32

V1.04

2025 2

Loongson Technology Corporation Limited

2

Building No.2, Loongson Industrial Park,

Zhongguancun Environmental Protection Park, Haidian District, Beijing

(Tel) 010-62546668

(Fax) 010-62600826

32

32

1.04

2020/09/17

0.90

0.91

1.00

1) 3.2.3
convertToIntegerExact… roundToIntegralExact

1.01 2) CSR

1) 2.2.3 2.2.5.1 3.2.5

1) Öù 0@ "D•s Öö°

1.02

	PC 2) 2.1.5 3) 2.2.1.10 MULH.WU U 4) 3.1.4.4 2Emin Emin 5) 3.2.4.6 MOVCF2FR 3.2.4.7 MOVCF2GR 0 0 0 6) 3.2.6.1 32 7) 4.2.3.3 4.2.3.4 TLB CSR.TLBIDX.NE 1 TLB TLB 8) 5.2.1 9) 5.4.3.4 “ INVTLB r0, r0 ” “ INVTLB 0, r0, r0 ” 10) 5.4.4 found_ppn 11) 6.2.2 12) 7.4.1 7-2 DATF DATM “ …… …… ” “ …… …… ” 13) 7.6.1 RDTIME RDCNTID 14) GRLEN 32
1.04	1) 2.2.1.6 32 2) 3 “ ” 3) 3.2.3.2 FFINT.{S/D}.L FTINT.L.{S/D}, 3.2.3.3 FTINT{RM/RP/RZ/RNE}.L.{S/D}

1

1.1

LoongArch Reduced Instruction Set Computing RISC

load/store load/store

32 64 LA32 LA64 LA64

LA32 “ ” LA32

LA64

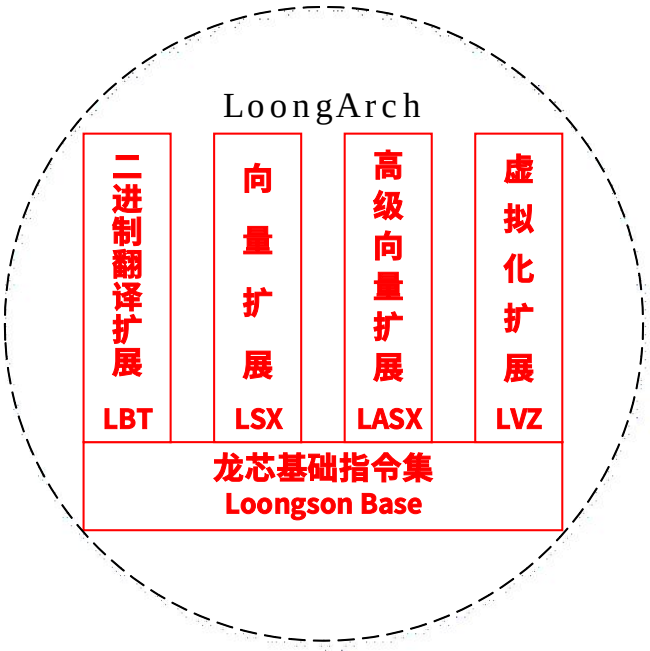
LA32 LA64

Loongson Base 1-1

Loongson Binary Translation LBT Loongson Virtualization LVZ

Loongson SIMD Extension LSX Loongson Advanced SIMD Extension

LASX



1-1

32 LA32

2 32

1.3

32

“ F ”

“ .XX ”

.B .H .W .BU .HU .WU

ADD.W ADD.D BLT[U] BLT BLTU

ADD[I].{W/D} ADD.W ADD.D ADDI.W ADDI.D

1.4.2

32 Control and Status Register CSR

CSR CSR.%%%%.####

%%%% #### CSR.CRMD.PLV CRMD

PLV

2

32

2.1

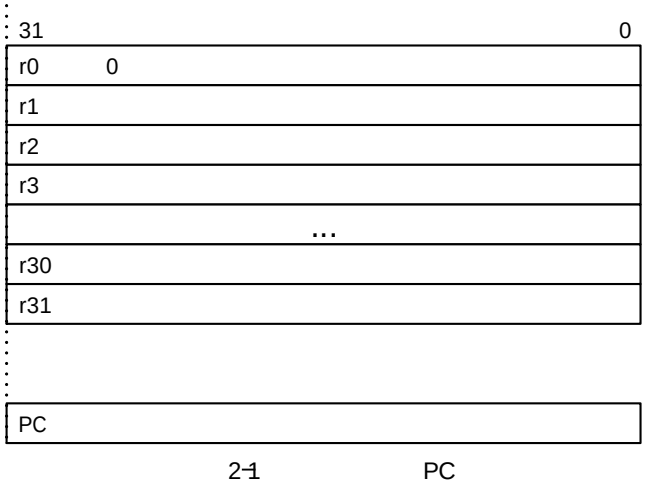
[1]

2.1.1

5 bit b Byte B 8b
Halfword H 16b Word W 32b LA32

2.1.2

General-purpose Register GR Program
Couner PC 2-1



2-1 PC

[1] Linux user mode

2.1.2.1

GR 32 r0~r31 0 r0 0 GR 32

32

GR BL 1 r1

Application Binary Interface ABI r1

2.1.2.2 PC

PC 1 PC

PC PC

GR

2.1.3

32 2 Privilege LeVel PLV PLV0 PLV3

PLV3 PLV0

4.1

2.1.4

Exception Interrupt /

“ / ” “ / ”

/

[1]

SYSCALL SYS

BREAK BRK

INE

IPE

4

ADEF

[1] 32

Cache Cache

Cache Cache Cache

IBAR store

Cache Cache code 8 9 CACOP

Hit Invalidate I-Cache Hit Invalidate and Writeback D-Cache

2.1.8

4 ADEF

[1] ALE

2.1.9

32 Weakly Consistency WC

1.

2.

3.

32 DBAR IBAR LL-SC

[1]

2 4 8

128 16 256 32

2.2

2.2.1

2.2.1.1 ADD.W, SUB.W

```
add.w    rd, rj, rk
sub.w    rd, rj, rk
```

ADD.W rj rk [31:0] rd

```
ADD.W:
        = G       + G
    G       =     31:0
```

SUB.W rj rk [31:0] rd

```
SUB.W:
        = G       - G
    G       =     31:0
```

2.2.1.2 ADDI.W

```
addi.w    rd, rj, si12
```

ADDI.W rj 12 si12 32 rd

```
ADDI.W:
        = G       +     E     (  12, 32)
    G       =     31:0
```

2.2.1.3 LU12I.W

```
lu12i.w    rd, si20
```

LU12I.W 20 si20 12 0 rd

```
LU12I.W:
    G      =    20, 12' 0
```

ORI 12

2.2.1.4 SLT[U]

slt rd, rj, rk

sltu rd, rj, rk

SLT rj rk

rd 1 0

```
SLT:
    G      = (      (G     ) <      (G     )) ? 1 : 0
```

SLTU rj rk

rd 1 0

```
SLTU:
    G      = (        (G     ) <        (G     )) ? 1 : 0
```

SLT SLTU

2.2.1.5 SLT[U]I

slti rd, rj, si12

sltui rd, rj, si12

SLTI rj 12 si12

rd 1 0

```
SLTI:
        =     E     (  12, 32)
    G      = (      (G     ) <      (   )) ? 1 : 0
```

SLTUI rj 12 si12

rd 1 0

```
SLTUI:
        =     E     (  12, 32)
    G      = (        (G     ) <        (   )) ? 1 : 0
```

