PRELIMINARY DATA

The flow chart, *Figure 1*, contains a number of functions which abstract out some the details. Those functions are described in this section. Starting with those used in

3.5 Architectural state

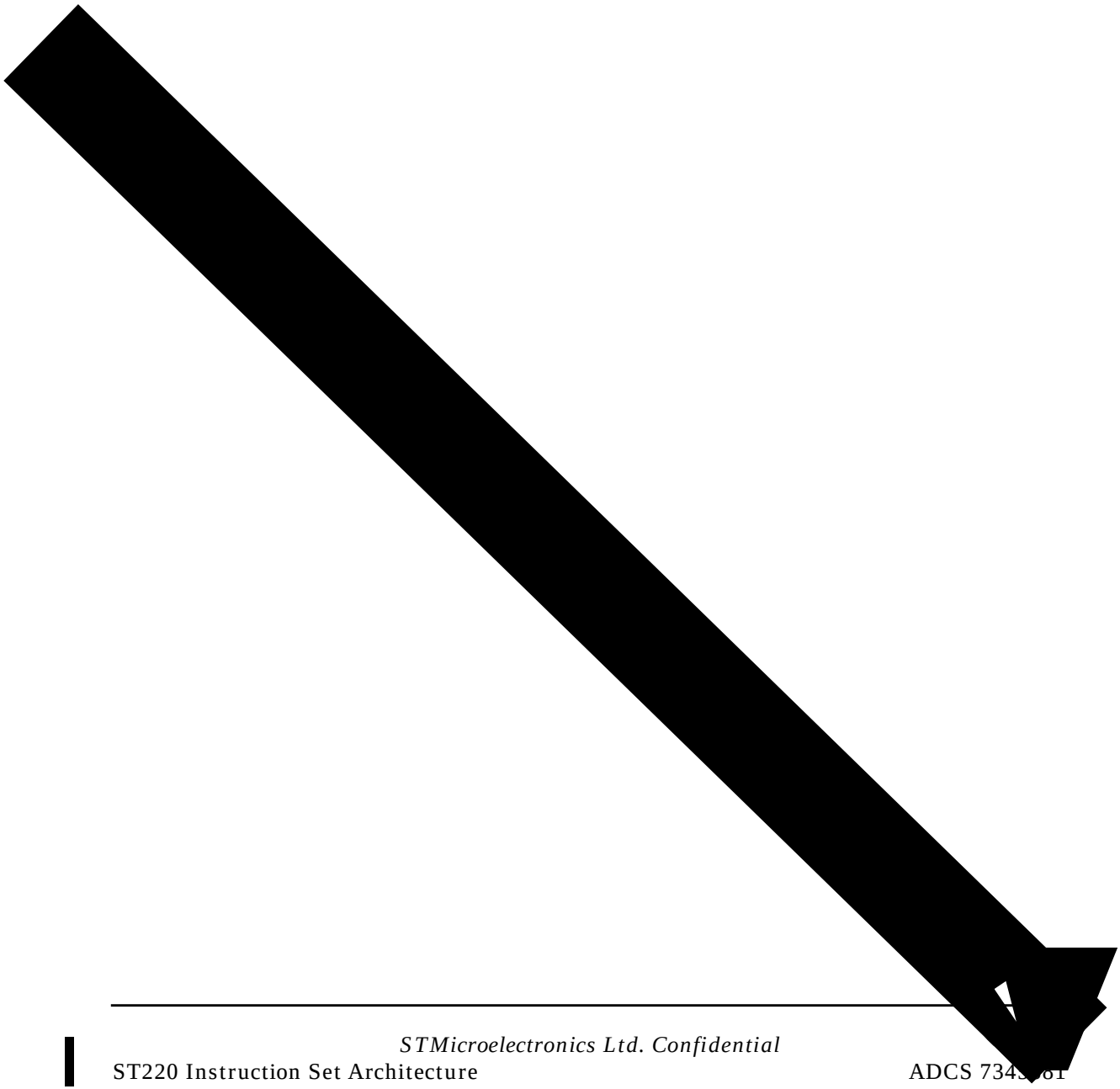
terrupts

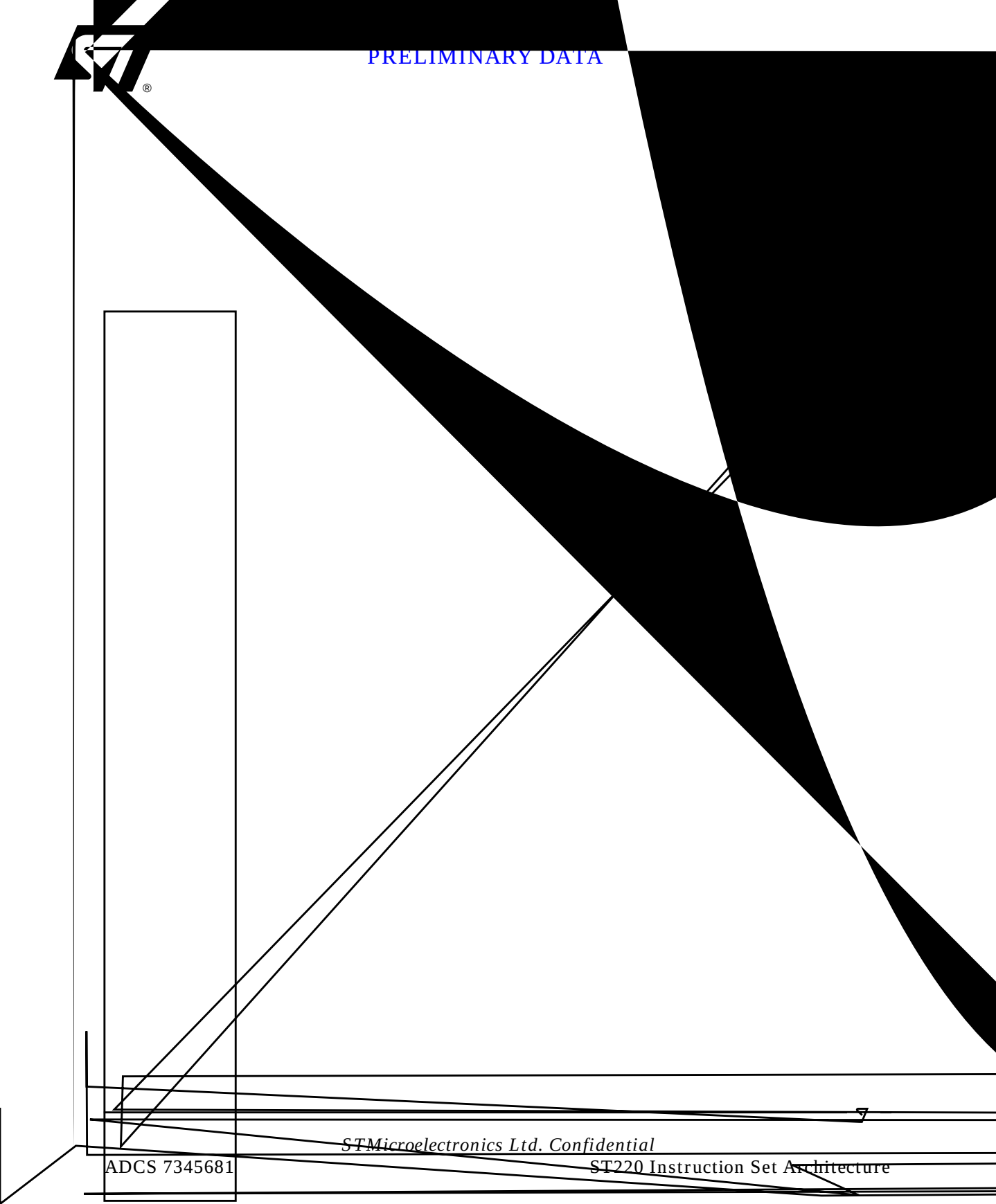
uring interrupts

during debug interrupts

used during debug

PRELIMINARY DATA



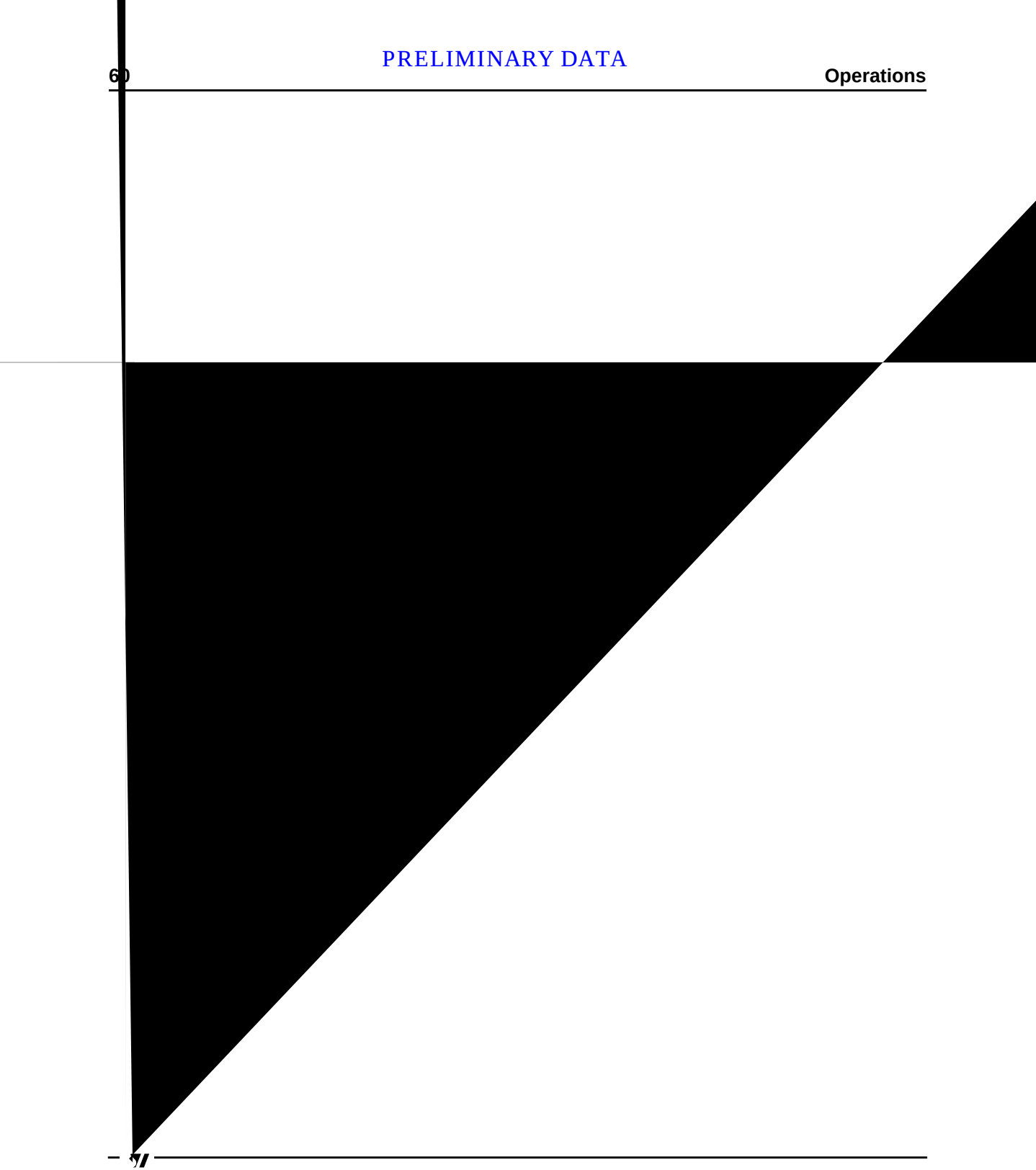


R_{idest} ← Register(result);

The function Register (

Immediate

andc R_{idest} = R



andl $B_{ibdest} = R_{src1}$

bswap Register

bswap R_{dest} = R_{src2}

Semantics:

Description:

Byte swap

Restrictions:

No address/bundle restrictions.

