MIPS Reference Data

1

CORE INSTRUCTION	ON SE	Т			OPCODE			
		FOR-			/ FUNCT			
NAME, MNEMO		MAT		(1)	(Hex)			
Add	add	R	R[rd] = R[rs] + R[rt]	()	0 / 20 _{hex}			
Add Immediate	addi	I	R[rt] = R[rs] + SignExtImm	(1,2)	8 _{hex}			
Add Imm. Unsigned		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	Ι	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 Equal	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+8;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)				
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	$R[rt] = R[rs] \mid ZeroExtImm$	(3)	*****			
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.	sltu	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)	$28_{ m hex}$			
Store Conditional	sc	I	M[R[rs]+SignExtImm] = R[rt]; $R[rt] = (atomic) ? 1 : 0$	(2,7)	$38_{ m hex}$			
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)	$2b_{\text{hex}}$			
Subtract	sub	R	R[rd] = R[rs] - R[rt]	(1)	0 / 22 _{hex}			
Subtract Unsigned	subu	R	R[rd] = R[rs] - R[rt]		$0/23_{hex}$			
	(2) Sig (3) Ze (4) Br) May cause overflow exception) SignExtImm = { 16{immediate[15]}, immediate }) ZeroExtImm = { 16{1b'0}, immediate }) BranchAddr = { 14{immediate[15]}, immediate, 2'b0 }						

- (5) $JumpAddr = \{ PC+4[31:28], address, 2'b0 \}$
- (6) Operands considered unsigned numbers (vs. 2's comp.)
- (7) Atomic test&set pair; R[rt] = 1 if pair atomic, 0 if not atomic

BASIC INSTRUCTION FORMATS

\mathbf{R}		opcode		rs		rt	rd		shamt	funct	
	31	26	25	21	20	16	15	11	10	5 5	0
I		opcode		rs		rt			immediat	te	
	31	26	25	21	20	16	15				0
\mathbf{J}	Г	opcode					addre	SS			
	31	26	25								0

© 2014 by Elsevier, Inc. All rights reserved. From Patterson and Hennessy, Computer Organization and Design, 5th ed.

ARITHMETIC CORE INSTRUCTION SET OPCODE / FMT /FT FOR-/ FUNCT NAME, MNEMONIC **OPERATION** (Hex) MAT FI if(FPcond)PC=PC+4+BranchAddr (4) 11/8/1/--Branch On FP True bolt FI if(!FPcond)PC=PC+4+BranchAddr(4) 11/8/0/--Branch On FP False bc1f div Lo=R[rs]/R[rt]; Hi=R[rs]%R[rt] 0/--/-1a (6) 0/--/--/1b divu R Lo=R[rs]/R[rt]; Hi=R[rs]%R[rt] Divide Unsigned 11/10/--/0 FP Add Single add.s FR F[fd] = F[fs] + F[ft]FP Add ${F[fd],F[fd+1]} = {F[fs],F[fs+1]} +$ add.d FR 11/11/--/0 {F[ft],F[ft+1]} Double 11/10/--/y FP Compare Single c.x.s* FR FPcond = (F[fs] op F[ft])? 1:0 FP Compare $FPcond = (\{F[fs], F[fs+1]\} op$ c.x.d* FR 11/11/--/y {F[ft],F[ft+1]})?1:0 Double * (x is eq, lt, or le) (op is ==, <, or <=) (y is 32, 3c, or 3e)div.s FR F[fd] = F[fs] / F[ft]11/10/--/3 FP Divide Single ${F[fd],F[fd+1]} = {F[fs],F[fs+1]} /$ FP Divide div.d FR 11/11/--/3 Double ${F[ft],F[ft+1]}$ FP Multiply Single mul.s FR F[fd] = F[fs] * F[ft] 11/10/--/2 FP Multiply ${F[fd],F[fd+1]} = {F[fs],F[fs+1]} *$ mul.d FR 11/11/--/2 Double ${F[ft],F[ft+1]}$ 11/10/--/1 FP Subtract Single sub.s FR F[fd]=F[fs] - F[ft] $\{F[fd],\!F[fd\!+\!1]\} = \{F[fs],\!F[fs\!+\!1]\} \text{ --}$ FP Subtract sub.d FR 11/11/--/1 {F[ft],F[ft+1]} Double (2) 31/--/--Load FP Single F[rt]=M[R[rs]+SignExtImm] Load FP F[rt]=M[R[rs]+SignExtImm]; (2) 35/--/--F[rt+1]=M[R[rs]+SignExtImm+4] Double 0 /--/--/10 R Move From Hi mfhi R[rd] = HiMove From Lo mflo R[rd] = Lo0 /--/--/12 R[rd] = CR[rs]10 /0/--/0 Move From Control mfc0 R $\{Hi,Lo\} = R[rs] * R[rt]$ 0/--/--/18Multiply mult R Multiply Unsigned ${Hi,Lo} = R[rs] * R[rt]$ 0/--/--/19 multu 0/--/--/3 R[rd] = R[rt] >> shamtShift Right Arith. sra R Store FP Single swc1 M[R[rs]+SignExtImm] = F[rt](2) 39/--/--M[R[rs]+SignExtImm] = F[rt];(2) 3d/--/--Store FP M[R[rs]+SignExtImm+4] = F[rt+1]

