

Instruções Nativas		Instruções Virtuais		DETI-UA - ACI
Transferência Memória-Registro ( <i>Load</i> )		Transferência Memória-Registro ( <i>Load</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	bge    Rsrc, Src, Label	
lw      Rdst, addr	addiu   Rdst, Rsrc, Imm		bgeu   Rsrc, Src, Label	
lwc1    CReg, addr	addu    Rdst, Rsrc1, Rsrc2	Transferência Registro-Memória ( <i>Store</i> )	bgt    Rsrc, Src, Label	
ldc1    CReg, addr	div     Rsrc1, Rsrc2	s.d    FPsrc, addr	bgtu   Rsrc, Src, Label	
Transferência Registro-Memória ( <i>Store</i> )	divu    Rsrc1, Rsrc2	s.s    FPsrc, addr	ble    Rsrc, Src, Label	
sb      Rsrc, addr	mult    Rsrc1, Rsrc2		bleu   Rsrc, Src, Label	
sw      Rsrc, addr	multu   Rsrc1, Rsrc2	Transferência Registro-Registro ( <i>Move</i> )	blt    Rsrc, Src, Label	
swc1    Creg, addr	sub     Rdst, Rsrc1, Rsrc2	move   Rdst, Rsrc	bltu   Rsrc, Src, Label	
sdc1    Creg, addr	subu    Rdst, Rsrc1, Rsrc2		beq    Rsrc, Src, Label	
Transferência Registro-Registro ( <i>Move</i> )	Cálculo c/ Inteiros: Op. Lógicas Bitwise	Manipulação de Const. ( <i>Load Imm/sym</i> )	bne    Rsrc, Src, Label	
mfhi    Rdst	and    Rdst, Rsrc1, Rsrc2	la      Rdst, sym		
mflo    Rdst	andi   Rdst, Rsrc, Imm	li      Rdst, IMM		
mthi    Rsrc	nor    Rdst, Rsrc1, Rsrc2			
mtlo    Rsrc	or      Rdst, Rsrc1, Rsrc2			
mfcl    Rdst, Creg	ori    Rdst, Rsrc, Imm			
mtcl    Rsrc, Creg	xor    Rdst, Rsrc1, Rsrc2			
mov.d   FPdst, FPsrc	xori   Rdst, Rsrc, Imm			
mov.s   FPdst, FPsrc				
	Cálculo c/ Inteiros: Operações de Shift			
	sll    Rdst, Rsrc1, Imm5			
	sllv   Rdst, Rsrc1, Rsrc2			
	sra    Rdst, Rsrc1, Imm5			
	srav   Rdst, Rsrc1, Rsrc2			
	srl    Rdst, Rsrc1, Imm5			
	srlv   Rdst, Rsrc1, Rsrc2			
	Cálculo em Vírgula Flutuante			
	add.p   FPdst, FPsrc1, FPsrc2			
	sub.p   FPdst, FPsrc1, FPsrc2			
	div.p   FPdst, FPsrc1, FPsrc2			
	mul.p   FPdst, FPsrc1, FPsrc2			

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 c1	c1	Coprocessador nº 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).