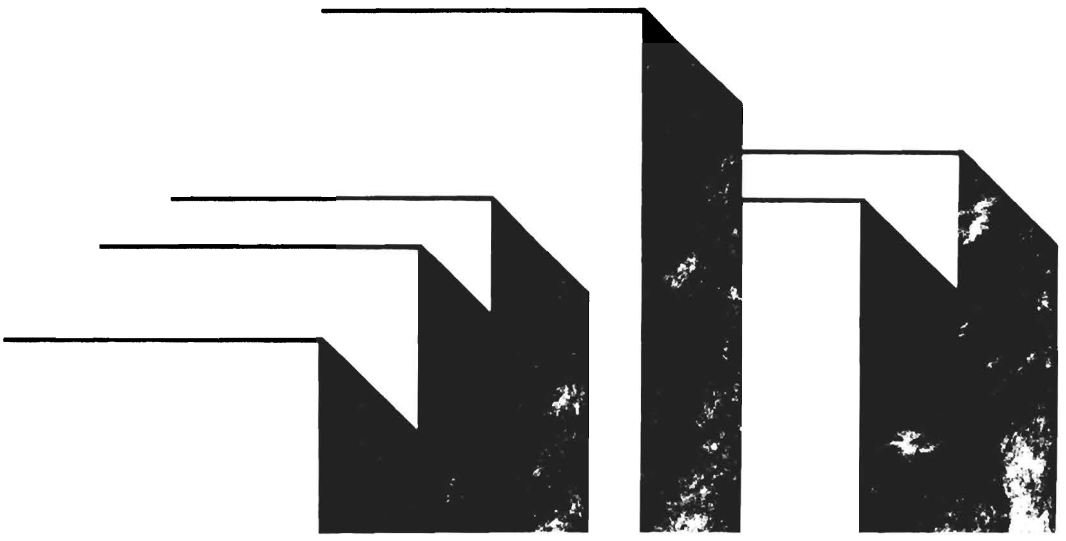
Publication Number
SA22-7095-1

IBM System/370 Extended Architecture

Interpretive Execution

IBM



IBM System/370 Extended Architecture

Interpretive Execution

Publication Number
SA22-7095-1

File Number
S370-01

Second Edition (September 1985)

This edition obsoletes SA22-7095-0. Text has been added to describe interception format 2. In addition, several sections have been clarified and corrected. Significant changes are identified by a vertical bar in the left margin.

Changes are made periodically to the information herein; before using this publication in connection with the operation of IBM equipment, refer to the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

PREFACE

This publication provides, for reference purposes, a definition of the machine functions performed by the System/370 extended-architecture (370-XA) interpretive-execution facility. It describes each function at the level of detail needed to prepare an assembler-level program that relies on that function. The information in this publication is provided principally for use by assembler-language programmers, although anyone concerned with the functional details of 370-XA systems operating in the interpretive-execution mode will find it useful.

The reader of this publication should become familiar with the IBM 370-XA Principles of Operation, SA22-7085, and, for the definition of a guest operating in the System/370 mode, the IBM System/370 Principles of Operation, GA22-7000. Terms and concepts are referred to in this publication which are explained in the Principles of Operation and which, for the most part, are not covered again in this publication.

This page is intentionally left blank.

CHAPTER 1. INTRODUCTION

This publication describes the interpretive-execution facility of the System/370 extended architecture (370-XA). The interpretive-execution facility improves the efficiency with which a specialized component of a control program can oversee the execution of programs in a virtual-machine environment. Execution of these programs is achieved through a combination of capabilities provided by the CPU while it is in the interpretive-execution mode and services supplied by supporting programs.

The interpretive-execution facility provides complete handling of many aspects of the architecture of an interpreted machine, or, when such handling is not provided, presents virtual-machine status in a form convenient for further support-program processing.

The interpreted machine is viewed as a virtual machine called the guest. Facilities appropriate to the mode of the guest, either System/370 mode or 370-XA mode, are provided when the CPU is in the interpretive-execution mode. The term host refers to the real machine and to the control program which both manages real-machine resources and provides services to the guest program or interpreted machine. The interpreted and host machines execute guest and host programs, respectively.

The interpretive-execution facility is invoked by executing the START INTERPRETIVE EXECUTION (SIE) instruction, which causes the CPU to enter the interpretive-execution mode and to commence execution of the guest program under control of the operand of the instruction, called the state description. Certain operations encountered in the guest cannot be performed in the interpretive-execution mode, and some may optionally have been designated to cause interpretive execution to be discontinued. In these cases, the CPU exits from the interpretive-execution mode, the execution of the START INTERPRETIVE EXECUTION instruction is completed, and the instruction in the host program that follows the START INTERPRETIVE EXECUTION instruction (or that follows an EXECUTE instruction, as appropriate) is designated as the next instruction to be executed. This process is called interception, and it includes saving the state of the guest in the state description and providing information about the reason for exiting from the interpretive-execution mode.

The CPU may also exit from the interpretive-execution mode by interrupting (a host I/O interruption, for example) execution of the START INTERPRETIVE EXECUTION instruction. The START INTERPRETIVE EXECUTION instruction is an interruptible instruction; that is, the parameters in the state description for continuing the execution of the guest are updated in such a way that if the host old PSW is loaded, causing the START INTERPRETIVE EXECUTION instruction (or an EXECUTE instruction, as appropriate) to be reexecuted, execution resumes at the interrupted point in the guest.

The state description specifies the type of system to be interpreted, the area of host storage representing guest main storage, the contents of some of the program-addressable guest registers, the addresses of related control tables, bits for controlling the operation of optional facilities, areas for displaying information concerning an interception, and information about other aspects of the operation.

Two methods of representing guest main storage are provided. In the pageable-storage mode, guest main storage consists of a portion of the host primary address space. That is, guest absolute addresses, to which an offset is added, are treated as host primary virtual addresses. In the preferred-storage mode, guest main storage consists of the first portion of host main storage. That is, guest absolute addresses are treated unmodified as host absolute addresses. With both methods, guest address-transformation mechanisms, including dynamic address translation and prefixing, are provided.

Guest key-controlled storage protection is provided, with guest storage keys set as the real-machine keys for the real-machine storage assigned to the guest. All other protection mechanisms are provided as well for the guest. In addition, if the pageable-storage mode is specified, then host page protection is applicable to guest references to guest main storage for both System/370 mode and 370-XA-mode guests.

When the pageable-storage mode is specified, a separate control area, called the reference-and-change-preservation (RCP) area, is provided for retaining the values of the reference bits and change bits that are separately applicable to the guest and to the host for each individual 4K-byte block of guest main storage. A true indication of the reference-and-change status can be

obtained by a logical OR of the appropriate bits from the RCP area and the reference and change bits in the real-machine storage key. Reference-and-change bits in the real-machine storage key, but not those in the RCP area, are set by references to storage.

Timing facilities are, for the most part, fully interpreted for the guest. A guest time-of-day (TOD) clock value is obtained by adding a constant found in the state description to the value of the host TOD clock. The guest CPU-timer and guest clock-comparator values are specified in the state description, with the function of these facilities provided in the interpretive-execution mode. Most instructions associated with the timing facilities are executed in the same manner in the interpretive-execution mode as when the CPU is not in that mode. When the System/370 mode is specified, the guest interval timer is also maintained and very nearly reflects the time spent in the interpretive-execution mode. Guest interruption processing associated with guest timing facilities is performed in part. The operation of host timing facilities is not affected by the interpretive-execution mode.

Most instructions are executed in the same manner in the interpretive-execution mode as when the CPU is not in that mode. In the interpretive-execution mode, however, some instructions are given special treatment, such as those that are:

- Not interpreted. Interception is mandatory for these instructions. This group includes most of the I/O instructions and certain other privileged instructions.
- Interpreted as operation exceptions. When the System/370 interpretive-execution mode is specified, this group includes 370-XA instructions that are invalid in the System/370 mode. Conversely, System/370 instructions that are invalid in the 370-XA mode are recognized as operation exceptions for 370-XA mode guests.
- Conditionally interpreted depending on the setting of control bits.
- Handled differently. This is generally because of special characteristics of the interpretive-execution environment. This group includes the storage-key-handling instructions.

