

Instruções da Máquina Nativa			Instruções da Máquina Virtual						
Transferência Memória-Registo (<i>Load</i>)		Cálculo c/ Inteiros: Operações Aritméticas	Transferência Memória-Registo (<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	Transferência Registo-Memória (<i>Store</i>)		bge	Rsrc, Src, Label			
lwcx	CReg, addr	addu Rdst, Rsrc1, Rsrc2			s.d	FPsrc, addr			
Transferência Registo-Memória (<i>Store</i>)		div Rsrc1, Rsrc2			s.s	FPsrc, addr			
		divu Rsrc1, Rsrc2	Transferência Registo-Registo (<i>Move</i>)						
		mult Rsrc1, Rsrc2							
multu Rsrc1, Rsrc2	move	Rdst, Rsrc							
sw	Rsrc, addr	sub Rdst, Rsrc1, Rsrc2	Manipulação de Const. (<i>Load Imm/sym</i>)		blt	Rsrc, Src, Label			
swcx	Creg, addr	subu Rdst, Rsrc1, Rsrc2			bltu	Rsrc, Src, Label			
Transferência Registo-Registo (<i>Move</i>)		Cálculo c/ Inteiros: Op. Lógicas Bitwise			bnez	Rsrc, Label			
mfhi	Rdst	and Rdst, Rsrc1, Rsrc2	la	Rdst, sym	Arquitetura de Computadores II DETI-UA				
mflo	Rdst	andi Rdst, Rsrc, Imm	li	Rdst, IMM					
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	Tabela resumida das instruções da arquitetura MIPS			
Manipulação de Const. (<i>Load Immediate</i>)		sll	Rdst, Rsrc1, Imm5	mul	Rdst, Rsrc, Src				
		sllv	Rdst, Rsrc1, Rsrc2	mulo	Rdst, Rsrc, Src				
		sra	Rdst, Rsrc1, Imm5	mulou	Rdst, Rsrc, Src				
lui	Rdst, Imm	srav	Rdst, Rsrc1, Rsrc2	mulo	Rdst, Rsrc, Src		Tabela I: Registos do MIPS e convenção de uso		
Instruções de Comparação		srl	Rdst, Rsrc1, Imm5	mulou	Rdst, Rsrc, Src				
		srlv	Rdst, Rsrc1, Rsrc2	neg	Rdst, Rsrc				
		Cálculo em Vírgula Flutuante		negu	Rdst, Rsrc				
slt	Rdst, Rsrc1, Rsrc2	abs.p	FPdst, FPsrc	rem	Rdst, Rsrc, Src	Nome Lóg. Nome Real Uso Convencionado			
sltu	Rdst, Rsrc1, Rsrc2	add.p	FPdst, FPsrc1, FPsrc2	remu	Rdst, Rsrc, Src				
slti	Rdst, Rsrc, Imm	c.eq.p	FPsrc1, FPsrc2	Cálculo c/ Inteiros: Op. Lógicas Bitwise					
sltiu	Rdst, Rsrc, Imm	c.le.p	FPsrc1, FPsrc2	not	Rdst, Rsrc				
Salto Relativo (<i>Branch</i>) e Absoluto (<i>Jump</i>)		c.lt.p	FPsrc1, FPsrc2	Cálculo c/ Inteiros: Operações de Rotate					
		cvt.d.s	FPdst, FPsrc	rol	Rdst, Rsrc, Src				
		cvt.d.w	FPdst, FPsrc	ror	Rdst, Rsrc, Src				
		cvt.s.d	FPdst, FPsrc	Instruções de Comparação					
		cvt.s.w	FPdst, FPsrc					seq	Rdst, Rsrc, Src
		cvt.w.d	FPdst, FPsrc					sge	Rdst, Rsrc, Src
		cvt.w.s	FPdst, FPsrc	sgeu	Rdst, Rsrc, Src				
		div.p	FPdst, FPsrc1, FPsrc2	sgt	Rdst, Rsrc, Src				
		mul.p	FPdst, FPsrc1, FPsrc2	sgtu	Rdst, Rsrc, Src				
		neg.p	FPdst, FPsrc	sle	Rdst, Rsrc, Src				
		sub.p	FPdst, FPsrc1, FPsrc2	sleu	Rdst, Rsrc, Src				
		Manipulação de Exceções e Traps		sne	Rdst, Rsrc, Src				
		break	n	JLA, AO, TOS, Jan / 2017					
		Nop							
Eret									
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_k(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	C_z	Coprocessador nº z
CReg	Registo do Coprocessador C_z	Src	Rsrc ou IMM
sym	Endereço do símbolo (label) sym	Imm5	Valor imediato (constante) de 5 bits

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 <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>).
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.
.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) no 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	Reserva <i>n</i> bytes no segmento de dados, sem inicializar
.equ name, val	Atribui ao símbolo "name" o valor "val"
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 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>	Inclui o ficheiro especificado no campo "filename"

Tabela IV: System Calls da placa DETPIC32			
Protótipo equivalent em C	\$v0	Parâmetros de entrada	Retorno
char inkey(void)	1		\$v0
char getChar(void)	2		\$v0
void putChar(char ch)	3	\$a0 = character	
unsigned int readInt(unsigned int base)	4	\$a0 = base	\$v0
int readInt10(void)	5		\$v0
void printInt(unsigned int val, unsigned int base)	6	\$a0 = val, \$a1 = base	
void printInt10(int val)	7	\$a0	
void printStr(char *str)	8	\$a0 = str	
void readStr(char *buffer, unsigned int nc)	9	\$a0 = buffer, \$a1 = nc	
void exit(int code)	10	\$a0 = exit code	
unsigned int readCoreTimer(void)	11		\$v0
void resetCoreTimer(void)	12		

printInt(), "base": 16 lsbits – [2.. 16], 16 msbits – número de caracteres com que o resultado é apresentado (o valor por omissão é 0, i.e. sem formatação)

Tabela VI: Registos do CP0 do MIPS		
Nome Lógico	Nome Real	Conteúdo
\$BadVAddr	\$8	Endereço de memória inválido que causou a excepção
\$Status	\$12	Interrupt mask & Enable bits
\$Cause	\$13	Tipo de excepção e interrupt bits
\$EPC	\$14	Endereço da instrução que causou a excepção
Tabela VII: Valores dos bits [5..2] do registo Cause		
Valor	Nome	Significado
0	INT	External Interrupt
4	ADDRL	Add error exception (load or store)
5	ADDRS	Add error exception (fetch)
6	IBUS	Bus error on instruction fetch
7	DBUS	Bus error on data load or store
8	SYSCALL	Syscall exception
9	BKPT	Break point exception
10	RI	Reserved instruction exception
12	OVF	Overflow exception