FLOATING-POINT INSTRUCTION FORMATS

FR		pcode		fmt			ft			fs		fd	funct	
	31	26	25		21	20		16	15	11	10	6	5	0
FI		pcode		fmt			ft				ir	nmediate	•	
	31	26	25		21	20		16	15					0

PSEUDOINSTRUCTION SET

Double

NAME	MNEMONIC	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 Equal	bge	$if(R[rs] \ge R[rt]) PC = Label$
Load Immediate	li	R[rd] = immediate
Move	move	R[rd] = R[rs]

REGISTER NAME, NUMBER, USE, CALL CONVENTION

NAME	NUMBER	USE	PRESERVEDACROSS
NAME	NUMBER	USE	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

	OPCOD	ES. BASI	E CONVER	SIC	ON. A	SCIL	SYMB	OLS		9	
					J.11, 7					Hexa-	ASCII
		. ,		Bi	narv	Deci-					
(I)					iidi j	mal			mal		
Sub.f 00 0001				00	0000	0			64		
Set	(1)	011									
Sale	i	srl									
Deep											
Deb Deb											
Diez Srav negf 00 0110 6	_										
bgtz		srlv									
addiu jar											
Stiti						8				48	
Sitiu movz		-					9			49	
Stitu	slti			00	1010	10	a	LF	74	4a	J
ori xori break xori trunc.wf 00 1101 13 d CR 77 4d M uni sync floor.wf 00 1111 15 f SI 78 4e N unfhi 01 0000 16 10 DLE 80 50 P unfhi 01 0001 18 12 DC2 82 52 R unifo movn.f 01 0010 18 12 DC2 82 52 R unifo movn.f 01 0010 20 14 DC4 84 54 T unifo 01 0101 21 15 NAK 85 55 U unifo 01 1010 22 16 SYN 86 56 V unifo 01 1010 22 16 SYN 86 56 V unifo 01 1010 22 18 EM 89 59 Y div <td>100000000000000000000000000000000000000</td> <td></td> <td></td> <td></td> <td></td> <td>11</td> <td>b</td> <td>VT</td> <td>75</td> <td>4b</td> <td>K</td>	100000000000000000000000000000000000000					11	b	VT	75	4b	K
ori break xori trunc.wf 00 11101 13 d CR 77 4d M unthi sync floor.wf 00 1111 15 f SI 79 4f O mfhi 01 0000 16 10 DLE 80 50 P mflo movo.f 01 0010 17 11 DCI 81 51 P mflo movo.f 01 0010 18 12 DC2 82 52 R mlo movo.f 01 0010 20 14 DC4 84 54 T 01 0101 21 15 NAK 85 55 U 01 0101 22 16 SYN 86 56 V 01 0110 22 16 SYN 86 56 V 01 1011 27 15 ESM 89 59 Y div 01 1010 26 18 CAN 88	andi	syscall	round.w.f	00	1100	12	С	FF	76	4c	L
Note Color Sync Sync Color Sync Sync	ori			00	1101	13	d	CR	77	4d	\mathbf{M}
(2) mthi mflo movz, f ol 0000 16 10 DLE 80 50 P ol 0000 17 11 DC1 81 51 Q mthi mflo movz, f ol 0010 17 11 DC1 81 51 Q ol 0001 17 11 DC1 81 51 Q ol 0010 18 12 DC2 82 52 R ol 01010 20 14 DC4 84 54 T ol 01010 12 15 NAK 85 55 U ol 01010 22 16 SYN 86 56 V ol 01011 23 17 ETB 87 57 W ol 01011 23 17 ETB 87 57 W ol 01010 24 18 CAN 88 58 X ol 01000 24 18 CAN 88 58 X ol 01010 25 19 EM 89 59 Y ol 01101 27 15 ESC 91 55 C ol 1100 28 1c FS 92 5c ol 1100 28 1c FS 92 5c ol 1110 29 1d GS 93 5d] ol 1110 30 1c RS 94 5c ol 1111 31 1f US 95 5f ol 1111 31 1f US 95 62 b lw subu 10 0010 33 21 ! 97 61 a lw subu 10 0010 33 21 ! 97 61 a lw subu 10 0010 33 21 ! 97 61 a lw subu 10 0010 33 21 ! 97 61 a lw subu 10 0010 34 22 " 98 62 b lw subu 10 0011 35 23 # 99 63 c lw subu 10 0010 34 22 " 98 62 b lw subu 10 0010 36 24 \$ 100 64 d lw or 10 0101 37 25 % 100 64 d lw or 10 0101 37 25 % 100 66 f lw sw subu 10 0010 34 22 " 98 62 b lw subu 10 1010 38 26 & 100 66 f lw sw subu 10 1010 38 26 & 100 66 f lw sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 42 2a * 106 6a j sw subu 10 1010 53 35 5 117 75 u w subu 10 1010 54 36 6 118 76 v c.uef, f 11 0100 52 34 4 116 74 t ldc1 c.uef, f 11 0100 53 35 5 117 75 u w subu 10 1010 54 36 6 118 76 v c.uef, f 11 0101 53 35 5 117 75 u w subu 10 111 55 37 7 119 77 w subu 10 111 55 37 7 119 77 w subu 10 111 110 61 30 30 50 51 123 76 b c.uef, f 11 1110 61 30 30 50 51 123 76 b c.uef, f 11 1110 61 30 30 51 123 76 b c.uef, f 11 1110 61 30 30 51 123 76 b c.uef, f 11 1111 61 30 36 7 127 76 DEL	xori			00	1110	14	e	SO		4e	N
