LOAD AND STORE OPERATIONS

The load operations replace the contents of index (X) registers, or arithmetic (A) registers, with information from storage. The storage contents remain unchanged. The store operations replace information in storage with information from one or more of the X or A registers. The register contents remain unchanged. The count to storage and swap with storage instructions act as both a load and a store and thus may change both the register and storage.

The load and store instructions have one of the following formats:

Short	op	i	j	k	
Long	op	i	j	k	h
	9	5	5	5	24

For each operation the i field designates the register, or registers, to be loaded or stored. The j and k fields designate two index registers which are added together to calculate the effective address of the storage information. In the long format the h field is also added in forming the effective address.

All addresses generated by the main processor are considered to be virtual addresses by the mapping mechanism. This mechanism transforms (maps) the virtual address into the address of a physical location in storage. The mapping mechanism deals with 36 bit virtual addresses (ea). The low order 24 bits are called the effective address (eal) and the high order 12 bits are called the key (eak).

The eak is specified by one of four key registers: problem normal key PNK, problem alternate key PAK, supervisory normal key SNK, and supervisory alternate key SAK. Which key is used is defined by the MPM mode, which is either problem or supervisory, and the instruction code, which specifies either normal or alternate. The following table describes the key specification:

	Supervisory Mode	Problem Mode
Normal Key	SNK	PNK
Alternate Key	SAK	PAK

The eal may be computed in two ways. In normal indexing the index quantities are aligned so that the low order bits of each are added together. In true indexing the quantity from X^k is doubled by shifting it left one position prior to addition. The eal addition is computed modulo 2^{24} for both types of indexing.

The following table describes normal and true indexing for both long and short formats:

	Short Format	Long Format
Normal Indexing	eal $\leftarrow X^j + X^k$	eal $\leftarrow X^j + X^k + h$
True Indexing	eal $\leftarrow X^j + 2 \times X^k$	eal $\leftarrow X^j + 2 \times X^k + h$

additions are computed modulo 2^{24}

X^j, X^k = contents of the index registers specified by the j and k fields of the instruction

h = literal field of the instruction

Index load and store operands are 24 bits long. The length of the operands for arithmetic loads and stores is specified in the operation code. Three lengths may be specified: half (24 bits), single (48 bits), and double (96 bits). When loading half word quantities the 24 bit number is expanded to 48 bits when placed in the arithmetic register as follows: if the instruction calls for the left half to be loaded, 0's replace the low order 24 bits of the register; if the right half is loaded, the high order bit of the half word is copied into the high order 24 bits of the register. When storing half word quantities, the selected half of the arithmetic register is stored and the register contents are uneffected.

Index register X^0 is specified to be a source of 0's. To specify single indexing or no indexing, either the j or k field or both should be set to zero. When X^0 is stored, 0's replace the 24-bit storage contents located by the effective address. Information loaded into X^0 is not recoverable from X^0.

Similarly, A^0 is specified to contain 0's. If A^0 is used as a source in a store operation, the length of the zero quantity stored is determined by the operand length specified in the instruction. Information loaded into A^0 is not recoverable from A^0. If A^0 is specified by the i field of a load arithmetic double instruction, register A^1 is also set to 0's.

Multiple Load and Store

The multiple load operations replace the contents of blocks of successive arithmetic (A) or index (X) registers with information taken from consecutive storage locations. (Register number 0 is considered to be the successor of register number 31.) Storage remains unchanged.

The multiple store operations reverse the process, that is, information from the registers is stored in consecutive storage locations. The registers are unchanged.

The multiple load and store operations have the following format:

op	i	j	k	h

For each operation the i-field designates the initial register to be loaded or stored; j gives the number of registers to be loaded or stored; and the modulo 2^{24} sum of index register k and the literal, h, gives the effective address of the first storage location.

Registers X^0 and A^0 are sources of 0's; information loaded into them is not recoverable.

The value of the i-field must be even when X-unit operands are specified. If it is not, the low order bit of the field is forced to 0, exception bit RS is set, and the operation proceeds. The use of the register pair $X^{0,1}$ in multiple load and store instructions results in the loading or storing 24 0's for X^0 and the 24 data bits for X^1.

Exceptions

Every virtual address generated for a load or store arithmetic, a load or store arithmetic double, or any multiple load or store, must be divisible by 2. If it is not, the BV (boundary value) exception bit is set to 1, and the operation proceeds using the address minus one as the storage address. Similarly the virtual address for STMZ and STMZA must be divisible by 64. If it is not, the BV bit is set to 1, and the operation proceeds using the address with the seven low order bits forced to 0's as the storage address.

