MIPS Reference Data

ATV.	Ì
	y

CORE INSTRUCTI	CORE INSTRUCTION SET OPCODE							
NAME, MNEMO	NIC	FOR- MAT			/ FUNCT (Hex)			
Add	add	R	R[rd] = R[rs] + R[rt]	(1)	0 / 20 _{hex}			
Add Immediate	addi	I	R[rt] = R[rs] + SignExtImm	(1,2)	8 _{hex}			
Add Imm. Unsigned	addiu	I	R[rt] = R[rs] + SignExtImm	(2)	9 _{hex}			
Add Unsigned	addu	R	R[rd] = R[rs] + R[rt]		0 / 21 _{hex}			
And	and	R	R[rd] = R[rs] & R[rt]		0 / 24 _{hex}			
And Immediate	andi	I	R[rt] = R[rs] & ZeroExtImm	(3)	c _{hex}			
Branch On Equal	beq	I	if(R[rs]==R[rt]) PC=PC+4+BranchAddr	(4)	4 _{hex}			
Branch On Not Equa	bne	I	if(R[rs]!=R[rt]) PC=PC+4+BranchAddr	(4)	5 _{hex}			
Jump	j	J	PC=JumpAddr	(5)	2 _{hex}			
Jump And Link	jal	J	R[31]=PC+4;PC=JumpAddr	(5)	3 _{hex}			
Jump Register	jr	R	PC=R[rs]		$0/08_{hex}$			
Load Byte Unsigned	lbu	I	R[rt]={24'b0,M[R[rs] +SignExtImm](7:0)}	(2)	24 _{hex}			
Load Halfword Unsigned	lhu	I	R[rt]={16'b0,M[R[rs] +SignExtImm](15:0)}	(2)	25 _{hex}			
Load Linked	11	I	R[rt] = M[R[rs] + SignExtImm]	(2,7)	30_{hex}			
Load Upper Imm.	lui	I	$R[rt] = \{imm, 16'b0\}$		f _{hex}			
Load Word	lw	I	R[rt] = M[R[rs] + SignExtImm]	(2)	23 _{hex}			
Nor	nor	R	$R[rd] = \sim (R[rs] \mid R[rt])$		$0/27_{hex}$			
Or	or	R	$R[rd] = R[rs] \mid R[rt]$		0 / 25 _{hex}			
Or Immediate	ori	I	R[rt] = R[rs] ZeroExtImm	(3)	d _{hex}			
Set Less Than	slt	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0		0 / 2a _{hex}			
Set Less Than Imm.	slti	I	R[rt] = (R[rs] < SignExtImm)? 1	: 0 (2)	a _{hex}			
Set Less Than Imm. Unsigned	sltiu	I	R[rt] = (R[rs] < SignExtImm) ? 1:0	(2,6)	b _{hex}			
Set Less Than Unsig.		R	R[rd] = (R[rs] < R[rt]) ? 1 : 0	(6)	0 / 2b _{hex}			
Shift Left Logical	sll	R	$R[rd] = R[rt] \ll shamt$		0 / 00 _{hex}			
Shift Right Logical	srl	R	R[rd] = R[rt] >>> shamt		0 / 02 _{hex}			
Store Byte	sb	I	M[R[rs]+SignExtImm](7:0) = R[rt](7:0)	(2)	28 _{hex}			
Store Conditional	sc	I	M[R[rs]+SignExtImm] = R[rt]; R[rt] = (atomic) ? 1 : 0	(2,7)	38 _{hex}			
Store Halfword	sh	I	M[R[rs]+SignExtImm](15:0) = R[rt](15:0)	(2)	29 _{hex}			
Store Word	sw	I	M[R[rs]+SignExtImm] = R[rt]	(2)	2b _{hex}			
Subtract	sub	R	R[rd] = R[rs] - R[rt]	(1)	0 / 22 _{hex}			
$\label{eq:subtract} \begin{tabular}{lllllllllllllllllllllllllllllllllll$								

