

ARITHMETIC CORE INSTRUCTION SET

			55 (2)	OLCODE
				/ FMT /FT
		FOR-	-	/ FUNCT
NAME, MNEMO	NIC	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//v
FP Compare			$FPcond = (\{F[fs], F[fs+1]\} op$	
Double	c.x.d*	FR	{F[ft],F[ft+1]})?1:0	11/11//y
* (x is eq. lt, or	rle) (op is	==, <, or <=) (y is 32, 3c, or 3e)	
			F[fd] = F[fs] / F[ft]	11/10//3
ED Divide	div.d		${F[fd],F[fd+1]} = {F[fs],F[fs+1]} /$	11/11//3
Double	aiv.a	ГК	{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	mul.d	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} *$	11/11//2
Double	muı.a	I'IX	{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	sub.d	FR	${F[fd],F[fd+1]} = {F[fs],F[fs+1]} -$	11/11//1
Double	sub.u	I'IX	$\{F[ft],F[ft+1]\}$	11/11//1
Load FP Single	lwc1	I	F[rt]=M[R[rs]+SignExtImm] (2)	31//
Load FP	ldc1	I	F[rt]=M[R[rs]+SignExtImm]; (2)	35//
Double	Idel		F[rt+1]=M[R[rs]+SignExtImm+4]	
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)	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	SuCI	1	M[R[rs]+SignExtImm+4] = F[rt+1]	Ju//

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	e
	31 26	25 21	20 16	15		0

PSEUDOINSTRUCTION SET

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 Equa	l bge	$if(R[rs] \ge R[rt]) PC = Label$
Load Immediate	1i	R[rd] = immediate
Move	move	R[rd] = R[rs]

REGISTER NAME, NUMBER, USE, CALL CONVENTION

NUMBER USE		PRESERVEDACROSS A CALL?
0	The Constant Value 0	N.A.
1	Assembler Temporary	No
2-3	Values for Function Results and Expression Evaluation	No
4-7	Arguments	No
8-15	Temporaries	No
16-23	Saved Temporaries	Yes
24-25	Temporaries	No
26-27	Reserved for OS Kernel	No
28	Global Pointer	Yes
29	Stack Pointer	Yes
30	Frame Pointer	Yes
31	Return Address	Yes
	0 1 2-3 4-7 8-15 16-23 24-25 26-27 28 29 30	0 The Constant Value 0 1 Assembler Temporary 2-3 Values for Function Results and Expression Evaluation 4-7 Arguments 8-15 Temporaries 16-23 Saved Temporaries 24-25 Temporaries 26-27 Reserved for OS Kernel 28 Global Pointer 29 Stack Pointer 30 Frame Pointer

MIPS Reference Data

	110		ence Butu	4	
CORE INSTRUCTI	ON SE	T			OPCODE
		FOR-			/ FUNCT
NAME, MNEMO		MAT	- ((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	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+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)	$30_{ m hex}$
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	$R[rd] = R[rs] \mid R[rt]$		$0/25_{ m hex}$
Or Immediate	ori	I	$R[rt] = R[rs] \mid ZeroExtImm$	(3)	d_{hex}
Set Less Than	slt	R	$R[rd] = (R[rs] \le R[rt]) ? 1 : 0$		$0/2a_{hex}$
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	$R[rd] = (R[rs] \le R[rt]) ? 1 : 0$	(6)	$0 \: / \: 2b_{hex}$
Shift Left Logical	sll	R	$R[rd] = R[rt] \le shamt$		$0 / 00_{ m hex}$
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	$\begin{aligned} M[R[rs] + SignExtImm] &= R[rt]; \\ R[rt] &= (atomic) ? 1 : 0 \end{aligned}$	(2,7)	$38_{ m hex}$
Store Halfword	sh	I	M[R[rs]+SignExtImm](15:0) = R[rt](15:0)	(2)	$29_{ m hex}$
Store Word	SW	I	M[R[rs]+SignExtImm] = R[rt]	(2)	$2b_{hex}$
Subtract	sub	R	R[rd] = R[rs] - R[rt]	(1)	$0/22_{ m hex}$
Subtract Unsigned	subu	R	R[rd] = R[rs] - R[rt]		$0/23_{ m hex}$
	(2) Sig	nExtI	se overflow exception mm = { 16{immediate[15]}, immediate 16{11b20}, immediate	ediate	}

