GA22-7072-0
File No. S370-01

Systems

IBM System/370 Extended Facility

IBM

Preface

This publication describes the IBM System/370 extended facility or feature, which is available on all processor complexes and some models of System/370.

This manual is intended for system programmers and IBM Field Engineering personnel. The reader should be familiar with the general machine functions of System/370, as described in the *IBM System/370 Principles of Operation,* GA22-7000, and with the MVS system.

The following reading is considered corequisite:

IBM Virtual Machine Facility/370 System Extensions General Information Manual, GC20-1827

MVS/System Extensions: Debugging Handbook (Volume 2), SD23-0002

OS/VS2 Data Areas, SYB8-0606. (This document is on microfiche)

OS/VS2 MVS/System Extensions General Information Manual, GC28-0872

OS/VS2 System Logic Library, Volumes 1-7, SBOF-8210

First Edition (January 1978)

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

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, Product Publications, Department B98, PO Box 390, Poughkeepsie, N.Y., U.S.A., 12602. Comments become the property of IBM.

Contents

The IBM System/370 extended facility or feature is available on all processor complexes and some models of System/370. Developed to support the internal structure of the MVS control program, this facility enhances the capabilities and performance of the MVS system when it is used with the MVS/System Extensions program product (Program No. 5740-XE1). It contains:

Common-segment bit
Low-address protection
INVALIDATE PAGE TABLE ENTRY instruction
TEST PROTECTION instruction
Four lock-handling instructions
Six tracing instructions
FIX PAGE instruction
SVC ASSIST instruction
Virtual-machine extended-facility assist

The common-segment-bit function permits a single segment-table entry in the translation-lookaside buffer (TLB) to serve for a segment shared by all segment tables. Thus, separate TLB entries are not required for each segment table sharing that common segment. Use of the common-segment-bit function increases the effective size of the TLB and thus improves CPU performance.

Low-address protection can be used to prevent system programs, which are executed with a PSW key of zero, from modifying the contents of the first 512 bytes of main storage. Since these locations contain information critical to proper operation of the entire system, the use of low-address protection can prevent system failures caused by inadvertent modifications in this area by the CPU program.

Use of the instruction INVALIDATE PAGE TABLE ENTRY permits the invalidation of a single page-table entry in the translation-lookaside buffers of all configured CPUs. It is therefore not necessary, when a page is stolen, to purge the entire TLB in order to eliminate a single entry. For configurations containing more than one CPU sharing main storage, it is also not necessary to use programmed inter-CPU communication via the SIGNAL PROCESSOR instruction when a page-table entry is invalidated. Use of INVALIDATE PAGE TABLE ENTRY in a demand-paging environment improves CPU performance.

Supervisor services test for potential protection violations before performing services on behalf of application programs. TEST PROTECTION can be used to perform such tests without causing program interruptions for protection exceptions. When TEST PROTECTION is used, it is not necessary to establish and release functional recovery routines before and after such testing. The elimination of the overhead associated with such recovery routines, when testing for user protection errors, improves CPU performance.

Twelve instructions are provided that depend on the particular conventions, fields, and control-block formats of the MVS system. Use of these 12 instructions in the MVS/System Extensions program product improves CPU performance.

The virtual-machine extended-facility assist permits the 12 MVS-dependent instructions to be executed directly by a virtual machine without program interruptions to invoke routines for their simulation. Use of the virtual-machine extended-facility assist for these instructions thus eliminates the CPU performance degradation that would otherwise arise from the use of simulation techniques. This assist is used by the VM/370 System Extensions program product (Program No. 5748-XE1).

COMMON-SEGMENT BIT

Bit 30 of the segment-table entry is defined as the common-segment bit. When this bit is a one, it indicates that the associated segment is common to all segment tables. The entry in the translation-lookaside buffer (TLB) formed from a segment-table entry with the common bit on may be used for the segment index and segment size for which it was fetched, regardless of the value of the segment-table origin in control register 1.