BASIC INSTRUCTION FORMATS

R	opcode	rs	rt	rd	shamt	funct
	31 26	25 21	20 16	15 11	10 6	5 0
I	opcode	rs	rt		immediate	•
	31 26	25 21	20 16	15		0
J	opcode			address		
	31 26	25	-	-	-	0

ARITHMETIC CORE INSTRUCTION SET

			2	/ FMT /FT
		FOR-		/ FUNCT
NAME, MNEMO	ONIC	MAT	OPERATION	(Hex)
Branch On FP True	bc1t	FI	if(FPcond)PC=PC+4+BranchAddr (4)	11/8/1/
Branch On FP False	bc1f	FI	if(!FPcond)PC=PC+4+BranchAddr(4)	11/8/0/
Divide	div	R	Lo=R[rs]/R[rt]; Hi=R[rs]%R[rt]	0//-1a
Divide Unsigned	divu	R	Lo=R[rs]/R[rt]; Hi=R[rs]%R[rt] (6)	0//-1b
FP Add Single	add.s	FR	F[fd] = F[fs] + F[ft]	11/10//0
FP Add Double	add.d	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} + {F[ft],F[ft+1]}$	11/11//0
FP Compare Single	c.x.s*	FR	FPcond = (F[fs] op F[ft]) ? 1 : 0	11/10//y
FP Compare Double	c.x.d*	FR	$FPcond = ({F[fs],F[fs+1]}) op $ ${F[ft],F[ft+1]})?1:0$	11/11//y
			==, <, or <=) (y is 32, 3c, or 3e)	
FP Divide Single	div.s	FR	F[fd] = F[fs] / F[ft]	11/10//3
FP Divide Double	div.d	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} / {F[ft],F[ft+1]}$	11/11//3
FP Multiply Single	mul.s	FR	F[fd] = F[fs] * F[ft]	11/10//2
FP Multiply Double	mul.d	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} * {F[ft],F[ft+1]}$	11/11//2
FP Subtract Single	sub.s	FR	F[fd]=F[fs] - F[ft]	11/10//1
FP Subtract Double	sub.d	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} - {F[ft],F[ft+1]}$	11/11//1
Load FP Single	lwc1	I	F[rt]=M[R[rs]+SignExtImm] (2)	31//
Load FP Double	ldc1	I	F[rt]=M[R[rs]+SignExtImm]; (2) F[rt+1]=M[R[rs]+SignExtImm+4]	35//
Move From Hi	mfhi	R	R[rd] = Hi	0 ///10
Move From Lo	mflo	R	R[rd] = Lo	0 //-12
Move From Control	mfc0	R	R[rd] = CR[rs]	10 /0//0
Multiply	mult	R	$\{Hi,Lo\} = R[rs] * R[rt]$	0//-18
Multiply Unsigned	multu	R	$\{Hi,Lo\} = R[rs] * R[rt] $ (6)	
Shift Right Arith.	sra	R	R[rd] = R[rt] >> shamt	0//-3
Store FP Single	swcl	I	M[R[rs]+SignExtImm] = F[rt] (2)	
Store FP Double	sdc1	I	M[R[rs]+SignExtImm] = F[rt]; (2) M[R[rs]+SignExtImm+4] = F[rt+1]	3d//

OPCODE

FLOATING-POINT INSTRUCTION FORMATS

FR	opcode	fmt	ft	fs	fd	funct
	31 26	25 21	20 16	15 11	10 6	5 0
FI	opcode	fmt	ft		immediate	2
	31 26	25 21	20 16	15		0

PSEUDOINSTRUCTION SET

NAME	MNEMONIC	C OPERATION
Branch Less Than	blt	if(R[rs] < R[rt]) PC = Label
Branch Greater Than	bgt	if(R[rs]>R[rt]) PC = Label
Branch Less Than or Equal	ble	$if(R[rs] \le R[rt]) PC = Label$
Branch Greater Than or Equa	l bge	if(R[rs]>=R[rt]) PC = Label
Load Immediate	li	R[rd] = immediate
Move	move	R[rd] = R[rs]

