File No. S360-32
Form C28-6666-0

OS

IBM Systems Reference Library

IBM System/360 Operating System

Guide to Model 91 Support

This publication is a guide to considerations that must be understood when programming on the IBM System/360 Model 91. The information in this publication applies to the MVT version of the IBM System/360 Operating System.

PREFACE

This publication describes features of the IBM System/360, Model 91, as they apply to the IBM System/360 Operating System, Multiprogramming with a Variable Number of Tasks (MVT). All users of the Model 91 should be aware of the implications of these features. The following publications, if read as prerequisites to, or in conjunction with, the associated sections of this publication, provide the necessary background:

IBM System/360 Operating System:

Supervisor and Data Management Services, Form C28-6646

Supervisor and Data Management Macro Instructions, Form C28-6647

Storage Estimates, Form C28-6551

System Generation, Form C28-6554

PL/I Language Specifications, Form C28-6571

Operator's Guide, Form C28-6540

Messages and Codes, Form C28-6631

PL/I (F) Programmer's Guide, Form C28-6594

FORTRAN IV Library Subprograms, Form C28-6596

System programmers and, particularly, programmers writing problem code in the assembly language must be thoroughly familiar with the associated material in the following publications:

IBM System/360 Principles of Operation, Form A22-6821

IBM System/360 Model 91 Functional Characteristics, Form A22-6907

The latter publication contains information that is unique to the Model 91 and may amend corresponding material in the Principles of Operation publication.

First Edition (November 1967)

This publication corresponds to Release 14.

Specifications contained herein are subject to change from time to time. Any such change will be reported in subsequent revisions or Technical Newsletters.

This publication was prepared for production using an IBM computer to update the text and to control the page and line format. Page impressions for photo-offset printing were obtained from an IBM 1403 Printer using a special print chain.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form is provided at the back of this publication for reader's comments. If the form has been removed, comments may be addressed to IBM Corporation, Programming Systems Publications, Department D58, PO Box 390, Poughkeepsie, N. Y. 12602

ILLUSTRATIONS

FIGURES

TABLES

INTRODUCTION

This publication presents information of which you should be aware when programming with the MVT version of the IBM System/360 Operating System for the IBM System/360 Model 91. The major sections of the book are concerned with the following:

- Instruction-processing techniques of the Model 91, including the concepts of an imprecise interruption and a pipe-line drain, and the sequence of the instruction-executions.

- Considerations regarding the results of floating-point operations and the manner in which these results (on the Model 91) are different from the results of the same operations on the other models.

- Changes to the supervisory functions of the operating system. These functions include the consideration of user-exit specifications for imprecise interruptions, a broadening of interruption-handling features, and the introduction of a routine to simulate decimal instructions.

- Language processor considerations for Model 91 programming. The main areas here are concerned with FORTRAN and PL/I programs.

- The application of the CENPROCS, the SUPRVSOR, and the SCHEDULR macro instructions to Model 91 programming. The operand parameter requirements and changes for these macro instructions are described.

- The use of the IBM 2250 Display Unit Model 1 as a system operator's console. General considerations are presented, and sections pertaining to the description of the device and to operator techniques are included.

- An estimate of the storage requirements of the operating system for the Model 91. This section includes both the main storage and the auxiliary storage requirements.

INSTRUCTION PROCESSING

The IBM System/360 Model 91 (hereinafter referred to as the Model 91) is designed for scientific programming applications that require high speed internal processing. This speed is achieved by using efficient algorithms and concurrent operations (i.e., parallel, out-of-sequence processing of instructions within the various elements of the central processing unit (CPU)). The physical ordering of instructions within a program determines the instruction decoding order but does not necessarily determine the order of instruction execution. The decoder unit of the instruction processor identifies each instruction and delivers the instruction to the appropriate execution unit (or element) of the CPU. Each instruction is executed by the unit to which it has been assigned and at a time determined by the relationship between that unit and other execution units.

As a consequence of the routing of instructions to different execution units, there may be, concurrently, several instructions at different phases of execution in the CPU. Under certain conditions (see the following list of events), the decoder halts operation, and the instructions that have already been decoded are executed. This latter action is referred to as a "pipe-line drain" and is initiated by one of the following events:

- A program exception is detected.
- A request is received from an input/output channel.
- A signal is received from either the timer, the interrupt key, or an external unit.
- A machine malfunction occurs.
- A status-switching instruction (e.g., LOAD PSW, SUPERVISOR CALL) is encountered.

INTERRUPTIONS

Supervisor Call, external, machine check, and I/O interruptions are handled on the Model 91 in the same manner as for other models. (For details, see the Model 91 Functional Characteristics publication.)

Due to the concurrency of operations that may occur on the Model 91, the detection of a program exception (i.e., a hardware occurrence) may occur during the execution of an instruction that is no longer addressed by the current program status word (PSW). A program interruption caused in this manner is termed 'imprecise' since the offending instruction cannot be precisely identified by the instruction address in the PSW.

Note: In this discussion, an interruption is a suspension of normal program activities.

Whenever a program exception occurs, the decoding of further program instructions ceases, and the execution of all instructions that have been decoded (i.e., that are in the process of being executed) is completed before the resulting interruption is handled. During the period of time required for this completion process (i.e., the pipe-line drain), one or more additional program exceptions may be detected. If so, the result is a multiple-imprecise-interruption situation, which is one that contains a series of program exceptions, in addition to the one originally triggering the program interruption.

Because a program exception may result in either an imprecise or a precise interruption, a method has been established so that the program old PSW indicates which type has been encountered. The content and significance of the pertinent PSW fields for interruption indication are discussed in the section "Specify Program Interruption Exit," and they are outlined in Appendix A. Since the instruction that causes a program interruption may not always be identified (i.e., may be imprecise), on the Model 91 a program that depends upon only precise program interruptions may not be executed as anticipated.

SEQUENTIAL INSTRUCTION EXECUTION

For a program section that requires the serial or sequential execution of instructions, the following 'no-operation' instruction also initiates a pipe-line drain:

```
    BCR      M,0               where M ≠ 0
```

The implementation of this instruction on the Model 91 insures that all the instructions preceding it are executed before the instruction succeeding it is decoded. It is recommended that use of this instruction be minimized since such use may affect the performance of the Model 91.

It should be noted that isolating an instruction by preceding it and succeeding it with the aforementioned BCR instruction eliminates multiple imprecise interruptions from more than one instruction by virtue of the pipe-line drain effect. However, since multiple exceptions may occur in one instruction, this technique does not eliminate a multiple imprecise interruption nor does it change an imprecise interruption into a precise interruption. The use of the BCR instruction does not assure a programmer that he can fix up an error situation. In general, the only information available will be the address of the no-operation (i.e., BCR) instruction. The length of the instruction preceding the BCR instruction is not recorded, and generally there is no way to determine what that instruction is.

NONSEQUENTIAL INSTRUCTION EXECUTION

The CPU maintains a logical consistency with respect to its own operations, including the beginning and ending of I/O operations, but it does not assume responsibility for such consistency in the operations performed by asynchronous units. Consequently, for any asynchronous unit that depends upon a strict adherence to sequential (or serial) execution, a problem program must set up its own procedures to ensure the proper instruction sequence.

The following example taken from the Program Controlled Interruption (PCI) appendage for dynamic buffer allocation in the Basic Telecommunications Access Method (BTAM) illustrates the dependence of an asynchronous channel program upon serial execution. The example also demonstrates the use of the previously-mentioned 'BCR' instruction to effect serial execution.

The purpose of the PCI appendage is to maintain an uninterrupted transmission of data into main storage. The controlling factors in this transmission are the availability of buffers and the ability (of the appendage routine) to modify and chain two channel programs. Each channel program consists of the following two channel command word (CCW) chains:

Chain 1:	CCW1	READ	(into a buffer, with Data Chaining (CD) and PCI flags)
	CCW2	READ	(into CCW3, with Skip (SKIP) and Suppress Length Indication (SLI) flags)
Chain 2:	CCW3	READ	(into a buffer with CD and PCI flags)
	CCW4	READ	(into CCW1 with SKIP and SLI flags)

Description of the channel programs follows:

1. In Chain 1, CCW1 is initialized with the first available buffer address.

2. The address fields for CCW2 and CCW4 initially contain the storage addresses of CCW3 and CCW1, respectively.

3. When the PCI interruption in CCW1 occurs, the PCI appendage routine determines the address of the next available buffer and stores it into the address field of CCW3.

4. The command code in CCW2 is changed from a 'READ' to a 'transfer-in-channel' (TIC), and the command code in CCW4 is reset to 'READ' as shown in Chain 2. (Note: the first time through the channel program, CCW4 is already set to 'READ'.)

5. When the PCI interruption in CCW3 occurs, the PCI appendage routine determines the address of the next available buffer (after the one indicated in step 3) and stores it into the address field of CCW1.

6. The command code in CCW4 is changed from a 'READ' to a 'TIC', and the command code in CCW2 is reset to 'READ' as shown in Chain 1.

Steps 3 through 6 of the preceding sequence of events continue until the input data transmission has been completed. The PCI appendage instructions that accomplish the alteration of the CCWs are:

```
ST    (available buffer address into CCW3 (or CCW1))

BCR   15, 0

MVI   (into CCW2 (or CCW4), the code for a 'TIC' command)

MVI   (into CCW4 (or CCW2), the code for a 'READ' command)
```

On the Model 91, use of the BCR instruction effects a pipe-line drain to ensure that the proper buffer address is stored before the READ command is changed to a TIC command. If, as could happen on the Model 91 when the BCR instruction is omitted, the MVI instructions were completed out of sequence (i.e., before the ST instruction), the channel could fetch a READ (into buffer) command with an old buffer address. In such a situation, the new input would overlay the old data, and the results could be serious.

