Instruções da Máquina Nativa			Instruções da Máquina Virtual						
Transferé	encia Memória-Registo (<i>Load</i>)		Inteiros: Operações Aritméticas	Transfe	rência Memória-Registo (<i>Load</i>)		elativo (Branch)		
lb	Rdst,addr	add	Rdst,Rsrc1,Rsrc2	1.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			bge	Rsrc,Src,Lab	oel	
lwcz	CReg,addr	addu	Rdst,Rsrc1,Rsrc2	Transfe	rência Registo-Memória (Store)	bgeu	Rsrc,Src,Lal	oel	
		div	Rsrc1,Rsrc2	s.d	FPsrc,addr	bgt	Rsrc,Src,Lal	oel	
Transferé	encia Registo-Memória (Store)	divu	Rsrc1,Rsrc2	s.s	FPsrc,addr	bgtu	Rsrc,Src,Lal	oel	
sb	Rsrc,addr	mult	Rsrc1,Rsrc2			ble	Rsrc,Src,Lal	oel	
sw	Rsrc,addr	multu	Rsrc1,Rsrc2	Transfe	rência Registo-Registo (Move)	bleu	Rsrc,Src,Lal	oel	
$\mathrm{SWC}Z$	Creg,addr	sub	Rdst,Rsrc1,Rsrc2	move	Rdst,Rsrc	blt	Rsrc,Src,Lal	oel	
		subu	Rdst,Rsrc1,Rsrc2			bltu	Rsrc,Src,La	oel	
Transferé	ència Registo-Registo (Move)	Cálculo c/	Inteiros: Op. Lógicas Bitwise	Manipu	lação de Const. (Load Imm/sym)	bnez	Rsrc,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	1.d	FPdst,sym		Arquitet	ura de Comp	utadores II
mtlo	Rsrc	or	Rdst,Rsrc1,Rsrc2	1.s	FPdst,sym		,	DETI-UA	
mfcz	Rdst,Creg	ori	Rdst,Rsrc,Imm						
mtcz	Rsrc,Creg	xor	Rdst,Rsrc1,Rsrc2	Cálculo	c/ Inteiros: Op. Aritméticas	Tabela resumida das instruções da arquitetura MIPS		da arquitetura MIPS	
mov.d	FPdst,FPsrc	xori	Rdst,Rsrc,Imm	abs	Rdst,Rsrc				
mov.s	FPdst,FPsrc	Cálculo c/	Inteiros: Operações de Shift	div	Rdst,Rsrc,Src				
		sll	Rdst,Rsrc1,Imm5	divu	Rdst,Rsrc,Src				
Manipula	ıção de Const. (Load Immediate)	sllv	Rdst,Rsrc1,Rsrc2	mul	Rdst,Rsrc,Src				
lui	Rdst,Imm	sra	Rdst,Rsrc1,Imm5	mulu	Rdst,Rsrc,Src		Tabela I: Reg	istos do MIPS e	convenção de uso
		srav	Rdst,Rsrc1,Rsrc2	mulo	Rdst,Rsrc,Src	Nome I	.óg. Nome Real	Uso Convencion	ado
Instruçõe	s de Comparação	srl	Rdst,Rsrc1,Imm5	mulou	Rdst,Rsrc,Src	\$zero	\$0	Constante 0	
slt	Rdst,Rsrc1,Rsrc2	srlv	Rdst,Rsrc1,Rsrc2	neg	Rdst,Rsrc	\$at	\$1	Reservado pelo ass	emblador
sltu	Rdst,Rsrc1,Rsrc2	Cálculo e	m Vírgula Flutuante	negu	Rdst,Rsrc	\$v0\$v	1 \$2\$3	Cálculo de expresso	ões e valor de retorno das funções.
