amdahl Architecture Specification

TITLE 470V/7 Functional Specification

AUTHOR Richard Tobias

470V/7 Functional Specification

12-22-77

REV	SRN	DATE	REV	SRN	DATE	APPROVALS	DATE
E	903754	1-5-78				ARCHITECT Tobias	1-4-78
						REV. ARCH Grant	1-5-78
						MGR ARCH [illegible]	1-5-78
						ENGR M. R. Clements	1-5-78

AMDAHL COMPANY PRIVATE

P/N REV 820104-600 E

NEXT PAGE 2

470V/7
FUNCTIONAL SPECIFICATION

CONTENTS

INTRODUCTION

The Amdahl 470V/7 is a data processing system which is based on the capabilities of IBM System/370. Portions of two IBM publications -

1. IBM System/370 Principles of Operation (GA22-7000), and
2. IBM System/360 and System/370 I/O Interface - Channel to Control Unit OEM Information (GA22-6974),

provide the basic external specifications for Amdahl 470V/7. Differences, where they exist, and features which are not supported by System/370 are described by this document.

Supported System/370 CPU Features

System/370 CPU features supported by Amdahl 470V/7 are:

* Universal instruction set and associated computing facilities including:
    - Byte oriented operands
    - General Registers
    - Control Registers
    - Storage Protection
    - Interval Timer
    - Time-of-day clock
    - Floating point registers and instructions
    - Basic system console functions
* Extended precision floating point
* Direct Control feature
* CPU timer and clock comparator feature
* CPU dynamic address translation
* Program Event Recording (PER)
* Conditional swapping - Compare and Swap instructions
* PSW key handling

Supported System/370 I/O Features

System/370 I/O features supported by Amdahl 470V/7 are:

* Channel indirect data addressing
* CLEAR I/O instruction

- Bus extension to 2 bytes
- Error alert feature (Disconnect-In tag line)
- Command retry procedure
- High speed data transfer feature (Data-In and Data-Out tag lines)
- Fast release instruction
- Limited channel logout
- I/O extended logout

Non-Supported System/370 Features

The 470V/7 does not support the multiprocessing feature of System/370. In particular, shared main storage, prefixing, CPU signalling and response and TOD clock synchronization functions are not available. Extensions to external interruption, multiprocessing control register positions and multiprocessing instructions are also not available.

Limited CPU serialization is available. All interruptions cause serialization, as do SUPERVISOR CALL, LOAD PSW, PURGE TLB, READ DIRECT and WRITE DIRECT instructions. However, BRANCH ON CONDITION, COMPARE AND SWAP, STORE CLOCK, TEST AND SET and all I/O instructions do not cause CPU serialization.

Extensions to System/370

Amdahl 470V/7 provides features which are extensions to System/370 architecture. The major extensions are listed below:

- Feature control register
- Channel dynamic address translation
- 31 bit storage address feature
- 16 additional channels
- Branch and store feature
- Selective TLB purge feature
- 4K byte key-in-storage feature
- Set key and clear feature
- Hardware measurement interface
- Functional DIAGNOSE operations
- Cycle Counter

FEATURE CONTROL REGISTER

Amdahl 470V/7 has features which are extensions to IBM System/370. The 32 bit feature control register (FCR) provides a means for maintaining and manipulating control information for some of these features. This register is not part of addressable storage. The instruction LOAD FEATURE CONTROL REGISTER provides a means for loading control information from main storage into FCR, whereas the STORE FEATURE CONTROL REGISTER instruction permits information to be transferred from FCR to main storage.

Format

The format of the feature control register is shown below.

```
+--+--+--+--+--+--+--+--+------------------------------+
|KC|CD|RM|CL|CE|BS|PG|PE|//////////////////////////////|
+--+--+--+--+--+--+--+--+------------------------------+
 0  1  2  3  4  5  6  7  8                             31
```

Set Key and Clear Feature (KC) - Bit 0 controls the set key and clear feature. When the bit is one, the feature is enabled and the SET STORAGE KEY instruction can specify either the standard set key operation or the set key and clear operation. When the bit is zero, the feature is disabled.

Channel Dynamic Address Translation (CD) - Bit 1 controls the channel dynamic address translation feature. When the bit is one, the feature is enabled, when the bit is zero, the feature is disabled.

Real Address Modifier (RM) - When bit 2 is zero, all real addresses are 24 bits, no matter how or under what condition they were generated; when the bit is one, real address size is not restricted.

Channel Storage Address Extension (CL) - Bit 3 controls the channel storage address extension feature. When the bit is one, the feature is enabled and 31 bit channel storage addresses are allowed; when the bit is zero, the feature is disabled.

Channel Extension (CE) - Bit 4 controls the channel extension feature. When the bit is one, the feature is enabled and 16 additional channels are allowea; when the bit is zero, the feature is disabled.

Branch and Store Feature (BS) - Bit 5 controls whether branch and store instructions can be executed. When the bit is one, the feature is enabled and the instructions are valid; when the bit is zero, the feature is disabled.

Selective TLB Purge (PG) - Bit 6 controls the selective TLB purge feature. When the bit is one, the feature is enabled and selective purge instructions are valid; when the bit is zero, the feature is disabled.

4K Byte Protection (PE) - Bit 7 controls the protection extension feature. When the bit is one, the feature is enabled and 4K byte blocks of real storage are protected with a single key; when the bit is zero, the feature is disabled.

New Instructions

Two instructions under the diagnose code are provided to change and inspect the feature control register.

Instruction	Mnemonic	Type	Exceptions*	Code
Ld feat ctrl reg	LFCR	S	M,A	8301
Sto feat ctrl reg	STFCR	S	M,A	8302

Both of these instructions are privileged instructions and fall under the System Control class of instructions as defined in the IBM System/370 Principles of Operation.

LOAD FEATURE CONTROL REGISTER

```
    LFCR       D2(B2)           <S>
  +-------------------+----+---------------+
  |        8301       | B2 |      D2       |
  +-------------------+----+---------------+
   0                   16   20           31
```

Although the Feature Control Register (FCR) is architecturally specified to be 4 bytes long, 470V/7 implements only the first byte; thus the one-byte second operand is loaded into FCR. The second operand remains unchanged in main storage. If an accessing exception is recognized for the one byte second

* A Accessing Exception
M Privileged Operation Exception

operand, the operation is suppressed.

A serialization function is performed. CPU operation is delayed until all previous accesses to main storage have been completed, as observed by channels. No subsequent instructions, their operands, or dynamic address translation entries are fetched until the execution of the subject instruction is complete.

When a feature is not installed, the corresponding FCR bit remains "0" even if an attempt has been made to set it to "1".

Resulting Condition Code:

0 All defined FCR bits have been loaded successfully

1 An attempt has been made to activate (set the corresponding FCR bits to "1") features which are not installed.

2 -

3 -

Program Exceptions

Privileged Operation
Access (fetch, operand 2)

STORE FEATURE CONTROL REGISTER

```
STFCR      D2(B2)          <S>
+-------------------+----+---------------+
|        8302       | B2 |      D2       |
+-------------------+----+---------------+
0                    16   20            31
```

The contents of the one-byte Feature Control Register are stored at the byte location designated by the second operand address. The contents of FCR remain unchanged. If an accessing exception is recognized, the operation is suppressed.

Condition Code:

The code remains unchanged.

Program Exceptions:

Privileged Operation
Access (store, operand 2)

CHANNEL DYNAMIC ADDRESS TRANSLATION

The channel dynamic address translation feature provides the channel with facilities to use either real or logical addresses for data and command words.

Control

When the channel dynamic address translation feature is installed it is controlled by feature control register bit 1 (CD).

CD = 0	Feature disabled or not installed
CD = 1	Feature enabled

With the feature enabled, each subchannel can operate in either real or logical mode. To control the mode, two additional channel commands are avialable:

TRANSFER VIRTUAL AND LOCK (TVL)
TRANSFER AND LOCK (TL)

START I/O (SIO) and START I/O FAST RELEASE (SIOF) instructions always begin channel operations in real address mode. Operation proceeds in real mode until the channel program executes a TVL command. This changes the mode to logical and forces any further TVL or TL commands in the same channel program to cause a program check condition. Alternatively, while still in real mode, the channel program may execute a TL command. This does not change the mode, but, like TVL, forces any further TVL or TL commands in the channel program to cause a program check condition.

Subchannel Translation Word (STW)

A channel operating in logical mode interprets all storage addresses as logical addresses except those of permanently assigned locations. Translation control parameters, which correspond to the CPU parameters in control registers 0 and 1, are transferred to a subchannel when a transfer-virtual-and-lock command is issued. These parameters are contained in 4 bytes called the subchannel translation word (STW).

Only the general format of STW is described here; a definition of each field will be given with the description of the facility with which the field is associated.