Note: In this example, it is assumed that the PCI appendage instructions necessary to alter the format of the CCWs were executed before the channel fetched the "READ xx (SKIP,SLI)" command. If, in a given situation, this is not the case, then the fetching of the "READ xx (SKIP,SLI)" command is executed without transfer of data and causes a normal termination. Then the appropriate channel program must be restarted at the expense of additional input/output time.

For the Model 91, floating-point operations are somewhat different than they are on other models. The three cases involving differences are discussed in the following sections.

FLOATING-POINT DIVIDE

On the Model 91, the quotient of a floating-point divide operation may be different from the quotient obtained on other models using the same dividend and divisor. If the division results in a remainder of zero, the quotient is the same regardless of the machine model. For nonzero remainders, the quotient on the Model 91 will always be either equal to or greater than the quotient (for the same division problem) on other models. If the quotient is greater, its absolute value is greater by a numerical amount equal to one in the low-order bit position (of the quotient). For short-precision results, the low-order bit is bit 31. For long-precision results, the low-order bit is bit 63.

It is possible for this one-bit difference to be reflected by an increase in the characteristic of the quotient. In this case, there could be a set of dividend/divisor pairs which could produce varying results between the Model 91 and other models. One example of this might be a dividend/divisor pair that produces an overflow condition on the Model 91 and a number within the normal range on another model. Another example might produce a normal number on the Model 91 and an underflow condition on another model.

EXPONENT OVERFLOW

A floating-point exponent overflow exception results in the maximum floating-point number (see the publication IBM System/360 Principles of Operation) being placed in the 'result' register. The correct sign of the result is appended to the result in the register. For operations using long-precision, all 56 of the fraction bits in the register are set to one. For operations using short-precision, the low-order 32 bits in the register remain unchanged. In addition and subtraction, the condition code reflects the sign of the result. This exception produces an imprecise interruption.

EXPONENT UNDERFLOW

The result of an exponent underflow exception is that a true zero is placed in the 'result' register. For long-precision operations, all 56 of the fraction bits in the register are set to zero. For short-precision operations, the low-order 32 bits in the register remain unchanged. In addition and subtraction, the condition code is set to zero. This exception produces an imprecise interruption if the mask bit (bit 38) in the PSW is set to one.

Note: Whenever an interruption occurs on other models of System/360, system routines provide the setting of the result register when requested. To maintain compatibility, these operations are performed in the hardware of the Model 91 since the imprecise interruption prohibits the programming technique.

SUPERVISOR SERVICES

For the Model 91, supervisor services that differ from those on other models are the use of the Specify Program Interruption Exit (SPIE) macro instruction, the Program First-Level Interruption Handler (PFLIH) routine, and the Decimal Simulator (IEAXDS00) routine.

SPECIFY PROGRAM INTERRUPTION EXIT

The SPIE macro instruction permits a problem program to selectively process program interruptions rather than have the control program abnormally terminate a task. A programmer may specify his own exit routine and the types of interruptions which cause control to be given to the routine. If a program interruption of a type not specified in the SPIE macro instruction occurs, the control program's abnormal termination routine is given control.

For precise program interruptions on the Model 91 (as for all program interruptions on other models of the IBM System/360) , there is one, and only one, program exception recorded (in the PSW) per interruption. However, to indicate imprecise and multiple imprecise interruptions, the introduction of ten new positional bit settings in the program old PSW is provided . A unique bit in the PSW is defined for each of the ten exceptions that can cause imprecise interruptions. Therefore, as a result of a pipe-line drain after an exception, one interruption could have caused all ten types of program exceptions to be recorded in the PSW. Multiple occurrences of a given type would not be shown since the appropriate bit is set only when the first of any given type of exception occurs.

Because of the possibility of imprecise interruptions and the introduction of new positions in the program old PSW, two additional considerations are required whenever a user exit routine is called to service program interruptions.

If imprecise interruptions cause the recording of a type or types of exceptions not specified by a SPIE macro instruction, the task is abnormally terminated. Therefore, to ensure that a user's exit routine receives control, all ten of the exceptions that can cause an imprecise interruption should be specified in the SPIE macro instruction, although only one exception may be of interest in a given situation. Within the user's exit routine, the recorded program exceptions can be screened, and the exit routine can determine the severity of the situation (i.e., either terminate the task or continue processing) without processing all of the exceptions that may have been present.

The second required consideration for exit routines is the recognition of imprecise and precise interruption cases. This can be done by interrogating both the interruption code and the instruction-length code in the program old PSW.[1] The two cases can be distinguished as follows:

[1] Appendix A contains details of the program old PSW.

Program PSW	Interruption	
Bits	Precise	Imprecise
16-25	zero	at least one nonzero
26-31	at least one nonzero	zero
32-33	at least one nonzero	zero

Although certain types of interruptions are normally imprecise on the Model 91 (see the IBM System/360 Model 91 Functional Characteristics publication), a SPIE routine should be aware of situations in which the program old PSW records these interruptions as precise. For example, if the Decimal Simulator routine (see a later section in this publication) detects protection or data exceptions, precise interruptions are simulated.

PROGRAM FIRST-LEVEL INTERRUPTION HANDLER

The Program First-Level Interruption Handler (PFLIH) routine is redesigned for the Model 91 to ensure that the program interruption mechanism is handled in a consistent manner. Three major differences should be recognized.

The first difference occurs only when a decimal simulator routine is included in the control program. Under this condition, a decimal operation code (which always will appear on the Model 91 as an invalid operation code) causes the PFLIH routine to invoke the decimal simulator routine.

The second difference is that if a decimal simulator routine is not included in the control program, and if the TESTRAN program is in control when a decimal operation code is encountered, the PFLIH routine returns control to a TESTRAN error routine, which terminates the task.

The third difference is the inclusion of coding to verify that either an imprecise or a multiple imprecise interruption contains only those exceptions that are a subset of the exceptions specified by the execution of a SPIE macro instruction. If this condition is satisfied, a user's routine is given control. Otherwise, the task is abnormally terminated.

Note: When an imprecise or a multiple-imprecise program interruption causes a task to be terminated, the completion code is "0C0" since the last digit reflects the decimal content of bits 26-31 in the program old PSW.

DECIMAL SIMULATOR

With the exception of the Edit (EDIT) and Edit and Mark (EDMK) instructions, the Model 91 is not equipped with the decimal instruction feature. An optional Decimal Simulator (IEAXDS00) routine is provided to accept decimal arithmetic instructions.

When a decimal operation exception is detected, a precise program interruption occurs. If a decimal simulator routine is not included in the operating system, either the task is terminated or control goes to a programmer exit routine if a SPIE macro instruction specifying invalid operation codes has been executed. If a decimal simulator routine is included in the operating system, it is given control. As far as the individual programmer is concerned, problem programs will be executed without modification; that is, the effect of the simulated decimal arithmetic operation is transparent to a user.

Speed?

All conditions which would ordinarily (if the decimal instruction feature were provided) result in a program interruption (e.g., addressing exceptions, data exceptions, etc.) are detected. In the general case, all exceptions are treated as if they were detected by the hardware except that the condition code and the contents of main storage are unchanged. Each condition is noted by setting the appropriate interruption code in the program old PSW. Control is returned to the interruption handler routine for action such as passing control to a SPIE routine or terminating the task. All exceptions occurring during decimal simulation are recorded by the Decimal Simulator routine as precise interruptions.

On the Model 91, there are two cases in which the IBM Decimal Simulator (IEAXDS00) routine takes special action. The first case involves a decimal overflow condition. A problem answer is calculated, and the result is placed in a main storage location that is addressed by the first operand. The result consists of the low-order portion of the true result and is truncated to the length specified by the first operand. The condition code in the PSW is set to three. If the decimal overflow mask bit in the PSW is zero, the Decimal Simulator routine returns control to the program from which it was given control. If the mask bit is one, control is returned to the Program First-Level Interruption Handler routine.

In the second case, if the TESTRAN program is in either the 'trace' or the 'go back' mode, the Decimal Simulator routine recognizes the situation and returns control as specified by the TESTRAN program. If an exception is detected during the course of a decimal simulation, the Decimal Simulator routine returns control to the Program First Level Interruption Handler routine, which eventually returns control to the TESTRAN program.

Note: The use of decimal instructions should be minimized when the Decimal Simulator routine is used on a Model 91. Decimal Simulation requires both a relatively long execution time and that the central processing unit (CPU) be disabled during simulation. Thus, functions relying upon time-dependent responses would not be able to utilize the CPU in a consistently responsive manner.

eg PUT DATA;

The majority of language programs written for the IBM System/360 Operating System require no special modifications for execution on the Model 91. However, certain differences do exist as indicated in the following sections.

ASSEMBLER

Programs coded in the assembly language can be executed without any further considerations beyond those already mentioned in this publication. However, more efficient use of the CPU with resulting improvement in performance can be gained by a study of the coding and time considerations outlined in the appendixes of the publication IBM System/360: Model 91 Functional Characteristics. A22-6907-2 (May69)

Note: If a program uses decimal arithmetic, a decimal simulator routine must be included in the control program.

COBOL AND RPG

Compared to other models, increases in performance on the Model 91 may be realized only for programs in which the use of decimal arithmetic is minimized.

FORTRAN

For programs coded in FORTRAN, the major areas in which considerations must be taken when applied to the Model 91 involve:

- The action of the object time program interruption exit routine.
- A boundary adjustment routine.

PROGRAM INTERRUPTION EXIT ROUTINE

The library subroutine that handles interruptions has been modified to recognize precise, imprecise, and multiple imprecise interruptions. Multiple imprecise interruptions may require that the subroutine set more than one of the indicators (for divide check, exponent overflow, and exponent underflow).

