

Instruções da Máquina Nativa				Instruções da Máquina Virtual				DETI-UA - ACI			
Transferência Memória-Registro ( <i>Load</i> )		Cálculo c/ Inteiros: Operações Aritméticas		Transferência Memória-Registro ( <i>Load</i> )		Salto Relativo ( <i>Branch</i> )					
lb	Rdst, addr	add	Rdst, Rsrc1, Rsrc2	l.d	FPdst, addr	b	Label				
lbu	Rdst, addr	addi	Rdst, Rsrc, Imm	l.s	FPdst, addr	beqz	Rsrc, Label				
lw	Rdst, addr	addiu	Rdst, Rsrc, Imm			bnez	Rsrc, Label				
lwcz	CReg, addr	addu	Rdst, Rsrc1, Rsrc2	Transferência Registro-Memória ( <i>Store</i> )		bge	Rsrc, Src, Label				
Transferência Registro-Memória ( <i>Store</i> )		div	Rsrc1, Rsrc2	s.d	FPsrc, addr	bgeu	Rsrc, Src, Label				
sb	Rsrc, addr	divu	Rsrc1, Rsrc2	s.s	FPsrc, addr	bgt	Rsrc, Src, Label				
sw	Rsrc, addr	mult	Rsrc1, Rsrc2	Transferência Registro-Registro ( <i>Move</i> )		bgtu	Rsrc, Src, Label				
swcz	Creg, addr	multu	Rsrc1, Rsrc2	move	Rdst, Rsrc	ble	Rsrc, Src, Label				
		sub	Rdst, Rsrc1, Rsrc2			bleu	Rsrc, Src, Label				
		subu	Rdst, Rsrc1, Rsrc2			blt	Rsrc, Src, Label				
Transferência Registro-Registro ( <i>Move</i> )		Cálculo c/ Inteiros: Op. Lógicas Bitwise		Manipulação de Const. ( <i>Load Imm/sym</i> )		bltu	Rsrc, Src, Label				
mfhi	Rdst	and	Rdst, Rsrc1, Rsrc2	la	Rdst, sym	beq	Rsrc, Src, Label				
mflo	Rdst	andi	Rdst, Rsrc, Imm	li	Rdst, IMM	bne	Rsrc, Src, Label				
mthi	Rsrc	nor	Rdst, Rsrc1, Rsrc2	l.d	FPdst, sym						
mtlo	Rsrc	or	Rdst, Rsrc1, Rsrc2	l.s	FPdst, sym						
mfcz	Rdst, Creg	ori	Rdst, Rsrc, Imm	Cálculo c/ Inteiros: Op. Aritméticas							
mtcz	Rsrc, Creg	xor	Rdst, Rsrc1, Rsrc2	abs	Rdst, Rsrc						
mov.d	FPdst, FPsrc	xori	Rdst, Rsrc, Imm	div	Rdst, Rsrc, Src						
mov.s	FPdst, FPsrc	Cálculo c/ Inteiros: Operações de Shift		divu	Rdst, Rsrc, Src						
Manipulação de Const. ( <i>Load Immediate</i> )		sll	Rdst, Rsrc1, Imm5	mul	Rdst, Rsrc, Src						
lui	Rdst, Imm	sllv	Rdst, Rsrc1, Rsrc2	mulu	Rdst, Rsrc, Src						
Instruções de Comparação		sra	Rdst, Rsrc1, Imm5	mulo	Rdst, Rsrc, Src						
slt	Rdst, Rsrc1, Rsrc2	srav	Rdst, Rsrc1, Rsrc2	mulou	Rdst, Rsrc, Src						
sltu	Rdst, Rsrc1, Rsrc2	srl	Rdst, Rsrc1, Imm5	neg	Rdst, Rsrc						
slti	Rdst, Rsrc, Imm	srlv	Rdst, Rsrc1, Rsrc2	negu	Rdst, Rsrc						
sltiu	Rdst, Rsrc, Imm	Cálculo em Vírgula Flutuante		rem	Rdst, Rsrc, Src						
Salto Relativo ( <i>Branch</i> ) e Absoluto ( <i>Jump</i> )		abs.p	FPdst, FPsrc	remu	Rdst, Rsrc, Src						
bczf	Label	add.p	FPdst, FPsrc1, FPsrc2	Cálculo c/ Inteiros: Op. Lógicas Bitwise							
bczt	Label	c.eq.p	FPsrc1, FPsrc2	not	Rdst, Rsrc						
beq	Rsrc1, Rsrc2, Label	c.le.p	FPsrc1, FPsrc2	Cálculo c/ Inteiros: Operações de Rotate							
bgez	Rsrc, Label	c.lt.p	FPsrc1, FPsrc2	rol	Rdst, Rsrc, Src						
bgezal	Rsrc, Label	cvt.d.s	FPdst, FPsrc	ror	Rdst, Rsrc, Src						
bgtz	Rsrc, Label	cvt.d.w	FPdst, FPsrc	Instruções de Comparação							
blez	Rsrc, Label	cvt.s.d	FPdst, FPsrc	seq	Rdst, Rsrc, Src						
bltz	Rsrc, Label	cvt.s.w	FPdst, FPsrc	sge	Rdst, Rsrc, Src						
bltzal	Rsrc, Label	cvt.w.d	FPdst, FPsrc	sgeu	Rdst, Rsrc, Src						
bne	Rsrc1, Rsrc2, Label	cvt.w.s	FPdst, FPsrc	sgt	Rdst, Rsrc, Src						
j	Label	div.p	FPdst, FPsrc1, FPsrc2	sgtu	Rdst, Rsrc, Src						
jal	Label	mul.p	FPdst, FPsrc1, FPsrc2	sle	Rdst, Rsrc, Src						
jalr	Rsrc	neg.p	FPdst, FPsrc	sleu	Rdst, Rsrc, Src						
jrr	Rsrc	sub.p	FPdst, FPsrc1, FPsrc2	sne	Rdst, Rsrc, Src						
		Manipulação de Exceções e Traps									
		break	n								
		nop									
		eret									
		syscall									