The format and fields of STW are:

```
+--------+----------------------+------+----+------+---+
| Length | Segment Table Origin |  PS  | CR |  SS  | 0 |
+--------+----------------------+------+----+------+---+
 0        8                      26     28   29     31
```

Bits	Name of field	Mnemonic	Facility
0- 7	Segment table length	LEN	Dyn Addr Trans
8-25	Segmemt table origin	STO	Dyn Addr Trans
26-27	Page size control	PS	Dyn Addr Trans
28	Real addr size control	CR	31 bit Stor Addr
29-30	Segment size control	SS	Dyn Addr Trans

TVL and TL Commands

TVL and TL commands may be executed only when channel dynamic address translation is enabled. Furthermore, only one of these commands is allowed per channel program. That is, in a channel program, after the execution of either a TVL or TL command, detection of subsequent TVL or TL commands causes a program check conditon to be generated.

If the first channel command word (CCW) designated by the channel address word (CAW) contains either a TVL or a TL command, the subchannel executes the command as a part of the SIO or SIOF instruction; that is, the TVL or TL command is considered a logical extension of the CAW, and the next command in the program is considered the first command for the purposes of initial device selection and status report to the program.

TRANSFER VIRTUAL AND LOCK   TVL

```
+-----------+----------------------------+
| xx01 0000 | Next CCW address (logical)|      CCW word 1
+-----------+----------------------------+
 16          8                          31

+----------------------------------------+
|      Subchannel Translation Word       |      CCW word 2
+----------------------------------------+
 32                                     63
```

A TVL command sets the subchannel to logical mode, transfers the STW to the subchannel, and "locks" the subchannel to prevent further address mode change by the channel program. Bits 0 and 1 of the CCW contain command modifier bits, while the logical address of the next CCW is contained in bits 8-31. Bits 32-63 contain the subchannel translation word.

A TVL command neither initiates an I/O operation at the channel nor informs the I/O device of the execution of the command.

A TVL command is not allowed in a CCW immediately following a CCW which specified data chaining, and is also not allowed if a previous TVL or TL command has been executed in the channel program. Specification errors are recognized if CCW bits 29-31 and 63 do not contain zeros, or if the page size or segment size fields of the STW contain an invalid code. If either a command restriction or a specification error is detected, a program check condition is generated and an interrupt signalled.

An error detected while attempting to translate the logical address and fetch the next CCW causes an appropriate check condition to be generated and an interrupt signalled. These conditions are described in the chapter "Dynamic Address Translation Process".

TRANSFER AND LOCK TL

```
+-----------+----------------------------+
| xx10 0000 |  Next CCW address (real)   |        CCW word 1
+-----------+----------------------------+
 0           8                          31

+----------------------------------------+
|////////////////////////////////////////|        CCW word 2
+----------------------------------------+
 32                                     63
```

A TL command "locks" the subchannel in real address mode. Bits 0 and 1 of the CCW contain command modifier bits and bits 8-31 contain the real address of the next CCW. Bits 32-63 are ignored.

A TL command does not initiate any I/O operation at the channel, and the I/O device is not signalled of the execution of the command.

A TL command is not allowed in a CCW immediately following a CCW which specified data chaining, and is also not allaowed if a previous TVL or TL command has been executed in the channel program. Furthermore, a specification error is recognized if CCW bits 29-31 do not contain zeros. If a command restriction or a specification error is detected, or the designated address of the next CCW is unavailable, a program check condition is generated and an interrupt is signalled. If the location of the next CCW is protected against fetching, a protection check condition is generated and an interrupt signalled.

Real and Logical Channel Addresses

Channel Command Word (CCW) addresses which are always real addresses are

a. the CCW address contained in bits 8-31 of the Channel Address Word (CAW); and
b. the CCW address contained in bits 8-31 of a CCW which specifies a TL command.

CCW addresses which can be either real or logical addresses depending on the subchannel address mode are

a. the CCW address contained in bits 8-31 of a CCW which specifies a transfer-in-channel (TIC) command; and
b. the CCW address stored in bits 8-31 of the Channel Status Word (CSW).

The CCW address contained in bits 8-31 of a CCW which specifies a TVL command is always a logical address.

An Indirect Address Word (IDAW) address contained in bits 8-31 of a CCW indicating channel indirect data addressing is a real address or a logical address depending on the subchannel address mode.

Data addresses which may be either real or logical addresses depending on the subchannel address mode are

a. the address contained in bits 8-31 of a CCW which specifies READ, WRITE, SENSE, CONTROL, or READ BACKWARD commands and not channel indirect data addressing; and
b. the address contained in an IDAW.

Variations/Additions to Channel Status Word

The CCW address portion of the CSW (bits 8-31) can contain either a real or a logical address depending on the subchannel address mode at the time the CSW is stored.

The program check bit of the CSW (bit 42) is set when one or more of the following conditions are detected -

a. Invalid CCW address specification - Either the CAW or one of the "transfer" commands (TIC, TVL, TL) did not designate the next CCW on a double-word boundary; that is, the low-order 3 bits of the CCW address did not contain zeros.

b. Invalid CCW address - The channel has attempted to fetch a CCW from a real storage location not available to the channel. This condition can occur 1) when a real storage CCW address is specified by the channel program (see Real and Logical Channel Address section); 2) when a CCW address is generated by the channel during command chaining with the subchannel in real address mode; and 3) when the subchannel is in logical address mode and a logical CCW address is translated to a real storage address.

c. Invalid command code - The command code in the first CCW designated be the CAW or in a CCW fetched on command chaining has an invalid pattern - 'xx00 0000' or 'xx11 0000', where x is a command modifier bit and can be either 0 or 1; or a TVL or TL command is specified in a CCW fetched on data chaining. Other than for TVL or TL, the code is not tested for validity during data chaining.

d. Invalid count - A CCW other than one specifying a transfer command (TIC, TVL, TL) contains a zero value in bits 48-63.

e. Invalid IDAW address specification - Channel indirect data addressing is specified, and the data address does not designate the first IDAW on an integral word boundary.

f. Invalid IDAW address - The channel has attempted to fetch an IDAW from a real storage location not available to the channel. This condition can occur 1) when a real storage IDAW address is specified by the channel program (see Real and Logical Channel Address section); or 2) when an IDAW address is sequentially generated by the channel with the subchannel in real address mode; or 3) when the subchannel is in logical address mode and a logical IDAW address is translated to a real storage address.

g. Invalid data address - The channel has attempted to move data to or from a real storage location not available to the channel. This condition can occur 1) when a real storage data address is specified by the channel program (see Real and Logical Channel Address section); 2) when a data address is generated by the channel with the subchannel in real address mode; or 3) when the subchannel is in logical address mode and a logical data address is translated to a real storage address.

h. Invalid IDAW specification - when 24 bit addressing mode is specified, bits 0-7 of IDAW are not all zeros; or when 31 bit addressing mode is specified, bit 0 of IDAW is not zero.

i. Invalid CAW format - The CAW does not contain zeros in bit positions 4-7.

j. Invalid CCW format - A CCW other than one specifying a "transfer" command (TIC, TVL, TL) does not contain zeros in bit positions 38-39.

k. Invalid sequence - A CCW has been fetched which contains a command which is invalid at the time it is ready for executuon. The condition can occur when 1) a CCW specifies TVL or TL when the subchannel is in a "locked" state; 2) a CCW specifies TVL or TL when the channel dynamic address translation is not installed or not enabled; 3) the first CCW designated by the CAW specifies a TIC command; or 4) a CCW specifies TIC when the immediately preceeding CCW also specified TIC.

l. Translation fault - A situation has occurred during dynamic address translation which prevents the process from completing. More details about the incomplete translation are provided in the extended status byte of the CSW.

The channel control check bit of the CSW (bit 45) is set when conditions described in IBM System/370 Principles of Operation occur. The bit is also set when a subchannel operating in logical mode attempts to signal an interrupt after channel dynamic address translation has beem disabled; that is, after feature control register bit 1 (CD) has been set to "0".

Channel Status Word Extension

When the channel dynamic address translation feature is enabled an additional byte is required to transmit translation status to the program. This additional status byte, called the extended channel status byte (ESB), is a logical extension of the channel status word (CSW). Whenever channel dynamic address translation is enabled and the channel status portion of the CSW is updated, the extended channel status byte is stored at real location 184.

The format of the extended channel status byte is:

```
+----+----+----+----+--------------------+
¦ LM ¦ SI ¦ PI ¦ SC ¦                    ¦     MS location 184
+----+----+----+----+--------------------+
  0    1    2    3    4                7
```

Logical Mode (LM): Bit 0

0 - subchannel was in real mode when ESB stored
1 - subchannel was in logical mode when ESB stored

Segment Invalid (SI): Bit 1

Bit 1, when one, indicates a segment invalid condition occurred during the execution of a channel program in the logical mode. SI is always zero when LM is zero.

Page Invalid (PI): Bit 2

Bit 2, when one, indicates a page invalid condition occurred during the execution of a channel program in the logical mode. PI is always zero when LM is zero.