REGISTER NAME, NUMBER, USE, CALL CONVENTION

NAME	NUMBER	USE	PRESERVEDACROSS A CALL?
\$zero	0	The Constant Value 0	N.A.
\$at	1	Assembler Temporary	No
\$v0-\$v1	2-3	Values for Function Results and Expression Evaluation	No
\$a0-\$a3	4-7	Arguments	No
\$t0-\$t7	8-15	Temporaries	No
\$s0-\$s7	16-23	Saved Temporaries	Yes
\$t8-\$t9	24-25	Temporaries	No
\$k0-\$k1	26-27	Reserved for OS Kernel	No
\$gp	28	Global Pointer	Yes
\$sp	29	Stack Pointer	Yes
\$fp	30	Frame Pointer	Yes
\$ra	31	Return Address	Yes

Copyright 2009 by Elsevier, Inc., All rights reserved. From Patterson and Hennessy, Computer Organization and Design, 4th ed.

OPCODES, BASE CONVERSION, ASCII SYMBOLS

Opcode Funct (31:26) (5:0) (5:			(2) MIDS						Horro	A CCII
Secondary Seco			(2) MIPS	ъ.	Deci-			Deci-		I
(31:26) (5:00)	*			Binary						I
Substrain Subs		(=)		00 0000						
Section	(1)	sll								
jal										
Deg										
Deb	jal	sra	div.f							
Diez Srav neg.f 00 0110 6 6 ACK 70 46 F	beq	sllv	sqrt. f							
Dgtz	bne		abs f							
addi	blez	srlv	mov.f	00 0110	6	6	ACK	70	46	F
addiu jalr	bgtz	srav	${\tt neg.}\!f$	00 0111		7	BEL	71	47	G
Sitiu movz 00 1010 10 a LF 74 4a J K andi syscall round.wf 00 1010 12 c FF 76 4c L c c L c c L c c	addi	jr		00 1000	8	8	BS	72	48	Н
Stiu	addiu	jalr		00 1001	9	9	HT	73	49	I
Stiu	slti	movz		00 1010	10	a	LF	74	4a	J
andi	sltiu			00 1011		b	VT	75	4b	
ori break xori trunc.wf ceil.wf 00 1101 13 d CR 77 4d M 1ui sync floor.wf 00 1111 15 f SI 79 4f N mfhi mfhi 01 0000 16 10 DLE 80 50 P mflo movz.f 01 0010 18 12 DC2 82 52 R mflo movn.f 01 0010 19 13 DC3 83 53 S 01 0101 20 14 DC4 84 54 T 01 0101 21 15 NAK 85 55 U 01 0101 22 16 SYN 86 56 V mult 01 1010 22 16 SYN 86 56 V mult 01 1001 25 19 EM 89 59 Y div 01 1010 26 1a			round.w.f	00 1100	12	С	FF	76	4c	
Note										
101										
(2) mthi mflo movz.f 01 0000 16 10 DLE 80 50 P mflo mflo movz.f 01 0010 18 12 DC2 82 52 R mtlo movn.f 01 0011 19 13 DC3 83 53 S 01 01 010 20 14 DC4 84 54 T 01 01010 21 15 NAK 85 55 U 01 0110 22 16 SYN 86 56 V 01 0111 23 17 ETB 87 57 W 01 0110 24 18 CAN 88 58 X mult 01 1000 24 18 CAN 88 58 X mult 01 1010 25 19 EM 89 59 Y div div 01 1010 26 1a SUB 90 5a Z divu 01 1010 28 1c FS 92 5c \ 01 110 29 1d GS 93 5d] 01 1110 29 1d GS 93 5d] 01 1110 30 1e RS 94 5e \ 01 1111 31 1f US 95 5f \ 01 1110 30 1e RS 94 5e \ 01 1111 31 1f US 95 5f \ 1h add cvt.s.f 10 0001 33 21 ! 