Tabela I: Registos do MIPS e convenção de uso		
Nome Lóg.	Nome Real	Uso Convencionado
\$zero	\$0	Constante 0
\$at	\$1	Reservado pelo assembler
\$v0..\$v1	\$2..\$3	Cálculo de expressões e valor de retorno das
\$a0..\$a3	\$4..\$7	Primeiros 4 parâmetros das funções
\$t0..\$t7	\$8..\$15	Geral (não são preservados pelas funções)
\$s0..\$s7	\$16..\$23	Geral (não podem ser alterados pelas funções)
\$t8..\$t9	\$24..\$25	Geral (não são preservados pelas funções)
\$k0..\$k1	\$26..\$27	Reservado pelo <i>kernel</i> do S.O.
\$gp	\$28	Ponteiro para área global ( <i>Global Pointer</i> )
\$sp	\$29	<i>Stack Pointer</i>
\$fp	\$30	<i>Frame Pointer</i>
\$ra	\$31	Endereço de retorno das funções ( <i>Return Address</i> )

Tabela II: Registos da FPU do MIPS e convenção de uso	
Nome Lógico	Uso Convencionado
\$f0(\$f1) ... \$f2(\$f3)	Cálculo de expressões e valor de retorno das funções
\$f4(\$f5) ... \$f10(\$f11)	Geral (não são preservados pelas funções)
\$f12(\$f13) ... \$f14(\$f15)	Passagem de parâmetros para funções.
\$f16(\$f17) ... \$f18(\$f19)	Geral (não são preservados pelas funções)
\$f20(\$f21) ... \$f30(\$f31)	Geral (não podem ser alterados pelas funções)

Rev 2021 - MBC, JLA, AO, LAU, ACP

Tabela III: Notação			
<b>Imm</b>	Valor imediato (constante) de 16 bits	<b>addr</b>	Endereço na forma <b>Imm(Rsrc) = (Rsrc) + Imm</b>
<b>IMM</b>	Valor imediato de 32 bits	<b>B<sub>k</sub>(Rsrc)</b>	<b>Byte</b> índice <b>k</b> de <b>Rsrc</b>
<b>Rsrc (1, 2)</b>	Registo fonte (1 ou 2)	<b>FPdst</b>	Registo destino do coprocessador aritmético
<b>(Rsrc)</b>	Conteúdo de <b>Rsrc</b>	<b>FPsrc (1, 2)</b>	Registo fonte do coprocessador aritmético (1 ou 2)
<b>Rdst</b>	Registo destino	<b>C<sub>z</sub></b>	Coprocessador nº <b>z</b>
<b>CReg</b>	Registo do Coprocessador <b>C<sub>z</sub></b>	<b>Src</b>	<b>Rsrc</b> ou <b>IMM</b>
<b>sym</b>	Endereço do símbolo (label) sym	<b>Imm5</b>	Valor imediato (constante) de 5 bits