When bit 30 of the segment-table entry is zero, the rules for the use of the TLB entry are those that apply when the System/370 extended facility is not installed.

LOW-ADDRESS PROTECTION

Low-address protection, when active, prohibits instructions from storing into locations 0 through 511. The range test is applied before any dynamic address translation or prefixing is performed. Low-address protection is active whenever bit 3 of control register 0 is one.

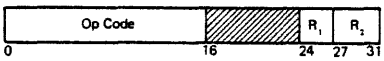
Low-address protection is not applied to accesses made by the CPU or channel for interruptions, updating the interval timer, logout, storing the CSW by an I/O instruction or interruption, storing channel identification during execution of STORE CHANNEL ID (STIDC), and the initial-program-loading and store-status functions, nor is it applied to data stores by a channel during I/O data transfer.

If storing is attempted to a location protected by low-address protection, the protected area is unchanged, and a program interruption for protection occurs, with suppression or termination of the operation.

ATTRIBUTES OF SYSTEM/370 EXTENDED-FACILITY INSTRUCTIONS

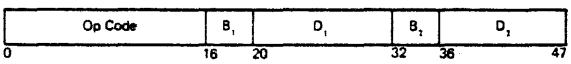
The System/370 extended facility provides 14 instructions. The two instructions INVALIDATE PAGE TABLE ENTRY and TEST PROTECTION are not dependent on specific conventions and control-block formats of the MVS control program. The other 12 instructions depend on the conventions and control-block formats of the MVS control program. Although these 12 MVS-dependent instructions will operate in any environment, they may not be useful when used outside MVS. These 12 MVS-dependent instructions include four lock-handling instructions, six trace instructions, the FIX PAGE instruction, and the SVC ASSIST instruction.

The INVALIDATE PAGE TABLE ENTRY instruction uses the RRE format:



This format permits two general registers to be specified, as for RR-format instructions. However, the RRE format uses a 16-bit operation code.

The 12 MVS-dependent instructions and TEST PROTECTION use the SSE instruction format of the form;



This format provides for addressing a first and second operand in a manner identical to that used for SS instructions. It differs from the SS format in that the operation code consists of 16 bits.

Operand Addressing

In addition to the two operands explicitly designated by the $D_1(B_1)$ and $D_2(B_2)$ fields, many System/370 extended-facility instructions use operands at specific, fixed main-storage locations or pointed to by general registers. Furthermore, many operands fetched from main storage are in turn used for addressing still other operands.

In EC mode, the translation-mode bit of the PSW controls address translation for all main-storage operand references of all System/370 extended-facility instructions, except for INVALIDATE PAGE TABLE ENTRY.

MAPL Control Block

The two System/370 extended-facility instructions FIX PAGE and SVC ASSIST use a special 16-word parameter list (MAPL) as an implicit operand. The first eight words are used by SVC ASSIST, and the remainder of the MAPL is used for FIX PAGE. This parameter list must be aligned on a doubleword boundary, and the last three bytes of the word at location A4 hex contain the address of the MAPL.

The MAPL layout is:

Offset			
Decimal	*(Hex)*	*Name*	*Content*
0-3	(0-3)	MPLSVTA	SVC-table address
4-7	(4-7)	MPLSVRBP	Prefix length in bytes of an SVRB
8-11	(8-B)	MPLESVC1	Entry address for SVCs of type 1
12-15	(C-F)	MPLRSVC1	Exit address for SVCs of type 1
16-19	(10-13)	MPLESVC2	Entry address for SVCs of types 2, 3, and 4
20-23	(14-17)	MPLRSVC2	Exit address for SVCs of types 2, 3, and 4
24-27	(18-1B)	MPLESVC6	Entry address for SVCs of type 6
28-31	(1C-1F)	MPLRSVC6	Exit address for SVCs of type 6
32-33	(20-21)	MPLLCSA	16 times lowest virtual common page-frame number
34-35	(22-23)	MPLLPRIV	16 times lowest virtual private page-frame number
36-39	(24-27)	MPLPFTP	Page-frame-table origin
40-41	(28-29)	–	Reserved
42-43	(2A-2B)	MPLMAXFX	Maximum number of pages to be fixed
44-47	(2C-2F)	MPLCNTRS	Address of PVT + X'720'
48-51	(30-33)	–	Reserved
52-55	(34-37)	MPLPFAL	Entry address to move a page to a preferred frame
56-59	(38-3B)	MPLPFCM	Entry address if MPLMAXFX is reached
60-63	(3C-3F)	–	Reserved