Translation Specification Check (SC): Bit 3

Bit 3, when one, indicates a translation specification has occurred. SC is always zero when LM is zero.

Bits 4-7

Bits 4-7 are reserved and are stored as zeros.

Programming Notes

When the subchannel is in real mode, channel indirect data addressing permits a single 24 bit real address from the CCW to control the transmission of data that spans non-contiguous pages in real storage. When the subchannel is in logical mode, channel indirect data addressing makes it possible to map the 24 bit logical address from the CCW into another logical address. This logical address can then itself be translated into a non-contiguous set of pages in real storage.

Indirect data addressing in the logical mode requires that the channel fetch an additional word from real storage and translate its address. Depending on the I/O device in operation this may result in overrun. Thus, programmers should be careful in using this facility.

When channel dynamic address translation is enabled some implementations of the extended channel status byte may provide zeros in undefined bit positions, however programs should not depend on such zeros.

The channel dynamic address translation architecture operates as a superset of the channel architecture defined by IBM System/370 Principles of Operation, except for the use of real location 184. This location is used to store the extended status byte in 470 architecture but is " reserved" in System/370 architecture. Programs which are meant to be compatible should be written to be insensitive to this address.

31 BIT STORAGE ADDRESSING

The 31 bit storage addressing feature provides both the CPU and channels with facilities for generating 31 bit storage addresses. In addition, the dynamic address translation facility is provided the capability of furnishing a 31 bit real address. These added facilities extend both the logical and real addressing capability of the processor to 2,147,483,648 bytes (2G bytes).

Storage Address Control

When the processor is in basic control (BC) mode, storage addresses for instruction, operands, and channel data and control words are always 24 bit real addresses.

When the processor is in extended control (EC) mode and the 31 bit storage addressing feature is installed storage addresses are controlled as follows:

1. Storage addresses generated by the CPU are either logical addresses or real addresses depending on the "translation mode" bit (T) - bit 5 of the PSW.

T=0	CPU generates real addresses
T=1	CPU generates logical addresses

2. The length of addresses generated by the CPU (either logical or real, depending on T) is controlled by the "CPU storage address extension" bit (AE) - bit 4 of the PSW.

AE=0	CPU generates 24 bit addresses
AE=1	CPU generates 31 bit addresses

3. Storage addresses provided to and generated by channels are either logical addresses or real addresses as defined in the "Channel Dynamic Address Translation" chapter.

4. The length of channel control words and channel data addresses (either logical or real) is controlled by the "channel storage address extension" bit (CL) - bit 3 of the feature control register.

CL=0	channel addresses are 24 bits long
CL=1	channel addresses are 31 bits long

5. Real addresses provided directly by the channels or generated directly by the CPU will be limited to 24 bits if

so designated by the "real address modifier" bit (RM) - bit 2 of the feature control register.

RM=0	Force 24 bit real address*
RM=1	31 bit real address allowed

6. If CPU dynamic address translation is specified, the length of the real address furnished by the translation process is controlled by the "CPU real address extension" bit (RE) - bit 10 of control register 0.

RE=0	24 bit real address
RE=1	31 bit real address

7. If channel dynamic address translation is specified, the length of the real address furnished by the translation process is controlled by the "subchannel real address extension" bit (CR) - bit 28 of the subchannel translation word.

CR=0	24 bit real address
CR=1	31 bit real address

Programming notes:

When the processor is in BC mode, all storage addresses are 24 bit real addresses.

When the processor is in EC mode and neither CPU nor channel translation is specified, CPU generated storage addresses are either 24 or 31 bit real addresses depending an AE and RM; and channnel storage addresses are either 24 or 31 bit real addresses depending on CL and RM.

When the CPU is in translation mode, it generates either 24 or 31 bit logical addresses, depending on AE; while the real address furnished by the translation process is either 24 or 31 bits long depending on RE. When a subchannel is in translation mode, it provides either 24 or 31 bit logical addresses depending on CL; while the real address furnished by the translation process is either 24 or 31 bits long depending on CR.

* The forcing function causes zeros to be set into high-order real address bits 1-7.

Storage Address Formats

The storage address is designated as though it were contained in low-order bit positions of a 32 bitvord.

```
+---+--------+---------------------+
| 0 | 1    7 | 8                31 |
+---+--------+---------------------+
    --------------------------------
          |     ----------------------
          |          |
          |          +------------ 24 bit storage address
          +----------------------- 31 bit storage address
```

Logical addresses are divided into three parts:

```
segment index - address of a segment within a single logical
                space,
page index    - address of a page within a single segment,
byte index    - address of a byte within a single page.
```

The segment index field of a 24 bit logical address begins with address bit 8 and bits 1-7 are ignored while the same field of a 31 bit logical address begins with address bit 1. The exact formats depend on the logical segment and page sizes:

64K byte segments, 4K byte pages

```
+--------------------------+--------+------------------+
|       Segment Index      | Pg Idx |    Byte Index    |
+--------------------------+--------+------------------+
 1                          16       20              31
```

64K byte segments, 2K byte pages

```
+--------------------------+----------+----------------+
|       Segment Index      |  Pg Idx  |   Byte Index   |
+--------------------------+----------+----------------+
 1                          16         21            31
```

1M byte segments, 4K byte pages

```
+-----------------+----------------+-----------------+
|   Segment Index |   Page Index   |    Byte Index   |
+-----------------+----------------+-----------------+
 1                 12               20             31
```

1M byte segments, 2K byte pages

```
+-----------------+------------------+----------------+
|   Segment Index |    Page Index    |    Byte Index  |
+-----------------+------------------+----------------+
 1                 12                 21            31
```

31 bit Address Generation

In computing a 31 bit operand address, bits 1-31 of both the base and the index registers are treated as positive 31 bit binary integers and bit 0 is ignored. The 12 bit displacement with 19 high-order zeros is similarly treated as a positive 31 bit binary integer. The three values are added as 31 bit binary numbers, ignoring overflow, thus the sum is always 31 bits long.

If the contents of a general register or a word in storage is used as a 31 bit address, bit positions 1-31 are used as the address and bit position 0 is ignored.

Although 31 bit storage addressing may be specified for channels, CCW addresses specified by a channel program are always 24 bits long, and during a command chaining operation, 24 bit address arithmetic is always used by the channel when computing subsequent CCW addresses.

Other addresses generated by channels when channel 31 bit storage addressing is specified are computed according to 31 bit arithmetic rules.

Programming Note: The method of designating CCW addresses means channel programs must be located in the first 16M bytes of storage.

Values Effected by 31 bit Addressing

Instruction addresses, addresses of branch targets and of operands in storage, and program-event-recording (PER) addresses are all either 24 or 31 bits long depending on AE - the CPU 31 bit storage extension control bit. These addresses are either real or logical depending on T - the translation mode control bit (EC PSW bit 5).

> Instruction addresses are contained in bits 32-63 of the current PSW. 24 bit addresses occupy bits 40-63 and bits 32-39 are unassigned. 31 bit addresses occupy bits 33-63 and bit 0 is unassigned. A specification exception occurs immediately after the PSW becomes active when unassigned bit positions do not contain zeros.
>
> Branch target addresses and operand in storage addresses are formed using either 24 or 31 bit arithmetic rules, depending on AE.

> PER addresses: The starting and ending addresses are contained in Control Registers (CR) 10 and 11 respectively and the interrupt address is placed in the word at real storage location 152. In 24 bit address mode, PER comparisons use CR 10 and 11, bits 8-31 and ignore bits 0-7. 8 high-order zeros are appended before storing the interrupt address. In 31 bit mode, CR 10 and 11, bits 1-31 are compared and bit 0 is ignored. One high-order zero is appended before storing the interrupt address.

The translation exception address is a logical address which caused either a segment or a page translation exception. The address is either 24 or 31 bits long depending on AE, and is placed in the word at real storage location 144 when the exception is recognized. Before storing, 8 high-order zeros are appended to a 24 bit address, and one to a 31 bit address.

The machine check extended logout (MCEL) address and the failing storage address (FSA) are both real addresses, thus their lengths are a function of control parameters which define the CPU real address size (see Storage Address Control section).

> The MCEL address is contained in bits 8-31 or bits 1-31 of CR 15, depending on CPU real address size. Bits 0-7 or bit 0 respectively, are ignored.
>
> The FSA is placed in the word at real storage location 248 according to the format defined in the "Miscelaneous" chapter.

I/O data address specification depends on whether "channel indirect data addressing" (CIDA) is specified by the CCW.

> If CIDA is not specified, the starting location for data transfer is always given by a 24 bit address contained in CCW bits 8-31. Subsequent data addresses generated directly by the channel are computed using rules of either 24 or 31 bit arithmetic depending on CL - the channel address extension control bit.
>
> If CIDA is specified, the starting address is contained in an "indirect data address word" (IDAW). 24 bit addresses are given by bits 8-31 and bits 0-7 are unassigned. 31 bit addresses are given by bits 1-31 and bit 0 is unassigned. A program check condition is generated when unassigned bit positions do not contain zeros. Note, subsequent data addresses generated directly by the channel are computed using reles of either 24 or 31 bit arithmetic depending on CL.