Tabela IV: <i>System Calls</i> do MARS			
Protótipo equivalent em C	\$v0	Parâmetros de entrada	Retorno
<b>void</b> print_int10( <b>int</b> value)	1	\$a0 = value (int)	
<b>void</b> print_float( <b>float</b> value)	2	\$f12 = value (float)	
<b>void</b> print_double( <b>double</b> value)	3	\$f12 = value (double)	
<b>void</b> print_string( <b>char</b> *str)	4	\$a0 = str	
<b>int</b> read_int( <b>void</b> )	5		\$v0
<b>float</b> read_float( <b>void</b> )	6		\$f0
<b>double</b> read_double( <b>void</b> )	7		\$f0
<b>void</b> read_string( <b>char</b> *buf, <b>int</b> length)	8	\$a0 = buf, \$a1 = length	
<b>void</b> *sbrk( <b>int</b> amount)	9	\$a0 = amount	\$v0
<b>void</b> exit( <b>void</b> )	10		
<b>void</b> _print_char( <b>char</b> value)	11	\$a0 = value (char)	
<b>char</b> read_char( <b>void</b> )	12		\$v0
<b>void</b> print_int16( <b>unsigned int</b> value)	34	\$a0 = value (unsigned int)	
<b>void</b> print_int2( <b>unsigned int</b> value)	35	\$a0 = value (unsigned int)	
<b>void</b> print_intu10( <b>unsigned int</b> value)	36	\$a0 = value (unsigned int)	

Tabela V - Directivas do Assembler	
Directivas	Descrição
<b>Para controlo dos Segmentos</b>	
.data [address]	Coloca os próximos itens no segmento de dados do utilizador (opcionalmente a partir de <i>address</i> ).
.text [address]	Coloca os próximos itens no segmento de código do utilizador (opcionalmente a partir de <i>address</i> ).
.kdata [address]	Coloca os próximos itens no segmento de dados do <i>kernel</i> (opcionalmente a partir de <i>address</i> ).
.ktext [address]	Coloca os próximos itens no segmento de código do <i>kernel</i> (opcionalmente a partir de <i>address</i> ).
<b>Para criação de constantes e variáveis em memória:</b>	
.ascii str	Armazena uma <i>string</i> em memória sem lhe acrescentar o terminador '\0'.
.eqv label, valor	Substitui todas as ocorrências de <i>label</i> no programa por <i>valor</i> .
.asciiz str	Armazena uma <i>string</i> em memória acrescentando-lhe o terminador '\0'.
.byte b <sub>1</sub> , ..., b <sub>n</sub>	Armazena as grandezas de 8 bits b <sub>1</sub> , ..., b <sub>n</sub> em sucessivos bytes de memória.
.half h <sub>1</sub> , ..., h <sub>n</sub>	Armazena as grandezas de 16 bits h <sub>1</sub> , ..., h <sub>n</sub> em sucessivas meias palavras de memória.
.word w <sub>1</sub> , ..., w <sub>n</sub>	Armazena as grandezas de 32 bits w <sub>1</sub> , ..., w <sub>n</sub> em sucessivas palavras de memória.
.float f <sub>1</sub> , ..., f <sub>n</sub>	Armazena f <sub>1</sub> , ..., f <sub>n</sub> em vírgula flutuante, precisão simples (32 bits) no seg. de dados.
.double d <sub>1</sub> , ..., d <sub>n</sub>	Armazena d <sub>1</sub> , ..., d <sub>n</sub> em vírgula flutuante, precisão dupla (64 bits) no seg. de dados.
.space n	Reserva <i>n</i> bytes no segmento de dados, sem inicializar
<b>Para controlo do alinhamento:</b>	
.align n	Alinha o próximo item num endereço múltiplo de 2 <sup>n</sup> .
<b>Para referências externas:</b>	
.globl sym	Declara que o símbolo <i>sym</i> é global e pode ser referenciado em outros ficheiros.
.extern sym size	Declara que o item associado a <i>sym</i> ocupa <i>size</i> bytes e é um símbolo global.