Provision is made for handling guest I/O as follows:

- A means is provided for recording a pending guest I/O interruption and recognizing interception when the guest PSW is enabled for I/O interruptions.
- Interpretive execution of the TEST CHANNEL instruction is conditional and is modified from the native definition. Except for the TEST CHANNEL instruction, all I/O instructions and the channel-set-switching instructions cause interception, with information about the instruction provided in the state description at interception.

A means is provided for recording a pending guest external interruption; interception is recognized when the guest PSW is enabled for external interruptions. Guest program interruptions and guest supervisor-call interruptions are conditionally interpreted, depending on the setting of controls.

Interpretive execution can provide the effect of a guest shared-main-storage multiprocessing configuration. Multiple host CPUs can be simultaneously employed in this activity. Each guest CPU is defined by a separate state description, and each state description specifies the same guest main storage and (for pageable-storage-mode guests) RCP areas. An additional area called the system-control area (SCA) is used for control purposes; the same SCA is designated by each state description of the guest multiprocessing configuration.

On entry to the interpretive-execution mode, the contents of the state description are subjected to a number of checks for errors and for consistency. Thereafter, during interpretive execution, checking is performed with respect to items fetched from the state description. If an invalid condition is found, control is returned to the host program by means of a validity interception.

The following chapters give a detailed description of the operation of the CPU in the interpretive-execution mode. This description includes the definition of the START INTERPRETIVE EXECUTION instruction, a description of the format and contents of the state description, a discussion of guest storage (including the addressing mechanisms and the handling of storage keys), information about guest multiprocessing, and a description of the special handling of some instructions.

START INTERPRETIVE EXECUTION

SIE $D_2(B_2)$ [S]

The CPU is placed in the interpretive-execution mode and performs the functions of the interpreted machine, called the guest. The state description, which begins at the location designated by the second-operand address, contains information pertinent to the state of the guest. The CPU performing the interpretive execution by executing the START INTERPRETIVE EXECUTION instruction is called the host.

The CPU remains in the interpretive-execution mode until either interception or a host interruption occurs. Interception is recognized if conditions are encountered in the guest that cannot be handled in the interpretive-execution mode or for which special assistance is supplied by a host program. Interception consists in updating the state description to indicate the cause of the interception and to reflect the current state of the guest, leaving the interpretive-execution mode, and completing the execution of the START INTERPRETIVE EXECUTION instruction. Host interruption processing consists in updating the state description to reflect the current state of the guest, exiting from the interpretive-execution mode, storing host interruption parameters as required, and setting the instruction address in the old host PSW to designate the interrupted START INTERPRETIVE EXECUTION instruction (or EXECUTE, as appropriate).

Host program-event recording (PER) for instruction fetching applies to the fetch of the START INTERPRETIVE EXECUTION instruction and is indicated, if applicable, as for other interruptible instructions.

A serializing function and a checkpoint-synchronization function are performed on entry to and exit from the interpretive-execution mode.

Special Conditions

In the interpretive-execution mode, host PER monitoring for general-register alteration is not applied, and host PER monitoring for storage alteration is not applied to the state description, to the reference-and-change-preservation (RCP) area, to the system-control area (SCA), or to the host main-storage area defined as guest main storage. Host PER does not apply to the execution of guest instructions.

Host tracing does not apply to guest operations.

Host PSW bits 16 (address-space control) and 32 (addressing mode) must be zero and one, respectively; otherwise, a special-operation exception is recognized, and instruction execution is suppressed.

The second-operand address is a real address, and accesses to the second-operand location are not subject to key-controlled storage protection. If the operand is not designated on a 256-byte boundary, or if bits 1-19 of the operand address are zeros or are equal to bits 1-19 of the host prefix, then a specification exception is recognized and instruction execution is suppressed.

Host page-translation, segment-translation, and translation-specification exceptions may be recognized during host dynamic address translation for access to the RCP area and guest storage.

Condition Code: The code remains unchanged.

Program Exceptions:

- Addressing (fetch and store, operand 2)
- Operation (if the interpretive-execution facility is not installed)
- Page translation
- PER event (I-fetch only)
- Privileged operation
- Segment translation
- Special operation
- Specification
- Translation specification

Programming Note

The START INTERPRETIVE EXECUTION instruction may be executed with host dynamic address translation (DAT) either on or off. In either case, if the pageable-storage mode is specified, the host primary address space contains guest main storage, and guest-type translation-lookaside buffer (TLB) entries may be formed and used during execution.

This page is intentionally left blank.

The START INTERPRETIVE EXECUTION instruction designates an operand called the state description that is located in host real storage. Fields in the state description provide information about the guest initial state on entry to the interpretive-execution mode, provide guest control and status information used in the interpretive-execution mode, and are used to store information about the guest state on exit from the interpretive-execution mode.

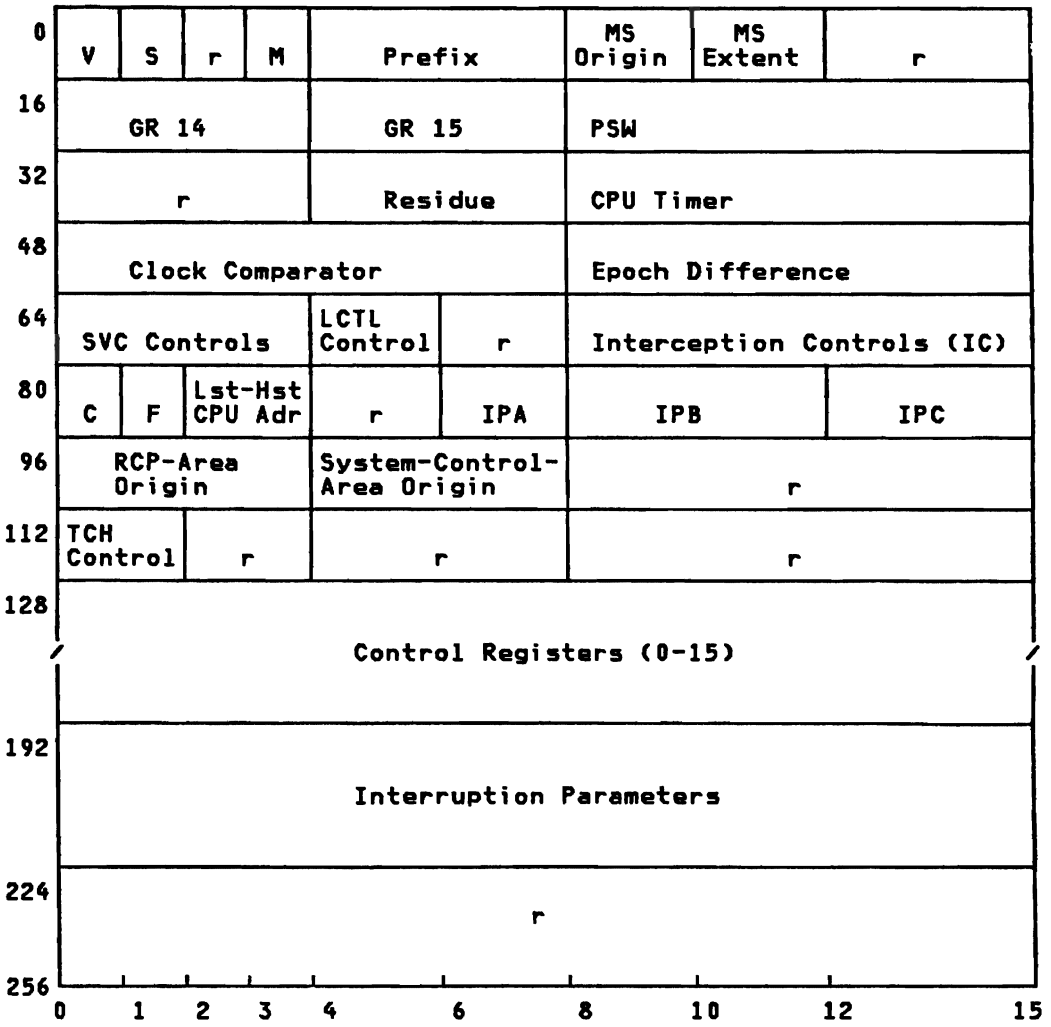
After the CPU enters the interpretive-execution mode, it is undefined whether changes to the state description by the channel subsystem or by another CPU will affect the guest. Fetch references to the state description by the channel subsystem or by another CPU may or may not obtain the current state of the guest. However, bits in the fields labeled "intervention requests" and "TCH control" may be set to ones by means of an interlocked update by one CPU while the fields are concurrently being used to control interpretive execution in another CPU, with the assurance that the change will be observed by the CPU in the interpretive-execution mode. The contents of these two fields are said to be dynamically observed or recognized.