IDAW address: When CIDA is specified, the 24 bit address of the first IDAW of a list is contained in CCW bits 8-31. Subsequent IDAW addresses are computed directly by the channel using rules of either 24 or 31 bit arithmetic, depending on CL.

The I/O extended logout (IOEL) pointer is contained in bits 8-31 or 1-31 of the word at real storage location 172 depending on CL. Bits 0-7 or bit 0 are ignored.

The segment table entry address is a real address formed by adding the 18 bit segment table origin (concatenated with 6 low-order zeros) to the segment index portion of the logical address. The address is computed using rules of either 24 or 31 bit arithmetic, depending on RE - the CPU real address extension control bit.

The page table address is a real address formed by adding the page index portion of the logical address to the page table origin (PTO). The address is computed using rules of either 24 or 31 bit arithmetic, depending on RE.

The page table entry indicates the availability of the page and contains the high-order bits of the real address. The entry is either 2 or 4 bytes long depending on RE.

Instructions Modified by 31 bit Storage Addressing

PSW specification exception recognition is modified because previously unassigned bit 4 of the EC mode PSW is assigned as AE - the CPU storage extension control bit.

LOAD PSW (LPSW)
SET SYSTEM MASK (SSM)
STORE THEN "OR" SYSTEM MASK (STOSM)

Immediately after loading a new PSW or loading or modifying the system mask, a specificaion exception is recognized and a program interruption occurs if the resulting PSW specifies EC mode and

the 31 bit storage addressing feature is not installed and the contents of PSW bit positions 0, 2-4, 16-17, and 24-31 are not all zeros; or

the 31 bit storage addressing feature is installed and the contents of bit positions 0, 2-3, 16-17, and 24-31 are not all zeros.

Register or storage location usage is modified, depending on AE.

BRANCH ON CONDITION (BCR)
BRANCH ON COUNT (BCTR)

The contents of the general register specified by the R2 field are used as the branch address

24 bit address is located in bits 8-31 and bits 0-7 are ignored.
31 bit address is located in bits 1-31 and bit 0 is ignored.

LOAD ADDRESS (LA)

The second operand address is inserted in the general register specified by the R1 field.

24 bit address placed in bits 8-31 and bits 0-7 are set to zeros.
31 bit address placed in bits 1-31 and bit 0 is set to zero.

MONITOR CALL (MC)

The first operand address forms the monitor code and is placed in the word at real storage location 156 (byte locations 156-159).

COMPARE LOGICAL LONG (CLCL)
MOVE LONG (MVCL)

The location of the left-most byte of the first and second operands is designated by the general registers specified by the R1 and R2 field respectively.

24 bit address is contained in bits 8-31, bits 0-7 are ignored.
31 bit address contained in bits 1-31, bit 0 is ignored.

When the instruction is interrupted or completed, the incremented operand address values occupy the R1 and R2 register.

24 bit address is contained in bits 8-31 and bits 0-7 are set to zero.
31 bit address is contained in bits 1-31 and bit 0 is set to zero.

TRANSLATE AND TEST (TRT)

When the function byte is non-zero, the related argument address, which points to the argument last translated is placed in general register 1.

24 bit argument address placed in bits 8-31, bits 0-7 are set to zero.
31 bit argument address placed in bits 1-31, bit 0 is set to zero.

EDIT AND MARK (EDMK)

If the significance indicator was off before examination of the result character and the character is a zoned source digit, its address is placed in general register 1.
24 bit character address placed in bits 8-31, bits 0-7 are set to zero.
31 bit character address placed in bits 1-31, bit 0 is set to zero.

Register usage is modified depending on the CPU real address size control parameters.

INSERT STORAGE KEY (ISK)
SET STORAGE KEY (SSK)

A 2K byte block of real storage is designated by the general register specified by the R2 field.

24 bit address: Bits 8-20 address the 2K byte block. Bits 0-7 and 21-27 are ignored. Bits 28-31 must contain zeros or a specification exception is recognized.
31 bit address: Bits 1-20 address the 2K byte block. Bits 0 and 21-27 are ignored. Bits 28-31 must contain zeros or a specification exception is recognized.

RESET REFERENCE BIT (RRB)

A 2K byte block of real storage is designated by the second operand address.
24 bit address: Bits 8-20 address the 2K byte block. Bits 0-7 and 21-31 are ignored.
31 bit address: Bits 1-20 address the 2K byte block. Bits 0 and 21-31 are ignored.

Operand usage and register usage are modified depending on AE and RE respectively.

LOAD REAL ADDRESS (LRA)

The second operand address is either a 24 or a 31 bit logical address, depending on AE, and is translated by means of the dynamic address translation facility (without use of the TLB) regardless of whether translation is specified and regardless of whether the PSW specifies BC or EC mode.

The result is either a 24 or a 31 bit real address, depending on RE, and is one of three possible addresses.

1. Real address which results from successful translation.
2. Real address which designates the location of the segment table entry if it falls outside the segment table or is invalid
3. Real address which designates the location of the page table entry if it falls outside the page table or is invalid.

In each case the result is placed in the general register designated by the R1 field.

24 bit address: Placed in bits 8-31 and bits 0-7 are set to zero.
31 bit address: Placed in bits 1-31 and bit 0 is set to zero.

BRANCH AND STORE FEATURE

The branch and store feature provides users of the 31 bit storage addressing feature instructions which store a 31 bit link address in a general register.

Instruction	Mnemonic	Type	Exceptions*	Code
Branch and store	BASR	RR	0	0D
Branch and store	BAS	RX	0	4D

Control

The branch and store feature is controlled by feature control register bit 5 (BS).

BS=1 Feature is enabled and BAS, BASR are valid
BS=0 Feature is disabled or not installed

Instruction Definition

BRANCH AND STORE

```
 BASR      R1,R2          <RR>
+---------+----+----+
¦   0D    ¦ R1 ¦ R2 ¦
+---------+----+----+
 0         8    12 15
```

```
 BAS       R1,D2(X2,B2)   <RX>
+---------+----+----+----+--------------+
¦   4D    ¦ R1 ¦ X2 ¦ B2 ¦      D2      ¦
+---------+----+----+----+--------------+
 0         8    12   16   20          31
```

* 0 operation exception, if branch and store feature not enabled

This instruction

* first computes the branch target address,
* then performs the store operation by placing the next sequential instruction address in the general register designated by R1 and
* finally executes the branch by replacing the next sequential instruction address of the PSW with the branch target address.

In the RX format, the second operand address is the branch target address, while in the RR format the contents of the general register designated by R2 is the branch target address. In the latter case, if the R2 field contains zeros, the branch is not executed, but the store operation is still performed.

The instruction length code is 1 for the RR format and 2 for the RX format.

When the processor is either in BC mode or in EC mode with CPU 24 bit address generation specified, bit positions 8-31 of the R1 register are updated by the store operation and zeros are placed in bit positions 0-7. The RX format branch target address is the contents of bit positions 8-31 of the R2 register and bits 0-7 are ignored.

When 31 bit address mode is specified, bit positions 1-31 of the R1 register are updated by the store operation and bit 0 is set to zero. The RX format branch target address is the contents of bits 1-31 of the R2 register and bit 0 is ignored.

Condition Code:

The code remains unchanged.

Program Exceptions:

Operation (if the Branch and Store instruction feature is not installed).

Programming Note:

If the branch and store feature is enabled, BAS and BASR instructions are valid in BC mode and EC mode whether or not the 31 bit storage addressing feature is enabled.

SELECTIVE TLB PURGE FEATURE

The selective TLB purge feature provides instructions which purge either single entries or single address spaces from the translation lookaside buffer (TLB).

Instruction	Mnemonic	Type	Exceptions*	Code
Purge page	PPG	S	M,O	B2F0
Purge single user	PSU	S	M,O	B2F1

Control

The selective TLB purge feature is controlled by feature control control register bit 6 (PG).

PG=1 Feature is enabled. PPG and PSU are valid.
PG=0 Feature is disabled or not installed.

Instruction Definition

PURGE PAGE

```
PPG         D2(B2)          <S>
+-------------------+----+--------------+
|        B2F0       | B2 |      D2      |
+-------------------+----+--------------+
0                   16   20            31
```

All translation lookaside buffer (TLB) entries which contain the real page address designated by the second operand are made invalid. No change is made to addressable storage or registers. The TLB appears cleared of the original contents of these entries for all following instructions.

If no TLB entries containing the real page address exist, the instruction is equivalent to a no-operation.

* M - privileged operation exception
O - operation exception, if selective purge feature not enabled

A serialization function is performed. CPU operation is delayed until all previous accesses by the CPU to main storage have been completed, as observed by the channels. No subsequent instructions, their operands, or dynamic address translation entries are fetched until the execution of this instruction is complete.