slti	Rdst,Rsrc,Imm	abs.p	FPdst,FPsrc	rem	Rdst,Rsrc,Src	\$a0\$a3	3 \$4\$7	Primeiros 4 parâme	tros das funções
sltiu	Rdst,Rsrc,Imm	add.p	FPdst,FPsrc1,FPsrc2	remu	Rdst,Rsrc,Src	\$t0\$t7	\$8\$15	Geral (não são pres	ervados pelas funções)
		c.eq.p	FPsrc1,FPsrc2	Cálculo	c/ Inteiros: Op. Lógicas Bitwise	\$s0\$s7	\$16\$23	Geral (não podem s	er alterados pelas funções)
Salto Rela	ativo (Branch) e Absoluto (Jump)	c.le.p	FPsrc1,FPsrc2	not	Rdst,Rsrc	\$t8\$t9	\$24\$25	Geral (não são pres	ervados pelas funções)
bczf	Label	c.lt.p	FPsrc1,FPsrc2			\$k0\$k	1 \$26\$27	Reservado pelo ker	nel do S.O.
bczt	Label	cvt.d.s	FPdst,FPsrc	Cálculo	c/ Inteiros: Operações de <i>Rotate</i>	\$gp	\$28	Ponteiro para área g	global (Global Pointer)
beq	Rsrc1,Rsrc2,Label	cvt.d.w	FPdst,FPsrc	rol	Rdst,Rsrc,Src	\$sp	\$29	Stack Pointer	
bgez	Rsrc,Label	cvt.s.d	FPdst,FPsrc	ror	Rdst,Rsrc,Src	\$fp	\$30	Frame Pointer	
bgezal	Rsrc,Label	cvt.s.w	FPdst,FPsrc			\$ra	\$31	Endereço de retorn	o das funções (Return Address)
bgtz	Rsrc,Label	cvt.w.d	FPdst,FPsrc	Instruçõ	ões de Comparação		•	•	
blez	Rsrc,Label	cvt.w.s	FPdst,FPsrc	seq	Rdst,Rsrc,Src				
bltz	Rsrc,Label	div.p	FPdst,FPsrc1,FPsrc2	sge	Rdst,Rsrc,Src				
bltzal	Rsrc,Label	mul.p	FPdst,FPsrc1,FPsrc2	sgeu	Rdst,Rsrc,Src				
bne	Rsrc1,Rsrc2,Label	neg.p	FPdst,FPsrc	sgt	Rdst,Rsrc,Src				
j	Label	sub.p	FPdst,FPsrc1,FPsrc2	sgtu	Rdst,Rsrc,Src				
jal	Label	Manipula	ção de Excepções e Traps	sle	Rdst,Rsrc,Src				
jalr	Rsrc	break	n	sleu	Rdst,Rsrc,Src				
jr	Rsrc	Nop		sne	Rdst,Rsrc,Src				
J =				•					
] _		Eret							ILA, AO, TOS, Jan / 2017

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	Cz	Coprocessador nº z
CReg	Registo do Coprocessador Cz	Src	Rsrc ou IMM
sym	Endereço do símbolo (label) sym	Imm5	Valor imediato (constante) de 5 bits

Directivas	Descrição			
Para controlo dos Segmento	,			
.data [address]	Coloca os próximos items no segmento de dados do utilizador (opcionalmente a partir de address)			
.text [address]	Coloca os próximos items no segmento de código do utilizador (opcionalmente a partir de <i>address</i>).			
.kdata [address]	Coloca os próximos items no segmento de dados do kernel (opcionalmente a partir de address).			
.ktext [address]	Coloca os próximos items no segmento de código do kernel (opcionalmente a partir de address).			
Para criação de constantes o	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'.			
.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, \ldots, 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 n bytes no segmento de dados, sem inicializar			
.equ name, val	Atribui ao símbolo "name" o valor "val"			
Para controlo do alinhamen	to:			
.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></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
<pre>void printInt(unsigned int val, unsigned int base)</pre>	6	a0 = val, a1 = base	
void printInt10(int val)	7	\$a0	
<pre>void printStr(char *str)</pre>	8	a0 = str	
<pre>void readStr(char *buffer, unsigned int nc)</pre>	9	a0 = buffer, a1 = nc	
void exit(int code)	10	\$a0 = exit code	
<pre>unsigned int readCoreTimer(void)</pre>	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 [52] 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		