The figure "State Description" shows the format of the state description. The numbers in this figure represent byte offsets. A detailed description of each of the fields, presented sequentially as formatted, follows the figure. The numbers in these figures represent bit positions in the fields.

Programming Note

This publication assigns meanings to various fields, and bit positions within fields, of the state description, RCP area, and SCA. The remaining fields and bit positions are reserved (shown as "r" in the figures).

In order to operate as intended if and when functions are assigned to reserved fields and bit positions, the program should store zeros in reserved fields and bit positions of a state description that has not yet been used as the operand of the START INTERPRETIVE EXECUTION instruction before the CPU enters the interpretive-execution mode and should ignore the contents of those fields and bit positions after the CPU exits from the interpretive-execution mode. Although the CPU may store values in some reserved fields and bit positions, the program should not depend on such values.



Explanation:

C	Interception code	M	Mode controls
F	Interception status	MS	Main storage
IPA	Instruction parameter A	r	Reserved
IPB	Instruction parameter B	S	State controls
IPC	Instruction parameter C	V	Intervention requests

State Description

CONTROL OF OPERATIONS

This section describes the controls for handling pending requests for action and the controls over the modes of interpretive execution.

INTERVENTION REQUESTS (V)

Intervention-request bits are provided to return control to the host to handle a guest I/O interruption, a guest external interruption, or an externally initiated request to stop.

When set to one, an intervention-request bit results in interception under the conditions described below.

r: Reserved

P: 1 A stop interception is requested.

0 A stop interception is not requested.

I: 1 An I/O interception is requested. Interception occurs when a guest PSW I/O-mask bit is one. That is, in the BC mode (System/370) at least one of the guest PSW bits 0-6 is one, or in the EC mode (System/370) or 370-XA mode, PSW bit 6 is one.

0 An I/O interception is not requested.

E: 1 An external interception is requested. Interception occurs when the guest PSW external mask (PSW bit 7) is one.

0 An external interception is not requested.

An intervention-request bit is inspected under these circumstances:

- On entry to the interpretive-execution mode
- When the new PSW is loaded after the execution of a guest LOAD PSW instruction is completed, after the PSW is modified by a guest SET SYSTEM MASK or STORE THEN OR SYSTEM MASK instruction, and after a guest interruption occurs.

  An intervention request is recognized after it is established that no early PSW specification exception exists, before corresponding interruptions enabled by the new or modified PSW occur, and before the next instruction is fetched.
- Periodically when more than one host CPU is in the configuration. Periodic inspection may or may not be provided when a host configuration consists of only one CPU or when the host TOD clock is in the error, stopped, or not-operational state.

Inspection of an intervention-request bit may also occur at other times.

The intervention-request bits may be changed from zero to one by means of an interlocked update by one CPU while they are concurrently being used to control the interpretive-execution mode in another CPU, with the assurance that the change will be observed by the CPU in the interpretive-execution mode. However, if the bits are dynamically changed again from one to zero, it is unpredictable which value is observed. Changes to the contents of this field by the channel subsystem during interpretive execution may or may not be recognized and may be lost.

An intervention-request bit is not set to zero on the occurrence of the corresponding interception.

STATE CONTROLS (S)

The state-control byte provides a location to record a pending guest interval-timer external-interruption condition. Although the condition exists when the interval timer has been decremented from a positive number or zero to a negative number, it may not be recognized immediately. A bit in the byte is set to indicate whether there is a pending guest interval-timer interruption on exit from the interpretive-execution mode, and is examined on entry to the interpretive-execution mode. The bit is also examined when the guest becomes enabled for interval-timer interruptions.

T: 1 If a System/370-mode guest is enabled for interval-timer interruptions (bit 7 of the PSW and bit 24 of control register 0 are both ones) and the interval-timer-activation-control (D) bit of the mode-control field is zero, then a guest external interruption for the interval timer is requested.

0 A guest external interruption for the interval timer is not requested.

r: Reserved; may be set to zero.

The T bit is ignored unless the System/370 mode is specified and the D bit is zero.

When the corresponding guest interruption occurs, it results in a mandatory interception that places the interruption parameters in the state description and sets the T bit to zero.

MODE CONTROLS (M)

The mode-control byte specifies whether the guest is executed in the System/370 or 370-XA mode and whether guest main storage is represented as a portion of the host primary address space or is considered to be assigned to the corresponding portion of host absolute storage. In addition, in the System/370 mode, control is provided over whether the interval timer is active or inactive.

r: Reserved

MM: 00 Invalid
01 The guest is in the System/370 mode.
10 The guest is in the 370-XA mode.
11 Invalid

G: 0 Pageable-storage mode
1 Preferred-storage mode

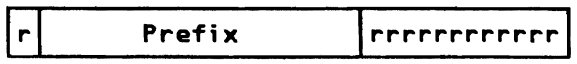
D: 0 Interval timer active
1 Interval timer inactive

The D bit is ignored unless MM is 01.

DEFINITION OF GUEST STORAGE

This section describes how the placement of guest storage within host storage is specified and how the amount of guest storage to be made available is specified. Provision of prefixing for the guest is also described.

PREFIX



This field contains the contents of the guest prefix register. Prefixing is considered to be installed for all guests. A validity interception occurs if the prefix value does not designate an available location within guest main storage. Bits 1 through 7 may or may not be ignored when the System/370 mode is specified; if they are not ignored, a nonzero value results in a validity interception. Bit position 0 and bit positions 20-31 are reserved.

If the System/370 mode and the pageable-storage mode are specified, then prefixing is applied to guest dynamic-address-translation (DAT) table references for both implicit translation and for LOAD REAL ADDRESS. If the preferred-storage mode or the 370-XA mode is specified, then prefixing may or may not be applied to DAT-table references for implicit translation and for LOAD REAL ADDRESS.

MAIN-STORAGE ORIGIN

When the pageable-storage mode is specified, bits 1-15 of this field, with 16 zero bits appended on the right, designate the virtual-storage location within the host primary address space that represents guest absolute-storage address 0. The location corresponds to an address which is a multiple of 64K bytes. If the preferred-storage mode is specified and any of bits 1-15 of this field is one, then a validity interception occurs. Bit position 0 of this field is reserved.

The same origin value within the same host address space may be designated in another state description for the purpose of interpreting a guest shared-main-storage multiprocessing configuration.

MAIN-STORAGE EXTENT

Bits 1-15 of this field specify the maximum size of guest main storage in both the pageable-storage and preferred-storage modes. Bit position 0 is reserved. The size of guest main storage is the value of bits 1-15 of this field, plus one, multiplied by 64K bytes. That is, a main-storage-extent value of n specifies (n + 1) times 64K bytes of guest main storage.

If the preferred-storage mode is specified and if the state description or the host prefix area would be included within the storage available to the guest, then a validity interception occurs on entry to the interpretive-execution mode.

When the pageable-storage mode is specified, a validity interception may be recognized on entry to the interpretive-execution mode if guest main storage appears to wrap in the host primary address space, or if an invalid translation format is specified in host control register 0. Alternatively, these conditions may be recognized instead as host program exceptions when a translation is performed.

GENERAL REGISTERS AND THE PSW

The contents of two of the guest general registers are obtained from the state description. Described below are the fields that contain the contents for these registers and the requirement to preserve the host contents for these registers. Also described is the field containing the guest PSW that is used on entry to the interpretive-execution mode and that is stored on exit from the interpretive-execution mode.

GR 14 AND GR 15

Contents for guest general registers 14 and 15 are obtained from these fields on entry to the interpretive-execution mode and are saved in these fields on exit. The contents of host general registers 14 and 15 are saved on entry to the interpretive-execution mode and are restored on exit. General registers 0-13 are shared by the guest and the host; that is, the contents of these registers are not changed on entry to or on exit from the interpretive-execution mode.