(2) mthi mflo mflo mflo mflo mflo mflo mflo mflo	lui	sync		00	1111	15	f	SI	79	4f	O
(2)				01	0000	16	10	DLE	80	50	P
mflo mtlo movz.f movn.f 01 0010 11 19 13 DC3 82 52 R S R novn.f 01 0011 19 13 DC3 83 53 S T W L	(2)						11		81		Q
Mathematics	. /		movz.f								
01 0100 20									83		
Display											
Mult											
mult 01 0111 23 17 ETB 87 57 W multu 01 1000 24 18 CAN 88 58 X multu 01 1010 25 19 EM 89 59 Y divu 01 1010 26 1a SUB 90 5a Z 01 1010 28 1c FS 92 5c \ 01 1110 29 1d GS 93 5d] 1b add cvt.sf 10 0000 32 20 Space 96 60 \ 1h addu cvt.df 10 0001 33 21 ! 97 61 a 1w subu 10 0010 34 22 " 98 62 b 1w subu 10 0101 37 25 % 101 64 d 1bu and cvt.wf 10 0100											
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 1010 26 1a SUB 90 5a Z 01 1101 27 1b ESC 91 5b [01 1101 29 1d GS 93 5d] 1b add cvt.sf 10 0000 32 20 Space 96 60 * 1b add cvt.sf 10 0000 32 20 Space 96 60 * 1b add cvt.sf 10 0000 32 20 Space 96 60 * 1b add cvt.sf 10 0001 33 21 ! 97 61 a 1b sub											
Multu div Oli 1001 25 19 EM 89 59 Y Oli 1010 26 1a SUB 90 5a Z Oli 1010 27 1b ESC 91 5b [Oli 1100 28 1c FS 92 5c N Oli 1110 29 1d GS 93 5d] Oli 1110 30 1e RS 94 5e N Oli 1111 31 1f US 95 5f D Oli 1111 35 23 # 99 G3 C D Oli 1111 35 23 # 99 G3 C D Oli 111 35 23 # 99 G3 C D Oli 111 35 23 # 99 G3 C D Oli 111 35 23 # 99 G3 C D Oli 111 35 23 # 99 G3 C D Oli 111 35 23 # 99 G3 C D Oli 111 37 25 % Oli 101 65 e D Oli 101 37 25 % Oli 101 65 e D Oli 101 39 27 P Oli 36 F G D Oli 101 70 F Oli 101 70		mult		01	1000	24					X
div divu		multu		01	1001	25	19		89	59	Y
Diagram Diag				01	1010		1a	SUB	90	5a	Z
Di 1100									91		[
Display											
Display											
Description											^
The bound of the content of the co											
1h addu cvt.df 10 0001 33 21 ! 97 61 a 1w subu 10 0010 34 22 " 98 62 b 1bu and cvt.wf 10 0100 36 24 \$ 100 64 d 1bu and cvt.wf 10 0101 37 25 % 101 65 e 1w xor 10 0110 38 26 & 102 66 f sb 10 1000 40 28 (104 68 h sw1 slt 10 1000 40 28 (104 68 h sw1 slt 10 1010 42 2a * 105 69 i sw1 slt 10 1100 42 2a * 106 6a j sw1 slt 10 1101 43 2b + 107 6b k swr	1b	add	cvt.s.f								
lwl subu 10 0010 34 22 " 98 62 b lw subu 10 0010 34 22 " 99 63 c lbu and cvt.wf 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 sb 10 1001 40 28 (104 68 h sh 10 1001 41 29) 105 69 i swl slt 10 1010 42 2a * 106 6a j swl slt 10 1010 42 2a * 106 6a j swr slt 10 1101 45 2d - 109 6d m											a
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 sb 10 1000 40 28 (104 68 h sw slt 10 1001 41 29) 105 69 i sw slt 10 1011 43 2b + 106 6a j sw sltu 10 1011 43 2b + 107 6b k swr 10 1110 46 2e . 110 6c 1 swr 10 1111 47 2f / 111 6f o lwcl											
1bu								#			
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$			cvt.w.f								
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	100000000000000000000000000000000000000										
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$											
Sb								,			
sm 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 6e n 11 tge c.ff 11 0000 48 30 0 112 70 p lwc1 tgeu c.unf 11 0010 50 32 2 114 72 r p pref tltu c.ueqf 11 0010 52 34 4 116 74 t t <	sb							(