Condition Code:

The code remains unchanged.

Program Exception:

Operation if selective purge feature is not enabled.
Privileged operation if selective purge feature is enabled.

PURGE SINGLE USER

```
     PSU                      <S>
    +-------------------+--------------------+
    |        B2F1       |////////////////////|
    +-------------------+--------------------+
     0                  16                 31
```

All entries in the translation lookaside buffer associated with the current translation parameters of control registers 0 and 1 are made invalid. No change is made to addressable storage or registers. All such entries appear cleared of their original contents for all following instructions. Wben no such entries exist in the TLB, the instruction is equivalent to a no-operation.

A serialization function is performed. CPU operation is delayed until all previous accesses by the CPU to main storage have been completed, as observed by channels. No subsequent instructions, their operands, or dynamic address translation entries are fetched until the execution of this instruction is complete.

Bits 16-31 of the instruction are ignored.

Condition Code:

The code remains unchanged.

Program Exception:

Operation if selective purge feature not enabled.
Privileged operation if selective purge feature is enabled.

4K BYTE PROTECTION FEATURE

This feature provides the ability to protect 4096 byte blocks of real storage. Each block has an address that is a multiple of 4096 and a single key is associated with each block. The key format, the protection action and the types of accesses subject to protection remain the same as that defined for the standard 2K byte protection.

Control

When the 4K byte protection feature is installed it is controlled by feature control register bit 7 (PE).

PE=1	Feature is enabled
PE=0	Feature is disabled or not installed

Even though the 4K byte portection feature is enabled, it is inoperative if the processor is in translation mode with 2K byte page size specified (EC PSW bit 5=1 and control register 0 bits 8-9=01).

Instruction Modification

When the feature is enabled and operative, INSERT STORAGE KEY, SET STORAGE KEY and RESET REFERENCE BIT instructions logically access a single key associated with a 4096 byte block of real storage. Block addressing for each instruction is detailed below.

INSERT STORAGE KEY
SET STORAGE KEY

Bits 8-19 (1-19 for a 31 bit address) of the register designated by the R2 field address a 4096 byte block of real storage. Bits 0-7 (just 0 for a 31 bit address) and 20-27 are ignored. Bits 28-31 must be zero; otherwise a specification exception is recognized.

RESET REFERENCE BIT

Bits 9-19 (1-19 for a 31 bit address) of the second operand address designate a 4096 byte block of real storage. Bits 0-7 (just 0 for a 31 bit address) and 20-31 are ignored.

SET KEY AND CLEAR FEATURE

This feature allows the SET STORAGE KEY (SSK) instruction to optionally set the key in storage and clear the associated real storage block, or just set the key in storage. Bits 28-31 of the R2 register specify the option.

The "set key and clear" operation places the contents of the R1 register into the key in storage which as associated with the block designated by the R2 register, and stores zeros in all byte locations of the designated real storage block.

Control

When the set key and clear feature is installed, it is controlled by feature control register bit 0 (KC).

KC=1	Feature is enabled
KC=0	Feature is disabled or not installed

Specification

When the set key and clear feature is enabled, bits 28-31 of the register designated by the R2 field of the SSK instruction define the operation to be performed. If bits 28-31 contain zeros, the standard SSK instruction (as defined by *IBM System/370 Principles of Operation* and modified by 470V/7 features - 31 bit storage address and 4K byte protection) is executed. If bits 28-31 have a non-zero value, the set key and clear operation is executed. The definition of the remainder of the R2 register contents as well as the R1 register contents is unchanged.

When the feature is disabled, bits 28-31 of the R2 register must contain zeros or a specification exception is recognized.

HARDWARE MEASUREMENT INTERFACE

The hardware measurement interface (HMI) facility provides the 470V/7 with an interface suitable for attaching a commercial hardware monitor. HMI is fully specified in "470V/7 Hardware Measurement Interface Functional Architecture Specification" P/N 820105-600. Two instructions under the diagnose code are provided for processor - HMI communication.

Instruction	Mnemonic	Type	Exceptions*	Code
Load HMI	LHMI	S	M,A	8303
Store HMI	STHMI	S	M,A	8304

Instruction Definition

LOAD HARDWARE MEASUREMENT INTERFACE

```
LHMI      D2(B2)          <S>
+--------------------+----+--------------+
|        8303        | B2 |      D2      |
+--------------------+----+--------------+
0                    16   20            31
```

The word designated by the second operand address is made available to the Hardware Measurement Interface (HMI). Particular implementations of HMI may utilize less than the full word, however accessing exceptions are checked for the entire 4 bytes of the second operand. Recognition of an accessing exception causes the operation to be suppressed.

Resulting Condition Code

0 HMI loaded successfully

1 Data not transferred because data previously transferred to HMI has not yet been used

2 Data not transferred because HMI detected an error in previously transferred data

3 HMI not installed

* M - privileged operation
A - accessing exception

Program Exceptions

Privileged operation
Access (fetch, operand2)

STORE HARDWARE MEASUREMENT INTERFACE

```
    STHMI    D2(B2)        <S>
  +--------------------+----+--------------+
  |         8304       | B2 |      D2      |
  +--------------------+----+--------------+
   0                   16   20           31
```

A word from the Hardware Measurement Interface (HMI) is stored in the location designated by the second operand address. Particular implementations of HMI may provide less than a full word, however accessing exceptions are checked for the entire 4 bytes of the second operand. Recognition of an accessing exception causes the operation to be suppressed.

Resulting Condition Code:

0 HMI data stored and valid

1 HMI data stored but not valid

2 -

3 HMI not installed

Program Exception:

Privileged operation
Access (store, operand 2)

Programming Note:

Although a particular HMI implementation may provide zeros in undefined bit positions of operand 2, the program should not depend on such zeros.

DYNAMIC ADDRESS TRANSLATION PROCESS

Dynamic address translation may be specified for instruction and data addresses generated by the CPU; and for data and control word addresses generated by the channels. Logical addresses are translated into real addresses using the appropriate segment and page table entries. The location and format of these entries depends on the address translation parameters, the maximum real address size and the logical address itself.

CPU translation control parameters are located in control registers (CR) 0 and 1; and the channel translation control parameters are located in the subchannel translation words (STW). When translation is active, the maximum CPU real address size is specified by CR0 bit 10 and the maximum subchannel real address size is specified by STW bit 28.

Translation Control Parameters

Segment table length (LEN)

CPU control	CR1 bits 0-7
Channel control	STW bits 0-7

LEN+1 is the number of 64 byte blocks in the segment table. Thus the length in bytes is given by (LEN+1)*64.

Segment table origin (STO)

CPU control	CR1 bits 8-25
Channel control	STW bits 8-25

STO with 6 low-order zeros appended forms a 24 bit real address which designates the beginning of the segment table. The table begins on a 64 byte boundary within the low-order 16M bytes of real storage.

Segment size (SS)

CPU control	CR0 bits 11-12
Channel control	STW bits 28-29

SS designates the size of a logical segment.

00	64K bytes
10	1M byte
01,11	invalid states

Page size (PS)

CPU control	CRO bits 8-9
Channel control	STW bits 26-27

PS designates the size of a logical page.

01	2K bytes
10	4K bytes
00,11	invalid states

Segment Table Entry

The segment table entry designates the availability and the logical accessability of the segment as well as designating the origin and length of the corresponding page table. A segment table entry has the following format:

```
+-----+-----+--------------------+-----+---+
| Len |/////| Page Table Origin  |/////| I |
+-----+-----+--------------------+-----+---+
0     4     8                    29    31
```

Page Table Length (Len): bits 0-3

Len+1 divided by sixteen is the fraction of the maximum page table length which is available, where the maximum length depends on both the segment size and the page size.

Page Table Origin (PTO): bits 8-28

PTO with 3 low-order zeros appended forms a 24 bit real address which designates the beginning of the page table. The table must begin on an 8 byte boundary within the low-order 16M bytes of real storage.

Segment Invalid (I): bit 31

0 - segment available, translation may proceed using the designated page table
1 - segment not available

Bits 4-7 and 29-30 are ignored

Page Table Entry

The page table entry indicates the availability of the page, and contains the high-order bits of the real address. The entry has four possible formats depending on the logical page size and the real address size.

24 bit real address (RE=0)

```
                   +--------------------+--+--+--+
                   ¦8-------RPA-------20¦PI¦CI¦//¦  2K byte pages
                   +--------------------+--+--+--+
                    0                    13 14 15

                   +-----------------+--+--+--+--+
                   ¦8-------RPA----19¦PI¦0 ¦CI¦//¦  4K byte pages
                   +-----------------+--+--+--+--+
                    0              12 13 14 15
```

31 bit real address (RE=1)

```
    +----------+-------------------------+--+--+--+
    ¦//////////¦1--------RPA-----------20¦PI¦CI¦//¦  2K byte pages
    +----------+-------------------------+--+--+--+
     0         9                         29 30 31

    +----------+----------------------+--+--+--+--+
    ¦//////////¦1--------RPA--------19¦PI¦0 ¦CI¦//¦  4K byte pages
    +----------+----------------------+--+--+--+--+
     0         9                    28 29 30 31
```

Real page address (RPA)

2 byte format:	2K page	0-12;	4K page	0-11
4 byte format:		9-28;		9-27

These bits provide the high-order portion of the real address. The entire real address is obtained by appending the contents of the byte-index field of the logical address to RPA.

