Lx Assembler

Version 1.0
March 1, 2000

Comments beginning with the strings '**#APP**' or **#NO_APP**' also have a special interpretation and should not be used.

"Directives" on page 15. Any statement that begins with a letter is part of an Lx instruction bundle. The bundle terminator '`;;`' is also a statement.

A statement is terminated by a newline character. Newlines within character constants are an exception – they do not terminate a statements. It is an error to end any statement with end-of-file; the last character of any input file must be a newline. Statements may be written on multiple lines provided that each line other

Example

```
$r0.1       general purpose register 1 of cluster 0
$r1.3       general purpose register 3 of cluster 1
$b0.3       branch register 3 of cluster 0
```

2.9 Assembler Directives

```
.__longjmp          .__setjmp
._longjmp           ._setjmp
.call               .comment
.endp               .entry
.import             .proc
.return             .sversion
.type
```

2.9.8 Restrictions on Standard Directives

A number of restrictions currently apply to standard directives. In most cases, this is a temporary situation, but we have not yet had an opportunity to examine the implications of the directives.

The following directives are not currently permitted in bss or text sections

```
.ascii, .asciz, .byte, .data1, .data2, .data4, .data8,  .double,
.float, .hword, .longd, 196dd, quad, .real4,
.single, .short, .stringd, word
```

The following directives are not currently permitted in the text section

```
.align, .skip, .space
```

The following directives are currently ignored.

```
as -o my-object-file.o mumble.s
as -omy-object-file.o mumble.s
```

3.1.2 Input and Output Files

We use the phrase

When the resulting bits are to be interpreted as a 2's complement number we write `sign_extend(foo[h:l])` to denote the 2's complement number obtained by copying the sign bit of `foo[h:l]`(ha(n)1


xor
0
0 0 0 0 0 1 1 0 1
dest
src2
src1

add

Syntax

(0) **add** dest = src1, src2
(1) **add** idest = src1, isrc2

Operands

dest : GR destination
src1 : GR source 1

addcg

Syntax

(0) **addcg** dest, bdest = src1, src2, scond

Operands

dest : GR destination
bdest : BR destination
src1 : GR source 1
src2 : GR source 2
scond : BR select condition or carry

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

andc

Syntax

(0) **andc** dest = src1, src2
(1) **andc** idest = src1, isrc2

Operands

dest : GR destination
src16(t5( )]TJ 0 0 scn3.012 0 TD.0014 Tc0.0007 Tw[(: G)6.7(R)2.1(s)7.4(our)5.7(c)7.4(e)0t5(1)]TJ5.2289 1.1566 TD.00

br

brf

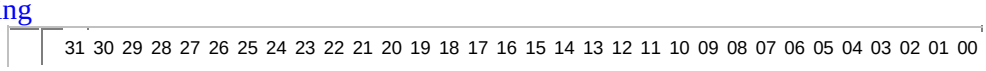
Syntax

(0) **brf** bcond, btarg

Operands

bcond : BR branch condition
btarg : Branch target offset

Encoding



call

Syntax

(0) **call** linkd = btarg
(1) **call** linkd = link

linkd

cmpeq

Syntax

cmpge

Tc07 Tc07 Tc07 Tc07- Tc07

cmpgeu

Syntax

(0) **cmpgeu** dest = src1, src2
(1) **cmpgeu** idest = src1, isrc2
(2) **cmpgeu** bdest = src1, src2
(3) **cmpgeu** ibdest = src1, isrc2

Operands

cmpgt

Syntax

(0)

cmpltu

Syntax

(0) **cmpltu**

cmpne

ax]TJ3 0 0 scn12.77170 TD( )Tj-TT1081 Tf9.96 0 0 9.96 90.0406 764284 Tm0

goto

Syntax

g(例 T goto

p

imml

Syntax

Operands

Encoding

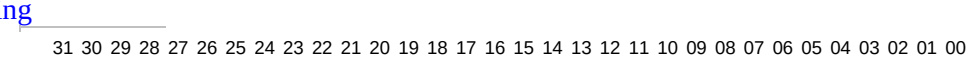
imml Encoding

immr

Syntax

Operands

Encoding



ldb

Syntax

(0) **ldb** idest = isrc2[src1]

ldbu.d

Syntax

(0) ldbu.d idest = isrc2[src1c1c1c1ax

ldh

Syntax

(0) **ldh** idest = isrc2[src1]

ldhu.d

ldw idest = isrc2[src1] (0)