Lock-Interface Table

The four lock-handling instructions may access the lock-interface-table prefix. This table is located by an address contained in the word in main storage following the second-operand word. The prefix of the lock-interface table consists of four words, arranged as follows:

Offset		
Decimal	*(Hex)*	*Field Name*
−16	(−10)	LITOLOC
−12	(−C)	LITRLOC
−8	(−8)	LITOCMS
−4	(−4)	LITRCMS

PROGRAM INTERRUPTIONS

PER Events

The System/370 extended-facility instructions are subject to PER interruption controls, except that branch events are not recognized when these instructions cause branching.

Privileged-Operation Exceptions

All System/370 extended-facility instructions are privileged.

INSTRUCTIONS

The instructions described in this section are listed in the table "Instruction Summary" together with their operation codes and the program-interruption conditions that can be recognized when they are executed.

In the format shown in the instruction description, the op code is given in hex, which is signified by enclosing its value in single quotation marks ('XXXX').

Name	Characteristics							Code
INVALIDATE PAGE TABLE ENTRY	RRE	XF	M	A_1				B221
TEST PROTECTION	SSE	XF	M	A_2				E501
OBTAIN LOCAL LOCK	SSE	XF	M	A	SP	R	ST	E504
RELEASE LOCAL LOCK	SSE	XF	M	A	SP	R	ST	E505
OBTAIN CMS LOCK	SSE	XF	M	A	SP	R	ST	E506
RELEASE CMS LOCK	SSE	XF	M	A	SP	R	ST	E507
TRACE SVC INTERRUPTION	SSE	XF	M	A	SP		ST	E508
TRACE PROGRAM INTERRUPTION	SSE	XF	M	A	SP		ST	E509
TRACE INITIAL SRB DISPATCH	SSE	XF	M	A	SP		ST	E50A
TRACE I/O INTERRUPTION	SSE	XF	M	A	SP		ST	E50B
TRACE TASK DISPATCH	SSE	XF	M	A	SP		ST	E50C
TRACE SVC RETURN	SSE	XF	M	A	SP		ST	E50D
FIX PAGE	SSE	XF	M	A	SP	R	ST	E502
SVC ASSIST	SSE	XF	M	A	SP	R	ST	E503

Legend:

A	Access exceptions
A_1	Addressing, protection, and translation-specification exceptions
A_2	Addressing and translation-specification exceptions only
M	Privileged-operation exception
R	PER general-register-alteration event
RRE	RRE instruction format
SP	Specification exception
SSE	SSE instruction format
ST	PER storage-alteration event
XF	System/370 extended facility

Instruction Summary

INVALIDATE PAGE TABLE ENTRY

IPTE R_1,R_2 *[RRE]*

The register designated by the R_1 field contains a segment-table entry, of which only the page-table origin is used. The register designated by the R_2 field contains a virtual address, of which only the page index is used. A program interruption for translation specification occurs if the format of bits 8-12 of control register 0 is invalid. Otherwise, the real address of a page-table entry is computed using the two operands. This entry is accessed, using the PSW key, to fetch a page address and to store a one in the page-invalid bit.