PSW

This field contains the contents of the guest current PSW. The PSW contains the control and status information necessary to initiate guest operation on entry to the interpretive-execution mode. On exit from the interpretive-execution mode, the contents of the guest current PSW are stored in this field. When the exit from the interpretive-execution mode is due to interception for a guest interruption, the guest PSW stored in the state description is the PSW which would have been stored as the guest-interruption old PSW.

CONTROL OF TIMING

This section describes the two fields that specify the contents of the guest CPU-timer and clock-comparator registers. Also described are the residue field, used in maintaining the guest interval timer, and the epoch-difference field, used in providing a guest TOD-clock value.

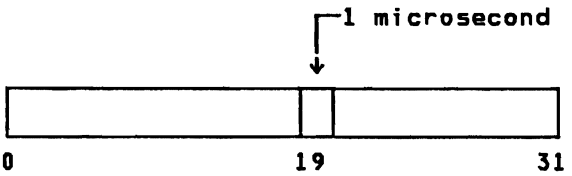
Both host and guest timing facilities are provided. All aspects of the guest timing facilities are provided in the interpretive-execution mode, except for (1) the SET CLOCK instruction (for which interception is mandatory), (2) guest TOD-clock synchronization (which is handled by the host), and (3) guest external interruption conditions for the interval timer, clock comparator, and CPU timer (for which interception is mandatory).

Between the instant that some condition causing exit from the interpretive-execution mode is initially recognized and the instant that exit from the interpretive-execution mode finally occurs, additional time may be reflected in the values placed in the state description for the guest CPU timer and interval timer. The conditions specified by these values and the enabled state of the guest may indicate that an interruption is due. Interruptions and interceptions which become due during exit from the interpretive-execution mode are not necessarily recognized instead of the condition which initiated the exit, even if the new condition is ordinarily handled at a higher priority.

If a host CPU is in the interpretive-execution mode when the TOD clock accessed by that CPU enters the running state or is changed by another CPU sharing the same TOD clock, the values of the guest interval timer, residue field, and guest CPU timer may be unpredictably changed. Also, guest interruptions due to the interval timer, CPU timer, and clock comparator may be lost, delayed, or erroneously generated.

RESIDUE COUNTER

The residue-counter field has the following format:



The format of the residue-counter field corresponds to bits 32-63 of the TOD clock. This field is used to maintain an accurate accumulation of the CPU time spent in the interpretive-execution mode that has not yet been accounted for by decrementing the interval timer in guest storage. This field is ignored when the interval timer is not active.

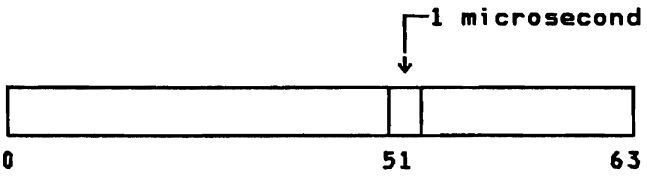
On entry to the interpretive-execution mode, the interval timer in guest storage is adjusted by decrementing bit position 23 once for each integral multiple of 3,333 microseconds contained in the residue counter. The corresponding amount is deducted from the residue counter, with the remainder treated as an initial elapsed-time value. The adjustment may or may not occur when a

preexisting interception-causing condition is recognized on entry to the interpretive-execution mode.

On exit from the interpretive-execution mode, the residue-counter field contains the sum of the value of the residue counter at entry, plus the CPU time considered to have elapsed in the interpretive-execution mode since entry, minus 3,333 microseconds for each time the interval timer in guest storage was decremented in bit position 23.

CPU TIMER

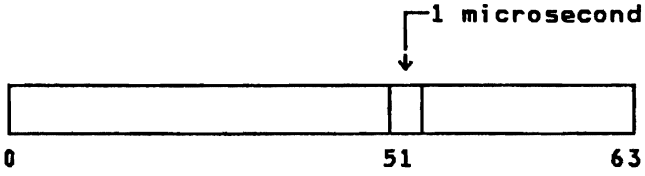
The CPU-timer field has the following format:



This field is provided for the guest CPU-timer value. The guest CPU timer is decremented only when the CPU is in the interpretive-execution mode and is decremented at the same rate as the host CPU timer.

CLOCK COMPARATOR

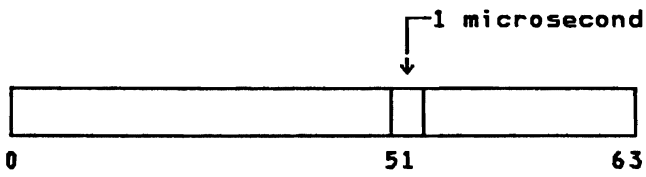
The clock-comparator field has the following format:



This field contains the guest clock-comparator value.

EPOCH DIFFERENCE

The epoch-difference field has the following format:



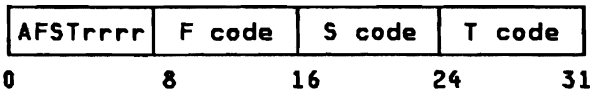
This field contains the difference between the guest and the host TOD-clock epochs, expressed as a 64-bit unsigned integer. Since bit position 51 represents one microsecond, this field has the same format as the TOD clock. The value of the guest TOD clock is the sum of the epoch difference and the value of the host TOD clock (with any carry out of bit 0 ignored). Bits representing higher resolution than provided by the host TOD clock may or may not be lost.

CONTROL OF INTERCEPTIONS

The term "conditional interception" refers to functions that are executed for the guest unless a specified condition is encountered that causes control to be returned to the host by the process called interception. Described below are some of the controls that can cause interception when the related function is handled for the guest.

SVC CONTROLS

The first byte of this field specifies whether all guest SUPERVISOR CALL (SVC) instructions, or only those for which the effective I field matches a code, cause instruction interception to be recognized. The next three bytes contain the code values used in the I-field-matching process.



A: 1 Instruction interception is recognized for every guest SUPERVISOR CALL instruction.
0 Instruction interception is recognized when the effective I field of a guest SUPERVISOR CALL instruction matches a control code, provided that the code is enabled for matching by one or more of the next three bits.

F: 1 Instruction interception is recognized for a guest SUPERVISOR CALL instruction for which the effective I field is equal to the F code.
0 A match with the F code is not recognized.

S: Same as for the F bit, but applies to S code

T: Same as for the F bit, but applies to T code

r: Reserved

F code: A one-byte value to be compared with the contents of the effec-

tive I field of SUPERVISOR CALL instructions if matching is specified by a one in the F bit

S code: Same as for F code but with respect to the S bit

T code: Same as for F code but with respect to the T bit

LCTL CONTROLS

The bits of this field are numbered, starting from the left, to correspond to control-register numbers. When a bit is one and a guest LOAD CONTROL (LCTL) instruction designates the corresponding control register in the range of registers to be loaded, instruction execution is suppressed, and instruction interception is recognized.

INTERCEPTION CONTROLS (IC)

A bit value of one results in interception when the associated function is encountered in the guest.

Bit Position	Associated Function
0	Program interruption (operation exception)
1	Program interruption (privileged-operation exception)
2	Program interruption (other than mandatory program-interruption, operation-exception, and privileged-operation-exception conditions)
4	TEST AND SET, for condition code 1
5	COMPARE AND SWAP, for condition code 1
6	COMPARE DOUBLE AND SWAP, for condition code 1
7	INVALIDATE PAGE TABLE ENTRY
9	LOAD PSW
10	PURGE TLB
11	SET SYSTEM MASK
13	STORE CONTROL
14	STORE THEN AND SYSTEM MASK
15	STORE THEN OR SYSTEM MASK
16	STORE CLOCK
17	INSERT STORAGE KEY and INSERT STORAGE KEY EXTENDED
18	SET STORAGE KEY and SET STORAGE KEY EXTENDED
19	RESET REFERENCE BIT and RESET REFERENCE BIT EXTENDED
20	PROGRAM CALL
21	PROGRAM TRANSFER
22	TEST PROTECTION
23	LOAD ADDRESS SPACE PARAMETERS
25	SET CPU TIMER and STORE CPU TIMER
26	SET CLOCK COMPARATOR and STORE CLOCK COMPARATOR

Unassigned bits are reserved.