SLTI SLTUI

SLTUI

2.2.1.6 PCADDU12I

pcaddu12i rd, si20

PCADDU12I 20 si20 12 0 PC

rd

PCADDU12I:

```
G = C + 20, 12' 0
```

2.2.1.7 AND, OR, NOR, XOR

and rd, rj, rk

or rd, rj, rk

nor rd, rj, rk

xor rd, rj, rk

AND rj rk

rd

AND:

```
G = G & G
```

OR rj rk rd

OR:

```
G = G G
```

NOR rj rk

rd

NOR:

```
G = (G G )
```

XOR rj rk

rd

XOR：

```
G       = G         G
```

2.2.1.8 ANDI, ORI, XORI

```
andi        rd, rj, ui12
ori         rd, rj, ui12
xori        rd, rj, ui12
```

ANDI rj 12

rd

ANDI：

```
G       = G       &     E     (  12, 32)
```

ORI rj 12

rd

ORI：

```
G       = G             E     (  12, 32)
```

XORI rj 12

rd

XORI：

```
G       = G             E     (  12, 32)
```

2.2.1.9 NOP

NOP “ andi r0, r0, 0” 4 PC 4

2.2.1.10 MUL.W, MULH.W[U]

```
mul.w       rd, rj, rk
mulh.w      rd, rj, rk
mulh.wu     rd, rj, rk
```

MUL.W rj rk [31:0]
rd

```
MUL.W:
        =    (G    ) *    (G    )
  G     =     31:0
```

MULH.W rj rk [63:32]
rd

```
MULH.W:
        =    (G    ) *    (G    )
  G     =     63:32
```

MULH.WU rj rk
[63:32] rd

```
MULH.WU:
        =     (G    ) *     (G    )
  G     =     63:32
```

2.2.1.11 DIV.W[U], MOD.W[U]

```
div.w    rd, rj, rk
mod.w    rd, rj, rk
div.wu   rd, rj, rk
mod.wu   rd, rj, rk
```

DIV.W DIV.WU rj rk rd

```
DIV.W:
        =    (G    ) /    (G    )
  G     =     31:0
DIV.WU:
        =     (G    ) /     (G    )
  G     =     31:0
```

MOD.W MOD.WU rj rk

rd

```
MOD.W:
        =     (G     ) %     (G     )
    G   =       31:0
MOD.WU:
        =     (G     ) %     (G     )
    G   =       31:0
```

DIV.W MOD.W DIV.WU MOD.WU

/ DIV.W/MOD.W DIV.WU/MOD.WU

0

2.2.2

2.2.2.1 SLL.W, SRL.W, SRA.W

```
sll.w   rd, rj, rk
srl.w   rd, rj, rk
sra.w   rd, rj, rk
```

SLL.W rj rd

```
SLL.W:
      = LL(G     , G     4:0 )
    G   =   31:0
```

SRL.W rj rd

```
SRL.W:
      =  L(G     , G     4:0 )
    G   =   31:0
```

SRA.W rj rd

```
SRA.W:
      =  A(G     , G     4:0 )
    G   =   31:0
```

rk [4:0]

2.2.2.2 SLLI.W, SRLI.W, SRAI.W

```
slli.w    rd, rj, ui5
srli.w    rd, rj, ui5
srai.w    rd, rj, ui5
```

SLLI.W rj rd

```
SLLI.W:
    = LL(G , 5)
G   = 31:0
```

SRLI.W rj rd

```
SRLI.W:
    = L(G , 5)
G   = 31:0
```

SRAI.W rj rd

```
SRAI.W:
    = A(G , 5)
G   = 31:0
```

5 ui5

2.2.3

2.2.3.1 BEQ, BNE, BLT[U], BGE[U]

```
beq     rj, rd, offs16
bne     rj, rd, offs16
blt     rj, rd, offs16
bge     rj, rd, offs16
bltu    rj, rd, offs16
bgeu    rj, rd, offs16
```

BEQ rj rd

BEQ:

```
G ==G :
C = C + E ( 16, 2' 0 , 32)
```

BNE rj rd

BNE:

```
G !=G :
C = C + E ( 16, 2' 0 , 32)
```

BLT rj rd

BLT:

```
(G ) < (G ) :
C = C + E ( 16, 2' 0 , 32)
```

BGE rj rd

BGE:

```
(G ) >= (G ) :
C = C + E ( 16, 2' 0 , 32)
```

BLTU rj rd

BLTU:

```
(G ) < (G ) :
C = C + E ( 16, 2' 0 , 32)
```

BGEU rj rd

BGEU:

```
(G ) >= (G ) :
C = C + E ( 16, 2' 0 , 32)
```

16 offs16 2

PC

offs16<<2

2.2.3.2 B

b offs26

B 26 offs26 2
PC

B:

```
C = C + E ( 26, 2' 0 , 32)
```

offs26<<2

2.2.3.3 BL

bl offs26

BL PC 4 1 r1
26 offs26 2
PC

BL:

```
G 1 = C + 4
C = C + E ( 26, 2' 0 , 32)
```

LA ABI 1 r1 ra

offs26<<2

2.2.3.4 JIRL

jirl rd, rj, offs16

JIRL PC 4 rd
16 offs16 2
rj

JIRL:

```
G = C + 4
C = G + E ( 16, 2' 0 , 32)
```

rd 0 JIRL

rd 0 rj 1 offs16 0 JIRL

offs16<<2

2.2.4

2.2.4.1 LD.{B[U]/H[U]/W}, ST.{B/H/W}

```
ld.b    rd, rj, si12
ld.h    rd, rj, si12
ld.w    rd, rj, si12
ld.bu   rd, rj, si12
ld.hu   rd, rj, si12
st.b    rd, rj, si12
st.h    rd, rj, si12
st.w    rd, rj, si12
```

LD.{B/H} / rd LD.W

rd

```
LD.B:
        = G     +     E     (  12, 32)
    A     C         C    (     )
        = A                 (     )
       = M     L    (     , B  E)
    G     =     E     (    , 32)
LD.H:
        = G     +     E     (  12, 32)
    A     C         C    (     )
        = A                 (     )
           = M     L    (     , HALF   D)
    G     =     E     (         , 32)
LD.W:
        = G     +     E     (  12, 32)
    A     C         C    (     )
        = A                 (     )
       = M     L    (     ,    D)
    G     =
```

LD.{BU/HU } / rd

LD.BU:

```
        = G      +     E      (  12, 32)
   A      C          C    (      )
        = A                  (      )
        = M     L    (      , B   E)
   G      =     E     (     , 32)
```

LD.HU:

```
        = G      +     E      (  12, 32)
   A      C          C    (      )
        = A                  (      )
          = M     L    (      , HALF   D)
   G      =     E     (         , 32)
```

ST.{B/H/W} rd [7:0]/[15:0]/[31:0]

ST.B:

```
        = G      +     E      (  12, 32)
   A      C          C    (      )
        = A                  (      )
   M          (G      7:0 ,      , B   E)
```

ST.H:

```
        = G      +     E      (  12, 32)
   A      C          C    (      )
        = A                  (      )
   M          (G      15:0 ,      , HALF   D)
```

ST.W:

```
        = G      +     E      (  12, 32)
   A      C          C    (      )
        = A                  (      )
   M          (G      31:0 ,      ,    D)
```

rj 12 si12

LD.{H[U]/W} ST.{B/H/W}

2.2.4.2 PRELD

preld hint, rj, si12

PRELD Cache Cache rj
12 si12 Cache
PRELD hint Cache hint 0~31 32
hint=0 load Cache hint=8 store Cache hint
NOP
PRELD Cache cached NOP

