INDEX ARITHMETIC

The index arithmetic instruction set performs binary arithmetic on operands serving as addresses, index quantities, and counts. Except in literal index arithmetic, operands are 24 bits long. In literal index arithmetic one of the operands is contained in the instruction and is either 5 or 24 bits long.

The index arithmetic instructions have the following formats:

(I)

op	i	j	k

(II)

op	i	j	k

(III)

op	i	j	

(IV)

op	i	j		h

Operations are performed between index register X^j and X^k (format I), between X^j and the 5-bit k-literal (format II), or between X^j and the 24-bit h-literal (format IV). When format III is used, the single operand is X^j. The result replaces the contents of register X^i. Four index divide operations also replace the contents of X^{i+1} with part of its result. Three add-and-test operations set condition bit c_i and X^j. Except for these three the contents of X^j and X^k are not changed.

Index register X^0 is identically equal to zero. Since the contents of X^0 is always zero, results which are placed in X^0 are not recoverable.

Number Representation

The number representation used for 24-bit index quantities has the property that the operands and results for the instructions add, subtract, and multiply can be interpreted either as 2's complement integers in the range -2^{23} to $2^{23}-1$ or as positive integers, modulo 2^{24}, the range 0 to $2^{24}-1$. These forms are termed "signed" and "unsigned" index integers. Since signed divide is different from unsigned divide two sets of divide instructions are provided. Similarly both signed and unsigned compare instructions are provided where necessary; see Section 6.

2's Complement Arithmetic

In 2's complement arithmetic the bits of the 24-bit index operands and results have the following arithmetic weights:

weight	-2^{23}	2^{22}	2^{21}		2^{23-m}		2^3	2^2	2^1	2^0
operand bit position	0	1	2		m		20	21	22	23

for $1 \leq m \leq 23$

Numbers in this format are called signed index integers.

The integers which can be represented in 2's complement form in 24 bits range from -2^{23} to $2^{23}-1$. No special indication is given if a result is outside this range (overflows).

The 5-bit literal in instruction format II is considered as an integer in 2's complement form. Thus the bits have the following arithmetic weights:

weight	-2^4	2^3	2^2	2^1	2^0
operand bit position	0	1	2	3	4

Thus the range of representable literal values is -2^4 to 2^4-1.

Before participating in an operation the 5-bit quantity is extended to a 24-bit quantity by appending 19 high order bits equal in value (i.e. 0 or 1) to the high order bit of the k field. This operation transforms the 5-bit literal to a 24-bit signed index integer, and does not change the value of the number.

Modulo 2^{24} Arithmetic

In modulo 2^{24} arithmetic the bits of the 24-bit index operands and results have the following arithmetic weights:

weight	2^{23}	2^{22}	2^{21}		2^{23-m}		2^2	2^1	2^0
operand bit position	0	1	2		m		21	22	23

for $0 \leq m \leq 23$

Numbers in this format are called unsigned index integers.

The integers which can be represented in modulo 2^{24} form in 24 bits range from 0 to $2^{24}-1$; and by definition of modulo 2^{24} arithmetic all results are within this range.

The instructions with format II can be used in modulo 2^{24} arithmetic. If the high order bit of the k field is 0, the representable literal values are in the range 0 to 2^4-1. If the high order bit of the k field is 1, the representable literals are in the range $2^{24}-16$ to $2^{24}-1$.

Add Index — AX

	i	j	k

The contents of X^i are replaced by the low order 24 bits of the sum formed by the addition of the index integers in X^j and X^k.

Exceptions: none

Subtract Index — SX

	i	j	k

The contents of X^i are replaced by the low order 24 bits of the difference formed by the subtraction of the index integer in X^k from X^j.

Exceptions: none

Multiply Index — MX

	i	j	k

The contents of X^i are replaced by the low order 24 bits of the product of the index integers in X^j and X^k.