97 61 a lbu and cvt.w.f 10 1001 35 23 # 99 63 c lbu and cvt.w.f 10 1001 35 23 # 99 63 c lbu and cvt.w.f 10 1001 37 25 % 100 64 d lhu or 10 0110 38 26 & 102 66 f la sw sub 10 1001 37 25 % 100 64 d lhu or 10 1011 39 27 ' 103 67 g sb sh 10 1000 42 28 (104 68 h lo 10 1001 38 26 & 102 66 f la sw sub 10 1001 34 22 " 100 64 d lhu or 10 1011 37 25 % 100 64 d lhu or 10 1011 37 25 % 100 64 d lhu or 10 1011 37 25 % 100 64 d lhu or 10 1011 37 25 % 100 64 d lhu or 10 1011 37 25 % 100 64 d lhu or 10 1011 37 25 % 100 64 d lhu or 10 1011 37 25 % 100 66 f lo 10 101 37 25 % 100 66 f lo 10 101 37 25 % 100 66 f lo 10 101 37 25 % 100 66 f lo 10 101 37 25 % 100 66 f lo 10 101 37 25 % 100 66 f lo 10 101 37 25 % 100 66 f lo 10 101 37 25 % 100 66 f lo 10 101 37 25 % 100 66 m lo 10 101 44 22 , 100 66 m lo 10 101 44 22 , 100 66 m lo 10 101 44 22 1 lo 10 10 60 a lo 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2		sync								
(2) mthi mflo movz.f 01 0001 17 11 DC1 81 51 Q mtlo movn.f 01 0010 18 12 DC2 82 52 R 01 01 0001 19 13 DC3 83 53 S 01 01 0100 20 14 DC4 84 54 T 01 01010 21 15 NAK 85 55 U 01 01010 22 16 SYN 86 56 V 01 0111 23 17 ETB 87 57 W 01 0111 23 17 ETB 87 57 W 01 01 0101 25 19 EM 89 59 Y 01 01 010 26 18 SUB 90 5a Z 01 01 010 26 18 SUB 90 5a Z 01 01 010 27 1b ESC 91 5b [01 1101 29 1d GS 93 5d] 01 1101 29 1d GS 93 5d] 01 1101 29 1d GS 93 5d] 01 1110 30 1e RS 94 5e ^ 01 1111 31 1f US 95 5f _ 10 10 000 32 20 Space 96 60 5	101		110011119							
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	(2)									
mtlo	(2)		movz f							
01 0100			- ·							
		IIICIO	illovii.j							
Mult										
mult 01 1000 24 18 CAN 88 58 X div 01 1001 25 19 EM 89 59 Y divu 01 1010 26 1a SUB 90 5a Z divu 01 1011 27 lb ESC 91 5b [01 1101 29 ld GS 93 5d]] 01 1101 29 ld GS 93 5d]] 01 1110 30 le RS 94 5e ^ 1 01 1111 31 lf US 95 5f _ 1 1b add cvt.s.f 10 0000 32 20 Space 96 60 * 1h addu cvt.d.f 10 0001 33 21 ! 97 61 a a 1w sub 10 0011 35 23 # 99 63 c b 1bu and cvt.w.f 10 0100 36 24 \$ 100 64 d d 1bu or 10 0101 37 25 % 101 65 e e 1wr										-
multu 01 1001 25 19 EM 89 59 Y div 01 1010 26 1a SUB 90 5a Z 01 1010 28 1c FSC 91 5b [01 1110 28 1c FSS 92 5c \ 01 1110 30 1e RS 94 5e \ 01 1111 31 1f US 95 5f _ 1b add cvt.s.f 10 0000 32 20 Space 96 60 \ 1h addu cvt.d.f 10 0001 34 22 " 98 62 b 1w subu 10 0101 37 25 % 101 65 e 1bu and cvt.w.f 10 0100 36 24 \$ 100 64 d 1bu and cvt.w.f 10 0100 32										
div divu										
Dot 1100 28										L
Ol 1101 29		divu								
Ol 1110 30 1e RS 94 5e A Ol 1111 31 1f US 95 5f										\
11										
1b										^