idest : GR destination (imm. format)
isrc2 : Immediate source 2

max

Syntax

(0) **max** dest = src1, src2
(1) **max** idest = src1, isrc2

Operands

dest : GR destination
src1 : GR source 1

maxu

Syntax

minu

Syntax

(0) **minu** dest = src1, src2
(1) **minu** idest = src1, isrc2

Operands

dest : GR destination
src1 : GR source 1
src2 : GR source 2
idest

mulh

Syntax

(0) **mulh** dest = src1, src2
(1) **mulh** idest = src1, isrc2

Operands

dest : GR destination
src1 : GR source 1
src2 : GR source 2
idest

mulhh

Syntax

(0) **mulhh** dest = src1, src2
(1

mulhhu

Syntax

(0) **mulhhu** dest = src1, src2
(1) **mulhhu** idest = src1, isrc2

Operands

dest : GR destination
src1 : GR source 1

mulhs

Syntax

(0) **mulhs** dest = src1, src2
(1) **mulhs** idest = src1, isrc2

mulhu

Syntax

(0) **mulhu** dest = src1, src2
(1

mull

Syntax

(0) **mull** dest = src1, src2

mullh

Syntax

(0) **mullh** dest = src1, src2
(1) **mullh** idest = src1, isrc2

Operands

dest : GR destination
src1 : GR source 1
src2 : GR source 2
idest : GR destination (: GR destina(: G)7.1(R)2.5(des)7.8(ts)0.00sidest :2.0007 6501.904 07 Tc1.9027.1(R)2.5(d

mullhu

Syntax

(0) **mullhu** dest = src1, src2
(1) **mullhu** idest = src1, isrc2

Operands

dest : GR destination

mulll

Syntax

(0) **mulll** dest = src1, src2
(1) **mulll** idest = src1, isrc2

Operands

dest : GR destination
src1

mullu

nandl

Syntax

(0) **nandl**

orl

Syntax

recv

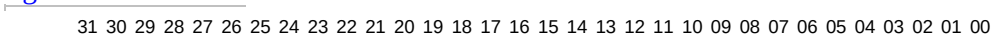
Syntax

(0) **recv** idest = icbus

Operands

idest : GR destination (imm. format)
icbus : Intercluster bus

Encoding



send

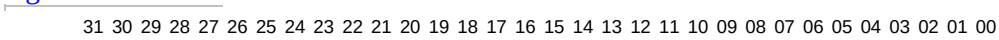
Syntax

(0) **send** icbus = src2

Operands

icbus : Intercluster bus
src2 : GR source 2

Encoding



sh1add

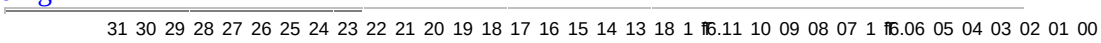
Syntax

(0) **sh1add** dest = src1, src2
(1) **sh1add** idest = src1, isrc2

Operands

dest : GR destination
src1 : GR source 1
src2 : GR source 2
idest : GR destination (imm. format)
isrc2 : Immediate source 2

Encoding



sh2add

Syntax

(0) **sh2add** dest = src1, src2
(1) **sh2add** idest = src1, isrc2

Operands

sh3add

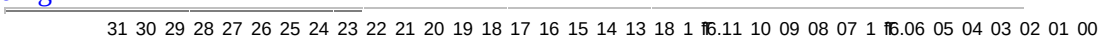
Syntax

(0) **sh3add** dest = src1, src2
(1) **sh3add** idest = src1, isrc2

Operands

dest : GR destination
src1 : GR source 1
src2 : GR source 2
idest : GR destination (imm. format)
isrc2 : Immediate source 2

Encoding



sh4add

Syntax

(0) **sh4add** dest = src1, src2
(1) **sh4add** idest = src1, isrc2

Operands

shl

Syntax

(0) **shl** dest = src1, src2
(1) **shl** idest = src1, isrc2

Operands

shru

Syntax

(0) **shru**

slctf

Syntax

stb

Syntax

(0) **stb** isrc2[src1] = src2

Operands

isrc2 : Immediate source 2

sub

Syntax

(0) **sub** dest = src2, src1
(1) **sub** idest = isrc2, src1

Operands

dest : GR destination
src2 : GR14 ĉ34(our5.57c)-7.74e 2

sxtb

Syntax

(0) **sxtb** dest = src2

Operands

dest : GR destination
src2 : GR source 2

Encoding

sxth

Syntax

(0) **sxth** dest = src2

Operands

dest : GR destination
src2 : GR source 2

Encoding

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

syscall

Syntax

(0)

```
#define TEST_STH(a,t)     Exception = TEST_ADDRESS(a,16,WRITE)#define TEST_STW(a,t)
#define DO_LDB(a,t)     if (!Exception) t = INT8(*a)#define DO_LDB_D(a,t) d ? 0 : 1
                                      #define DO_LDB_U(a,t)   if (!Exception) t = U

                                                          #define DO_LDH(a,t)     if (!
```