The mapping mechanism checks the validity of all virtual addresses in two ways. First, if the virtual address does not correspond to an actual physical location, a missing address exception occurs and exception bit MA is set to 1. Second, if the virtual address of a store instruction refers to an area to which store access is not permitted, a protected address exception occurs and exception bit PA is set to 1. The setting of the MA or PA exception bit results in a type 2 interruption condition. See the chapter "Interruptions" for further details.

Each storage address generated for multiple load and store operations is individually checked for MA and PA exceptions.

When a double precision A-unit operand is specified by the instruction code, 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, exception bit RS is set, and the operation proceeds. These fifteen A-unit double precision quantities are specifiable; namely the data in register pairs specified by 2, 4, 6, ... , 30. The double precision quantity specified by A^0 is defined to be 96 0's, so that register A^1 is not the low order half of any double precision quantity.

<u>Load Index</u> (half word format) LXH

	i	j	k

$eal \leftarrow X^j + X^k$

$eak \leftarrow$ normal key

$X^i \leftarrow M^{ea}$

Exceptions	Exception bit
missing address	MA

<u>Load Index</u> LX

	i	j	k	h

$eal \leftarrow X^j + X^k + h$

$eak \leftarrow$ normal key

$X^i \leftarrow M^{ea}$

Exceptions	Exception bit
missing address	MA

<u>Load Index per Alternate Key</u> LXA

	i	j	k	h

$eal \leftarrow X^j + X^k + h$

$eak \leftarrow$ alternate key

$X^i \leftarrow M^{ea}$

This instruction is identical to LX except that in forming the storage address the alternate key is used.

Exceptions	Exception bit
missing address	MA

Store Index (half word format) STXH

	i	j	k

$eal \leftarrow X^j + X^k$

$eak \leftarrow$ normal key

$M^{ea} \leftarrow X^i$

Exceptions	Exception bit
missing address	MA
protected address	PA

Store Index STX

	i	j	k	h

$eal \leftarrow X^j + X^k + h$

$eak \leftarrow$ normal key

$M^{ea} \leftarrow X^i$

Exceptions	Exception bit
missing address	MA
protected address	PA

Store Index per Alternate Key STXA

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{alternate key}$$

$$M^{ea} \leftarrow X^i$$

This instruction is identical to STX except that in forming the storage address the alternate key is used.

Exceptions	Exception bit
missing address	MA
protected address	PA

Count to Storage

CNTS

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$X^i \leftarrow M^{ea}$$

$$\text{If } M^{ea} \neq 2^{24}-1\text{:}\quad M^{ea} \leftarrow M^{ea} + 1$$

$$\text{If } M^{ea} = 2^{24}-1\text{:}\quad M^{ea} \leftarrow M^{ea}$$

The contents of the memory location is loaded into X^i. The contents of M^{ea} is treated as an unsigned integer and is incremented by one, modulo 2^{24}. If this would cause M^{ea} to go to zero, the add is suppressed. The fetch from ea and the subsequent storing into it are interlocked so that no intervening accesses are permitted.

Exceptions	Exception bit
missing address	MA
protected address	PA

Swap with Storage SWS

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$\text{If } M^{ea} \neq 0\text{:}\quad X^i \leftarrow M^{ea}$$

$$\text{If } M^{ea} = 0\text{:}\quad X^i \leftarrow M^{ea} \text{ and } M^{ea} \leftarrow X^i$$

The contents of the memory location is loaded into X^i. If the contents of M^{ea} is zero (twenty-four 0's), M^{ea} is replaced by the original contents of X^i. If M^{ea} is different from zero, M^{ea} is not changed. The fetch from M^{ea} and the (potential)subsequent storing into it are interlocked so that no intervening accesses are permitted.

Exceptions	Exception bit
missing address	MA
protected address	PA

Swap with Storage per Alternate Key SWSA

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{alternate key}$$

$$\text{If } M^{ea} \neq 0\text{:}\quad X^i \leftarrow M^{ea}$$

$$\text{If } M^{ea} = 0\text{:}\quad X^i \leftarrow M^{ea} \text{ and } M^{ea} \leftarrow X^i$$

This instruction is identical to SWS except that the alternate key is used.

Exceptions	Exception bit
missing address	MA
protected address	PA

<u>Load Arithmetic</u> (half word format) LAH

	i	j	k

eal $\leftarrow X^j + X^k$

eak $\leftarrow$ normal key

$A^i \leftarrow M^{ea}$

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV

<u>Load Arithmetic</u> LA

	i	j	k	h

eal $\leftarrow X^j + X^k + h$

eak $\leftarrow$ normal key

$A^i \leftarrow M^{ea}$

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV

Load Arithmetic per Alternate Key LAA

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{alternate key}$$

$$A^i \leftarrow M^{ea}$$

This instruction is identical to LA except that in forming the storage address the alternate key is used.