Modifications are made to the message that is issued for the following program exceptions:

- Fixed-point divide
- Decimal divide
- Floating-point exponent overflow
- Floating-point exponent underflow
- Floating-point divide

The format of the message issued is

IHC210I PROGRAM INTERRUPT (x) OLD PSW IS y

where x represents one of the letters P (for precise), I (for imprecise), or M (for multiple imprecise of different classes), and y is the hexadecimal representation of the program old PSW.

BOUNDARY ADJUSTMENT ROUTINES

Specification of optional routines provide boundary adjustments for, or simulation of, instructions that cause specification exceptions. However, the nature of these 'fix-up' routines requires the identification of the instruction that causes the exception. Since specification exceptions on the Model 91 generate imprecise interruptions, boundary adjustments cannot be made. Thus, when the FCRTRAN library is specified for the Model 91 (see the section "System Generation"), boundary alignment must not be requested.

Note: If boundary alignment is requested when the FORTRAN library is specified for the Model 91, the related task eventually may be terminated for the following reason. Boundary alignment is made with respect to the instruction addressed by the program old PSW, but since the related interruption may have been imprecise, the old PSW may not contain the address of the offending instruction.

PL/I

PL/I programs that are to be run on the Model 91 should be compiled using the "M91" compiler option. (The standard default for this option is "NOM91".) This option may be set during system generation time or may be specified in the PARM field of the EXEC statement used for a compilation. In the implementation of this option, the PL/I (F) processor and library have been modified to permit source language processing of imprecise program interruptions. The compiler inserts 'no-operation' instructions (see the section "Compilation") at certain points in the program, and an External Symbol Dictionary (ESD) entry is created for the PL/I library module IHEM91, which is required only when an object program is to be executed on a Model 91. Module IHEM91 provides the facilities:

- for detecting multiple-exception imprecise interruptions
- for setting the value that is returned by the ONCOUNT built-in function (discussed further on)
- and for raising the appropriate PL/I conditions

Programs that are compiled under the "M91" option are also executable on other IBM System/360 models supported by the IBM System/360 Operating System.

COMPILATION

The implementation of the PL/I compilation process for the Model 91 is indicated in the following manner . A method of localizing imprecise program interruptions to a particular segment of a program by using the 'no-operation' instruction "BCR M,0" has been previously described (see the section "Sequential Execution"). Specification of the "M91" option causes the PL/I compiler to generate object-code BCR instructions to insure that interruption processing results in the correct action as specified in a PL/I source program. The situations in which the BCR instruction is generated are:

- Before an "ON" statement.
- Before a "REVERT" statement.
- Before compiled code to set the "SIZE" condition.
- Before compiled code to change prefix options.
- For a null statement. (This feature provides the programmer with source language control over the timing of program interruptions.)
- Before every statement if the "STMT" option (in the PARM field of the EXEC statement) is used. (This is an important debugging tool.)

The average PL/I program can be executed without significant degradation in speed or space requirements. However, use of the "STMT" option will cause both the execution time and the object-code space requirements to increase in direct proportion to the number of statements in the program. The benefits obtained as a result of the debugging aid that this option gives, should offset and justify these increases.

EXECUTION

If the M91 compiler is specified, the PL/I library module IHEM91 is included when the object module is linkage-edited, and it is called when an imprecise interruption is detected. This module provides the facilities for detecting multiple-exception imprecise interruptions, setting the value returned by the ONCOUNT built-in function, and raising the appropriate PL/I conditions. When a (PL/I) program interruption on the Model 91 results from multiple program exceptions, only one of the PL/I conditions (given in (1) and (2) below) is raised for each type of exception indicated. The order of processing of the exceptions is as follows:

1. PL/I conditions in the order:

    - UNDERFLOW
    - FIXEDOVERFLOW or SIZE
    - ERROR if system action is required for either FIXEDOVERFLOW or SIZE
    - OVERFLOW
    - ERROR if system action is required for CVERFLOW
    - ZERODIVIDE
    - ERROR if system action is required for ZERODIVIDE

    The conditions FIXEDOVERFLOW and SIZE cannot occur together since the same hardware condition raises both of them.

2. Other hardware exceptions in the order:

   - data
   - specification
   - addressing
   - protection

Conditions and exceptions are raised in the preceding order until one of the following situations occurs:

- A "GO TO" statement arising out of an 'ON-unit' is executed. All other exceptions will then be lost.
- The 'ERROR-condition' is raised. If the program is terminated as a result of this action (i.e., system action causing the ERROR condition to be raised, followed by the FINISH condition), messages will be printed to indicate the nature of the unprocessed exceptions. The exceptions themselves will not be processed.

The ONCOUNT Built-In Function

If a program interruption is a multiple-exception imprecise interruption, the "ONCOUNT" built-in function provides a count (in the form of a binary integer) of the number of exception types (not PL/I ON-conditions), including the current one, remaining to be processed. If this ("ONCOUNT") function is used when only a single exception has occurred, or if it is used outside of an 'ON-unit', a count value of zero is indicated.

MODEL 91 OBJECT-TIME DIAGNOSTIC MESSAGES

If PL/I object-time diagnostic messages are issued as a result of an imprecise interruption, the words "AT OFFSET..." are replaced by "NEAR OFFSET...", since in these circumstances the instruction causing the interruption cannot be precisely identified.

After a multiple-exception imprecise interruption on a Model 91, certain exceptions will remain unprocessed if the ERROR condition is raised before all the exceptions have been handled. If the program subsequently is terminated as a direct result of the ERROR condition being raised in these circumstances, one or more of the following messages will be printed out.

IHE810I PROTECTION EXCEPTION UNPROCESSED AFTER MULTIPLE-EXCEPTION IMPRECISE INTERRUPT

IHE811I ADDRESSING EXCEPTION UNPROCESSED AFTER MULTIPLE-EXCEPTION IMPRECISE INTERRUPT

IHE812I SPECIFICATION EXCEPTION UNPROCESSED AFTER MULTIPLE-EXCEPTION IMPRECISE INTERRUPT

IHE813I DATA EXCEPTION UNPROCESSED AFTER MULTIPLE-EXCEPTION IMPRECISE INTERRUPT

IHE814I ZERODIVIDE UNPROCESSED AFTER MULTIPLE-EXCEPTION IMPRECISE INTERRUPT

IHE815I OVERFLOW UNPROCESSED AFTER MULTIPLE-EXCEPTION IMPRECISE INTERRUPT

SYSTEM ENVIRONMENT RECORDING

The System Environment Recording (SER) function components have been constructed specifically for the Model 91. As such, they are required for reliability and serviceability considerations.

SER1

The resident version (SER1) of the System Environment Recording function is the only one that supports the Model 91.

EREP

The version of the Environment Recording Edit and Print (EREP) function used by the Model 91 is compatible with the MVT configuration of the operating system. This version will run as a job under the MVT option of the IBM System/360 Operating System.

SYSTEM GENERATION (SYSGEN)

When the Starter Operating System is used by the IBM System/360 Operating System to generate a configuration of the MVT version on a Model 91, the INHIBIT OVERLAP switch on the system control panel must be in the "inhibit overlap" position. By inhibiting overlap, instructions will be executed serially and precise program interruptions will occur for all exceptions except protection exceptions. Portions of the starter operating system depend on the adherence to these two factors for correct execution.

Regardless of the generating system used, the following macro instructions are required when generating a new MVT version of the operating system for the Model 91. It is essential that all required parameters and values be specified as indicated.

- CENPROCS
- SUPRVSOR

If FORTRAN is included in the system, certain operands must be specified in the FORTLIB macro instruction.

If PL/I is included in the system, certain operands must be specified in the PL1 macro instruction.

In addition, there are system generation considerations for using the IBM 2250 Display Unit, Model 1 as an operator's console (as discussed in the next main section of this publication). These considerations pertain to the use of the SCHEDULR macro instruction.

THE CENPROCS MACRO INSTRUCTION

The CENPROCS macro instruction describes the central processing unit. The format of this macro instruction, using the conventions described in the System Generation publication, is as follows:

Name	Operation	Operand

	CENPROCS	MODEL=91
		$\text{STORAGE=}\left\{\begin{array}{c}\text{J}\\ \text{K}\\ \text{JK}\\ \text{KK}\\ \text{L}\\ \text{JL}\\ \text{KL}\end{array}\right\}$
		INSTSET=UNIV
		[FEATURE=(feature[,feature])]

Operand Parameters

MODEL=91
    specifies the Model 91 as the central processing unit.

STORAGE=size
    specifies the size of main storage as one of the following values:

Value	Storage Size (in bytes)
J	1,048,576
K	2,097,152
JK	3,145,728
KK	4,194,304
L	4,194,304
JL	5,242,880
KL	6,291,456

INSTSET=UNIV
    specifies that the central processing unit has the Universal Instruction Set. The value of UNIV is implied although the decimal feature on the Model 91 includes only the "EDIT" and "EDMK" instructions. The inclusion of the Decimal Simulator routine (IEAXDS00) is NOT to be implied by the use of UNIV.

FEATURE=feature_n
    specifies the optional features that may be installed in the central processing unit as one or more of the following values:

TIMER (interval timer)
PROTECT (storage protection)

On the Model 91, both features are implied whether or not this parameter is coded.

THE SUPRVSOR MACRO INSTRUCTION

The task supervisor options are selected through the use of this macro instruction. The format of this macro instruction, using the conventions described in the System Generation publication, is shown below. In the discussion of the operands for this macro instruction, only those descriptions that either amend or are to replace the descriptions given in the System Generation publication are shown here. The remaining operand descriptions are as shown in the System Generation publication.