CPU page invalid (PI)

2 byte format:	2K page	13;	4K page	12
4 byte format:		29;		28

0 - page available to CPU processes. Real addresses may be formed using RPA.

1 - page not available to CPU processes.

Channel page invalid (CI)

2 byte format:	2K page	14;	4K page	14
4 byte format:		30;		30

If the channel dynamic address translation feature is enabled, this bit defines the availability of the page to Channel processes.

0 - page available to channel processes. Real addresses may be formed using RPA.
1 - page not available to channel processes.

If the channel dynamic address translation feature is not enabled, this bit is defined to be zero.

Other bits defined to be zero

2 byte format:	2K page	--;	4K page	13
4 byte format:		--;		29

Unassigned bits

2 byte format:	2K page	15;	4K page	15
4 byte format:		0-8,31;		0-8,31

Translation Process

Translation from a logical address to a real address is performed by means of a segment table and a page table. CPU translation is specified by EC mode PSW bit 5 and controlled by parameters of control registers 0 and 1. Channel translation is indicated when the requesting subchannel is in logical address mode. This mode results when a channel program executes a transfer virtual and lock (TVL) command when channel dynamic address translation is active. The translation is controlled by parameters of the subchannel translation word.

Preliminary checking

When a CPU logical address requires translation, control register 0 bits 8-12 are examined. If an invalid code is contained in either bits 8-9 or bits 11-12, a translation specification exception is recognized. All possible STW specification exceptions were detected during execution of the TVL command, and hence were reported at that time via a program check condition in the CSW.

Segment table access

The segment table entry address is computed in the following manner:

a. Append 6 low-order zeros to the segment table origin forming the 24 bit origin address (bits 8-31),
b. align the low-order bit of the segment index portion of the logical address with bit 29 of the origin address just formed, and
c. add the above quantities as binary integers. Note the resulting address designates a location on a word boundary.

The addition may provide an address greater than 2**24-1 if 31 bit real addressing is available to the initiator of the request.

The segment index is compared against the segment table length to establish whether the addressed entry is within the table. The low-order 4 bits of the segment index are ignored and the value of the remaining segment index bits is checked to be less than or equal to the length value. If it is not, the addressed entry lies outside the defined table. A CPU initiated access causes a segment translation exception; while a channel initiated access causes translation specification and program check conditions to be set in the extended CSW.

If the computed segment table entry address of a CPU initiated access lies outside installed real storage, an addressing exception is recognized. If a channel caused segment table entry access lies outside installed real storage, a program check condition is set in the CSW.

The segment table entry address is used to fetch a 4 byte word from real storage. Bit 31 - the segment invalid bit - indicates the availability of the corresponding page table. If the bit is a one, the segment is invalid and the page table is unavailable. In this case, a CPU initiated access causes a segment translation exception; while a channel initiated access causes segment invalid and program check conditions to be set in the extended CSW.

Page table access

The origin and length of the page table are specified by the appropriate fields of the segment table entry. The page table entry address computation depends on whether 24 bit or 31 bit real addressing is specified.

Three low-order zeros are appended to the origin forming the 24 bit page table origin address (bits 8-31). When 24 bit real addressing is specified, page table entries are 2 bytes long, therefore the low-order bit of the page index portion of the logical address is aligned with bit 30 of the origin address, a binary addition is performed ignoring overflow and the 24 bit result is the page table entry address.

When 31 bit real addressing is specified, the page table entry is 4 bytes long. In this case, the low-order bit of the page index is aligned with bit 29 of the origin address, and the addition is performed retaining any overflow. The resulting address may be greater than 2**24-1.

The page index is compared against the page table length to establish whether the addressed entry is within the table. The value of the high-order 4 bits of the page index is checked to be less than or equal to the length value. If it is not, the entry lies outside the page table. A CPU initiated access causes a page translation exception; while a channel initiated access causes translation specification check and program check conditions to be set in the extended CSW.

If the computated page table entry address of a CPU initiated access lies outside installed real storage, an addressing exception is recognized and the operation is suppressed. If a channel caused page table entry access lies outside installed real storage, a program check condition is set in the CSW.

The page table entry address is used to fetch either a 2 byte or a 4 byte entry from real storage. The appropriate invalid bit of the entry is inspected and if it is a one the page is unavailable. A CPU initiated access causes a page translation exception; while a channel initiated access causes page invalid and program check conditions to be set in the extended CSW.

Bit positions which are defined to be zero are inspected. If a one is detected and the access was CPU initiated, a translation specification exception is recognized; while if the access was channel initiated, translation sepcification check and program check conditions are set in the extended CSW.

Real address formation

The real page address, obtained from the page table entry, and the byte index portion of the logical address are concatenated to form either a 24 bit or a 31 bit real address depending on which address length is specified for the request.

Summary of Translation Error Conditions

Error conditions which may occur during the translation process are compiled below. For each condition, the action taken, depending on the translation initiator - CPU or channel, is specified.

Control parameter specification error

CPU control parameter errors which are recognized when an attempt is made to use the parameters are

page size (PS) - invalid state
segment size (SS) - invalid state

A translation specification exception is recognized and the operation suppressed

Channel control parameters in the subchannel translation word are checked for errors during the execution of a TVL command. No further checking takes place when channel address translation actually occurs.

Segment table entry outside defined table

CPU - segment translation exception recognized and the operation is nullified.
Channel - translation specification check and program check conditions stored in extended status byte and channel status portions respectively, of CSW.

segment table entry outside available main storage

CPU - addressing exception recognized and the operation is suppressed.
Channel - program check stored in CSW.

Segment unavailable: segment table entry bit 31 is a one

CPU - segment translation exception recognized and the operation is nullified.
Channel - segment invalid and program check conditions

stored in extended CSW.

Page table entry outside defined table

CPU - page translation exception recognized and the operation is nullified.
Channel - translation specification check and program check conditions are stored in the extended CSW.

Page table entry address outside available main storage

CPU - addressing exception recognized and the operation is suppressed.
Channel - program check stored in CSW.

Page unavailable because appropriate page table entry invalid bit is a one

CPU - page translation exception recognized and the operation is nullified.
Channel - page invalid and program check conditions stored in extended CSW.

Page table entry specification error: bits defined to be zero contain ones.

CPU - translation specification exception recognized and the operation is nullified.
Channel - translation specification check and program check condition stored in extended CSW.

Programming note:

Segment and page table entry accesses to real storage are not subject to memory protection.

When an error is detected during a CPU initiated translation, the operation is either nullified or suppressed. If a translation error condition (translation specification, segment invalid, page invalid) occurs during fetching of the CCW which initiates an I/O operation, the I/O operation is not initiated. If an error occurs after the I/O operation has been initiated, the channel signals the I/O device to terminate the operation when a request to transfer the next data byte is submitted from the I/O device. Command chaining is suppressed when translation errors occur. The CSW contains the address of the last executed CCW, plus eight. The CSW count field is unpredictable.

CYCLE COUNTER

The cycle counter provides two capabilities for the user of a 470V/7 system. The first and normal mode of operation makes available to the system programmer a value which is decremented every processor cycle. When this value reaches zero, it rolls over to its highest value and continues while processor operation proceeds normally. The second mode of operation allows the user to stop the 470V/7 clocks (gated clocks only) a predetermined number of cycles after a preset reference point has been passed.

Format

The cycle counter is a 32 bit word having the following format -

```
+----+--------------------+
| ab |       count        |
+----+--------------------+
 0    2                  31
```

Bits 0-1 (ab) are the "starting condition" code. These bits encode an event which defines the "preset reference point" mentioned above.

ab

00 - not armed, cycle counter is inoperative
01 - address compare event, decrementing begins when an address compare pulse is received from the addrss compare circuitry
10 - process state event, decrementing begins when the CPU enters PROCESS STATE
11 - immediate event, decrementing begins on the cycle immediately after the counter is loaded.

Bits 2-31 contain a 30 bit positive binary value. When initally loaded this value is the "predetermined number of cycles" mentioned above. When the starting condition occurs, this is the value which is decremented.

Operation

The cycle counter mode of operation is controlled by bit 7 of the Clock Control State Register (CCSR) - the "cycle counter clock stop enable" bit.

0 - continuous decrement mode (this is the normal mode of operation)
1 - clock stop mode

The entire CCSR is loaded by the Load Clock Control State Register command from the console processor.