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

(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

OPCODES, BASE CONVERSION, ASCII SYMBOLS

MIPS (1) MIPS (2) MIPS (5:0) (6:0) (6:0) (6:0) (6:0) (7:0)			(2) MIPS						Heva	ASCII
(3i:26) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (5:0) (7:				D:	Deci-			Deci-		
(31:26) (3:09) (3:09) (3:00)				Binary	mal			mal		
Subf 00 0001				00.000						
jal sra mulf 00 0010 2 2 STX 66 42 B beq silv sqrtf 00 0010 3 3 ETX 67 43 C beq silv sqrtf 00 0010 4 4 EOT 68 44 D bee silv movf 00 0010 5 ENQ 69 45 E betz srav negf 00 0110 6 6 ACK 70 46 F bgtz srav negf 00 0110 6 6 ACK 70 46 F bgtz srav negf 00 0110 8 8 BS 72 48 H addid jalr 00 1001 9 9 HT 73 49 I sltii movz 00 1010 10 a LF 74 4a J sltii movz 00 1010 10 a LF 74 4a J sltii movz 00 1010 11 b VT 75 4b K srav 00 1011 11 b VT 75 4b K srav 00 1011 11 b VT 75 4b K srav 00 1011 13 d CR 77 4d M srav 00 1011 13 d CR 77 4d M srav 00 1011 13 d CR 77 4d M srav 00 1011 15 f SI 79 4f O mrhi 01 1000 16 10 DLE 80 50 P mrho movsf 01 0010 17 11 DC1 81 51 Q mrho movsf 01 0010 17 11 DC1 81 51 Q mrho movsf 01 0010 12 15 NAK 85 55 U 01 0101 22 16 SYN 86 56 V divu 01 1010 24 18 CAN 88 S8 X mult 01 1000 24 18 CAN 88 58 X mult 01 1000 25 19 EM 89 59 Y divu 01 1010 26 1a SUB 90 5a Z divu 01 1010 27 1b ESC 91 5b [lb add cvt.sf 10 0000 32 20 Space 96 60 V srav 10 0010 34 22 9 96 3c C lb add cvt.sf 10 0001 33 21 9 96 3c C srav 10 0010 34 22 9 96 3c C lb add cvt.sf 10 0000 32 20 Space 96 60 V srav 10 0010 37 25 9 101 66 6 6 lw subu 10 0010 37 25 9 10 66 6 lw subu 10 0010 33 21 9 9 63 c lo 1100 40 28	(1)	sll								
jal										
Dee										
Dec										
Diez Srav movf		sllv								
bgtz										
addi										
addiu jalr			neg.f							
Sitiu movx 00 1010 10 a LF 74 4a J										
Sitiu movn										
andi										
ori xori break xori trunc.xf 00 1101 13 d CR 77 4d M usyrc floor.xf 00 1110 14 e SO 78 4e N mfhi mfhi 01 0000 16 10 DLE 80 50 P mthi movxf 01 0010 17 11 DCL 83 50 P mthi movxf 01 0010 18 12 DC2 82 52 R mtho movxf 01 0010 20 14 DC4 84 54 T 01 0101 21 15 NAK 85 55 U 10 1011 22 18 CAN 88 58 X mult 01 1010 24 18 CAN 88 58 X div 01 1010 26 1a SUB 90 5a Z div 01 1010 26 1a<										
Note			round.w.f							
1ui		break								
(2) mthi movzf mtho movzf 01 0000 16 10 DLE 80 50 P mtho movzf 01 0010 17 11 DC1 81 51 Q mtho movnf 01 0011 19 13 DC3 83 53 S										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	lui		floor.w.f							
mflo movsf 01 0010										
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	(2)									
mtlo			movz.f							
		mtlo	movn.f				DC3			
							DC4			
mult										
mult										
multu div 01 1001 25 19 EM 25 19 SUB 90 5a Z 2 10 1010 26 1a SUB 90 5a Z 2 10 1010 26 1a SUB 90 5a Z 2 10 1010 27 1b ESC 91 5b [01 1101 29 1d GS 93 5d] 01 1110 30 1e RS 94 5e ^ 101 1110 30 1e RS 94 5e ^ 101 1110 30 1e RS 94 5e ^ 101 1110 31 1f US 95 5f _ 1 10 1111 31 1f US 95 5f _ 1 10 1111 31 1f US 95 5f _ 1 10 1111 31 1f US 95 5f _ 1 10 1111 31 1f US 95 5f _ 1 10 1111 31 1f US 95 5f _ 1 10 1111 31 1f US 95 5f _ 1 10 1111 31 1f US 95 5f _ 1 10 1111 31 1f US 95 5f _ 1 10 1011 35 23 # 99 63 60 5 10 10 101 34 22 " 98 62 b 10 10 101 35 23 # 99 63 c 10 10 101 35 23 # 99 63 c 10 10 10 10 10 36 24 \$ 100 64 d 10 10 10 10 36 24 \$ 100 64 d 10 10 10 10 37 25 % 101 65 e 10 10 10 10 38 26 & 102 66 f f 10 10 10 10 39 27 ' 103 67 g 10 10 10 10 10 10 10 10 10 10 10 10 10										
div 01 1010 26 1a SUB 90 5a Z		mult		01 1000		18	CAN	88		
divu		multu		01 1001	25	19			59	
01 1100		div		01 1010	26	1a	SUB	90	5a	Z
01 1100		divu		01 1011	27	1b	ESC	91	5b	ſ
