V1.11

2025 3

Loongson Technology Corporation Limited

2

Building No.2, Loongson Industrial Paae

1.11

2020/04/27

0.80	┌
0.90	┌
0.91	┌
1.00	┌

1.01

1) 3.2.3 convertToIntegerExact ň roundToIntegralExact ┌
2) 4.2.1 7.2.3 CSR CSR ┌
3) 5.1 LA32 36 ┌
4) CSR ┌

1.02

1) 2.2.1.3 ┘ 2.2.4 ┘ 2.2.5.3 ┘ 2.2.7.2 ┘ 3.2.5 ┌
1) FSCALEB.S/D ┘ FLOGB.S/D FRINT.S/D 6 LA64 ┌
2) 2.2.7.2 LL/SC ┌
3) 2.1.4 ┘ 2.1.5 5.2.1 ┌
4) 5.4.2 [log_2PS-1:12] [PS-1:12] ┌
5) 5.4.4 SignalException(PPI) ┌
6) 7.5.3 CSR TLBELO0 TLBELO1 LA32 PPN PALEN<36 ┌
7) 7.5.4 ASID INVTLB TLB ASID

	2) 2.2.3.3 sa2 sa3 0 ┌ 3) 2.2.10.5 CPUCFG 1 25 ┌ 4) 3.1.4.4 2Emin Emin ┌ 5) 3.2.2.1 CUNE IEEE 754-2008 compareQuietNotEqual compareSignalingNotEqual ┌ 6) 3.2.4.6 MOVCF2FR 3.2.4.7 MOVCF2GR 0 0 0 ┌ 7) 4.2.4.3 4.2.4.4 TLB CSR.TLBIDX.NE 1 TLB TLB ┌ 8) 4.2.5.1 Huge Page ┌ 9) 4.2.5.2 seq req Huge Page ┌ 10) 5.2 4 Ņ ņ PALEN VALEN ┌ 11) 5.2.1 ┌ 12) 5.4.3.4 Ņ INVTLB r0, r0 ņ Ņ INVTLB 0, r0, r0 ņ ┌ 13) 5.4.4 found_ppn ┌ 14) 5.4.5 PA $\log_2$PageSize 24 16MB ┌ 15) 6.2.1 CFG.VS=0 Ņ $2^{(CSR.ECFG.VS+2)}$ ¥ (ecode+64) ņ Ņ $2^{(CSR.ECFG.VS+2)}$ ¥ ecode ņ ┌ 16) 6.2.2 (ADE) (ALE) ┌ 17) 7.4.1 7-2 DATF DATM Ņ ň ň ň ň ņ Ņ ň ň ň ň ņ ┌ 18) 7.4.5 7-6 VS Ņ ň ň ņ Ņ ň ň ņ ┌ 19) 7.5.8 7.5.9 7-29 7-30 LDDIR ┘LDPTE ┌ 20) 7.5.8 7-29 PTEWIDTH 2 3 256 512 ┌
1.10	1) TiesToEven Ņ ņ ┌ 2) FTINT.W.D ┘ FTINT.L.S ┘ FTINTRM.W.D ┘ FTINTRM.L.S ┘ FTINTRP.W.D ┘ FTINTRP.L.S ┘ FTINTNE.W.D ┘ FTINTNE.L.S FR[fd] ┌ 3) CPUCFG Ņ 0x ņ ┌ LoongArchV1.1 ┌
1.11	

1) 2.2.1.3 2 r

2) 2.

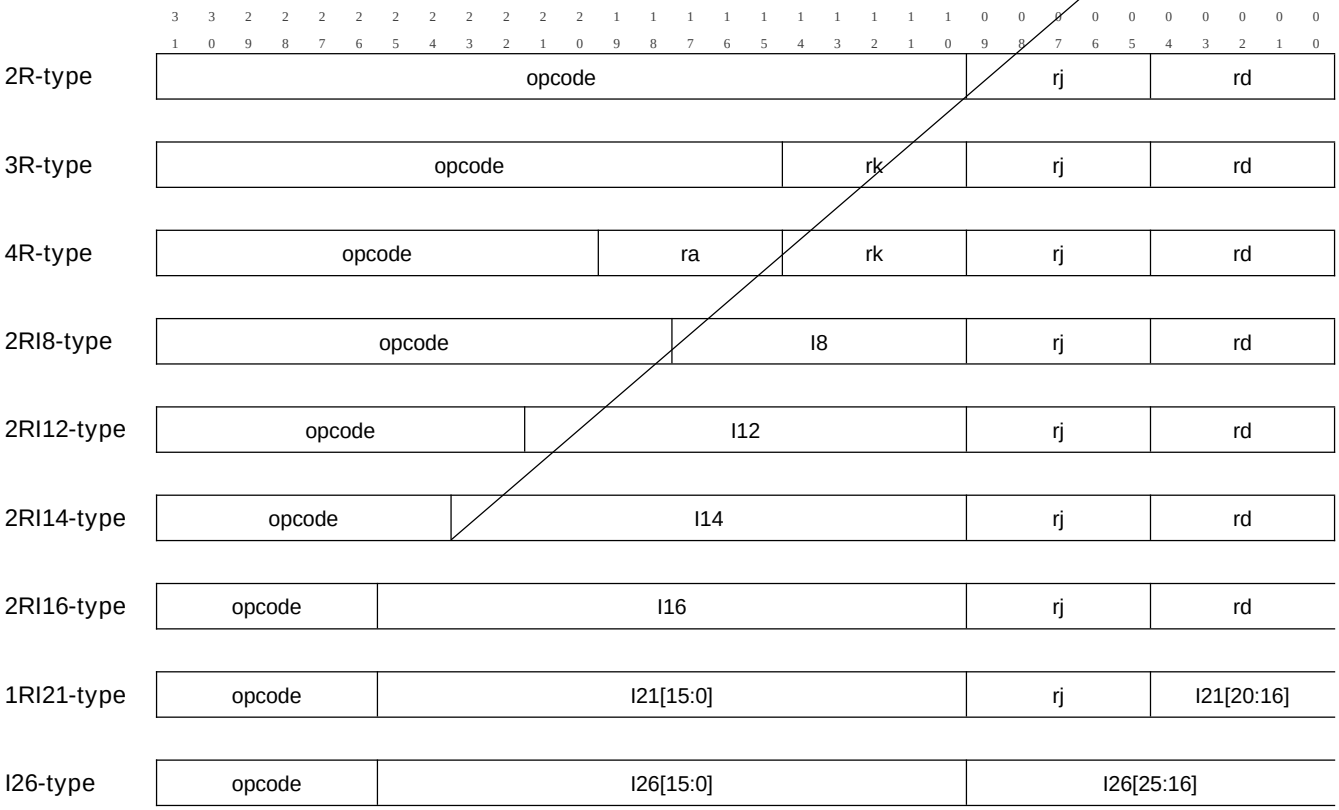
1

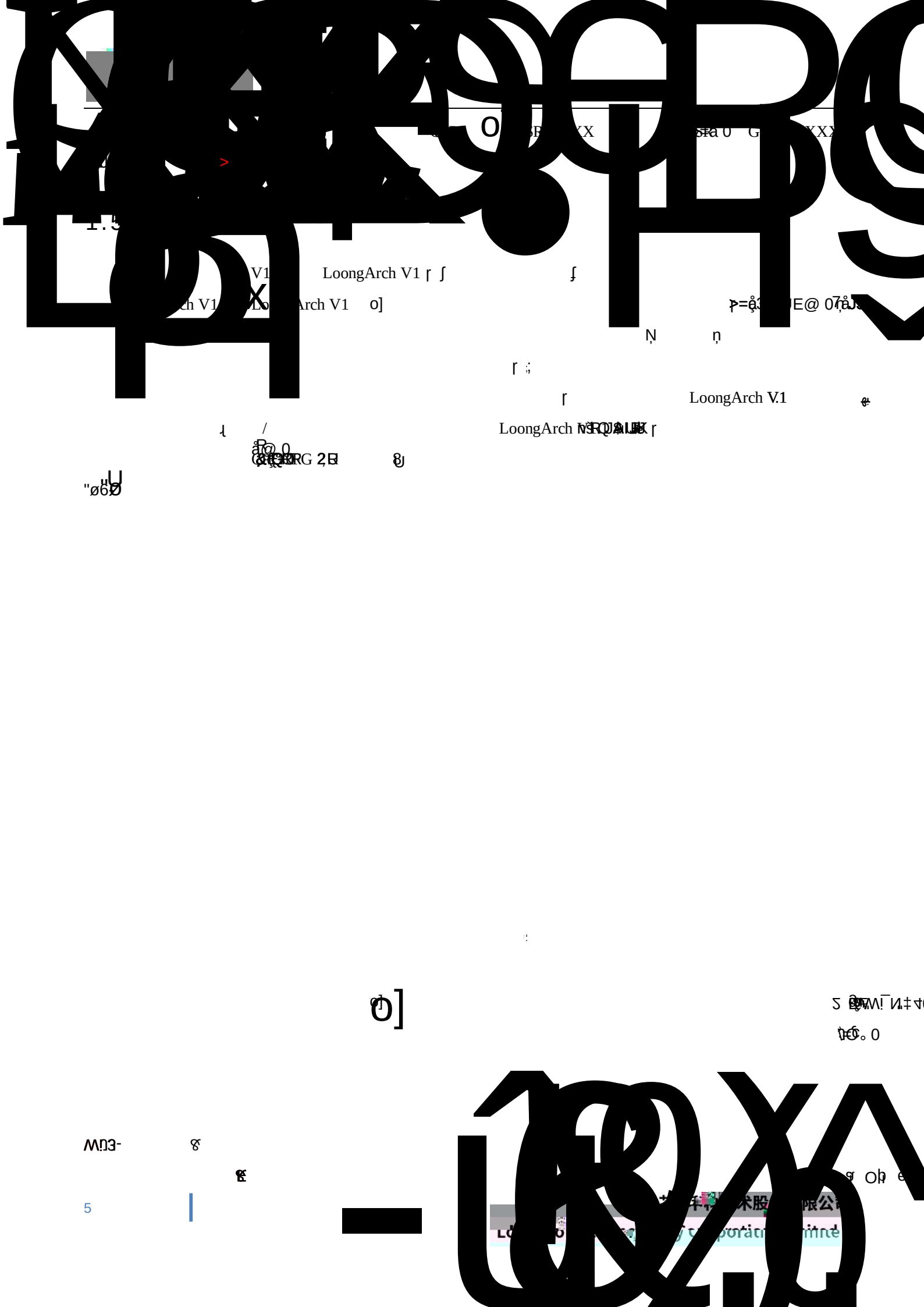
SIMD

128

256

1-1





2

r r

r

2.1

[1] r

r

r

2.1.1

5 bit b 丬 Byte B 8b 丬

Halfword H 16b 丬 Word W 32b 丬 Doubleword D

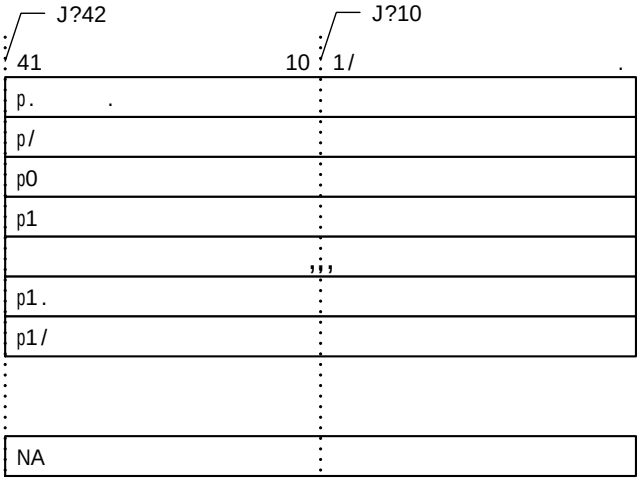
64b r LA32 r

丬 丬 r

2.1.2

General-purpose Register GR Program

Couner PC 2-1 r



2-1 PC

[1] r Linux user mode r

2.1.2.1

GR 32 r0~r31 0 r0 0 ┌GR GRLEN ┌LA32 GR 32 LA64 GR 64 ┌ ┌ 32 GR ┌ BL 1 r1 ┌ Application Binary Interface ABI r1 ┌

2.1.2.2PC

PC 1 ┌PC ┤ ┌ PC ┌PC GR ┌

2.1.3

r

[1]

SYSCALL SYS

BREAK BRK

INE

0,/,1

IPE

4

ADEF

ADEM

FPE 1,/,2

2.1.5

LA32 $0\sim2^{31}-1$

LA64 $0\sim2^{VALEN-1}-1$ VALEN

64 VALEN [40, 48]

CPUCFG 0x1 VALEN VALEN

2.1.6

2.1.7

Coherent Cached CC

Strongly-ordered UnCached SUC Weakly-ordered UnCached WUC

MAT Memory Access Type [2] MAT

[1]

[2]

r

r r

m} , Ů

2.1.9

Weakly Consistency WC

1.

2.

3.