swl slt 10 1010 42 2a * 106 6a j sw sltu 10 1010 42 2a * 106 6a j lo 10101 43 2b + 107 6b k 10 11000 44 2c , 108 6c 1 10 1101 45 2d - 109 6d m cache 10 1110 46 2e . 110 6e n lwc1 tgeu c.sff 11 0000 48 30 0 112 70 p lwc2 tlt c.egf 11 0010 50 32 2 114 72 r pref tltu c.uegf 11 0010 50 32 2 114 72 r pref tltu c.ueff 11 0010 52 34 4 116 74 t 116 1											
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 . 1110 6e n cache 10 1111 47 2f / 111 6f o 11 tge c.ff 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 1114 72 r pref tltu c.ueqf 11 0010 53 33 3 115 73 s teq c.oltf 11 0100 52 34 4 116 74 t ldc1 c.ultf 11 0110 53 35 5 117 75 u ldc2 tne c.olef 11 0110 54 36 6 118 76 v c.ulef 11 0100 55 38 8 120 78 x swc1 c.seff 11 1000 56 38 8 120 78 x swc2 c.seff 11 1000 57 39 9 121 79 y swc2 c.ngl.f 11 1010 58 3a : 122 7a z c.ngl.f 11 1010 58 3a : 122 7a z c.ngl.f 11 110 62 3e > 125 7d } sdc1 c.ngl.f 11 1100 60 3c < 124 7c	200000	slt									
10 1100								+			
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$											
swr lo 1110 46 2e . 110 6e n cache 10 1111 47 2f / 111 6f o lwc1 tge c.ff 11 0000 48 30 0 112 70 p lwc1 tgeu c.unf 11 0001 49 31 1 113 71 q pref tltu c.ueqf 11 0010 50 32 2 114 72 r pref tltu c.ueqf 11 0010 52 34 4 116 74 t ldc1 c.ultf 11 0100 52 34 4 116 74 t ldc2 tne c.olef 11 0110 53 35 5 117 75 u ldc2 tne c.olef 11 0111 55 37 7 119 77 w sc c.sff 11 0000											
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$	swr								1000		
$\begin{array}{cccccccccccccccccccccccccccccccccccc$											
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$		tae	c.f.f								
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$		_									
pref tltu c.ueqf 11 0011 51 33 3 115 73 s teq c.oltf 11 0100 52 34 4 116 74 t ldc1 c.ultf 11 0101 53 35 5 117 75 u ldc2 tne c.olef 11 0110 54 36 6 118 76 v c.ulef 11 0111 55 37 7 119 77 w sc c.sff 11 1000 56 38 8 120 78 x swc1 c.nglef 11 1001 57 39 9 121 79 y swc2 c.seqf 11 1001 58 3a : 122 7a z c.ngl-f 11 1011 59 3b ; 123 7b { sdc1 c.ngef 11 1100 60 3c 124 7c	72/2000/00/2000										
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 0110 55 37 7 119 77 w sc c.sf.f 11 1000 56 38 8 120 78 x swc1 c.ngl.ef 11 1001 57 39 9 121 79 y swc2 c.seq.f 11 1001 58 3a : 122 7a z c.ngl.ef 11 101 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.ngt.f 11 1110 62 3e > 126 7e ~ c.ngt.f 11 1111 63 3f ? 127 7f DEL	10.000										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	PICI										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	1dc1	204									
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		tne									
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	1402	2110									
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	SC										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$											
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$											
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	SWCZ							:			
sdc1 c.ngef 11 1101 61 3d = 125 7d } sdc2 c.lef 11 1110 62 3e > 126 7e ~ c.ngtf 11 1111 63 3f ? 127 7f DEL								,			1
sdc2 c.lef 11 1110 62 3e > 126 7e ~ c.ngtf 11 1111 63 3f ? 127 7f DEL	ada1										1
c.ngt.f 11 1111 63 3f ? 127 7f DEL											
	sacz										
	(1)	1-(21-20)		11	1111	0.5	31		127	/1	DEL