Name	Operation	Operand
	SUPRVSOR	[RESIDNT=(function[,function]...)] OPTIONS=(option [,option] ...) $\left[\text{WAIT=}\left\{\begin{array}{l}\underline{\text{SINGLE}}\\ \text{MULTIPLE}\end{array}\right\}\right]$ $\left[\text{TIMER=}\left\{\begin{array}{l}\underline{\text{TIME}}\\ \text{INTERVAL}\\ \text{JOBSTEP}\end{array}\right\}\right]$ [TRACE=number] SER=SER1

Operand Parameters

OPTIONS=(option [,option$_n$])
specifies the supervisor options as described in the System Generation publication and/or by the following text:

DECSIM
specifies the inclusion of a decimal simulator routine. When the operating system includes COBCL, PL/I, or RPG, the DECSIM option should be specified. For processor-type system generation, using either COBOL, PL/I, or RPG, a decimal simulator routine should already reside in the operating system being generated.

SER=SER1
specifies the resident version of the system environment recording function. By using the SER1 value, a nucleus that can only be used on the Model 91 is generated. This parameter is required.

Note: For the Model 91, the use of the COMM option is recommended.

THE FORTLIB MACRO INSTRUCTION

The FORTLIB macro instruction specifies that the FORTRAN subroutine library (SYS1.FORTLIB) is to be included in the operating system being generated. In the discussion of the operands for this macro instruction, only those descriptions that either amend or replace the descriptions given in the System Generation publication are shown here. The remaining operand descriptions are as shown in the System Generation publication.

Name	Operation	Operand
	FORTLIB	DESIGN=$\begin{Bmatrix}G\\H\end{Bmatrix}$ [UNIT=name] [VOLNO=serial] [UNTABLE=number] [OBJERR=unit] [ONLNRD=unit] [ONLNPCH=unit] BOUNDRY=NOALIGN

Operand Parameters

BOUNDRY=NOALIGN
specifies that the execution-time boundary-alignment routine should be excluded from the system.

THE PL1 MACRO INSTRUCTION

The PL1 macro instruction specifies that the PL/I compiler is to be included in the operating system being generated. In the discussion of the operand field for this macro instruction, only the modifications (for the Model 91) that are not given in the System Generation publication are described here.

Name	Operation	Operand
	PL1	DESIGN=F
		$\left[\text{OBJCODE=}\left\{\begin{matrix}\underline{\text{NOM91}}\\ \text{M91}\end{matrix}\right\}\right]$
		[DELETE=(item[,item]...)]

Operand Parameters

OBJCODE

specifies the default option at compilation time for the inclusion in the object program of both coding and an additional module to support the Model 91. M91 specifies the inclusion of the coding and the module; NOM91 specifies that the coding and the module are not to be included.

DELETE

(The list of items which can be specified for this operand also includes M91 and NOM91. See the System Generation publication.)

THE SCHEDULR MACRO INSTRUCTION

Programming support for the 2250 System Operator's Console allows an IBM 2250 Display Unit Model 1 to be specified as either a primary or an alternate console. The SCHEDULR macro instruction described in the System Generation publication, must be used to specify job scheduler options. The CONSOLE and ALTCONS operands may specify the address of a 2250 equipped with a 4K or 8K buffer, an alphameric keyboard, a Light Pen, and a character generator. The address of the 2250 can be specified in either the CONSOLE or ALTCONS operand, but not in both.

Note: When the primary and alternate console devices are specified at system generation time, the considerations given in the following table must be observed.

Primary Console	Alternate Console	Acceptable Combination
2250	non-2250	Yes
non-2250	2250	Yes
non-2250	non-2250	Yes
2250	2250	No

The information in this section describes how to use the IBM 2250 Display Unit Model 1, as a system operator's console on the IBM System/360 Model 91. The functions and uses of the general display areas of the 2250 screen are explained in detail. In addition, operating techniques not covered in the publication IBM System/360 Operating System Operator's Guide, hereinafter referred to as the operator's guide, are included here. Information for starting the system, controlling input and output, controlling jobs through commands and statements, operating techniques, and answering messages, that is found in the operator's guide is applicable to this programming support.

PROGRAMMING SUPPORT

Use of the IBM 2250 Display Unit Model 1, (referred to here as the 2250) as a console input/output device on the IBM System/360 Model 91 is supported by System/360 Model 91 programming support. With the 2250, faster communication is provided between the operator and the system than a typewriter could provide, and more information can be shown than is feasible with a typewriter. The programming support enables the operator to perform the following functions:

- Observe displayed messages from the system and from the problem programs.
- Reply to messages which require replies.
- Issue commands to the system.
- Request display of system status information.

A hard copy of messages, replies, and commands can be provided.

Although the 2250 is a standard feature on the Model 91, the use of a 2250 as a system operator's console is optional, i.e., an installation can decide to use some other device as its primary console. The 2250 can be specified at system generation time as a primary console along with either a 1052 Printer-Keyboard or a composite console made up of two devices, a card reader and a printer, in order to provide for a hard copy of commands and messages. Messages are displayed by the 2250 and can be printed by a 1052 or by a printer. Commands can be entered and displayed by the 2250 and can be printed by means of the 1052 or by a printer.

Any 2250 can be specified at system generation time as the primary or the alternate console (see the section "The SCHEDULR Macro Instruction"). The 2250 can be switched from primary to alternate console status or from alternate to primary console status, as described in the operator's guide.

Note: When switching the 2250 from an alternate console status to a primary console status, it may be necessary to depress the ALT and END keys to get a display.

The operator can make the 2250 available for problem program usage by placing it in alternate console status. However, when a 2250 in alternate console status is allocated to a problem program, it is not available as a primary console until the job it is allocated to has been terminated.

MACHINE CONSIDERATIONS

The 2250 attached to the maintenance console as a standard feature of the Model 91, or any other 2250 attached to the Model 91 via an input/output channel, may be used as the system operator's console. Use of a 2250 as a primary system operator's console makes it unavailable for use with problem programs.

The 2250 attached to the maintenance console must be equipped with the following features:

- 8192-byte Buffer
- Light Pen Detect
- Alphameric Keyboard
- Character Generator

Any other 2250 to be used as a system operator's console requires only a 4096-byte buffer, but it must be equipped with the other features listed above.

2250 ATTACHED TO THE MAINTENANCE CONSOLE

The 2250 attached to the Model 91 maintenance console can be connected by a manual switch to the system control panel (maintenance mode), or can be connected to the Model 91 via a 2860 Selector channel or a Selector subchannel of a 2870 Multiplexor channel (non-maintenance mode).

In maintenance mode, the 2250 is not under program control, is connected only to the system control panel, and is not available as a system operator's console or problem program device. (See the publication, IBM System/360 Model 91, Functional Characteristics.) It can only be used as an output device capable of displaying storage, CPU registers, or storage control registers.

In non-maintenance mode, the 2250 is connected to an input/output channel and may be used either as a system operator's console or as a problem program device. In this mode, the 2250 is entirely under program control.

OTHER 2250'S

Any other 2250 included in the Model 91 configuration, but not attached to the maintenance console, can be attached and connected to the Model 91 via a 2860 Selector channel or by a Selector subchannel of a 2870 Multiplexor channel. It may be used as either a system operator's console or a problem program device, but in either case it cannot be used in maintenance mode.

The basic display illustrated in Figure 1 will be available to the system operator. This display is divided into four general areas. The line numbers for each area are shown in parentheses. These line numbers appear in the figure for illustration purposes only and are not in the actual screen display. The display areas are:

1. Write-to-Operator (WTO) Message Display Area (lines 3 through 39).

2. Write-to-Operator With Reply (WTOR) Message Display Area (lines 41 through 44).

3. Entry Display Area (lines 46 and 47).

4. Control Display Area (lines 49 and 50).

Single asterisks preceding alphameric characters indicate options that can be selected with the Light Pen. Light Pen detects on these characters (see note) result in an action of some kind being taken.

Note: In the case of header lines, the Light Pen detect is ignored.

The functions and uses of these areas are explained in detail in the paragraphs that follow.

WRITE-TO-OPERATOR MESSAGE DISPLAY AREA

The Write-to-Operator (WTO) message display area displays all messages issued by a WTO macro instruction, or its equivalent, in the control program or in a problem program.

The WTO area consists of 37 lines, each capable of containing a single message. The header line for this area is "*WTO MESSAGES" (see Figure 1, line 2). A maximum of one 74-character line is allowed for each message. Lines containing more than 74 characters are truncated. Each new message is entered at the bottom of the area, provided that a line is available and the MSGE HOLD option is NO (see "MSGE HOLD * NO"). As each new message is entered, the old messages move up one line.

All messages in the WTO area must be deleted by means of the Light Pen. When a Light Pen detect is received on any of the displayed characters in a message, the message will be flagged and a set of action options will appear on the right-hand side of the line (see Figure 2, line 39). If necessary the last characters of the message are temporarily overwritten. The Light Pen must then be used to select one of the following action options:

YES -- delete the selected message. The message will be erased and all messages being displayed above it will move down one line.

NO -- do not delete the selected message. The original message text will be restored with no change.

ALL -- delete the selected message and all messages displayed above it. The selected messages will be erased.

After a message (or messages) has been deleted, a new message can be entered on the bottom line unless the MSGE HOLD option in the Control Display area is YES (see "MSGE HOLD * YES"). New messages can be allowed to enter the WTO area by selecting the MSGE HOLD * NO option in the Control Display area.

A warning message is displayed in the Control Display area (see Figure 3, line 49) and an audible alarm is sounded when a new message cannot be displayed. The warning is deleted when the new message is displayed.

