1

Machine coding of selected MIPS instructions

R type fields		rs	rt	rd	sa	fn	mnem \$ rd,\$rs,\$r		
(bits):	6	5 5 5 5		6	mnem \$rs,\$rt or mnem \$rd				
Instruction	Decimal Coding in Memory						Coding in Assembly Language		
sll	0	0	2	1	3	0	sll \$1,\$2, 3	shift-logical-left	
jr	0	31	0	0	0	8	jr \$31 <i>(5)</i>	jump-to-register	
mflo	0	0	0	1	0	18	mflo \$1	move-from-LO	
mult	0	2	3	0	0	24	mult \$2,\$3	signed-multiply	
multu	0	2	3	0	0	25	multu \$2,\$3	unsigned-multiply	
add	0	2	3	1	0	32	add \$1,\$2,\$3	add	
addu	0	2	3	1	0	33	addu \$1,\$2,\$3	unsigned-add	
sub	0	2	3	1	0	34	sub \$1,\$2,\$3	subtract	
subu	0	2	3	1	0	35	subu \$1,\$2,\$3	unsigned-subtract	
and	0	2	3	1	0	36	and \$1,\$2,\$3	bitwise-and	
or	0	2	3	1	0	37	or \$1,\$2,\$3	bitwise-or	
slt	0	2	3	1	0	42	slt \$1,\$2,\$3	set-if-less-than	
sltu	0	2	3	1	0	43	sltu \$1,\$2,\$3	unsigned-set-if-less-than	
		immJ (word-address)						mnom hyto, addross	
J- type fields	орс	im	ımJ (v		ddress	s)	mnem hyte-addre	DCC	
(bits):	6		•	26		s)	mnem byte-addre		
, J.	6	im Decimal	Codin	26 g in Me	emory	s)	Coding in Assem		
(bits): Instruction	6		Codin	26 g in Mo	emory	s)	Coding in Assembly 40000	bly Language jump	
(bits):	6		Codin	26 g in Me	emory	5)	Coding in Assem	bly Language	
(bits): Instruction j jal I- type	6	Decimal rs	Codin 10 10 rt	26 g in Mo	emory (4) (4) imm	5)	Coding in Assembly 40000 jal 40000 mnem \$rd,\$rs,imr	bly Language jump jump-and-link m ^{(1) (2)} or	
(bits): Instruction j jal	6 2 3	Decimal	Codin 10 10	26 g in Mo	emory (4) (4)	5)	Coding in Asseml j 40000 jal 40000	bly Language jump jump-and-link m ^{(1) (2)} or	
(bits): Instruction j jal I- type	6 [2 3 opc 6 [Decimal rs	10 10 10 rt 5	26 g in Mo 000 000	emory (4) (4) imm 16 emory		Coding in Assembly 40000 jal 40000 mnem \$rd,\$rs,imr mnem \$rt, displace Coding in Assembly	bly Language jump jump-and-link m (1) (2) or tement(\$rs) (2) bly Language	
(bits): Instruction j jal I- type field-width (bit):	6 [2 3 opc 6 [4	rs 5 Decimal	10 10 10 rt 5	26 g in Me 000 000 g in Me	emory (4) (4) imm 16 emory 00 (3)	3)	Coding in Assembly 40000 jal 40000 mnem \$rd,\$rs,imr mnem \$rt, displace Coding in Assemble \$1,\$2,400	bly Language jump jump-and-link m (1) (2) or ement(\$rs) (2) bly Language branch-if-equal	
(bits): Instruction j jal I- type field-width (bit): Instruction beq bne	6 [2 3 opc 6 [4 5 5	rs 5 Decimal	10 10 10 rt 5 Codin 2	26 g in Me 000 000 g in Me	emory (4) (4) imm 16 emory 00 (3)	3)	Coding in Assemble j 40000 jal 40000 mnem \$rd,\$rs,imr mnem \$rt, displace Coding in Assemble j \$1,\$2,400 bne \$1,\$2,400	bly Language jump jump-and-link on (1) (2) or tement(\$rs) (2) bly Language branch-if-equal branch-if-not-equal	
(bits): Instruction j jal I- type field-width (bit): Instruction beq bne addi	6 [2 3 opc 6 [4 5 8 8	rs 5 Decimal 1 1	10 10 10 rt 5 Codin 2 2	26 g in Me 000 000 g in Me	emory (4) (4) imm 16 emory 00 (3) 100	3)	Coding in Assembly 40000 jal 40000 mnem \$rd,\$rs,imr mnem \$rt, displace Coding in Assemble \$1,\$2,400 bne \$1,\$2,400 addi \$1,\$2,100	bly Language jump jump-and-link m (1) (2) or ement(\$rs) (2) bly Language branch-if-equal branch-if-not-equal	