DBAR IBAR DBAR AM LL-SC

2.1.9.1 load

load

dbar 0x700

load CPUCFG.3.LD_SEQ_SA[bit23] 1

2.2

LA64 LA32

2-1 LA32 GR 32

32 rd

2-1 LA32

	@CC-V RTA-V @CCH-V @KRK-V KT01H-V RKS RKST RKSH RKSTH OB@CCH OB@CCT01H OB@K@T01H @MC NQ MNQ WNQ @MCM NQM @MCH NQH WNQH LTK-V LTKG-V LTKG-VT CHU-V LNC-V CHU-VT LNC-VT
	RKK-V RQK-V RQ@-V QNSQ-V RKKH-V RQKH-V RQ@H-V QNSQH-V
	DWS-V-A DWS-V-G BKN-V BKY-V BSN-V BSY-V AXSDOHBJ-V QDUA-1G AHSQDU-3A AHSQDU-V ARSQHMR-V ARSQOHBJ-V L@RJDPY L@RJMDY
	ADP AMD AKS AFD AKST AFDT ADPY AMDY A AK IHQK
	KC-A KC-G KC-V KC-AT KC-GT RS-A RS-G RS-V OQDKC

	KK-V+ RB-V
	CA@Q+ HA@Q
	RXRB@KK+ AQD@J+ QCSHLDK-V+ QCSHLDG-V+ BEF

GR LA32 32 LA64

64 r r

2.2.1

2.2.1.1 ADD.{W

```
tmp = (GR[rj][31:0]<<(sa2+1)) + GR[rk][31:0]
GR[rd]
```

rd 1 0 ┌

SLT:

```
GR[rd] = (signed(GR[rj]) < signed(GR[rk])) ? 1 : 0
```

SLTU rj rk

rd 1 0 ┌

SLTU:

```
GR[rd] = (unsigned(GR[rj]) < unsigned(GR[rk])) ? 1 : 0
```

SLT SLTU ┌

2.2.1.6 SLT[U]I

```
slti    rd, rj, si12
sltui   rd, rj, si12
```

SLTI rj 12 si12

rd 1 0 ┌

SLTI:

```
tmp = SignExtend(si12, GRLEN)
GR[rd] = (signed(GR[rj]) < signed(tmp)) ? 1 : 0
```

SLTUI rj 12 si12

rd 1 0 ┌

SLTUI:

```
tmp = SignExtend(si12, GRLEN)
GR[rd] = (unsigned(GR[rj]) < unsigned(tmp)) ? 1 : 0
```

SLTI SLTUI ┌

SLTUI ┌

2.2.1.7 PCADDI, PCADDU12I, PCADDU18I, PCALAU12I

```
pcaddi      rd, si20
pcaddu12i   rd, si20
pcaddu18i   rd, si20
pcalau12i   rd, si20
```

PCADDI 20 si20 2 0 PC

rd ┌

NOR rj rk rd

r

```
NOR:
    GR[rd] = ~(GR[rj] | GR[rk])
```

XOR rj rk rd

r

```
XOR:
    GR[rd] = GR[rj] ^ GR[rk]
```

ANDN rk rj

rd r

```
ANDN:
    GR[rd] = GR[rj] & (~GR[rk])
```

ORN rk rj

rd r

```
ORN:
    GR[rd] = GR[rj] | (~GR[rk])
```

r

2.2.1.9 ANDI, ORI, XORI

andi rd, rj, ui12

ori rd, rj, ui12

xori rd, rj, ui12

ANDI rj 12

rd r

```
ANDI:
    GR[rd] = GR[rj] & ZeroExtend(ui12, GRLEN)
```

ORI rj 12

rd r

```
ORI:
    GR[rd] = GR[rj] | ZeroExtend(ui12, GRLEN)
```

XORI rj 12

rd ┌

```
XORI:
    GR[rd] = GR[rj] ^ ZeroExtend(ui12, GRLEN)
```

┌

2.2.1.10 NOP

NOP Ņandi r0, r0, 0” ┌ 4 PC 4

┌

2.2.1.11 MUL.{W/D}, MULH.{W[U]/D[U]}

mul.w rd, rj, rk

mulh.w rd, rj, rk

mulh.wu rd, rj, rk

mul.d rd, rj, rk

mulh.d rd, rj, rk

mulh.du rd, rj, rk

MUL.W rj [31:0] rk [31:0] [31:0]

rd ┌

```
MUL.W:
    product = signed(GR[rj][31:0]) * signed(GR[rk][31:0])
    GR[rd] = SignExtend(product[31:0], GRLEN)
```

MULH.W rj [31:0] rk [31:0]

[63:32] rd ┌

```
MULH.W:
    product = signed(GR[rj][31:0]) * signed(GR[rk][31:0])
    GR[rd] = SignExtend(product[63:32], GRLEN)
```

MULH.WU rj [31:0] rk [31:0]

[63:32] rd ┌

```
MULH.WU:
    product = unsigned(GR[rj][31:0]) * unsigned(GR[rk][31:0])
    GR[rd] = SignExtend(product[63:32], GRLEN)
```

2.2.1.13 DIV.{W[U],D[U]}, M[U].{W[U],D[U]}

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

DIV.W DIV.WU rj [31:0] rk [31:0]

rd

```
DIV.W:
    quotient = signed(GR[rj][31:0]) / signed(GR[rk][31:0])
    GR[rd] = SignExtend(quotient[31:0], GRLEN)
DIV.WU:
    quotient = unsigned(GR[rj][31:0]) / unsigned(GR[rk][31:0])
    GR[rd] = SignExtend(quotient[31:0], GRLEN)
```

MOD.W MOD.WU rj [31:0] rk [31:0]

rd

```
MOD.W:
MOD.WU:
```

ROTR.W rj [31:0] rd Γ

```
ROTR.W:
    tmp = ROTR(GR[rj][31:0], GR[rk][4:0])
    GR[rd] = SignEmig      ή
```

2.2.2.3 SLL.D, SRL.D, SRA.D, ROTR.D

```
sll.d    rd, rj, rk
srl.d    rd, rj, rk
sra.d    rd, rj, rk
rotr.d   rd, rj, rk
```

SLL.D rj [63:0] rd

```
SLL.D:
    GR[rd] = SLL(GR[rj][63:0], GR[rk][5:0])
```

SRL.D rj [63:0] rd

```
SRL.D:
    GR[rd] =SRL(GR[rj][63:0], GR[rk][5:0])
```

SRA.D rj [63:0] rd

```
SRA.D:
    GR[rd] = SRA(GR[rj][63:0], GR[rk][5:0])
```

ROTR.D rj [63:0] rd

```
ROTR.D:
    GR[rd] = ROTR(GR[rj][63:0], GR[rk][5:0])
```

rk [5:0]

2.2.2.4 SLLI.D, SRLI.D, SRAI.D, ROTRI.D

```
slli.d    rd, rj, ui6
srli.d    rd, rj, ui6
srai.d    rd, rj, ui6
rotri.d   rd, rj, ui6
```

SLLI.D rj [63:0] rd

```
SLLI.D:
    GR[rd] = SLL(GR[rj][63:0], ui6)
```

SRLI.D rj [63:0] rd

```
SRLI.D:
    GR[rd] =SRL(GR[rj][63:0], ui6)
```

SRAI.D rj [63:0] rd r

```
SRAI.D:
    GR[rd] = SRA(GR[rj][63:0], ui6)
```

ROTRI.D rj [63:0] rd r

```
ROTRI.D:
    GR[rd] = ROTR(GR[rj][63:0], ui6)
```

6 ui6 r

2.2.3

2.2.3.1 EXT.W.{B/H}

ext.w.b rd, rj

ext.w.h rd, rj

EXT.W.B rj [7:0] rd r

```
EXT.W.B:
    GR[rd] = SignExtend(GR[rj][7:0], GRLEN)
```

EXT.W.H rj [15:0] rd r

```
EXT.W.H:
    GR[rd] = SignExtend(GR[rj][15:0], GRLEN)
```

2.2.3.2 CL{O/Z}.{W/D}, CT{O/Z}.{W/D}

clo.w rd, rj clo.d rd, rj

clz.w rd, rj clz.d rd, rj

cto.w rd, rj cto.d rd, rj

ctz.w rd, rj ctz.d rd, rj

CLO.W rj [31:0] 31 0 N1”

rd r

```
CLO.W:
    GR[rd] = CLO
```

CLZ.W rj [31:0] 31 0 Ņ0”

rd ŗ

```
CLZ.W:
    GR[rd] = CLZ(GR[rj][31:0])
```

CTO.W rj [31:0] 0 31 Ņ1”

rd ŗ

```
CTO.W:
    GR[rd] = CTO(GR[rj][31:0])
```

CTZ.W rj [31:0] 0 31 Ņ0”

rd ŗ

```
CTZ.W:
    GR[rd] = CTZ(GR[rj][31:0])
```

CLO.D rj [63:0] 63 0 Ņ1”

rd ŗ

```
CLO.D:
    GR[rd] = CLO(GR[rj][63:0])
```

CLZ.D rj [63:0] 63 0 Ņ0”

rd ŗ

```
CLZ.D:
    GR[rd] = CLZ(GR[rj][63:0])
```

CTO.D rj [63:0] 0 63 Ņ1”

rd ŗ

```
CTO.D:
    GR[rd] = CTO(GR[rj][63:0])
```

” CTZ.D 63rj Z 0 Ņ0

2.2.3.3 BYTEPICK.{W/D}

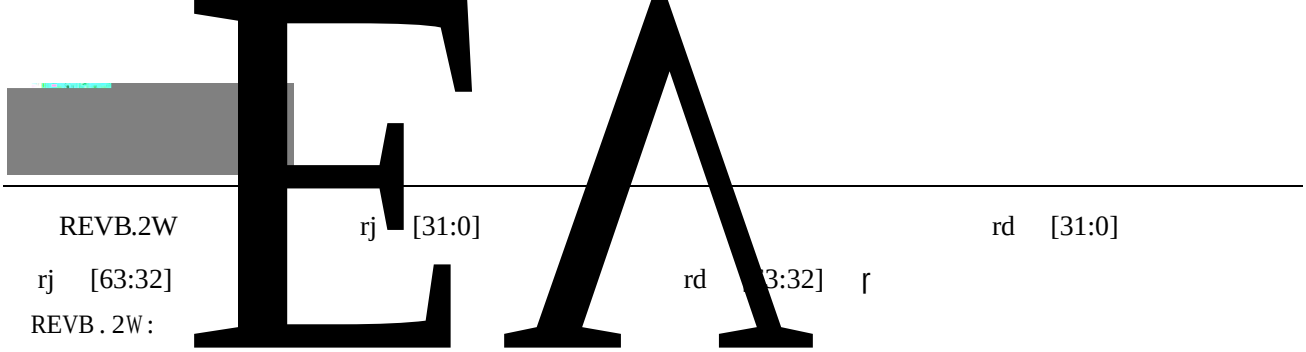
bytepick.w rd, rj, rk, sa2 bytepick.d rd, rj, rk, sa3

BYTEPICK.W rk [31:0] rj [31:0] 64 8

sa2 4 32

rd Γ

```
BYTEPICK.W:
    tmp = {GR[rk][31:0], GR[rj][31:0]}
    GR[rd] = SignExtend(tmp[8×(8-sa2)-1
```



REVB.2W rj [31:0] rd [31:0]

rj [63:32] rd [63:32] ¡

REVB.2W:

```
tmp0 = {GR[rj][7:0], GR[rj][15:8], GR[rj][23:16], GR[rj][31:24]}
tmp1 = {GR[rj][39:32], GR[rj][47:40], GR[rj][55:48], GR[rj][63:56]}
GR[rd] = {tmp1, tmp0}
```

REVB.D rj [63:0] 8 rd ¡

REVB.D:

```
GR[rd] = {GR[rj][7:0], GR[rj][15:8], GR[rj][23:16], GR[rj][31:24], GR[rj][3
9:32], GR[rj][47:40], GR[rj][55:48], GR[rj][63:56]}
```

2.2.3.5 REVH.{2W/D}

revh.2w rd, rj

revh.d rd, rj

REVH.2W rj [31:0] 2 rd [31:0]

rj [63:32] 2 rd [63:32] ¡

REVH.2W:

```
tmp0 = {GR[rj][15:0], GR[rj][31:16]}
tmp1 = {GR[rj][47:32], GR[rj][63:48]}
GR[rd] = {tmp1, tmp0}
```

REVH.D rj [63:0] 4 rd¡

REVH.D:

```
GR[rd] = { GR[rj][15:0], GR[rj][31:16], GR[rj][47:32], GR[rj][63:48]}
```

2.2.3.6 BITREV.{4B/8B}

bitrev.4b rd, rj bitrev.8b rd, rj

BITREV.4B rj [7:0] 8 [7:0] rj [15:8] 8 [15:8] rj [23:16] 8 [23:16] rj [31:24] 8 [31:24] 32 rd ¡

BITREV.4B:

```
bstr32[31:24] = BITREV(GR[rj][31:24])
bstr32[23:16] = BITREV(GR[rj][23:16])
bstr32[15: 8] = BITREV(GR[rj][15: 8])
bstr32[ 7:
```

BITREV.8B rj [7:0] 8 rd [7:0]
rj [15:8] 8 rd [15:8] rj [23:16] 8
rd [23:16] rj [31:24] 8
rd [31:24] rj [39:32] 8 rd [39:32]
rj [47:40] 8 rd [47:40] rj [55:48]
8 rd [55:48] rj [63:56] 8
rd [63:56]

BITREV.8B:

```
GR[rd][63:56] = BITREV(GR[rj][63:56])
GR[rd][55:48] = BITREV(GR[rj][55:48])
GR[rd][47:40] = BITREV(GR[rj][47:40])
GR[rd][39:32] = BITREV(GR[rj][39:32])
GR[rd][31:24] = BITREV(GR[rj][31:24])
GR[rd][23:16] = BITREV(GR[rj][23:16])
GR[rd][15: 8] = BITREV(GR[rj][15: 8])
GR[rd][ 7: 0] = BITREV(GR[rj][ 7: 0])
```

2.2.3.7 BITREV.{W/D}

bitrev.w rd, rj bitrev.d rd, rj

BITREV.W rj [31:0] 32 [31:0] 32

BSTRINS.D rd [msbd:lsbd] rj [msbd-lsbd:0] rd

r

```
BSTRINS.D:
    maskv = {(msbd-lsbd+1){1'b1}}<<lsbd
    GR[rd] = GR[rd][63:0]&~maskv[63:0] | (GR[rj][msbd-lsbd:0]<<lsbd)&maskv[63:0]
```

2.2.3.9 BSTRPICK.{W/D}

bstrpick.w rd, rj, msbw, lsbw bstrpick.d rd, rj, msbd, lsbd

BSTRPICK.W rj [msbw:lsbw] 32 32

rd r

```
BSTRPICK.W:
    bstr32[31:0] = ZeroExtend(GR[rj][msbw-lsb], 3. b
    GR[rd] = SignExtend( btr32[31:0], GR>N)
```

BSTRPICK.D rj [msbd:lsbd] 64 rd r

```
BSTRPICK.D:
    GR[rd] = ZeroExtend(GR[rj][msbd-lsb], 64)
```

2.2.3.10 MASKEQZ, MASKNEZ

maskeqz rd, rj, rk

masknez rd, rj, rk

MASKEQZ MASKNEZ r

MASKEQZ rk 0 rd 0 rj

r

```
MASKEQZ:
    GR[rd] = (GR[rk]==0) ? 0 : GR[rj]
```

MASKNEZ rk 0 rd 0 rj

r

```
MASKNEZ:
    GR[rd] = (GR[rk]!=0) ? 0 : GR[rj]
```

2.2.4

2.2.4.1 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 r

```
BEQ:
    if GR[rj]==GR[rd] :
        PC = PC + SignExtend({offs16, 2'b0}, GRLEN)
```

BNE rj rd r

```
BNE:
    if GR[rj]!=GR[rd] :
        PC = PC + SignExtend({offs16, 2'b0}, GRLEN)
```

