GA22-6955-1
File No. S/370-13

Systems

709/7090/7094/7094 II Compatibility Feature for IBM System/370 Models 165, 165 II, and 168

IBM

Preface

This publication contains information about the IBM 7094 compatibility feature (#7119), which adds interpretive facilities to System/370 Models 165, 165 II, and 168 for use by the IBM 7094 emulator program. The combination of the feature and the program (referred to as the 7094 emulator) allows execution on IBM System/370 Models 165, 165 II, and 168 of programs written for the IBM 709, 7090, 7094, and 7094 II Data Processing Systems.

The IBM 7094 compatibility feature, operating in conjunction with the IBM 7094 integrated emulator program, simulates the operation of an IBM 7094 Data Processing System in an IBM System/370. Operating together, the emulator program and the compatibility feature consititute a hardware-aided simulator referred to as an *emulator.* Through the emulator, a System/370 Model 165, 165 II, or 168 is, in effect, a 7094 that executes 7094 programs. The emulator requires a minimum region or partition, when operating under a System/360 or System/370 control program, of approximately 380,000 bytes.

Prerequisite Publications

To obtain maximum benefit from this publication, the reader should be familiar with the information contained in the following publications:

IBM System/370 System Summary, GA22-7001.
IBM System/370 Principles of Operation, GA22-7000.
7094 OS Emulator on Models 165/168 Reference, GC27-6951.

Organization of This Publication

To assist the reader, the information in this publication has been organized as follows:

The "Introduction" contains a general description of emulation and provides basic information on the compatibility feature and the emulator program.

"Emulator Organization" describes the functions of the compatibility feature and the emulator program and explains their relationship to each other. This section also includes information on acceptable data formats and program/feature communication.

"Emulator Instruction Set" describes the format, application, and action of each instruction provided by the compatibility feature.

The Appendixes contain information concerning timing, conversion of graphics, and internal codes and addresses.

Second Edition (May 1973)
This is a major revision of, and obsoletes GA22-6955-0, and Technical Newsletter GN22-0427.

Abbreviations and Notation Forms

The abbreviations (other than the mnemonics of the instructions) and notation forms used in this manual are:

AC	accumulator
ASC	address store compare
BCD	binary-coded decimal
BRNS	branch not successful
BRS	branch successful
B2, D2	second-operand base and displacement of RS-instruction formats
CM	condition met
CNM	condition not met
C(x)	contents of x (example: C(FPR 2))
DAT	dynamic address translation
DIL	Do 7094 Interpretive Loop
E	execution
EA	effective address
EBCDIC	extended binary-coded decimal interchange code
FPR	floating-point register
FTM	floating-point trap mode
GR	general register
hex	hexadecimal
HSD	high-speed Do Interpretive Loop
I	instruction
ILC	instruction-length code
I/O	input/output
K	= 1,024 (in System/370)
LSDIL	low-speed Do Interpretive Loop
MCAR	maintenance control address register
MCDR	maintenance control data register
MP	multiprocessing
MQ	multiplier quotient
MTM	multiple tag mode
M1	the mask field of some instructions
nRx	n times the contents of Rx (example: 8R1)
op code	operation code
OVF	overflow
PDIL	privileged Do Interpretive Loop
PER	program event recording
ROS	read-only storage
RR	System/370 register-to-register instruction format
RS	System/370 register-and-storage instruction format
R1	first operand register of RR- and RS-instruction formats
R2	second operand register of the RR-instruction format
R3	third operand register of the RS-instruction format
S	sign
SR	storage register
SS	System/370 storage-to-storage instruction format
SU	subroutine base address
TCM	transfer conditions met
TCNM	transfer conditions not met
TTM	transfer trap mode
UNF	underflow
WCS	writable control storage
Y	the address field of a 7094 instruction (bits 21-35)
94ASC	7094 address store compare
94AR1	15-bit address register (7094)
94AR2	21-bit address register (7094)
94IC	instruction counter (7094)
94IR	instruction register (7094)

When used with its associated emulator program, the 709/7090/7094/7094 II compatibility feature facilitates the transition from IBM 709, 7090, 7094, or 7094 II Data Processing Systems to System/370. The compatibility feature adds special instructions and registers to System/370. The emulator program employs these facilities and those of the System/370 universal instruction set to simulate 7094 instructions. The compatibility feature and the emulator program together constitute the emulator.

Through the emulator, a System/370 Model 165, 165 II, or 168 interprets and executes programs written for the IBM 709, 7090, 7094, or 7094 II, with internal performance about 50 percent higher than that of the 7094 II, depending on the instructions used.

The term 7094 is used in this publication when no distinction is being made between the 709, 7090, 7094, and 7094 II.

Installation of the compatibility feature does not affect normal operation of the Model 165, 165 II, or 168. On the Model 168, the compatibility feature may be run with dynamic address translation (DAT), or in conjunction with multiprocessing (MP).

This feature also provides hardware for instruction retry. All 7094 instructions can be retried under the same conditions specified for the Models 165, 165 II, and 168. However, restrictions apply to several instructions if instruction retry occurs. (These instructions are LD9, TSX9, MVEB9, MVED9, MVDB9, and MVDD9, all described later.)

Emulation is accomplished on an instruction-by-instruction basis; therefore, each 7094 instruction is executed by way of a subroutine, except when high-speed conditions are met.

Purpose of the Emulator

The purpose of this emulator is to aid in the transition from an IBM 709, 7090, 7094, or 7094 II Data Processing System to System/370. With the emulator operating under a System/360 or System/370 control program, a System/370 Model 165, 165 II, or 168 executes current 7094 programs and programming systems with little or no reprogramming. The emulator under OS/360 control can be multiprogrammed with other user problem programs, including other 7094 programs.

Emulator Organization

COMPATIBILITY FEATURE COMPONENTS AND FUNCTIONS

The compatibility feature may be considered as having three major components: the operation-code converter, the address converter, and the emulation instruction set. The compatibility feature also provides additional registers and their associated indicators. These indicators are displayed by using the emulator microfiche frames in the system microfiche viewer.

Operation-Code Converter

The operation-code converter analyzes each 7094 operation code and develops a System/370 24-bit binary address. This address is used to branch into the emulator program routine set up to emulate that specific operation code. The process is repeated for each 7094 instruction as it is fetched from the area of main/virtual storage that emulates 7094 storage. The emulated 7094 storage and 7094 instruction subroutines are separately relocatable under operating system control. In this manner, each 7094 operation code causes the emulator program to begin execution at the unique address associated with that operation code. The addresses produced by the operation-code converter are on doubleword boundaries. A minimum separation of 16 bytes is provided between them.

Address Converter

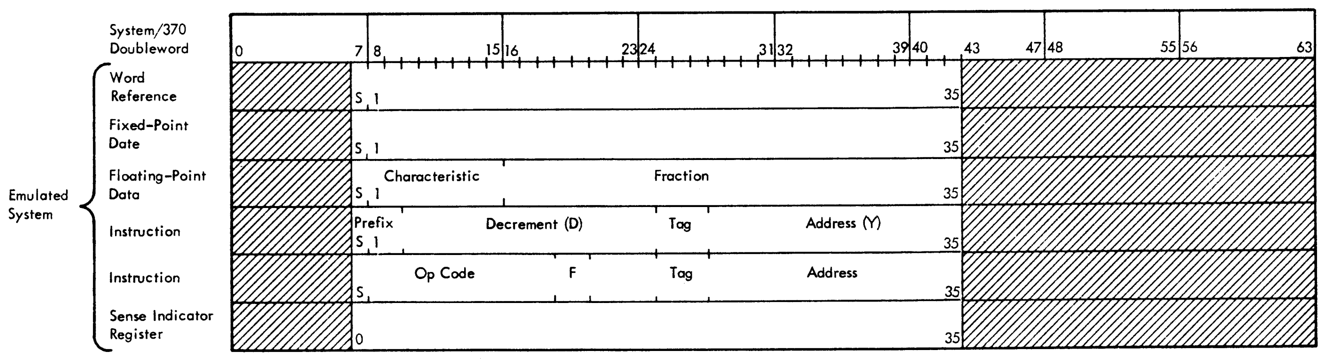
The address converter analyzes the contents of the address portion of each 7094 instruction as it is fetched, and develops a corresponding System/370 24-bit binary address from it. Reserved in main/virtual storage are 262,144 bytes to hold the 7094 core image and 16,384 bytes for the emulation subroutines. Each 7094 word is emulated by a System/370 doubleword as shown in Figure 1. 7094 storage wraparound occurs at 7094 location 77777 (octal). The conversion process includes a self-check of the address converter. An address converter error results in a machine-check interruption.

Emulator Instruction Set

The emulator instruction set contains interpretive instructions that put 7094 data or instructions in a form that is usable by System/370. It also contains corresponding instructions that are direct counterparts of certain 7094 instructions. Examples of the first type of instruction are DIL9 (Do 7094 Interpretive Loop), MVED9 (Move and Encode Decimal), and MVDD9 (Move and Decode Decimal). Examples of the second type are TIX9 (Transfer on Index), AXT9 (Address to Index True), and SLW9 (Store Logical Word).

Registers and Indicators

The compatibility feature adds to the Models 165, 165 II, and 168 certain registers and indicators that are used during emulation. The status of these registers and indicators can be displayed on the system microfiche viewer. The two most important additions to the Models 165, 165 II, and 168 data path are the 7094 instruction counter (94IC) and the 7094 instruction register (94IR). The 94IC contains the address of the next 7094 instruction to be executed. The 94IR contains the next 7094 instruction. Also added are a 15-bit address register (94AR1), a 21-bit address register



Note: Bits 0-6 and 43-63 must contain 0's for proper operation.

Figure 1. 7094-Word Formats in System/370 Storage

(94AR2), and special triggers. These registers and their associated indicators are listed in Table 1 (Appendix A) with their logout assignments. The indicators are located on microfiche panels A6, B5, and B6.

The use of most registers depends on the special instruction being executed. However, certain register assignments are fixed by the design of the compatibility feature.

REGISTER ASSIGNMENTS

The general registers (Figure 2) and floating-point registers (Figure 3) are used by the compatibility feature for communicating with the emulator program. The assignment of these registers is as follows:

GR 0 and GR 1: Working Registers

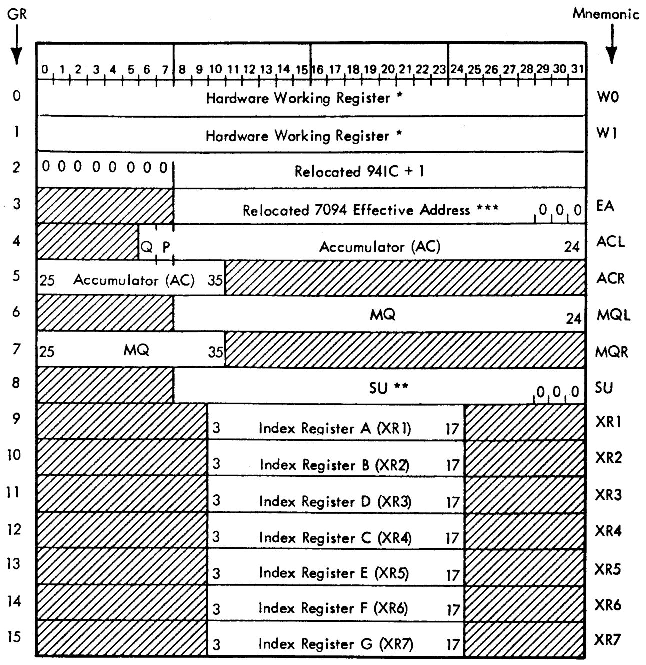
These registers are used as working registers by certain instructions described later: FAS9, DFAS9, FM9, DFM9, and FD9. The registers are not restored to their original value.

GR 2: 94IC + 1

This register (bits 8-28) is used to save the instruction count plus 1 each time a DIL9 instruction is executed. GR 2 is not updated by 7094 instructions performed in high-speed mode.

GR 3: 7094 Effective Address

The 7094 effective address (EA) is stored by DIL9 in bits 8-28 of GR 3. Bits 29-31 are all 0's to specify doubleword boundaries. In high-speed operations, GR 3 bits 4-24 contain the core relocate value from FPR 2 bits 4-24.



GR	0 1 2 3 4 5 6 7	8 9 10 11 12 13 14 15	16 17 18 19 20 21 22 23	24 25 26 27 28 29 30 31	Mnemonic
0	Hardware Working Register *				W0
1	Hardware Working Register *				W1
2	0 0 0 0 0 0 0 0	Relocated 94IC + 1			
3	▨	Relocated 7094 Effective Address ***		0 0 0	EA
4	▨ Q P	Accumulator (AC)		24	ACL
5	25 Accumulator (AC) 35	▨			ACR
6	▨	MQ		24	MQL
7	25 MQ 35	▨			MQR
8	▨	SU **		0 0 0	SU
9	▨	3 Index Register A (XR1) 17		▨	XR1
10	▨	3 Index Register B (XR2) 17		▨	XR2
11	▨	3 Index Register D (XR3) 17		▨	XR3
12	▨	3 Index Register C (XR4) 17		▨	XR4
13	▨	3 Index Register E (XR5) 17		▨	XR5
14	▨	3 Index Register F (XR6) 17		▨	XR6
15	▨	3 Index Register G (XR7) 17		▨	XR7

Signs of AC and MQ are held in hardware.

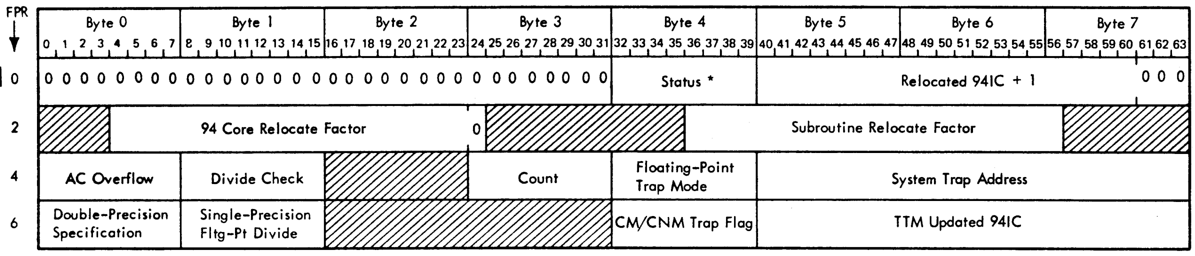
All crosshatched areas must be 0's for proper operation.

* May be used during execution of instructions provided by the emulator. When used, original contents are destroyed.