1h addu cvt.s.f 10 0000 32 20 space 50 60 lw subu 10 0001 34 22 " 98 62 b lw subu 10 0010 34 22 " 98 62 b lw subu 10 0010 36 24 \$ 100 64 d lbu and cvt.w.f 10 0100 36 24 \$ 100 64 d lbu and cvt.w.f 10 0101 37 25 % 101 65 e lw xor 10 0111 39 27 ' 103 67 g sb 10 1000 40 28 (104 68 h swl slt 10 1010 42 2a * 106 6a j swl slt 10 1011 43 2b + 107 6b k										
1w sub										٠,
lw subu 10 0011 35 23 # 99 63 c lbu and cvt.w.f 10 0100 36 24 \$ 100 64 d lhu or 10 0101 37 25 % 101 65 e lwr xor 10 0110 38 26 & 102 66 f nor 10 0101 39 27 ' 103 67 g sb 10 1000 40 28 (104 68 h sw slt 10 1001 42 2a * 106 6a j sw slt 10 1010 42 2a * 106 6a j sw slt 10 1010 42 2a * 106 6a j sw slt 10 1011 43 2b + 107 6b k <			cvt.d.f							
1bu and cvt.w,f 10 0100 36 24 \$ 100 64 d	lwl	sub								b
Thu or 10 0101 37 25 % 101 65 e		subu								
lwr xor 10 0110 38 26 & 102 66 f sb 10 0111 39 27 ' 103 67 g sb 10 1000 40 28 (104 68 h sh 10 1001 41 29) 105 69 i swl slt 10 1010 42 2a * 106 6a j sw sltu 10 1101 43 2b + 107 6b k sw sltu 10 1101 45 2d - 109 6d m swr 10 1101 45 2d - 109 6d m cache 10 1111 47 2f / 111 6f n cache 10 1111 47 2f / 111 6f n cache 11 10001 49 31 1	lbu	and	$\mathtt{cvt.w.} f$							d
Nor	lhu	or								
Sb	lwr	xor		10 0110				102	66	f
sh 10 1001 41 29) 105 69 i swl slt 10 1010 42 2a * 106 6a j sw sltu 10 1011 43 2b + 107 6b k 10 1100 44 2c , 108 6c 1 10 1101 45 2d - 109 6d m swr 10 1110 46 2e . 110 6e n cache 10 1111 47 2f / 111 6f o 11 tge c.f.f 11 0000 48 30 0 112 70 p lwc1 tgeu c.unf 11 0001 50 32 2 114 72 r pref tttu c.eaf 11 0010 50 32 2 114 72 r pref 11 0010 50 32 <t< td=""><td></td><td>nor</td><td></td><td></td><td></td><td></td><td></td><td></td><td>67</td><td>g</td></t<>		nor							67	g
swl slt 10 1010 42 2a * 106 6a j sw sltu 10 1011 43 2b + 107 6b k 10 1100 44 2c , 108 6c 1 10 1101 45 2d - 109 6d m swr 10 1110 46 2e . 110 6e n cache 10 1111 47 2f / 111 6f o n 11 tge c.f.f 11 0000 48 30 0 112 70 p 1wc1 tgeu c.un.f 11 0001 50 32 2 114 72 r r pref tltu c.ueq.f 11 0011 51 33 3 115 73 s t t 11 010 52 34 4 116 74 t t 11 010 52<	sb			10 1000	40	28	(104	68	h
sw sltu 10 1011 43 2b + 107 6b k 10 1100 44 2c , 108 6c 1 10 1100 44 2c , 108 6c 1 swr 10 1110 45 2d - 109 6d m swr 10 1111 47 2f / 111 6e n cache 10 1111 47 2f / 111 6e n cache 11 10000 48 30 0 112 70 p lwc1 tgeu c.unf 11 0001 50 32 2 114 72 r p pref tltu c.ueq,f 11 0010 50 32 2 114 72 r r r r r r r r r r r r r r	sh			10 1001	41	29)	105	69	i
10 1100	swl	slt		10 1010	42	2a	*	106	6a	j
swr 10 1101 45 2d - 109 6d m cache 10 1110 46 2e . 110 6e n cache 10 1111 47 2f / 111 6f o 11 tge c.f.f 11 0000 48 30 0 112 70 p lwc1 tgeu c.unf 11 0001 49 31 1 113 71 q lwc2 tlt c.eqf 11 0010 50 32 2 114 72 r r pref tltu c.ueq.f 11 0011 51 33 3 115 73 s s 10c c.ult,f 11 0100 52 34 4 116 74 t t 1dc1 c.ult,f 11 0101 53 35 5 117 75 u 11 1011 53 35 5 117 75 u	sw	sltu		10 1011	43	2b	+	107	6b	k
