Subtract Unsigned

MIPS Reference Data

CORE INSTRUCTION SET OPCODE											
FOR- / FUNCT											
NAME, MNEMO		MAT	- ((Hex)						
Add	add	R	R[rd] = R[rs] + R[rt]		0 / 20 _{hex}						
Add Immediate	addi	Ι	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 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	$R[rt]=\{24\text{'b0,M}[R[rs]\\+SignExtImm](7:0)\}$	(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	$R[rt] = R[rs] \mid ZeroExtImm$	(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.	sltu	R	R[rd] = (R[rs] < R[rt]) ? 1 : 0	(6)	0 / 2b _{hex}						
Shift Left Logical	sll	R	$R[rd] = R[rt] \le shamt$		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	$\begin{aligned} M[R[rs]+SignExtImm] &= R[rt]; \\ R[rt] &= (atomic) ? 1 : 0 \end{aligned}$	(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}						

(1)	May cause overfl	ow exception	
(2)	SignExtImm = {	16{immediate[15]}, immediate	1

- (3) $ZeroExtImm = \{ 16\{1b'0\}, immediate \}$
- (4) 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 > 1 VIEW 11 CHEM 4VIEW AND

 $R \quad R[rd] = R[rs] - R[rt]$

BAS	SIC II	NSTRU	JCTIO	N FOI	RMA	TS	7 갓 V	2	ち	見思	i34	48(54)	5 6001	16/	notaled a
	R	opc	ode	rs			rt			rd		shamt	funct	1001	adel
KH	1	31	26	25	21	20		16	15	11	10	6	5	0	302
	I	opc	ode	rs			rt				iı	mmediate	•		아 .
		31	26	25	21	20		16	15					0	by ofthe
	J	opc	ode						á	nddress					

DX240 memory

ARITHMETIC CORE INSTRUCTION SET

			FMT/FT
F	OR-	-	/ FUNCT
	ИAТ	OPERATION	(Hex)
Branch On FP True bclt	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 add.d	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} +$	11/11//0
Double add.d	I IX	$\{F[ft],F[ft+1]\}$	11/11//0
FP Compare Single c.x.s*	FR	FPcond = (F[fs] op F[ft]) ? 1 : 0	11/10//y
FP Compare	FR	$FPcond = ({F[fs],F[fs+1]}) op$	11/11//v
Double		$\{F[ft],F[ft+1]\}\)?1:0$	11/11//y
		==, <, or <=) (y is 32, 3c, or 3e)	
	FR	F[fd] = F[fs] / F[ft]	11/10//3
FP Divide	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} /$	11/11//3
Double		$\{F[ft],F[ft+1]\}$	
1 7 0	FR	F[fd] = F[fs] * F[ft]	11/10//2
FP Multiply mul.d	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} *$	11/11//2
Double		$\{F[ft],F[ft+1]\}$	
U	FR	F[fd]=F[fs] - F[ft]	11/10//1
FP Subtract	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} -$	11/11//1
Double		$\{F[ft],F[ft+1]\}$	
Load FP Single lwc1	I		31//
Load FP	Ι	$F[rt]=M[R[rs]+SignExtImm]; \qquad (2)$	35//
Double		F[rt+1]=M[R[rs]+SignExtImm+4]	
Move From Hi mfhi	R	[]	0 ///10
Move From Lo mflo	R	F -3 -	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)	0///19
Shift Right Arith. sra	R	R[rd] = R[rt] >>> shamt	0//-3
Store FP Single swc1	I	M[R[rs]+SignExtImm] = F[rt] (2)	39//
Store FP sdc1	ī	M[R[rs]+SignExtImm] = F[rt]; (2)	3d//
Double	1	M[R[rs]+SignExtImm+4] = F[rt+1]	Ju//

OPCODE

FLOATING-POINT INSTRUCTION FORMATS

FR	Ē.	code	fmt		ft			fs	fd	funct	
	31	26	25	21	20	16	15	11	10 6	5	0
FI	op	code	fmt		ft				immediate	e	٦
	31	26	25	21	20	16	15				0

PSEUDOINSTRUCTION SET

NAME	MNEMONIC	OPERATION
Branch Less Than	blt	$if(R[rs] \le R[rt]) PC = Label$
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 A CALL?
i	\$zero	0	The Constant Value 0	N.A.
ĺ	\$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
K	\$sp	29	Stack Pointer	Yes
	\$fp	30	Frame Pointer	Yes
	\$ra	31	Return Address	No