** SU = Relocated subroutine base address (in bits 8-31) or decrement (in bits 14-28).

*** Contains core relocate factor when operating in high-speed DIL mode.

Figure 2. System/370 General Register Assignments



FPR	Byte 0 (0-7)	Byte 1 (8-15)	Byte 2 (16-23)	Byte 3 (24-31)	Byte 4 (32-39)	Byte 5 (40-47)	Byte 6 (48-55)	Byte 7 (56-63)
0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	Status *	Relocated 94IC + 1		0 0 0
2	▨ 94 Core Relocate Factor			0 ▨	▨	Subroutine Relocate Factor		▨
4	AC Overflow	Divide Check	▨	Count	Floating-Point Trap Mode	System Trap Address		
6	Double-Precision Specification	Single-Precision Fltg-Pt Divide	▨	▨	CM/CNM Trap Flag	TTM Updated 94IC		

Note: Crosshatched areas must be zero for proper operation.

* Status and relocated 94IC are placed in FPR 0 on an interruption. The contents of FPR 0 have no meaning when the 7094 program is running. The status is as follows:

FPR 0 Bit	Status
32	Count mode
33	Trap
34	Transfer trap mode
35	Multiple tag mode
36	Accumulator sign
37	94ASC
38	Not used
39	MQ sign

Figure 3. System/370 Floating-Point Register Assignments

The 7094 effective address is determined in the following manner. If the 7094 instruction does not fit the Boolean expression (below) for indexing, the effective address is defined as bits 21-35 of the instruction. If the 7094 instruction does fit the indexing expression, the true address is defined as the 7094 effective address with indexing and indirect addressing taking place correspondingly.

Purpose	Boolean Expression
Indexing	$(\overline{1} \cdot \overline{2}) \cdot (\overline{8} + \overline{9}) \cdot (18 + 19 + 20)$
Indirect Accessing	$(\overline{1} \cdot \overline{2}) \cdot (\overline{8} + \overline{9}) \cdot (\overline{3} + \overline{4} + \overline{5}) \cdot 12 \cdot 13$

Note: The digits in the Boolean expressions refer to bit positions within the 7094 instruction.

Indexing also takes into account whether or not the 7094 is in multiple tag mode. After indexing and indirect addressing are accomplished, the core relocate value from FPR 2 bits 4-24 is added to provide the relocated effective address, but only if the instruction is not a ±07xx or ±005x 7094 instruction. The ±07xx and ±005x 7094 instructions use the address portion of the instruction as an operation modifier mask. Since this changes the meaning of the instruction, the relocate value is not added.

GR 4 and GR 5: Accumulator

The 7094 accumulator bits Q, P, 1-24 are stored in bit positions 6-31 of GR 4. Accumulator bits 25-35 are stored in bit positions 0-10 of GR 5. The sign position is in a separate hardware latch.

GR 6 and GR 7: MQ Register

The 7094 MQ register bits 1-24 are stored in bit positions 8-31 of GR 6. MQ bits 25-35 are stored in bit positions 0-10 of GR 7. The MQ sign is in a separate hardware latch. Bits 6 and 7 of GR 6 may be set to indicate MQ overflow or underflow as a result of a floating-point operation; they are reset by the emulator interruption routine (program).

GR 8: Subroutine Address or Decrement

The relocated subroutine base address (SU) is stored in bit positions 8-31 (bits 0-7 are set to 0's) for all 7094 instructions that do not have a 1 in position 1 or 2 of the operation field; bits 29-31 are 0's. Storage blocks of 16 bytes are allocated to each subroutine. The subroutine relocate value is obtained from FPR 2 bits 36-56 (the high-order 21 bits of the 24-bit address). If the 7094 instruction has a 1 in position 1 or 2 of the operation field, the decrement field (bits 3-17 of the 7094 instruction) is stored in bit positions 10-24 of GR 8.

GR 9-GR 15: Index Registers

Bits 3-17 of the index registers are stored in bit positions 10-24 of GR 9-GR 15. (For index register assignment, see Figure 2.)

FPR 0: Status and 7094 Instruction Counter

FPR 0 bits 32-39 are used to save the status of the 7094 program when leaving an emulator program. Bits 40-60 are used to save the contents of the 94IC. When entering an emulator program for the first time after an interruption, an emulator prefix instruction is executed and causes these values to be restored to the proper triggers and registers.

FPR 2: 7094 Core Relocate Value and Subroutine Relocate Value

FPR 2 bits 4-24 must contain the 7094 core relocate value, and bits 36-56 must contain the subroutine relocate value. Bit 24 must be 0. Both relocate values are the high-order 21 bits of a 24-bit address.

FPR 4: 7094 AC Overflow and Divide-Check Indicators, DIL Counter, Floating-Point Trap Mode, System Trap Address

The 7094 AC (accumulator) overflow indicator is located in byte 0 of FPR 4 (Figure 3). This byte is all 1's when the indicator is on. It is set and reset by the appropriate emulator instructions. The 7094 divide-check indicator is located in byte 1. This byte is all 1's when the indicator is on, and is reset by the emulator program.

The emulator program provides a count in byte 3. This count is reduced by 1 each time a DIL9 is executed in count mode. Since count mode forces low speed, this count can be used to cause an interruption after a fixed number of 7094 instructions has been executed. Count mode is entered with an SM9 instruction. When a 7094 trap is serviced by a DIL9, the count is *not* reduced.

The floating-point trap mode indicator is located in byte 4 of FPR 4. This byte is set and reset by programming, with the following codes:

Not 7094 II and not in floating-point trap mode	00000000
Not 7094 II and in floating-point trap mode	11111111
7094 II	00000001

The system trap address (bytes 5-7) contains the System/370 address to which program control is transferred during execution of a normal DIL when the interruption trigger is on. If a divide-check or floating-point trap occurs, program control is transferred to the system trap address plus 8.

If byte 3 decrements to 0 during execution of a DIL9 when the emulator is in count mode, program control is transferred to the system trap address plus 16.

FPR 6: Floating-Point Trap (Special Conditions), Transfer Trap Mode Flag, Updated 7094 Instruction Counter

Byte 0 is set to 1's if the floating-point trap is caused by specifying an odd-even pair of operands for a double-precision operation. This trap can occur only if the floating-point trap mode indicator (byte 4 of FPR 4) is set and the system being emulated is other than a 7094 II.

Byte 1 is set to 1's if the instruction causing the floating-point trap is an FD9 (Single-Precision Floating-Point Divide).

Byte 4 is reserved for a 7094 transfer trap mode flag. Bytes 5-7 are reserved for the value of the previous 7094 instruction location plus 1. For a description of the conditions under which these bytes are set, see "Transfer Trap Mode (TTM)."

MANUAL CONTROLS AND INDICATORS

Power-On Reset, IPL Reset, and System Reset Pushbuttons

Each of these pushbuttons resets all mode triggers and all control triggers.

CPU Reset Pushbutton

This pushbutton resets all control triggers but does not reset the mode triggers or the 7094 status valid trigger.

Indicators and Logout

The indicators added by the compatibility feature are listed in Table 1 in Appendix A. These indicators, located on microfiche panels A6, B5, and B6, can be displayed on the system microfiche viewer. On a logout, they are placed in the System/370 logout area.

EMULATOR PROGRAM FUNCTIONS

The emulator program is an interpretive simulator that uses the compatibility feature and other System/370 facilities. It includes appropriate routines for interpreting and simulating 7094 instructions and for providing the control and communication facilities required by the emulator. The emulator program uses the emulation instruction set and the universal instruction set to provide routines that simulate the execution-time actions of those 7094 instructions that are emulated. These routines are entered via the operation-code converter described earlier. In general, communication between the emulator program and the compatibility feature is through the general and floating-point registers.

DATA REPRESENTATION

Each 7094 36-bit word is simulated by bits 7-42 of a System/370 doubleword (Figure 1). The unused bits of each doubleword must be 0's to ensure proper operation of the compatibility feature. The accumulator (AC) and multiplier-quotient (MQ) register signs are maintained in separate latches.

Conversion of characters and internal data formats is described in detail in Appendix B.

STORAGE ALLOCATION

Reserved in System/370 main/virtual storage are 256K (262,144) bytes on a quadword boundary to hold the 7094 main/virtual storage image; also reserved are 16K (16,384) bytes on a doubleword boundary for emulation subroutines.

PROGRAM EVENT RECORDING (PER) ON MODELS 165 II and 168

With Program Event Recording (PER) masked on, certain emulator feature instructions cause successful branch PER events. These instructions are: DIL9 (except where no reference to a subroutine is made, as in a high-speed DIL); BA9 (where the contents of GR 1 are less than 15); D9, FD9, DFD9, and high-speed divides when a divide-check occurs; and DFAS9, DFM9, DFD9, DLD9, and high-speed double precision 7094 instructions (where a double precision specification occurs). (The last two groups branch to a system trap address.)

Other PER events (general register alteration, main/virtual storage alteration, and instruction fetch) are handled normally.

Emulator Instruction Set

This section contains descriptions of the primary and secondary instructions, the types of exceptions to which they are susceptible, and applicable indicators and condition codes set by execution of the instructions.

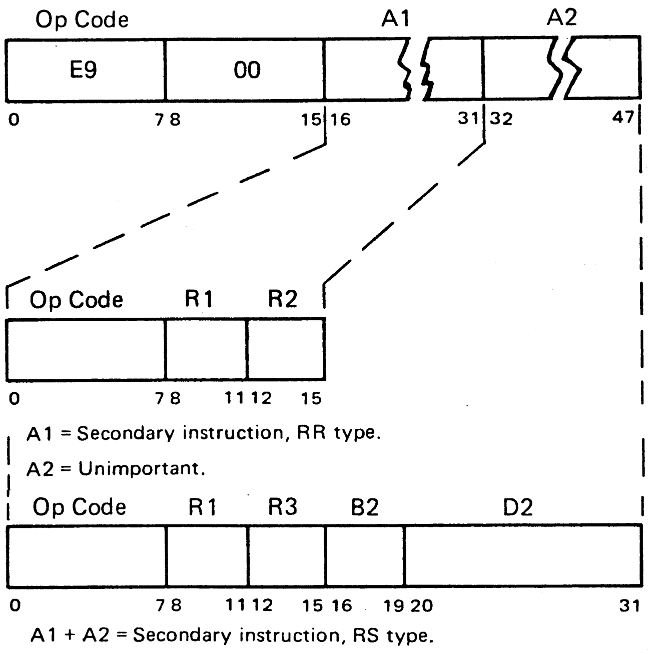
PRIMARY INSTRUCTION

The System/370 Models 165, 165 II, and 168 emulate the 7094-type instructions by way of the SS-format emulator instruction, given the mnemonic EMU. This is the only primary instruction of this feature, and has a special format. The first two bytes contain the op code E9 and the emulator flag, and the next four bytes contain any of the RR- or RS-format secondary instructions. The first two bytes are, in effect, a prefix for each of the secondary instructions.

The secondary EMU instruction must be a defined instruction and the emulator flag, bits 8-15, must be zero. If either or both of these conditions are not met, the primary SS instruction terminates with a specification exception on the Model 165 and an operation exception on the Models 165 II/168. The emulator flag is examined only when the E9 op code is actually executed to make emulator status valid. All other times the emulator flag field is ignored.

EMU (Emulator-Feature Instruction)

SS Format



A1 = Secondary instruction, RR type.
A2 = Unimportant.

A1 + A2 = Secondary instruction, RS type.

Program Interruptions:	Operation
	Access
	Specification (Model 165)

The EMU instruction causes a specification exception (Model 165), or an operation exception (Models 165 II/168), if it is the subject of the XEC9 instruction, or if byte 1 is nonzero.

Indicators: EMU status trigger

Before executing the secondary instruction, the EMU instruction examines the EMU status trigger. If the trigger is on, EMU proceeds to the secondary instruction. If it is off, EMU loads the 7094 hardware status, turns on the EMU status trigger, and begins again to execute the entire instruction.

SECONDARY INSTRUCTIONS

The secondary instructions, in conjunction with the System/370 instruction set, perform the same functions as the listed instructions from the emulated system. The R1, R2, and R3 fields are used in some of these instructions as modifiers to the operation code. Many instructions require some fixed value to ensure proper operation. C(Y) refers to the contents of a 7094 main/virtual storage word, bit positions S and 1-35.

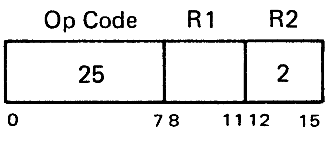
If a special instruction turns an indicator on or off, the indicator is noted at the end of the instruction description. An indicator can be a light or a byte in the FPR's; differentiation is made by identifying the indicator lights.

For all floating-point instructions, the original AC Q- and P-bits are treated as an extension of the 8-bit AC exponent. Exceptions to this are noted for the appropriate instructions. The MQ exponent is also 10 bits, with the two high-order bits (MQ pseudo Q- and P-bits) stored in GR 6 bits 6 and 7, respectively.

A floating-point trap occurs if the final AC Q- or P-bit is not 0, or if GR 6 bits 6 and 7 are not 0. If a floating-point trap does occur, the floating-point trap interruption routine in the emulator program decodes these bits and sets the 7094 AC/MQ OVF or UNF indicators. After this decoding, the interruption routine resets GR 6 byte 0.

DIL9 (Do Interpretive Loop)

RR Format



R1 = 0 = Nonprivileged DIL (DIL9) (Trap Allowed)
= 1 = Privileged DIL (PDIL9) (Trap Not Allowed)

Description

The DIL9 interprets 7094 instructions. Hardware added to speed up this interpretation process includes a 36-bit instruction register (94IR), a 21-bit 7094 instruction counter (94IC), a 15-bit address register (94AR1), and a 21-bit address register (94AR2).

The DIL9 exists in two forms, a normal DIL and a high-speed DIL (HSD). The HSD eliminates fetching a System/370 subroutine from storage if all HSD conditions exist; otherwise, a normal DIL occurs.