Exceptions: none

Divide with Remainder, Index DRX

	i	j	k

The contents of X^i is replaced by the signed quotient formed by dividing the signed index integer dividend in X^j by the signed index integer divisor in X^k, and the contents of X^{i+1} are replaced by the signed remainder. The value of the i-field is assumed to be even; if it is not, the low order bit of the i-field is forced to 0, bit RS is set, and the operation proceeds.

The signed index divide is performed as follows:

1. If the divisor is zero, both the quotient and remainder are set to zero, the index divide by zero exception bit (XDZ) is set to 1, and the remaining steps are omitted.

2. If the dividend is -2^{23} and the divisor is -1 (so that the true quotient 2^{23} cannot be represented), the quotient is set to zero, the remainder is set to -2^{23}, and the remaining steps are omitted.

3. The dividend is divided by the divisor to form an exact quotient.

4. If the exact quotient is an integer, it forms the result quotient. The result remainder is zero.

5. If the exact quotient was not an integer, the integer part of the exact quotient forms the result quotient. That is, the exact quotient is rounded toward zero. The remainder is defined as:

$$\text{remainder} = \text{dividend} - (\text{quotient} \times \text{divisor})$$

Exceptions	Exception bit
$X^k = 0$	XDZ
i odd	RS

Divide Index DX

	i	j	k

The contents of X^i is replaced by the signed quotient formed by the division of the signed index integer dividend in X^j by the signed index integer divisor in X^k. The quotient is defined as in DRX.

Exceptions	Exception bit
$X^k = 0$	XDZ

Remainder Index RX

	i	j	k

The contents of X^i is replaced by the signed remainder formed by the division of the signed index integer dividend in X^j by the signed index integer divisor in X^k. The remainder is defined as in DRX.

Exceptions	Exception bit
$X^k = 0$	XDZ

Divide with Remainder, Unsigned Index DRUX

	i	j	k

The contents of X^i is replaced by the unsigned quotient formed by dividing the unsigned index integer dividend in X^j by the unsigned index integer divisor in X^k, and the contents of X^{i+1} are replaced by the unsigned remainder. The value of the i-field is assumed to be even; if it is not, the lower order bit of the i-field is forced to 0, bit RS is set, and the operation proceeds.

The unsigned index divide is performed as follows:

1. If the divisor is zero, both the quotient and remainder are set to zero, the index divide by zero exception bit (XDZ) is set to 1, and the remaining steps are omitted.
2. The dividend is divided by the divisor to form an exact quotient.
3. If the exact quotient is an integer, it forms the result quotient. The result remainder is zero.
4. If the exact quotient was not an integer, the integer part of the exact quotient forms the result quotient. That is, the exact quotient is rounded toward zero. The remainder is defined as:

$$\text{remainder} = \text{dividend} - (\text{quotient} \times \text{divisor})$$

Exception	Exception bit
$X^k = 0$	XDZ
i odd	RS

Divide Unsigned Index DUX

	i	j	k

The contents of X^i is replaced by the quotient formed by the division of the unsigned index integer dividend in X^j by the unsigned index integer divisor in X^k. The quotient is defined as in DRUX.

Exception	Exception bit
$X^k = 0$	XDZ

Remainder Unsigned Index RUX

	i	j	k

The contents of X^i is replaced by the remainder formed by the division of the unsigned index integer dividend in X^j by the unsigned index integer divisor in X^k. The remainder is defined as in DRUX.

Exception	Exception bit
$X^k = 0$	XDZ

<u>Add Index to Short Constant</u> AXC

	i	j	k

The contents of X^i are replaced by the low order 24 bits of the sum formed by the addition of the 24 bit number in X^j and the number in the literal k-field. The 5-bit k-field is extended to a 24-bit quantity before the addition by appending 19 high-order bits equal in value to the high order bit of the k-field.

Exceptions: none

<u>Add Index to Constant</u> AXK

	i	j	//////	h

The contents of X^i are replaced by the low order 24 bits of the sum formed by the addition of the index integers in X^j and the literal h-field.

Exceptions: none

Multiply Index by Constant MXK

	i	j		h

The contents of X^i are replaced by the low order 24 bits of the product of the index integers in X^j and in the literal h-field.