WRITE-TO-OPERATOR WITH REPLY MESSAGE DISPLAY AREA

The Write-to-Operator with Reply (WTOR) message display area displays all messages issued by WTOR macro instructions, or their equivalents, in the control program or in a problem program.

The WTOR area consists of four lines, each capable of containing a single message (see Figure 1, line 41 through 44). The header line for this area is "*WTOR MESSAGES" (see Figure 1, line 40). A maximum of one 74-character line is allowed for each message. A new message is displayed on any available line in the WTOR area and on the bottom line of the WTO area if there is a line available in both areas. An audible alarm is sounded as each new message is displayed.

When a reply to a WTOR message is received and accepted (see "Entry Display Area"), the corresponding message is deleted from the WTOR and the reply will be displayed in the WTO area.

WTOR messages may be deleted by means of the Light Pen in the same manner as WTO messages, except that when an individual message is selected, the ALL option will not appear on the right hand side of the line. Messages will be automatically deleted if the issuing task is terminated.

A warning message similar to the one shown for the WTO area is displayed in the Control Display area and an audible alarm is sounded when a new message cannot be displayed. The warning is deleted when the message is displayed.

ENTRY DISPLAY AREA

The Entry Display area is used to enter and display operator commands and to enter replies to WTOR messages. The header line for this area is "***USE KEYBOARD TO ENTER COMMANDS***" (see Figure 1, line 45).

This area consists of two lines which may contain a single reply or command. A reply or a command may contain a maximum of 126 characters in the two lines.

The Entry Display area is initialized with non-displayable characters. Therefore, the cursor will not be displayed except when the operator has caused it to move, by means of the BACKSPACE or JUMP keys, so that it is positioned at a displayable character, e.g., alphabetic, numeric, special, or blank.

Replies or commands are entered under cursor control by means of the alphameric keyboard. The cursor moves to the right one position for every character in the reply or command placed in the Entry area.

To indicate the end of the reply or command, the operator depresses the ALT and END keys. After the END key is depressed and the entry is read, the Entry Display area is blanked, and the cursor is repositioned at the left of the Entry Display area. The entry will be displayed in the WTO area and truncated to 74 characters.

To make a correction to a reply or command, the operator depresses the BACKSPACE key and, after proper repositioning, retypes the desired entry. To indicate cancellation of replies or commands, the operator depresses the ALT and CANCEL keys. When the cancelation is completed, the Entry Display area is blanked, and the cursor is repositioned at the left of the Entry Display area.

CAUTION: Other attentions, e.g., Light Pen, should not be issued until the Entry Display area has been blanked; otherwise the previous attention will be lost.

CONTROL DISPLAY AREA

The Control Display area of the screen (see Figure 1, lines 49 and 50) displays the options available to the operator via the Light Pen for selecting and controlling information to be displayed. WTO and WTOR warning messages are displayed in this area as they occur (see Figure 3, line 49). The header line for this area is "***USE LIGHT PEN TO SELECT OPTIONS***" (see Figure 1, line 48).

The operator selects an option by detecting via the Light Pen on any of the displayed characters in the desired option. These options are:

- *UNITS
- *COMMANDS
- MSGE HOLD * YES * NO
- MSGE PRINT * YES * NO

The functions of these options are described in the paragraphs that follow.

UNITS

This option produces a unit status table of 3-digit hexadecimal addresses of all input/output devices in the system and their current status (see Figure 3, lines 10 through 25). A maximum of 96 device addresses can be displayed. The header line for the unit status table is "*UNIT STATUS UNIT STATUS..." (see Figure 3, line 9). The address of each device is shown under the UNIT heading and the status of the device (i.e., ALLOC (on line and allocated), UNALLOC (on line and unallocated), OFFLINE, RDR (Reader), WTR (Writer), SYSRES (System Residence), CONSOLE, or ALTCONS (Alternate Console)) is shown under the STATUS heading.

The unit status table is displayed in the WTO area. The table may temporarily overlay WTO messages in the area. The presence of the unit status table does not affect the adding of messages to the WTO area. Messages not overlayed by the unit status table are retained.

This table may be deleted by a Light Pen detect on any character of the line, "***USE LIGHT PEN TO ERASE UNITS***". The messages overlayed by the table are restored.

COMMANDS

This option produces a list of operator command verbs. The list is displayed on 8 lines in the WTO area (see Figure 2, lines 27 through 34). The table may be temporarily written over WTO messages in the area. Older messages are displayed above the list and newer messages are displayed below it. The list is positioned in the WTO area below the unit status table so that both the command verb list and the unit status table may be displayed at the same time.

After the list of command verbs is displayed, the operator must display the format of a specific command by directing the Light Pen at any character in the desired command verb. The format of the selected command will be displayed in place of the command verb list (see Figure 3, lines 27 through 34). The formats of the commands duplicate those described in the operator's guide, including brackets to indicate optional parameters and braces to indicate a choice of parameters. Parameters shown in upper case in the operator's guide will be displayed using large-size characters; parameters shown in lower case will be displayed using basic-size characters.

The command format display may be deleted by a Light Pen detect on any character of the line, "***USE LIGHT PEN TO ERASE COMMAND***". The messages displaced by the command format are redisplayed in the WTO area.

MSGE HOLD * YES

This option is selected by a Light Pen detect on the YES in MSGE HOLD * YES (see Figure 2, line 50). The YES is underlined when in effect. This option prevents all messages issued by WTO macro instructions or their equivalents either in the control program or in the problem program from being entered into the WTO area. Messages may be deleted by Light Pen option while the MSGE HOLD option is YES.

MSGE HOLD * NO

This option is selected by a Light Pen detect on the NO in MSGE HOLD * NO (see Figure 2, line 50). The NO is underlined when in effect. Selection of this option negates the MSGE HOLD * YES option. The operator must use this option following a MSGE HOLD * YES option before he can again receive messages in the WTO area. When the system is initialized, the MSGE HOLD option is NO.

MSGE PRINT * YES

This option is selected by a Light Pen detect on the YES in MSGE PRINT * YES (see Figure 2, line 49). The YES is underlined when in effect. This option allows a hard-copy of WTO and WTOR messages to be printed by an alternate console, i.e., typewriter or printer, if one is present. When the system is initialized, the MSGE PRINT option is YES.

MSGE PRINT * NO

This option is selected by a Light Pen detect on the NO in MSGE PRINT * NO (see Figure 2, line 49). The NO is underlined when in effect. This option bypasses printing a hard-copy of WTO and WTOR messages by an alternate console, i.e., typewriter or printer.

ERROR HANDLING

Both synchronous and asynchronous errors may occur on the 2250. Synchronous errors occur during an input/output operation. Asynchronous errors occur at any time during display regeneration, but not while an input/output operation is being performed. The following paragraphs explain how the programming support for the 2250 System Operator's Console handles these errors.

SYNCHRONOUS ERRORS

When synchronous errors occur on either the primary or alternate console, and the 2250 is specified as the primary console, the operator can switch the alternate console to primary status by pressing the INTERRUPT key, as described in the operator's guide.

Synchronous errors occurring on the 2250 cause the screen to be blanked. Synchronous errors occurring on the alternate console cause the output of messages to be discontinued.

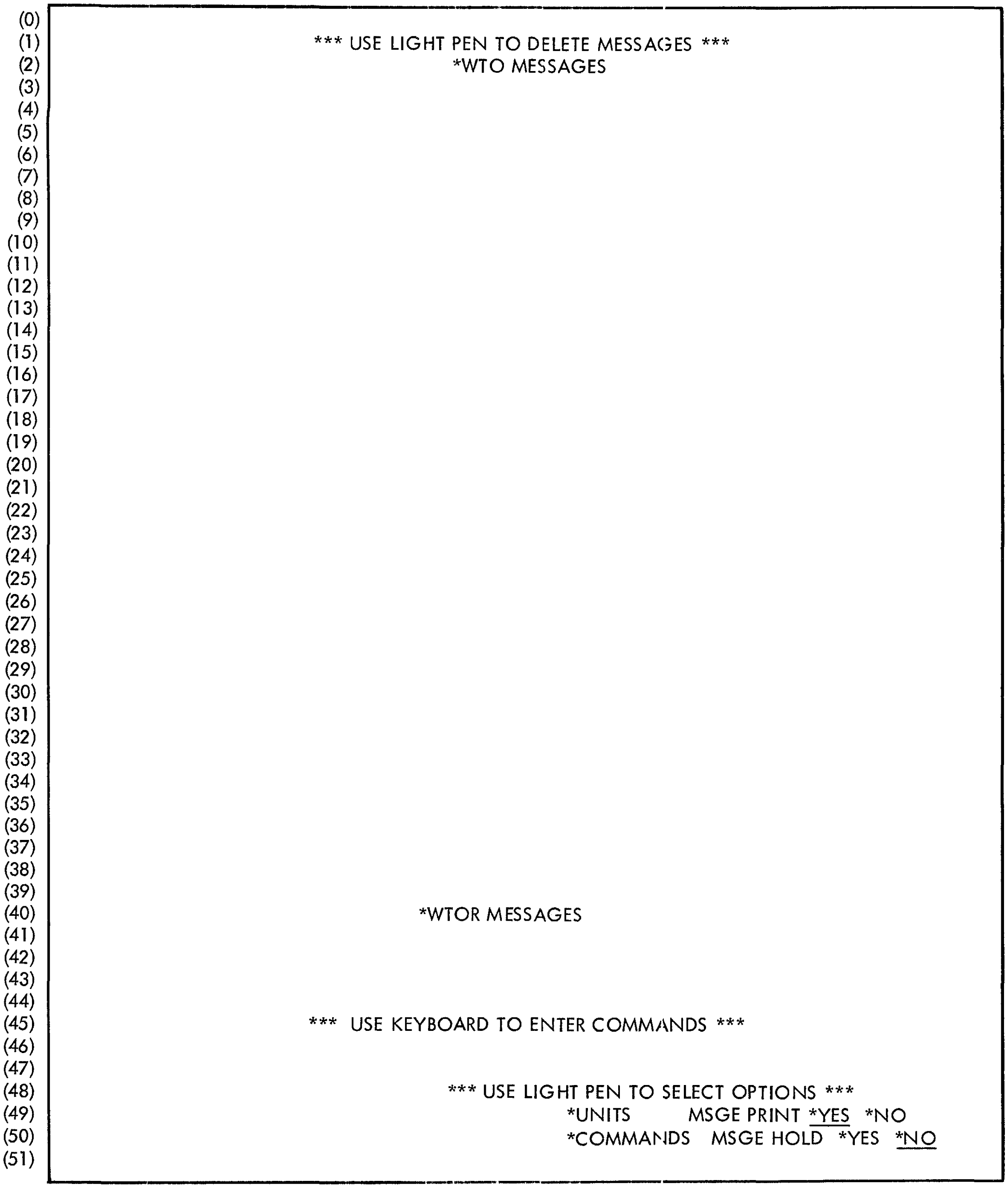
ASYNCHRONOUS ERRORS

When an asynchronous error occurs on the 2250, the screen is blanked and the following message appears:

> "A 2250 ERROR HAS OCCURRED. RECENT LIGHT PEN OR KEYBCARD ACTION
> MAY NEED TO BE REPEATED. PLEASE USE LIGHT PEN HERE * TO RESTART
> DISPLAY. BUFFER ADDRESS OF ERROR IS (xxxx)"

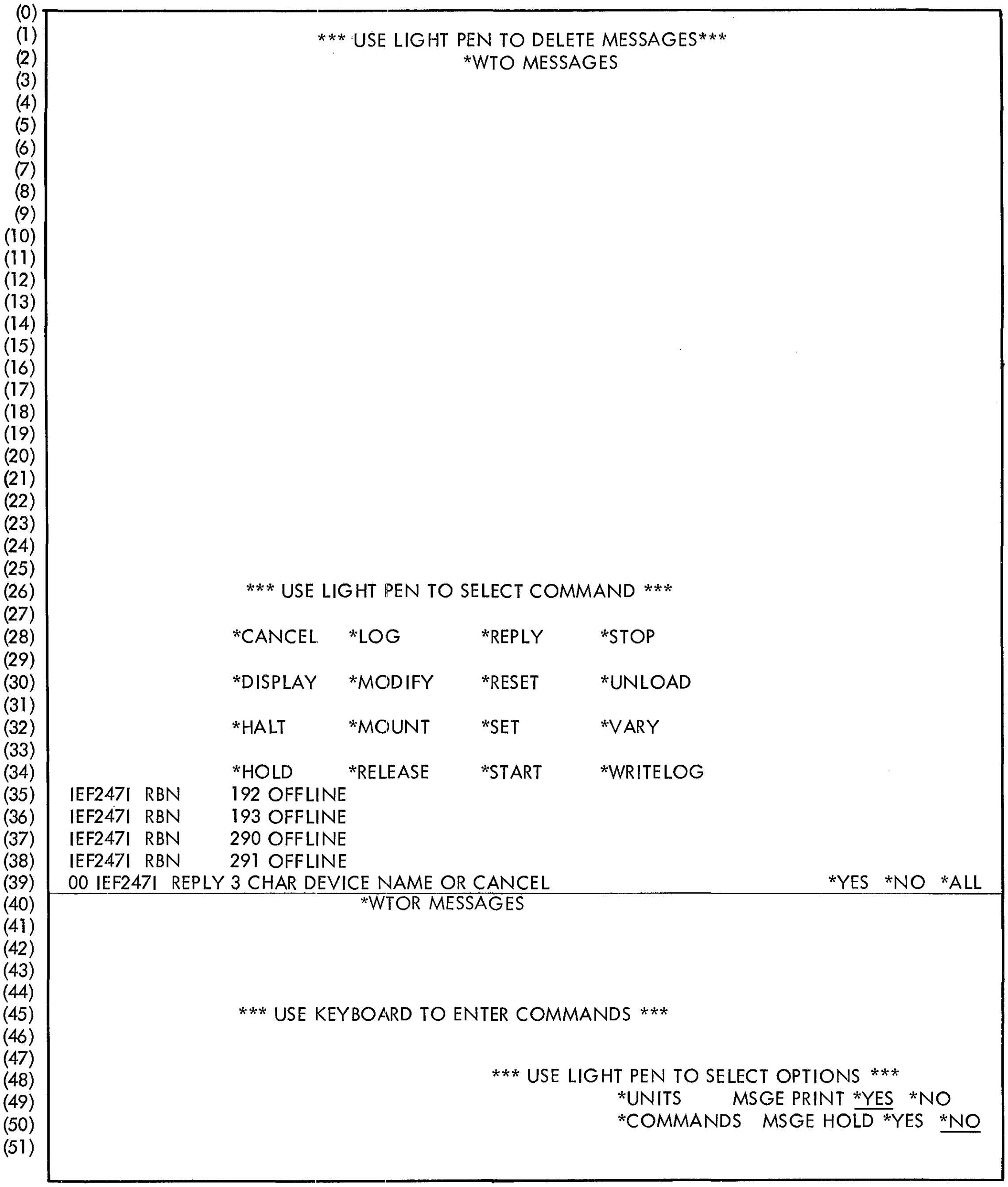
The message which notifies the operator of the occurrence of an asynchronous error will contain the buffer address of the byte which caused the error. If the error occurs twice consecutively at the same buffer address, the programming support will either: (a) bypass the 2250 and continue message output using the hard-copy console, if one is available; or (b) place the buffer address and hexadecimal "0800," i.e., data check, in Register 15 before putting the system into a Wait state.

When the operator performs a Light Pen Detect on any character in the message, the display will be restored. The displays restored will include WTO, WTOR, and warning messages, and MSGE HOLD and MSGE PRINT options. However, Unit Status, command formats or message delete options (YES-NO-ALL) will not be restored, and the Entry area will be blanked.



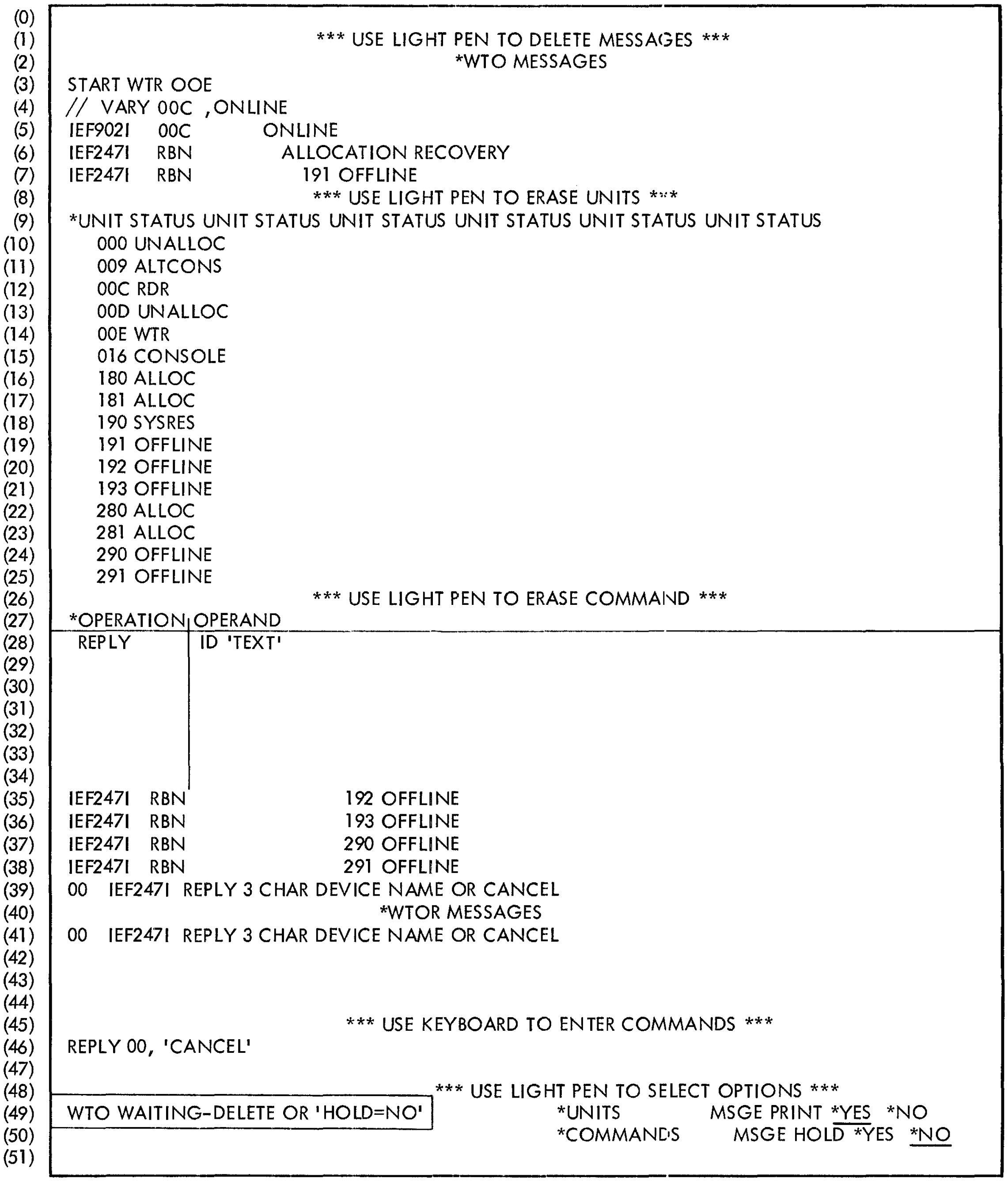
```
(0)
(1)                  *** USE LIGHT PEN TO DELETE MESSAGES ***
(2)                             *WTO MESSAGES
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)
(16)
(17)
(18)
(19)
(20)
(21)
(22)
(23)
(24)
(25)
(26)
(27)
(28)
(29)
(30)
(31)
(32)
(33)
(34)
(35)
(36)
(37)
(38)
(39)
(40)                          *WTOR MESSAGES
(41)
(42)
(43)
(44)
(45)                 ***  USE KEYBOARD TO ENTER COMMANDS ***
(46)
(47)
(48)                         *** USE LIGHT PEN TO SELECT OPTIONS ***
(49)                                *UNITS        MSGE PRINT *YES  *NO
(50)                                *COMMANDS   MSGE HOLD  *YES  *NO
(51)
```