The translation-lookaside buffers (TLBs) of all configured CPUs are purged of copies of page-table entries for which all of the following conditions hold:

1. At the time the copy was formed, the value of bits 8-12 of control register 0 in the CPU containing the TLB copy was the same as the current value of bits 8-12 of control register 0 of the CPU executing IPTE.
2. The page-table origin and page-index values used to compute the address of the page-table entry from which the TLB copy was made are the same as the page-table origin and page-index values of the operands of the IPTE instruction being executed.
3. The page address in the TLB copy is the same as that in the page-table entry invalidated by the IPTE instruction being executed.

Each receiving CPU purges its TLB between units of operation after having completed all stores that use TLB entries to be purged.

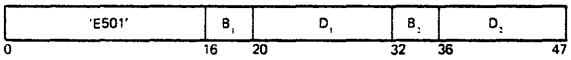
Condition Code: The code is unchanged.

Program Exceptions:

Operation (if System/370 extended feature not installed)
Privileged operation
Protection
Addressing
Translation specification

TEST PROTECTION

TPROT $D_1(B_1),D_2(B_2)$ [SSE]



Using bits 24-27 of the second-operand address as a test key, the first-operand location is tested for protection-exception conditions, including low-address protection if it is active. The condition code is set to reflect the test result.

Condition Code:

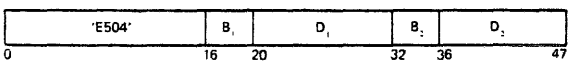
0 Both fetching and storing are permitted with the test key
1 Fetching but not storing is permitted with the test key
2 Neither fetching nor storing is permitted with the test key
3 A page- or segment-translation exception was encountered in attempting to translate the first-operand address.

Program Exceptions:

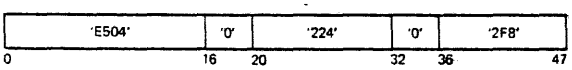
Operation (if System/370 extended feature not installed)
Privileged operation
Addressing
Translation specification

OBTAIN LOCAL LOCK

General Form



Form Used in MVS/System Extensions Program Product



If the local lock in the ASCB addressed by the first-operand word is not held, it is replaced, using an interlocked update with the value from PSALCPUA; the local-lock bit of the highest lock-held-indicator word fetched from the second-operand location is set to one; and zeros are placed in general register 13.

Otherwise, the updated instruction address, prefixed by eight zeros, is placed in general register 12; the contents of LITOLOC are placed in general register 13; and bits 8-31 of the contents of LITOLOC are placed in the instruction-address portion of the PSW.

Serialization occurs before fetching the local lock and, if the lock is obtained, again after the lock is updated.

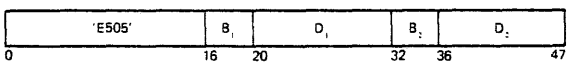
Condition Code: The code remains unchanged.

Program Exceptions:

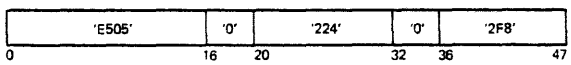
Operation (if System/370 extended feature not installed)
Privileged operation
Access
Specification

RELEASE LOCAL LOCK

General Form



Form Used in MVS/System Extensions Program Product



If the highest lock-held-indicator word fetched from the second-operand location shows that the executing CPU holds the local lock and does not hold a CMS lock, and if the word after the local lock word in the ASCB addressed by the first-operand word is zero, then the doubleword containing the lock is set to zero using interlocked update. Also, the local-lock bit of the highest lock-held-indicator word is set to zero, and zeros are placed in general register 13.

Otherwise, the updated instruction address, prefixed by eight zeros, is placed in general register 12; the contents of LITRLOC are loaded into general register 13; and bits 8-31 of the contents of LITRLOC are placed in the instruction address portion of the PSW.

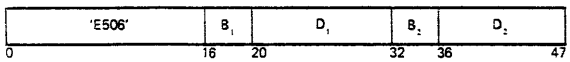
Condition Code: The code remains unchanged.