PRELD MMU

2.2.5

2.2.5.1 LL.W, SC.W

ll.w rd, rj, si14
sc.w rd, rj, si14

LL.W SC.W “ - - ” LL.W
rd SC.W
LL.W LL

2.2.6

2.2.6.1 DBAR

dbar hint

DBAR load/store hint

hint 0 load/store

“ DBAR 0” “ DBAR 0 ” load/store

2.2.7.3 RDCNTV{L/H}.W, RDCNTID

rdcntvl.w rd

rdcntvh.w rd

rdcntid rj

32 64 Stable Counter

Stable Counter 0 1 1

3

32 32

IEEE 754-2008

3-1

3-1

	FADD.S, FSUB.S, FMUL.S, FDIV.S, FMADD.S, FMSUB.S, FNMADD.S, FNMSUB.S, FMAX.S, FMIN.S, FMAXA.S, FMINA.S, FABS.S, FNEG.S, FSQRT.S, FRECIP.S, FRSQRT.S, FCOPYSIGN.S, FCLASS.S
	FCMP.cond.S
	FFINT.S.W, FTINT.W.S, FTINTRM.W.S, FTINTRP.W.S, FTINTRZ.W.S, FTINTRNE.W.S,
	FMOV.S, FSEL, MOVGR2FR.W, MOVFR2GR.S, MOVGR2FCSR, MOVFCSR2GR, MOVFR2CF, MOVCF2FR, MOVGR2CF, MOVCF2GR
	BCEQZ, BCNEZ
	FLD.S, FST.S

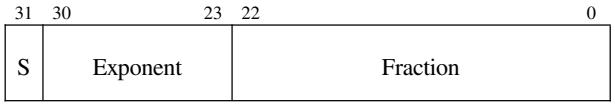
3.1

3.1.1

IEEE 754-2008

3.1.1.1

32



S Exponent Fraction 3-2

3-2

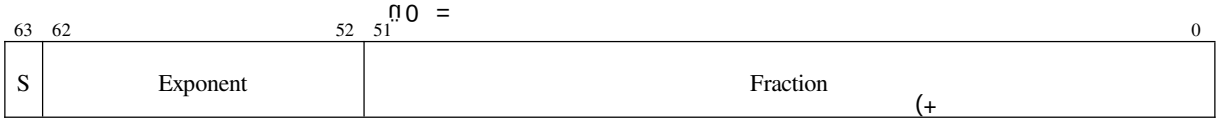
Exponent	Fraction	S	bit[22]	V
0	=0	0	0	+0
		1	0	-0
0	!=0	0		$+2^{-126} \times (0.\text{Fraction})$

Exponent	Fraction	S	bit[22]	V
		1		$-2^{-126} \times (0.\text{Fraction})$
[1, 0xFE]		0		$+2^{(\text{Exponent}-127)} \times (1.\text{Fraction})$
		1		$-2^{(\text{Exponent}-127)} \times (1.\text{Fraction})$
0xFF	=0	0	0	(+∞)
		1	0	(-∞)
0xFF	!=0		0	(Signaling Not a Number, SNaN)
			1	(Quiet Not a Number, QNaN)

± ∞ SNaN QNaN IEEE 754-2008

3.1.1.2

64



S Exponent Fraction 3-3

3-3

Exponent	Fraction	S	bit[51]	V
0	=0	0	0	+0
		1	0	-0
0	!=0	0		$+2^{-1022} \times (0.\text{Fraction})$
		1		$-2^{-1022} \times (0.\text{Fraction})$
[1, 0x7FE]		0		$+2^{(\text{Exponent}-1023)} \times (1.\text{Fraction})$
		1		$-2^{(\text{Exponent}-1023)} \times (1.\text{Fraction})$
0x7FF	=0	0	0	(+∞)
		1	0	(-∞)
0x7FF	!=0		0	(Signaling Not a Number,

fa fj fk fa fj fj fk

SNaN QNaN

SNaN 1

1

0 1

SNaN QNaN QNaN

QNaN QNaN

QNaN 0x7FC00000 QNaN 0x7FF8000000000000

3.1.2

Word W 32b

3.1.3

Floating-point Register FR

Condition Flag Register CFR Floating-point Control and Status Register FCSR

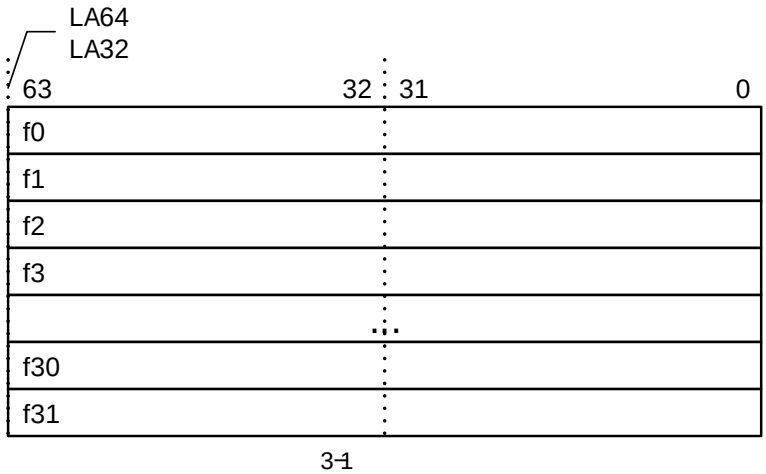
3.1.3.1

FR 32 f0~f31

FR 32 FR 64 LA32 LA64

32

FR



3-1

[31:0]

[63:32]

3.1.3.2

CFR 1 fcc0 CFR 1

1 0

3.1.3.3

FCSR 4 fcsr0~fcsr3 32 fcsr1~fcsr3 fcsr0

fcsr1~fcsr3 fcsr0 fcsr1~fcsr3 fcsr0

fcsr0 3-4

3-4 FCSR0

4:0	Enables	RW	VZOUI 4 V 3 Z 2 O 1 U 0 I
7:5	0	R0	0
9:8	RM	RW	4 0 RNE IEEE 754-2008 roundTiesToEven 1 RZ IEEE 754-2008 roundTowardZero 2 RP IEEE 754-2008 roundTowardsPositive 3 RM IEEE 754-2008 roundTowardsNegative
15:10	0	R0	0
20:16	Flags	RW	Flags VZOUI 20 V 19 Z 18 O 17 U 16 I
23:21	0	R0	0
28:24	Cause	RW	VZOUI 28 V 27 Z 26 O 25 U 24 I
31:29	0	R0	0

FCSR1 FCSR0 Enables FCSR0

FCSR2 FCSR0 Cause Flags FCSR0

FCSR3 FCSR0 RM FCSR0

3.1.4

IEEE 754-2008

Inexact (I)

Underflow (U)

Overflow (O)

Division by Zero (Z)

Invalid Operation (V)

FCSR0 Cause

FCSR0 Cause

FCSR0 Enables

1 Enable 1

Enable 0 FCSR0

Flag 1

3-5

3-5

I

	RNE	± 2^-126	0, subnormal, ± 2^-1022	normal
	RZ		0, subnormal	
U	RP	+2^-126	0, subnormal, +2^-1022	normal
	RM	-2^-126	0, subnormal, -2^-1022	normal
	RNE		+ ∞ - ∞	
	RZ			
O	RP	æ	Ϟ	ش

QNaN

3.1.4.2 (Z)

0

3.1.4.3 (O)

3.1.4.4 (U)

0

$(-2^{Emin}, 2^{Emin})$ Emin

3.2

3.2.1

3.2.1.1 F{ADD/SUB/MUL/DIV}.{S/D}

fadd.s	fd, fj, fk	fadd.d	fd, fj, fk
fsub.s	fd, fj, fk	fsub.d	fd, fj, fk
fmul.s	fd, fj, fk	fmul.d	fd, fj, fk
fdiv.s	fd, fj, fk	fdiv.d	fd, fj, fk

FADD.{S/D} fj / fk / / fd IEEE 754-2008 addition(x,y)

FADD.S:

```
F       31:0  = F 32          (F       31:0 , F       31:0 )
```

FADD.D:

```
F       = F 64          (F      , F      )
```

FSUB.{S/D} fj / fk / / fd IEEE 754-2008 subtraction(x,y)