1110 30 1e RS 94 5e ^ 1111 31 1f US 95 5f _ 1111 31 1f US 35 23 # 99 63 c 1111 37 25 % 101 65 e 1111 37 25 % 101 65 e 1111 38 26 & 102 66 f g g 111 38 27 103 67 g g 111 35 35 7 103 67 g 111 31 1f 1f 1f 1f 1f 1f				01 1100	28	1c	FS	92	5c	Ţ
1110 30 1e RS 94 5e ^ 1111 31 1f US 95 5f _ 1111 31 1f US 35 23 # 99 63 c 1111 37 25 % 101 65 e 1111 37 25 % 101 65 e 1111 38 26 & 102 66 f g g 111 38 27 103 67 g g 111 35 35 7 103 67 g 111 31 1f 1f 1f 1f 1f 1f				01 1101	29	1d	GS	93		1
10				01 1110	30	1e	RS	94	5e	^
10 10 10 10 10 10 10 10				01 1111	31	1f	US	95	5f	
10	lb	add	cvt.s.f							-
1	1h									a
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	lwl				34	22	"	98		b
Du							#			
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	lbu	and	cvt.w.f		36	24	\$	100		
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$										
Note		xor								
Second S							,			
swl 10 1001 41 29 105 69 i swl slt 10 1010 42 2a * 106 6a j sw sltu 10 1010 42 2a * 106 6a j sw 10 1101 43 2b + 107 6b k swr 10 1101 44 2c . 108 6c 1 10 1101 45 2d - 109 6d m cache 10 1110 46 2e . 110 6e n 11 tge c.ff 110000 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 tdc1 c.untf 11 01010 52 34 4	sb						(h
swl slt 10 1010 42 2a * 106 6a j sw sltu 10 1011 43 2b + 107 6b k 10 1010 44 2c , 108 6c I swr 10 1101 45 2d - 109 6d m cache 10 1110 46 2e . 110 6e n 11 tge c.ff 11 0000 48 30 0 112 70 p 1wc1 tgeu c.unf 11 0010 50 32 2 114 72 r pref tltu c.uegf 11 0010 50 32 2 114 72 r tdc1 c.ultf 11 0100 52 34 4 116 74 t 1dc2 tne c.olef 11 0110 53 35 5 <										
Second S		slt					*			
10 1100 44 2c 108 6c 1 10 101 45 2d 109 6d m										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	- "									
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$,			-
cache 101111 47 2f / 111 6f o 11 tge c.ff 11 0000 48 30 0 112 70 p 1wc1 tgeu c.unf 11 0001 49 31 1 113 71 q 1wc2 tlt c.eqf 11 0010 50 32 2 2 114 72 r 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 0110 53 35 5 117 75 u ldc2 tee c.olef 11 0110 54 36 6 118 76 v sc c.sef 11 1000 56 38 8 120 78 x swc1 c.nglef 11 1001 57 39 9 121 79 y swc2 c.seqf 11 1010 58 3a : 122 7a z c.ngl-f 11 1101 59 3b ; 123 7b { sdc1 c.ngl-f 11 1100 60 3c c.ngl-f 11 1101 61 3d = 125 7d } sdc2 c.lef 11 1110 62 3e > 126 7e ~ c.ngt-f 11 1110 62 3e > 127 7f DEL	swr									
$ \begin{array}{cccccccccccccccccccccccccccccccccccc$										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$		tre	c ff							
lwc2 tlt c.eqf 11 0010 50 32 2 114 72 r pref tltu c.ueqf 11 0010 51 33 3 115 73 s teq c.oltf 11 0010 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 sc c.sff 11 1001 57 37 7 719 77 w swc1 c.seqf 11 1001 57 39 9 121 79 y swc2 c.seqf 11 1010 58 3a : 122 7a z c.ngtf 11 1000 50 36 : 123 7b { sdc1 c.ngtf 11 100 60 3c : 124 7c <td></td>										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$										
teq c.oltf 11 0100 52 34 4 116 74 t										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	brer									
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	1 de 1	ceq								
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$		tne								
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	±ucz	rite								
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	0.0		c.uie.j							
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$										
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	swc2						:			
$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$;		7/b	
$\begin{array}{c ccccccccccccccccccccccccccccccccccc$										Į
c.ngt f 11 1111 63 3f ? 127 7f DEL (1) opcode(31:26) == 0										
(1) $\operatorname{opcode}(31:26) == 0$	sdc2									
				11 1111	63	3f	?	127	7f	DEL