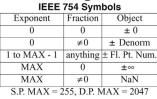
if $fmt(25:21) == 17_{ten} (11_{hex}) f = d (double)$

IEEE 754 FLOATING-POINT STANDARD

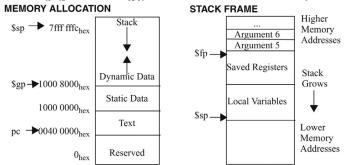
(3)

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

IEEE Single Precision and Double Precision Formats:



		ormats.			
S	Expo	onent	Fraction		
31	30	23 22			0
S	Ex	ponent	Fraction	75	
63	62	52 51			0



DATA ALIGNMENT

	Double Word											
	Wo	rd		Word								
Halfv	vord	Half	word	Hal	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

LF HON CONTR	OL HEGISTERS.	CHUSE	AND STAT	03	
В	Interru	ıpt	Exce	eption	
D	Masl	c	C	ode	
31	15	8	6	2	
	Pendii	ıg	U	J E	I
	Interru	ipt	N	M L	E
	15	0		1 1	0

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

 (OL: 11)	J.1 00	DEC					
Number	Name	Cause of Exception	Number	Name	Cause of Exception		
0	Int	Interrupt (hardware)	9	Bp	Breakpoint Exception		
4	AdEL	Address Error Exception		RI	Reserved Instruction		
7	Auel	(load or instruction fetch)		KI	Exception		
5	AdES	Address Error Exception	11	CpU	Coprocessor		
5	AuLS	(store)	11	СрО	Unimplemented		
6	IDE	IRE	IBE	Bus Error on	12	Ov	Arithmetic Overflow
0	IDL	Instruction Fetch	12	Ov	Exception		
7	DBE	Bus Error on	13	Tr	Trap		
	DDL	Load or Store	13	11			
- Q	Sve	Syccall Exception	15	EDE	Floating Point Exception		

SIZE PREFIXES

	PREFIX	SYMBOL	SIZE	PREFIX	SYMBOL	SIZE	PREFIX	SYMBOL	SIZE	PREFIX	SYMBO
10 ³	Kilo-	К	210	Kibi-	Ki	1015	Peta-	Р	250	Pebi-	Pi
106	Mega-	М	220	Mebi-	Mi	1018	Exa-	Е	260	Exbi-	Ei
109	Giga-	G	230	Gibi-	Gi	1021	Zetta-	Z	270	Zebi-	Zi
1012	Tera-	т	240	Tebi-	Ti	1024	Yotta-	Y	280	Yobi-	Yi