Figure 1. Basic Display



```
(0)
(1)                     *** USE LIGHT PEN TO DELETE MESSAGES***
(2)                                *WTO MESSAGES
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(10)
(11)
(12)
(13)
(14)
(15)
(16)
(17)
(18)
(19)
(20)
(21)
(22)
(23)
(24)
(25)
(26)               *** USE LIGHT PEN TO SELECT COMMAND ***
(27)
(28)              *CANCEL   *LOG        *REPLY      *STOP
(29)
(30)              *DISPLAY  *MODIFY     *RESET      *UNLOAD
(31)
(32)              *HALT     *MOUNT      *SET        *VARY
(33)
(34)              *HOLD     *RELEASE    *START      *WRITELOG
(35)  IEF247I  RBN   192 OFFLINE
(36)  IEF247I  RBN   193 OFFLINE
(37)  IEF247I  RBN   290 OFFLINE
(38)  IEF247I  RBN   291 OFFLINE
(39)  00 IEF247I  REPLY 3 CHAR DEVICE NAME OR CANCEL                *YES  *NO  *ALL
(40)                          *WTOR MESSAGES
(41)
(42)
(43)
(44)
(45)              *** USE KEYBOARD TO ENTER COMMANDS ***
(46)
(47)
(48)                                *** USE LIGHT PEN TO SELECT OPTIONS ***
(49)                                         *UNITS      MSGE PRINT *YES  *NO
(50)                                         *COMMANDS   MSGE HOLD *YES  *NO
(51)
```

Figure 2. Basic Display With Command List and Action Options



```
(0)
(1)                          *** USE LIGHT PEN TO DELETE MESSAGES ***
(2)                                          *WTO MESSAGES
(3)   START WTR OOE
(4)   // VARY 00C ,ONLINE
(5)   IEF902I   00C        ONLINE
(6)   IEF247I   RBN           ALLOCATION RECOVERY
(7)   IEF247I   RBN              191 OFFLINE
(8)                          *** USE LIGHT PEN TO ERASE UNITS *'*
(9)   *UNIT STATUS UNIT STATUS UNIT STATUS UNIT STATUS UNIT STATUS UNIT STATUS
(10)     000 UNALLOC
(11)     009 ALTCONS
(12)     00C RDR
(13)     00D UNALLOC
(14)     00E WTR
(15)     016 CONSOLE
(16)     180 ALLOC
(17)     181 ALLOC
(18)     190 SYSRES
(19)     191 OFFLINE
(20)     192 OFFLINE
(21)     193 OFFLINE
(22)     280 ALLOC
(23)     281 ALLOC
(24)     290 OFFLINE
(25)     291 OFFLINE
(26)                         *** USE LIGHT PEN TO ERASE COMMAND ***
(27)  *OPERATION|OPERAND
(28)   REPLY     | ID 'TEXT'
(29)             |
(30)             |
(31)             |
(32)             |
(33)             |
(34)             |
(35)  IEF247I  RBN               192 OFFLINE
(36)  IEF247I  RBN               193 OFFLINE
(37)  IEF247I  RBN               290 OFFLINE
(38)  IEF247I  RBN               291 OFFLINE
(39)  00   IEF247I  REPLY 3 CHAR DEVICE NAME OR CANCEL
(40)                                   *WTOR MESSAGES
(41)  00   IEF247I  REPLY 3 CHAR DEVICE NAME OR CANCEL
(42)
(43)
(44)
(45)                           *** USE KEYBOARD TO ENTER COMMANDS ***
(46)  REPLY 00, 'CANCEL'
(47)
(48)                                 *** USE LIGHT PEN TO SELECT OPTIONS ***
(49)  WTO WAITING-DELETE OR 'HOLD=NO'        *UNITS        MSGE PRINT *YES  *NO
(50)                                         *COMMANDS      MSGE HOLD *YES  *NO
(51)
```

Figure 3. Basic Display With Unit Status, Command Format and Warning Message

The operating techniques described in the following paragraphs are not included in the operator's guide. They are to be used by the operator when the 2250 is the primary system operator's console on the Model 91. The operator uses the keyboard on the 2250 in the same manner as the keyboard on the 1052 Printer-Keyboard, Model 7, except that there are no REQUEST or EOB keys on the 2250. There is no need for a REQUEST key, and in place of the ALT and EOB keys, the operator uses the ALT and END keys on the 2250.

HOW TO START THE SYSTEM

Starting the system includes Initial Program Loading (IPL), readying the nucleus, and readying the scheduler, as described in the operator's guide. These procedures are the same except that while the nucleus is being readied by the Nucleus Initialization Program (NIP), the 2250 system operator's console will function as follows:

- Old messages will be automatically deleted when the screen is full, and new messages will be displayed.
- A Light Pen detect will be considered as a permanent error. The screen will be blanked. Another IPL will be required.

HOW TO ENTER COMMANDS FROM THE 2250 KEYBOARD

- Type the command.
- Signal end by pressing the ALT key and the END key.

To correct errors, the operator may cancel the entire line by pressing the ALT key and CANCEL key, and retype the command. If the entire line does not need to be canceled, the operator may correct errors by backspacing. He presses the BACKSPACE key, and after proper repositioning, retypes the desired entry.

HOW TO USE THE LIGHT PEN TO DELETE MESSAGES

- Point the Light Pen at the desired message in the WTO area.
- Activate the Light Pen.
- Point the Light Pen to the desired action option at the right of the message.
- Activate the Light Pen.

Note: Complete the above steps before attempting any other action.

HOW TO USE THE LIGHT PEN TO SELECT OPTIONS

- Point the Light Pen at the desired option in the Control Display area.
- Activate the Light Pen.

HOW TO USE THE LIGHT PEN TO DISPLAY COMMANDS

- Point the Light Pen at the COMMANDS option in the Control Display area.
- Activate the Light Pen.
- Point the Light Pen at the command verb desired for a display of the command format.
- Activate the Light Pen.

MESSAGES AND COMPLETION CODES

Programming support for the IBM 2250 System Operator's Console will utilize the same Wait State PSW error codes described in the Messages and Completion Codes publication. The error codes used are Nucleus Initialization Program (NIP) error codes and Supervisor error codes.

NUCLEUS INITIALIZATION PROGRAM (NIP) ERROR CODES

Error codes 000002 through 000007 are the applicable error codes used by the 2250 System Operator's Console programming support. The conditions causing these error codes to appear and the responses of the operator remain the same.

SUPERVISOR ERROR CODES

An error code of 0000E2 is used by the 2250 System Operator's Console programming support when a machine check or a channel failure has occurred and the system environment recording (SER) option, SER1, is not present in the system. The operator must perform the IPL procedure with the SEREP program, which is a stand-alone program to record machine environment, and give the results to the Customer Engineer.

Table 1 gives the estimates of storage requirements for the Model 91. The values in this table either must be added to the corresponding sizes normally computed for the MVT configuration or are requirements for the modules that are unique to the Model 91. The required storage includes only those items which result from the system generation specifications of MODEL=91 in the CENPROCS macro instruction and SER = SER1 in the SUPRVSOR macro instruction.

To use the information in Table 1, refer in the Storage Estimates publication to the sections on estimating the requirements for the following:

- Fixed main storage
- Dynamic main storage
- Auxiliary storage

ESTIMATING THE FIXED MAIN STORAGE REQUIREMENT

The following steps 1-3 should be performed, and of the steps 4-7, only those that are appropriate need be considered.

1. To compute the constant main storage requirement for MVT, add the amounts given in Table 1 for PFLIH (162 bytes) and the diagnostic scan-out area (1264 bytes) to the fixed amount required by the control program. The total diagnostic scan-out area on the Model 91 contains 1520 bytes, but this includes the 256-byte scan-out area that is standard on other models.

2. Determine a round-up factor and add it in so that the fixed amount required by the control program plus the amount required for the MVT control program options is a multiple of 2K (2048 bytes).

3. Since SER1 is required for the Model 91, the size (5900 bytes) given in Table 1 must be added to the basic fixed amount to determine the total fixed amount.

4. If decimal simulation is requested, the basic fixed storage requirement must be increased by the amount (3520 bytes) specified for IEAXDS00.

5. When the 2250 is selected as a system operator's console, there are two considerations:

   a. Fixed main storage: This must be increased by 4096 bytes for the Display Console Message (DCM) table, which is placed in the nucleus.

   b. Link pack area: The following routines from the link library are added to those always loaded into the link pack area:

Routine	Size (in bytes)
IEECVOP1	1410
IEECVOP2	2840
Total	4250

Consequently, the link pack area size requirement must be computed for this configuration, first by adding the above 4250 bytes to the total number of bytes contained in the routines always loaded into the link pack area, and then by rounding up the result to a multiple of 2K.