Normal DIL

If R1=0 (nonprivileged DIL), the next instruction to be executed is determined by the 7094 trap trigger. If the trigger is on, DIL9 branches to the trap address specified by FPR 4 bits 40-63. The 94IC and 94IR are unchanged. If the 7094 trap trigger is off, or if R1=1 (privileged DIL), then DIL9:

1. *Branches to a System/370 emulation subroutine.* DIL9 gates the 7094 operation code (held in 94IR bits S and 1-11) through a decoder into 94AR1 bits 18-27. Bit 28 of the 94AR1 is 0 to allow subroutines of 16 bytes. If the instruction uses an index register as an operand, the decoder gates the 94IR tag (bits 18-20) into 94IR1 bits 25-27 and modifies 94IR1 bits 18-24, resulting in a branch to one of eight different subroutines as designated by the tag. FPR 2 bits 36-56 are added to provide a relocated subroutine address in the 94AR2, which is gated to the System/370 instruction counter (IC), thus causing the branch.
2. *Generates the effective address and places it in GR 3.* If the 7094 instruction is ±005x, 94IR bits 21-35 are left in GR 3. If the instruction is ±07xx, 94IR bits 21-35 may be indexed, if applicable, and the result left in GR 3. In all other cases (after indexing, if applicable) FPR 2 bits 4-24 are added to 94IR bits 21-35, resulting in a relocated effective address which is left in GR 3. If the instruction is indirectly addressable, and the flag (in 94IR bits 12 and 13) contains 1's, the indirect address is fetched from storage, and bits 18-35 of the 94IR are replaced by the contents of the storage specified by the indirect address. The relocated effective address is generated and left in GR 3. In all cases, the address is in GR 3 bits 8-28, and all other GR 3 bits are 0's. (Indexing may be done before and/or after indirect addressing.)
3. *Places the SU or decrement in GR 8.* The decrement is gated from the 94IR into bit positions 10-24 of GR 8 if the operation code of the instruction in the 94IR has a 1 in position 1 or 2 (±1xxx, ±2xxx, or ±3xxx). Otherwise, the relocated subroutine base address is gated into bit positions 8-31 of GR 8. It is identical with the address gated into the System/370 instruction counter as in item 1 above.
4. *Increments the 94IC by 1 and fills the 94IR per the 94IC.* During execution of instruction "n", the 94IR contains instruction n+1, with the 94IC also at n+1. The updated 94IC (n+1) is placed into bit positions 8-28 of GR 2.
5. *Refills the 94IR if 94ASC is on.* The 94ASC (address store compare) indicator is tested; if the indicator is on, it is turned off, the 94IR is refilled from the 94IC, and the DIL is restarted. The 94ASC would have been turned on if a comparison done by the previous DIL or high-speed store was successful and the previous instruction was a store type. (A store-type instruction is a +160, +320, or ±06xx.) A full comparison during the DIL9 is made between the 94IC and the effective address of the instruction to be executed. If the comparison is successful and a store-type instruction is to be executed, the 94ASC trigger is set.
6. *Decrements FPR 4 byte 3 by 1.* In count mode, DIL9 reduces the count by 1 and tests for a count-equal-zero condition. If the count is 0 at this time, the 94IC is not updated and the transfer is to the system trap address plus 16 rather than to the subroutine address.

High-Speed DIL

If all of the following conditions are met, the DIL9 instruction will do both the 7094 instruction interpretation and execution instead of branching to a subroutine to simulate execution of the 7094 instruction:

1. The instruction in the 94IR is a high-speed instruction. (See Appendix E.)
2. The instruction is not under control of a privileged DIL.
3. The 94 trap trigger is not set.
4. The 94ASC trigger is not set.
5. Transfer trap mode is not set.
6. The instruction in the 94IR does not have a multiple tag, or multiple tag mode is not set.
7. The instruction in the 94IR does not specify an index modifier-type instruction with 0 as an index register.
8. Count mode is not set.

When the above conditions are detected, the following occurs:

1. The core relocate value in FPR 2 bits 4-24 is placed in GR 3 bits 4-24.
2. A branch to a microprogram subroutine is performed to execute the 7094 instruction (including fetching of operands).
3. The next 7094 instruction is fetched from storage (per the 94IC+1) and placed in the 94IR.
4. When execution of the 7094 instruction is complete, the next 7094 instruction is examined to determine if it also can be processed as a high-speed instruction. If it can, the above actions are repeated unless a System/370 interruption is pending. An interruption will terminate high-speed operations, which allows the interruption to

be taken. The PSW will contain the address of the last DIL9 instruction, and the ILC will either reflect the length of the System/370 instruction preceding the DIL9 in the Model 165 or, in the Model 165 II/168, will contain a value of 3.

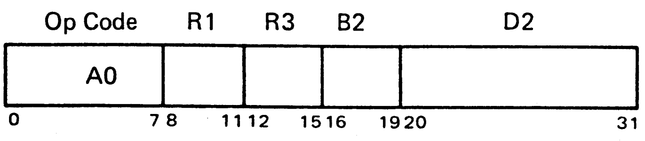
When the above conditions are not met, a normal DIL occurs. Note that when an HSD occurs no reference is made to the subroutine in main/virtual storage.

Program Interruption: Access

Indicators: 94ASC (light)
HST (light) (only on HSD operations)

STO9 (Store)

RS Format



R1 = Any even-numbered GR (normally 4)
R3 = Unimportant

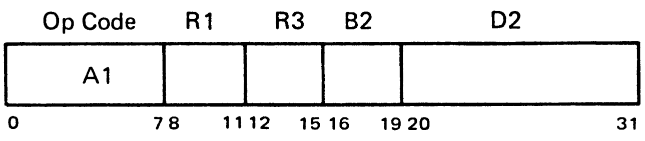
Description

This instruction is equivalent to STO (Store AC) when R1=4. The contents of two successive GR's, starting at the GR specified by R1, are stored at the second operand location, C(B2)+D2. The accumulator sign in all cases is stored in bit 7 of the second operand location.

Program Interrupton: Access
Indicators: None

STQ9 (Store MQ)

RS Format



R1 = Any even-numbered GR (normally 6)
R3 = Unimportant

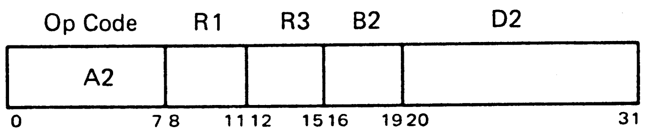
Description

This instruction is equivalent to the STQ (Store MQ) when R1=6. The contents of two successive GR's, starting at the GR specified by R1, are stored at the second operand location, C(B2)+D2. The MQ sign bit in all cases is stored in bit 7 of the second operand location.

Program Interruption: Access
Indicators: None

SLW9 (Store Logical Word)

RS Format



R1 = Any even-numbered GR (normally 4)
R3 = Unimportant

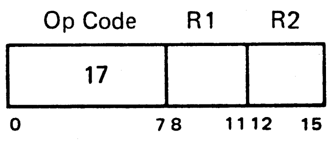
Description

This instruction is equivalent to the SLW (Store Logical Word) when R1=4. The contents of two successive GR's, starting at the GR specified by R1, are stored at the second operand location, C(B2)+D2.

Program Interruption: Access
Indicators: None

SQP9 (Set Q, P)

RR Format



R1 = Unimportant
R2 = 0=XCA (Exchange AC and MQ)
1=SSP (Set Sign Plus)
2=SSM (Set Sign Minus)
3=CHS (Change Sign)

Description

This instruction is equivalent to the instruction specified by R2.

Program Interruption: None
Indicators: None

LD9 (Load)

RS Format

Op Code	R1	R3	B2	D2
B3				
0 7	8 11	12 15	16 19	20 31

R1 = Unimportant for CLA, CLS, CAL, LDQ
= Index GR for LXA and LXD
R3 = 0 for CLA (Clear and Add)
1 for CLS (Clear and Subtract)
2 for CAL (Clear and Add Logical Word)
3 for LDQ (Load MQ)
4 for LXD (Load Index from Decrement)
5 for LXA (Load Index from Address)
B2 ≠ 4 or 6 for CLA, CLS, and CAL; ≠ R1 for LXA and LXD;
≠ 6 or 7 for LDQ
Note: The results are unpredictable if an instruction retry occurs and B2 = 4 or 5 (CLA, CLS, CAL) or B2 = 6 or 7 (LDQ) or B2 = R1 (LXA, LXD)

Description

This instruction is equivalent to the instruction specified by R3. The location of the C(Y) is specified by the second operand address, C(B2)+D2.

Program Interruption: Access
Indicators: None

ACL9 (Add and Carry Logical)

RR Format

Op Code	R1	R2
16		4
0 7	8 11	12 15

R1 = Effective address (normally 3)

Description

This instruction is equivalent to the instruction ACL (Add and Carry Logical Word).

Program Interruption: Access
Indicators: None

AS9 (Add, Subtract)

RS Format

Op Code	R1	R3	B2	D2
B5				
0 7	8 11	12 15	16 19	20 31

R1 = Any even-numbered GR (normally 4)
R3 = 0 = ADD (Add)
1 = ADM (Add Magnitude)
4 = SUB (Subtract)
5 = SBM (Subtract Magnitude)

Description

This instruction is equivalent to the instruction as specified by R3 if R1=4. The location of the C(Y) is specified by the second operand address, C(B2)+D2.

Program Interruption: Access
Indicators: AC overflow

SHFT9 (Shift)

RR Format

Op Code	R1	R2
11	3	
0 7	8 11	12 15

R2 = 0 = LLS (Long Left Shift)
1 = LRS (Long Right Shift)
2 = ALS (Accumulator Left Shift)
3 = ARS (Accumulator Right Shift)
4 = LGR (Logical Right Shift)
5 = LGL (Logical Left Shift)
6 or 7 = RQL (Rotate MQ Left)

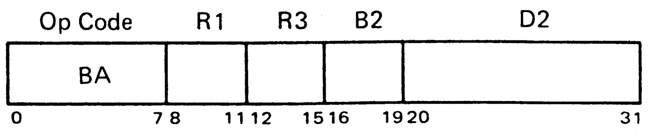
Description

This instruction is equivalent to the instruction as specified by R2. The shift amount is obtained from bit positions 21-28 of GR 3 (effective address).

Program Interruption: None
Indicators: AC overflow (LGL, LLS, and ALS)

ST9 (Store Address or Index)

RS Format



R1 = Even GR for STA (Store Address)
= Index GR affected for SXA, SXD (Store Index in Address, Store Index in Decrement)
R3 = 4 = SXA (Store Index in Address)
0 = STA (Store Address)

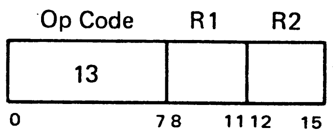
Description

This instruction is equivalent to the instruction as specified by R3. The operand, C(Y), is specified by the second operand address, C(B2)+D2.

Program Interruption: Access
Indicators: None

AXT9 (Address to Index True)

RR Format



R1 = Index register affected
R2 = Any GR (normally 3, the effective address GR)

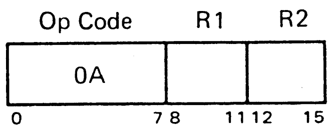
Description

This instruction is equivalent to AXT (Address to Index True). The contents of bit positions 14-31 of the GR specified by R2 are placed into bit positions 10-27 of the GR specified by R1. Zeros are inserted into bit positions 0-9 and 28-31 of the GR specified by R1.

Program Interruption: None
Indicators: None

PAX9 (Place Address in Index)

RR Format



R1 = Even source GR, normally 4 (AC)
R2 = Destination Index GR

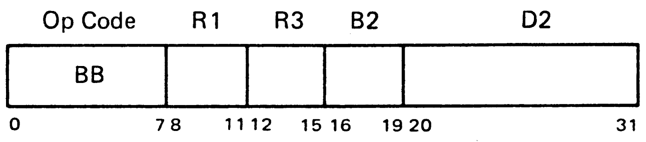
Description

This instruction is equivalent to PAX (Place Address in Index) when R1 = 4.

Program Interruption: None
Indicators: None

XEC9 (Execute)

RS Format



R1 = 0 = Normal Execute (Trap allowed)
1 = Privileged Execute (No trap allowed)
R3 = Any GR

Description

This instruction is equivalent to XEC (Execute). XEC9 incorporates DIL9 into its operation; it does not trap if R1=1. The address of the instruction to be executed is specified by the second operand address, C(B2)+D2. In addition, the contents of GR 3 plus 8 are placed in bit positions 0-31 of the GR specified by R3, thus storing the effective address. The updated 941C contents will *not* be placed into the GR specified by R2.

Note: The ASC light, if on, is turned off.

Program Interruption: Access
Indicators: ASC (light)

TIX9 (Transfer on Index)

RS Format

Op Code	R1	R3	B2	D2
BE				
0 7	8 11	12 15	16 19	20 31

R1 = Decrement GR (normally 8)
R3 = Index register affected (See Figure 2 for GR address.)
C(B2) +D2 = Effective address

Description

If the contents of the GR specified by R3 are greater than the contents of the GR specified by R1, the decrement, the number in the R3-specified GR is reduced by the decrement, the address specified by C(B2)+D2 is placed in the 94IC, and the contents of that address are placed in the 94IR. The computer then takes the next sequential System/370 instruction. If the contents of the R3-specified GR are less than or equal to the decrement, the contents of that GR are unchanged and the next sequential System/370 instruction is executed.

Program Interruption: Access
Indicators: CM-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

TXI9 (Transfer with Index Incremented)

RS Format

Op Code	R1	R3	B2	D2
A7				
0 7	8 11	12 15	16 19	20 31

R1 = Decrement GR (normally 8)
R3 = Index register affected (See Figure 2 for GR address.)
C(B2) +D2 = Effective address

Description

The decrement in bit positions 10-24 of the GR specified by R1 is added to the contents of the GR specified by R3. The address specified by C(B2)+D2 is placed in the 94IC and the contents of that address are placed in the 94IR. The next sequential System/370 instruction is executed.

Program Interruption: Access
Indicators: CN-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

TXL9 (Transfer on Index Low or Equal)

RS Format

Op Code	R1	R3	B2	D2
BC				
0 7	8 11	12 15	16 19	20 31

R1 = Decrement GR (normally 8)
R3 = Index register (See Figure 2 for GR address.)
C(B2) +D2 = Effective address

Description

If the contents of the R3-specified GR are less than or equal to the contents of the R1-specified GR, the address specified by C(B2)+D2 is placed in the 94IC and the contents of that location are placed in the 94IR. The next sequential System/370 instruction is then executed.

Program Interruption: Access
Indicators: CM-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

TXH9 (Transfer on Index High)

RS Format

Op Code	R1	R3	B2	D2
BD				
0 7	8 11	12 15	16 19	20 31

R1 = Decrement GR (normally 8)
R3 = Index register (See Figure 2 for GR address.)
C(B2) +D2 = Effective address

Description

If the contents of the GR specified by R3 are greater than the decrement in the GR specified by R1, the address specified by C(B2)+D2 is placed in the 94IC and the contents of that address are placed in the 94IR. The next sequential System/370 instruction is then executed.

Program Interruption: Access
Indicators: CM-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

TSX9 (Transfer and Set Index)

RS Format

Op Code	R1	R3	B2	D2
B1				
0 7	8 11	12 15	16 19	20 31

R1 = Not equal to R3
R3 = Index register affected (See Figure 2 for GR address.)
C(B2) +D2 = Effective address
B2 ≠ R3 Results are unpredicable if B2 = R3 when a retry occurs.