Program Exceptions:

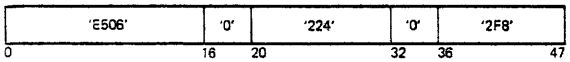
Operation (if System/370 extended feature not installed)
Privileged operation
Access
Specification

OBTAIN CMS LOCK

General Form



Form Used in MVS/System Extensions Program Product



If the highest lock-held-indicator word fetched from the second-operand location shows that the executing CPU holds the local lock and does not hold a CMS lock, and if the CMS lock addressed by bits 8-31 of general register 11 is not held, the lock is replaced, using an interlocked update, with the first-operand word. Also, the highest lock-held indicator is set to show that a CMS lock is held, and zeros are placed in general register 13.

Otherwise, the updated instruction address, prefixed by eight zeros, is placed in general register 12; the contents of LITOCMS are loaded into general register 13; and bits 8-31 of the contents of LITOCMS are placed in the instruction-address portion of the PSW.

Serialization occurs before fetching the lock and, if the lock is obtained, again after the lock is updated.

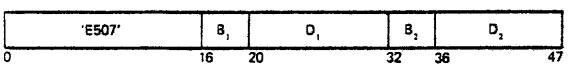
Condition Code: The code remains unchanged.

Program Exceptions:

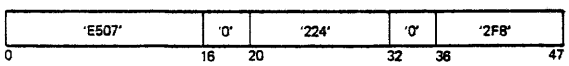
Operation (if System/370 extended feature not installed)
Privileged operation
Access
Specification

RELEASE CMS LOCK

General Form



Form Used in MVS/System Extensions Program Product



If the contents of the CMS lockword addressed by bits 8-31 of general register 11 equal the contents of the first-operand word and if the word after the CMS lockword is zero, then the doubleword containing the lockword is set to zero using interlocked update. Also, the highest lock-held-indicator word fetched from the second-operand location is set to show that no CMS lock is held, and zeros are placed in general register 13.

Otherwise, the updated instruction address, prefixed by eight zeros, is placed in general register 12; the contents of LITRCMS are loaded into general register 13; and bits 8-31 of the contents of LITRCMS are placed in the instruction address portion of the PSW.

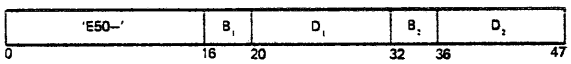
Condition Code: The code remains unchanged.

Program Exceptions:

Operation (if System/370 extended feature not installed)
Privileged operation
Access
Specification

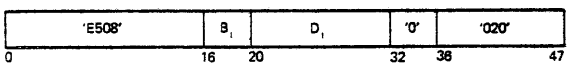
TRACE INSTRUCTIONS

General Form

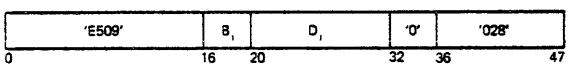


Specific Forms Used in MVS/System Extensions Program Product

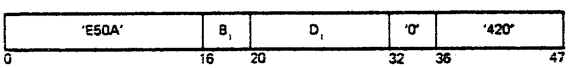
TRACE SVC INTERRUPTION



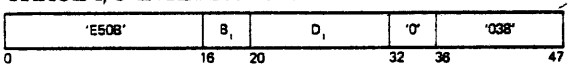
TRACE PROGRAM INTERRUPTION



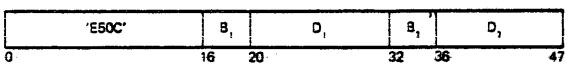
TRACE INITIAL SRB DISPATCH



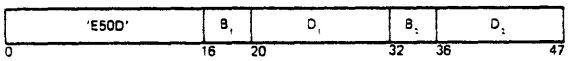
TRACE I/O INTERRUPTION



TRACE TASK DISPATCH



TRACE SVC RETURN



All six TRACE instructions use implicit logical addresses less than 4,095. TRACE INITIAL SRB DISPATCH uses the contents of bit positions 8-31 of general register 0 as the address of an SRB to access fields in that SRB. TRACE TASK DISPATCH uses the contents of bit positions 8-31 of general register 3 as the address of an RB plus 16 (10 hex) to access fields in that RB. The first operand of all TRACE instructions is a halfword field. The second operand of all TRACE instructions is the PSW to be traced.

