

Tabela IV: Notação			
Imm	Valor imediato (constante) de 16 bits	addr	Endereço na forma Imm(Rsrc) = (Rsrc) + Imm
IMM	Valor imediato de 32 bits	B_k (Rsrc)	Byte índice k de Rsrc
Rsrc (1, 2)	Registo fonte (1 ou 2)	FPdst	Registo destino do coprocessador aritmético
(Rsrc)	Conteudo de Rsrc	FPsrc (1, 2)	Registo fonte do coprocessador aritmético (1 ou 2)
Rdst	Registo destino	C_z	Coprocessador nº z
CReg	Registo do Coprocessador C_z	ADDR	Um dos modos de endereçamento da Tabela I
		Src	Rsrc ou IMM
(¹) Estas instruções podem usar operandos na forma “ Rrdt, IMM ” sendo Rrdt simultaneamente o registo fonte e destino			

Tabela V - Directivas do Assembler	
Directivas	Descrição
Para controlo dos Segmentos	
.data [address]	Coloca os próximos items no segmento de dados do ut ilizador (opcionalmente a partir de <i>address</i>).
.text [address]	Coloca os próximos items no segmento de código do u tilizador (opcionalmente a partir de <i>address</i>).
.kdata [address]	Coloca os próximos items no segmento de dados do <i>kernel</i> (opcionalmente a partir de <i>address</i>).
.ktext [address]	Coloca os próximos items no segmento de código do <i>kernel</i> (opcionalmente a partir de <i>address</i>).
Para criação de constantes e variáveis em memória:	
.ascii str	Armazena uma <i>string</i> em memória sem lhe acrescentar o terminador '\0'.
.asciiz str	Armazena uma <i>string</i> em memória acrescentando-lhe o terminador '\0'.
.byte b ₁ , ..., b _n	Armazena as grandezas de 8 bits b ₁ , ..., b _n em sucessivos bytes de memória.
.half h ₁ , ..., h _n	Armazena as grandezas de 16 bits h ₁ , ..., h _n em sucessivas meias palavras de memória.
.word w ₁ , ..., w _n	Armazena as grandezas de 32 bits w ₁ , ..., w _n em sucessivas palavras de memória.
.float f ₁ , ..., f _n	Armazena f ₁ , ..., f _n em vírgula flutuante, precisão simples (32 bits) n o seg. de dados.
.double d ₁ , ..., d _n	Armazena d ₁ , ..., d _n em vírgula flutuante, precisão dupla (64 bits) no seg. de dados.
.space n	Aloca <i>n</i> bytes (SPIM só deixa usar esta directiva no segmen to .data).
Para controlo do alinhamento:	
.align n	Alinha o próximo item num endereço múltiplo de 2 ⁿ .
Para referências externas:	
.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 size bytes e é um símbolo global.

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