BLT rj rd

r

```
BLT:
    if signed(GR[rj]) < signed(GR[rd]) :
        PC = PC + SignExtend({offs16, 2'b0}, GRLEN)
```

BGE rj rd

r

```
BGE:
    if signed(GR[rj]) >= signed(GR[rd]) :
        PC = PC + SignExtend({offs16, 2'b0}, GRLEN)
```

BLTU rj rd

r

```
BLTU:
    if unsigned(GR[rj]) < unsigned(GR[rd]) :
        PC = PC + SignExtend({offs16, 2'b0}, GRLEN)
```

B

offs26<<2 ┌

2.2.4.4 BL

bl offs26

BL PC 4 1 r1 ┌

26 offs26 2

PC ┌

BL:

```
GR[1] = PC + 4
PC = PC + SignExtend({offs26, 2'b0}, GRLEN)
```

LA ABI 1 r1 ra ┌

offs26<<2 ┌

2.2.4.5 JIRL

jirl rd, rj, offs16

JIRL PC 4 rd ┌

16 offs16 2

rj ┌

JIRL:

```
GR[rd] = PC + 4
PC = GR[rj] + SignExtend({offs16, 2'b0}, GRLEN)
```

rd 0 JIRL ┌

rd 0 rj 1 offs16 0 JIRL ┌

offs16<<2 ┌

```
LDX.D:
    vaddr = GR[rj] + GR[rk]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    GR[rd] = MemoryLoad(paddr, DOUBLEWORD)
```

LDX.{BU/HU/WU} / / rd ſ

```
LDX.BU:
    vaddr = GR[rj] + GR[rk]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    byte = MemoryLoad(paddr, BYTE)
    GR[rd] = ZeroExtend(byte, GRLEN)
LDX.HU:
    vaddr = GR[rj] + GR[rk]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    halfword = MemoryLoad(paddr, HALFWORD)
    GR[rd] = ZeroExtend(halfword, GRLEN)
LDX.WU:
    vaddr = GR[rj] + GR[rk]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    word = MemoryLoad(paddr, WORD)
    GR[rd] = ZeroExtend(word, GRLEN)
```

STX.{B/H/W/D} rd [7:0]/[15:0]/[31:0]/[63:0] ſ

```
STX.B:
    vaddr = GR[rj] + GR[rk]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    MemoryStore(GR[rd][7:0], paddr, BYTE)
STX.H:
    vaddr = GR[rj] + GR[rk]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    MemoryStore(GR[rd][15:0], paddr, HALFWORD)
STX.( ξ .( ξ
```

```
v    = G  rj] + GR    ]
A    ssCo   lianceCh   k(v   d
pa     dressTr                  r)
AddressSompliancecheck(vaddr)
MemoryStor nSwrkd{n^n^lmaddrmRpCB_@SpTy
```

```
R.D:
addr = GR[rj] + SignExtend({si14, 2
ddressComplianceCheck(vaddr)
paddr = AddressTr ation(vaddr)
MemoryStore(GR[rd][63:0], paddr, DOUBLEWORD)
```

14 si14 2 rj

r

si14<<2 r

LDPTR.{W/D} S

LDPTR.{W/D} ɻ STPTR

r

2.2.5.4 PRELD

preld

PRELD

12

PRELD hint Cache hint 0~31 32

r hint=0 load Cache hint=8 store Cache r hint

PRELD

r

2.2.5.5 PRELDX

preldx hint, rj, rk

PRELDX hint Cache ┌hint 0~31 32
┌ hint=0 load Cache hint=2 load Cache hint=8
store Cache ┌ hint NOP ┌
PRELDX Cache cached NOP
┌
PRELDX MMU ┌

2.2.6

2.2.6.1 LD{GT/LE}.{B/H/W/D}, ST{GT/LE}.{B/H/W/D}

```
ldgt.b    rd, rj, rk
ldgt.h    rd, rj, rk
ldgt.w    rd, rj, rk
ldgt.d    rd, rj, rk
ldle.b    rd, rj, rk
ldle.h    rd, rj, rk
ldle.w    rd, rj, rk
ldle.d    rd, rj, rk
stgt.b    rd, rj, rk
stgt.h    rd, rj, rk
stgt.w    rd, rj, rk
stgt.d    rd, rj, rk
stle.b    rd, rj, rk
stle.h    rd, rj, rk
stle.w    rd, rj, rk
stle.d    rd, rj, rk
```

LDGT/LDLE.{B/H/W} / / rd
LDGT/LDLE.D rd ┌
STGT/STLE.{B/H/W/D} rd [7:0]/[15:0]/[31:0]/[63:0] ┌
rj ┌
┌
LDGT.{B/H/W/D} STGT.{B/H/W/D} rj rk
┌ LDLE.{B/H/W/D} STLE.{B/H/W/D}
rj rk
┌

```
LDGT.B:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]>GR[rk] :
        byte = MemoryLoad(paddr, BYTE)
        GR[rd] = SignExtend(byte, GRLEN)
    else :
        RaiseException(BCE) #Bound Check Error Exception
LDGT.H:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]>GR[rk] :
        halfword = MemoryLoad(paddr, HALFWORD)
        GR[rd] = SignExtend(halfword, GRLEN)
    else :
        RaiseException(BCE) #Bound Check Error Exception
LDGT.W:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]>GR[rk] :
        word = MemoryLoad(paddr, WORD)
        GR[rd] = SignExtend(word, GRLEN)
    else :
        RaiseException(BCE) #Bound Check Error Exception
LDGT.D:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]>GR[rk] :
        GR[rd] = MemoryLoad(paddr, DW)
```

```
LDLE.B:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]<=GR[rk] :
        byte = MemoryLoad(paddr, BYTE)
        GR[rd] = SignExtend(byte, GRLEN)
    else :
        RaiseException(BCE) #Bound Check Error Exception
LDLE.H:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]<=GR[rk] :
        halfword = MemoryLoad(paddr, HALFWORD)
        GR[rd] = SignExtend(halfword, GRLEN)
    else :
        RaiseException(BCE) #Bound Check Error Exception
LDLE.W:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]<=GR[rk] :
        word = MemoryLoad(paddr, WORD)
        GR[rd] = SignExtend(word, GRLEN)
    else :
        RaiseException(BCE) #Bound Check Error Exception
LDLE.D:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]<=GR[rk] :
        GR[rd] = MemoryLoad(paddr, DOUBLEWORD)
    else :
        RaiseException(BCE) #Bound Check Error Exception
```

```
STGT.B:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]>GR[rk] :
        MemoryStore(GR[rd][7:0], paddr, BYTE)
    else :
        RaiseException(BCE) #Bound Check Error Exception
STGT.H:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]>GR[rk] :
        MemoryStore(GR[rd][15:0], paddr, HALFWORD)
    else :
        RaiseException(BCE) #Bound Check Error Exception
STGT.W:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]>GR[rk] :
        MemoryStore(GR[rd][31:0], paddr, WORD)
    else :
        RaiseException(BCE) #Bound Check Error Exception
STGT.D:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]>GR[rk] :
        MemoryStore(GR[rd][63:0], paddr, DOUBLEWORD)
    else :
        RaiseException(BCE) #Bound Check Error Exception
```

```
STLE.B:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]<=GR[rk] :
        MemoryStore(GR[rd][7:0], paddr, BYTE)
    else :
        RaiseException(BCE) #Bound Check Error Exception
STLE.H:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]<=GR[rk] :
        MemoryStore(GR[rd][15:0], paddr, HALFWORD)
    else :
        RaiseException(BCE) #Bound Check Error Exception
STLE.W:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]<=GR[rk] :
        MemoryStore(GR[rd][31:0], paddr, WORD)
    else :
        RaiseException(BCE) #Bound Check Error Exception
STLE.D:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]<=GR[rk] :
        MemoryStore(GR[rd][63:0], paddr, DOUBLEWORD)
    else :
        RaiseException(BCE) #Bound Check Error Exception
```

2.2

2.2.7 AM{SWAP/ADD/AND/OR/XOR/MAX/MIN}[_DB].{W/D}, AM{MAX/MIN}[_DB]{WU/DU}

	amswap.w	rd, rk, rj	amswap_db.w	rd, rk, rj
rd,	amswap.d	rd, rk, rj	amswap_db.d	rd, rk, rj
	amadd.w	rd, rk, rj	amadd_db.w	rd, rk, rj
	amadd.d	rd, rk, rj	amadd_db.d	rd, rk, rj
	amand.w	rd, rk, rj	amand_db.w	rd, rk, rj
	amand.d	rd, rk, rj	amand_db.d	rd, rk, rj
	amor.w	rd, rk, rj	amor_db.w	rd, rk, rj
	amor.d	rd, rk, rj	amor_db.d	rd, rk, rj
	amxor.w	rd, rk, rj	amxor_db.w	rd, rk, rj
	amxor.d	rd, rk, rj	amxor_db.d	rd, rk, rj
	ammax.w	rd, rk, rj	ammax_db.w	rd, rk, rj
	ammax.d	rd, rk, rj	ammax_db.d	rd, rk, rj
	ammi&	m" ƀ		

rk ┌ AMOR[_DB].{W/D}

rk ┌AMXOR[_DB].{W/D}

rk ┌ AMMAX[_DB].{W/D}

rk ┌AMMIN[_DB].{W/D}

rk ┌ AMMAX[_DB].{WU/DU}

2.2.7.3 AMCAS[_DB].{B/H/W/D}

amcas.b	rd, rk, rj	amcas_db.b	rd, rk, rj
amcas.h	rd, rk, rj	amcas_db.h	rd, rk, rj
amcas.w	rd, rk, rj	amcas_db.w	rd, rk, rj
amcas.d	rd, rk, rj	amcas_db.d	rd, rk, rj

AMCAS[_DB].{B/H/W/D} / / / Compare-and-Swap

/ rd [7:0]/[15:0]/[31:0]/[63:0]

rk [7:0]/[15:0]/[31:0]/[63:0]

SC LLbit rd Γ

LL-SC LLbit 0

E

LLbit rd ŗ

LL-SC LLbit

```
CRC.W.D.W:
    chksum = CRC32(GR[rk][31:0], GR[rj][63:0], 64, 0xEDB88320)
```

2.2.10.2 BREAK

break code

BREAK ┌

code ┌

2.2.10.3 ASRT{LE/GT}.D

asrtle.d rj, rk

asrtgt.d rj, rk

rj rk

┌ ASRTLE.D rj rk

ASRTGT.D rj rk ┌

2.2.10.4 RDTIME{L/H}.W, RDTIME.D

rdtimel.w rd, rj rdtime.d rd, rj

rdtimeh.w rd, rj

64 Stable Counter ┌ Stable

Counter 0 1 1 0 ┌

Counter ID ┌

┌

RDTIME{L/H}.W RDTIME.D Stable Counter

rd Counter ID rj ┌ Stable Counter

RDTIMEL.W Counter [31:0] RDTIMEH.W Counter [63:32] RDTIME.D

64 Counter ┌ 64 RDTIME{L/H}.W 32

rd ┌ RDTIME{L/H}.W 32 64 Counter ┌

2.2.10.5 CPUCFG

cpucfg rd, rj

CPUCFG ┌

CPUCFG

┌

CPUCFG rj

rd ┌ 32 LA64

┌

08	KAS^@QL	0	@QL
1/	KAS^LHOR	0	LHOR
10	KROV	0	
11	K@L	0	@L)
13	GOSV	0	O`fd S`akd V`kjdq
14	EQDBHOD	0	EQDBHOD-zR.C| ɻ EQRPQSD-zR.C| 3
		017	UEQDBHOD-zR.C| ɻ UEQRPQSD-zR.C| 3
		145	WUEQDBHOD-zR.C| ɻ WUEQRPQSD-zR.C|
		3	ɼ
15	CHU21	0 53	CHU-VZT\ LNC-VZT\ 21
16	K@L^AG	0	@LzRV@O.@CC|Z^CA\-zA.G| 7
17	K@LB@R	0	@LB@RZ^CA\-zA.G.V.C| 7
18	KK@BP^RBQDK	0	KK@BP-zV.C| ɻ RBQDK-zV.C| 3
2/	RBP	0	RB-P
/	BBCL@	0	B`bgd Bngdqdms CL@
0	REA	0	Rsnqd Ehkk Ateedq REA
1	TB@BB	0	tb`bb vhm
2	KKDWB	1ɔ	

/w2

	0	K0 HT Tmhex	0 K0 HT^Oqdrdms B`bgd B`bgd
	1	K0 C Oqdrdms	0 B`bgd
	2	K1 HT Oqdrdms	0 B`bgd B`bgd
	3	K1 HT Tmhex	0 K1 HT^Oqdrdms B`bgd B`bgd
	4	K1 HT Oqhu`sd	0 K1 HT^Oqdrdms B`bgd
	5	K1 HT Hmbktrhud	0 K1 HT^Oqdrdms B`bgd K0
	6	K1 C Oqdrdms	0 B`bgd
	7	K1 C Oqhu`sd	0 B`bgd
	8	K1 C Hmbktrhud	0 B`bgd K0
	0/	K2 HT Oqdrdms	0 B`bgd B`bgd
	00	K2 HT Tmhex	0 K2 HT^Oqdrdms B`bgd B`bgd
	01	K2 HT Oqhu`sd	0 K2 HT^Oqdrdms B`bgd
	02	K2 HT Hmbktrhud	0 K2 HT^Oqdrdms B`bgd K0 K1
	03	K2 C Oqdrdms	0 B`bgd
	04	K2 C Oqhu`sd	0 B`bgd
	05	K2 C Hmbktrhud	0 B`bgd K0 K1
/w00	049/	V`x,0	,0 0/ K0 HT^Oqdrdms B`bgd
	12905	Hmcdw,knf1	knf$_1$' B`bgd ( 0/ K0 HT^Oqdrdms B`bgd
	2/913	Khmdrhyd,knf1	knf$_1$'B`bgd ( 0/ K0 HT^Oqdrdms B`bgd
/w01	049/	V`x,0	,0 0/ K0 C Oqdrdms B`bgd
	12905	Hmcdw,knf1	knf$_1$' B`bgd ( 0/ K0 C Oqdrdms B`bgd
	2/913	Khmdrhyd,knf1	knf$_1$'B`bgd ( 0/ K0 C Oqdrdms B`bgd
/w02	049/	V`x,0	,0 0/ K1 HT Oqdrdms B`bgd
	12905	Hmcdw,knf1	knf$_1$' B`bgd ( 0/ K1 HT Oqdrdms B`bgd
	2/913	Khmdrhyd,knf1	knf$_1$'B`bgd ( 0/ K1 HT Oqdrdms B`bgd
/w03	049/	V`x,0	,0 0/ K2 HT Oqdrdms B`bgd
	12905	Hmcdw,knf1	knf$_1$' B`bgd ( 0/ K2 HT Oqdrdms B`bgd
	2/913	Khmdrhyd,knf1	knf$_1$'B`bgd ( 0/ K2 HT Oqdrdms B`bgd