swr 10 1101 45 2d - 109 6d m cache 10 1110 46 2e . 110 6e n cache 10 1111 47 2f / 111 6f o 11 tge c.f.f 11 0000 48 30 0 112 70 p lwc1 tgeu c.unf 11 0001 49 31 1 113 71 q lwc2 tlt c.eqf 11 0010 50 32 2 114 72 r r pref tltu c.ueq.f 11 0011 51 33 3 115 73 s s 10c c.ult,f 11 0100 52 34 4 116 74 t t 1dc1 c.ult,f 11 0101 53 35 5 117 75 u 11 1011 53 35 5 117 75 u				10 1100	44	2c	,	108	6c	1
cache 10 1111 47 2f / 111 6f o 11 tge c.f.f 11 0000 48 30 0 112 70 p lwc1 tgeu c.un.f 11 0001 49 31 1 113 71 q lwc2 tlt c.eq.f 11 0010 50 32 2 114 72 r r pref tltu c.ueq.f 11 0010 51 33 3 115 73 s s teq c.olt.f 11 0100 52 34 4 116 74 t t d 11010 53 35 5 117 75 u d d c.ule.f 11 0110 54 36 6 118 76 v c.ule.f 11 0111 55 37 7 119 77 w s swc c.sf.f 11 1000 56 38 8 120 78					45	2d	-	109	6d	m
cache 10 1111 47 2f / 111 6f o 11 tge c.f.f 11 0000 48 30 0 112 70 p lwc1 tgeu c.un.f 11 0001 49 31 1 113 71 q lwc2 tlt c.eq.f 11 0010 50 32 2 114 72 r pref tltu c.ueq.f 11 0010 51 33 3 115 73 s teq c.olt.f 11 0100 52 34 4 116 74 t ldc1 c.ult.f 11 0101 53 35 5 117 75 u ldc2 tne c.ole.f 11 0110 54 36 6 118 76 v c.ule.f 11 0111 55 37 7 119 77 w sc c.sf.f 11 1000 56	swr			10 1110	46	2e		110	6e	n
lwc1 tgeu c.un.f 11 0001 49 31 1 113 71 q lwc2 tlt c.eq.f 11 0010 50 32 2 114 72 r pref tltu c.ueq.f 11 0011 51 33 3 115 73 s teq c.olt.f 11 0100 52 34 4 116 74 t ldc1 c.ult.f 11 0101 53 35 5 117 75 u ldc2 tne c.ole.f 11 0110 54 36 6 118 76 v c.ule.f 11 0101 55 37 7 119 77 w sc c.sf.f 11 1000 56 38 8 120 78 x swc1 c.ngl.f 11 1001 57 39 9 121 79 y swc2 c.seq.f 11 1010 58<	cache				47	2f	/	111	6f	0
lwc1 tgeu c.un.f 11 0001 49 31 1 113 71 q lwc2 tlt c.eq.f 11 0010 50 32 2 114 72 r pref tltu c.ueq.f 11 0011 51 33 3 115 73 s teq c.olt.f 11 0100 52 34 4 116 74 t ldc1 c.ult.f 11 0101 53 35 5 117 75 u ldc2 tne c.ole.f 11 0110 54 36 6 118 76 v c.ule.f 11 0101 55 37 7 119 77 w sc c.sf.f 11 1000 56 38 8 120 78 x swc1 c.ngl.f 11 1001 57 39 9 121 79 y swc2 c.seq.f 11 1010 58<		tge	c.f.f							
lwc2 tlt c.eq.f 11 0010 50 32 2 114 72 r pref tltu c.ueq.f 11 0011 51 33 3 115 73 s teq c.olt,f 11 0100 52 34 4 116 74 t ldc1 c.ult,f 11 0101 53 35 5 117 75 u ldc2 tne c.ole,f 11 0110 54 36 6 118 76 v c.ule,f 11 0111 55 37 7 119 77 w sc c.sf,f 11 1000 56 38 8 120 78 x swc1 c.ngl,f 11 1001 57 39 9 121 79 y swc2 c.seq,f 11 1010 58 3a : 122 7a z c.ngl,f 11 1010 59 3b ;										- 1