The word at location 84 (54 hex) contains the address of the trace-table-entry (TTE) header. The trace-table-entry header has doubleword alignment and consists of three words. Each TRACE instruction performs an interlocked update of the first word of the header. The updated value contains the address of a trace-table entry (TTE). Each TRACE instruction stores 32 bytes in a TTE, which must be on a 32-byte boundary. Condition code 0 is set, and the updated word in the TTE header is the old value plus 32 (20 hex) unless that value is logically greater than or equal to the value of the third word in the TTE header. In the latter case, condition code 1 is set, and the updated TTE header word contains the value from the second word of the TTE header.

The accompanying TTE diagram contains a row for each field of a 32-byte TTE and a column for each TRACE instruction. For a particular row and column, the entry shows from which register, operand, PSA field, or other source a value is copied into the TTE at that offset (row) for that TRACE instruction (column).

Condition Code:

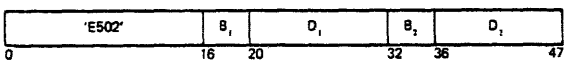
0 TTE placed at next sequential location.
1 TTE placed at table start because table wrapped.
2 –
3 –

Program Exceptions:

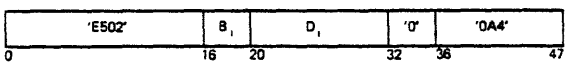
Operation (if System/370 extended feature not installed)
Privileged operation
Access
Specification

FIX PAGE

General Form



Form Used in MVS/System Extensions Program Product



Decimal (Hex) Byte Offsets within Trace-Table Entry		Instruction					
		TRACE SVC INTERRUPTION	TRACE PROGRAM INTERRUPTION	TRACE INITIAL SRB DISPATCH	TRACE I/O INTERRUPTION	TRACE TASK DISPATCH	TRACE SVC RETURN
Bytes 0-1		2nd Operand Bytes 0-1	2nd Operand Bytes 0-1	2nd Operand Bytes 0-1	2nd Operand Bytes 0-1	2nd Operand Bytes 0-1	2nd Operand Bytes 0-1
Byte 2	Bits 0-3	'2'	'3'	'4'	'5'	'7'	'8'
	Bits 4-7	'0'	'0'	'0'	FLCIOAA Bits 12-23	'0'	'0'
Byte 3		FLCSVCN Byte 1	FLCPICOD Byte 1	'0'		RBINTCOD Byte 1	2nd Operand Byte Minus 17
Bytes 4-7		2nd Operand Bytes 4-7	2nd Operand Bytes 4-7	2nd Operand Bytes 4-7	2nd Operand Bytes 4-7	2nd Operand Bytes 4-7	2nd Operand Bytes 4-7
Bytes 8-9		General Register 15	General Register 15	'0'	FLCCSW	General Register 15	General Register 15
Bytes 10-11 (A-B)				SRBPASID			
Bytes 12-15 (C-F)		General Register 0	FLCTEA	General Register 0		General Register 0	General Register 0
Bytes 16-19 (10-13)		General Register 1	General Register 1	General Register 1	'0'	General Register 1	General Register 1
Byte 20 (14)	Bits 0-1	FLCSILC Bits 5-6	FLCPIILC Bits 5-6	'0'	2nd Operand Byte 2	RBINLNTH Bits 5-6	2nd Operand Byte Minus 19 Bits 5-6
	Bits 2-7	2nd Operand Bits 18-23	2nd Operand Bits 18-23			2nd Operand Bits 18-23	2nd Operand Bits 18-23
Byte 21 (15)		Second byte of the two-byte CPU address					
Bytes 22-23 (16-17)		1st Operand Bytes 0-1	1st Operand Bytes 0-1	1st Operand Bytes 0-1	1st Operand Bytes 0-1	1st Operand Bytes 0-1	1st Operand Bytes 0-1
Bytes 24-27 (18-1B)		PSATOLD	PSATOLD	SRBPTCB	PSATOLD	PSATOLD	PSATOLD
Bytes 28-31 (1C-1F)		Bytes 3-6 of doubleword that would be stored by the instruction STORE CLOCK					