3

r

IEEE 754-2008 r

r

r

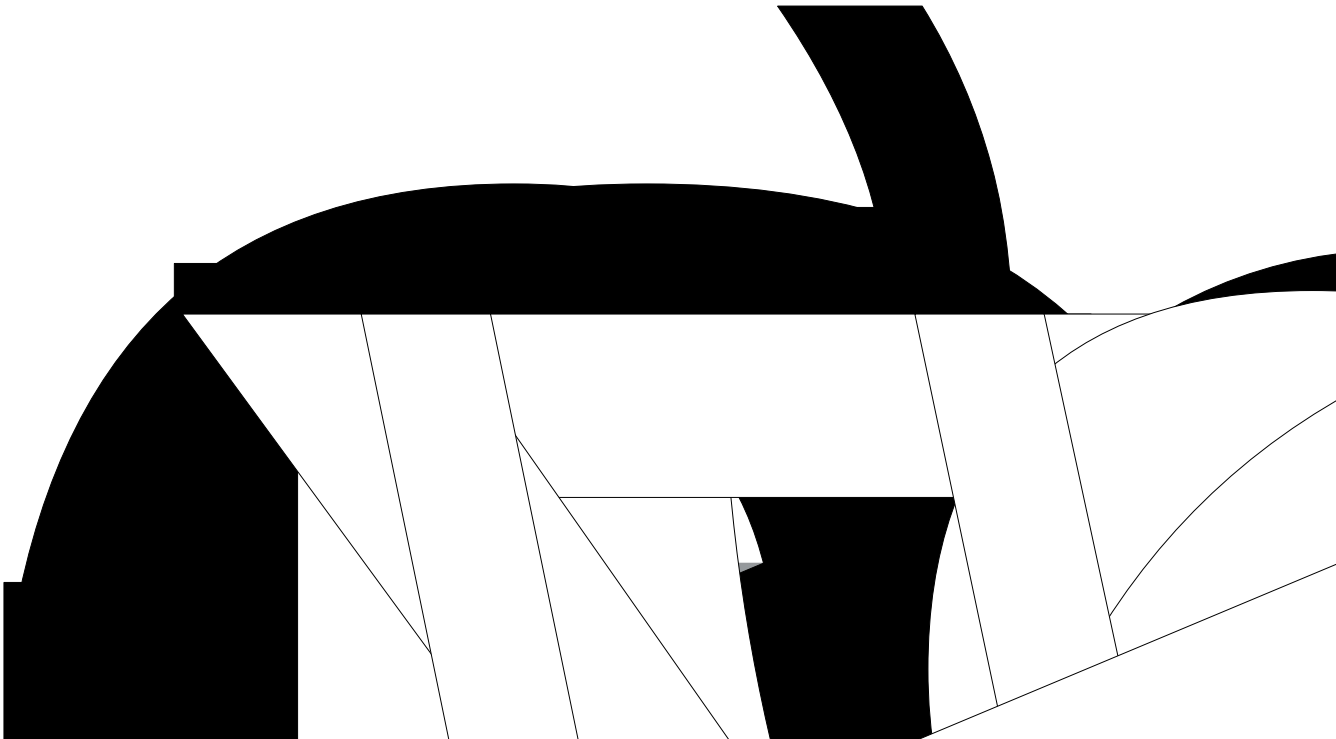
3-1 r

LA32 LA64 r

MOVGR2FR.D ɻ MOVFR2GR.D ɻ FSCALEB.S/D ɻ FLOGB.S/D ɻ FRINT.S/D ɻ FRECIPE.S/D ɻ FRSQRTE.S/D ɻ FLD{GT/LE}.{S/D} FST{GT/LE}.{S/D} 20 LA64 r

3-1

E@CC-R+ ERTA-R+ ELTK-R+ ECHU-R+ EL@CC-R+ ELRTA-R+ EML@CC-R+ EMLRTA-R+ EL@W-R+ ELHM-R+
EL@W@-R+ ELHM@-R+ E@AR-R+ EMDF-R+ ERPQS-R+ EQDBHO-R+ EQRPQS-R+ EBNOXRHFM-R+ EBK@RR-R
EBLO-bnmc-R
EEHMS-R-V+ ESHMS-V-R+ ESHMSQL-V-R+ ESHMSQO-V-R+ ESHMSQY-V-R+ ESHMSQMD-V-R+
ELNU-R+ ERDK+ LNUFQ1EQ-V+ LNUEQ1FQ-R+ LNUFQ1EBRQ+ LNUEBRQ1FQ+ LNUEQ1BE+
LNUBE1EQ+ LNUFQ1BE+ LNUBE1FQ
-RA



EEE

onmdms	Eq`bshnm	R	ahsZ11\	U
/	</	/	/	*/
		0	/	,/
/	</	/		*1'015¥'/-Eq`bshnm(
		0		,1'015¥'/-Eq`bshnm(
/wED\		/		*1'Dwonmdms,016(¥'0-Eq`bshnm(
		0		,1'Dwonmdms,016(¥'0-Eq`bshnm(
/wEE	</	/	/	'*Ÿ(
		0	/	',Ÿ(
/wEE	</		/	'Rhfm`khmf Mns ` Mtladq+ RM`M(
			0	'Pthds Mns ` Mtladq+ PM`M(

QNaN ɾ QNaN SNaN

NaN ɾ

fj fk fj fk

fa ɻ fj fk fa fj fj fk ɾ

SNaN QNaN

SNaN 1 ɾ

א t خ

d.
d/
dO
d1
,,,
d1.
d1/

QMD *Ÿ ,Ÿ

QY

N

QO *Ÿ

QL ,Ÿ

Y /

U

3.1.4.5 (l)

FPU

r

r

r

3.2

division(x,y)

fnmsub.s fd, fj, fk, fa fnmsub.d fd, fj, fk, fa

FMADD.{S/D}

FMADD.S:

```
    FR[fd][31:0] = FP32_fusedMultiplyAdd(FR[fj][31:0], FR[fk][31:0], FR[fa][31:
0])
```

FMADD.D:

```
    FR[fd] = FP64_fusedMultiplyAdd(FR[fj], FR[fk], FR[fa])
```

FMSUB.{S/D}

```
FR[fd] = FP64_maxNumMag(FR[fj], FR[fk])
```

FMINA.{S/D} fj / fk /

fd ┌ IEEE 754-2008 minNumM

FSQRT.S:

```
FR[fd][31:0] = FP32_squareRoot(FR[fj][31:0])
```

FSQRT.D:

```
FR[fd] = FP64_squareRoot(FR[fj])
```

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

FRECIP.S:

```
FR[fd][31:0] = FP32_division(1.0, FR[fj][31:0])
```

FRECIP.D:

```
FR[fd] = FP64_division(1.0, FR[fj])
```

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

FRSQRT.S:

```
FR[fd][31:0] = FP32_division(1.0, FP_squareRoot(FR[fj][31:0]))
```

FRSQRT.D:

```
FR[fd] = FP64_division(1.0, FP_squareRoot(FR[fj]))
```

3.2.1.7 F{SCALEB/LOGB/COPYSIGN}.{S/D}

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

FSCALEB.{S/D} fj / a fk / N a*2^N / fd ┌ IEEE 754-2008 scaleB(x, N) ┌

FSCALEB.S:

```
FR[fd][31:0] = FP32_scaleB(FR[fj][31:0], FR[fk][31:0])
```

FSCALEB.D:

```
FR[fd] = FP64_scaleB(FR[fj], FR[fk])
```

FLOGB.{S/D} fj / 2 / fd ┌ IEEE 754-2008 logB(x) ┌

FLOGB.S:

```
FR[fd][31:0] = FP32_logB(FR[fj][31:0])
```

FLOGB.D:

```
FR[fd] = FP64_logB(FR[fj])
```

FCOPYSIGN.{S/D} fj / fk

/ / fd ┌

IEEE 754-2008 copySign(x, y) ┌

```
COPYSIGN.S:
    FR[fd][31:0] = FP32_copySign(FR[fj][31:0], FR[fk][31:0])
FCOPYSIGN.D:
    FR[fd] = FP64_copySign(FR[fj], FR[fk])
```

3.2.1.8 FCLASS.{S/D}

fclass.s fd, fj fclass.d fd, fj

fj 10

Ahs/	Ahs0	Ahs1	Ahs2	Ahs3	Ahs4	Ahs5	Ahs6	Ahs7	Ahs8
RM`M	PM`M	mdf`shud u`ktd				onrhshud u`ktd			
		Ÿ	mnq l`k	rtamnq l`k	/	Ÿ	mnq l`k	rtamnq l`k	/

1 ┌

IEEE-754-2008 class(x) ┌

```
FCLASS.S:
    FR[fd][31:0] = FP32_class(FR[fj][31:0])
FCLASS.D:
    FR[fd] = FP64_class(FR[fj])
```

3.2.1.9 F{RECIPE/RSQRTE}.{S/D}

frecipe.s fd, fj frecipe.d fd, fj

frsqrte.s fd, fj frsqrte.d fd, fj

FRECIPE.{S/D} fj / 1.0

/ fd ┌ 2^{-14} ┌

2^{N} 2^{-N} ┌ QNaN ↓ SNaN ↓ ±∞ ↓ £0 ↓

FRECIP.{S/D} ┌

```
FRECIPE.S:
    FR[fd][31:0] = FP32_reciprocal_estimate(FR[fj][31:0])
FRECIPE.D:
    FR[fd] = FP64_reciprocal_estimate(FR[fj])
```

FRSQRTE.{S/D} fj / /

1.0 / f

2^{-14} ┌

2^{2N} 2^{-N} ┌ QNaN ┤ SNaN ┤ ±∞ ┤ £0 ┤

FRSQRT.{S/D} ┌

FRSQRTE.S:

```
    FR[fd][31:0] = FP32_reciprocal_squareroot_estimate(FR[fj][31:0])
```

FRSQRTE.D:

```
    FR[fd] = FP64_reciprocal_squareroot_estimate(FR[fj])
```

3.2.2

3.2.2.1 FCMP.cond.{S/D}

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

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

FCVT.S.D fj

fd ŗ

FCVT.S.D:

FR[fd][31:0] ₫32Ĵ_convertFormat(FGCFR[fj][31:0],FP64)

FCVT.D.S fj

fd ŗ

FCVT.D.S:

FR[fd] ₫64Ĵ_convertFormat(FGCFR[fj][31:0], FP32)

IEEE 754-2008 convertFormat(x) ŗ

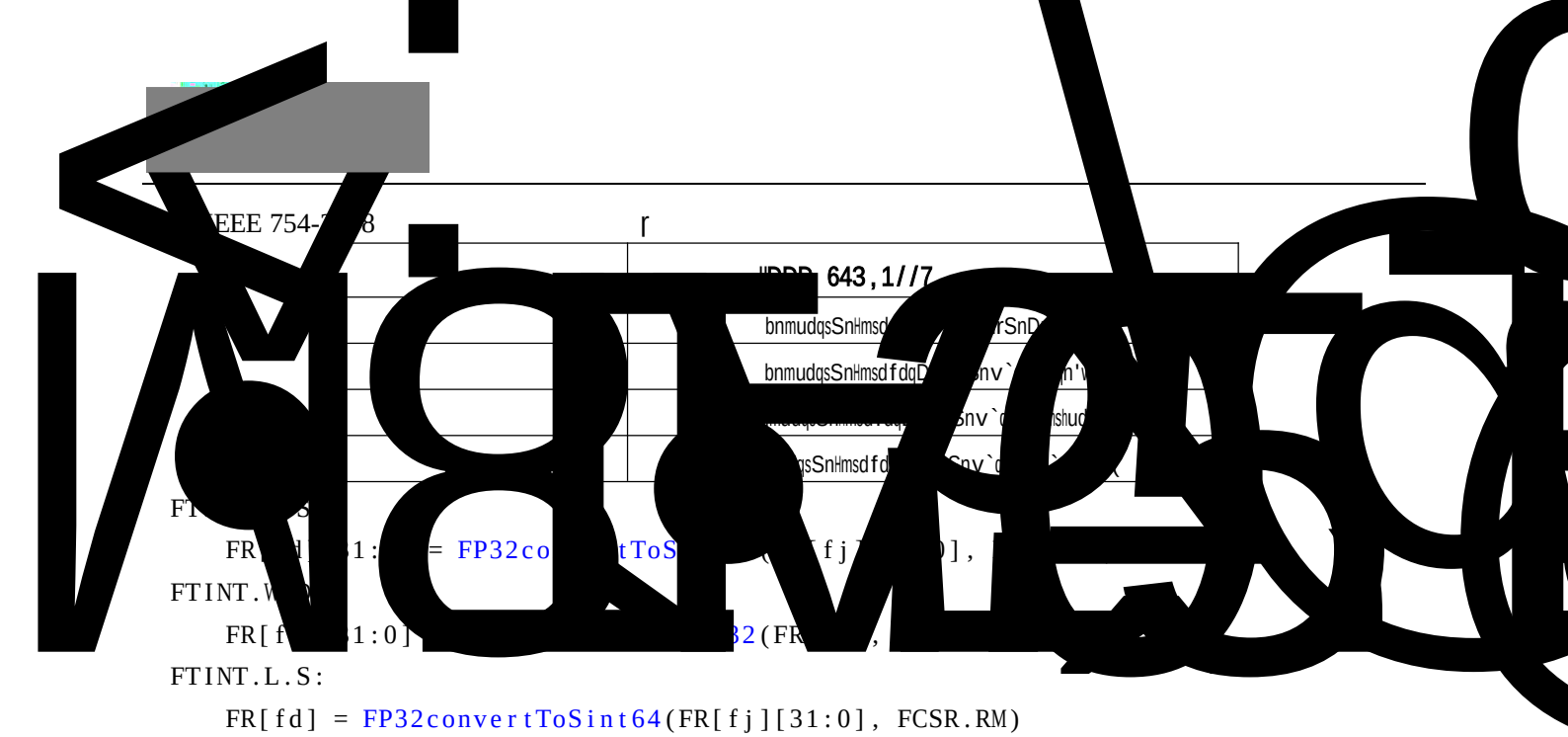
3.2.3.2 FFINT.{S/D}.{W/L}, FTINT.{W/L}.{S/D}