pref tltu c.ueq.f 11 0011 51 33 3 115 73 s teq c.olt.f 11 0100 52 34 4 116 74 t ldc1 c.ult.f 11 0101 53 35 5 117 75 u ldc2 tne c.ole.f 11 0110 54 36 6 118 76 v c.ule.f 11 0111 55 37 7 119 77 w sc c.sf.f 11 1000 56 38 8 120 78 x swc1 c.ngl.f 11 1001 57 39 9 121 79 y swc2 c.seq.f 11 1010 58 3a : 122 7a z c.ngl.f 11 1011 59 3b ; 123 7b { sdc1 c.nge,f 11 1100 60 3c 124 7c <td></td>										
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$										
ldc1 c.ult.f 11 0101 53 35 5 117 75 u ldc2 tne c.ole.f 11 0110 54 36 6 118 76 v c.ule.f 11 0111 55 37 7 119 77 w sc c.sf.f 11 1000 56 38 8 120 78 x swc1 c.ngle.f 11 1001 57 39 9 121 79 y swc2 c.seq.f 11 1010 58 3a : 122 7a z c.ngl.f 11 1011 59 3b ; 123 7b { sdc1 c.nge,f 11 1100 60 3c <	FIJE									
ldc2 tne c.ole.f 11 0110 54 36 6 118 76 v c.ule.f 11 0111 55 37 7 119 77 w sc c.sf.f 11 1000 56 38 8 120 78 x swc1 c.ngle.f 11 1001 57 39 9 121 79 y swc2 c.seq.f 11 1010 58 3a : 122 7a z c.ngl.f 11 1101 59 3b ; 123 7b { sdc1 c.nge.f 11 1100 60 3c < 124 7c sdc1 c.nge.f 11 1101 61 3d = 125 7d } sdc2 c.le.f 11 1110 62 3e > 126 7e ~ c.ngt.f 11 1111 63 3f ? 127 7f DEL	ldc1	004						-		
c.ule.f 11 0111 55 37 7 119 77 w sc c.sf.f 11 1000 56 38 8 120 78 x swc1 c.ngle.f 11 1001 57 39 9 121 79 y swc2 c.seq.f 11 1010 58 3a : 122 7a z c.ngl.f 11 1011 59 3b ; 123 7b { sdc1 c.nge.f 11 1100 60 3c < 124 7c sdc1 sdc2 c.le.f 11 1101 62 3e > 126 7e ~ c.ngt.f 11 1111 63 3f ? 127 7f DEL		tne								
sc c.sf.f 11 1000 56 38 8 120 78 x swc1 c.ngle,f 11 1001 57 39 9 121 79 y swc2 c.seq,f 11 1010 58 3a : 122 7a z c.ngl,f 11 1011 59 3b ; 123 7b { sdc1 c.nge,f 11 1100 60 3c < 124 7c sdc1 c.nge,f 11 1101 61 3d = 125 7d } sdc2 c.le,f 11 1110 62 3e > 126 7e ~ c.ngt,f 11 1111 63 3f ? 127 7f DEL	1402	CIIC								
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	80									
swc2 c.seq.f 11 1010 58 3a : 122 7a z c.ngl.f 11 1011 59 3b ; 123 7b { c.lt.f 11 1100 60 3c < 124 7c sdc1 c.nge.f 11 1101 61 3d = 125 7d } sdc2 c.le.f 11 1110 62 3e > 126 7e ~ c.ngt.f 11 1111 63 3f ? 127 7f DEL										
c.ngl.f 11 1011 59 3b ; 123 7b { c.lt.f 11 1100 60 3c <										
c.lt.f 11 1100 60 3c 124 7c sdc1 sdc1 c.nge.f 11 1101 61 3d = 125 7d } sdc2 c.le.f 11 1110 62 3e > 126 7e ~ c.ngt.f 11 1111 63 3f ? 127 7f DEL	SWCZ									
sdc1 c.nge.f 11 1101 61 3d = 125 7d } sdc2 c.le.f 11 1110 62 3e > 126 7e ~ c.ngt.f 11 1111 63 3f ? 127 7f DEL										1
sdc2 c.le.f 11 1110 62 3e > 126 7e ~	1 -									
c.ngt.f 11 1111 63 3f ? 127 7f DEL										
	sdc2									
	(1) 5 = -	da(21:20)		11 1111	63	31		12/	/1	DEL