Once the cycle counter is loaded with a start code other than "00" the occurance of the event specified by the start code causes decrementing to begin. The count contained in bits 2-31 is reduced by one every cycle thereafter until the count equals zero. Cycle counter action at this point depends on the mode of operation. If the mode is "continuous decrement", the underflow indicator is set and the decrementing process continues with the value 2**30-1 immediately following zero.

If the mode is "clock stop", gated clock propagation to the entire system is halted. This action causes propagation of a Clock Stop interrupt to the system console. The underflow indicator is not set in this mode of operation.

Underflow Indicator

As noted above, the underflow indicator is set when the cycle counter is in "continuous decrement" mode and the count passes through zero. Once the underflow indicator is set it remains pending until reset by execution of a SET CYCLE COUNTER instruction, a STORE CYCLE COUNTER instruction or a Cycle Counter Reset command from the console processor. Other general system or CPU resets also cause underflow reset.

Error Condition

The cycle counter is self-checking. The decrementor predicts byte parity based on the input value and and parity is independently generated from the decrementor output value. If the generated parity and the predicted parity differ, the error indicator is set and at the same time an external damage pulse is transmitted to the CPU.

Once the <u>error</u> indicator is set it remains pending until specifically reset. The reset conditions are the same as those specified for the <u>underflow</u> indicator.

If the processor is not masked for the external damage class of interrupts, both the machine check interrupt code (MCIC) indication "external damage" and the region code indicating "cycle counter error" are stored.

<u>Loading the Cycle Counter</u>

The cycle counter is loadable by either the 470V/7 SET CYCLE COUNTER instruction or the console processor Load Cycle Counter command. SET CYCLE COUNTER can be executed during normal supervisor state operation of the 470V/7 processor. The console processor Load Cycle Counter command is sent through the CPU-console Interface Processor hardware and is executed whether of not 470V/7 processor clocks are running.

SET CYCLE COUNTER

```
  SCC         D2(B2)          <S>
+-------------------+----+--------------+
|       8305        | B2 |      D2      |
+-------------------+----+--------------+
 0                  16   20           31
```

The current contents of the cycle counter are replaced by the contents of the word designated by the second operand address. This word has the format described in the "format" section of this chapter.

After the cycle counter value is set, the clock is placed in the stopped state and reamins stopped until the event defined by the "start code" occurs. The cycle counter then enters the run state and decrements as described in the "Operation" section of this chapter.

A serialization function is performed. CPU operation is delayed until all previous accesses by the CPU to main storage have been completed, as observed by the channels. No subsequent instructions, their operands or dynamic address translation entries are fetched until execution of this instruction is complete.

Recognition of protection or accessing exceptions causes the operation to be supperessed.

The cycle counter is loaded regardless of any pending underflow or error conditions, however the condition code is set accordingly. As proviously noted, pending error or underflow conditions are reset by execution of this instruction.

Condition Code:

0 - no conditions pending

1 - pending underflow

2 - pending error

3 -

Program Exceptions:

Privileged Operation
Access (fetch, operand 2)

Load Cycle Counter command

This console processor command loads the cycle counter, however unlike the SET CYCLE COUNTER instruction, does not reset pending underflow or error conditions. For more details refer to "470V/7 Console Operation Specification".

Examining the Cycle Counter

The cycle counter can be examined by either the 470V/7 STORE CYCLE COUNTER instruction or the scan-out process of the system console diagnostic mode. STORE CYCLE COUNTER can be executed during normal supervisory state operation of the 470V/7 processor.

STORE CYCLE COUNTER

```
STCC       D2(B2)          <S>
+-------------------+----+--------------+
|        8306       | B2 |      D2      |
+-------------------+----+--------------+
0                   16   20            31
```

The current count value and starting condition of the cycle counter are stored in the word designated by the second operand address. The stored word has the format described in the "Format" section of this chapter.

A serialization function is performed. CPU operation is delayed until all previous accesses by the CPU to main storage have been completed, as observed by the channels. No subsequent instructions, their operands or dynamic address translation entries are fetched until execution of this instruction is complete.

The condition code is set indicating pending underflow or error conditions. Similiarly to the SET CYCLE COUNTER instruction, pending conditions are reset by execution of this instruction.

<u>Condition</u> <u>Code</u>:

0 - no conditions pending

1 - pending underflow

2 - pending error

3 -

<u>Program</u> <u>Exceptions</u>:

Privileged Operation
Access (store, operand 2)

<u>Programming</u> <u>Note</u>:

In the normal, continuous decrement mode of operation, the cycle counter can provide a high resolution timer for the system programmer. A SET CYCLE COUNTER instruction with either "process state" or "immediate" as the start condition will allow immediate cycle counter decrementing (the processor is in process state when the instruction is executed). If the full value of 2**30-1 is loaded, about 28-32 seconds (depending on the processor cycle time) elapses before underflow is reached.

As previously noted, the underflow condition remains pending unless directly (by console processor command) or indirectly (by 470V/7 cycle counter instruction) reset. There is no way to tell whether the cycle counter has passed through zero one time or multiple times between STORE CYCLE COUNTER executions other than inherently knowing the absolute elapsed time range.

Cycle Counter Relationship to DIAGNOSE STOP

Execution of the DIAGNOSE STOP instruction will not change the cycle counter mode of operation, however it does cause the start code to be reset. If the cycle counter is decrementing when the start code is reset, decrementing is halted. If the counter is not decrementing because a previously loaded start event has not yet occurred, the start event is lost; however the count remains unchanged. In either case, if cycle counter use is desired after execution of DIAGNOSE STOP, it must be reloaded.

DIAGNOSE OPERATIONS

The DIAGNOSE operations are all S format instructions as follows:

```
+-------------------+----+--------------+
¦      OP (83xx)    ¦ Bx ¦      Dx      ¦
+-------------------+----+--------------+
 0                   16   29           31
```

where the hexidecimal op-code '83xx' specifies the exact operation to be performed.

All DIAGNOSE operations cause a serialization function. CPU operation is delayed until all previous accesses by the CPU to main storage have been completed, as observed by channels. No subsequent instructions, their operands, or dynamic address translation entries are fetched until the execution of this operation is complete.

All DIAGNOSE operations are privileged instructions.

Instruction List

All defined DIAGNOSE functions for 470V/7 are listed below:

Instruction	Mnemonic	Type	Exceptions*	Code
Diagnose no-operation	DNOOP	S	M	8300
Load feat ctrl reg	LFCR	S	M,A	8301
Store feat ctrl reg	STFCR	S	M,A	8302
Load HMI	LHMI	S	M,A	8303
Store HMI	STHMI	S	M,A	8304
Load cycle counter	LCC	S	M,A	8305
Store cycle counter	STCC	S	M,A	8306
Store ECC	STECC	S	M,A	8308
Release line	RLSL	S	M	830E
Diagnose stop	DSTOP	S	M	83EB
Undefined stop	USTOP	S	M	83yy**

Notes * M - Privileged operation exception
A - Accessing exception
** yy can be any of the following hex values:

07, 0B, 0F
13, 23, ... F3
17, 27, ... F7
1B, 2B, ... DB, FB

1F, 2F, ... FF

8333 is defined to be the universal "undefined stop".

Undefined DIAGNOSE op-codes are denoted by 83zz where zz is any hex value not defined in the above list. Executing a DIAGNOSE with an undefined op-code gives unpredictable results. The state of the CPU, the contents of registers or storage locations as well as the progress of an I/O operation could be affected. The resulting state of the conditon code is also unpredictable. (See also IBM System/370 Principles of Operation, "Diagnose", pp 103,4).

Instruction Specification

DIAGNOSE NO-OPERATION

```
DNOOP                   <S>
+-------------------+-------------------+
|        8300       |///////////////////|
+-------------------+-------------------+
0                   16                31
```

A serialization function is performed. CPU operation is delayed until all previous accesses by the CPU to main storage are completed, as observed by channels. No subsequent instructions, their operands, or dynamic address translation entries are fetched until the executuion of this instruction is complete.

Bits 16-31 of the instruction are ignored.

Condition Code: The code remains unchanged.

Program Exception: Privileged operation.

LOAD FEATURE CONTROL REGISTER
STORE FEATURE CONTROL REGISTER

These two instructions are part of the feature control register support and are defined in the chapter "Feature Control Register".

LOAD HARDWARE MEASUREMENT INTERFACE
STORE HARDWARE MEASUREMENT INTERFACE

These two instructions are part of the hardware measurement interface feature and are defined in the chapter "Hardware Measurement Interface".

LOAD CYCLE COUNTER
STORE CYCLE COUNTER

These two instructions are part of the cycle counter feature and are defined in the "Cycle Counter" chapter.

STORE ERROR CORRECTION CODE

```
  STECC     D2(B2)           <S>
+-------------------+----+--------------+
|        8308       | B2 |      D2      |
+-------------------+----+--------------+
 0                   16   20          31
```

The error correction codes of the most recent line transferred from main store to the high speed buffer are stored at the word location designated by the second operand address. An accessing exception causes the operation to be suppressed.