No latency constraints.

Exceptions:

None.

cmpeq Register - Register

cmpeq $R_{dest} = R_{src1}, R_{src2}$

Semantics:

Description:

Test for equality

Restrictions:

No address/bundle restrictions.

No latency constraints.

Exceptions:

None.

cmpne Register - Immediate

cmpne R = R , isrc2

s	00	1	1	0	0001	isrc2	idest	src1

Semantics:

Description: Test for inequality **Restrictions:** No address/bundle restrictions.No latency constrain

goto Link Register

goto LR

Semantics:

Description:

immr

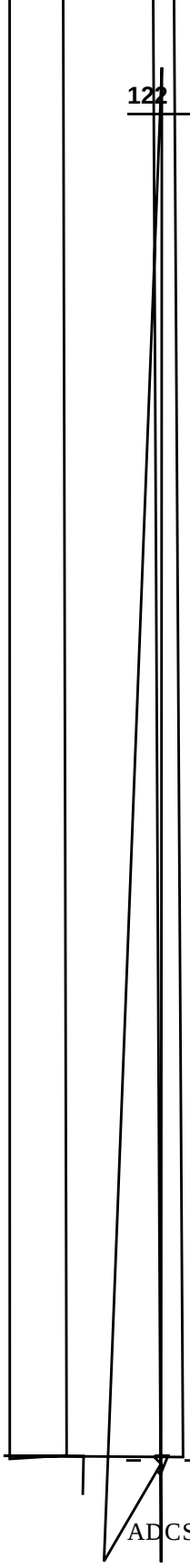
immr imm

Semantics:

Description:

Long immediate for next syllable

Restri BT /.7



maxu Immediate

maxu R_{idest} = R_{src1}, isrc2

Semantics:

Description:

Unsigned maximum

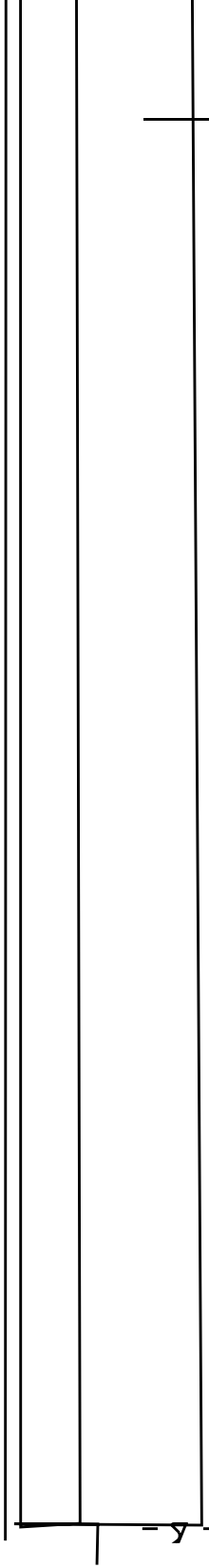
Restrictions:

No address/bundle restrictions.

No latency constraints.

Exceptions:

None.



minu

idest src1

Semantics:

operand-1 = ZeroExtend $_{32}(R_{src1})$;
operand-2 = ZeroExtend $_{32}$(Imm(isrc2));
IF (operan-1 < operan-2)
result = operan-1;Tj0 -i07r70 7.78.t

Description:

Unsigned minimum

Restrictions:

No address/bundle restrictions.

No latency constraints.

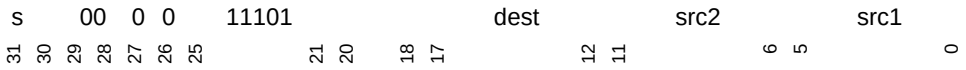
Exceptions:

None.

mulh

mulhh Register

mulhh $R_{dest} = R_{src1}, R_{src2}$



Semantics:

operand1 ← SignExtend

Description:

Upper-half-word by upper-half-word signed multiply

Restrictions:

Must be encoded at odd word addresses.