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

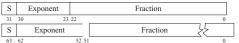
if $fmt(25:21) = 17_{ten} (11_{hex}) f = d (double)$

IEEE 754 FLOATING-POINT STANDARD

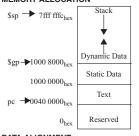
 $(-1)^S \times (1 + Fraction) \times 2^{(Exponent - Bias)}$ where Single Precision Bias = 127, Double Precision Bias = 1023.

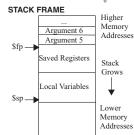
ILLL	7 JT Oyili	
Exponent	Fraction	Object
0	0	± 0
0	≠0	± Denorm
1 to MAX - 1	anything	± Fl. Pt. Num
MAX	0	±∞
MAX	≠0	NaN
CDMAV-2	55 D P N	4 A Y = 2047

IEEE Single Precision and **Double Precision Formats:**



MEMORY ALLOCATION





DATA ALIGNMENT

	Double Word									
Word Word										
Halfv	vord	Half	word	Hal	fword	Halfword				
Byte	Byte	Byte	Byte	Byte Byte		Byte	Byte			

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

EXCEPTION CONTROL REGISTERS: CALISE AND STATUS

FIION CONTROL REGISTERS. CAUSE AND STATUS									
Interrupt Exception									
Mask		Code							
15 8		6	2						
Pending		U	ΕI						
Interrupt		M	L E						
	Interrupt Mask 15 8 Pending	Interrupt Mask 15 8 Pending	Interrupt Exception Mask Code 15 8 6 Pending U						

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

EXCEPTION CODES

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

SIZE PREFIXES (10x for Disk, Communication; 2x 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.