INTERCEPTION PARAMETERS

Exit from the interpretive-execution mode takes one of two forms. One form is exit by a host stop function or interruption, after which, for host interruptions, the condition can be handled by the host program. The second form is exit by interception, which causes execution of host instructions to resume with the instruction following the START INTERPRETIVE EXECUTION instruction (or an EXECUTE instruction, as appropriate). Normally, interception results from a function or condition encountered in the guest that must be handled by a host program. To allow the condition to be efficiently analyzed by the host program, descriptive information about the condition is stored in the state description at interception. The information provided is described in this section.

INTERCEPTION CODE (C)

The code stored in this field during interception indicates the reason for the interception. The contents of this field are changed only at interception.

Code	Interception Condition Applicable to the Guest
04	Instruction
08	Program interruption
12	Instruction and program interruption
16	External request
20	External interruption
24	I/O request
28	Wait state
32	Validity
40	Stop request
44	Operation exception

Unassigned codes are reserved.

The definitions of the conditions under which each code is stored are given below.

Code 04 (Instruction)

Instruction interception is indicated by code 04 when interception is mandatory or when interception is conditional and the interception control for the instruction is one. The above conditions also apply to the target instruction of EXECUTE.

Some instructions are valid either in the 370-XA mode or in the System/370 mode, but not in both modes. If instruction interception is either mandatory or conditional in the mode in which the instruction is valid, then it is undefined whether instruction interception is recognized unconditionally or an operation exception is recognized in the invalid mode.

When code 04 is stored, guest instruction execution is suppressed, except for the COMPARE AND SWAP, COMPARE DOUBLE AND SWAP, and TEST AND SET instructions (which are completed).

Code 08 (Program Interruption)

Interception is mandatory, and interception code 08 is stored, for guest program interruptions caused by these exceptions:

- Protection
- Addressing
- Specification
- Special operation

Otherwise, program interruptions cause interception with code 08 only if they are of a type for which the corresponding interception control, bit 1 or bit 2, is set to one.

A guest protection exception may be due to a guest key-mismatch condition, a guest page-protection or a guest segment-protection condition, a guest low-address-protection condition, or a host page-protection condition.

A guest addressing exception may be due to an invalid guest or host address.

The parameters of the interruption are placed in the state description.

Code 12 (Instruction and Program Interruption)

Instruction interception is indicated by code 12 when conditional interception is recognized because execution is completed with condition code 1 set for the COMPARE AND SWAP, COMPARE DOUBLE AND SWAP, and TEST AND SET instructions and a guest PER event has also been recognized. The PER event results in the storing of the corresponding program-interruption parameters in the state description. Code 12 is stored regardless of whether the interception control for the program interruption is also set to cause interception.

If a PER I-fetch event is applicable, then it is undefined whether or not bit 6 (IF) of the interception-status field is set.

Code 16 (External Request)

Interception is due to bit 7 of the intervention-request field being one when PSW bit 7 is one.

Code 20 (External Interruption)

Interception is mandatory for guest external interruptions. Guest external interruptions are generated for the following facilities:

- Guest clock comparator
- Guest CPU timer
- Guest interval timer

The parameters of the interruption are placed in the state description.

Code 24 (I/O Request)

Interception is due to bit 6 of the intervention-request field being one either when any of bits 0-6 of a BC-mode (System/370) PSW is one or when bit 6 of an EC-mode (System/370) PSW or 370-XA PSW is one.

Code 28 (Wait State)

Interception is due to bit 14 of the guest PSW being one when no other interruption condition or interception condition (including intervention requests) can be recognized.

Code 32 (Validity)

Validity interception may be recognized if a mode is specified that is invalid or not installed. Validity interception caused by values in the state description fields is discussed in the appropriate sections of this chapter.

Before any reference is made to guest main storage on entry to the interpretive-execution mode, a validity check is made of the mode controls and, in the preferred-storage mode, of the state description and the host prefix area to assure that neither lies within guest main storage. Interception for invalid mode controls takes precedence over all other reasons for interception. Validity interception caused by access exceptions is discussed in Chapter 4, "Storage."

A validity interception recognized during the interpretive execution of a guest instruction may result in suppression, nullification, or termination of the guest unit of operation.

Code 40 (Stop Request)

Interception is due to bit 5 of the intervention-request field being one.

Code 44 (Operation Exception)

Interception is due to an attempt to execute an instruction and one of the following situations:

- The operation code is valid in neither the 370-XA nor System/370 modes
- The operation code is not valid in the mode specified but is valid and subject to either conditional or mandatory interception in the other mode
- The operation code is valid in the mode specified but is not installed

It is undefined whether an operation code in the last two categories may instead be handled as an instruction interception. If it is handled as an instruction interception, then other applicable exceptions may be recognized in preference to instruction interception.

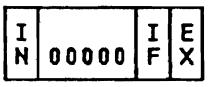
If an operation exception is recognized and interception-control bit 0 is zero, then a guest program interruption (operation exception) occurs. If an operation exception is recognized, and interception-control bit 0 is one, then an interception is recognized and the following actions occur:

1. Interception code 44, operation exception, is stored.
2. The first two bytes of the instruction are stored in instruction parameter A (IPA). If interception format 2 is not installed, then no instruction information is stored in instruction parameter B. If interception format 2 is installed, then the remaining bytes, if any, of the instruction are stored in instruction parameter B. The remaining bytes of instruction parameter B (IPB) and instruction parameter C (IPC) are set to zero. Instruction execution is suppressed.

   The instruction-length code (ILC) in a System/370-mode-guest BC-mode PSW is not necessarily valid. Interruption parameters are not generated, except for the interruption-code portion of a System/370 BC-mode PSW, which may specify operation exception.
3. The EX bit (bit 7) of interception status is set to specify whether the invalid operation code is the target of an EXECUTE instruction.
4. The IF bit (bit 6) of interception status is set to specify whether an instruction-fetch PER event is applicable.

INTERCEPTION STATUS (F)

On interception for a guest instruction, this field specifies whether a guest I-fetch PER event is applicable and whether the guest instruction is the target of a guest EXECUTE instruction. This field is also used to specify that interception format 2 is used on interception for a guest instruction. Zero is stored in this field for interception codes other than 04, 12, and 44.



IN: 1 Interception format 2 is installed; The combined IPA and IPB fields contain the bytes of the instruction for which interception is indicated.

0 Interception format 2 is not installed; The IPA field contains the first two bytes of the instruction, and the IPB and IPC fields contain operand-addressing information when applicable.

IF: 1 An instruction-fetch PER event is applicable to the instruction for which interception has been recognized. When the instruction is the target of a guest EXECUTE instruction, the instruction-

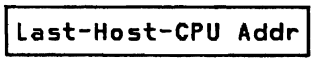
fetch event is indicated when it is applicable to either the EXECUTE instruction or the target instruction or both. This condition is indicated only with interception codes 04 and 44; the bit may or may not be set for interception code 12.

0 An instruction-fetch PER event has not been recognized. The bit is always set to the appropriate value for interception codes 04 and 44; the bit may or may not be set for interception code 12.

EX: 1 Interception is indicated for an instruction that is the target of a guest EXECUTE instruction. The condition is indicated with interception codes 04, 12, and 44.

0 Interception is indicated for an instruction that is not the target of an EXECUTE instruction. The condition is indicated with interception codes 04, 12, and 44.

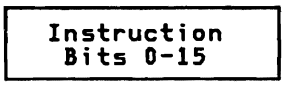
LAST-HOST-CPU ADDRESS



If guest-type TLB entries are retained after exit from the interpretive-execution mode, this field contains the 16-bit host CPU address (that would be obtained by a host STORE CPU ADDRESS instruction) of the CPU that most recently executed a START INTERPRETIVE EXECUTION instruction whose operand was this state description. On entry to the interpretive-execution mode, the contents of this field are compared with the host CPU address. If the field matches the address, then existing guest-type TLB entries can be used. If the field does not match the address, then existing guest-type TLB entries for this guest are cleared. If guest-type TLB entries are cleared on exit from the interpretive-execution mode, the contents of this field are either unchanged or set to the CPU address as described above.

The host program may store the hex value FFFF in this field to force all guest-type TLB entries for this guest to be cleared on the CPU on entry to the interpretive-execution mode.

INSTRUCTION PARAMETER A (IPA)



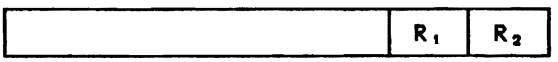
If the interception code is 04, 12, or 44, then this field contains the first two bytes (bit positions 0-15) of the instruction causing the interception, as modified by EXECUTE if applicable. Zero is stored in this field for interception codes other than 04, 12, and 44.