Cannot write the link register (LR).

This instruction must be followed by 2 bundles before R_{dest} can be read.

Exceptions:

None.

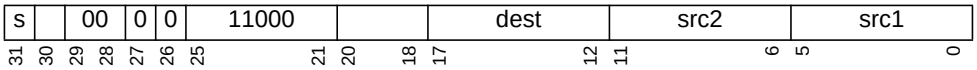
mulhhu Immediate

mulhhu R_{idest} = R

mulhs

mulhu Register

mulhu R_{dest} = R , R_{src2}



Semantics:

operand1 ← $ZeroExtend_{32}(R_{src1})$;
operand2 ← $SignExtend_{32}$

(operand2 >> 16

←

Description:

Word by upper-half-word unsigned multiply

Restrictions:

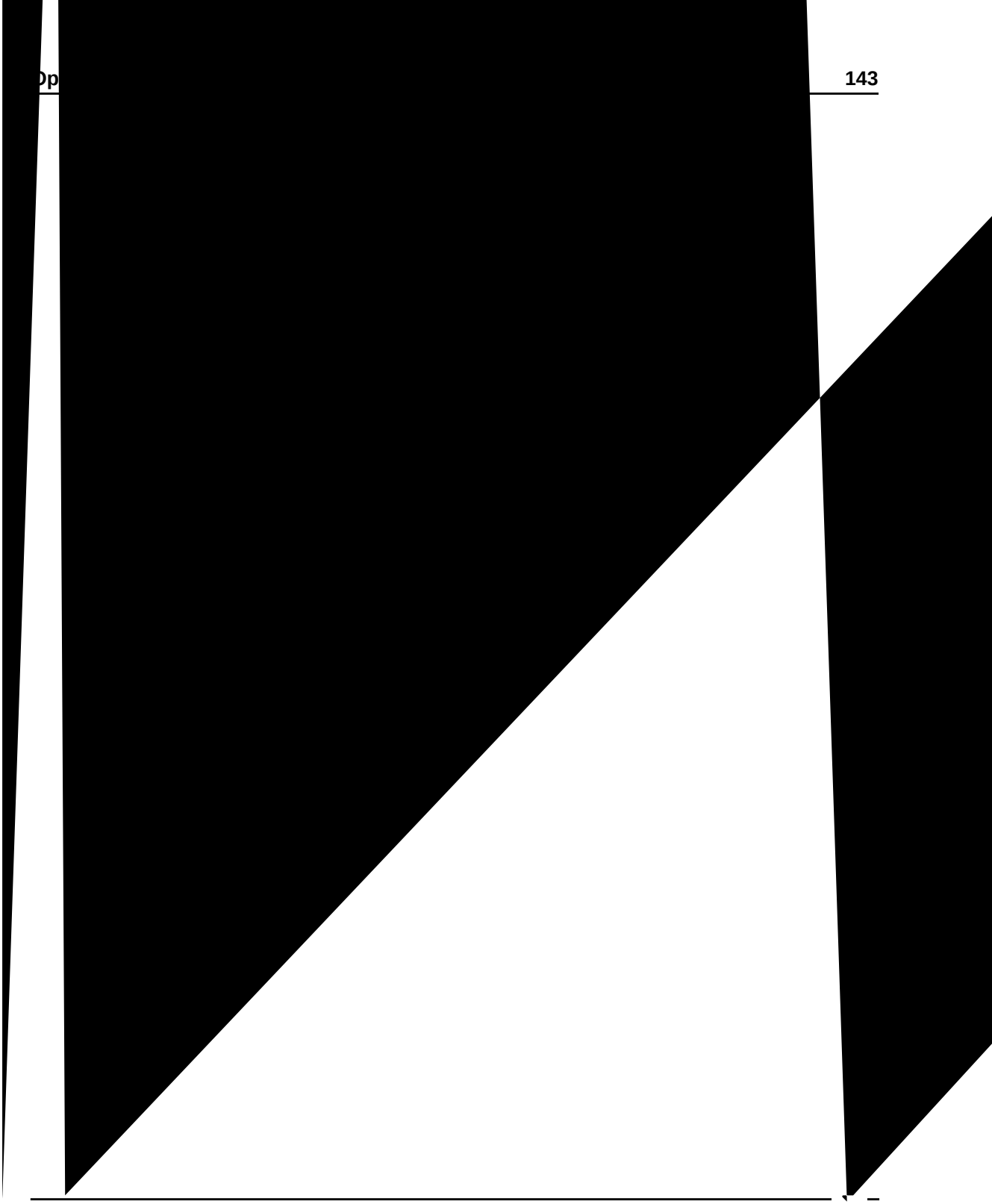
Must be encoded at odd word addresses.