FSUB.S:

```
F       31:0  = F 32            (F       31:0 , F       31:0 )
```

FSUB.D:

```
F       = F 64            (F      , F      )
```

FMUL.{S/D} fj / fk / / fd IEEE 754-2008 multiplication(x,y)

FMUL.S:

```
F       31:0  = F 32              (F       31:0 , F       31:0 )
```

FMUL.D:

```
F       = F 64              (F      , F      )
```

FDIV.{S/D} fj / fk / / fd IEEE 754-2008 division(x,y)

```
FDIV.S:
    F       31:0  = F 32          (F       31:0 , F       31:0 )
FDIV.D:
    F       = F 64          (F      , F      )
```

3.2.1.2 F{MADD/MSUB/NMADD/NMSUB}.{S/D}

fmadd.s	fd, fj, fk, fa	fmadd.d	fd, fj, fk, fa
fmsub.s	fd, fj, fk, fa	fmsub.d	fd, fj, fk, fa
fnmadd.s	fd, fj, fk, fa	fnmadd.d	fd, fj, fk, fa
fnmsub.s	fd, fj, fk, fa	fnmsub.d	fd, fj, fk, fa

FMADD.{S/D} fj / fk /
fa / /
fd

```
FMADD.S:
    F       31:0  = F 32       M        A   (F       31:0 , F       31:0 , F       31:
0 )
FMADD.D:
    F       = F 64       M        A   (F      , F      , F      )
```

FMSUB.{S/D} fj / fk /
fa / /
fd

```
FMSUB.S:
    F       31:0  = F 32       M        A   (F       31:0 , F       31:0 , -F       3
1:0 )
FMSUB.D:
    F       = F 64       M        A   (F      , F      , -F      )
```

FNMADD.{S/D} fj / fk /
fa / /
fd

```
FNMADD.S:
    F       31:0  = -F 32       M        A   (F       31:0 , F       31:0 , F       3
1:0 )
FNMADD.D:
    F       = -F 64       M        A   (F      , F      , F      )
```

FNMSUB.{S/D}	fj	/		fk	/

```
FMAXA.S:
    F       31:0  = F 32     N  M  (F       31:0 , F       31:0 )
FMAXA.D:
    F       = F 64     N  M  (F      , F      )
```

FMINA.{S/D} fj / fk / fd IEEE 754-2008 minNumMag(x,y)

```
FMINA.S:
    F       31:0  = F 32     N  M  (F       31:0 , F       31:0 )
FMINA.D:
    F       = F 64     N  M  (F      , F      )
```

3.2.1.5 F{ABS/NEG}.{S/D}

fabs.s	fd, fj	fabs.d	fd, fj
fneg.s	fd, fj	fneg.d	fd, fj

FABS.{S/D} fj / 0 fd IEEE 754-2008 abs(x)

```
FABS.S:
    F       31:0  = F 32     (F       31:0 )
FABS.D:
    F       = F 64     (F      )
```

FNEG.{S/D} fj / fd IEEE 754-2008 negate(x)

```
FNEG.S:
    F       31:0  = F 32        (F       31:0 )
FNEG.D:
    F       = F 64        (F      )
```

3.2.1.6 F{SQRT/RECIP/RSQRT}.{S/D}

fsqrt.s	fd, fj	fsqrt.d	fd, fj
frecip.s	fd, fj	frecip.d	fd, fj
frsqrt.s	fd, fj	frsqrt.d	fd, fj

FSQRT.{S/D} fj / /

fd IEEE 754-2008 squareRoot(x)

FSQRT.S:

```
F      31:0  = F 32          (F      31:0 )
```

FSQRT.D:

```
F      = F 64          (F      )
```

FRECIP.{S/D} fj / 1.0 / fd IEEE 754-2008 division(1.0,x)

FRECIP.S:

```
F      31:0  = F 32          (1.0, F      31:0 )
```

FRECIP.D:

```
F      = F 64          (1.0, F      )
```

FRSQRT.{S/D} fj / / 1.0 / fd IEEE 754-2008 rSqrt(x)

FRSQRT.S:

```
F      31:0  = F 32          (1.0, F            (F      31:0 ))
```

FRSQRT.D:

```
F      = F 64          (1.0, F            (F      ))
```

3.2.1.7 FCOPYSIGN.{S/D}

fcopysign.s fd, fj, fk fcopysign.d fd, fj, fk

FCOPYSIGN.{S/D} fj /

SNaN	QNaN	negative value				positive value			
		∞	normal	subnormal	0	∞	normal	subnormal	0

1

IEEE-754-2008 class(x)

FCLASS.S:

F 31:0 = F 32 (F 31:0)

FCLASS.D:

F = F 64 (F)

3.2.2

3.2.2.1 FCMP.cond.{S/D}

fcmp.cond.s cc, fj, fk fcmp.cond.d cc, fj, fk

cc cond 22

	cond	True Condition	QNaN IEEE 754-2008
CAF	0x0		
CUN	0x8	UN	compareQuietUnordered
CEQ	0x4	EQ	compareQuietEqual
CUEQ	0xC		

3.2.3

3.2.3.1 FCVT.S.D, FCVT.D.S

fcvt.s.d fd, fj fcvt.d.s fd, fj

FCVT.S.D fj

fd

FCVT.S.D:

```
F      31:0  = F 32        F     (F     , F 64)
```

FCVT.D.S fj

fd

FCVT.D.S:

```
F      = F 64        F     (F      31:0 , F 32)
```

IEEE 754-2008 convertFormat(x)

3.2.3.2 FFINT.{S/D}.{W/L}, FTINT.{W/L}.{S/D}

ffint.s.w	fd, fj	ftint.w.s	fd, fj
ffint.s.l	fd, fj	ftint.l.s	fd, fj
ffint.d.w	fd, fj	ftint.w.d	fd, fj
ffint.d.l	fd, fj	ftint.l.d	fd, fj

FFINT.{S/D}.{W/L} fj / /

/ fd IEEE 754-2008 convertFromInt(x)

FFINT.S.W:

```
F      31:0  = F 32        F   I  (F      31:0 ,  IN 32)
```

FFINT.S.L:

```
F      31:0  = F 32        F   I  (F     ,  IN 64)
```

FFINT.D.W:

```
F      = F 64        F   I  (F      31:0 ,  IN 32)
```

FFINT.D.L:

```
F      = F 64        F   I  (F     ,  IN 64)
```

FTINT.{W/L}.{S/D} fj / /

/ fd FCSR

IEEE 754-2008

	IEEE 754-2008
	convertToIntegerExactTiesToEven(x)
	convertToIntegerExactTowardZero(x)
	convertToIntegerExactTowardPositive(x)
	convertToIntegerExactTowardNegative(x)

FTINT.W.S:

F 31:0 = F 32 32(F 31:0 , FC . M)

FTINT.W.D:

F = F 64 32(F , FC . M)

FTINT.L.S:

F = F 32 64(F 31:0 , FC . M)

FTINT.L.D:

F = F 64 64(F , FC . M)

3.2.3.3 FTINT{RM/RP/RZ/RNE}.{W/L}.{S/D}

ftintrm.w.s	fd, fj	ftintrp.w.s	fd, fj
ftintrm.w.d	fd, fj	ftintrp.w.d	fd, fj
ftintrm.l.s	fd, fj	ftintrp.l.s	fd, fj
ftintrm.l.d	fd, fj	ftintrp.l.d	fd, fj
ftintrz.w.s	fd, fj	ftintrne.w.s	fd, fj
ftintrz.w.d	fd, fj	ftintrne.w.d	fd, fj
ftintrz.l.s	fd, fj	ftintrne.l.s	fd, fj
ftintrz.l.d	fd, fj	ftintrne.l.d	fd, fj

FTINTRM.{W/L}.{S/D} fj / /

/ fd “ ”

FTINTRM.W.S:

F 31:0 = F 32 32(F 31:0 , 3)

FTINTRM.W.D:

F = F 64 32(F , 3)