Description

This instruction subtracts 1 from the 94IC and places the 2's complement of the difference in the GR designated by R3. The address specified by C(B2)+D2 is placed in the 94IC and the contents of that address are placed in the 94IR.

The next sequential System/370 instruction is then executed.

Program Interruption: Access
Indicators: CM-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

TC9 (Transfer Condition)

RS Format

Op Code	R1	R3	B2	D2
B9				
0 7	8 11	12 15	16 19	20 31

R1 designates the FPR whose byte 0 contains the accumulator overflow flag (normally FPR 4). For TMI and TPL, R1 = 0.
R3 = 2 = TMI (Transfer on Minus)
= 6 = TPL (Transfer on Plus)
= 8 = TOV (Transfer on Overflow)
= 9 = TNO (Transfer on No Overflow)
C(B2) +D2 = Effective address

Description

If:

1. R3=2 and the accumulator sign is minus (1),
2. R3=6 and the accumulator sign is plus (0),
3. R3=8 and the accumulator overflow flag is hexadecimal FF, or
4. R3=9 and the flag is 0,

then the address specified by C(B2)+D2 is placed in the 94IC, the contents of that address are placed in the 94IR, and the next sequential System/370 instruction is executed. If R3=8 or 9, FPR 4 byte 0 is set equal to 0.

Program Interruption: Access
Indicators: AC Overflow
CM-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

M9 (Multiply)

RS Format

Op Code	R1	R3	B2	D2
AF	6			
0 7	8 11	12 15	16 19	20 31

R3 = 0 = MPY (Multiply)
= 1 = MPR (Multiply and Round)
C(B2) +D2 = Effective address

Description

This instruction is equivalent to the instruction specified by R3. The address of the C(Y) is specified by the second operand address C(B2)+D2.

Program Interruption: Access
Indicators: None

D9 (Divide)

RR Format

Op Code	R1	R2
15		6
0 7	8 11	12 15

R1 = Effective address

Description

This instruction is equivalent to the DVH (Divide or Halt) or DVP (Divide or Proceed). If a divide-check condition results from the divide operation, a trap request is initiated. The *program* determines whether the instruction is a DVH or a DVP.

Program Interruption: Access
Indicators: Divide check

FAS9 (Floating-Point Add-Subtract)

RS Format

Op Code	R1	R3	B2	D2
AB	4			
0 7	8 11	12 15	16 19	20 31

R3 = 0 = FAD (Floating Add)
1 = UFA (Unnormalized Floating Add)
2 = FAM (Floating Add Magnitude)
3 = UAM (Unnormalized Floating Add Magnitude)
4 = FSB (Floating Subtract)
5 = UFS (Unnormalized Floating Subtract)
6 = FSM (Floating Subtract Magnitude)
7 = USM (Unnormalized Floating Subtract Magnitude)

Description

This instruction is equivalent to the instruction specified by R3. The address of the C(Y) is specified by the second operand address, C(B2)+D2.

Note: For the floating-point instructions specified by the R3 field, the following occurs if the AC Q- or P-bit is not 0.

1. The prealignment of the AC and storage (SR) operands is done by assuming the AC exponent to be greater than or equal to the SR exponent. As a result, the SR fraction is shifted right per the exponent difference.
2. The operand signs are changed. The new SR sign equals the original AC sign or AC P-bit while the new AC sign equals the original SR sign.
3. The original AC Q- and P-bits are ignored for exponent calculations.

Program Interruption: Access
Indicators: None

FM9 (Floating-Point Multiply)

RS Format

Op Code	R1	R3	B2	D2
A9	6			
0 7	8 11	12 15	16 19	20 31

R3 = 0 = FMP (Floating Multiply)
= 1 = UFM (Unnormalized Floating Multiply)

Description

This instruction is equivalent to the instruction specified by R3. The address of the C(Y) is specified by the second operand address, C(B2)+D2.

Program Interruption: Access
Indicators: None

FD9 (Floating-Point Divide)

RS Format

Op Code	R1	R3	B2	D2
AE	4	0		
0 7	8 11	12 15	16 19	20 31

Description

This instruction is equivalent to the FDP or FDH instruction. The address of the C(Y) is specified by the second operand address, C(B2)+D2.

Program Interruption: Access
Indicators: Divide check
SP FP divide (single-precision floating-point divide)

DFAS9 (Double-Precision Floating-Point Add-Subtract)

RS Format

Op Code	R1	R3	B2	D2
AA	4			
0 7	8 11	12 15	16 19	20 31

R3 = 0 = DFAD (Double-Precision Floating-Point Add)
= 1 = DUFA (Double-Precision Unnormalized Floating-Point Add)
= 2 = DFAM (Double-Precision Floating-Point Add Magnitude)
= 3 = DUAM (Double-Precision Unnormalized Floating-Point Add Magnitude)
= 4 = DFSB (Double-Precision Floating-Point Subtract)
= 5 = DUFS (Double-Precision Unnormalized Floating-Point Subtract
= 6 = DFSM (Double-Precision Floating-Point Subtract Magnitude)
= 7 = DUSM (Double-Precision Unnormalized Floating-Point Subtract Magnitude)

Description

This instruction is equivalent to the instruction specified by R3. The address of the C(Y) is specified by the second operand address, C(B2)+D2.

Note: If the accumulator Q- or P-bits are not 0 for the instructions specified by the R3 field, the 8-bit accumulator exponent is treated as being greater than or equal to the SR exponent. The original Q- and P-bits are ignored for exponent calculations.

Program Interruption: Access
Indicators: Double-precision specification

DFM9 (Double-Precision Floating-Point Multiply)

RS Format

Op Code	R1	R3	B2	D2
AC	4			
0 7	8 11	12 15	16 19	20 31

R3 = 0 = DFMP (Double-Precision Floating-Point Multiply)
= 1 = DUFM (Double-Precision Unnormalized Floating-Point Multiply)

Description

This instruction is equivalent to the instruction specified by R3. The location of C(Y) is specified by the second operand address, C(B2)+D2. The results derived may be different from those derived in the 7094 systems. Where there are differences, the results obtained under emulation are more accurate because the 7094 systems data flow is 36 bits; therefore, double-precision floating-point multiply is accomplished by using a different algorithm from that used by the emulator feature, which has 54-bit fractions in its data flow.

The fractional result obtained is equal to the high-order 54 bits of the unrounded product.

Program Interruption: Access
Indicator: Double-precision specification

DFD9 (Double-Precision Floating-Point Divide)

RS Format

Op Code	R1	R3	B2	D2
AD	4	0		
0 7	8 11	12 15	16 19	20 31

Description

This instruction is equivalent to the DFDP (Double-Precision Floating-Point Divide or Proceed) or DFDH (Double-Precision Floating-Point Divide or Halt). The location of

C(Y) is specified by the second operand address, C(B2)+D2. The results derived may be different from those derived in 7094 systems. The fractional results obtained by DFD9 are equal to the true 54-bit quotient.

DFD9 generates a divide check if the 54-bit dividend fraction is equal to or greater than twice the 54-bit divisor fraction or if the divisor fraction is 0. If a divide check occurs, the accumulator bit positions S, Q, P, 1-35, and MQ positions 1-35 remain unchanged. The MQ sign is made equal to the accumulator (AC) sign.

Program Interruptions: Access
Indicators: Double-precision specification
Divide check

BC9 (Branch on Condition)

RS Format

Op Code	R1	R3	B2	D2
A6				
0 7	8 11	12 15	16 19	20 31

R1 = M1 (mask)
R3 = Unimportant
C(B1) +D2 = Effective address of the next instruction if the branch is successful.

Condition Code	*M1 Bit*
0	8
1	9
2	10
3	11

Description

The BC9 instruction is used in conjunction with System/370 instructions to branch on a condition code. A System/370 instruction can set a condition code, and the M1 field of the BC9 can select the condition code desired.

If any of the mask bits corresponding to the condition code is a 1, the address specified by C(B2)+D2 is placed in the 94IC and the contents of that address are placed in the 94IR; otherwise, the 94IC and 94IR remain unchanged. The next sequential System/370 instruction is then executed.

Program Interruption: Access
Indicators: CM-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

SKC9 (Skip on Condition)

RR Format

Op Code	R1	R2
OE		
0 7	8 11	12 15

R1 = M1 (mask)
R2 = 0 or 15 (See following description)

Condition Code	*M1 Bit*
0	8
1	9
2	10
3	11

Description

The SKC9 instruction is used with System/370 instructions to skip on a condition. A System/370 instruction can set a condition code, and the M1 field of the SKC9 can select the condition code desired. If the mask bit corresponding to the condition code is a 1, the contents of the 94IC are increased by 1 (R2=1) or 2 (R2=15), respectively, and the 94IR is set as specified by the new 94IC; otherwise, the 94IC and 94IR remain unchanged. The next sequential System/370 instruction is then executed.

Program Interruption: Access
Indicators: None

BA9 (Branch on Address)

RR Format

Op Code	R1	R2
12		2
0 7	8 11	12 15

R1 = GR Containing address (GR 3, when used with DIL9)

Description

This instruction examines bits 14-28 of the GR designated by R1. If the value of this address is greater than 14 (octal 16), the next sequential System/370 instruction is executed. If the address is less than or equal to 14 (octal 16), an automatic branch on that number takes place as shown on the following chart. This instruction is designed primarily to handle 7094 operation codes of ±0760. The instruction assumes that the relocated subroutine base address is in GR 8. If the branch is successful, GR 8 is updated with the new address.

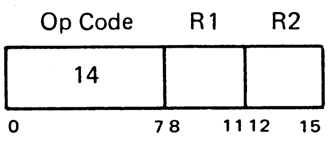
Program Interruption: Access
Indicators: None

Bits 21-35 (GR 3) (Octal)	*System/370 Address (Hex) +0760*	*0760*
0	000010	002010
1	000020	002020
2	000030	002030
3	000040	002040
4	000050	002050
5	000060	002060
6	000070	002070
7	000080	002080
10	000090	002090
11	0000A0	0020A0
12	0000B0	0020B0
13	0000C0	0020C0
14	0000D0	0020D0
15	0000E0	0020E0
16	0000F0	0020F0

Note: Add the subroutine relocate value (FPR 2 bits 36-56) to the System/370 address to obtain the address that is the branch destination and that is placed in GR 8.

SM9 (Set Mode)

RR Format



R1 and R2 (See description of instruction bits.)

Description

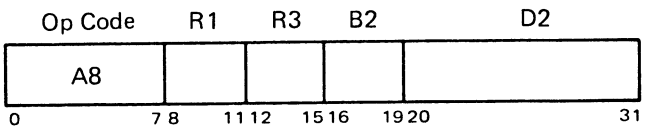
The following bits are selected by the contents of instruction bit positions 8-14 and are either set or reset if instruction bit position 15 is a 1 or a 0, respectively.

Bit	*Meaning*
8	Not used.
9	Transfer trap mode.
10	Not used.
11	Multiple tag mode.
12	Count mode.
13	Not used.
14	7094 trap.
15	Set or reset mode triggers, as designated by instruction bit positions 8-14.

Program Interruption: None
Indicators: Multiple tag mode (light)
Transfer trap mode (light)
7094 trap (light)
Count mode (light)

MVEB9 (Move and Encode Binary)

RS Format



R1 = Starting address of 7094 words
R3 = 7094 word count
C(B2) +D2 = System/370 buffer starting-byte address

Description

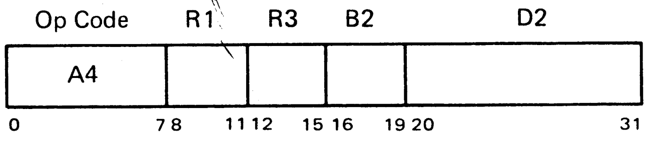
The Move and Encode Binary (Load Binary) instruction takes the 7094 words starting at the System/370 address specified by R1 and loads them into contiguous bytes starting at the byte address specified by C(B2)+D2. In the Model 165, the number of 7094 words that are to be moved and translated is specified by the contents of the GR specified by R3. In the Model 165 II/168, only bits 25-31 of the GR specified by R3 are used for the word count. Each 7094 word is considered as six 6-bit bytes. Each 7094 byte is translated to an EBCDIC 8-bit byte. (See Table 3 in Appendix B.) A 7094 word count of 0 transfers no characters.

The results of this instruction are unpredictable if the source operand field overlaps the destination operand field and the word count is greater than 1.

Program Interruption: Access
Indicators: None

MVED9 (Move and Encode Decimal)

RS Format



R1 = Starting address of 7094 words
R3 = 7094 word count
C(B2) +D2 = System/370 buffer starting-byte address

Description

The Move and Encode Decimal (Load Decimal) instruction is identical with the MVEB9 (Load Binary) instruction except for the translation (see Table 3, Appendix B).

Program Interruption: Access

Indicators: None

MVDB9 (Move and Decode Binary)

RS Format

Op Code	R1	R3	B2	D2
B0				
0 7	8 11	12 15	16 19	20 31

R1 = Starting address of 7094 words
R3 = 7094 word count
C(B2) +D2 = System/370 buffer starting-byte address

Description

The Move and Decode Binary instruction takes contiguous 8-bit bytes, starting at the System/370 byte address specified by C(B2)+D2, and loads them into 7094 words starting at the System/370 address specified by the GR designated by R1. In the Model 165, the number of 7094 words to be formed is specified by the GR designated by R3. In the Model 165 II/168, only bits 25-31 of the GR specified by R3 are used for the word count. Each group of six contiguous 8-bit bytes are translated (as shown in Table 3) into six 6-bit bytes and stored at the appropriate System/370 doubleword address according to the 7094 storage format (Figure 1). A 7094 word count of 0 transfers no characters.

Program Interruption: Access

Indicators: None

MVDD9 (Move and Decode Decimal)

RS Format

Op Code	R1	R3	B2	D2
A3				
0 7	8 11	12 15	16 19	20 31

R1 = Starting address of 7094 words
R3 = 7094 word count
C(B2) +D2 = System/370 buffer starting-byte address

Description

The Move and Decode Decimal (Unload Decimal) instruction is identical with the MVDB9 (Unload Binary) instruction except for the translation (see Table 3 in Appendix B).

Program Interruption: Access

Indicators: None

ISIC9 (Insert-Set Instruction Counter)

RS Format

Op Code	R1	R3	B2	D2
A5				
0 7	8 11	12 15	16 19	20 31

R1 = Any GR
R3 = 0 = IIC9 (Insert 94IC into the GR designated by R1)
  1 = IICM9 (Insert 94IC minus 1 into GR designated by R1)
  2 = SIC9 (Set contents of GR designated by R1 into 94IC)
  3 = LAF9 (Load Address Fix)
C(B2) +D2 = Unimportant (but must be valid)

Description

IIC9	The contents of the 94IC are placed into bits 8-28 of the GR specified by R1. All remaining GR bits are set to 0. The 94IC remains unchanged.
IICM9	The contents of the 94IC minus 1 are placed into bits 8-28 of the GR specified by R1. All remaining GR bits are set to 0. The 94IC remains unchanged.
SIC9	The contents of bits 8-28 of the GR specified by R1 replace the contents of the 94IC, and the contents of that address are placed into the 94IR.
LAF9	The contents of the GR specified by R1 are reduced by 8. If bits 0-7 of the GR specified by R1 are all 1's, the instruction is a No Operation.