Exceptions: none

Divide with Remainder Index by Constant DRXK

	i	j		h

The signed index integer in X^j is divided by the signed index integer divisor in the literal h-field. The signed index integer quotient replaces the contents of X^i and the signed index integer remainder replaces the contents of X^{i+1}. The value of the i-field is assumed to be even.

The quotient, remainder, and exception are defined as in DRX.

Exception	Exception bit
h-field = 0	XDZ
i odd	RS

<u>Divide Index by Constant</u> DXK

	i	j		h

The signed index integer dividend in X^j is divided by the signed index integer divisor in the literal h-field. The signed index integer quotient replaces the contents of X^i.

The quotient and exception are defined as in DRX.

Exception	Exception bit
h-field = 0	XDZ

<u>Remainder Index by Constant</u> RXK

	i	j		h

The signed index integer dividend in X^j is divided by the signed index integer divisor in the literal h-field. The signed index integer remainder replaces the contents of X^i.

The remainder and exception are defined as in DRX.

Exception	Exception bit
h-field = 0	XDZ

<u>Divide with Remainder Unsigned Index by Constant</u> DRUXK

	i	j		h

The unsigned index integer in X^j is divided by the unsigned index integer divisor in the literal h-field. The unsigned index integer quotient replaces the contents of X^i and the unsigned index integer remainder replaces the contents of X^{i+1}. The value of the i-field is assumed to be even.

The quotient, remainder, and exception are defined as in DRUX

Exceptions	Exception bit
h-field = 0	XDZ
i odd	RS

Divide Unsigned Index by Constant DUXK

	i	j	▨	h

The unsigned index integer dividend in X^j is divided by the unsigned index integer divisor in the literal h-field. The unsigned index integer quotient replaces the contents of X^i.

The quotient and exception are defined as in DRUX.

Exception	Exception bit
h-field = 0	XDZ

Remainder Unsigned Index by Constant RUXK

	i	j	▨	h

The unsigned index integer dividend in X^j is divided by the unsigned index integer divisor in the literal h-field. The unsigned index integer remainder replaces the contents of X^i.

The remainder and exception are defined as in DRUX.

Exception	Exception bit
h-field = 0	XDZ

Set Positive, Index SPX

	i	j	▨

The contents of register X^i are replaced by the absolute value of the signed index integer in X^j.

If the integer in X^j is -2^{23}, the contents of X^i are replaced by the value zero.

Exceptions: none

Set Negative, Index SNX

	i	j	▨

The contents of register X^i are replaced by the negative of the absolute value of the signed index integer in X^j.

Exceptions: none

Add Index and Test AXT

	i	j	k

The contents of X^j are replaced by the low-order 24 bits of the sum formed by the addition of the signed index integers in X^j and X^k.

If the original value of X^j_0 is different from the new X^j_0, condition bit c_i is set to 1; otherwise c_i is set to 0. Thus the condition bit is set to 1 when the addition causes the sign of the index integer in X^j to change (with zero considered positive).

Exception	Exception bit
c_{24} set to 0 or c_{25} set to 1	CC

<u>Add Index to Short Constant and Test</u> AXCT

	i	j	k

The contents of X^j are replaced by the low-order 24 bits of the sum formed by the addition of the signed index integer in X^j and the signed integer in the literal k-field. The 5-bit k-field is extended to a 24-bit quantity before the addition by appending 19 high-order bits equal in value to the high order bit of the k-field.

If the original value of X^j_0 is different from the new X^j_0, condition bit c_i is set to 1; otherwise, c_i is set to 0.

Exception	Exception bit
c_{24} set to 0 or c_{25} set to 1	CC

<u>Add Index to Constant and Test</u> AXKT

	i	j	▨	h

The contents of X are replaced by the low order 24 bits of the sum formed by the addition of the signed index integers in X^j and the literal h-field.

If the original value of X^j_0 is different from the new X^j_0, condition bit c_i is set to 1; otherwise, c_i is set to 0.

Exception	Exception bit
c_{24} set to 0 or c_{25} set to 1	CC