FTINTRM.L.S:

F = F 32 64(F 31:0 , 3)

FTINTRM.L.D:

F = F 64 64(F , 3)

FTINTRP.{W/L}.{S/D} fj / /

/ fd , “ ”

FTINTRP.W.S:

F 31:0 = F 32 32(F 31:0 , 2)

FTINTRP.W.D:

F = F 64 32(F , 2)

FTINTRP.L.S:

F = F 32 64(F 31:0 , 2)

FTINTRP.L.D:

F = F 64 64(F , 2)

FTINTRZ.{W/L}.{S/D} fj / /

/ fd “ ”

FTINTRZ.W.S:

F 31:0 = F 32 32(F 31:0 , 1)

FTINTRZ.W.D:

F = F 64 32(F , 1)

FTINTRZ.L.S:

F 31:0 = F 32 64(F 31:0 , 1)

FTINTRZ.L.D:

F = F 64 64(F , 1)

FTINTRNE.{W/L}.{S/D} fj / /

/ fd “ ”

FTINTRNE.W.S:

F 31:0 = F 32 32(F 31:0 , 0)

FTINTRNE.W.D:

F = F 64 32(F , 0)

FTINTRNE.L.S:

F = F 32 64(F 31:0 , 0)

FTINTRNE.L.D:

F = F 64 64(F , 0)

IEEE 754-2008

	IEEE 754-2008
FTINTRNE.{W/L}.{S/D}	convertToIntegerExactTiesToEven(x)
FTINTRZ.{W/L}.{S/D}	convertToIntegerExactTowardZero(x)
FTINTRP.{W/L}.{S/D}	convertToIntegerExactTowardPositive(x)
FTINTRM.{W/L}.{S/D}	convertToIntegerExactTowardNegative(x)

3.2.4

3.2.4.1 FMOV.{S/D}

fmov.s fd, fj fmov.d fd, fj

FMOV.{S/D} fj / fd fj
/

FMOV.S
```
F 31:0 = F 31:0
```
FMOV.d
```
F = F
```

IEEE 754 Cause Flags

3.2.4.2 FSEL

fsel fd, fj, fk, ca

FSEL FSEL ca 0 fj
fd fk fd

FSEL
```
F = CF ? F : F
```

3.2.4.3 MOVGR2FR.W, MOVGR2FRH.W

movgr2fr.w fd, rj
movgr2frh.w fd, rj

MOVGR2FR.W rj fd 32 64 fd
32

MOVGR2FR.W
```
F 31:0 = G
```

MOVGR2FRH.W rj fd 32 fd 32

MOVGR2FRH.W
```
F 63:32 = G
F 31: 0 = F 31:0
```

3.2.4.4 MOVFR2GR.S, MOVFRH2GR.S

```
movfr2gr.s      rd, fj
movfrh2gr.s     rd, fj
```

MOVFR2GR/MOVFRH2GR.S fj 32 / 32 rd

```
MOVFR2GR.S
    G       = F       31:0
MOVFRH2GR.S
    G       = F       63:32
```

3.2.4.5 MOVGR2FCSR, MOVFCSR2GR

```
movgr2fcsr      fcsr, rj
movfcsr2gr      rd, fcsr
```

MOVGR2FCSR rj fcsr

MOVGR2FCSR FCSR0 Cause Enables 1 FCSR1 Enables FCSR2 Cause Cause Enables 1 MOVGR2FCSR

```
MOVGR2FCSR
    FC          = G
```

MOVFCSR2GR fcsr 32 rd

```
MOVFCSR2GR
    G       = FC
```

fcsr

3.2.4.6 MOVFR2CF, MOVCF2FR

```
movfr2cf   cd, fj
movcf2fr   fd, cj
```

MOVFR2CF fj cd

```
MOVFR2CF
    CF      = F       0
```

MOVCF2FR cj fd fd 0

```
MOVCF2FR
```

```
F       0   =       (CF      , 64)
```

3.2.4.7 MOVGR2CF, MOVCF2GR

```
movgr2cf  cd, rj
movcf2gr  rd, cj
```

MOVGR2CF rj cd

```
MOVGR2CF
    CF      = G       0
```

MOVCF2GR cj rd rd 0

```
MOVCF2GR
    G       0   =     E      (CF      , 32)
```

3.2.5

3.2.5.1 BCEQZ, BCNEZ

```
bceqz     cj, offs21
bcnez     cj, offs21
```

BCEQZ cj 0

BCNEZ cj 0

21 offs21 2

PC

```
BCEQZ:
        CF      ==0 :
          C =  C +     E      (     21, 2' 0 , 32)
BCNEZ:
        CF      !=0 :
          C =  C +     E      (     21, 2' 0 , 32)
```

dk^ Ô

3.2.6

3.2.6.1 FLD.{S/D}, FST.{S/D}

fld.s	fd, rj, si12	
fld.d	fd, rj, si12	
fst.s	fd, rj, si12	
fst.d	fd, rj, si12	

FLD.S fd 32 64 fd 32 FLD.D fd

FLD.S:

```
= G      +     E     (  12, 32)
A      C         C    (      )
       = A                 (     )
      = M     L   (     ,    D)
F      31:0  =
```

FLD.D:

```
= G      +     E     (  12, 32)
A      C         C    (     )
       = A                 (     )
            = M     L   (     , D  BLE   D)
F      =
```

FST.S fd 32 FST.D fd

FST.S:

```
= G      +     E     (  12, 32)
A      C         C    (     )
       = A                 (     )
M          (F      31:0 ,      ,    D)
```

FST.D:

```
= G      +     E     (  12, 32)
A      C         C    (     )
       = A                 (     )
M          (F      63:0 ,      , D  BLE   D)
```

rj 12 si12

FLD.{S/D} FST.{S/D}

4

4.1

32 2 Privilege LeVel PLV PLV0 PLV3
CSR.CRMD PLV
PLV0
PLV3
Linux PLV0 PLV3

4.2

PLV0 PLV3 Hit CACOP

4.2.1 CSR

4.2.1.1 CSRRD, CSRWR, CSRXCHG

```
csrrd      rd, csr_num
csrwr      rd, csr_num
csrxchg    rd, rj, csr_num
```

CSRRD CSRWR CSRXCHG CSR CSRRD CSR
rd CSRWR rd CSR CSR
rd CSRXCHG rj rd
CSR 1 CSR CSR
rd
CSR CSR 14 csr_num
CSR CSR 0 CSR csr_num 0 1 CSR csr_num 1
32 CSR 32
CSR CSR 0
CSRWR CSRXCHG CSR
CSR

4.2.2 Cache

4.2.2.1 CACOP

cacop code, rj, si12

CACOP Cache Cache
rj 12 si12 CACOP VA
Cache
CACOP Cache Cache 5 code code[2:0]
Cache code[4:3]
code[2:0]=0 Cache code[2:0]=1 Cache code[2:0]=2
Cache
code[4:3]=0 Cache Store Tag Cache tag 0 Cache
1<<Way 1<<Index Cache Cache 1<<Offset
Cache VA[Way-1:0] VA[Index+Offset-1:Offset] Cache
code[4:3]=1 Cache Index Invalidate / Invalidate and Writeback
Cache
Cache Cache
Cache Cache Cache
Cache
code[4:3]=2 Cache Hit Invalidate / Invalidate and Writeback
Cache CACOP VA
load Cache Cache
CACOP TLB CACOP
Cache
code[4:3]=3 Cache

4.2.3 TLB

4.2.3.1 TLBSRCH

tlbsrch

CSR.ASID CSR.TLBEHI TLB
CSR.TLBIDX Index CSR.TLBIDX NE 0 CSR.TLBIDX

NE 1

TLB شي

4.2.3.5 INVTLB

invtlb op, rj, rk

INVTLB TLB TLB

op 5

rj [9:0] ASID “ ASID”