Program Interruption: None for IIC9, IICM9, and LAF9
Access for SIC9

Indicators: None

SKAC9 (Skip on Accumulator)

RS Format

Op Code	R1	R3	B2	D2
BF	4			
0 7	8 11	12 15	16 19	20 31

R3 = 0 = CAS (Compare Accumulator with Storage)
  1 = PBT (P-bit test)
C(B2) +D2 = Effective address

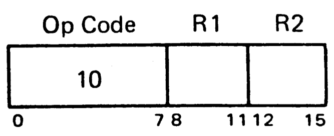
Description

This instruction is equivalent to the instruction specified by R3.

Program Interruption: Access
Indicators: None

PXD9 (Place Index in Decrement)

RR Format



R1 = Source GR (normally an index register)
R2 = Unimportant

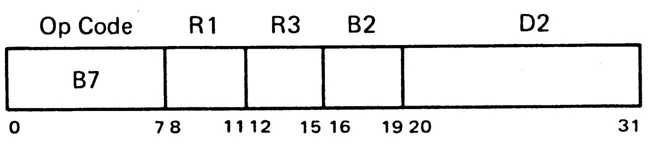
Description

This instruction resets ACR(GR 5) to 0, resets the AC sign to 0, and places the contents of the source GR specified by R1 into ACL(GR 4).

Program Interruption: None
Indicators: None

TNX9 (Transfer No Index)

RS Format



R1 = Decrement GR (normally 8)
R3 = Index register affected

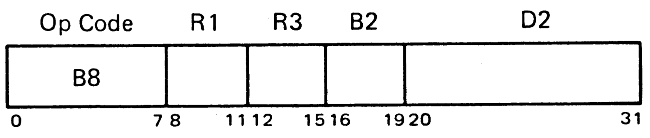
Description

If the contents of the GR specified by R3 are greater than the decrement in the GR specified by R1, the number in the GR specified by R3 is reduced by the decrement, and the next sequential System/370 instruction is executed. If the contents of the GR specified by R3 are less than or equal to the decrement, the contents of the R3-specified register are unchanged, the address specified by C(B2)+D2 is placed in the 94IC, and the contents of that address are placed in the 94IR. The next sequential System/370 instruction is then executed.

Program Interruption: Access
Indicators: CM-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

BAC9 (Branch on Accumulator)

RS Format



R1 = Any even-numbered GR (normally 4)
R3 = 0 = TNZ (Transfer on No Zero)
1 = TZE (Transfer on Zero)

Description

If R3=0 and the contents of the *even-odd pair* of GR's designated by R1 (bit positions 4-59) are nonzero, or if R3=1 and the contents of the pair of GR's designated by R1 (bit positions 4-59) are 0, then the address specified by C(B2)+D2 is placed in the 94IC and the contents of that address are placed in the 94IR. The next sequential System/370 instruction is then executed. If the above condition is not met, the 94IC and 94IR are left unchanged, and the next sequential System/370 instruction is executed.

Program Interruption: Access
Indicators: CM-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

TLQ9 (Transfer Low Quotient)

RR Format

Op Code	R1	R2
0F		4
0 7	8 11	12 15

R1 = Effective address (normally 3)

Description

If the MQ is algebraically less than the accumulator (including S, Q, P, and 1-35), the address specified by the GR designated by R1 is placed in the 94IC and the contents of that address in the 94IR. The computer then takes the next sequential System/370 instruction. If the MQ is algebraically greater than or equal to the AC (including S, Q, P, and 1-35), the next System/370 instruction is executed.

Program Interruption: Access

Indicators: CM-CNM trap flag
94 trap (light)
(See "Transfer Trap Mode (TTM)")

DLD9 (Double Load)

RS Format

Op Code	R1	R3	B2	D2
B6				
0 7	8 11	12 15	16 19	20 31

R1 = Unimportant
R3 = Unimportant

Description

This instruction is equivalent to the DLD instruction. The address of the C(Y) is specified by the second operand address, C(B2)+D2.

Note: If the address of the C(Y) is odd and the system emulated is not the 7094 II, the C(Y) is placed in the AC and the MQ. If trap mode is indicated, a trap occurs.

Program Interruption: Access

Indicators: Double-precision specification

PDX9 (Place Decrement in Index)

RR Format

Op Code	R1	R2
0C		
0 7	8 11	12 15

R1 = Source GR, normally 4 (AC)
R2 = Destination index GR

Description

This instruction is equivalent to PDX (Place Decrement in Index) when R1 = 4.

Program Interruption: None

Indicators: None

PDC9 (Place Complement of Decrement in Index)

RR Format

Op Code	R1	R2
0D		
0 7	8 11	12 15

R1 = Source GR, normally 4 (AC)
R2 = Destination index GR

Description

This instruction is equivalent to PDC (Place Complement of Decrement in Index) when R1 = 4.

Program Interruption: None

Indicators: None

TRANSFER TRAP MODE (TTM)

The following applies to all secondary emulator transfer instructions. When the emulator is in transfer trap mode (TTM), the same functions occur as when not in TTM. These functions include placing the effective address in the 94IC (if the condition is met) and reloading the 94IR. However, three additional events occur in TTM:

1. The 94 trap trigger is set so that DIL transfers to the trap subroutine.
2. If the condition is not met (CNM), hexadecimal 3F is placed in FPR 6, byte 4. The remaining bytes of FPR 6 are unaltered.
3. If the condition is met (CM), hexadecimal FF is placed in FPR 6 byte 4; the 94IC of the transfer instruction plus 1 is placed in bit positions 40-60 of FPR 6, and 0's are placed in bit positions 61-63.

In Model 165 II/168, when high-speed conditions are not met, FPR 6 is changed as described above regardless of TTM. However, the 94 trap trigger is set only in TTM.

When high-speed conditions are met, FPR 6 (bytes 4-7) is not changed.

In the Model 165, when not in TTM, FPR 6 is changed as shown below. (The 94 trap trigger is set only in TTM.)

1. CM: BAC9, TC9, TLQ9, and high-speed TLQ, TMI, TNO, TOV, TPL, TNZ, and TZE; the contents of FPR 6 are the same as in 3 above.
2. CM: BC9, TIX9, TNX9, TXH9, TXL9, TSX9, TXI9, and high-speed TSX; hexadecimal FF is placed in FPR 6 byte 6. No other change is made.
3. CNM: BAC9, BC9, TIX9, TNX9, TXH9, TXL9, and high-speed TNZ and TZE; hexadecimal 3F is placed in FPR 6 byte 4. The remaining bytes remain unchanged.
4. For all other cases, FPR 6 remains unchanged.

Appendix A. Compatibility Feature Indicators

Table 1. Indicators Added by the Compatibility Feature

7094 Indicators			Indicator Pos on Micro-fiche (Mods 165/165 II-168)	Logout (Model 165/Model 165 II-168)	
				Address (Dec)	Bit
			Frame B5		
CONTROL STORE BITS 108-125	(PARITY)	108	B01	P* +872/1056	00
	(EB)	109	B03		02
	(EB)	110	B04		03
	(EB)	111	B05		04
	(EB)	112	B06		05
	(EC)	113	B08		09
	(EC)	114	B09		10
	(EC)	115	B10		11
	(EC)	116	B11		12
	(EC)	117	B12		13
	(ED)	118	B14		17
	(ED)	119	B15		18
	(ED)	120	B16		19
	(EA)	121	B18		21
	(EF)	122	B20		25
	(EG)	123	B22		27
	(EG)	124	B23		28
CONTROL STORE BITS 108-125	(EG)	125	B24		29
94 OP CODE REGISTER		S	C07		40
		1	C08		41
		2	C09		42
		3	C10		43
		4	C11		44
		5	C12		45
		6	C13		48
		7	C14		49
		8	C15		50
		9	C16		51
		10	C17		52
94 OP CODE REGISTER		11	C18	P* +872/1056	53
SIGN TRIGGERS					
ACCUMULATOR (AC)			E01	P* + 880/1064	32
MULTIPLIER QUOTIENT (MQ)			E02		33
STORAGE REGISTER			E03		34
7094 STATUS VALID			E06		37
7040-44 MODE TRIGGER			E07		40
COUNT MODE TRIGGER			E08		41
CONTROL TRIGGERS					
WA & WB TO SERIAL ADDER			E11		44
A4-7 + B4-7 GATE			E12		45
AR1 TO PB42-56 GATE			E13		48
AR1 TO SI(40-60)			E14		49
AR2 TO SI(40-60)			E15	P* + 880/1064	50

Table 1. Indicators Added by the Compatibility Feature (Cont)

7094 Indicators		Indicator Pos on Micro-fiche (Mods 165/165 II-168)	Logout (Model 165/Model 165 II-168) Address (Dec)	Bit
		Frame B5		
HS LAST CYCLE		E16	P* + 880/1064	51
IND ADDR		E17	↑	52
EMIT TO LSAL	0	E21		58
↑	1	E22		59
↓	2	E23	↓	60
EMIT TO LSAL	3	E24	P* +880/1064	61
BUFFER TRIGGERS				
AC SIGN		F01	P* + 888/1072	00
MQ SIGN		F02	↑	01
HIGH SPEED		F03		02
MODE TRIGGERS				
MULTIPLE TAG		F06		05
TRANSFER TRAP		F07		08
7094 TRAP		F08		09
CONTROL TRIGGERS				
PROTECT INTERRUPT		F11/NA		12
DST ADVANCE (Mod 165 with 94 EMU installed, or Mod 165 II/168)		F12		13
IR VALIDITY DELAYED		F13		16
LAST CYCLE DELAYED		F14		17
LOW-SPEED DIL		F15		18
HIGH-SPEED END OP		F16		19
HIGH-SPEED		F17		20
SELECT WCS		F18		21
EMIT TO LSAL5	0	F21		26
↑	1	F22		27
↓	2	F23		28
EMIT TO LSAL5	3	F24		29
CONTROL TRIGGERS				
IR ADV INTERRUPT		G11/NA		44
IR FETCH INTERRUPT		G12		45
IR FETCH (IRF)		G13		48
INDIRECT IR FETCH		G14		49
ADDRESS STORE COMPARE		G15		50
STORE TYPE		G16		51
PRIVILEGED EXECUTE		G17	↓	52
EMU IN EXEC (Mod 165 with STATX, or Mod 165 II/168)		G18	P* + 888/1072	53
94 INSTRUCTION COUNTER BUFFER	P1	J01	P* + 896/1080	32
↑	P2	J02	↑	33
	P3	J03		34
	8	J04		35
	9	J05		36
	10	J06		37
	11	J07		40
	12	J08		41
	13	J09		42
	14	J10		43
↓	15	J11	↓	44
94 INSTRUCTION COUNTER BUFFER	16	J12	P* + 896/1080	45

Table 1. Indicators Added by the Compatibility Feature (Cont)

7094 Indicators		Indicator Pos on Micro-fiche (Mods 165/165 II-168)	Logout (Model 165/Model 165 II-168) Address (Dec)	Bit
94 INSTRUCTION COUNTER BUFFER		Frame B5		
↑	17	J13	P* + 896/1080	48
	18	J14	↑	49
	19	J15		50
	20	J16		51
	21	J17		52
	22	J18		53
	23	J19		56
	24	J20		57
	25	J21		58
	26	J22		59
↓	27	J23	↓	60
94 INSTRUCTION COUNTER BUFFER	28	J24	P* + 896/1080	61
TRANSLATOR INPUT (Parity Error)		K11	P* + 904/1088	12
CS 108-125 (Parity Error)		K12	P* + 904/1088	13
		Frame B6		
94IR	P1-8	B01	P* + 912/1096	00
↑	P9-16	B02	↑	01
	PREFIX S	B07		08
	PREFIX 1	B08		09
	PREFIX 2	B09		10
	3	B10		11
	4	B11		12
	5	B12		13
	6	B13		16
	7	B14		17
	8	B15		18
	9	B16		19
	10	B17		20
	11	B18		21
	12	B19		24
	13	B20		25
	14	B21		26
	15	B22		27
	16	B23		28
	17	B24		29
	P17-24	C01		32
	P25-32	C02		33
	P33-35	C03		34
	TAG 18	C07		40
	TAG 19	C08		41
	TAG 20	C09		42
	21	C10		43
	22	C11		44
	23	C12		45
	24	C13		48
	25	C14		49
	26	C15		50
	27	C16		51
	28	C17		52
	29	C18		53
	30	C19		56
	31	C20		57
	32	C21		58
	33	C22		59
↓	34	C23	↓	60
94IR	35	C24	P* + 912/1096	61

Table 1. Indicators Added by the Compatibility Feature (Cont)

7094 Indicators		Indicator Pos on Micro-fiche (Mods 165/165 II-168)	Logout (Model 165/Model 165 II-168)	
			Address (Dec)	Bit
		Frame B6		
94IC	P1	E01	P* + 920/1104	32
↑	P2	E02	↑	33
	P3	E03		34
	8	E04		35
	9	E05		36
	10	E06		37
	11	E07		40
	12	E08		41
	13	E09		42
	14	E10		43
	15	E11		44
	16	E12		45
	17	E13		48
	18	E14		49
	19	E15		50
	20	E16		51
	21	E17		52
	22	E18		53
	23	E19		56
	24	E20		57
	25	E21		58
	26	E22		59
↓	27	E23	↓	60
94IC	28	E24	P* + 920/1104	61
AR2	INVERT A	G01	P* + 928/1112	32
AR2	INVERT B	G02	↑	33
AR1	INVERT A	G07		40
↑	P2	G08		41
	P3	G09		42
	14	G10		43
	15	G11		44
	16	G12		45
	17	G13		48
	18	G14		49
	19	G15		50
	20	G16		51
	21	G17		52
	22	G18		53
	23	G19		56
	24	G20		57
	25	G21		58
	26	G22		59
↓	27	G23	↓	60
AR1	28	G24	P* + 928/1112	61
AR2	P1	H01	P* + 936/1120	00
↑	P2	H02	↑	01
	P3	H03		02
	8	H04		03
	9	H05		04
	10	H06		05
	11	H07		08
	12	H08		09
↓	13	H09	↓	10
AR2	14	H10	P* + 936/1120	11

Table 1. Indicators Added by the Compatibility Feature (Cont)

7094 Indicators		Indicator Pos on Micro-fiche (Mods 165/165 II-168)	Logout (Model 165/Model 165 II-168) Address (Dec)	Bit
		Frame B6		
AR2	15	H11	P* + 936/1120	12
	16	H12		13
	17	H13		16
	18	H14		17
	19	H15		18
	20	H16		19
	21	H17		20
	22	H18		21
	23	H19		24
	24	H20		25
	25	H21		26
	26	H22		27
	27	H23		28
AR2	28	H24	P* + 936/1120	29
		Frame A6 Version 094		
INSTRUCTION Q1 STATUS BIT TRIGGERS				
EMU TYPE Q1		E16/D6	P* + 640/784	52/05
PRIV DIL Q1		E17/D18	P* + 640/784	53/21
INSTRUCTION Q2 STATUS BIT TRIGGERS				
EMU TYPE Q2		F16/E6	P* + 648/792	20/37
PRIV DIL Q2		F17/E18	P* + 648/792	21/53
INSTRUCTION Q3 STATUS BIT TRIGGERS				
EMU TYPE Q3		G16/F6	P* + 648/792	52/05
PRIV DIL Q3		G17/F18	P* + 648/792	53/21
INSTRUCTION Q4 STATUS BIT TRIGGERS				
EMU TYPE Q4		NA/G6	P* + NA/792	NA/37
PRIV DIL Q4		NA/G18	P* + NA/792	NA/53
E-UNIT EMULATOR TYPE		NA/H6	P* + NA/800	NA/05
DIL9 TO CONTROL STORAGE ADDRESS REGISTER (CSAR) TRIGGER		H01/NA	P* + 656/NA	00/NA
EMU I TIME		H02/NA	P* + 656/NA	01/NA

P* = Contents of control register 15 (bits 8-31) which is logged out at decimal address 508.

