

197-200 & アセンブリ

MIPSのオペランド

種類	例	備考
32個のレジスタ	\$s0-\$s7, \$t0-\$t9, \$gp, \$fp, \$zero, \$sp, \$ra, \$at, Hi, Lo	高速にアクセス可能なデータの格納場所。MIPSでは、演算の対象となるデータはレジスタに収めなければならない。MIPSのレジスタ\$zeroの値は常に0である。レジスタ\$atはアセンブラの擬似命令および大きな定数用に予約されている。HiおよびLoは32ビットのレジスタであり、乗算および除算の結果を保持するために使用される。
2 ³⁰ のメモリ語	メモリ[0], メモリ[4], ... メモリ[4294967292]	MIPSではデータ転送命令によってのみアクセスされる。MIPSではバイト・アドレス方式をとっている。したがって、順に並んだ語のアドレスは4刻みでずれる。メモリには配列のようなデータ構造や手続き呼び出しなどでスピル・アウトされたレジスタの値を保持する。

MIPSのアセンブリ言語

区分	命令	例	意味	備考
算術演算	add	add \$s1,\$s2,\$s3	\$s1=\$s2+\$s3	3オペランド、オーバーフローを検出
	subtract	sub \$s1,\$s2,\$s3	\$s1=\$s2-\$s3	3オペランド、オーバーフローを検出
	add immediate	addi \$s1,\$s2,100	\$s1=\$s2+100	定数を加算、オーバーフローを検出
	add unsigned	addu \$s1,\$s2,\$s3	\$s1=\$s2+\$s3	3オペランド、オーバーフローを検出しない
	subtract unsigned	subu \$s1,\$s2,\$s3	\$s1=\$s2-\$s3	3オペランド、オーバーフローを検出しない
	add imm. unsign.	addiu \$s1,\$s2,100	\$s1=\$s2+100	定数を加算、オーバーフローを検出しない
	move fr. copr. reg.	mfc0 \$s1,\$epc	\$s1=\$epc	例外PCと他の特殊レジスタをコピーするために使用
	multiply	mult \$s2,\$s3	Hi,Lo=\$s2×\$s3	HiとLoに64ビットの符号付きの積が収められる
	multiply unsigned	multu \$s2,\$s3	Hi,Lo=\$s2×\$s3	HiとLoに64ビットの符号なしの積が収められる
	divide	div \$s2,\$s3	Lo=\$s2÷\$s3,Hi=\$s2 mod \$s3	Loには商が、Hiには剰余が収められる
divide unsigned	divu \$s2,\$s3	Lo=\$s2÷\$s3,Hi=\$s2 mod \$s3	符号なしの商と剰余	
Move from Hi	mfhi \$s1	\$s1=Hi	Hiのコピーを得るために使用される	
Move from Lo	mflo \$s1	\$s1=Lo	Loのコピーを得るために使用される	
論理演算	and	and \$s1,\$s2,\$s3	\$s1=\$s2&\$s3	2つのレジスタ間のAND
	or	or \$s1,\$s2,\$s3	\$s1=\$s2 \$s3	2つのレジスタ間のOR
	and immediate	andi \$s1,\$s2,100	\$s1=\$s2&100	レジスタと定数間のAND
	or immediate	ori \$s1,\$s2,100	\$s1=\$s2 100	レジスタと定数間のOR
	shift left logical	sli \$s1,\$s2,10	\$s1=\$s2<<10	定数分左へシフト
	shift right logical	srl \$s1,\$s2,10	\$s1=\$s2>>10	定数分右へシフト
データ転送	load word	lw \$s1,100(\$s2)	\$s1=メモリ[\$s2+100]	メモリからレジスタへ転送
	store word	sw \$s1,100(\$s2)	メモリ[\$s2+100]=\$s1	レジスタからメモリへ転送
	load upper imm.	lui \$s1,100	\$s1=100×2 ¹⁶	定数を上位16ビットにロード
条件分岐	branch on equal	beq \$s1,\$s2,25	if(\$s1==\$s2) go to PC+4+100	等しいときに、PC相対分岐
	branch on not eq.	bne \$s1,\$s2,25	if(\$s1!=\$s2) go to PC+4+100	等しくないときに、PC相対分岐
	set on less than	slt \$s1,\$s2,\$s3	if(\$s2<\$s3) \$s1=1;else \$s1=0	より小さいかの判定、2の補数用
	set less than imm.	slti \$s1,\$s2,100	if(\$s2<100) \$s1=1;else \$s1=0	定数よりも小さいかの判定、2の補数用
	set less than uns.	sltu \$s1,\$s2,\$s3	if(\$s2<\$s3) \$s1=1;else \$s1=0	より小さいかの判定、自然数用
	set l.t.imm.uns.	sltiu \$s1,\$s2,100	if(\$s2<100) \$s1=1;else \$s1=0	定数よりも小さいかの判定、自然数用
無条件ジャンプ	jump	j 2500	go to 10000	目的のアドレスへジャンプ
	jump register	jr \$ra	go to \$ra	switch文や手続き呼び出しからの戻りに利用
	jump and link	jal 2500	\$ra=PC+4; go to 10000	手続き呼び出し用