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV

Store Arithmetic (half word format) STAH

	i	j	k

$$eal \leftarrow X^j + X^k$$

$$eak \leftarrow \text{normal key}$$

$$M^{ea} \leftarrow A^i$$

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV

Store Arithmetic STA

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$M^{ea} \leftarrow A^i$$

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV

Store Arithmetic per Alternate Key STAA

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{alternate key}$$

$$M^{ea} \leftarrow A^i$$

This instruction is identical to STA except that in forming the storage address the alternate key is used.

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV

Load Arithmetic Double (half word format) LDH

	i	j	k

$$eal \leftarrow X^j + X^k$$

$$eak \leftarrow \text{normal key}$$

$$A^{i,i+1} \leftarrow M^{ea,ea+2}$$

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV
i odd	RS

Load Arithmetic Double LD

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$A^{i,i+1} \leftarrow M^{ea,ea+2}$$

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV
i odd	RS

Store Arithmetic Double
(half word format)

STDH

	i	j	k

$$eal \leftarrow X^j + X^k$$

$$eak \leftarrow \text{normal key}$$

$$M^{ea,\,ea+2} \leftarrow A^{i,\,i+1}$$

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV
i odd	RS

Store Arithmetic Double

STD

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$M^{ea,\,ea+2} \leftarrow A^{i,\,i+1}$$

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV
i odd	RS

Load Arithmetic, True Indexing
(half word format)

LATH

	i	j	k

$eal \leftarrow X^j + 2 \times X^k$

$eak \leftarrow$ normal key

$A^i \leftarrow M^{ea}$

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV

Load Arithmetic, True Indexing

LAT

	i	j	k	h

$eal \leftarrow X^j + 2 \times X^k + h$

$eak \leftarrow$ normal key

$A^i \leftarrow M^{ea}$

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV

<u>Store Arithmetic, True Indexing</u>
(half word format)

STATH

	i	j	k

$$eal \leftarrow X^j + 2 \times X^k$$

$$eak \leftarrow \text{normal key}$$

$$M^{ea} \leftarrow A^i$$

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV

<u>Store Arithmetic, True Indexing</u>

STAT

	i	j	k	h

$$eal \leftarrow X^j + 2 \times X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$M^{ea} \leftarrow A^i$$

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV

Load Arithmetic, Left Half — LL

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$A^i_{0,1,2,\cdots,23} \leftarrow M^{ea}$$

$$A^i_{24,25,26,\cdots,47} \leftarrow 0\,[24]$$

Exceptions	Exception bit
missing address	MA

Load Arithmetic, Right Half — LR

	i	j	k	h

$$eal \leftarrow X^j + X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$A^i_{0,1,\cdots,23} \leftarrow M^{ea}_0\,[24]$$

$$A^i_{24,25,\cdots,47} \leftarrow M^{ea}$$

Exceptions	Exception bit
missing address	MA

<u>Store Arithmetic, Left Half</u> STL

	i	j	k	h

$eal \leftarrow X^j + X^k + h$

$eak \leftarrow$ normal key

$M^{ea} \leftarrow A^i_{0,1,2,\cdots,23}$

Exceptions	Exception bit
missing address	MA
protected address	PA

<u>Store Arithmetic, Right Half</u> STR

	i	j	k	h

$eal \leftarrow X^j + X^k + h$

$eak \leftarrow$ normal key

$M^{ea} \leftarrow A^i_{24,25,26,\cdots,47}$

Exceptions	Exception bit
missing address	MA
protected address	PA

Load Multiple Index

LMX

	i	j	k	h

number of registers loaded $\leftarrow$ j

$$eal \leftarrow X^{k} + h$$

$$eak \leftarrow \text{normal key}$$

$$X^{i,i+1} \leftarrow M^{ea,ea+1}$$

$$X^{i+2,i+3} \leftarrow M^{ea+2,ea+3}$$

$$\cdots$$

$$X^{i+j-2,i+j-1} \leftarrow M^{ea+j-2,ea+j-1}$$

If j is not divisible by 2, the number of registers loaded will be j-1. If j is zero or one, 32 registers will be loaded.