The character code for System/370 (EBCDIC) differs from the codes used in previous IBM systems. These differences and a summary of 7094 character codes are presented in Tables 2, 3, and 4.

Table 2 is divided into two parts. The leftmost column of the first part shows the BCD card code. The second and third columns show the FORTRAN and commercial characters, respectively, for the card code. The next two columns show the EBCDIC character and hexadecimal equivalent for the card code. The second part shows the EBCDIC card code for a character identical with the H-set character. The character and its hexadecimal equivalent are shown in the last two columns.

Table 3 translates 7094 bit configurations to EBCDIC bit configurations.

Table 4 summarizes the conversions of all characters in the H set. The leftmost column shows the form of the character (column 3) in 7094 core storage; representation is in octal notation. The second column shows the magnetic-tape code for the character as it appears after the 7094 writes it in even parity (decimal) on seven-track tape. Card codes that correspond to the 7094 core representations are shown in column 4. Columns 5 and 6 show the codes for the character after it has been written on nine-track tape. Nine-track tapes are always written in odd parity; however, the characters reflect the parity in which the seven-track tape would have been written. If the seven-track tape has been written in odd parity (binary), bit 1 of each nine-track tape character is a 0; if the seven-track tape has been written in even parity, bit 1 of each nine-track tape character is a 1. The seventh column shows the character printed by a System/370 printer for that 7094-core bit configuration.

The last two columns demonstrate the effect of the dual conversion option. As indicated by the column headings, the option can be used only for even-parity (decimal) operations. Where a single figure is given, the effect is the same with or without the option. Codes within parentheses represent input codes to be translated to corresponding 7094 core-storage codes; the resultant 7094 core-storage code is the same for both input codes. The code outside the parentheses is the code of the output character. An asterisk indicates that no tape character translates to the corresponding 7094 core-storage character.

Example: An H-set equal sign or A-set pound sign (#) (BCD card code 8-3) resides on seven-track, even-parity tape as an octal 13. When this code enters simulated 7094 core storage it remains an octal 13. On even-parity output to a seven-track tape, the dual conversion option changes the code to an octal 16. A System/370 printer interprets the code as an equal sign.

An EBCDIC equal sign (card code 8-6) resides on seven-track, even-parity tape as an octal 16. This character becomes an octal 13 in simulated 7094 core storage. On even-parity output to a seven-track tape, the dual conversion option leaves the code unchanged. A System/370 printer, therefore, writes this code as an equal sign.

Table 2. Dual Characters

Dual Characters					EBCDIC Equivalent to an H-Set Character		
26 Card Punch Code	FORTRAN or H Set	Commercial or A Set	EBCDIC		29 Card Punch Code	EBCDIC	
			Character	Hex Equivalent		Character	Hex Equivalent
8, 3	=	#	#	7B	8, 6	=	7E
8, 4	'			7C	8, 5	'	7D
12	+	&	&	50	12, 8, 6	+	4E
12,8,4	)	¤	<	4C	11, 8, 5	)	5D
0, 8, 4	(	%	%	6C	12, 8, 5	(	4D

Table 3. Translation, 7094 Bit Configurations to EBCDIC

7094 Image		Binary		Decimal BCD	
00	0000	0x00	0000	1111	0000
00	0001	1x11	0001	1111	0001
⋮	⋮	⋮	⋮	⋮	⋮
00	1001	1x11	1001	1111	1001
00	0000*				
00	1010	1x11	0000	1111	0000
00	1011	0x11	1011	0111	1011
⋮	⋮	⋮	⋮	⋮	⋮
00	1111	0x11	1111	0111	1111
01	0000	0x11	1010	0101	0000
01	0001	0x10	0001	1100	0001
01	0010	1x10	0010	1100	0010
⋮	⋮	⋮	⋮	⋮	⋮
01	1001	1x10	1001	1100	1001
01	1010	1x10	0000	1100	0000
01	1011	0x10	1011	0100	1011
⋮	⋮	⋮	⋮	⋮	⋮
01	1111	0x10	1111	0100	1111
10	0000	0x10	0000	0110	0000
10	0001	1x01	0001	1101	0001
⋮	⋮	⋮	⋮	⋮	⋮
10	1001	1x01	1001	1101	1001
10	1010	1x01	0000	1101	0000
10	1011	0x01	1011	0101	1011
⋮	⋮	⋮	⋮	⋮	⋮
10	1111	0x01	1111	0101	1111
11	0000	0x01	0000	0100	0000
11	0001	0x01	0001	0110	0001
11	0010	1x00	0010	1110	0010
⋮	⋮	⋮	⋮	⋮	⋮
11	1001	1x00	1001	1110	1001
11	1010	1x00	0000	1110	0000
11	1011	0x00	1011	0110	1011
⋮	⋮	⋮	⋮	⋮	⋮
11	1111	0x00	1111	0110	1111

* From EBCDIC to 7094 storage (BCD only)

x = one, when buffer is loaded from seven-track tape.
= zero, when buffer is loaded from nine-track tape or from 7094 storage.

Table 4. Character and Code Correspondence

7094 Core (Octal)	7-Track Decimal (Octal)	H-Set (FORTRAN) Character	26 Card Code	9-Track (Hex) Binary	9-Track (Hex) Decimal	EBCDIC Character	Dual Conversion Option 7-Track Decimal (Octal)	Dual Conversion Option 9-Track Decimal (Hex)
00	12	0	0	00	F0	0	12	F0
01	01	1	1	B1	F1	1	01	F1
02	02	2	2	B2	F2	2	02	F2
03	03	3	3	B3	F3	3	03	F3
04	04	4	4	B4	F4	4	04	F4
05	05	5	5	B5	F5	5	05	F5
06	06	6	6	B6	F6	6	06	F6
07	07	7	7	B7	F7	7	07	F7
10	10	8	8	B8	F8	8	10	F8
11	11	9	9	B9	F9	9	11	F9
12 (See Note)	12	*, 0		B0	F0	0	12	F0
13	13	=	8, 3	3B	7B	#	(16, 13) 16	(7E, 7B), 7E

Table 4. Character and Code Correspondence (continued)

7094 Core (Octal)	7-Track Decimal (Octal)	H-Set (FORTRAN) Character	26 Card Code	9 Track (Hex) Binary	9 Track (Hex) Decimal	EBCDIC Character	Dual Conversion Option 7-Track Decimal (Octal)	Dual Conversion Option 9-Track Decimal (Hex)
14	14	'	8, 4	3C	7C	@	(15, 14), 15	(7D, 7C), 7D
15	15			3D	7D	'	*, 14	*, 7C
16	16			3E	7E	=	*, 13	*, 7B
17	17			3F	7F	''	17	7F
20	60	+	12	3A	50	&	(76, 60), 76	(4E, 50), 4E
21	61	A	12, 1	21	C1	A	61	C1
22	62	B	12, 2	A2	C2	B	62	C2
23	63	C	12, 3	A3	C3	C	63	C3
24	64	D	12, 4	A4	C4	D	64	C4
25	65	E	12, 5	A5	C5	E	65	C5
26	66	F	12, 6	A6	C6	F	66	C6
27	67	G	12, 7	A7	C7	G	67	C7
30	70	H	12, 8	A8	C8	H	70	C8
31	71	I	12, 9	A9	C9	I	71	C9
32	72			A0	C0		72	C0
33	73	.	12, 8, 3	2B	4B	.	73	4B
34	74	)	12, 8, 4	2C	4C		(55, 74), 55	(5D, 4C), 5D
35	75			2D	4D	(	*, 34	*, 6C
36	76			2E	4E	+	*, 60	*, 50
37	77			2F	4F	|	77	4F
40	40	-	11	20	60	- (hyphen)	40	60
41	41	J	11, 1	91	D1	J	41	D1
42	42	K	11, 2	92	D2	K	42	D2
43	43	L	11, 3	93	D3	L	43	D3
44	44	M	11, 4	94	D4	M	44	D4
45	45	N	11, 5	95	D5	N	45	D5
46	46	O	11, 6	96	D6	O	46	D6
47	47	P	11, 7	97	D7	P	47	D7
50	50	Q	11, 8	98	D8	Q	50	D8
51	51	R	11, 9	99	D9	R	51	D9
52	52			90	D0		52	D0
53	53	$	11, 8, 3	1B	5B	+	53	5B
54	54	*	11, 8, 4	1C	5C	*	54	5C
55	55			1D	5D	)	*, 74	*, 4C
56	56			1E	5E	;	56	5E
57	57			1F	5F	(not)	57	5F
60	20	blank	blank	10	40	blank	20	40
61	21	/	0, 1	81	61	/	21	61
62	22	S	0, 2	82	E2	S	22	E2
63	23	T	0, 3	83	E3	T	23	E3
64	24	U	0, 4	84	E4	U	24	E4
65	25	V	0, 5	85	E5	V	25	E5
66	26	W	0, 6	86	E6	W	26	E6
67	27	X	0, 7	87	E7	X	27	E7
70	30	Y	0, 8	88	E8	Y	30	E8
71	31	Z	0, 9	89	E9	Z	31	E9
72	32			80	E0		32	E0
73	33	,	0, 8, 3	0B	6B	,	33	6B
74	34	(	0, 8, 4	0C	6C	%	(75, 34), 75	(4D, 6C), 4D
75	35			0D	6D	_ (underscore)	35	6D
76	36			0E	6E	>	36	6E
77	37			0F	6F	?	37	6F

Note: A 7094 octal 12 is written on seven-track tape as an octal 12 and on nine-track tape as a hexadecimal F0. Both print as 0 by the 7094. A seven-track octal 12, however, and a nine-track hexadecimal F0 appear as an octal 00 to the 7094. As output, both the octal 0 and octal 12 become an octal 12.

Appendix C. Numeric Index of Instructions

Operation Code	Mnemonic	Instruction	Page
0A	PAX9	Place Address in Index	14
0C	PDC9	Place Complement of Decrement in Index	23
0D	PDX9	Place Decrement in Index	23
0E	SKC9	Skip on Condition	19
0F	TLQ9	Transfer Low Quotient	23
10	PXD9	Place Index in Decrement	22
11	SHFT9	Shift	13
12	BA9	Branch on Address	19
13	AXT9	Address to Index True	14
14	SM9	Set Mode	20
15	D9	Divide	17
16	ACL9	Add and Carry Logical	13
17	SQP9	Set Q, P	12
25	DIL9	Do Interpretive Loop	10
A0	STO9	Store	12
A1	STQ9	Store MQ	12
A2	SLW9	Store Logical Word	12
A3	MVDD9	Move and Decode Decimal	21
A4	MVED9	Move and Encode Decimal	20
A5	ISIC9	Insert – Set Instruction Counter	21
A6	BC9	Branch on Condition	19
A7	TXI9	Transfer with Index Incremented	15
A8	MVEB9	Move and Encode Binary	20
A9	FM9	Floating-Point Multiply	17
AA	DFAS9	Double Precision Floating-Point Add-Subtract	18
AB	FAS9	Floating-Point Add-Subtract	17
AC	DFM9	Double Precision Floating-Point Multiply	18
AD	DFD9	Double Precision Floating-Point Divide	18
AE	FD9	Floating-Point Divide	18
AF	M9	Multiply	17
B0	MVDB9	Move and Decode Binary	21
B1	TSX9	Transfer and Set Index	16
B3	LD9	Load	13
B5	AS9	Add, Subtract	13
B6	DLD9	Double Load	23
B7	TNX9	Transfer on No Index	22
B8	BAC9	Branch on Accumulator	22
B9	TC9	Transfer Condition	16
BA	ST9	Store Address or Index	14
BB	XEC9	Execute	14
BC	TXL9	Transfer on Index Low or Equal	15
BD	TXH9	Transfer on Index High	16
BE	TIX9	Transfer on Index	15
BF	SKAC9	Skip on Accumulator Addresses	21
E9	EMU	Emulator Feature	10

The charts in this appendix contain the System/370 hexadecimal subroutine base addresses for all possible combinations of 7094 operation codes. In the cases where the 7094 operation code uses the index register as an operand, there are eight subroutine base addresses for that operation code corresponding to the tag field of positions 0-7. The charts are in two groups: group I comprises those operation codes not using an index register as an operand; group II, those operation codes that do use an index register as an operand.

To locate a subroutine address in main storage, add the 94 core relocate factor (FPR 2 bits 36-56) to the appropriate System/370 address.