INSTRUCTION PARAMETER B (IPB)

For some instructions, either bytes 2 and up, if any, or operand-addressing information, if applicable, is stored in this field at interception.

If interception format 2 is not installed, then operand-addressing information or zero is stored. If operand-addressing information is stored, then two classes are used. In the first class, an effective address is formed according to the current guest addressing mode and then is stored in this field as a 31-bit value. Zero is stored in bit position 0. This presentation is used at instruction interception for the operand address of the DIAGNOSE instruction (DIAGNOSE, for this purpose, is treated as an S-format instruction), RS-format, SI-format, and S-format instructions, and for the first-operand address for SSE-format instructions.

In the second class, the fourth byte of the RRE-format instruction is stored in the fourth byte of this field. The contents of bit positions 0-23 may or may not change.



Zero is stored for all interception codes other than 04. Zero is also stored for interception code 04 if the parameters of the instruction for which interception is indicated are contained within the IPA field, or if the instruction for which interception is indicated is COMPARE AND SWAP, COMPARE DOUBLE AND SWAP, or TEST AND SET.

If interception format 2 is installed, then, for interception codes 04, 12, and

44, bytes 2 and up, if any, of the instruction are stored in this field; unused bytes of this field are set to zero, and bit 0 of the interception-status field is set to one.

INSTRUCTION PARAMETER C (IPC)

If interception format 2 is not installed, addressing information is stored in this field upon interception for SSE-format instructions. The effective second-operand address is formed according to the current guest addressing mode and then is stored in this field as a 31-bit value. Zero is stored in bit position 0. Zero is stored in this field for all interception codes other than 04, or if the instruction for which interception (code 04) is indicated is not an SSE-format instruction.

If interception format 2 is installed, then zero is stored in this field at interception.

ORIGINS OF RELATED TABLES

Depending on the controls specified, additional tables are used to contain control information. The origins of these tables are obtained from the state description fields described below.

RCP-AREA ORIGIN

If the pageable-storage mode is specified, then the contents of bits 1-19 of this field, with 12 zeros appended on the right, designate the host primary virtual address of an area used to retain reference-bit and change-bit information. Bit position 0 and bit positions 20-31 are reserved. If the preferred-storage mode is specified, this field is ignored. The RCP area contains one byte for each 4K bytes of guest main storage, with the index of a byte within the area equal to the index of the corresponding 4K-byte block within guest main storage. The format of each RCP byte is described in the "Reference-and-Change Handling" section of Chapter 4, "Storage."

In the pageable-storage mode, a validity interception is recognized if the RCP-area-origin field specifies zero, or if wrapping of the RCP area is implied by the amount of guest main storage specified. A validity interception may or may not be recognized if the implied size of the RCP area exceeds the length of the host primary address space.

SYSTEM-CONTROL-AREA ORIGIN

The contents of bits 1-27 of this field, with four zeros appended on the right, designate the host real address of a sixteen-byte system-control area (SCA). Bit position 0 and bit positions 28-31 are reserved.

A validity interception is recognized, either during execution of the INVALIDATE PAGE TABLE ENTRY instruction or on entry to the interpretive-execution mode if (1) bits 1-19 of the SCA origin match bits 1-19 of the host prefix value, (2) bits 1-19 of the SCA origin are zero when bits 20-27 are not zero, (3) the SCA origin designates an invalid address, or (4) the SCA origin designates storage within the guest extent and the preferred-storage mode is specified. The value of the SCA origin also determines whether or not other CPUs are signaled during the execution of a guest INVALIDATE PAGE TABLE ENTRY instruction. See Chapter 4, "Storage," for details.

OTHER CONTROLS

This section describes a control field related to guest I/O operations.

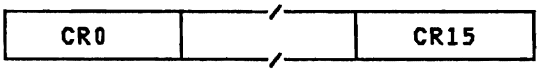
TCH CONTROL

The bits of this field are numbered from left to right to correspond to System/370 channel addresses 0-15. Interpretive execution of a TEST CHANNEL instruction is completed by setting condition code 0 unless (1) a privileged-operation exception is recognized, or (2) instruction interception is recognized because the bit in the TCH control field corresponding to the designated channel is one or because a

channel address larger than 15 is specified. Instruction execution is suppressed for interception.

This field may be changed dynamically by means of an interlocked update by one CPU while the fields are concurrently being used to control interpretive execution in another CPU, with the assurance that the change will be observed by the CPU in the interpretive-execution mode.

CONTROL REGISTERS



Contents for guest control registers 0-15 are obtained from these fields on entry to the interpretive-execution mode and are saved in these fields on exit. The contents of host control registers are preserved on entry to the interpretive-execution mode and are restored on exit. The registers appear in ascending order of register numbers, starting with guest control register 0.

INTERRUPTION PARAMETERS

If interception is caused by a guest external or guest program interruption for which the native definition specifies storing at real storage locations 128 through 159, then storing occurs instead in the state description at locations 192 through 223. Locations 192 through 223 of the state description are formatted in a manner identical to locations 128 through 159 of real storage.

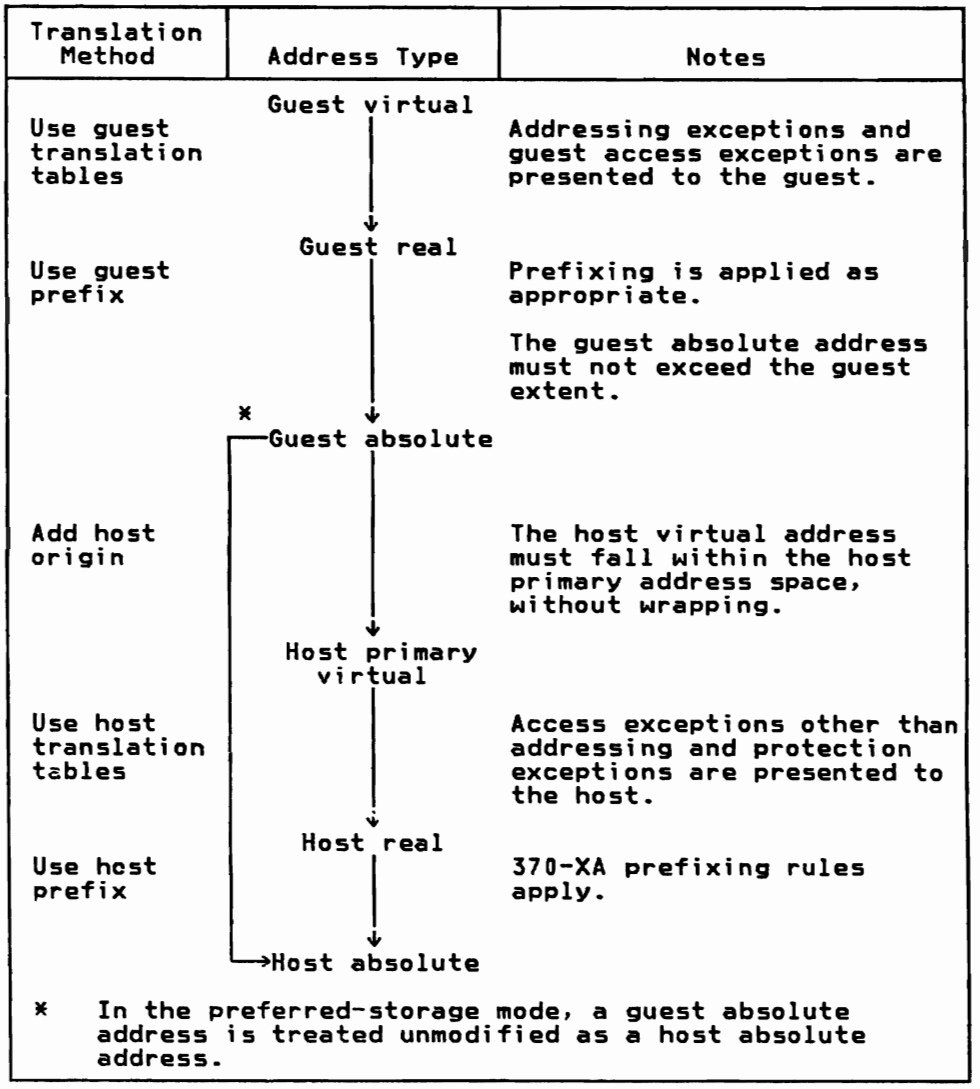
ADDRESSING

Two mechanisms are provided for representing guest main storage. One mechanism represents guest main storage as a contiguous portion of the host primary address space. The origin of guest main storage within the host primary address space is provided in the state description. This is called the pageable-storage mode.