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

FFINT.{S/D}.{W/L} fj , Ð / K¿ ĭ ʁo ° ش /

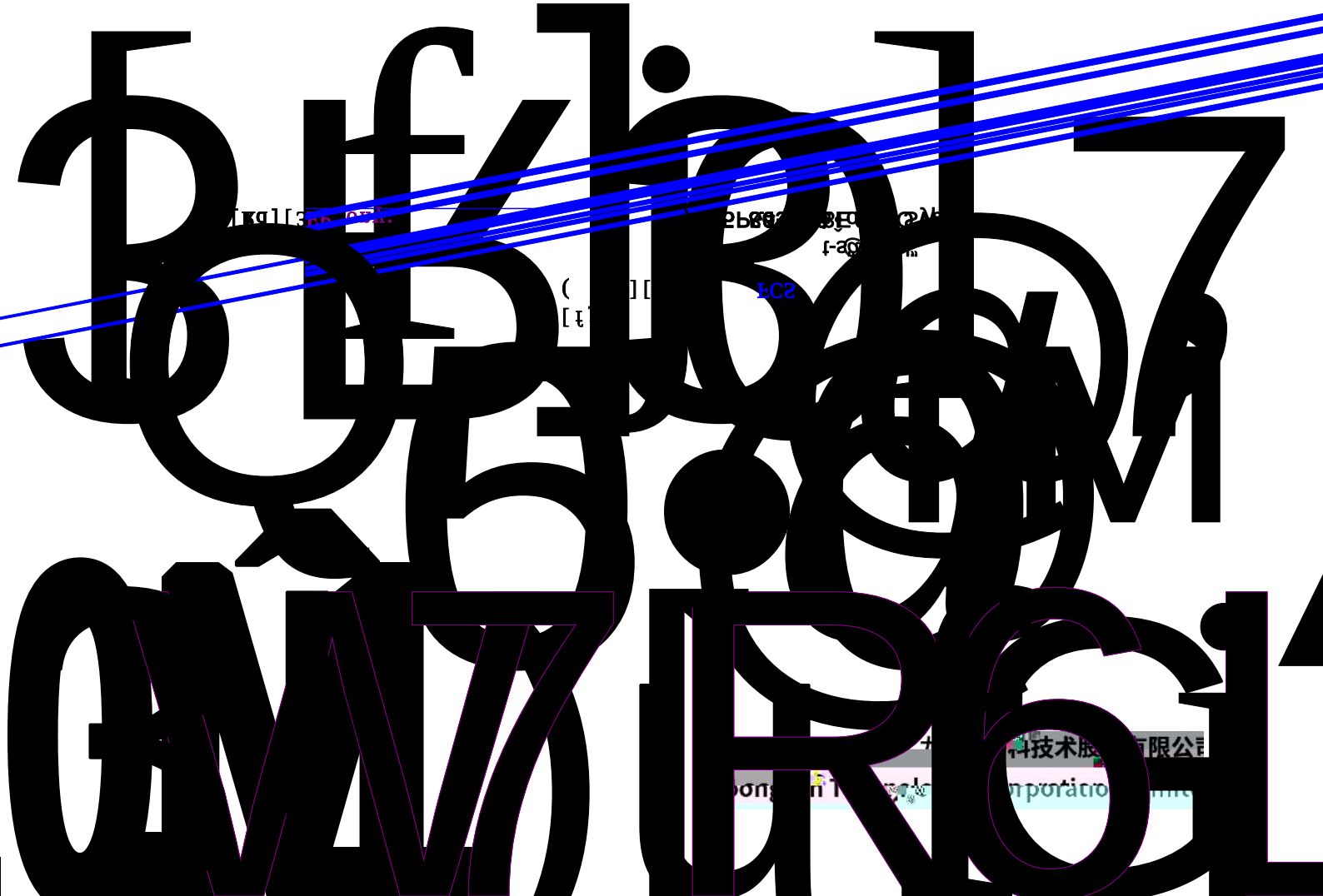
/ "VK¿

K¿



```
FTINT.W.S:
    FR[fd][31:0] = FP32convertToSint32(FR[fj][31:0], FCSR.RM)
FTINT.L.S:
    FR[fd] = FP32convertToSint64(FR[fj][31:0], FCSR.RM)
FTINT.L.D:
    FR[fd] = FP64convertToSint64(FR[fj], FCSR.RM)
```

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



```
FTINTRP.W.S:
    FR[fd][31:0] = FP32convertToSint32(FR[fj][31:0], 2)
FTINTRP.W.D:
    FR[fd][31:0] = FP64convertToSint32(FR[fj], 2)
FTINTRP.L.S:
    FR[fd] = FP32convertToSint64(FR[fj][31:0], 2)
FTINTRP.L.D:
    FR[fd] = FP64convertToSint64(FR[fj], 2)
```

FTINTRZ.{W/L}.{S/D} fj / /
/ fd Ņ ņ ŗ

```
FTINTRZ.W.S:
    FR[fd][31:0] = FP32convertToSint32(FR[fj][31:0], 1)
FTINTRZ.W.D:
    FR[fd] = FP64convertToSint32(FR[fj], 1)
FTINTRZ.L.S:
    FR[fd][31:0] = FP32convertToSint64(FR[fj][31:0], 1)
FTINTRZ.L.D:
    FR[fd] = FP64convertToSint64(FR[fj], 1)
```

FTINTRNE.{W/L}.{S/D} fj / /
/ fd Ņ ņ ŗ

```
FTINTRNE.W.S:
    FR[fd][31:0] = FP32convertToSint32(FR[fj][31:0], 0)
FTINTRNE.W.D:
    FR[fd][31:0] = FP64convertToSint32(FR[fj], 0)
FTINTRNE.L.S:
    FR[fd] = FP32convertToSint64(FR[fj][31:0], 0)
FTINTRNE.L.D:
    FR[fd] = FP64convertToSint64(FR[fj], 0)
```

IEEE 754-2008 ŗ

	GCCC 532+0..6
DRGLRPLC,yU-J{,yQ-B{	aml tcprRmGlrcecpCv_arRgcqRmCtcl&v'
DRGLRPX,yU-J{,yQ-B{	aml tcprRmGlrcecpCv_arRmu_pbXcpm&v'
DRGLRPN,yU-J{,yQ-B{	aml tcprRmGlrcecpCv_arRmu_pbNmqgrgtc&v'
DRGLRPK,yU-J{,yQ-B{	aml tcprRmGlrcecpCv_arRmu_pbLce_rgtc&v'

frint fd, frint.d fj

FRINT.{S/D} / /

/ fd r

┌ MOVGR2FCSR FCSR0 Cause Enables 1
FCSR1 Enables ┘ FCSR2 Cause Cause Enables 1 MOVGR2FCSR
┌

```
MOVGR2FCSR
    FCSR[fcsr] = GR[rj][31:0]
```

MOVFCSR2GR fcsr 32 rd ┌

```
MOVFCSR2GR
    GR[rd] = SignExtend(FCSR[fcsr], GRLEN)
```

fcsr ┌

3.2.4.6 MOVFR2CF, MOVCF2FR

movfr2cf cd, fj

movcf2fr fd, cj

MOVFR2CF fj cd ┌

```
MOVFR2CF
    CFR[cd] = FR[fj][0]
```

MOVCF2FR cj fd fd 0 ┌

```
MOVCF2FR
    FR[fd] = ZeroExtend(CFR[cj], 64)
```

3.2.4.7 MOVGR2CF, MOVCF2GR

movgr2cf cd, rj

movcf2gr rd, cj

MOVGR2CF rj cd ┌

```
MOVGR2CF
    CFR[cd] = GR[rj][0]
```

MOVCF2GR cj rd rd 0 ┌

```
MOVCF2GR
    GR[rd] = ZeroExtend(CFR[cj], GRLEN)
```

3.2.5

3.2.5.1 BCEQZ, BCNEZ

bceqz cj, offs21

bcnez cj, offs21

BCEQZ cj 0 г

BCNEZ cj 0 г

```
BCEQZ:
    if CFR[cj]==0 :
        PC = PC + SignExtend({offs21, 2'b0}, GRLEN)
BCNEZ:
    if CFR[cj]!=0 :
        PC = PC + SignExtend({offs21, 2'b0}, GRLEN)
```

21 offs21 2

PC г

offs21<<2 г

3.2.6

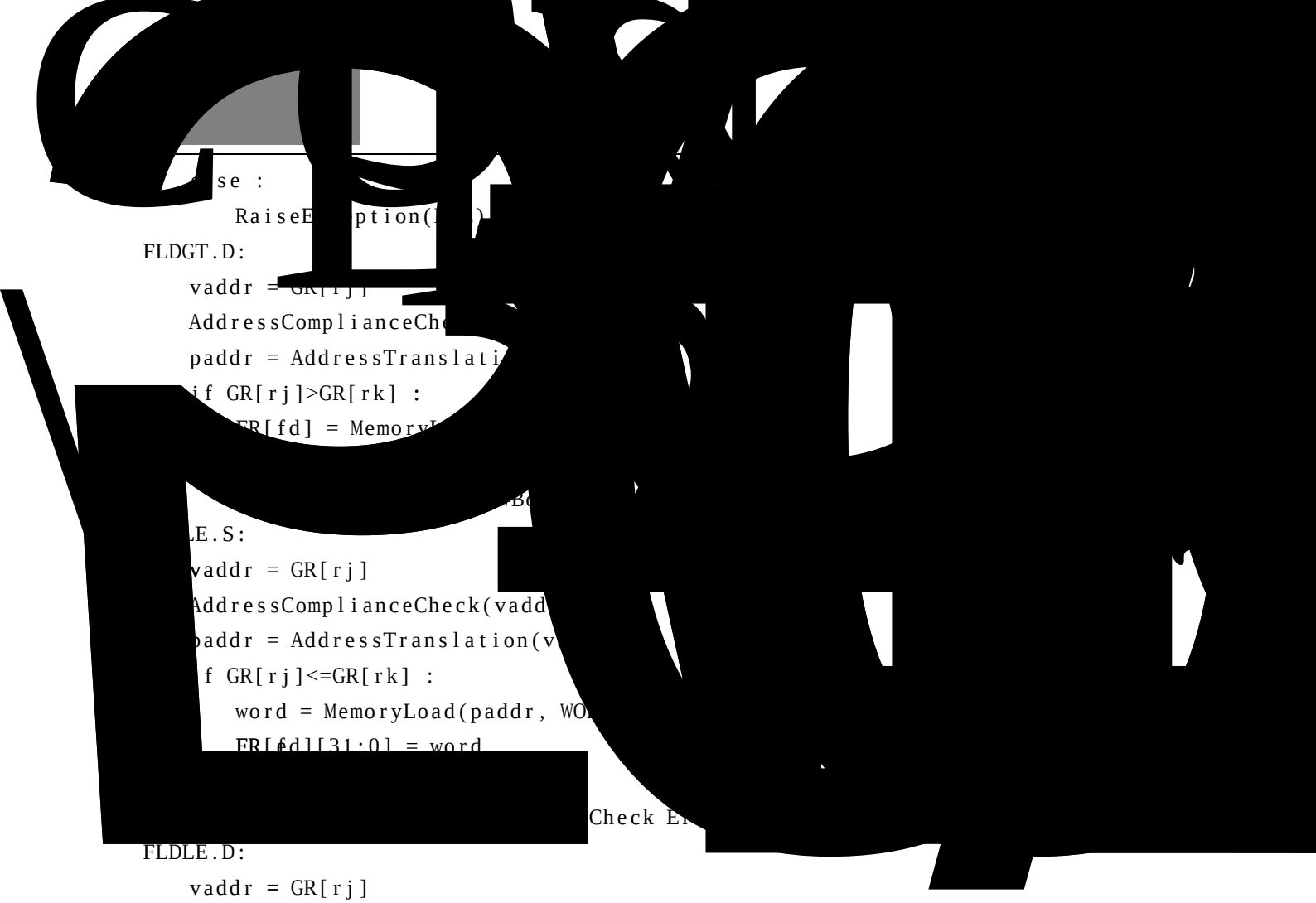
3.2.6.1 FLD.{S/D}, FST.{S/D}

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

fsu ش

```
    r
FSTX.S:
    vaddr = GR[rj] + GR[rk]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    MemoryStore(FR[fd][31:0], paddr, WORD)
FSTX.D:
    vaddr = GR[rj] + GR[rk]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    MemoryStore(FR[fd][63:0], paddr, DOUBLEWORD)
```

e e



```
    else :
        RaiseException(
FLDGT.D:
    vaddr = GR[rj]
    AddressComplianceCh
    paddr = AddressTransla
    if GR[rj]>GR[rk] :
        FR[fd] = Memory
                          B
LE.S:
    vaddr = GR[rj]
    AddressComplianceCheck(vadd
    paddr = AddressTranslation(v
    if GR[rj]<=GR[rk] :
        word = MemoryLoad(paddr, WO
        FR[fd][31:0] = word
                                  Check E
FLDLE.D:
    vaddr = GR[rj]
    AddressComplianceCheck(vaddr)
    paddr = AddressTranslation(vaddr)
    if GR[rj]<=GR[rk] :
        FR[fd] = MemoryLoad(paddr, DOUBLEWORD)
    else :
        RaiseException(BCE) #Bound Check Error Exception(BC
```

FST{GT/LE}.{S/D}

4.1

4 Privilege LeVel PLV PLV0~PLV3 ┌

CSR.CRMD PLV ┌

PLV0

┌PLV1~PLV3 MMU

┌

Linux PLV0 PLV3 ┌

4.2

PLV0 ┌ CSR.MISC

RPCNTL1/RPCNTL2/RPCNTL3 1 PLV1/PLV2/PLV3 CSRRD

┌

4.2 ✗

CSR CSR 0

r CSRWR CSRXCHG

CSR CSR r

4.2.2 IOCSR

4.2.2.1 IOCSR{RD/WR}.{B/H/W/D}

iocsrrd.b rd, rj

iocsrrd.h rd, rj

iocsrrd.w rd, rj

iocsrrd.d rd, rj

iocsrwr.b rd, rj

iocsrwr.h rd, rj

iocsrwr.w rd, rj

iocsrwr.d rd, rj

IOCSR{RD/WR}.{B/H/W/D} IOCSR r

IOCSR r IOCSR

r IOCSR r IOCSR{RD/WR}.{B/H/W/D}

IOCSR rj r

IOCSRRD.{B/H/W} IOCSR / /

rd IOCSRRD.D IOCSR

rd r

IOCSRWR.{B/H/W/D} rd [7:0]/[15:0]/[31:0]/[63:0] IOCSR

r

IOCSRRD.D IOCSRWR.D LA64 r

IOCSR r IOCSR

r

4.2.3 Cache

4.2.3.1 CACOP

cacop code, rj, si12

CACOP Cache Cache r

12 12 COP VA

Cach r

COP Cache Ca code de[2:0]

