

Arquitectura de Computadores II – Tabela resumida das instruções da arquitectura MIPS

Instruções da Máquina Nativa			Instruções da Máquina Virtual					
Transferência Memória-Registro (<i>Load</i>)		Cálculo c/ Inteiros: Operações Aritméticas	Transferência Memória-Registro (<i>Load</i>)		Salto Relativo (<i>Branch</i>)			
lb	Rdst,addr	add	Rdst,Rsrc1,Rsrc2	la	Rdst,ADDR	b	Label	
lbu	Rdst,addr	addi	Rdst,Rsrc,Imm	ld	Rdst,ADDR	beqz	Rsrc,Label	
lh	Rdst,addr	addiu	Rdst,Rsrc,Imm	ulh	Rdst,ADDR	bge	Rsrc,Src,Label	
lhu	Rdst,addr	addu	Rdst,Rsrc1,Rsrc2	ulhu	Rdst,ADDR	bgeu	Rsrc,Src,Label	
lw	Rdst,addr	div	Rsrc1,Rsrc2	ulw	Rdst,ADDR	bgt	Rsrc,Src,Label	
lwl	Rdst,addr	divu	Rsrc1,Rsrc2	Transferência Registro-Memória (<i>Store</i>)		bgtu	Rsrc,Src,Label	
lwr	Rdst,addr	mult	Rsrc1,Rsrc2	sd	Rdst,ADDR	ble	Rsrc,Src,Label	
Transferência Registro-Memória (<i>Store</i>)		multu	Rsrc1,Rsrc2	ush	Rdst,ADDR	bleu	Rsrc,Src,Label	
sb	Rsrc,addr	sub	Rdst,Rsrc1,Rsrc2	usw	Rdst,ADDR	blt	Rsrc,Src,Label	
sh	Rsrc,addr	subu	Rdst,Rsrc1,Rsrc2	Manipulação de Const. (<i>Load Immediate</i>)		bltu	Rsrc,Src,Label	
sw	Rsrc,addr	Cálculo c/ Inteiros: Op. Lógicas Bitwise		li	Rdst,IMM	bnez	Rsrc,Label	
swl	Rsrc,addr	and	Rdst,Rsrc1,Rsrc2	Cálculo c/ Inteiros: Op. Aritméticas				
swr	Rsrc,addr	andi	Rdst,Rsrc,Imm	abs	Rdst,Rsrc	Tabela I: Modos de Endereçamento		
Transferência Registro-Registro (<i>Move</i>)		nor	Rdst,Rsrc1,Rsrc2	div	Rdst,Rsrc,Src	Modo de Endereçamento	Cálculo do Endereço	
mfhi	Rdst	or	Rdst,Rsrc1,Rsrc2	divu	Rdst,Rsrc,Src	(reg)	Conteúdo do registro reg	
mflo	Rdst	ori	Rdst,Rsrc,Imm	mul	Rdst,Rsrc,Src	Imm	Constante imm	
mthi	Rsrc	xor	Rdst,Rsrc1,Rsrc2	mulo	Rdst,Rsrc,Src	imm(reg)	Conteúdo do registro reg + constante imm	
mtlo	Rsrc	xori	Rdst,Rsrc,Imm	mulou	Rdst,Rsrc,Src	sym	Endereço do símbolo (<i>label</i>) sym	
mfcz	Rdst,Creg	Cálculo c/ Inteiros: Operações de Shift		neg	Rdst,Rsrc	sym +/- imm	Endereço do símbolo sym +/- constante imm	
mtcz	Rsrc,Creg	sll	Rdst,Rsrc1,Rsrc2	negu	Rdst,Rsrc	sym +/- imm (reg)	Conteúdo de reg + endereço de sym +/- const. imm	
Manipulação de Const. (<i>Load Immediate</i>)		sllv	Rdst,Rsrc1,Rsrc2	rem	Rdst,Rsrc,Src			
lui	Rdst,Imm	sra	Rdst,Rsrc1,Rsrc2	remu	Rdst,Rsrc,Src	Tabela II: Registos do MIPS e convenção de uso		
		srav	Rdst,Rsrc1,Rsrc2	Cálculo c/ Inteiros: Op. Lógicas Bitwise		Nome Lóg.	Nome Real	Uso Convencionado
Instruções de Comparação		srl	Rdst,Rsrc1,Rsrc2	not	Rdst,Rsrc	\$zero	\$0	Constante 0
slt	Rdst,Rsrc1,Rsrc2	srlv	Rdst,Rsrc1,Rsrc2	Cálculo c/ Inteiros: Operações de Rotate		\$at	\$1	Reservado pelo assembler
sltu	Rdst,Rsrc1,Rsrc2	Manipulação de Exceções e Traps		rol	Rdst,Rsrc,Src	\$v0..\$v1	\$2..\$3	Cálculo de expressões e valor de retorno das funções.
slti	Rdst,Rsrc,Imm	break	n	ror	Rdst,Rsrc,Src	\$a0..\$a3	\$4..\$7	Primeiros 4 parâmetros das funções
sltiu	Rdst,Rsrc,Imm	nop		Instruções de Comparação		\$t0..\$t7	\$8..\$15	Geral (não são preservados pelas funções)
Salto Relativo (<i>Branch</i>) e Absoluto (<i>Jump</i>)		eret		seq	Rdst,Rsrc,Src	\$s0..\$s7	\$16..\$23	Geral (não podem ser alterados pelas funções)
beq	Rsrc1,Rsrc2,Label	syscall		sge	Rdst,Rsrc,Src	\$t8..\$t9	\$24..\$25	Geral (não são preservados pelas funções)
bgez	Rsrc,Label			sgeu	Rdst,Rsrc,Src	\$k0..\$k1	\$26..\$27	Reservado pelo <i>kernel</i> do S.O.
bgezal	Rsrc,Label			sgt	Rdst,Rsrc,Src	\$gp	\$28	Ponteiro para área global (<i>Global Pointer</i>)
bgtz	Rsrc,Label			sgtu	Rdst,Rsrc,Src	\$sp	\$29	<i>Stack Pointer</i>
blez	Rsrc,Label			sle	Rdst,Rsrc,Src	\$fp	\$30	<i>Frame Pointer</i>
bltz	Rsrc,Label			sleu	Rdst,Rsrc,Src	\$ra	\$31	Endereço de retorno das funções (<i>Return Address</i>)
bltzal	Rsrc,Label			sne	Rdst,Rsrc,Src			
bne	Rsrc1,Rsrc2,Label							
j	Label							
jal	Label							
jalr	Rsrc							
jr	Rsrc							

Arquitectura de Computadores II – Tabela resumida das instruções da arquitectura MIPS

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	C_z	Coprocessador nº z
Rdst	Registo destino	ADDR	Um dos modos de endereçamento da Tabela I
CReg	Registo do Coprocessador C_z	Src	Rsrc ou IMM

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.
.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) 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	Aloca <i>n</i> bytes
.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 <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 <i>size</i> bytes e é um símbolo global.
.include <filename>	Inclui o ficheiro especificado no campo "filename"

Tabela IV: System Calls do 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 exceção
\$Status	\$12	Interrupt mask & Enable bits
\$Cause	\$13	Tipo de exceção e interrupt bits
\$EPC	\$14	Endereço da instrução que causou a exceção
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