GROUP I: OPERATIONS NOT USING AN INDEX REGISTER AS AN OPERAND

7094 Op Code		System/370 Address (Hex)
+0000	(HTR)	00000
−0000		02000
+0001		01110
−0001		03110
+0002		01120
−0002		03120
+0003		01130
−0003		03130
+0004		01140
−0004		03140
+0005		01150
−0005		03150
+0006		01160
−0006		03160
+0007		01170
−0007		03170
+0010		01180
−0010		03180
+0011		01190
−0011		03190
+0012		011A0
−0012		031A0
+0013		011B0
−0013		031B0
+0014		011C0
−0014		031C0
+0015		011D0
−0015		031D0
+0016		011E0
−0016		031E0
+0017		011F0
−0017		031F0
+0020	(TRA)	00100
−0020		02100
+0021	(TTR)	00110
−0021	(ESNT)	02110
+0022	(TRCA)	00120
−0022	(TRCB)	02120
+0023		00130
−0023		02130
+0024	(TRCC)	00140
−0024	(TRCD)	02140
+0025		00150
−0025		02150
+0026	(TRCE)	00160
−0026	(TRCF)	02160
+0027	(TRCG)	00170
−0027	(TRCH)	02170
+0030	(TEFA)	00180
−0030	(TEFB)	02180
+0031	(TEFC)	00190
−0031	(TEFD)	02190
+0032	(TEFE)	001A0
−0032	(TEFF)	021A0
+0033	(TEFG)	001B0
−0033	(TEFH)	021B0
+0034		001C0
−0034		021C0
+0035		001D0
−0035		021D0
+0036		001E0
−0036		021E0
+0037		001F0
−0037		021F0
+0040	(TLQ)	00200
−0040		02200
+0041	(IIA)	00210
−0041		02210
+0042	(TIO)	00220
−0042	(RIA)	02220
+0043	(OIA)	00230
−0043		02230
+0044	(PAI)	00240
−0044		02240

7094 Op Code	System/370 Address (Hex)
+0045	00250
−0045	02250
+0046 (TIF)	00260
−0046 (PIA)	02260
+0047	00270
−0047	02270
+0050	00280
−0050	02280
+0051 (IIR)	00290
−0051 (IIL)	02290
+0052	002A0
−0052	022A0
+0053	002B0
−0053	022B0
+0054 (RFT)	002C0
−0054 (LFT)	022C0
+0055 (SIR)	002D0
−0055 (SIL)	022D0
+0056 (RNT)	002E0
−0056 (LNT)	022E0
+0057 (RIR)	002F0
−0057 (RIL)	022F0
+0060 (TCOA)	00300
−0060 (TCNA)	02300
+0061 (TCOB)	00310
−0061 (TCNB)	02310
+0062 (TCOC)	00320
−0062 (TCNC)	02320
+0063 (TCOD)	00330
−0063 (TCND)	02330
+0064 (TCOE)	00340
−0064 (TCNE)	02340
+0065 (TCOF)	00350
−0065 (TCNF)	02350
+0066 (TCOG)	00360
+0066 (TCNG)	02360
+0067 (TCOH)	00370
−0067 (TCNH)	02370
+0070	00380
−0070	02380
+0071	00390
−0071	02390
+0072	003A0
−0072	023A0
+0073	003B0
−0073	023B0
−0074	03380
+0075	003D0
−0075	023D0
+0076	003E0
−0076	023E0
+0077	003F0
−0077	023F0

7094 Op Code	System/370 Address (Hex)
+0100 (TZE)	00400
−0100 (TNZ)	02400
+0101	00410
−0101	02410
+0102	00420
−0102	02420
+0103	00430
−0103	02430
+0104	00440
−0104	02440
+0105	00450
−0105	02450
+0106	00460
−0106	02460
+0107	00470
−0107	02470
+0110	00480
−0110	02480
+0111	00490
−0111	02490
+0112	004A0
−0012	024A0
+0113	004B0
−0113	024B0
+0114 (CVR)	004C0
−0114 (CAQ)	024C0
+0115	004D0
−0115	024D0
+0116	004E0
−0116	024E0
+0117	004F0
−0117	024F0
+0120 (TPL)	00500
−0120 (TMI)	02500
+0121	00510
−0121	02510
+0122	00520
−0122	02520
+0123	00530
−0123	02530
+0124	00540
−0124	02540
+0125	00550
−0125	02550
+0126	00560
−0126	02560
+0127	00570
−0127	02570
+0130	00580
−0130 (XCL)	02580
+0131 (XCA)	00590
−0131	02590
+0132	005A0
−0032	025A0

7094 Op Code	System/370 Address (Hex)
+0133	005B0
−0133	025B0
+0134	005C0
−0134	025C0
+0135	005D0
−0135	025D0
+0136	005E0
−0136	025E0
+0137	005F0
−0137	025F0
+0140 (TOV)	00600
−0140 (TNO)	02600
+0141	00610
−0141	02610
+0142	00620
−0142	02620
+0143	00630
−0143	02630
+0144	00640
−0144	02640
+0145	00650
−0145	02650
+0146	00660
−0146	02660
+0147	00670
−0147	02670
+0150	00680
−0150	02680
+0151	00690
−0151	02690
+0152	006A0
−0152	026A0
+0153	006B0
−0153	026B0
+0154	006C0
−0154 (CRQ)	026C0
+0155	006D0
−0155	026D0
+0156	006E0
−0156	026E0
+0157	006F0
−0157	026F0
+0160	00700
−0160	02700
+0161 (TQO)	00710
−0161	02710
+0162 (TQP)	00720
−0162	02720
+0163	00730
−0163	02730
+0164	00740
−0164	02740
+0165	00750
−0165	02750

7094 Op Code	System/370 Address (Hex)
+0166	00760
−0166	02760
+0167	00770
−0167	02770
+0170	00B80
−0170	02B80
+0171	00B90
−0171	02B90
+0172	00BA0
−0172	02BA0
−0173	00BB0
−0173	02BB0
+0174	00BC0
−0174	02BC0
+0175	00BD0
−0175	02BD0
+0176	00BE0
−0176	02BE0
+0177	00BF0
−0177	02BF0
+0200 (MPY)	00800
−0200 (MPR)	02800
+0201	00810
−0201	02810
+0202	00820
−0202	02820
+0203	00830
−0203	02830
+0204 (VLM)	00840
−0204	02840
+0205	00850
−0205	02850
+0206	00860
−0206	02860
+0207	00870
−0207	02870
+0210	00880
−0210	02880
+0211	00890
−0211	02890
+0212	008A0
−0212	028A0
+0213	008B0
−0213	028B0
+0214	008C0
−0214	028C0
+0215	008D0
−0215	028D0
+0216	008E0
−0216	028E0
+0217	008F0
−0217	028F0
+0220 (DVH)	00900
−0220	02900

7094 Op Code		System/370 Address (Hex)
+0221	(DVP)	00910
−0221		02910
+0222		00920
−0222		02920
+0223		00930
−0223		02930
+0224	(VDH)	00940
−0224		02940
+0225	(VDP)	00950
−0225		02950
+0226		00960
−0226		02960
+0227		00970
−0227		02970
+0230		00980
−0230		02980
+0231		00990
−0231		02990
+0232		009A0
−0232		029A0
+0233		009B0
−0233		029B0
+0234		009C0
−0234		029C0
+0235		009D0
−0035		029D0
+0236		009E0
−0236		029E0
+0237		009F0
−0237		029F0
+0240	(FDH)	00A00
−0240	(DFDH)	02A00
+0241	(FDP)	00A10
−0241	(DFDP)	02A10
+0242		00A20
−0242		02A20
+0243		00A30
−0243		02A30
+0244		00A40
−0244		02A40
+0245		00A50
−0245		02A50
+0246		00A60
−0246		02A60
+0247		00A70
−0247		02A70
+0250		00A80
−0250		02A80
+0251		00A90
−0251		02A90
+0252		00AA0
−0252		02AA0
+0253		00AB0
−0253		02AB0

7094 Op Code		System/370 Address (Hex)
+0254		00AC0
−0254		02AC0
+0255		00AD0
−0255		02AD0
+0256		00AE0
−0256		02AE0
+0257		00AF0
−0257		02AF0
+0260	(FMP)	00B00
−0260	(UFM)	02B00
+0261	(DFMP)	00B10
−0261	(DUFM)	02B10
+0262		00B20
−0262		02B20
+0263		00B30
−0263		02B30
+0264		00B40
−0264		02B40
+0265		00B50
−0265		02B50
+0266		00B60
−0266		02B60
+0267		00B70
−0267		02B70
+0270		00B80
−0270		02B80
+0271		00B90
−0271		02B90
+0272		00BA0
−0272		02BA0
+0273		00BB0
−0273		02BB0
+0274		00BC0
−0274		02BC0
+0275		00BD0
−0275		02BD0
+0276		00BE0
−0276		02BE0
+0277		00BF0
−0277		02BF0
+0300	(FAD)	00C00
−0300	(UFA)	02C00
+0301	(DFAD)	00C10
−0301	(DUFA)	02C10
+0302	(FSB)	00C20
−0302	(UFS)	02C20
+0303	(DFSB)	00C30
−0303	(DUFS)	02C30
+0304	(FAM)	00C40
−0304	(UAM)	02C40
+0305	(DFAM)	00C50
−0305	(DUAM)	02C50
+0306	(FSM)	00C60
−0306	(USM)	02C60

7094 Op Code	System/370 Address (Hex)
+0307 (DFSM)	00C70
−0307 (DUSM)	02C70
+0310	00880
−0310	02880
+0311	00890
−0311	02890
+0312	008A0
−0312	028A0
+0313	008B0
−0313	028B0
+0314	008C0
−0314	028C0
+0315	008D0
−0315	028D0
+0316	008E0
−0316	028E0
+0317	008F0
−0317	028F0
+0320 (ANS)	00D00
−0320 (ANA)	02D00
+0321	00D10
−0321	02D10
+0322 (ERA)	00D20
−0322	02D20
+0323	00D30
−0323	02D30
+0324	00D40
−0324	02D40
+0325	00D50
−0325	02D50
+0326	00D60
−0326	02D60
+0327	00D70
−0327	02D70
+0330	00980
+0330	02980
+0331	00990
−0331	02990
+0332	009A0
−0332	029A0
+0333	009B0
−0333	029B0
+0334	009C0
−0334	029C0
+0335	009D0
−0335	029D0
+0336	009E0
−0336	029E0
+0337	009F0
−0337	029F0
+0340 (CAS)	00E00
−0340 (LAS)	02E00
+0341	00E10
−0341	02E10

7094 Op Code	System/370 Address (Hex)
+0342	00E20
−0342	02E20
+0343	00E30
−0343	02E30
+0344	00E40
−0344	02E40
+0345	00E50
−0345	02E50
+0346	00E60
−0346	02E60
+0347	00E70
−0347	02E70
+0350	00A80
−0350	02A80
+0351	00A90
−0351	02A90
+0352	00AA0
−0352	02AA0
+0353	00AB0
−0353	02AB0
+0354	00AC0
−0354	02AC0
+0355	00AD0
−0355	02AD0
+0356	00AE0
−0356	02AE0
+0357	00AF0
−0357	02AF0
+0360	00F00
−0360	02F00
+0361 (ACL)	00F10
−0361	02F10
+0362	00F20
−0362	02F20
+0363	00F30
−0363	02F30
+0364	00F40
−0364	02F40
+0365	00F50
−0365	02F50
+0366	00F60
−0366	02F60
+0367	00F70
−0367	02F70
+0370	00B80
−0370	02B80
+0371	00B90
−0371	00B90
+0372	00BA0
−0372	02BA0
+0373	00BB0
−0373	02BB0
+0374	00BC0
−0374	02BC0

7094 Op Code	System/370 Address (Hex)
+0375	00BD0
−0375	02BD0
+0376	00BE0
−0376	02BE0
+0377	00BF0
−0377	02BF0
+0400 (ADD)	01000
−0400 (SBM)	03000
+0401 (ADM)	01010
−0401	03010
+0402 (SUB)	01020
−0402	03020
+0403	01030
−0403	03030
+0404	01040
−0404	03040
+0405	01050
−0405	03050
+0406	01060
−0406	03060
+0407	01070
−0407	03070
+0410	01080
−0410	03080
+0411	01090
−0411	03090
+0412	010A0
−0412	030A0
+0413	010B0
−0413	030B0
+0414	010C0
−0414	030C0
+0415	010D0
−0415	030D0
+0416	010E0
−0416	030E0
+0417	010F0
−0417	030F0
+0420 (HPR)	01100
−0420	03100
+0421	01110
−0421	03110
+0422	01120
−0422	03120
+0423	01130
−0423	03130
+0424	01140
−0424	03140
+0425	01150
−0425	03150
+0426	01160
−0426	03160
+0427	01170
−0427	03170

7094 Op Code	System/370 Address (Hex)
+0430	01180
−0430	03180
+0431	01190
−0431	03190
+0432	011A0
−0432	031A0
+0433	011B0
−0433	031B0
+0434	011C0
−0434	031C0
+0435	011D0
−0435	031D0
+0436	011E0
−0436	031E0
+0437	011F0
−0437	031F0
+0440 (IIS)	01200
−0440	03200
+0441 (LDI)	01210
−0441	03210
+0442 (OSI)	01220
−0442	03220
+0443 (DLD)	01230
−0443	03230
+0444 (OFT)	01240
−0444	03240
+0445 (RIS)	01250
−0445	03250
+0446 (ONT)	01260
−0446	03260
+0447	01270
−0447	03270
+0450	01280
−0450	03280
+0451	01290
−0451	03290
+0452	012A0
−0452	032A0
+0453	012B0
−0453	032B0
+0454	012C0
−0454	032C0
+0455	012D0
−0455	032D0
+0456	012E0
−0456	032E0
+0457	012F0
−0457	032F0
+0460	01300
−0460	03300
+0461	01310
−0461	03310
+0462	01320
−0462	03320

7094 Op Code	System/370 Address (Hex)
+0463	01330
−0463	03330
+0464	01340
−0464	03340
+0465	01350
−0465	03350
+0466	01360
−0466	03360
+0467	01370
−0467	03370
+0470	01180
−0470	03180
+0471	01190
−0471	03190
+0472	011A0
−0472	031A0
+0473	011B0
−0473	031B0
+0474	011C0
−0474	031C0
+0475	011D0
−0475	031D0
+0476	011E0
−0476	031E0
+0477	011F0
−0477	031F0
+0500 (CLA)	01400
−0500 (CAL)	03400
+0501	01410
−0501 (ORA)	03410
+0502 (CLS)	01420
−0502	03420
+0503	01430
−0503	03430
+0504	01440
−0504	03440
+0505	01450
−0505	03450
+0506	01460
−0506	03460
+0507	01470
−0507	03470
+0510	00880
−0510	02880
+0511	00890
−0511	02890
+0512	008A0
−0512	028A0
+0513	008B0
−0513	028B0
+0514	008C0
−0514	028C0
+0515	008D0
−0515	028D0