The second mechanism represents guest absolute addresses as corresponding to the equivalent host absolute addresses; host dynamic address translation (DAT) and host prefixing are not applied. This is called the preferred-storage mode.

In both cases, the amount of guest main storage that is provided is specified in the state description.

The guest may also use DAT. A guest virtual address is first translated to a guest real address, using guest translation tables. Guest prefixing is then applied to produce a guest absolute-storage address. When the preferred-storage mode is specified, a guest absolute-storage address is treated as the corresponding host absolute-storage address. When the pageable-storage mode is specified, a guest absolute address is converted to a host primary virtual address by adding to it the host virtual address at which guest storage begins in the host primary address space (from the main-storage-origin field in the state description). The host address is then translated by using host translation tables. Finally, host prefixing is applied. Access exceptions (other than addressing and protection exceptions) for host addresses result in host interruptions, while access exceptions for guest addresses result in guest interruptions. All addressing and protection exceptions are considered guest exceptions. The address-translation mechanisms are illustrated in the figure "Translating Addresses for Interpretive Execution."



Translating Addresses for Interpretive Execution

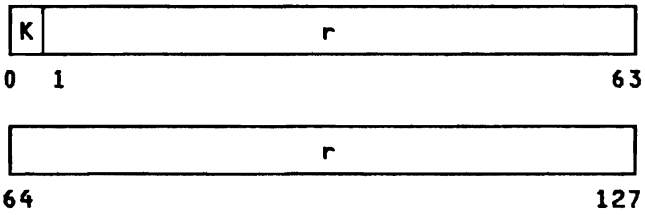
If the System/370 mode and the preferred-storage mode are specified, then it is undefined whether or not guest prefixing is applied to guest DAT-table-entry addresses for references due to implicit address translation or due to the execution of a LOAD REAL ADDRESS instruction. However, guest prefixing always applies to the operand address designated by the INVALIDATE PAGE TABLE ENTRY instruction.

The support of DAT in the System/370 mode may or may not include support for the 1M-byte segment size and for the 2K-byte page size. All models support the 64K-byte segment size and the 4K-byte page size. When the page size is 4K bytes, the effective page-frame real address always includes page-table-entry bits 13-14, which are appended on the left of the page-frame real address, resulting in a 14-bit value which allows 26-bit real-storage addressing for a System/370-mode guest.

Guest-type TLB entries may be formed to provide a rapid means of translating guest addresses to host addresses. These entries may or may not be retained when the CPU exits from the interpretive-execution mode. The entries thus formed are further differentiated from each other by guest environmental information that is taken into account to maintain the usability of the entries. The relevant environmental information includes guest translation parameters and translation tables, host translation parameters and translation tables, the host real address of the state description, the host real address of the system-control area (SCA), and the storage definition and the storage and architecture modes of the guest. Guest-type TLB entries that were previously formed under a

matching state-description address are cleared on entry to the interpretive-execution mode if the last-host-CPU-address field of the state description does not match the current CPU address.

The system-control area (SCA) provides a control for coordinating the activities associated with each CPU of the guest multiprocessing system. The same SCA origin should be designated in the state description for each guest CPU of that guest multiprocessing system. If the SCA origin has a value of zero, then the guest INVALIDATE PAGE TABLE ENTRY instruction is executed without signaling other CPUs. If the SCA origin has a nonzero value, then all CPUs in the configuration are signaled to clear those guest-type TLB entries formed under a matching SCA-origin value that are affected by the INVALIDATE PAGE TABLE ENTRY instruction. The access to the SCA is not subject to key-controlled storage protection, nor to low-address protection.



Bit 0 of the SCA is called the IPTE-interlock-control (K) bit. A guest INVALIDATE PAGE TABLE ENTRY instruction changes the K bit from zero to one by means of an interlocked-update function at the beginning of execution. The K bit is reset to zero by means of an interlocked-update function at the conclusion of execution. If the initial value of the K bit is one, then instruction interception is recognized. If the interlocked-update function observes a change in its operand during the update operation, then it is undefined whether or not instruction interception is recognized.

Bit positions 1-127 of the SCA are reserved.

ACCESS CONTROLS AND EXCEPTIONS

Guest accesses to guest storage, in all modes of interpretive execution, are subject to key-controlled storage protection as defined natively, using the real-machine storage keys. Guest accesses are also subject to guest low-address protection, to guest segment protection when the System/370 mode is specified, and to guest page protection and guest fetch-protection override when the 370-XA mode is specified. Host low-address protection is not applied to guest references to guest storage.

In addition, host page protection applies to guest references in the pageable-storage mode. This checking applies to guests in both the System/370 and 370-XA modes and includes the checking of guest references to guest main storage and the checking of implied references to the reference-and-change-preservation (RCP) area.

In general, disallowed storing causes a protection exception to be recognized, which results in a guest program interruption. However, an access exception encountered when referencing the guest prefix area to perform a guest interruption, or to update the interval timer, may result in a validity interception. Alternatively, a host page-translation, segment-translation, or translation-specification exception may be recognized as a host program exception, in which case the guest operation is nullified. When referencing the guest prefix area on entry to the interpretive-execution mode, an access exception may or may not result in a validity interception. In either the pageable-storage or preferred-storage mode, an addressing exception when referencing the guest prefix area results in a validity interception.

The host page-protection mechanism is taken into consideration when the condition code is set during the interpretive execution of a guest TEST PROTECTION instruction.

The INSERT STORAGE KEY, INSERT STORAGE KEY EXTENDED, SET STORAGE KEY, and SET STORAGE KEY EXTENDED instructions are interpreted by accessing the access-control, reference-and-change, and fetch-protection bits in the real-machine storage key. In the pageable-storage mode, additional information concerning the reference and change bits is kept in the reference-and-change-preservation (RCP) area.

In the pageable-storage mode, obtaining the host absolute address associated with the guest real address specified by the storage-key-handling instructions (and thereby specifying the real-machine storage key to be used) requires translating a host primary virtual address. The access to the RCP area also involves translating a host primary virtual address. It is undefined whether access exceptions for the RCP area are recognized before or after access exceptions for the real-machine storage key.

In the interpretive-execution mode, no checking occurs with respect to the host PSW key. No key-controlled storage protection is applicable to references to the state description or to references to the RCP area or SCA.

STORAGE-KEY HANDLING

Interpretive execution is provided for all of the storage-key-handling instructions. The results of the execution of the INSERT STORAGE KEY EXTENDED (ISKE), RESET REFERENCE BIT EXTENDED (RRBE), and SET STORAGE KEY EXTENDED (SSKE) instructions are appropriate to the architectural mode being interpreted (System/370 or 370-XA) in both the preferred-storage and pageable-storage modes. INSERT STORAGE KEY (ISK), RESET REFERENCE BIT (RRB), and SET STORAGE KEY (SSK) instructions are provided only for System/370-mode guests.

If the pageable-storage mode is specified, a page frame of host real storage may be the object of a translation for guest purposes and for host purposes simultaneously. The setting or resetting of the real-machine reference-and-change bits by guest RESET REFERENCE BIT, RESET REFERENCE BIT EXTENDED, SET STORAGE KEY, or SET STORAGE KEY EXTENDED instructions in conjunction with managing guest storage would leave the bits in an incorrect state for managing host storage. Conversely, changes to these bits by host actions result in the real-machine indicators incorrectly reflecting the status of guest storage. For this reason, three sets of reference-and-change bits are maintained. One set is the real-machine set. The bits in this set are the only indicators affected by storage accesses, including accesses by the channel subsystem. The other two sets are maintained in the RCP area. One set, the host RCP set, is used to retain status for the host when the real-machine set is changed by guest storage-key-handling operations. The other set, the guest RCP set, is provided to retain status for the guest when the real-machine set is changed by host storage-key-handling instructions.