Note: Only two modules are explicitly required to be resident in the link pack area when using the 2250 as a system operator's console. However, it is recommended that, for optimum use of the 2250, the following five modules from the SVC library also be placed into the link pack area:

Module	Size (in bytes)
IGC3107B	968
IGC3207B	1008
IGC3407B	887
IGC3507B	944
IGC3607B	828
	Total 4635

If this is done, the link pack area requirement should be increased by this amount (4571 bytes).

6. When TESTRAN is specified, the size of the PFLIH routine is increased by 22 bytes, which must be added to the total fixed main storage requirement.

   Note: If the MODE=TRACE operand in the TESTRAN macro instruction is also requested, the total fixed main storage requirement must be increased by 28 bytes, rather than the previously indicated 22 bytes.

ESTIMATING THE DYNAMIC MAIN STORAGE REQUIREMENT

Steps 1 and 2 provide dynamic main storage considerations for the Model 91.

1. The minimum storage requirement for the system utility program IFCEREP0 (EREP) is 21,300 bytes on the Model 91.
2. When a PL/I program is compiled for the Model 91, the dynamic main storage requirement for the object module is an additional 210 bytes (i.e., the object module requires 210 bytes more than the object module of the same program compiled for another model of IBM System/360.

ESTIMATING THE AUXILIARY STORAGE REQUIREMENT

The following text gives the auxiliary storage requirement considerations for the Model 91.

1. The size of the SYS1.NUCLEUS data set is computed by using the appropriate formula from the Storage Estimates publication.
2. The size of the SYS1.SVCLIB data set may require an additional number of tracks as indicated in Table 1 in order to contain the following modules if the 2250 is specified as a system operator's console:

Module	Function	Size (in bytes)
IGC3107B	Attention-handling routine	968
IGC3207B	Message buffer handling routine	1008
IGC3407B	Puts messages into table for display	887
IGC3507B	I/O routine for message display	944
IGC3607B	I/O routine for message deletion	828
IGC3807B	Routine to handle "units and commands" option	898
IGC3I07B	OPEN and CLOSE for the 2250	921
IGC3707B	Asynchronous error routine	820
	Total	7274

Note: At least two 256-byte records will be required for a directory of these modules.

3. SYS1.LINKLIB has a constant requirement for EREP as specified in Table 1.

4. If the 2250 is specified as a system operator's console, the size of the SYS1.LINKLIB may require an additional number of tracks as indicated in Table 1 in order to contain the following modules:

Module	Function	Size (in bytes)
IEECVOP1	Order program for "units" option	1410
IEECVOP2	Order program for "commands" option	2840
	Total	4250

Note: The number of 256-byte records in existence may suffice to contain the directory information for these modules.

5. The size of SYS1.PL1LIB may require one additional track in order to contain the additional 210 bytes. (However, the tracks currently defined in the Storage Estimates publication may be able to contain these bytes.)

Table 1. Additional Storage Estimates for the Model 91

	Component, Module, or Program	Storage Requirements			
		Bytes of Main Storage			Tracks of Auxiliary Storage on unit types indicated (8)
		Fixed	Dynamic	Link Pack Area (1)	
Required	PFLIH	162			
	SER1	5900			
	Scan-out Area	1264			
	EREP (2)		21300		5 of 2301 29 of 2311 15 of 2314
	SYS1.LOGREC (3)				3+D/90 of 2301 15+D/28 of 2311 10+D/33 of 2314
	IEAXDS00 (4)	3520			
	PL/I (5)		210		
	PFLIH (6)	28			
	2250 (7)				
	NUCLEUS	4096			
	LINK LIBRARY			4250	1 of 2301 2 of 2311 1 of 2314
	SVC LIBRARY			7274	1 of 2301 3 of 2311 2 of 2314

Notes:

(1) The modules can be executed in the transient area as well as being in the resident area, except for those from the link library.

(2) Three 256-byte directory records are required for allocation of newly-defined partitioned data sets.

(3) D=number of devices uniquely addressable.

(4) Includes PFLIH increase of 284 bytes to invoke IEAXDS00 (Decimal Simulator).

(5) Implies object-time library only.

(6) Includes 6 bytes for a trace routine and 22 bytes for TESTRAN code, both of which are optional.

(7) 2250 System Operator's Console Support.

(8) In some cases, it may be necessary to increase the values given in this table.

Table 2, below, indicates the pertinent bit settings in the PSW for the purpose of identifying precise and imprecise interruptions.

Table 2. PSW Bits for Interruption Identification

	Type of Interruption		
	Precise	Imprecise	
	Bits 16 to 25 = 0 ILC ≠ 0	Bits 27 to 31 = 0 ILC = 0	
Type of Exception	Bits 27 to 31	Bits 16 to 25	Notes
1. Operation	00001		(1)
2. Privileged Operation	00010		(1)
3. Execute	00011		(1)
4. Protection	00100	10000 00000	(2) (4)
5. Addressing	00101	01000 00000	(5)
6. Specification	00110	00100 00000	(5)
7. Data	00111	00010 00000	(2) (3)
8. Fixed-point Overflow	01000	00001 00000	(3)
9. Fixed-point Divide	01001	00000 10000	(3)
10. Decimal Overflow	01010		(1) (2)
11. Decimal Divide	01011		(1) (2)
12. Exponent Overflow	01100	00000 01000	(3)
13. Exponent Underflow	01101	00000 00100	(3)
14. Significance	01110	00000 00010	(3)
15. Floating-point Divide	01111	00000 00001	(3)

Notes:

(1) Always precise.

(2) Precise, as simulated by the Decimal Simulator routine.

(3) Normally imprecise, but is precise if INHIBIT OVERLAP switch is in the "inhibit overlap" position.

(4) Imprecise when detected, even in the "inhibit overlap" mode.

(5) Precise or imprecise according to the conditions detected (see the Functional Characteristics publication).

INDEX

IBM Technical Newsletter

File Number	S360-32
Re: Form No.	C28-6666-0
This Newsletter No.	N28-2308
Date	January 31, 1968
Previous Newsletter Nos.	None

IBM SYSTEM/360 OPERATING SYSTEM GUIDE TO MODEL 91 SUPPORT

This technical newsletter contains amendments to the Guide to Model 91 Support publication. Replacement and/or supplemental pages to be inserted in the publication are noted below. Corrections and additions to text and/or illustrations are indicated by a vertical bar to the left of the text or illustrations and a bullet (•) to the left of an illustration caption.

Pages to Be Inserted	Pages to Be Removed
Cover, Preface	Cover, Preface
11,12	11,12
17,18	17,18
21,22,22.1	21,22
25-28	25-28
33,34	33,34

Summary of Amendments

The modifications incorporated in this newsletter apply mainly to the following sections of the publication:

Program First Level Interruption Handler
FORTLIB Macro Instruction
PL1 Macro Instruction
Using and Operating the 2250

Note: Please file this cover letter at the back of the publication. Cover letters provide a quick reference to amendments and a means of checking receipt of all amendments.

IBM Corporation, Programming Systems Publications, P.O. Box 390, Poughkeepsie, N.Y. 12602

PRINTED IN U.S.A.

IBM Technical Newsletter

File Number	S360-32
Re: Form No.	C28-6666-0
This Newsletter No.	N28-2344
Date	May 1, 1968
Previous Newsletter Nos.	N28-2308

IBM SYSTEM/360 OPERATING SYSTEM GUIDE TO MODEL 91 SUPPORT

This technical newsletter corresponds to Release 15/16 and contains amendments to the Guide to Model 91 Support publication. Replacement and/or supplemental pages to be inserted in the publication are noted below. Corrections and additions to text and/or illustrations are indicated by a vertical bar to the left of the text or illustrations and a bullet (•) to the left of an illustration caption.

Pages to Be Inserted	Pages to Be Removed
37,38	37,38

Summary of Amendments

This newsletter incorporates changes made in auxiliary storage requirements for the EREP function as it applies to the Model 91.

Note: Please file this cover letter at the back of the publication. Cover letters provide a quick reference to amendments and a means of checking receipt of all amendments.

IBM Corporation, Programming Systems Publications, P.O. Box 390, Poughkeepsie, N.Y. 12602

PRINTED IN U.S.A.

READER'S COMMENT FORM

IBM System/360 Operating System
Guide to Model 91 Support

Form C28-6666-0

- Is the material:

	Yes	No
Easy to read?	□	□
Well organized?	□	□
Complete?	□	□
Well illustrated?	□	□
Accurate?	□	□
Suitable for its intended audience?	□	□

- How did you use this publication?
  - □ As an introduction to the subject
  - □ For additional knowledge
  - Other ....

- Please check the items that describe your position:
  - □ Customer personnel
  - □ IBM personnel
  - □ Manager
  - □ Systems Analyst
  - □ Operator
  - □ Programmer
  - □ Customer Engineer
  - □ Instructor
  - □ Sales Representative
  - □ Systems Engineer
  - □ Trainee
  - Other ....

- Please check specific criticism(s), give page number(s), and explain below:
  - □ Clarification on page(s) ....
  - □ Addition on page(s) ....
  - □ Deletion on page(s) ....
  - □ Error on page(s) ....

Explanation:

- Thank your for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS PLEASE . . .

This publication is one of a series which serves as reference for systems analysts, programmers and operators of IBM systems. Your answers to the questions on the back of this form, together with 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.

Please note: Requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold Fold

FIRST CLASS
PERMIT NO. 81
POUGHKEEPSIE, N.Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY

IBM Corporation
P.O. Box 390
Poughkeepsie, N.Y. 12602

Attention: Programming Systems Publications
Department D58

Fold Fold

IBM®

International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601
[USA Only]

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
[International]