Trace-Table-Entry Diagram

The contents of general register 1 are used as the virtual address of a location in a page to be fixed. The contents of general register 2 are used as the virtual address of a location in the last of a group of consecutive pages to be fixed. The contents of general register 0 are used as the real address of a location in the page frame containing the page to be fixed. The second operand is a word containing the address of the MAPL control block.

If the page to be fixed is a nucleus page, an LSQA/SQA page, or a V=R page, then the fix count for the page frame containing that page is not examined or incremented. If the fix count for the page frame containing the page to be fixed is zero, if the page frame is not in the preferred area, and if the page is either a private page of a second-level preferred user or a common page, then execution is completed by loading the next instruction address prefixed by eight zeros in general register 14; by loading the contents of MPLPFAL in general register 15; and by placing the contents of bits 8-31 of the contents of the MPLPFAL in the instruction-address part of the PSW. Otherwise, the fix count for the page frame containing the page to be fixed is incremented by one. If the incremented fix count is one, then the total system count of fixed frames is incremented by one, and either the number of private pages fixed in the current address space or the number of frames allocated to fixed common pages is incremented by one, depending on whether the page being fixed is a private or common page, respectively.

If the virtual address of the page fixed is less than the virtual address of the last page of the consecutive group of pages to be fixed, then general register 1 is incremented by 4,096, and execution is completed by placing the first-operand address in the instruction-address part of the PSW. Otherwise, the total system count of fixed frames is compared with the maximum fix-count threshold. If the threshold has been reached, instruction execution is completed by loading the next instruction address prefixed by eight zeros in general register 14; by loading the contents of MPLPFCM in general register 15; and by placing bits 8-31 of the contents of MPLPFCM in the instruction-address part of the PSW. Otherwise, general register 15 is set to zero, and normal instruction sequencing proceeds with the updated instruction address.

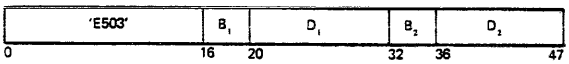
Condition Code: The code remains unchanged.

Program Exceptions:

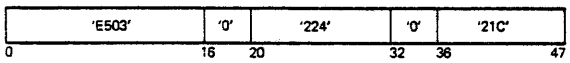
Operation (if System/370 extended feature not installed)
Privileged operation
Access
Specification

SVC ASSIST

General Form



Form Used in MVS/System Extensions Program Product



The first and second operands are words containing the addresses of the current ASCB and TCB, respectively.

Main-storage locations, which contain the MVS system status for the last SVC interruption, are tested to determine whether SVC-ASSIST action is to be taken. If SVC-ASSIST action is not taken, instruction execution is completed with normal instruction sequencing. No assist action is taken unless the CPU was enabled prior to the last SVC interruption, and a task is currently dispatched that holds no locks and for which SVC screening is not activated.

A type-1 SVC request is assisted only if the request is for an assistable type-1 function, if the only lock needed for the requested SVC function is the local lock, and if an attempt to obtain the local lock is successful.

A type-2, type-3, or type-4 SVC request is assisted only if the request is for an assistable function of type 2, 3, or 4, and if an attempt to dequeue an SVRB from the SVRB pool of the current address space is successful.