主要命令セットについてのMIPSアセンブリ言語。浮動小数点命令については第4章の図4.47を参照。MIPSの全命令セットのアセンブリ言語は付録Aで解説している。

コンピュータの構成と設計

MIPSの機械語

名前	形式	例						備考
		6ビット	5ビット	5ビット	5ビット	5ビット	6ビット	
add	R	0	18	19	17	0	32	add \$s1,\$s2,\$s3
sub	R	0	18	19	17	0	34	sub \$s1,\$s2,\$s3
addi	I	8	18	17	100			addi \$s1,\$s2,100
addu	R	0	18	19	17	0	33	addu \$s1,\$s2,\$s3
subu	R	0	18	19	17	0	35	subu \$s1,\$s2,\$s3
addiu	I	9	18	17	100			addiu \$s1,\$s2,100
mfc0	R	16	0	17	14	0	0	mfc0 \$s1,\$epc
mult	R	0	18	19	0	0	24	mult \$s2,\$s3
multu	R	0	18	19	0	0	25	multu \$s2,\$s3
div	R	0	18	19	0	0	26	div \$s2,\$s3
divu	R	0	18	19	0	0	27	divu \$s2,\$s3
mfhi	R	0	0	0	17	0	16	mfhi \$s1
mflo	R	0	0	0	17	0	18	mflo \$s1
and	R	0	18	19	17	0	36	and \$s1,\$s2,\$s3
or	R	0	18	19	17	0	37	or \$s1,\$s2,\$s3
andi	I	12	18	17	100			andi \$s1,\$s2,100
ori	I	13	18	17	100			ori \$s1,\$s2,100
sll	R	0	0	18	17	10	0	sll \$s1,\$s2,10
srl	R	0	0	18	17	10	2	srl \$s1,\$s2,10
lw	I	35	18	17	100			lw \$s1,100(\$s2)
sw	I	43	18	17	100			sw \$s1,100(\$s2)
lui	I	15	0	17	100			lui \$s1,100
beq	I	4	17	18	25			beq \$s1,\$s2,25
bne	I	5	17	18	25			bne \$s1,\$s2,25
slt	R	0	18	19	17	0	42	slt \$s1,\$s2,\$s3
slti	I	10	18	17	100			slti \$s1,\$s2,100
sltu	R	0	18	19	17	0	43	sltu \$s1,\$s2,\$s3
sltiu	I	11	18	17	100			sltiu \$s1,\$s2,100
j	J	2	2500					j 2500
jr	R	0	31	0	0	0	8	jr \$ra
jal	J	3	2500					jal 2500

MIPSの命令形式

命令区分	フィールド						備考	
フィールド長	6ビット	5ビット	5ビット	5ビット	5ビット	6ビット	MIPS命令の長さはすべて32ビット	
R形式	op	rs	rt	rd	shamt	funct	算術命令の形式	
I形式	op	rs	rt	address/immediate			データ転送・分岐・即値命令の形式	
J形式	op	target address						ジャンプ命令の形式

MIPSの主要な機械語コード、命令形式と、各フィールドには例としての値を記入した。opフィールドとfunctフィールド(それぞれ6ビット)で命令操作コードを構成する。rsフィールド(5ビット)はソース・レジスタを指示する。rtフィールド(5ビット)も通常はソース・レジスタ。rdフィールド(5ビット)はデスティネーション・レジスタを指示する。shamtはシフト量を表す(5ビット)。各フィールドの値の例はすべて10進数で表示。浮動小数点命令については第4章の図4.47を参照。MIPSの全命令セットの機械語は付録Aで解説している。

レジスタ \$s1 = r17
\$s2 = r18
\$s3 = r19

\$ra = r31
\$zero = r0

てお