Cannot write the link register (LR).

This instruction must be followed by 2 bundles before R_{dest} can be read.

Exceptions:

None.



[redacted]

nandl Register - Register

nandl $R_{dest} = R_{src1}, R_{src2}$

s		00		0	1	0		1011						dest				src2				src1									

Semantics:

Description:

Logical nand

Restrictions:

No address/bundle restrictions.

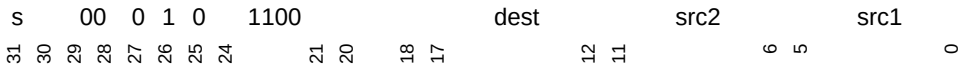
No latency constraints.

Exceptions:

None.

orl Register - Register

orl R_{dest} src2



Semantics:

Description:

Logical or

Restrictions:

No address/bundle restrictions.

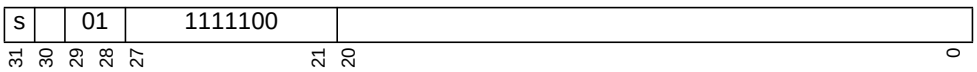
No latency constraints.

Exceptions:

None.

prgins

prgins



Semantics:

```
PurgeIns();
```

Description:

Purge the instruction cache

Restrictions:

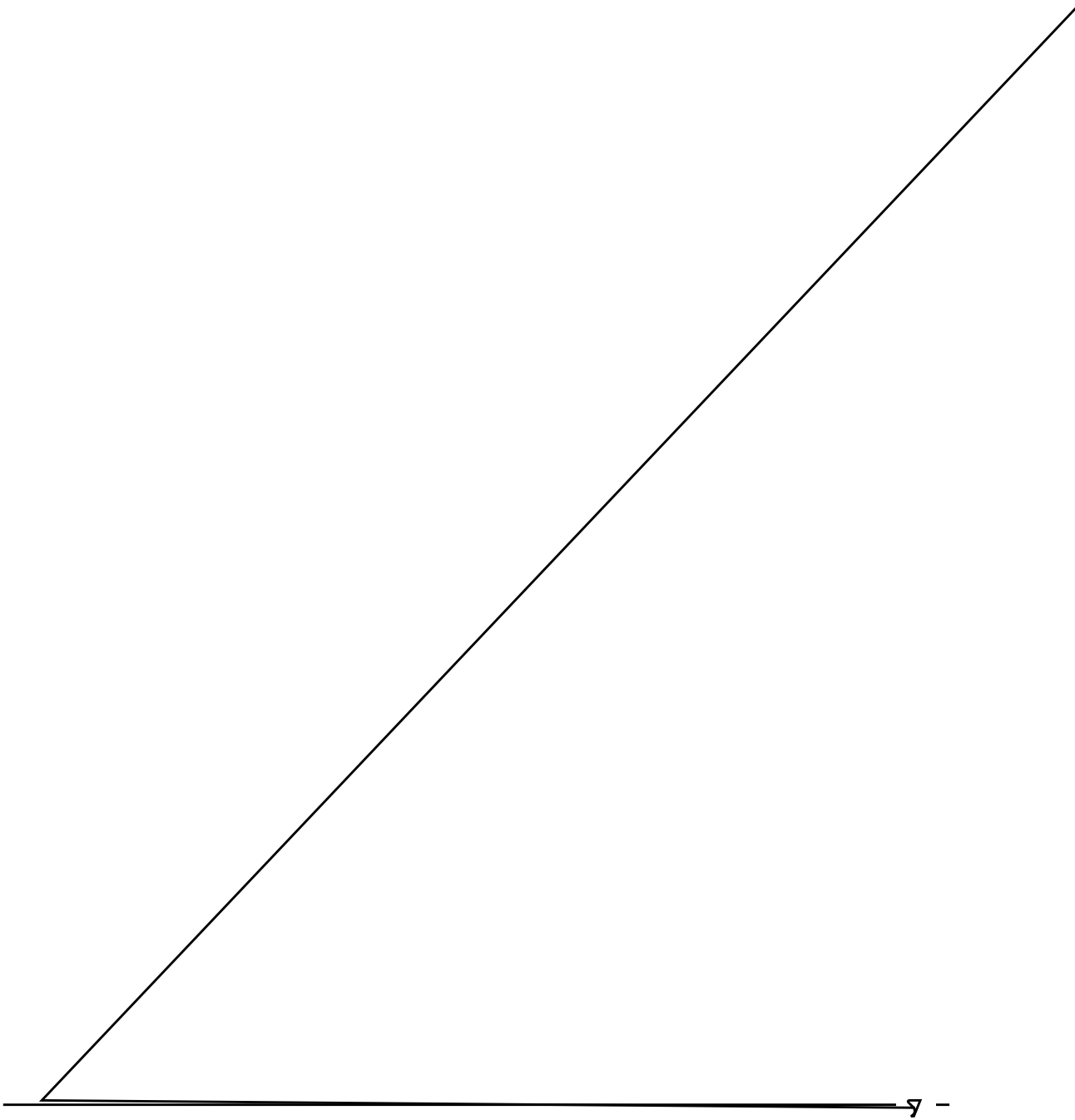
Must be the first syllable of a bundle.