A type-6 SVC request is assisted only if the request is for an assistable type-6 function for which no locks are needed.

Assist action consists in copying the information stored at the last SVC interruption into the current request block, saving all 16 general registers, loading the general registers as shown below, and then loading the instruction address portion of the PSW from bit positions 8-31 of general register 9.

GR No.	*Contents Loaded on Assist Action*
3	Address of CVT
4	Second operand
5	Address of current RB for type 1 or type 6; address of the acquired SVRB for type 2, 3, or 4
6	SVC entry-point address
7	First operand
8	Address of the RB for the program that was being executed at SVC interruption
9	Entry address from that MAPL field that is appropriate to the SVC type
11	Address of the MBCB part of the SVRB acquired; for type 2, 3, or 4 only
12	Address of requested SVCTABLE entry
13	Exit address from that MAPL field that is appropriate to the SVC type

Condition Code: The code remains unchanged.

Program Exceptions:

Operation (if System/370 extended feature not installed)
Privileged operation
Access
Specification

VIRTUAL-MACHINE EXTENDED-FACILITY ASSIST

The virtual-machine extended-facility assist is a part of the System/370 extended facility. This assist improves the performance of virtual machines when they execute any of the 12 MVS-dependent instructions for the System/370 extended facility.

This assist is active when bit positions 1 and 29 of control register 6 contain zero and one, respectively. Bit position 1 of control register 6 serves as the virtual-machine problem-state bit (zero indicates the supervisor state, and one indicates the problem state). The following table summarizes how bit position 15 of the PSW and bit positions 1 and 29 affect the actions that result when an MVS-dependent instruction is encountered.

PSW Bit 15	*CR Bit 1*	*CR Bit 29*	*Action*
0	0 or 1	0 or 1	Execute per instruction definition
1	0 or 1	0	Privileged-operation exception
1	1	1	Privileged-operation exception
1	0	1	Execution assisted

Assist action for the MVS-dependent instructions consists in their being executed exactly as if the PSW had contained a zero in the problem state bit. INVALIDATE PAGE TABLE ENTRY and TEST PROTECTION are not assisted, and all attempts to execute either instruction by virtual machines result in program interruptions for privileged-operation exceptions.

Note: The virtual-machine extended-facility assist is separate from the virtual-machine assist (VMA). Either facility or both facilities may be active simultaneously. Both assists use bit 1 of control register 6 as the problem-state bit of the virtual machine; hence, these facilities are mutually compatible.

DIRECT CONTROL AND SYSTEM/370 EXTENDED FACILITY

On CPUs with both the direct-control facility and the System/370 extended facility installed, the first-operand address of READ DIRECT and WRITE DIRECT is a real address. Also, storage-alteration events of the program-event-recording (PER) facility are not recognized when READ DIRECT is executed. This differs from the definition of direct control when the System/370 extended facility is not installed, where the first-operand address of READ DIRECT and WRITE DIRECT is subject to dynamic address translation and the execution of READ DIRECT is subject to PER monitoring for storage alteration.

Where more than one page reference is given, the major reference is first.

IBM System/370 Extended Facility (File No. S370-01) Printed in U.S.A. GA22-7072-0

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

IBM System/370
Extended Facility

Order No. GA22-7072-0

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM shall have the nonexclusive right, in its discretion, to use and distribute all submitted information, in any form, for any and all purposes, without obligation of any kind to the submitter. Your interest is appreciated.

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 and mailing address:

_____

_____

_____

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.)

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure-sensitive or other gummed tape to seal this form.

Reader's Comment Form

Cut or Fold Along Line

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

FIRST CLASS
PERMIT NO. 419
POUGHKEEPSIE,
NEW YORK

Business Reply Mail
No postage stamp necessary if mailed in U.S.A.

Postage will be paid by:

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 System/370 Extented Facility (File No. S370-01) Printed in U.S.A. GA22-7072-0

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601