

Instruções Nativas		Instruções Virtuais	
<b>Transferência Memória-Registro (<i>Load</i>)</b>	<b>Cálculo c/ Inteiros: Operações Aritméticas</b>	<b>Transferência Memória-Registro (<i>Load</i>)</b>	<b>Salto Relativo (<i>Branch</i>)</b>
lb Rdst, addr	add Rdst, Rsrc1, Rsrc2	l.d FPdst, addr 1	b Label
lbu Rdst, addr	addi Rdst, Rsrc, Imm	l.s FPdst, addr 1	bge Rsrc, Src, Label 2
lw Rdst, addr	addiu Rdst, Rsrc, Imm		bgeu Rsrc, Src, Label
lwcx CReg, addr	addu Rdst, Rsrc1, Rsrc2	<b>Transferência Registro-Memória (<i>Store</i>)</b>	bgt Rsrc, Src, Label 3
ldcx CReg, addr	div Rsrc1, Rsrc2	s.d FPsrc, addr 1	bgtu Rsrc, Src, Label
	divu Rsrc1, Rsrc2	s.s FPsrc, addr 1	ble Rsrc, Src, Label 2
<b>Transferência Registro-Memória (<i>Store</i>)</b>	mult Rsrc1, Rsrc2		bleu Rsrc, Src, Label
sb Rsrc, addr	multu Rsrc1, Rsrc2	<b>Transferência Registro-Registro (<i>Move</i>)</b>	blt Rsrc, Src, Label 3
sw Rsrc, addr	sub Rdst, Rsrc1, Rsrc2	move Rdst, Rsrc 1	bltu Rsrc, Src, Label
swcxc Reg, addr	subu Rdst, Rsrc1, Rsrc2		beq Rsrc, Src, Label 2
sdcxc Reg, addr		<b>Manipulação de Const. (<i>Load Imm/sym</i>)</b>	bne Rsrc, Src, Label
<b>Transferência Registro-Registro (<i>Move</i>)</b>	<b>Cálculo c/ Inteiros: Op. Lógicas Bitwise</b>	la Rdst, sym 2	
mfhi Rdst	and Rdst, Rsrc1, Rsrc2	li Rdst, IMM → 2 x for valor 32 bits 1 x for valor 16 bits	
mflo Rdst	andi Rdst, Rsrc, Imm		
mthi Rsrc	nor Rdst, Rsrc1, Rsrc2		
mtlo Rsrc	or Rdst, Rsrc1, Rsrc2		
mfccx Rdst, Creg	ori Rdst, Rsrc, Imm		
mtccx Rsrc, Creg	xor Rdst, Rsrc1, Rsrc2		
mov.d FPdst, FPsrc	xori Rdst, Rsrc, Imm	<b>Cálculo c/ Inteiros: Op. Aritméticas</b>	
mov.s FPdst, FPsrc		mul Rdst, Rsrc, Src → 1 para 32 2 para 64	
	<b>Cálculo c/ Inteiros: Operações de Shift</b>	mulo Rdst, Rsrc, Src	
<b>Manipulação de Const. (<i>Load Immediate</i>)</b>	sll Rdst, Rsrc1, Imm5	mulou Rdst, Rsrc, Src	
lui Rdst, Imm	sllv Rdst, Rsrc1, Rsrc2		
	sra Rdst, Rsrc1, Imm5	div Rdst, Rsrc, Src 1	
<b>Instruções de Comparação</b>	srav Rdst, Rsrc1, Rsrc2	divu Rdst, Rsrc, Src 1	
slt Rdst, Rsrc1, Rsrc2	srl Rdst, Rsrc1, Imm5	rem Rdst, Rsrc, Src 3	
sltu Rdst, Rsrc1, Rsrc2	srlv Rdst, Rsrc1, Rsrc2	remu Rdst, Rsrc, Src	
slti Rdst, Rsrc, Imm		<b>Cálculo c/ Inteiros: Operações de Rotate</b>	
sltiu Rdst, Rsrc, Imm	<b>Cálculo em Vírgula Flutuante</b>	rol Rdst, Rsrc, Src	
<b>Salto Relativo (<i>Branch</i>) e Absoluto (<i>Jump</i>)</b>	add.p FPdst, FPsrc1, FPsrc2	rор Rdst, Rsrc, Src	
bczf Label	sub.p FPdst, FPsrc1, FPsrc2		
bczt Label	div.p FPdst, FPsrc1, FPsrc2		
beq Rsrc1, Rsrc2, Label	mul.p FPdst, FPsrc1, FPsrc2		
bne Rsrc1, Rsrc2, Label			
bgez Rsrc, Label	neg.p FPdst, FPsrc		
bgtz Rsrc, Label	abs.p FPdst, FPsrc		
blez Rsrc, Label	cvt.d.s FPdst, FPsrc		
bltz Rsrc, Label	cvt.d.w FPdst, FPsrc		
j Label	cvt.s.d FPdst, FPsrc		
jal Label	cvt.s.w FPdst, FPsrc		
jalr Rsrc	cvt.w.d FPdst, FPsrc		
jr Rsrc	cvt.w.s FPdst, FPsrc		
<b>Manipulação de Exceções e Traps</b>	<b>Comparação em Vírgula Flutuante</b>		
break n	c.eq.p FPsrc1, FPsrc2		
nop	c.le.p FPsrc1, FPsrc2		
eret	c.lt.p FPsrc1, FPsrc2		
syscall			

Tabela III: Notação			
Imm	Valor imediato (constante) de 16 bits	addr	Endereço na forma Imm(Rsrc) = (Rsrc) + Imm
IMM	Valor imediato de 32 bits	B <sub>k</sub> (Rsrc)	Byte índice k de Rsrc
Rsrc (1, 2)	Registo fonte (1 ou 2)	FPdst	Registo destino do coprocessador aritmético
(Rsrc)	Conteúdo de Rsrc	FPsrc (1, 2)	Registo fonte do coprocessador aritmético (1 ou 2)
Rdst	Registo destino	Src	Rsrc ou IMM
CReg	Registo do Coprocessador Cz	cz	Coprocessador nº z (0 or 1)
sym	Endereço do símbolo (label) sym	Imm5	Valor imediato (constante) de 5 bits
Label	Endereço de uma instrução	.p	Precisão: substituir por .s ou .d

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

Tabela V - Directivas do Assembler	
Directivas	Descrição
Para controlo dos Segmentos	
.data [address]	Coloca os próximos itens no segmento de dados do utilizador (opcionalmente a partir de address).
.text [address]	Coloca os próximos itens no segmento de código do utilizador (opcionalmente a partir de address).
.kdata [address]	Coloca os próximos itens no segmento de dados do kernel (opcionalmente a partir de address).
.ktext [address]	Coloca os próximos itens no segmento de código do kernel (opcionalmente a partir de address).
Para criação de constantes e variáveis em memória:	
.ascii str	Armazena uma string em memória sem lhe acrescentar o terminador '\0'.
.asciiz str	Armazena uma string em memória acrescentando-lhe o terminador '\0'.
.space n	Reserva n bytes no segmento de dados, sem inicializar
.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.
.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.
.eqv label, valor	Substitui todas as ocorrências de label no programa por valor.
Para controlo do alinhamento:	
.align n	Alinha o próximo item num endereço múltiplo de 2 <sup>n</sup> .
Para referências externas:	
.globl sym	Declara que o símbolo sym é global e pode ser referenciado em outros ficheiros.
.extern sym size	Declara que o item associado a sym ocupa size bytes e é um símbolo global.
.include filename	Insere o conteúdo do ficheiro especificado (o nome do ficheiro é colocado entre aspas).