7094 Op Code	System/370 Address (Hex)
+0516	008E0
−0516	028E0
+0517	008F0
−0517	028F0
+0520 (ZET)	01500
−0520 (NZT)	03500
+0521	01510
−0521	03510
+0522 (XEQ)	01520
−0522	03520
+0523	01530
−0523	03530
+0524	01540
−0524	03540
+0525	01550
−0525	03550
+0526	01560
−0526	03560
+0527	01570
−0527	03570
+0530	01580
−0530	03580
+0531	01590
−0531	03590
+0533	015B0
−0533	035B0
+0536	015E0
−0536	035E0
+0537	015F0
−0537	035F0
+0540 (RCHA)	01600
−0540 (RCHB)	03600
+0541 (RCHC)	01610
−0541 (RCHD)	03610
+0542 (RCHE)	01620
−0542 (RCHF)	03620
+0543 (RCHG)	01630
−0543 (RCHH)	03630
+0544 (LCHA)	01640
−0544 (LCHB)	03640
+0545 (LCHC)	01650
−0545 (LCHD)	03650
+0546 (LCHE)	01660
−0546 (LCHF)	03660
+0547 (LCHG)	01670
−0547 (LCHH)	03670
+0550	00A80
−0550	02A80
+0551	00A90
−0551	02A90
+0552	00AA0
−0552	02AA0
+0553	00AB0
−0553	02AB0

7094 Op Code		System/370 Address (Hex)
+0554		00AC0
−0554		02AC0
+0555		00AD0
−0555		02AD0
+0556		00AE0
−0556		02AE0
+0557		00AF0
−0557		02AF0
+0560	(LDQ)	01700
−0560		03700
+0561		01710
−0561		03710
+0562		01720
−0562		03720
+0563		01730
−0563		03730
+0564	(ENB)	01740
−0564		03740
+0565		01750
−0565		03750
+0566		01760
−0566		03760
+0567		01770
−0567		03770
+0570		00B80
−0570		02B80
+0571		00B90
−0571		02B90
+0572		00BA0
−0572		02BA0
+0573		00BB0
−0573		02BB0
+0574		00BC0
−0574		02BC0
+0575		00BD0
−0575		02BD0
+0576		00BE0
−0576		02BE0
+0577		00BF0
−0577		02BF0
+0600	(STZ)	01800
−0600	(STQ)	03800
+0601	(STO)	01810
−0601		03810
+0602	(SLW)	01820
−0602	(ORS)	03820
+0603		01830
−0603	(DST)	03830
+0604	(STI)	01840
−0604		03840
+0605		01850
−0605		03850
+0606		01860
−0606		03860

7094 Op Code		System/370 Address (Hex)
+0607		01870
−0607		03870
+0610		00880
−0610		02880
+0611		00890
−0611		02890
+0612		008A0
−0612		028A0
+0613		008B0
−0613		028B0
+0614		008C0
−0614		028C0
+0615		008D0
−0615		028D0
+0616		008E0
−0616		028E0
+0617		008F0
−0617		028F0
+0620		01900
−0620	(SLQ)	03900
+0621	(STA)	01910
−0621		03910
+0622	(STD)	01920
−0622		03920
+0623		01930
−0623		03930
+0624		01940
−0624		03940
+0625	(STT)	01950
−0625	(STL)	03950
+0626		01960
−0626		03960
+0627		01970
−0627		03970
+0630	(STP)	01980
−0630		03980
+0631		01990
−0631		03990
+0632		019A0
−0632		039A0
+0633		019B0
−0633		039B0
+0635		019D0
−0635		039D0
+0637		019F0
−0637		039F0
+0640	(SCHA)	01A00
−0640	(SCHB)	03A00
+0641	(SCHC)	01A10
−0641	(SCHD)	03A10
+0642	(SCHE)	01A20
−0642	(SCHF)	03A20
+0643	(SCHG)	01A30
−0643	(SCHH)	03A30

7094 Op Code	System/370 Address (Hex)
+0644	01A40
–0644	03A40
+0645	01A50
–0645	03A50
+0646	01A60
–0646	03A60
+0647	01A70
–0647	03A70
+0650	00A80
–0650	02A80
+0651	00A90
–0651	02A90
+0652	00AA0
–0652	02AA0
+0653	00AB0
–0653	02AB0
+0654	00AC0
–0654	02AC0
+0655	00AD0
–0655	02AD0
+0656	00AE0
–0656	02AE0
+0657	00AF0
–0657	02AF0
+0660	01B00
–0660	03B00
+0661	01B10
–0661	03B10
+0662	01B20
–0662	03B20
+0663	01B30
–0663	03B30
+0664	01B40
–0664	03B40
+0665	01B50
–0665	03B50
+0666	01B60
–0666	03B60
+0667	01B70
–0667	03B70
+0670	01B80
–0670	03b80
+0671	01b90
–0671	03b90
+0672	01bA0
–0672	03BA0
+0673	01BB0
–0673	03BB0
+0674	01BC0
–0674	03BC0
+0675	01BD0
–0675	03BD0
+0676	01BE0
–0676	03BE0

7094 Op Code	System/370 Address (Hex)
+0677	01BF0
–0677	03BF0
+0700	01C00
–0700	03C00
+0701	01C10
–0701	03C10
+0702	01C20
–0702	03C20
+0703	01C30
–0703	03C30
+0704	01C40
–0704	03C40
+0705	01C50
–0705	03C50
+0706	01C60
–0706	03C60
+0707	01C70
–0707	03C70
+0710	00880
–0710	02880
+0711	00890
–0711	02890
+0712	008A0
–0712	028A0
+0713	008B0
–0713	028B0
+0714	008C0
–0714	028C0
+0715	008D0
–0715	028D0
+0716	008E0
–0716	028E0
+0717	008F0
–0717	028F0
+0720	00980
–0720	02980
+0721	00990
–0721	02990
+0722	009A0
–0722	029A0
+0723	009B0
–0723	029B0
+0724	009C0
–0724	029C0
+0725	009D0
–0725	029D0
+0726	009E0
–0726	029E0
+0727	009F0
–0727	029F0
+0730	01D80
–0730	03D80
+0731	01D90
–0731	03D90

7094 Op Code	System/370 Address (Hex)
+0732	01DA0
−0732	03DA0
+0733	01DB0
−0733	03DB0
+0735	01DD0
−0735	03DD0
+0736	01DE0
−0736	03DE0
+0740	00A80
−0740	02A80
+0741	00A90
−0741	02A90
+0742	00AA0
−0742	02AA0
+0743	00AB0
−0743	02AB0
+0744	00AC0
−0744	02AC0
+0745	00AD0
−0745	02AD0
+0746	00AE0
−0746	02AE0
+0747	00AF0
−0747	02AF0
+0750	01E80
−0750	03E80
+0751	01E90
−0751	03E90
+0752	01EA0
−0752	03EA0
+0753	01EB0
−0753	03EB0
+0755	01ED0
−0755	03ED0
+0757	01EF0
−0757	03EF0
+0760 (PSE)	01F00
−0760 (MSE)	03F00
+0761 (NOP)	01F10
−0761	03F10
+0762 (RDS)	01F20
−0762	03F20
+0763 (LLS)	01F30
−0763 (LGL)	03F30
+0764 (BSR)	01F40
−0764 (BSF)	03F40
+0765 (LRS)	01F50
−0765 (LGR)	03F50
+0766 (WRS)	01F60
−0766	03F60
+0767 (ALS)	01F70
−0767	03F70
+0770	01F80
−0770	03F80

7094 Op Code	System/370 Address (Hex)
+0771	01F90
−0771	03F90
+0772 (REW)	01FA0
−0772 (RUN)	03FA0
+0773	01FB0
−0773 (RQL)	03FB0
+0775	01FD0
−0775	03FD0
+0776 (SDN)	01FE0
−0776	03FE0
+0777	01FF0
−0777	03FF0
−1xxx	02D80

GROUP II: OPERATIONS USING AN INDEX REGISTER AS AN OPERAND

7094 Op Code	System/370 Tag	Hex Address
+0074	0 (TSX)	01380
	1	01390
	2	013A0
	3	013B0
	4	013C0
	5	013D0
	6	013E0
	7	013F0
+0532	0	01680
	1	01690
	2	016A0
	3	016B0
	4	016C0
	5	016D0
	6	016E0
	7	016F0
−0532	0	03680
	1	03690
	2	036A0
	3	036B0
	4	036C0
	5	036D0
	6	036E0
	7	036F0
+0534	0 (LXA)	01480
	1	01490
	2	014A0
	3	014B0
	4	014C0
	5	014D0
	6	014E0
	7	014F0

7094 Op Code	System/370 Tag	Hex Address
–0534	0 (LXD)	03480
	1	03490
	2	034A0
	3	034B0
	4	034C0
	5	034D0
	6	034E0
	7	034F0
+0535	0 (LAC)	01780
	1	01790
	2	017A0
	3	017B0
	4	017C0
	5	017D0
	6	017E0
	7	017F0
–0535	0 (LDC)	03780
	1	03790
	2	037A0
	3	037B0
	4	037C0
	5	037D0
	6	037E0
	7	037F0
+0634	0 (SXA)	01880
	1	01890
	2	018A0
	3	018B0
	4	018C0
	5	018D0
	6	018E0
	7	018F0
–0634	0 (SXD)	03880
	1	03890
	2	038A0
	3	038B0
	4	038C0
	5	038D0
	6	038E0
	7	038F0
+0636	0 (SCA)	01A80
	1	01A90
	2	01AA0
	3	01AB0
	4	01AC0
	5	01AD0
	6	01AE0
	7	01AF0
–0636	0 (SCD)	03A80
	1	03A90
	2	03AA0
	3	03AB0
	4	03AC0
	5	03AD0
	6	03AE0
	7	03AF0

7094 Op Code	System/370 Tag	Hex Address
+0734	0 (PAX)	01C80
	1	01C90
	2	01CA0
	3	01CB0
	4	01CC0
	5	01CD0
	6	01CE0
	7	01CF0
–0734	0 (PDX)	03C80
	1	03C90
	2	03CA0
	3	03CB0
	4	03CC0
	5	03CD0
	6	03CE0
	7	03CF0
+0737	0 (PAC)	01D00
	1	01D10
	2	01D20
	3	01D30
	4	01D40
	5	01D50
	6	01D60
	7	01D70
–0737	0 (PDC)	03D00
	1	03D10
	2	03D20
	3	03D30
	4	03D40
	5	03D50
	6	03D60
	7	03D70
+0754	0 (PXA)	01E00
	1	01E10
	2	01E20
	3	01E30
	4	01E40
	5	01E50
	6	01E60
	7	01E70
–0754	0 (PXD)	03E00
	1	03E10
	2	03E20
	3	03E30
	4	03E40
	5	03E50
	6	03E60
	7	03E70
+0756	0 (PCA)	00C80
	1	00C90
	2	00CA0
	3	00CB0
	4	00CC0
	5	00CD0
	6	00CE0
	7	00CF0

7094 Op Code	System/370 Tag	Hex Address
−0756	0 (PCD)	02C80
	1	02C90
	2	02CA0
	3	02CB0
	4	02CC0
	5	02CD0
	6	02CE0
	7	02CF0
+0774	0 (AXT)	00780
	1	00790
	2	007A0
	3	007B0
	4	007C0
	5	007D0
	6	007E0
	7	007F0
−0774	0 (AXC)	02780
	1	02790
	2	027A0
	3	027B0
	4	027C0
	5	027D0
	6	027E0
	7	027F0
+1xxx	0 (TXI)	00D80
	1	00D90
	2	00DA0
	3	00DB0
	4	00DC0
	5	00DD0
	6	00DE0
	7	00DF0

7094 Op Code	System/370 Tag	Hex Address
+2xxx	0 (TIX)	00E80
	1	00E90
	2	00EA0
	3	00EB0
	4	00EC0
	5	00ED0
	6	00EE0
	7	00EF0
−2xxx	0 (TNX)	02E80
	1	02E90
	2	02EA0
	3	02EB0
	4	02EC0
	5	02ED0
	6	02EE0
	7	02EF0
+3xxx	0 (TXH)	00F80
	1	00F90
	2	00FA0
	3	00FB0
	4	00FC0
	5	00FD0
	6	00FE0
	7	00FF0
−3xxx	0 (TXL)	02F80
	1	02F90
	2	02FA0
	3	02FB0
	4	02FC0
	5	02FD0
	6	02FE0
	7	02FF0

This appendix is a tabular listing of all the instructions that can be executed by means of a high-speed DIL.

7094 Op Code	Mnemonic
+0020	TRA
+0021	TTR
+0040	TLQ
+0074	TSX
+0100	TZE
–0100	TZE
+0131	XCA
+0140	TOV
–0140	TNO
+0200	MPY
–0200	MPR
+0220	DVH
+0221	DVP
+0241	FDH*
–0241	DFH*
+0241	FDP
–0241	DFDP
+0260	FMP
+0261	DFMP
+0300	FAD
+0301	DFAD
+0302	FSB
+0303	DFSB
+0340	CAS*
+0361	ACL
+0400	ADD
+0402	SUB
+0443	DLD*

7094 Op Code	Mnemonic
+0500	CLA
–0500	CAL
+0502	CLS
+0534	LXA
–0534	LXD
+0560	LDQ
–0600	STQ
+0601	STO
+0602	SLW
+0634	SXA
–0634	SXD
–0754	PXD
+0763	LLS
–0763	LGL
+0765	LRS
–0765	LGR
+0767	ALS
+0771	ARS
–0773	RQL
+0774	AXT
+1xxx	TXI
+2xxx	TIX
+3xxx	TXH
–3xxx	TXL

* Model 168 only.

Index

GA22-6955-1

709/7090/7094/7094 II Compatibility Feature for IBM System/370 Models 165, 165 II, and 168 (S/370-13) Printed in U.S.A. GA22-6955-1

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

709/7090/7094/7094 II Compatibility
Feature for IBM System/370 Models
165, 165 II, and 168

READER'S COMMENT FORM

Order No. GA22-6955-1

Your views about this publication may help improve its usefulness; this form will be sent to the author's department for appropriate action. Using this form to request system assistance or additional publications, however, will delay response. *For more direct handling of such request, please contact your IBM representative or the IBM Branch Office serving your locality.*

Possible topics for comment are:

Clarity Accuracy Completeness Organization Index Figures Examples Legibility

What is your occupation? ____________________

Number of latest Technical Newsletter (if any) concerning this publication: ____________

Please indicate in the space below if you wish a reply.

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.)

Cut or Fold Along Line

GA22-6955-1

Cut or Fold Along Line

Your comments, please . . .

This manual is a reference source for systems analysts, programmers and operators of IBM systems. Your comments will help us produce better publications for your use. Each reply will be carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Fold Fold

FIRST CLASS
PERMIT NO. 419
POUGHKEEPSIE, N.Y.

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 Fold

IBM®

**International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)**

**IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)**