Condition Code: The code remains unchanged.

Program Exceptions:

Privileged operation
Access (store, operand 2)

RELEASE LINE

```
  RLSL      D2(B2)           <S>
+-------------------+----+--------------+
|        830E       | B2 |      D2      |
+-------------------+----+--------------+
 0                   16   20          31
```

A line (32 bytes starting on a 32 byte boundary) is released from the buffer. The buffer is searched for the line designated by the second operand address which is always a real address.

* If the line does not exist in the buffer, the instruction has no function and is equivalent to a no-operation.

- If the line exists in the buffer in an unmodified state, it is made invalid in the buffer (since the same state of the line exists in real storage).

- If the line exists in the buffer in a modified state, the entire line from the buffer is transferred to real storage. This is the "back-storing" process. The line is then invalidated in the buffer.

Bits 8-26 or 1-26, depending on whether or not 31 bit real addressing is specified, of the second operand address designate a real storage line address. Bits 27-31 are ignored.

Condition Code:

0 - line not present in buffer
1 - line was unmodified in buffer, therefore it has been invalidated.
2 - line modified in buffer, therefore it has been backstored to msinstore and invalidated in the buffer.
3 -

Program Exception: Privileged operation

DIAGNOSE STOP

```
 DSTOP                    <S>
+--------------------+--------------------+
|        83EB        |////////////////////|
+--------------------+--------------------+
 0                    16                31
```

A serialization function is performed. CPU operation is delayed until all previous accesses by the CPU to main storage have been completed, as observed by channels. No subsequent instructions, their operands or dynamic address translation entries are fetched.

The CPU enters STOP state and raises the "Diagnose Stop" signal to the CPU-Console Interface (CCI).

Bits 16-31 of the instruction are ignored.

Condition Code: The code remains unchanged.

Program Exceptions: Privileged operation.

CHANNEL EXTENSION

The Channel Extension feature provides facilities for extending the number of channels to 32.

Extended channels follow the specifications of the *IBM System/370 Principles of Operation* except that the channel address portion of the I/O address may specify channel addresses 16-31.

The feature is controlled by the "channel extension" bit (CE) - bit 4 of the feature control register.

CE=0	Channels 0-15 are valid
CE=1	Channels 0-31 are valid

FAILING STORAGE ADDRESS

Certain errors associated with 470V/7 storage cause a failing storage address (FSA) to be stored at real storage location 248. In general, errors associated with main storage - both addressable data and key-in-storage, errors associated with cache - both data and tag sections, and errors associated with the TLB cause FSA to be formed and stored.

The general FSA format is shown

```
+---+-----+------------------------+-----+
¦ z ¦ xxx ¦  quarter line address  ¦ yyy ¦
+---+-----+------------------------+-----+
 0   1     4                        29 31
```

The values of z, xxx and yyy depend on the type of error which caused FSA to be stored. These errors and the corresponding z, xxx and yyy values are listed below.

An error detected in data which was stored into the buffer by either a CPU or a channel process:

z is set to "0"
xxx is set to "000"
yyy designates the byte of the quarter line

An error detected while data is being transferred from the buffer to main storage

z indicates the kind of bit which failed

0 - either a data bit or a parity bit
1 - longitudinal redundancy code (LRC) bit

xxx - designates the failing LRC bit when z=1
- not defined when z=0

yyy - designates the failing byte when z=0
- not defined when z=1

A correctable error detected while data is being transferred from main storage to the buffer

z indicates the kind of bit which failed

0 - data bit
1 - error correction code (ECC) bit

xxx - designates the failing bit of the byte when z=0
- designates the failing ECC bit (in conjunction with yyy) when z=1

yyy designates the failing byte when z=0

when z=1, xxx and yyy are coded to designate the failing ECC bit as shown below

xxx	yyy	ECC bit
001	000	C1
010	000	C2
100	000	C4
000	001	C8
000	010	C16
000	100	C32
000	000	C64
000	110	CP

Multiple bit errors detected when transferring data from main storage to buffer; or other errors whih cause FSA to be stored.

z is set to "0"
xxx set to "000"
YYY set to "000"

the low order 2 bits of the quarter line address are unpredictable.

STORE CPU ID INSTRUCTION

The STORE CPU ID instruction (STIDP) places 8 bytes of information in the storage location designated by the second operand address. The format of the 8 bytes is

```
+----+--------------+---------+---------+
| 07 |    00SSSS    |  0470   |  0000   |
+----+--------------+---------+---------+
 0    8              32        48     63
```

Bits 0-7 contain the version code. This field is set to '07' to indicate 470V/7.

Bits 8-31 contain the serial number. This field indictes in BCD format the serial number of the processor. The SSSS characters begin at '0000' for the 470V/7 engineering model and continue with '0001', '0002', ... for production machines.

Bits 32-47 contain the model number. This field contains four decimal digits which are set to '0470', the same as all other 470 processors.

Bits 48-63 are unused and are stored as zeros.

MACHINE CHANGE LEVEL

The change level of each 470V/7 is entered into a permanent location of the console processor storage. This information can be altered only when the processor is in "CE mode".

REGION CODE

When a machine check interrupt occurs and the "machine check interrupt code" is stored, ten bytes called the "region code" are stored starting at main storage location 252. These bytes contain model dependent information which more specifically define the location of the error. Individual region code bit assignments are given below.

Region Code Bit Assignment

real storage location	bit	error source
252	0	I-Unit pipeline control error
	1	E-Unit condition code error
	2	E-Unit LUCK 1 byte 0 parity error
	3	E-Unit LUCK 1 byte 1 parity error
	4	E-Unit LUCK 1 byte 2 parity error
	5	E-Unit LUCK 1 byte 3 parity error
	6	E-Unit LUCK 2 byte 0 parity error
	7	E-Unit LUCK 2 byte 1 parity error
253	0	E-Unit LUCK 2 byte 2 parity error
	1	E-Unit LUCK 2 byte 3 parity error
	2	E-Unit Multiplicand byte 0 parity error
	3	E-Unit Multiplicand byte 1 parity error
	4	E-Unit Multiplicand byte 2 parity error
	5	E-Unit Multiplicand byte 3 parity error
	6	E-Unit adder high-input phase error
	7	E-Unit adder low-input phase error
254	0	S-Unit compare register parity error
	1	S-Unit tag - control parity error
	2	S-Unit tag - key parity error
	3	S-Unit tag - ID parity error
	4	S-Unit store data parity error
	5	Main store read address parity error
	6	Main store write key parity error
	7	Main store write address parity error

255	0	S-Unit LRC error
	1	S-Unit move-out parity error
	2-¦---	S-Unit encoded byte or MSU indication
	3-¦	
	4	reserved
	5	S-Unit multiple byte error
	6	S-Unit primary/alternate TLB indication
	7	S-Unit translation register parity error
256	0	E-Unit multiplier parity error
	1	E-Unit byte adder input 1 parity error
	2	E-Unit byte adder input 2 parity error
	3	E-Unit byte adder input 3 parity error
	4	S-Unit TLB - valid bit error
	5	S-Unit TLB - key parity error
	6	S-Unit TLB - logical address parity error
	7	S-Unit real address register parity error
257	0	I-Unit result byte 0 parity error
	1	I-Unit result byte 1 parity error
	2	I-Unit result byte 2 parity error
	3	I-Unit result byte 3 parity error
	4	I-Unit EAG parity error (low-order 3 bytes)
	5	I-Unit EAG parity error (high-order byte)
	6	I-Unit instruction stream entrance parity error
	7	I-Unit store data parity error
258	0	S-Unit TLB - SBR ID parity error
	1	S-Unit SBR stack error
	2	C-Unit 1 I/O address parity error (from IU)
	3	reserved
	4	C-Unit 1 error on CSW store
	5	C-Unit 2 I/O address parity error (from IU)
	6	reserved
	7	C-Unit 2 error on CSW store.
259	0	E-Unit multiplier residue error
	1	E-Unit adder residue error
	2	I-Unit instruction stream exit parity error
	3	S-Unit cache LRU error
	4	I-Unit control register bytes 0-1 parity error
	5	I-Unit control register bytes 2-3 parity error
	6	I-Unit PSW bytes 0-1 parity error
	7	Cycle Counter Parity Error

	0	S-Unit execution key parity error
	1-¦	
	2-¦---	S-Unit encoded cache column
260	3-¦	
	4	S-Unit compare register SBR ID parity error
	5-¦	
	6-¦---	S-Unit encoded port ID
	7-¦	
	0	Main store interface error
	1	Main store configuration register parity error
	2	Main store Ref/Chg address parity error
261	3	Main store Ref/Chg op bus parity error
	4	Main store Ref/Chg data parity error
	5	S-Unit real address compare register par err
	6	S-Unit main store address register parity error
	7	S-Unit general word register parity error