0 op ASID rj r0

rk “ VA ” op

rk r0

op op

op	
0x0	
0x1	op=0
0x2	G=1
0x3	G=0
0x4	G=0 ASID ASID
0x5	G=0 ASID ASID VA VA
0x6	G=1 ASID ASID VA VA

4.2.4

4.2.4.1 ERTN

ertn

ERTN

PPLV PIE CSR.CRMD ERA

PPLV PIE CSR.PRMD ERA CSR.ERA

ERTN CSR.LLBCTL KLO 1 LLbit 0 LLbit

4.2.4.2 IDLE

idle level

IDLE

IDLE

5

5.1

$0\sim2^{PALEN}-1$

32 PALEN 36

32

5.2

32 PLV0 2^{32}

32 MMU

CSR.CRMD DA=1 PG=0 MMU

[PALEN-1:0] 0

CSR.CRMD DA=0 PG=1 MMU

“ ” “ ”

5.2.1 5.4

5.2.1

MMU

load/store

CSR.DMW0~CSR.DMW1

/

5.3

2.1.7 32 Coherent Cached

CC Strongly-ordered UnCached SUC

MMU CSR.CRMD.DATF

load/store CSR.CRMD.DATM

MMU load/store

load/store

CSR MAT load/store

MAT

0—— 1——

2/3——

5.4

TLB

load/store

5.4.1 TLB

TLB

5.4.2 TLB

TLB 5-1

VPPN	PS	G	ASID	E
PPN0	PLV0	MAT0	D0	V0
PPN1	PLV1	MAT1	D1	V1

5-1 TLB

TLB

(E) 1 1 TLB

(ASID) 10

TLB ASID TLB

ASID

(G) 1 1 ASID

TLB G 1

(PS) 6 MTLB 2

32 4KB 4MB TLB PS

12 21[1]

(VPPN) (VALEN-13) 32

TLB /2

TLB TLB

(V) 1 1

(D) 1 1

(MAT) 2

store store TLB V=0

TLB V=0

TLB V=1

CSR.CRMD.PLV

PLV

store TLB V=1

D 0

5.4.3.2 TLB

TLB TLB TLB

4.2.3

5.4.3.3 TLB CSR

TLB CSR TLB

TLB

BADV

TLBEHI

TLBELO0

TLBELO1

TLBIDX

ASID

PGDL

PGDH

PGD

TLBRENTRY

CSR TLB 7.4 CSR

5.4.3.4 TLB

32 TLB “ INVTLB 0, r0, r0”

5.4.4 TLB

TLB

```
# va:
# mem_type:                FETCH          LOAD   load      STORE   store
# plv                      CSR.CRMD.PLV
# pa:
# mat:
# VALEN:
# PALEN:
# TLB[]: TLB[N]    TLB     N
# TLB_ENTRIES: TLB

#     TLB
          = 0
              ( LB EN  IE ) :
       ( LB   .E==1)
     (( LB   .G==1)    ( LB   .A ID==C  .A ID.A ID))
     ( LB   .   N  ALEN-1:  LB   .  +1 ==    ALEN-1:  LB   .  +1 ) :
          (          ==0) :
                     = 1
                    =  LB   .
              (            ==0) :
                       =  LB   . 0
                       =  LB   .D0
                         =  LB   .MA 0
                         =  LB   . L 0
                         =  LB   .  N0
                :
                       =  LB   . 1
                       =  LB   .D1
                         =  LB   .MA 1
                         =  LB   . L 1
                         =  LB   .  N1
           :
           #

    (          ==0) :
           E         ( LB )        #  TLB

    (        ==0) :
                  :
        FE CH :        E         ( IF)      #
```

6

6.1

6.1.1

32 12 1
IPI 1 TI 8 HWI0~HWI7 2 SWI0~SWI1

CSR.ESTAT.IS[12]
0
CSR.ESTAT.IS[11]
CSR.TICLR TI 1
8 HWI[7:0]
CSR.ESTAT.IS[9:2]
CSR CSR.ESTAT.IS[1:0] 1
0
CSR.ESTAT.IS Int Number SWI0 0
SWI1 1 …… IPI 12

6.1.2

IPI
TI …… SWI0

6.1.3

6.2.1

6.1.4

CSR.ESTAT.IS CSR.ECFG.LIE
int_vec CSR.CRMD.IE=1 int_vec 0

6.2.3

6.2

6.2.1

TLB CSR.TLBRENTRY

CSR.EENTRY CSR.ESTA

Ecode IS

6.2.2

TLB

(ALE) > TLB [1]

6.2.3

CSR.CRMD PLV IE CSR.PRMD PPLV PIE CSR.CRMD PLV

0 IE 0

PC CSR.ERA

ERTN

CSR.PRMD PPLV PIE CSR.CRMD PLV IE

CSR.ERA

CSR.PRMD PPLV

PIE CSR.PRMD

[1] TLB TLB

6.3

PC 0x1C000000 MMU

7

7.1

7-1

0x0		CRMD
0x1		PRMD
0x2		EUEN
0x4		ECFG
0x5		ESTAT
0x6		ERA
0x7		BADV
0xc		EENTRY
0x10	TLB	TLBIDX
0x11	TLB	TLBEHI
0x12	TLB 0	TLBELO0
0x13	TLB 1	TLBELO1
0x18		ASID
0x19		PGDL
0x1A		PGDH
0x1B		PGD
0x20		CPUID
0x30~0x33		SAVE0~SAVE3
0x40		TID
0x41		TCFG
0x42		TVAL
0x44		TICLR
0x60	LLBit	LLBCTL
0x88	TLB	TLBRENTRY
0x98		CTAG
0x180~0x181		DMW0~DMW1

7.2

7.2.1

“ ” “ ”

RW——

R——

R0—— 0 CSR

0

W1—— 1 0 0

7.2.2

CSR CSR

0

CSRWR CSRXCHG rd

0

7.3

7.4

7.4.1 CRMD

7-2

1:0	PLV	RW	0 3 0 3 0 ERTN CSR.PRMD PPLV
2	IE	RW	0 1 ERTN CSR.PRMD PIE
3	DA	RW	TLB 1 ERTN CSR.ESTAT.Ecode=0x3F 0 DA PG 0 1 1 0
4	PG	RW	TLB 0 ERTN CSR.ESTAT.Ecode=0x3F 1 PG DA 0 1 1 0
6:5	DATF	RW	PG 1 DATF 0b01
8:7	DATM	RW	load store PG 1 DATM 0b01
31:9	0	R0	0

7.4.2 PRMD

7-3

1:0	PPLV	RW	CSR.CRMD PLV ERTN CSR.CRMD PLV
2	PIE	RW	CSR.CRMD IE ERTN CSR.CRMD IE
31:3	0	R0	0

7.4.3 EUEN

7-4

0	FPE	RW	0 3.2 FPD
31:1	0	R0	0

7.4.4 ECFG

7-5

9:0	LIE[9:0]	RW	CSR.ESTAT IS[9:0] 10 W)- ϰ0

7.4.5 ESTAT

7-6

			0 1 SWI0 SWI1
1:0	IS[1:0]	RW	1 0
		8	HWI0~HWI7
9:2	IS[9:2]	R	
10	0	R0	0

0

8

Ecode	EsubCode		
0xC	0	BRK	
0xD	0	INE	
0xE	0	IPE	
0xF	0	FPD	
0x12	0	FPE	
0x1A - 0x3E			
0x3F	0	TLBR	TLB

7.4.6 ERA

PC

7-8

31:0	PC	RW	PC

7.4.7 BADV

TLB

ADEF PC

ALE

load PIL

store PIS

PIF

PME

PPI

7-9

31:0	VAddr	RW	TLB

7.4.8 EENTRY

TLB

2	KLO	RW	ERTN LLBit 1 ERTN LLBit 0 KLO 1 ERTN
31:3	0	R0	0

7.5

7.5.1 TLB TLBIDX

TLB TLB 7-14 Index

12:0	0	R	0
31:13	VPPN	RW	TLBRD TLB VPPN TLBSRCH TLB VPPN TLBWR TLBFILL TLB VPPN TLB load store [31:13]