(bits): Instruction j jal I- type field-width (bit): Instruction beq bne addi slti	6 [2 3 opc 6	rs 5 Decimal 1 1 2 2	10 10 10 rt 5 Codin 2 2 1	26 g in Me 000 000 g in Me	emory (4) (4) imm 16 emory 00 (3)	3)	Coding in Assemble j 40000 jal 40000 mnem \$rd,\$rs,imr mnem \$rt, displace Coding in Assemble q \$1,\$2,400 bne \$1,\$2,400 addi \$1,\$2,100 slti \$1,\$2,100	bly Language jump jump-and-link m (1) (2) or tement(\$rs) (2) bly Language branch-if-equal branch-if-not-equal add immediate set-less-than-immed.	
(bits): Instruction j jal I- type field-width (bit): Instruction beq bne addi	6 [2 3 opc 6 [4 5 8 8	rs 5 Decimal 1 1 2 2 0	10 10 10 rt 5 Codin 2 2 1 1	26 g in Me 000 000 g in Me	emory (4) (4) imm 16 emory 00 (3) 100	3)	Coding in Assembly 40000 jal 40000 mnem \$rd,\$rs,imr mnem \$rt, displace Coding in Assemble \$1,\$2,400 bne \$1,\$2,400 addi \$1,\$2,100 slti \$1,\$2,100 lui \$1,100	bly Language jump jump-and-link m (1) (2) or ement(\$rs) (2) bly Language branch-if-equal branch-if-not-equal add immediate set-less-than-immed. load-upper-immediate	
(bits): Instruction j jal I- type field-width (bit): Instruction beq bne addi slti	6 [2 3 opc 6	rs 5 Decimal 1 1 2 2	10 10 10 rt 5 Codin 2 2 1	26 g in Me 000 000 g in Me	emory (4) (4) imm 16 emory 00 (3) 100 100	3)	Coding in Assemble j 40000 jal 40000 mnem \$rd,\$rs,imr mnem \$rt, displace Coding in Assemble q \$1,\$2,400 bne \$1,\$2,400 addi \$1,\$2,100 slti \$1,\$2,100	bly Language jump jump-and-link m (1) (2) or ement(\$rs) (2) bly Language branch-if-equal branch-if-not-equal add immediate set-less-than-immed. load-upper-immediate	
(bits): Instruction j jal I- type field-width (bit): Instruction beq bne addi slti lui	6 [2 3 opc 6	rs 5 Decimal 1 1 2 2 0	10 10 10 rt 5 Codin 2 2 1 1	26 g in Me 000 000 g in Me	emory (4) imm 16 emory 00 (3) 100 100	3)	Coding in Assembly 40000 jal 40000 mnem \$rd,\$rs,imr mnem \$rt, displace Coding in Assemble \$1,\$2,400 bne \$1,\$2,400 addi \$1,\$2,100 slti \$1,\$2,100 lui \$1,100	bly Language jump jump-and-link m (1) (2) or tement(\$rs) (2) bly Language branch-if-equal branch-if-not-equal w add immediate set-less-than-immed. load-upper-immediate	

(2) base/displacement addressing (16-bit, byte-address)

(4) 26-bit word-address (5) register-addressing.

MIPS Assembly convention for general-purpose register usage:

Will 6 7 33 cm biy convention for general purpose register asage.						
\$zero	\$0 , contains zero	\$at (reserved)	\$1, temporary for pseudo-codes			
\$v0 \$v1	\$2 \$3, return values from expression and procedure evaluation	\$a0 \$a3	\$4 \$7, arguments for procedure			
\$t0 \$t7	\$8 \$15 , temporaries	\$s0 \$s7	\$16 \$23, saved variables			
\$t8 \$t9	\$24 \$25, more temporaries	\$gp	\$28, global pointer			
\$sp	\$29, stack pointer	\$fp	\$30, frame pointer			
\$ra	\$31, return address					