Ca code r

code[2:0] Cache CPUCFG Cache r CPUCFG10=0x02C3D

code[2:0]=0 Cache code[2:0]=1 Cache code[2:0]=2

Cache code[2:0]=3 Cache r

code[4:3]=0 Cache Store Tag Cache tag 0 r Cache

1<<Way 1<<Index Cache Cache 1<<Offset

D) ªÐÐ% Ÿ P †D‡–'µð Cache VA[Way-1:0] VA[Index+Offset-1:Offset] Cache r

code[4:3]=1 Cache Index Invalidate / Invalidate and Writeback r

r Cache r

Cache Cache r

Cache r Cache Cache

Cache r

code[4:3]=2 Cache Hit Invalidate / Invalidate and Writeback r

Cache r CACOP VA

load Cache Cache r

CACOP TLB r CACOP

Cache r

Ca code[4:3]=3 Cache r

4.2.4 TLB

4.2.4.1 TLBSRCH

B O

I

4.2.4.2 TLBRD

tlbrd

LVZ TLBRD ┌

CSR.TLBIDX Index TLB ┌ TLB

TLB CSR.TLBEHI ┤ CSR.TLBELO0 ┤ CSR.TLBELO1 CSR.TLBIDX.PS

CSR.TLBIDX NE 0

STLB MTLB

4.2.4.5 TLBCLR

tlbclr

TLB CSR TLB TLB

LVZ TLBCLR

CSR.TLBIDX.Index MTLB (STLB) TLBCLR MTLB G=0 ASID CSR.ASID.ASID

CSR.TLBIDX.Index STLB (STLB) TLBCLR STLB CSR.TLBIDX.Index G=0 ASID CSR.ASID.ASID

4.2.4.6 TLBFLUSH

tlbflush

TLB CSR TLB TLB

LVZ TLBFLUSH

CSR.TLBIDX.Index .

op op r

no	
/w/	r
/w0	r no</ r
/w1	F<0 r
/w2	F</ r
/w3	F</ @RHC @RHC r
/w4	F</ @RHC @RHC U@ U@ r
/w5	F<0 @RHC @RHC U@ U@ r

4.2.5

LDPTE TLB PTE

CSR ŗ

rj [6] 1 rj HugePage ŗ

5

5.1

0~2PALEN-1 ┌

LA32 PALEN 36 32 ┌

LA64 PALEN 60 ┌

CPUCFG 0x1 PALEN PALEN ┌

5.2

┌ PLV0 LA32 2^{32}

LA64 2^{64} ┌ LA64 2^{64}

┌

┌

MMU ┌

CSR.CRMD DA=1 PG=0 MMU ┌

[PALEN-1:0] 0

┌ ┌

┌

CSR.CRMD DA=0 PG=1 MMU ┌

Ṇ ṇ Ṇ ṇ ┌

┌

5.2.1 5.4 ┌ LA64

[63:VALEN]

[VALEN-1] [VALEN-1] ADE ┌

┌

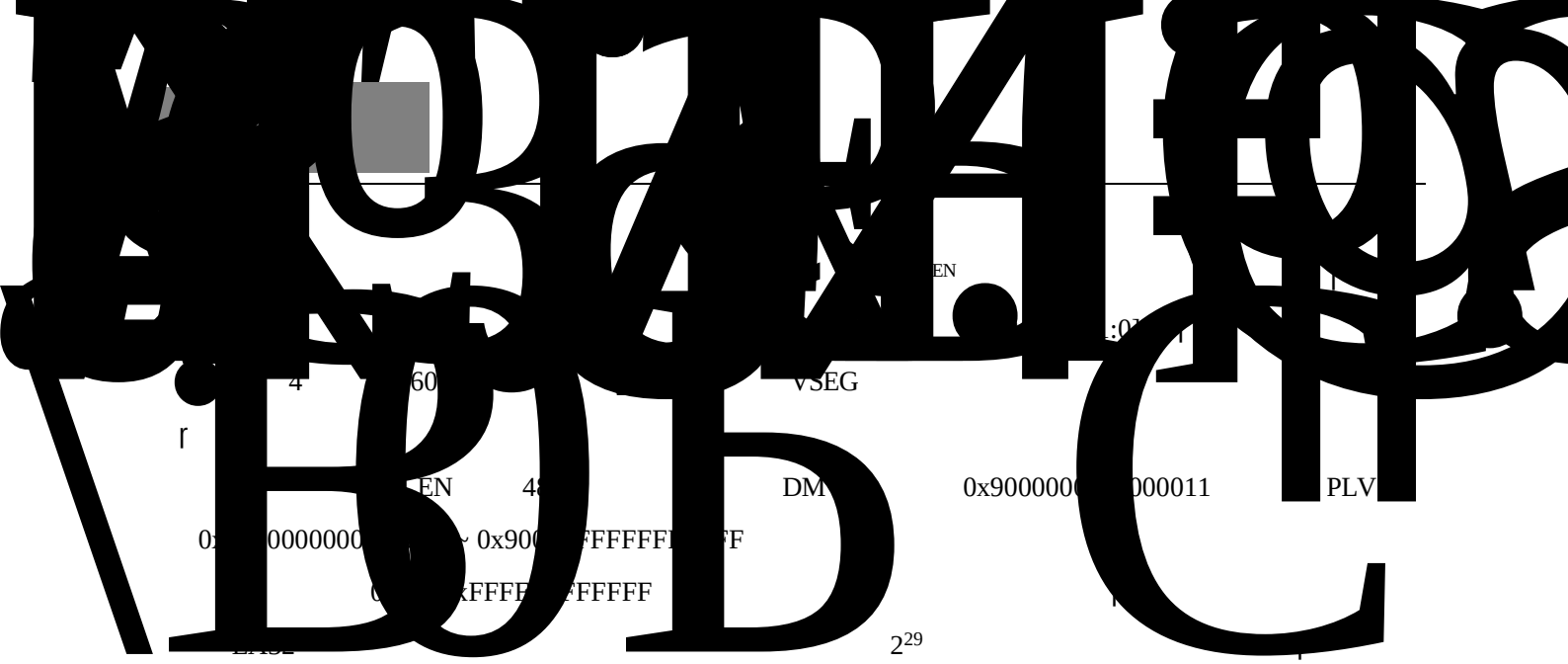
5.2.1

MMU

┌ load/store

load/store ┌

CSR.DMW0~CSR.DMW3 ┌



EN

4 60 VSEG

r

EN 4 DM 0x900000 000011 PLV

0x 0000000 ~ 0x900 FFFFFF FF

0 xFFFF FFFFFF

LA32 2^{29}

[28:0]

r 3 [31:29] [31:29]

r

DMW0 0x80000011 PLV0 0x80000000 ~ 0x9FFFFFFF

0x0 ~ 0x1FFFFFFF r

5.2.2 LA64 32

LA32 LA64

LA64 32 r CSR.MISC

VA32L1/VA32L2/VA32L3 1 PLV1/PLV2/PLV3 32 r

32 64 BL ┘JIRL

PCADD 32 64 r

5.2.3 LA64

LA64 r

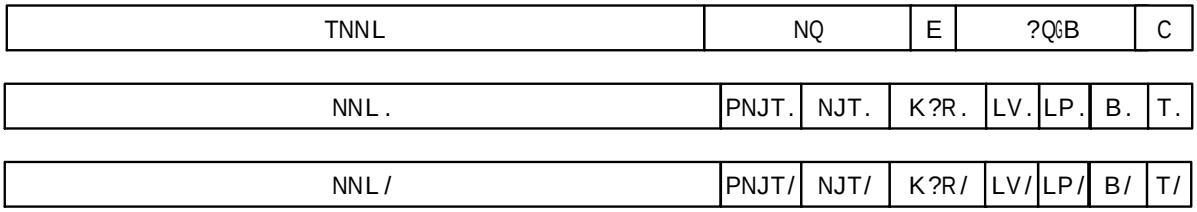
CSR.RVACFG RDVA 1~8

(VALEN-RDVA) r VALEN=48 RDVA 8

[63:40] [39] r

5.3

load/store	C



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

┌ 16KB PS=14 ┌

(VPPN) (VALEN-13) ┌

TLB /2

TLB ┌ TLB

┌

(V) 1 ┌ 1 ┌

(D) 1 ┌ 1 ┌

(NR) 1 ┌ 1 load ┌

LA64 ┌

(NX) 1 ┌ 1 ┌

LA64 ┌

(MAT) 2 ┌ ┌

3,1 ┌

PLV 2 ┌ ┌ RPLV=0

PLV RPLV=1 PLV ┌

RPLV 1 ┌ ┌

PLV ┌ LA64 ┌

(PPN) (PALEN-12) ┌ 4KB TLB PPN [PS-1:12]

┌

5.4.3 TLB

TLB ┌ 1.0x TLB TLB

┌

5.4.3.1 TLB

TLB TLB

TLB

┌ TLB

TLB TLB

TLB ┌ ┘ CSR

TLB CSR ┌ TLB

CSR.CRMD DA 1 PG 0

TLB TLB ┌

TLB CSR CSR TLB

CSR.TLBRERA.ISTLBR 1 ┌

load load TLB V=0

┌

store store TLB V=0

┌

TLB V=0

┌

TLB V=1

┌ RPLV=0 CSR.CRMD.PLV

PLV RPLV=1 CSR.CRMD.PLV PLV ┌

store TLB V=1

D 0 ┌

load TLB V=1

NR 1 ┌

TLB V=1

NX 1 ┌

5.4.3.2 TLB

TLB TLB ┘ ┘ ┘ TLB ┘

┌ 4.2.4

5.4.3.3 TLB CSR

TLB CSR TLB TLB

TLB r

BADV

TLBEHI

TLBELO0

TLBELO1

TLBIDX

ASID

STLBPS

PGDL

PGDH

PGD

PWCL

PWCH

TLBRENTRY

TLBRERA

TLBRBADV

TLBREHI

TLBRELO0

TLBRELO1

TLBRPRMD

TLBRSAVE

CSR TLB 5,2 CSR r

5.4.3.4 TLB

TLB Ņ INVTLB 0, r0, r0” r

5.4.4 TLB

TLB r STLB MTLB

STLB MTLB r

va:

```
elif (mtlb_found==1) :
    found_v = mfound_v
    found_d = mfound_d
    found_nr = mfound_nr
    found_nx = mfound_nx
    found_mat = mfound_mat
    found_plv = mfound_plv
    found_rplv = mfound_rplv
    found_ppn = mfound_ppn
    found_ps = mfound_ps
else :
    SignalException(TLBR)       #  TLB

if (found_v==0) :
    case mem_type :
        FETCH : SignalException(PIF)      #
        LOAD  : SignalException(PIL)      #  load
        STORE : SignalException(PIS)      #  store
elif (mem_type==FETCH) and (found_nx==1) :
    SignalException(PNX)        #
elif ((found_rplv==0) and (plv > found_plv)) or
     ((found_rplv==1) and (plv!= found_plv)) :
    SignalException(PPI)        #
elif (mem_type==LOAD) and (found_nr==1) :
    SignalException(PNR)        #
elif (mem_type==STORE) and (found_d==0)
     and ((plv==3) or (CSR.MISC[16+plv]==0)) :  #
    SignalException(PME)        #
else :
    pa = {found_ppn[PALEN-13:found_ps-12], va[found_ps-1:0]}
    mat = found_mat
```

■

5.4.5

LDDIR LDPTE

5-2 r

ṆP ṇ ṆW ṇ ŗ

TLB ŗ

TLB

LDPTE TLB ŗ

16KB 32MB ṆLDPTE rj, 0 ṇ ṆLDPTE rj, 1 ṇ TLB 16MB ŗ

TLB TLBR PGD

ŗ

6

6.1

ŗ ŗ

6.1.1

13 1 IPI 1 TI
1 PMI 8 HWI0~HWI7 2 SWI0~SWI1 ŗ
ŗ
CSR.ESTAT.IS[12] ŗ
ŗ 0
ŗ CSR.ESTAT.IS[11] ŗ
CSR.TICLR TI 1 ŗ
ŗ
[63] 1 ŗ
CSR.ESTAT.IS[10] ŗ [63] 0
ŗ
ŗ 8 HWI[7:0]
CSR.ESTAT.IS[9:2] ŗ
CSR CSR.ESTAT.IS[1:0] 1
0 ŗ
CSR.ESTAT.IS Int Number ŗ SWI0 0
SWI1 1 ňň IPI 12 ŗ

6.1.2

ŗ IPI
TI ňň SWI0 ŗ

6.1.3

r 4,0,/ r

64 r 0 SWI0 64 1 SWI1

65 ň ň r

6.1.4

CSR.ESTAT.IS CSR.ECFG.LIE

13 int_vec r CSR.CRMD.IE=1 int_vec 0

ł· ł· r

4,0,1 r

6.2

6.2.1

256

r CSR.MSGIS0~CSR.MSGIS3 64 CSR 0 255

r 256

r

6.2.2

256 255

254 ň ň 0 r

CSR.MSGIE.PT r

r

6.2.3

Ņ ņ CSR.EENTRY

Ņ ņ $2^{(CSR.ECFG.VS+2)}$ ¥ 78 0x4E r

Watch TLB

r

r

(ADE) >

0 IE 0 DA 1 PG 0

Watch CSR.CRMD WE CSR.TLBRPRMD PWE CSR.CRMD WE 0

PC [GRLEN-1:2] CSR.TLBRERA ERA CSR.TLBRERA IsTLBR 1

PC CSR.TLBRBADV [VALEN-1:13] CSR.TLBREHI VPPN

CSR.TLBRENTTRY ┌

ERTN TLB CSR.TLBRPRMD PPLV ┤ PIE CSR.CRMD PLV ┤ IE

Watch CSR.TLBRPRMD PWE CSR.CRMD WE CSR.CRMD DA 0 PG 1 CSR.TLBRERA IsTLBR 0

CSR.TLBRERA ┌

6.3.5

CSR.CRMD PLV ┤IE ┤DA ┤PG ┤DATF ┤DATM CSR.MERRCTL PPLV ┤PIE ┤PDA ┤ PPG ┤ PDATF ┤ PDATM CSR.CRMD PLV 0 IE 0 DA 1 PG 0 DATF 0 DATM 0