7.5.3 TLB TLBELO0, TLBELO1

TLBELO0 TLBELO1 TLB TLB
32 TLB TLB
TLBELO0 TLBELO1 TLBELO0 TLBELO1
7-16
TLBWR TLBFILL TLB G PPN0 V0 PLV0 MAT0 D0 PPN1 V1 PLV1 MAT1 D1 TLBELO0 TLBELO1
TLBRD TLB TLBELO0 TLBELO1

7-16 TLB

0	V	RW	V
1	D	RW	D
3:2	PLV	RW	PLV
5:4	MAT	RW	MAT
6	G	RW	G TLBFILL TLBWR TLBELO0 TLBELO1 G 1 TLB G 1 TLBRD TLB G 1 TLBELO0 TLBELO1 G 1
7	0	R	0
PALEN-5:8	PPN	RW	PPN
31:PALEN-4	0	R	0 PALEN=36

7.5.4 ASID

TLB ASID ASID

ASID

7-17

9:0	ASID	RW	load/store TLB ASID TLBSRCH TLB ASID TLBWR TLBFILL TLB ASID TLBRD TLB ASID
15:10	0	R	0
23:16	ASIDBITS	R	ASID
31:24	0	R0	0

7.5.5 PGDL

4KB

12 0

7-18

11:0	0	R	0
31:12	Base	RW	[VALEN-1] 0

7.5.6 PGDH

4KB

12 0

7-19

11:0	0	R	0
31:12	Base	RW	[VALEN-1] 1

7.5.7 PGD

CSR

7-20

11:0	0	R	0
31:12	Base	R	CSR.BADV 0 CSR.PGDL Base CSR.PGDH Base

7.5.8 TLB TLBRENTRY

TLB TLB

7-21 TLB

5:0	0	R	TLB [5:0] 0
31:6	PA	RW	TLB [31:6]

7.5.9 DMW0~DMW1

5.2.1

7-22

0	PLV0	RW	1 PLV0
2:1	0	R0	0
3	PLV3	RW	1 PLV3
5:4	MAT	RW	
24:6	0	R0	0
27:25	PSEG	RW	[31:29]
28	0	R0	0
31:29	VSEG	RW	[31:29]

7.6

7.6.1 TID

RDCNTID ID

7-23

31:0	TID	RW	CSR.CPUID CoreID

7.6.2 TCFG

TimeVal

7-24

0	En	RW	1 0
1	Periodic	RW	1 0 TimeVal 0 0
n-1:2	InitVal	RW	4 0
31:n	0	R	0

7.6.3 TVAL

TimeVal

7-25

n-1:0	TimeVal	R	
31:n	0	R	0

7.6.4 TICLR

0 1

7-26

0 CLR W1 bit 1 { {

8 A

8.1

“ 'd ” “##'d” “##'d” ##

“ 'b ” “##'b” “##'b” ##

“ 'h ” “##'h” “##'h” ##

A~F

8-1

_____ _____ (____, …): _____ return _____	

if _____1:
_____1
elif _____2:
_____2
else:
_____3	

case _____ of: __1: _____1 __2: _____2 default: _____	case
_____ ? TRUE_____ : FALSE_____	
for _____ in ____ : _____	for
range(N)	0 N-1 1
range(_____, _____, _____)	
break	
signed(…)	
unsigned(…)	
fp16(…)	
fp32(…)	
fp64(…)	

boolean	
bit	
integer	
bits(*N*)	N
ZeroExtend(____,*N*)	N
SignExtend(____,*N*)	N
isSNaN(____)	signaling NaN TRUE FALSE
isQNaN(____)	quiet NaN TRUE FALSE
SignalException(____)	
#	
=	

8-2

[*M*:*N*]	N M
{*N*{*M*}}	M N
{*N*, *M*, …}	N, M, …

8-3

+	
-	
*	
/	
%	
**	

8-4

==	
!=	
>	
<	
>=	
<=	

8.2.1

```
(N)  LL(    (N)  ,          ):
    ==0 :
          =
    :
          =    N-  -1:0 ,     1' 0
```

8.2.2

```
(N)   L(    (N)  ,          ):
    ==0 :
          =
    :
          =      1' 0  ,   N-1:
```

8.2.3

```
(N)   A(    (N)  ,          ):
    ==0 :
          =
    :
          =        N-1   ,   N-1:
```

8.2.4

```
 (32)   F 32            32(    (32)  ,     (2)   ):
          :
    2' 0 :          32          I      E          E   ( )
    2' 1 :          32          I      E              ( )
    2' 2 :          32          I      E                  ( )
    2' 3 :          32          I      E          N       ( )
```

8.2.5

```
(64)   F 32              64(    (32)   ,     (2)   ):
          :
   2' 0 :             64          I      E          E    ( )
   2' 1 :             64          I      E               ( )
   2' 2 :             64          I      E                    ( )
   2' 3 :             64          I      E          N         ( )
```

8.2.6

```
(32)   F 64              32(    (64)   ,     (2)   ):
          :
   2' 0 :             32          I      E          E    ( )
   2' 1 :             32          I      E               ( )
   2' 2 :             32          I      E                    ( )
   2' 3 :             32          I      E          N         ( )
```

8.2.7

```
(64)   F 64              64(    (64)   ,     (2)   ):
          :
   2' 0 :             64          I      E          E    ( )
   2' 1 :             64          I      E               ( )
   2' 2 :             64          I      E                    ( )
   2' 3 :             64          I      E          N         ( )
```

8.2.8

```
(32)   F 32        I       (    (N)   ,     (2)   ):
     F 32        I       E    ( )
```

8.2.9

```
(64)   F 64        I       (    (N)   ,     (2)   ):
     F 64        I       E    ( )
```

9 B

		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDCNTID.W	rj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0			rj			0	0	0	0	0
RDCNTVL.W	rd	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0			rd		
RDCNTVH.W	rd	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0							

		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
FMIN.D	fd, fj, fk	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	1	0	fk					fj					fd			
FMAXA.S	fd, fj, fk	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	0	1	fk					fj					fd			
FMAXA.D	fd, fj, fk	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	0	1	0	fk					fj					fd			
FMINA.S	fd, fj, fk	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	0	1	fk					fj					fd			
FMINA.D	fd, fj, fk	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1	1	0	fk					fj					fd			
FCOPYSIGN.S	fd, fj, fk	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	0	1	fk					fj					fd			
FCOPYSIGN.D	fd, fj, fk	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1	1	0	fk					fj					fd			
FABS.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	fj					fd			
FABS.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1	0	fj				fd			
FNEG.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	1	fj				fd			
FNEG.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	1	0	fj				fd			
FCLASS.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	1	0	1	fj				fd			
FCLASS.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	0	1	1	1	0	fj				fd			
FSQRT.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	0	1	fj				fd			
FSQRT.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	fj				fd			
FRECIP.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0	1	fj				fd			
FRECIP.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	0	1	1	0	fj				fd			
FRSQRT.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	1	0	0	1	fj				fd			
FRSQRT.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	0	0	1	1	0	1	0	fj				fd			
FMOV.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	1	0	1	fj					fd			
FMOV.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	0	1	1	0	fj					fd			
MOVGR2FR.W	fd, rj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0	0	1	rj					fd			
MOVGR2FRH.W	fd, rj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1	0	1	1	rj					fd			
MOVFR2GR.S	rd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1	1	0	1	fj					rd			
MOVFRH2GR.S	rd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	0	1	1	1	1	fj					rd			
MOVGR2FCSR	fcsr, rj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	0	0	rj					fcsr			
MOVFCSR2GR	rd, fcsr	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	0	0	1	0	fcsr				rd			
MOVFR2CF	cd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	0	0	fj					0	0	cd	
MOVCF2FR	fd, cj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	0	1	0	0	cj			fd			
MOVGR2CF	cd, rj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	1	0	rj					0	0	cd	
MOVCF2GR	rd, cj	0	0	0	0	0	0	0	1	0	0	0	0	1	0	1	0	0	1	1	0	1	1	1	0	0	cj			rd			
FCVT.S.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	0	1	1	0	fj					fd			
FCVT.D.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	1	0	0	1	fj					fd			
FTINTRM.W.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	fj					fd			
FTINTRM.W.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	0	0	1	0	fj				fd			
FTINTRP.W.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	1	0	0	0	1	fj					fd			
FTINTRP.W.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	1	0	0	1	0	fj					fd			
FTINTRZ.W.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	fj					fd			
FTINTRZ.W.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	0	0	0	1	0	fj					fd			
FTINTRNE.W.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	1	0	0	0	1	fj					fd			
FTINTRNE.W.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	0	1	1	0	1	0	1	1	0	0	1	0	fj					fd			