Must be followed by 3 bundles delay before issuing a syncins operation

No latency constraints.

Exceptions:

None.



sh1add Register

sh1add $R_{dest} = R_{src1}, R_{src2}$

Semantics:

Description:

Shift left one and accumulate

Restrictions:

No address/bundle restrictions.

No latency constraints.

Exceptions:

None.

shl Register

shl R_{dest} = R_{src1}, R_{src2}

s	00	0	0	00010	dest	src2	src1

Semantics:

Description:

Shift left

Restrictions:

No address/bundle restrictions.

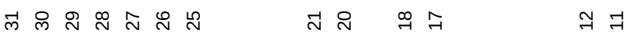
No latency constraints.

Exceptions:

None.

shru

shru R_{dest} = R_{src1}, R_{src2}



Semantics:

Description:

Logical shift right

Restrictions:

No address/bundle restrictions.

No latency constraints.

Exceptions:

None.

slctf Immediate

slctf $R_{idest} = B_{scond}, R_{src1}$, isrc2

stb

stb isrc2[R_{src1}] = R_{src2}

Semantics:

Description:

Store byte

Restrictions:

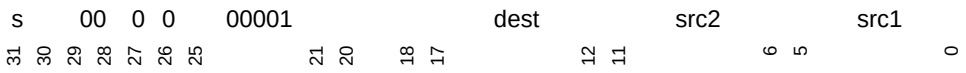
Uses the load/store unit, for which only operation is allowed per bundle.

No latency constraints.

Exceptions:

sub Register

sub $R_{dest} = R_{src2}, R_{src1}$



Semantics:

operand2 ← $SignExtend_{32}(R_{src2})$;
operand1 ← $SignExtend_{32}(R_{src1})$;
result ← result

Description:

Subtract

Restrictions:

No address/bundle restrictions.

No latency constraints.

Exceptions:

None.

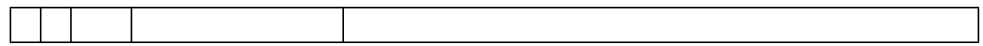
sub Immediate

sub R_{idest}

sxth Register

sxth R_{dest} = R_{src2}

Semantics:



Specific immediate extension, selects, extended arithmetic

Memory load, store

Control transfer