Watch CSR.CRMD WE CSR.MERRCTL PWE CSR.CRMD WE 0

PC CSR.MERRERA CSR.MERRCTL IsMERR 1

CSR.MERRINFO1 CSR.MERRINFO2

CSR.MERRENTRY ┌

ERTN CSR.MERRCTL PPLV ┤ PIE ┤ PDA ┤ PPG ┤ PDATF ┤ PDATM CSR.CRMD PLV ┤ IE ┤ DA ┤ PG ┤ DATF ┤ DATM

Watch CSR.MERRCTL PWE CSR.CRMD WE CSR.MERRCTL IsMERR 0

CSR.MERRERA ┌

6.4

r r

PC 0x1C000000 r MMU

0x1C000000 r

CSR.CRMD PLV=0 IE=0 DA=1 PG=0 DATF=0 DATM=0 WE=0

CSR.EUEN FPUen ɻ VPUen ɻ XVPUen ɻ BTUen 0

CSR.MISC 0

CSR.ECFG VS LIE 0

CSR.ESTAT IS[1:0] 0

CSR.RVACFG RDVA=0

CSR.TCFG En=0

CSR.LLBCTL KLO=0

CSR.TLBRERA IsTLBR=0

CSR.ERRCTL IsMERR=0

CSR.DMW PLV0~PLV3 0

CSR.PMCFG EvCode 0

CSR 0

CSR 0

CSR.DBG DS=0 r

r

TLB Cache

r

/w2/*m '/ q m q 04(			R@UDm
/w3/			SHC
/w30			SBEF
/w31			SU@K
/w32			BMSB
/w33			SHBKQ
/w5/	KKAhs		KKABSK
/w7/		0	HLOBSK0
/w70		1	HLOBSK1
/w77	SKA		SKAQDMSQX
/w78	SKA		SKAQA@CU
/w7@	SKA		SKAQDQ@
/w7A	SKA		SKAQR@UD
/w7B	SKA	/	SKAQDKN/
/w7C	SKA	0	SKAQDKN0
/w7D	SKA		SKAQDGH
/w7E	SKA		SKAQOQLC
/w8/			LDQQBSK
/w80		0	LDQQHMEN0
/w81		1	LDQQHMEN1
/w82			LDQQDMSQX
/w83			LDQQDQ@
/w84			LDQQR@UD
/w87			BS@F
/w`/		/	LRFHR/
/w`0		0	LRFHR0
/w`1		1	LRFHR1
/w`2		2	LRFHR2
/w`3			LRFHQ
/w`4			LRFHD
/w07/*m '/ q m q 2(		m	CLVm
/w1//*1m '/ q m q 20(		m	OLBEFm
/w1/0*1m '/ q m q 20(		m	OLBMSm

/w2//	kn`	qd		LVOB
/w2/0	kn`	qd		LVOR
/w20/*7m '/ǫmǫ02(	kn`	qd	m	LVOmBEF0
/w200*7m '/ǫmǫ02(	kn`	qd	m	LVOmBEF1
/w201*7m '/ǫmǫ02(	kn`	qd	m	LVOmBEF2
/w202*7m '/ǫmǫ02(	kn`	qd	m	LVOmBEF3
/w27/				EVOB
/w270				EVOR
/w28/*7m '/ǫmǫ02(			m 0	EVOmBEF0
/w280*7m '/ǫmǫ02(			m 1	EVOmBEF1
/w281*7m '/ǫmǫ02(			m 2	EVOmBEF2

S

\XB w„

7.2.2 LA32　LA64

32　　LA32　LA64　r

LA64　　CSR　　64

32　　r　　LA32　LA64　　r

7.2.3

CSR　　CSR

r

CSRWR　CSRXCHG　　rd

r

7.3

r

7.4

7.4.1　　CRMD

ł　　ł

r

7-2

09/	OKU	QV	r　　/}2 r　/　　2 r /　　r DQSM BRQ-DQQBSK-HrLDQQ<0　　BRQ-DQQBSK　OOKU BRQ-SKAQDQ@-HrSKAQ<0　　BRQ-SKAQOQLC　OOKU BRQ-OQLC　OOKU　　r

r

/ r

0 r

DQSM

1 HD QV BRQ-DQQBSK-HrLDQQ<0 BRQ-DQQBSK OHD

BRQ-SKAQDQ@-HrSKAQ<0 BRQ-SKAQOQLC OHD

BRQ-OQLC OHD r

r

SKA 0 r

DQSM

2 C@ QV BRQ-DQQBSK-HrLDQQ<0 BRQ-DQQBSK OC@

BRQ-SKAQDQ@-HrSKAQ<0 / r

C@ OF / ɻ 0 0 ɻ /

r

3 OF QV

8	VD	QV	r / r DQSM BRQ-DQQBSK-HrLDQQ<0 BRQ-DQQBSK 0VD BRQ-SKAQDQ@-HrSKAQ<0 BRQ-SKAQ0QLC 0VD BRQ-0QLC 0VD r
2090/	/	Q/	r / r

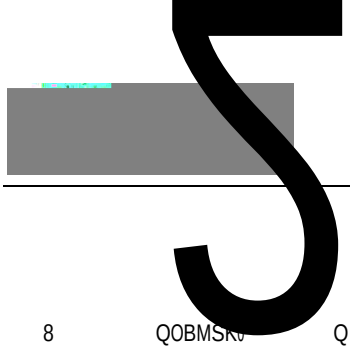
7.4.2 PRMD

TLB

r

7-4

/ EOD QV r / 2-1



		OKU0	r 0 OKU0	
8	QOBMSK0	QV	BRQQC OBMS	
		HOD r		
		OKU1	r 0 OKU1	
0/	QOBMSK1	QV	BRQQC OBMS	
		HOD r		
		OKU2	r 0 OKU2	
00	QOBMSK2	QV	BRQQC OBMS	
		HOD r		
		OKU/	kn`c.rsnqd 0 r 0	
			r	
01	@KBK/	QV	kn`c.rsnqd	
		0 r		
		OKU0	kn`c.rsnqd 0 r 0	
			r	
02	@KBK0	QV	kn`c.rsnqd	
		0 r		
		OKU1		
03	@KBK1	QV		

7.4.5 ECFG

r

7-6

019/	KHD	QV	r BRQ-DRS@S HR 02 r
04902	/	Q/	r / r
07905	UR	QV	r UR</ r UR </ 1UR r SKA UR r
20908	/	Q/	r / r

7.4.6 ESTAT

r

7-7

09/	HRZ09/\	QV	r / 0 RVH/ RVH0 r 0 / r
0191	HRZ0191\	Q	r 0 r0 HOH 0 SH 0 OLH 7 GVH/}GVH6 r r
02	/	Q/	r / r
03	/	Q/	BOTBEF-0-LRF^HMSZahs15\</ / r
03	LrfHms	Q	BOTBEF-0-LRF^HMSZahs15\</ 0 r
04	/	Q/	r / r
10905	Dbncd	Q	r SKA 6-7 Dbncd r
2/911	DrtaBncd	Q	r SKA 6-7 DrtaBncd r

20	/	Q/	r / r

7-8

Dbncd	DrtaBncd		
/w/		HMS	BRQ-DBEF-UR</ r
/w0		OHK	kn`c
/w1		OHR	rsnqd
/w2		OHE	
/w3		OLD	
/w4		OMQ	
/w5		OMW	
/w6		OOH	
/w7	/	@CDE	
	0	@CDL	
/w8		@KD	
/w@		ABD	
/wA		RXR	
/wB		AQJ	
/wC		HMD	
/wD		HOD	
/wE		EOC	
/w0/		RWC	017
/w00		@RWC	145
/w01	/	EOD	
	0	UEOD	
/w02	/	VODE	
	0	VODL	kn`c.rsnqd
/w03		ASC	
/w04		ASD	
/w05		FROQ	
/w06		GUB	
/w07	/	FBRB	BRQ
	0	FBGB	BRQ

7.4.9 BADI

r INT ɻ

CSR GCHC ɻ MERR r

7-11

209/	Hmrs	Q	r

7.4.10 EENTRY

r

7-12

009/ / Q /

7-13

29/	QAhsr	QV	/ r /}7 r r 7 r
2093	/	Q/	r / r

7.4.12 CPUID

r

7-14

79/	BnqdHC	Q	r r r / r
2098	/	Q/	r / r

7.4.13 1 PRCFG1

r

7-15 1

29/	R@UDMt l	Q	R@UD r
0093	Sh l dqAhsr	Q	Sh l dq 0 r
03901	URL`w	Q	BRQ-DBEF-UR r
20904	/	Q/	r / r

7.4.14 2 PRCFG2

r

7-16 2

FQKDM,09/	OR@UK	Q	SKA O`fd Rhyd r h 0 1^{h} r

7.4.15 3 PRCFG3

r

29/	SKASxod	Q	SKA / SKA 0 SKA LSKA 1 SKA LSKA * SKA RSKA r
0093	LSKADmsqhdr	Q	SKASxod</ / SKASxod<0 1 SKA 0 r
08901	RSKAV`xr	Q	SKASxod</ 0 / SKASxod<1 SKA 0 r
1491/	RSKARdsr	Q	SKASxod</ 0 / SKASxod<1 SKA 1RSKARdsr r
20915	/	Q/	r / r

7.4.16 L SAVE

^ r

r

q D I L XU Y S

2092	/	

7.4.18 1 IMPCT

ZæBY%Ø#J x T*ŸB±4÷.¯MÔ.Az/(Eùt E.2¨!7 â/p ÞÞ

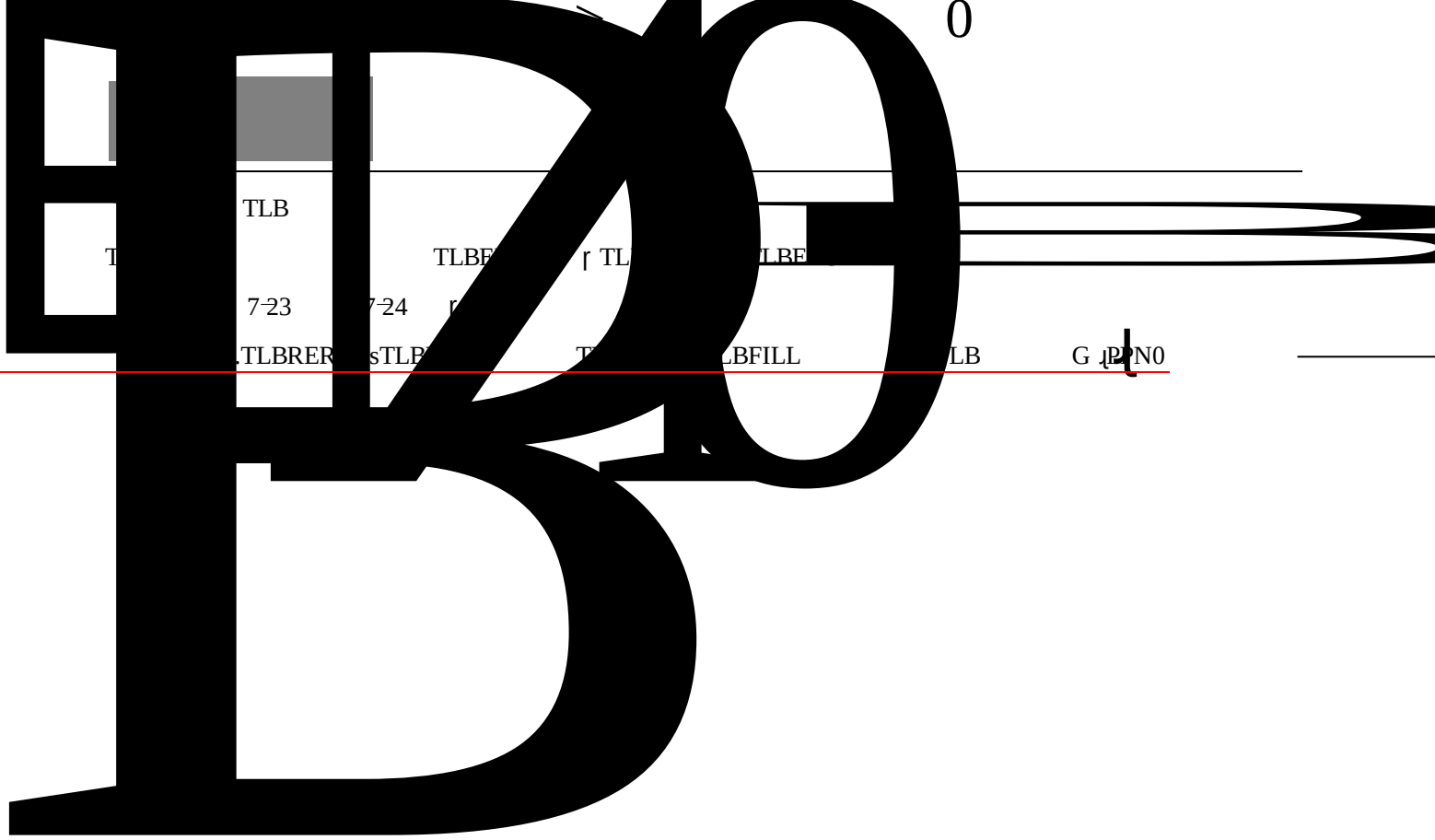
0 SKA SKA / SKA

SKA r

SKARQBG / 0 r

20 MD QV SKAQC SKA D r

SKAVQ SKAEHKK BRQ-SKAQDQ@-HrSKAQ</



291	OKU	QV	OKU	r
493	L@S	QV		

1391/	Chq1^a`rd	QV	1 r
1894	Chq1^vhcsg	QV	1 r / r
2092/	OSDVhcsg	QV	r / 53 0 017 1 145 2 401 r

7.5.9 PWCH

CSR.PWCL r

r 5.4.5 r

LA64 r

7-30

49/	Chq2^a`rd	QV	2 r
0095	Chq2^vhcsg	QV	2 r / r
0601	Chq3^a`rd	QV	3 r
1207	Chq3^vhcsg	QV	3 r / r
13	/	/	BOTBEF-1-GOSVZahs13\</ / r
13	GOSV^Dm	QV	BOTBEF-1-GOSVZahs13\<0 0 / r
2014	/	Q/	r / r