REFERENCE-AND-CHANGE HANDLING

The RCP area, provided only if the pageable-storage mode is specified, consists of one byte for each 4K bytes of guest main storage. Each RCP-byte number is the number of the associated 4K-byte block in guest main storage. The origin of the RCP area is designated by a field in the state description and resides in the host primary address space. The same RCP area can be designated in more than one state description when a guest shared-main-storage multiprocessing configuration is interpreted.

Five bits of each byte in the RCP area are used. The first bit is an interlock control. It is changed by each pageable-storage-mode guest storage-key-handling instruction by means of an interlocked update, thus ensuring exclusive use of the byte. The next two bits are used to save the host reference and host change indications, respectively. Similarly, two bits are provided for saving the guest reference and guest change indications. See the section "Instruction Execution" for more detail.

The format of each byte of the RCP area is illustrated below:

I RCP interlock control
HR RCP host reference indicator
HC RCP host change indicator
GR RCP guest reference indicator
GC RCP guest change indicator
r Reserved

INSTRUCTION EXECUTION

In the pageable-storage mode, the definition of the storage-key-handling instructions differs from the native definition as follows:

- If a host page-translation condition exists because the invalid bit is one in the host page-table entry corresponding to the second-operand address, then instruction interception is recognized and execution is suppressed for the INSERT STORAGE KEY, INSERT STORAGE KEY EXTENDED, SET STORAGE KEY, and SET STORAGE KEY EXTENDED instructions. The RESET REFERENCE BIT and RESET REFERENCE BIT EXTENDED instructions are completed using only RCP-byte information.

- Guest reference-and-change status is obtained as the logical OR of the real-machine set with the guest set in the RCP area. Guest reference-and-change alteration is brought about by (1) storing the logical OR of the real-machine set with the host set in the RCP area in the host set, (2) storing the operand bits in the guest set, and (3) storing zeros in the real-machine reference-and-change bits.

The following facilities, in addition to the basic computing functions defined in the IBM System/370 Principles of Operation, GA22-7000, are provided when the System/370 mode is specified for the guest:

- Commercial instruction set
- Floating point
- Extended-precision floating point
- Translation (with 1M-byte segments optional and 2K-byte pages optional)
- Extended (26-bit) real addressing
- CPU timer and clock comparator
- Conditional swapping
- PSW-key handling
- Multiprocessing
- Storage-key 4K-byte block
- Extended facility: INVALIDATE PAGE TABLE ENTRY, TEST PROTECTION, common segment bit, and low-address protection
- Test block (interception of the TEST BLOCK instruction is mandatory)
- Branch and save
- Dual address space
- Storage-key-instruction extensions
- Segment protection

The 370-XA architecture is provided when the 370-XA mode is specified for the guest.

Guest instructions not defined for the specified architectural mode but defined and interceptible for the other mode cause either an instruction interception or an operation exception to be recognized.

All control registers are considered to be installed in the guest.

Additional facilities may or may not also be offered in the interpretive-execution mode. Interpretive execution may be fully provided, or interpretive execution may be partially provided with the remainder provided by the host.

GUEST INSTRUCTION PROCESSING

Most guest instructions are fully interpreted, with the transfer of control within the guest program performed according to the native definition. However, some guest instructions require treatment by the host, and control is returned to the host program in these cases by a process called interception. Most guest instructions for which interception is recognized are suppressed prior to interception. However, for some instructions, interception occurs after the instruction has been completed.

The two general reasons for interception are (1) conditional interception has been selected, or (2) interception is mandatory because the instruction is not interpreted under the existing conditions. Normally, parameters of the guest instruction that are useful to the host in handling the guest operation are stored in the state description at interception.

Specification exceptions associated with the guest PSW or a guest EXECUTE instruction, and access exceptions for the instruction fetch of a guest EXECUTE instruction or first halfword of an instruction, are recognized ahead of instruction interception for the instruction due to be fetched. The privileged-operation, specification, and special-operation exceptions may be recognized ahead of instruction interception, except for specification-exception conditions introduced into the guest PSW by the instruction. For the conditionally interceptible instructions, all other applicable exceptions can also be recognized ahead of interception, except those that are normally recognized on completion of execution.

Interception is mandatory for the following instructions in the modes in which they are installed.

CLEAR CHANNEL
CLEAR I/O
CONNECT CHANNEL SET
CLEAR SUBCHANNEL
DIAGNOSE
DISCONNECT CHANNEL SET
HALT DEVICE
HALT I/O
HALT SUBCHANNEL
MODIFY SUBCHANNEL
RESET CHANNEL PATH

RESUME I/O
RESUME SUBCHANNEL
SET ADDRESS LIMIT
SET CHANNEL MONITOR
SET CLOCK
SET PREFIX
SIGNAL PROCESSOR
START I/O
START I/O FAST RELEASE
START INTERPRETIVE EXECUTION
START SUBCHANNEL
STORE CPU ADDRESS
STORE CHANNEL PATH STATUS
STORE CHANNEL REPORT WORD
STORE CHANNEL ID
STORE CPU ID
STORE PREFIX
STORE SUBCHANNEL
TEST BLOCK
TEST I/O
TEST PENDING INTERRUPTION
TEST SUBCHANNEL

INTERACTIONS OF FACILITIES

Exit from the interpretive-execution mode is part of the unit of operation in which an interception or interruption condition is recognized.

Exit from the interpretive-execution mode occurs before the host CPU enters the stopped state. The host PSW address designates the START INTERPRETIVE EXECUTION instruction (or an EXECUTE instruction, as appropriate). Thus, when the CPU is in the interpretive-execution mode, the host stop function is handled like a host interruption, except that there is no exchange of host PSWs. Similarly, exit from the interpretive-execution mode occurs before completing a host reset operation. If a host CPU reset is initiated while the CPU is in the interpretive-execution mode, the contents of the state description and various host registers, including the host PSW, the general registers, the prefix register, the CPU timer, and the clock comparator, are unpredictable. If the host rate control is in the instruction-step position when the START INTERPRETIVE EXECUTION instruction is executed, the CPU enters the interpretive-execution mode to execute a guest unit of operation, a guest interruption, or an interception, or to update the guest interval timer; then, the CPU exits from the interpretive-execution mode.

On entry to the interpretive-execution mode, a guest interruption or intervention request has higher priority than the execution of a guest unit of operation. The definition of a unit of operation applicable to the guest is the same as in native mode. Host asynchronous interruptions are recognized between guest units of operation. Machine-check interruptions occur after exit from the interpretive-execution mode is accomplished.

If a machine-check interruption occurs while the checkpoint-synchronization action that is part of the entry to the interpretive-execution mode is being performed, then execution of the START INTERPRETIVE EXECUTION instruction is nullified. If a machine-check interruption resulting from an exigent condition occurs during execution of a guest checkpoint-synchronization action in the interpretive-execution mode, the guest state and the values stored in the state description are unpredictable. The checkpoint-synchronization action on exit from the interpretive-execution mode occurs at completion or partial completion of the START INTERPRETIVE EXECUTION instruction.

Interruptions and interceptions which become due during exit from the interpretive-execution mode are not necessarily recognized ahead of the condition which initiated the exit from the interpretive-execution mode.

A host PER I-fetch event that applies in the interpretive-execution mode is handled in the same way that a PER event for any interruptible instruction is handled. That is, a host program interruption for PER occurs when (1) the START INTERPRETIVE EXECUTION instruction is completed because of interception, or (2) the PER I-fetch event is indicated concurrently with another host program interruption, or (3) the CPU exits from the interpretive-execution mode by partially completing the START INTERPRETIVE EXECUTION instruction. The mechanism is as if a host I/O, external, or machine-check interruption were about to occur -- except that a program interruption for PER occurs instead.

IBM System/370
Extended Architecture
Interpretive Execution

READER'S COMMENT FORM

Order No. SA22-7095-1

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Coding Retrieval Legibility

If you wish a reply, give your name, company, mailing address, and date:

_____

_____

_____

_____

What is your occupation? _____

Number of latest Newsletter associated with this publication: _____

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the front cover or title page.)

SA22-7095-1

Reader's Comment Form

Cut or Fold Along Line

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department B98
P.O. Box 390
Poughkeepsie, New York 12602

Fold and tape | Please Do Not Staple | Fold and tape

IBM

Publication Number
SA22-7095-1

File Number
S370-01

Printed in U.S.A.

IBM

SA22-7095-01