Exceptions	Exceptions
missing address	MA
ea not divisible by 2	BV
i odd	RS

<u>Load Multiple Index per Alternate Key</u> LMXA

	i	j	k	h

number of registers loaded ← j

$$eal \leftarrow X^{k} + h$$

$$eak \leftarrow \text{alternate key}$$

$$X^{i,i+1} \leftarrow M^{ea,ea+1}$$

$$X^{i+2,i+3} \leftarrow M^{ea+2,ea+3}$$

$$\cdots$$

$$X^{i+j-2,i+j-1} \leftarrow M^{ea+j-2,ea+j-1}$$

If j is not divisible by 2, the number of registers loaded will be j-1. If j is zero or one, 32 registers will be loaded.

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV
i odd	RS

Store Multiple Index STMX

	i	j	k	h

number of registers stored ← j

$$eal \leftarrow X^{k} + h$$

$$eak \leftarrow \text{normal key}$$

$$M^{ea,\, ea+1} \leftarrow X^{i,\, i+1}$$

$$M^{ea+2,\, ea+3} \leftarrow X^{i+2,\, i+3}$$

$$\cdots$$

$$M^{ea+j-2,\, ea+j-1} \leftarrow X^{i+j-2,\, i+j-1}$$

If j is not divisible by 2, the number of registers stored will be j-1. If j is zero or one, 32 registers will be stored.

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV
i odd	RS

Store Multiple Index per Alternate Key STMXA

	i	j	k	h

number of registers stored ← j

$$eal \leftarrow X^{k} + h$$

$$eak \leftarrow \text{alternate key}$$

$$M^{ea, ea+1} \leftarrow X^{i, i+1}$$

$$M^{ea+2, ea+3} \leftarrow X^{i+2, i+3}$$

$$\cdots$$

$$M^{ea+j-2, ea+j-1} \leftarrow X^{i+j-2, i+j-1}$$

If j is not divisible by 2, the number of registers stored will be j-1. If j is zero or one, 32 registers will be stored.

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV
i odd	RS

Load Multiple Arithmetic   LMA

	i	j	k	h

number of registers loaded ← j

$$eal \leftarrow X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$A^i \leftarrow M^{ea}$$

$$A^{i+1} \leftarrow M^{ea+2}$$

$$\cdots$$

$$A^{i+j-1} \leftarrow M^{ea+2j-2}$$

If j is zero, 32 registers will be loaded.

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV

Load Multiple Arithmetic per Alternate Key LMAA

	i	j	k	h

number of registers loaded ← j

$$eal \leftarrow X^k + h$$

eak ← alternate key

$$A^i \leftarrow M^{ea}$$

$$A^{i+1} \leftarrow M^{ea+2}$$

$$\cdots$$

$$A^{i+j-1} \leftarrow M^{ea+2j-2}$$

If j is zero, 32 registers will be loaded.

Exceptions	Exception bit
missing address	MA
ea not divisible by 2	BV

Store Multiple Arithmetic STMA

	i	j	k	h

number of registers stored ← j

$$eal \leftarrow X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$M^{ea} \leftarrow A^i$$

$$M^{ea+2} \leftarrow A^{i+1}$$

$$\cdots$$

$$M^{ea+2j-2} \leftarrow A^{i+j-1}$$

If j is zero, 32 registers will be stored.

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV

Store Multiple Arithmetic per Alternate Key STMAA

	i	j	k	h

number of registers stored ← j

$$eal \leftarrow X^{k} + h$$

$$eak \leftarrow \text{alternate key}$$

$$M^{ea} \leftarrow A^{i}$$

$$M^{ea+2} \leftarrow A^{i+1}$$

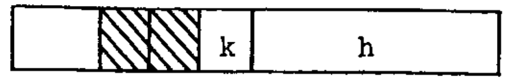
$$\cdots$$

$$M^{ea+2j-2} \leftarrow A^{i+j-1}$$

If j is zero, 32 registers will be stored.

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 2	BV

Store Multiple Zeros STMZ



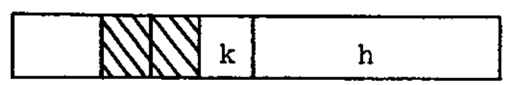
$$eal \leftarrow X^k + h$$

$$eak \leftarrow \text{normal key}$$

$$M^{ea, ea+1, \ldots, ea+63} \leftarrow 0 \; [1536]$$

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 64	BV

Store Multiple Zeros per Alternate Key STMZA



$$eal \leftarrow X^k + h$$

$$eak \leftarrow \text{alternate key}$$

$$M^{ea, ea+1, \ldots, ea+63} \leftarrow 0 \; [1536]$$

This instruction is identical to STMZ except that the alternate key is used.

Exceptions	Exception bit
missing address	MA
protected address	PA
ea not divisible by 64	BV