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

0 / 23_{hex}

OPCODES,	BASE	CONVERS	ION, ASCII	SYMBOLS

OPCOD	ES. BASI	CONVER	SION. A	SCIL	SYMB	OLS		9	
	(1) MIPS	(2) MIPS				ASCII	L .	Hexa-	ASCII
opcode	funct	funct	Binary	Deci-	deci-	Char-	Deci-	deci-	Char-
(31:26)	(5:0)	(5:0)	Dinary	mal	mal	acter	mal	mal	acter
(1)	sll	add.f	00 0000	0	0	NUL	64	40	(a)
(1)	211	sub.f	00 0000	1	1	SOH	65	41	A
j	srl	mul.f	00 0001	2	2	STX	66	42	В
jal	sra	div.f	00 0010	3	3	ETX	67	43	C
beq	sllv	sgrt.f	00 0011	4	4	EOT	68	44	D
1 -	SIIV	1 2	00 0100	5	5	ENQ	69	45	E
bne		abs f	l	6			70	46	F
blez	srlv	mov.f	00 0110	7	6 7	ACK BEL	71	47	G
bgtz	srav	neg.f	00 1000	8	8		72	48	H
addi	jr		l	9	9	BS HT	73	49	П I
addiu	jalr		00 1001	10			74		J
slti	movz		00 1010		a	LF		4a	
sltiu	movn		00 1011	11	b	VT	75	4b	K
andi	syscall	round.w.f	00 1100	12	c	FF	76	4c	L
ori	break	trunc.w.f	00 1101	13	d	CR	77	4d	M
xori		ceil.w.f	00 1110	14	e	SO	78	4e	N
lui	sync	floor.w.f	00 1111	15	f	SI	79	4f	O
(2)	mfhi		01 0000	16	10	DLE	80	50	P
(2)	mthi		01 0001	17	11	DC1	81	51	Q
	mflo	movz.f	01 0010	18	12	DC2	82	52	R
	mtlo	movn. f	01 0011	19	13	DC3	83	53	S
			01 0100	20	14	DC4	84	54	T
			01 0101	21	15	NAK	85	55	U
			01 0110	22	16	SYN	86	56	V
			01 0111	23	17	ETB	87	57	W
	mult		01 1000	24	18	CAN	88	58	X
	multu		01 1001	25	19	EM	89	59	Y
	div		01 1010	26	1a	SUB	90	5a	Z
	divu		01 1011	27	1b	ESC	91	5b	[
			01 1100	28	1c	FS	92	5c	/
			01 1101	29	1d	GS	93	5d]
			01 1110	30	1e	RS	94	5e	^
			01 1111	31	1f	US	95	5f	_
lb	add	cvt.s.f	10 0000	32	20	Space	96	60	-
lh	addu	$\mathtt{cvt.d}.f$	10 0001	33	21	!	97	61	a
lwl	sub		10 0010	34	22	"	98	62	b
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 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 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	0
11	tge	c.f. <i>f</i>	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 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.ngle f	11 1001	57	39	9	121	79	У
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	~
1				(2		• • • • • • • • • • • • • • • • • • • •	1 1 2 7		DIT

(1) opcode(31:26) == 0(2) opcode(31:26) == 17_{ten} (11_{hex}); if fmt(25:21)== 16_{ten} (10_{hex}) f = s (single);

11 1111

63

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

c.ngt.f

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:

IEEE 754 Symbols

Exponent	Fraction	Object
0	0	± 0
0	≠0	± Denorm
1 to MAX - 1	anything	± Fl. Pt. Num.
MAX	0	±∞
MAX	≠0	NaN

S.P. MAX = 255, D.P. MAX = 2047

S Exponent Fraction 31 30 S Exponent Fraction

MEMORY ALLOCAT	ION	STACK FRAME	Ü
\$sp → 7fff fffc _{hex}	Stack	Argumer Argumer	Addresses
\$gp →1000 8000 _{hex}	Dynamic Data	Saved Reg	Stack Grows
1000 0000 _{hex}	Static Data	Local Vari	ables
pc →0040 0000 _{hex}	Text	\$sp	Lower
$0_{ m hex}$	Reserved		Memory Addresses

DATA ALIGNMENT

	Double Word								
	Word				Word				
Halfv	Halfword		Halfword		Halfword		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	EI
	Interrupt		M	L E
	15	8	4	1 0

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

EXCEPTION CODES

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

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

TIET IXEO (10 101 Blok, Communication, 2 101 Mcmory)										
	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.

127

DEL