		31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10	09 08 07 06 05	04 03 02 01 00
FTINT.W.S	fd, fj	0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 0 1	fj	fd
FTINT.W.D	fd, fj	0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0	fj	fd
FFINT.S.W	fd, fj	0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 0 1 0 0	fj	fd
FFINT.D.W	fd, fj	0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0	fj	fd
SLTI	rd, rj, si12	0 0 0 0 0 0 1 0 0 0	si12	rj	rd
SLTUI	rd, rj, si12	0 0 0 0 0 0 1 0 0 1	si12	rj	rd
ADDI.W	rd, rj, si12	0 0 0 0 0 0 1 0 1 0	si12	rj	rd
ANDI	rd, rj, ui12	0 0 0 0 0 0 1 1 0 1	ui12	rj	rd
ORI	rd, rj, ui12	0 0 0 0 0 0 1 1 1 0	ui12	rj	rd
XORI	rd, rj, ui12	0 0 0 0 0 0 1 1 1 1	ui12	rj	rd
CSRRD	rd, csr	0 0 0 0 0 1 0 0	csr	0 0 0 0 0	rd
CSRWR	rd, csr	0 0 0 0 0 1 0 0	csr	0 0 0 0 1	rd
CSRXCHG	rd, rj, csr	0 0 0 0 0 1 0 0	csr	rj!=0,1	rd
CACOP	code, rj, si12	0 0 0 0 0 1 1 0 0 0	si12	rj	code
TLBSRCH		0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 0	0 0 0 0 0	0 0 0 0 0
TLBRD		0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 1 0 1 1	0 0 0 0 0	0 0 0 0 0
TLBWR		0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 1 1 0 0	0 0 0 0 0	0 0 0 0 0
TLBFILL		0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 1 1 0 1	0 0 0 0 0	0 0 0 0 0
ERTN		0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0	0 0 0 0 0	0 0 0 0 0
IDLE	level	0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 0 1	level	
INVTLB	op, rj, rk	0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 1	rk	rj	op
FMADD.S	fd, fj, fk, fa	0 0 0 0 1 0 0 0 0 0 0 1	fa	fk	fj	fd
FMADD.D	fd, fj, fk, fa	0 0 0 0 1 0 0 0 0 0 0 1 0	fa	fk	fj	fd
FMSUB.S	fd, fj, fk, fa	0 0 0 0 1 0 0 0 0 0 1 0 1	fa	fk	fj	fd
FMSUB.D	fd, fj, fk, fa	0 0 0 0 1 0 0 0 0 0 1 1 0	fa	fk	fj	fd
FNMADD.S	fd, fj, fk, fa	0 0 0 0 1 0 0 0 1 0 0 1	fa	fk	fj	fd
FNMADD.D	fd, fj, fk, fa	0 0 0 0 1 0 0 0 1 0 1 0	fa	fk	fj	fd
FNMSUB.S	fd, fj, fk, fa	0 0 0 0 1 0 0 0 1 1 0 1	fa	fk	fj	fd
FNMSUB.D	fd, fj, fk, fa	0 0 0 0 1 0 0 0 1 1 1 0	fa	fk	fj	fd
FCMP.cond.S	cd, fj, fk	0 0 0 0 1 1 0 0 0 0 0 1	cond	fk	fj	0 0	cd
FCMP.cond.D	cd, fj, fk	0 0 0 0 1 1 0 0 0 0 1 0	cond	fk	fj	0 0	cd
FSEL	fd, fj, fk, ca	0 0 0 0 1 1 0 1 0 0 0 0	0 0	ca	fk	fj	fd
LU12I.W	rd, si20	0 0 0 1 0 1 0	si20	rd
PCADDU12I	rd, si20	0 0 0 1 1 1 0	si20	rd
LL.W	rd, rj, si14	0 0 1 0 0 0 0 0	si14	rj	rd
SC.W	rd, rj, si14	0 0 1 0 0 0 0 1	si14	rj	rd
LD.B	rd, rj, si12	0 0 1 0 1 0 0 0 0 0	si12	rj	rd
LD.H	rd, rj, si12	0 0 1 0 1 0 0 0 0 1	si12	rj	rd
LD.W	rd, rj, si12	0 0 1 0 1 0 0 0 1 0	si12	rj	rd
ST.B	rd, rj, si12	0 0 1 0 1 0 0 1 0 0	si12	rj	rd
ST.H	rd, rj, si12	0 0 1 0 1 0 0 1 0 1	si12	rj	rd

		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
ST.W	rd, rj, si12	0	0	1	0	1	0	0	1	1	0	si12												rj					rd				
LD.BU	rd, rj, si12	0	0	1	0	1	0	1	0	0	0	si12												rj					rd				
LD.HU	rd, rj, si12	0	0	1	0	1	0	1	0	0	1	si12												rj					rd				
PRELD	hint, rj, si12	0	0	1	0	1	0	1	0	1	1	si12												rj					hint				
FLD.S	fd, rj, si12	0	0	1	0	1	0	1	1	0	0	si12												rj					fd				
FST.S	fd, rj, si12	0	0	1	0	1	0	1	1	0	1	si12												rj					fd				
FLD.D	fd, rj, si12	0	0	1	0	1	0	1	1	1	0	si12												rj					fd				
FST.D	fd, rj, si12	0	0	1	0	1	0	1	1	1	1	si12												rj					fd				
DBAR	hint	0	0	1	1	1	0	0	0	0	1	1	1	0	0	1	0	0	hint														
IBAR	hint	0	0	1	1	1	0	0	0	0	1	1	1	0	0	1	0	1	hint														
BCEQZ	cj, offs	0	1	0	0	1	0	offs[15:0]																0	0	cj			offs[20:16]				
BCNEZ	cj, offs	0	1	0	0	1	0	offs[15:0]																0	1	cj			offs[20:16]				
JIRL	rd, rj, offs	0	1	0	0	1	1	offs[15:0]																rj					rd				
B	offs	0	1	0	1	0	0	offs[15:0]																offs[25:16]									
BL	offs	0	1	0	1	0	1	offs[15:0]																offs[25:16]									
BEQ	rj, rd, offs	0	1	0	1	1	0	offs[15:0]																rj					rd				
BNE	rj, rd, offs	0	1	0	1	1	1	offs[15:0]																rj					rd				
BLT	rj, rd, offs	0	1	1	0	0	0	offs[15:0]																rj					rd				
BGE	rj, rd, offs	0	1	1	0	0	1	offs[15:0]																rj					rd				
BLTU	rj, rd, offs	0	1	1	0	1	0	offs[15:0]																rj					rd				
BGEU	rj, rd, offs	0	1	1	0	1	1	offs[15:0]																rj					rd				