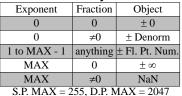
 $^{(1) \}text{ opcode}(31:26) == 0$

IEEE 754 FLOATING-POINT STANDARD

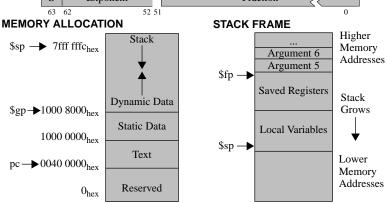
IEEE 754 Symbols

 $(-1)^S \times (1 + Fraction) \times 2^{(Exponent - Bias)}$ where Single Precision Bias = 127, Double Precision Bias = 1023.

IEEE Single Precision and Double Precision Formats:



S	Exponent	Fraction	
31	30 23	2	0
S	Exponent	Fractio	on >
	62	50 51	



DATA ALIGNMENT

Double Word							
Word Word							
Halfv	alfword Halfword		Half	fword	Halfword		
Byte	Byte	Byte	Byte	Byte	Byte	Byte	Byte
0	1	2	3	4	5	6	7

Value of three least significant bits of byte address (Big Endian)

EXCEPTION CONTROL REGISTERS: CAUSE AND STATUS

В	Interrupt		Exception Code	
D	Mask		Code	
31	15	8	6	2
	Pending		U	ΕI
	Interrupt		M	LE
	15	8	4	1 0

BD = Branch Delay, UM = User Mode, EL = Exception Level, IE =Interrupt Enable **EXCEPTION CODES**

NOET HON CODES												
Number	Name	Cause of Exception	Number	Name	Cause of Exception							
0	Int	Interrupt (hardware)	9	Вр	Breakpoint Exception							
4	AdEL	Address Error Exception	10	RI	Reserved Instruction							
		(load or instruction fetch)	10		Exception							
5	AdES	Address Error Exception	11	CpU	Coprocessor							
		(store)	11		Unimplemented							
6	IBE	Bus Error on	12	Ov	Arithmetic Overflow							
0		Instruction Fetch	12		Exception							
7	DBE	Bus Error on	13	Tr	Tron							
		Load or Store			Trap							
8	Sys	Syscall Exception	15	FPE	Floating Point Exception							

SIZE PREFIXES (10^x for Disk, Communication; 2^x for Memory)

	PRE-		PRE-		PRE-		PRE-
SIZE	FIX	SIZE	FIX	SIZE	FIX	SIZE	FIX
$10^3, 2^{10}$	Kilo-	$10^{15}, 2^{50}$	Peta-	10-3	milli-	10-15	femto-
$10^6, 2^{20}$	Mega-	$10^{18}, 2^{60}$	Exa-	10 ⁻⁶	micro-	10-18	atto-
$10^9, 2^{30}$	Giga-	$10^{21}, 2^{70}$	Zetta-	10-9	nano-	10-21	zepto-
$10^{12}, 2^{40}$	Tera-	$10^{24}, 2^{80}$	Yotta-	10-12	pico-	10-24	yocto-

The symbol for each prefix is just its first letter, except μ is used for micro.

⁽²⁾ opcode(31:26) == 17_{ten} (11_{hex}); if fmt(25:21)== 16_{ten} (10_{hex}) f = s (single); if $fmt(25:21) == 17_{ten} (11_{hex}) f = d (double)$