{hexa}
Store Word	sw	I	$M[R[rs] + \text{ImSinExt}] = R[rt]$	(2) 2b _{hexa}
Subtract	sub	R	$R[rd] = R[rs] - R[rt]$	(1) 0/22 _{hexa}
Subtract Unsigned	subu	R	$R[rd] = R[rs] - R[rt]$	0/23 _{hexa}

(1) Pode causar exceção de overflow

(2) $\text{ImSinExt} = \{16\{\text{imediato}[15]\}, \text{imediato}\}$

(3) $\text{ImZeroExt} = \{16\{1b'0\}, \text{imediato}\}$

(4) $\text{EndBranch} = \{14\{\text{imediato}[15]\}, \text{imediato}, 2'b0\}$

(5) $\text{JumpAddr} = \{\text{PC}[31:28], \text{address}, 2'b0\}$

(6) Operandos considerados números sem sinal (vs. compl. 2)

FORMATOS DE INSTRUÇÃO BÁSICOS

R	opcode	rs	rt	Rd	shamt	funct
	31	26 25	21 20	16 15	11 10	6 5 0
I	opcode	rs	rt	imediato		
	31	26 25	21 20	16 15		0
J	opcode	Endereço				
	31	26 25				0

2

CONJUNTO DE INSTRUÇÕES BÁSICAS ARITMÉTICAS

NOME	MNEMÔNICO	FORMATO	OPERAÇÃO	OPCODE/FMT/FT/FUNCT (HEXA)
Branch On FP True	belt	FI	if(FPcond) PC=PC+4+EndBranch (4)	11/8/1/-
Branch On FP False	bcif	FI	if(! FPcond) PC=PC+4+EndBranch (4)	11/8/0/-
Divide	div	R	$\text{Lo} = R[rs] / R[rt]$; $\text{Hi} = R[rs] \% R[rt]$	0/-/-/1a
Divide Unsigned	divu	R	$\text{Lo} = R[rs] / R[rt]$; $\text{Hi} = R[rs] \% R[rt]$ (6)	0/-/-/1b
FP Add Single	add.s	FR	$F[fd] = F[fs] + F[ft]$	11/10/-/0
FP Add Double	Add.d	FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} + \{F[ft], F[ft+1]\}$	11/11/-/0
FP Compare Single	C.x.s*	FR	$\text{FPcond} = (F[fs] \text{ op } F[ft]) ? 1 : 0$	11/10/-/y
FP Compare Double	C.x.d*	FR	$\text{FPcond} = (\{F[fs], F[fs+1]\} \text{ op } \{F[ft], F[ft+1]\}) ? 1 : 0$	11/11/-/y
* (x é eq, lt ou le) (op é ==, < ou <=) (y é 32, 3c ou 3e)				
FP Divide Single	div.s	FR	$F[fd] = F[fs] / F[ft]$	11/10/-/3
FP Divide Double	div.d	FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} / \{F[ft], F[ft+1]\}$	11/11/-/3
FP Multiply Single	mul.s	FR	$F[fd] = F[fs] * F[ft]$	11/10/-/2
FP Multiply Double	mul.d	FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} * \{F[ft], F[ft+1]\}$	11/11/-/2
FP Subtract Single	sub.s	FR	$F[fd] = F[fs] - F[ft]$	11/10/-/1
FP Subtract Double	sub.d	FR	$\{F[fd], F[fd+1]\} = \{F[fs], F[fs+1]\} - \{F[ft], F[ft+1]\}$	11/11/-/1
Load FP Single	lwc1	I	$F[rt] = M[R[rs] + \text{ImSinExt}]$ (2)	31/-/-/-
Load FP Double	ldc1	I	$F[rt] = M[R[rs] + \text{ImSinExt}]$; $F[rt+1] = M[R[rs] + \text{ImSinExt} + 4]$ (2)	35/-/-/-
Move From Hi	mfhi	R	$R[rd] = \text{Hi}$	0/-/-/10
Move From Lo	mflo	R	$R[rd] = \text{Lo}$	0/-/-/12
Move From Control	mfc0	R	$R[rd] = \text{CR}[rs]$	16/0/-/0
Multiply	mult	R	$\{\text{Hi}, \text{Lo}\} = R[rs] * R[rt]$	0/-/-/18
Multiply Unsigned	multu	R	$\{\text{Hi}, \text{Lo}\} = R[rs] * R[rt]$ (6)	0/-/-/19
Store FP Single	swc1	I	$M[R[rs] + \text{ImSinExt}] = F[rt]$ (2)	39/-/-/-
Store FP Double	sdc1	I	$M[R[rs] + \text{ImSinExt}] = F[rt]$; (2) $M[R[rs] + \text{ImSinExt} + 4] = F[rt+1]$	3d/-/-/-

FORMATOS DAS INSTRUÇÕES DE PONTO FLUTUANTE

FR	opcode	fnt	ft	fs	fd	funct
	31	26 25	21 20	16 15	11 10	6 5 0
FI	opcode	fnt	ft	imediato		
	31	26 25	21 20	16 15		0

CONJUNTO DE PSEUDO-INSTRUÇÕES

NOME	MNEMÔNICO	OPERAÇÃO
Branch Less Than	blt	if($R[rs] < R[rt]$) PC = Label
Branch Greater Than	bgt	if($R[rs] > R[rt]$) PC = Label
Branch Less Than or Equal	bge	if($R[rs] \leq R[rt]$) PC = Label
Branch Greater Than or Equal	bgt	if($R[rs] \geq R[rt]$) PC = Label
Load Immediate	li	$R[rd] = \text{imediato}$
Move	Move	$R[rd] = R[rs]$

NOME DO REGISTRADOR, NÚMERO, USO E CONVENÇÃO DE CHAMADA

NOME	NÚMERO	USO	PRESERVADO ENTRE CHAMADAS?
\$zero	0	O valor constante 0	N.A.
\$at	1	Temporário do montador	Não
\$v0-\$v1	2-3	Valores para resultados de função e avaliação de expressão	Não
\$a0-\$a3	4-7	Argumentos	Não
\$t0-\$t7	8-15	Temporários	Não
\$s0-\$s7	16-23	Temporários salvos	Sim
\$t8-\$t9	24-25	Temporários	Não
\$k0-\$k1	26-27	Reservado para kernel do SO	Não
\$gp	28	Ponteiro global	Sim
\$sp	29	Stack Pointer	Sim
\$fp	30	Frame Pointer	Sim
\$ra	31	Endereço de retorno	Sim

```
(1) opcode(31:26)==0
(2) opcode(31:26) = 17dec (11hexa); if fmt(25:21)==16dec (10hexa) f = s (single); if
    fmt(25:21)==17dec (11hexa) f = d (double)
```