7.5.10 STLB STLBPS

STLB r

7-31 STLB

49/	OR	QV	RSKA 1 r 05JA OR</wD r
2095	/	Q/	r / r

7.5.11 TLB TLBRENTRY

TLB r TLB

r

0	/	Q	/ r
FQKDM,091	OB	QV	SKA OB ZFQKDM,091\ r DQSM SKA HrSKAQ<0 BRQ-DQQBSK-HrLDQQ</ / r

7.5.14 TLB TLBRSAVE

r r

TLB SAVE TLB

TLB r

7-36 TLB

FQKDM,09/	C`s`	QV	r BRQ r

7.5.15 TLB TLBRELO0, TLBRELO1

TLBRELO0/1 TLB CSR.TLBRERA.IsTLBR=1 TLB

TLB r TLBRELO0/1

TLBELO0/1 r

TLBRELO0/1 CSR.TLBRERA.IsTLBR=1 TLBELO0/1 r

CSR.TLBRERA.IsTLBR TLBRD TLBELO0/1 r

CSR.TLBRERA.IsTLBR LDPTE TLBRELO0/1 r

7-37 TLB LA64

/	U	QV	U r
0	C	QV	C r
291	OKU	QV	OKU r
493	L@S	QV	L@S r
5	F	QV	F r SKAEHKK SKAVQ SKADKN/ SKADKN0 F 0 SKA F 0 r
0096	/	Q	/ r
O@KDM,0901	OOM	QV	OOM r

5/9O@KDM	/	Q	/ r
50	MQ	QV	MQ r
51	MW	QV	MW r
52	QOKU	QV	QOKU r QOKU</ OKU QOKU<0 OKU r

7-38 TLB LA32

/	U	QV	U r
0	C	QV	C r
291	OKU	QV	OKU r
493	L@S	QV	L@S r
5	F	QV ó^	F r SKAEHKK SKA^Q SKADKN/ SKADKN0 F 0 SKA F 0 r
6	/	Q	/ r
O@KDM , 497	OOM	QV	OOM r
209O@KDM , 3	/	Q	/ r O@KDM<25 r

7.5.16 TLB TLBREHI

TLBREHI TLB CSR.TLBRERA.IsTLBR=1 TLB TLB r TLBREHI TLBEHI r

TLBREHI CSR.TLBRERA.IsTLBR=1

7.6.2 TCFG

r TimeVal

r

7-45

7.6.5 TICLR

0 1 r

7-48

/	BKQ	V0	ahs 0 r / r
2090	/	Q/	r / r

7.7

7.7.1 MERRCTL

CSR r CSR

MERRERA ERRSAVE MERRCTL r

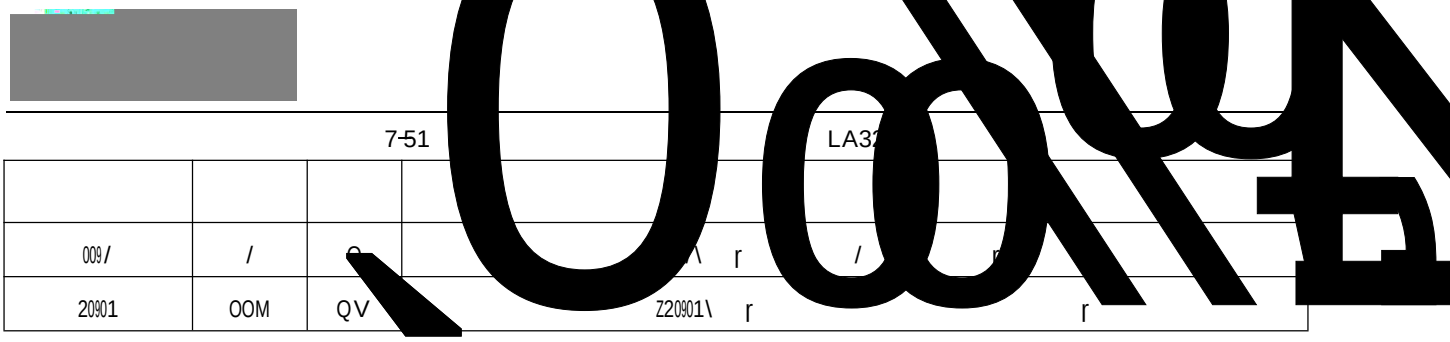
7-49

/	HrLDQQ	Q	0 r 0 r 0 DQSM / r BRQ 0 DQSM BRQ-BQLC OOKU ↓OHD DQSM BRQ-DQQDQ@ r
0	Qdo`hq`akd	Q	0 r
291	OOKU	QV	BRQ-BQLC OKU r HrLDQQ 0 DQSM BRQ-BQLC OKU r
3	OHD	QV	BRQ-BQLC HD r HrLDQQ 0 DQSM BRQ-BQLC HD r
4	/	Q	/ r

5 OVD QV HrLDQQ 0 DQSM

BRQ-BQLC VD r

6 OC@



7-51 LA32

009/	/		
20901	OOM	QV	Z20901\ r r

7.7.4 MERRERA

r

7-52

FQKDM,09/	OB	QV	OB r DQSM BRQ-DQQBSK-Hr LDQQ<0 r

7.7.5 MERRSAVE

r r

SAVE

r

7-53

FQKDM,09/	C`s`	QV	r BRQ r

7.8

r r

1 32 r CPUCFG.6.PMNUM[bit7:4]

ς

[redacted]

maddr mbyten

CSR.CRMD.WE=0

bytemask 8 MWPCFG1

VAddr MWPCFG3 Size r

7-59 load/store bytemask

LVOBEF2-Rhyd	LVOBEF0-U@ccqZ1?/\							
	/	0	1	2	3	4	5	6
/a/ /	/w/ /							
/a/0	/wE/				/w/E			
/a0/	/wEB		/wE2		/wBE		/w2E	
/a00	/wED	/wEC	/wEA	/wE6	/wDE	/wCE	/wAE	/w6E

L

7-60 load/store 1 I· ^

FQKDM,09/	U@ccq	QV	kn`c.rsnqd r

t !! B 3 I·

7-61 load/store 2

FQKDM,09/	L`rj	QV	kn`c.rsnqd r h /ǫh;FQKDM 0 h r

7-62 load/store 3

0 kn`c.rsnqd r Ṇ ṇ

/	CRNmkx	QV		r
			BRQ-CAF-CR<0	r
				r
0	OKU/	QV	OKU/	r 0 I· I· / I· I· r
1	OKU0	QV	OKU0	r 0 I· I· / I· I· r
2	OKU1	QV	OKU1	طى طى سخي طى

7-63 load/store 4

89/	@RHC	QV	@RHC
0490/	/	Q	/ r
12905	/	Q	/ r
20913	/	Q	/ r

7.9.4

7.9.e

7-69 4

89/	@RHC	QV	@RHC
0490/	/	Q	/ r
12905	/	Q	/ r
20913	/	Q	/ r

7.10

7.10.1 DBG

7-70

/	CR	Q	0 r 0 r 0 DQSM / r
690	CQdu	Q	r 0 r
7	CDH	Q	0 CDH r
8	CBK	Q	0 CBK r
0/	CEV	Q	0 CEV r
00	CLV	Q	0 kn`c.rsnqd CLV r
04901	/	Q/	/
10905	Dbncd	Q	r 6-7 SKA /w6 /wB /wD r
20911	/	Q/	/

7.10.2 DERA

7-71

52%	OB	QV	OB r BRQ-CAF-CR<0 DQSM r

7.10.3 DSAVE

r r

m" SAVE

r

7-72

7.11

7.11.1 0 3 MSGIS0

7-73 0

52%	HR	Q	/ 52 / 52 0 r

7-74 1

52%	HR	Q	/ 52 53 016 0 r

7-75 2

52%	HR	Q	/ 52 017 080 0 r

7-76 3

52%	HR	Q	/ 52 081 144 0 r

32 CSR LA64 CSR 64

8 A

8.1

r

Ņ'd ņ “##'d” “##'d” ##

Ņ'b ņ “ d

annkd`m	
ahs	
hmsdfdq	
ahsr'M(	M
YdqnDwsdmc'____+M(	M
RhfmDwsdmc'____+M(	M
hrRM`M'____(	rhfm`khmf M`M SQTD E@KRD
hrPM`M'____(	pthds M`M SQTD E@KRD
Rhfm`kDwbdoshnm'____(	
"	
<	

8-2

ZL9M\	M L
zMzL||	L M
zM+ L+ ň|	M+ L+ ň

8-3

*	
,	
)	
.	
$	
))	

8-4

<<	
<	
=	
;	
=<	
;<	

8-5

%	
{	
]	
}	
; ;	
==	
===	

8-6

`mc	
nq	
mns	

8-7

8-7

))	
}	
)+ .+ $	
*+ ,	
; ;+ ==+ ===	
%	
]+ {	
=+ ;+ =<+ ;<	
<<+ <	
mns	
`mc+ nq	

8.2

r

8.2.5 1

```
{bits(N) } CLO(bits(N) x):
    cnt = 0
    for i in range(N) :
        if x[N-1-i]==1'b0 :
            return cnt
        else :
            cnt = cnt + 1
```

8.2.6 0

```
{bits(N) } CLZ(bits(N) x):
    cnt = 0
    for i in range(N) :
        if x[N-1-i]==1'b1 :
            return cnt
        else :
            cnt = cnt + 1
```

8.2.7 1

```
{bits(N) } CTO(bits(N) x):
    cnt = 0
    for i in range(N) :
        if x[i]==1'b0 :
            return cnt
        else :
            cnt = cnt + 1
```

8.2.8 0

```
{bits(N) } CTZ(bits(N) x):
    cnt = 0
    for i in range(N) :
        if x[i]==1'b1 :
            return cnt
        else :
            cnt = cnt + 1
```

8.2.13

```
{bits(32) } FP64convertToSint32(bits(64) x, bits(2) rm):
    case {rm} of:
        {2'd0}: return Sint32_convertToIntegerExactTiesToEven(x)
        {2'd1}: return Sint32_convertToIntegerExactTowardZero(x)
        {2'd2}: return Sint32_convertToIntegerExactTowardPositive(x)
        {2'd3}: return Sint32_convertToIntegerExactTowardNegative(x)
```

8.2.14

```
{bits(64) } FP64convertToSint64(bits(64) x, bits(2) rm):
    case {rm} of:
        {2'd0}: return Sint64_convertToIntegerExactTiesToEven(x)
        {2'd1}: return Sint64_convertToIntegerExactTowardZero(x)
        {2'd2}: return Sint64_convertToIntegerExactTowardPositive(x)
        {2'd3}: return Sint64_convertToIntegerExactTowardNegative(x)
```

8.2.15

```
{bits(32) } FP32_roundToInteger(bits(N) x):
    return FP32_roundToIntegralExact(x)
```

8.2.16

```
{bits(64) } FP64_roundToInteger(bits(N) x):
    return FP64_roundToIntegralExact(x)
```

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
CLO.W	rd, rj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0			rj					rd		
CLZ.W	rd, rj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1			rj					rd		
CTO.W	rd, rj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0			rj					rd		
CTZ.W	rd, rj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1			rj					rd		
CLO.D	rd, rj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0			rj					rd		
CLZ.D	rd, rj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1			rj					rd		
CTO.D	rd, rj	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0			rj					rd		
CTZ.D																																	

		3 3 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1	1 1 1 0 0 0 0 0 0 0 0 0 0 0
		1 0 9 8 7 6 5 4 3 2 1 0 9 8 7 6 5 4 3	2 1 0 9 8 7 6 5 4 3 2 1 0
SYSCALL	code	0 0 0 0 0 0 0 0 0 0 0 1 0	-ﾍ

		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
FCLASS.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	0	1	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	1	0	1	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	1	0	1	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	1	0	1	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	1	0	1	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	1	0	1	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	1	0	1	0	0	0	1	1	0	1	0			fj					fd		
FRECIPE.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	1	1	0	1			fj					fd		
FRECIPE.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	1	1	1	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	9	8	7	6	5	4	3	2	1	0
FTINTRNE.L.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	1	0	0	1			fj					fd		
FTINTRNE.L.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	0	1	1	1	0	1	0			fj					fd		
FTINT.W.S	fd, fj	0	0	0	0	0	0	0	1	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	1	1	0	1	1	0	0	0	0	1	0			fj					fd		
FTINT.L.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0	1	0	0	1			fj					fd		
FTINT.L.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0			fj					fd		
FFINT.S.W	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	1	0	0			fj					fd		
FFINT.S.L	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0	0	1	1	0			fj					fd		
FFINT.D.W	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0	1	0	0	0			fj					fd		
FFINT.D.L	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0	1	0	0	1	0	1	0			fj					fd		
FRINT.S	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	1	0	0	0	1			fj					fd		
FRINT.D	fd, fj	0	0	0	0	0	0	0	1	0	0	0	1	1	1	1	0	0	1	0	0	1	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	2								rj					rd		

Œ ƒ P ₽ o g

ə! ƨ

0

		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
FST.S	fd, rj, si12	0	0	1	0	1	0	1	1	0	1						si12									rj					fd		
FLD.D	fd, rj, o(, אס																																

		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
AMSWAP.D	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	0	1			rk					rj					rd		
AMADD.W	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	1	0			rk					rj					rd		
AMADD.D	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	0	0	1	1			rk					rj					rd		
AMAND.W	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	0	1	0	0			rk					rj					rd		
AMAND.D	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	0	1	0	1			rk					rj					rd		
AMOR.W	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	0	1	1	0			rk					rj					rd		
AMOR.D	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	0	1	1	1			rk					rj					rd		
AMXOR.W	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	1	0	0	0			rk					rj					rd		
AMXOR.D	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	1	0	0	1			rk					rj					rd		
AMMAX.W	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	1	0	1	0			rk					rj					rd		
AMMAX.D	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	1	0	1	1			rk					rj					rd		
AMMIN.W	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	1	1	0	0			rk					rj					rd		
AMMIN.D	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	1	1	0	1			rk					rj					rd		
AMMAX.WU	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	1	1	1	0			rk					rj					rd		
AMMAX.DU	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	0	1	1	1	1			rk					rj					rd		
AMMIN.WU	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	1	0	0	0	0			rk					rj					rd		
AMMIN.DU	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	1	0	0	0	1			rk					rj					rd		
AMSWAP_DB.W	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	1	0	0	1	0			rk					rj					rd		
AMSWAP_DB.D	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	1	0	0	1	1			rk					rj					rd		
AMADD_DB.W	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	0	0			rk					rj					rd		
AMADD_DB.D	rd, rk, rj	0	0	1	1	1	0	0	0	0	1	1	0	1	0	1	0	1			rk					= ↾ ∉=