1401 SIMULATOR FOR OS/360

CORRECTION 3/69 360D-11.1.019

1401 SIMULATOR FOR OS/360 360D-11.1.019

CONTRIBUTED PROGRAM LIBRARY

CORRECTION 3/69

Program Contribution Form

Type III (IBM Employee)

IBM Corporation
Program Information Department (PID)
40 Saw Mill River Road
Hawthorne, New York 10532, U.S.A.
Attention: Program Control Desk

1 PROGRAM ORDER NUMBER (TO BE FILLED IN BY PID) 360D-11.1.019

2 SYSTEM TYPE (MACHINE) 360

3 SEARCH KEY
1401 SIMULATOR FOR
OS/360

4 AUTHORS' NAME(S) (IF DIFFERENT THEN SUBMITTER'S)
David Hutt

5 SUBMITTER'S NAME (DIRECT TECHNICAL INQUIRIES TO)
William Wilcox

6
IBM Corporation
c/o American Airlines Bldg.
Albany Post Road
Briarcliff Manor, New York 10510

7 TITLE OF PROGRAM
1401 Simulator for OS/360

8 PRIM.SUB. CODE 1.1.1

9 SECONDARY SUBJECT CODES . . . .

10 OPERATING OR MONITOR SYSTEM REQUIRED OS/360

11 TYPE OF SUBMITTAL PLEASE CHECK (√)
1 INITIAL PROGRAM ABSTRACT
2 INITIAL SUBMISSION OF PROGRAM
3 RESUBMITTAL OF UNANNOUNCED PROG.
4 X CORRECTION TO PROGRAM NUMBER

12 DATE OF SUBMITTAL 2/18/9

13 ABSTRACT (PLEASE LIMIT TO 150 WORDS. DESCRIBE PROGRAM AND PURPOSE. CLEARLY IDENTIFY MACHINE CONFIGURATION AND SOURCE LANGUAGE.)

This program allows the execution of 1401 programs on a SYSTEM/360 Model 30-75, under control of the Operating System, PCP, MFT, or MVT. No special hardware is required. 1401 features supported are advanced programming, sense switches, tapes, multiply, divide, 16K core, and all standard instructions except Select Stacker. Operator control is through the 1052 and includes tape and sense switch assignment, load from cards or tape, display and alter core, start reset, start, clear core, write tape mark, and rewind. The program is written in OS/360 Assembler Language. The minimum requirements are the same as for OS/360 and must include I/O devices used for simulation. If tape simulation is desired, tape drives must also be available. The simulator program requires approximately 70,000 bytes of core.

PLEASE ATTACH ADDITIONAL PAGES IF NECESSARY → TOTAL PAGES ATTACHED

PERMISSION TO PUBLISH: "I HEREBY GIVE IBM PERMISSION TO REPRINT, REPRODUCE AND DISTRIBUTE THIS PROGRAM TO ANYONE." PAGE 1

14 William B. Wilcox 2/18/69
SIGNATURE OF SUBMITTER AND DATE

TABLE OF CONTENTS

MAGNETIC TAPE KEY

This volume contains 3 files and 3 tape marks arranged as follows:

800bpi, 9-track

File 1 - Assembled Object Deck
300 Records, 80 Characters/Record, 40 Records/Block, 8 Blocks/File
TM

File 2 - Sample Program and Data
31 Records, 80 Characters/Record, 40 Records/Block, 1 Blocks/File
TM

File 3 - Simulator Source Program
2904 Records, 80 Characters, 40 Records/Block, 74 Blocks/File
TM

CORRECTION 3/69

USER INFORMATION

Purpose -- This program is designed to assist an installation in upgrading from a 1401 or a SYSTEM/360 Model 30 or 40 to a larger model SYSTEM/360 without the need for special hardware.

Advantages -- 1401 programs may be run without modification on a standard SYSTEM/360 under control of the Operating System. Either PCP, MFT or MVT may be used, the latter two allowing concurrent operation of multiple 1401 programs, along with regular OS/360 jobs.

Method -- The simulator has a separate subroutine for simulation of each 1401 operation, entered by an op-code selection routine, which determines the next type of instruction to be exucuted. 1401 core is simulated by 16, 000 bytes of S/360 core obtained dynamically. Supporting these primary routines are several control and operator communication routines. I/O operations are performed using two access methods: EXCP for tape operations and QSAM for reader, printer and punch.

Restrictions -- This simulator works best with I/O bound 1401 programs. With these, it can attain the maximum speed of the reader, printer, and punch, and very good speed on tapes. However, with a very CPU bound 1401 program, the simulator on a Model 50 is slightly slower than a 1401. All 7 track tapes are read and written with 556BPI density specified.

System Configuration -- The minimum system required for Simulator operation is the minimum required for operation of the Operating System. In addition, devices used by the simulator for simulating must be available, such as reader, printer, punch, or tapes. Enough core must be available to allow at least 70K for a problem program area. If tape simulation is not required, this core requirement may be reduced to 50K with the removal of the tape Buffer area. The simulator may use either 7-track or 9-track tapes.

PREPARATION FOR USE

Getting the Simulator Program Into the System

The first file of the tape contains the object deck for the simulator, which in most instances may be used without modification. It is the object deck produced as a result of assembling the source program which is included as the third file. The source program is included on the tape in case modifications are desired. In order to use the simulator program, your installation must Link Edit the object program and place the resulting load module in either SYS1. LINKLIB or a private library. A sample job to perform this function follows:

```
//jobname JOB MSGLEVEL=1

//LINKED EXEC PGM=IEWL PARM='XREF, LIST'

//SYSPRINT DD SYSOUT=A

//SYSUT1 DD UNIT=SYSDA, SPACE=(1024,(200, 20))

//SYSLMOD DD DSNAME=SYS1. LINKLIB(SIM1401), DISP=OLD

//SYSLIN DD *

                1401 Simulator object desk

/*
```

The member name used in the SYSLMOD statement is arbitrary; any member name may be substituted, provided it is not already used as a member name. Also, in the same statement, a private library may be substituted for SYS1. LINKLIB.

If the private library is not cataloged, UNIT and VOLUME parameters must also be supplied.

Job Control Language Used by the Simulator

The simulator program requires several data sets, and requires others only in certain cases. Therefore, some DD statements are required and some are optional. All required and optional statements will be considered.

1. JOB statement

There is nothing required in the JOB Statement that is peculiar to the simulator except the REGION parameter in MVT. It must specify at least 75K, unless the installation's default size is 75K or greater.

2. EXEC Statement

The Execute Statement must specify PGM=progname, where progname is the member name given to the simulator when it was Link Edited on to a library. There are no PARM field entries recognized by the simulator program.

3. Required DD Statements

A. WRITE - This is the ddname for 1401 print output. In addition to normal parameters (usually SYSOUT=A), the DCB parameters RECFM and BLKSIZE must be supplied. RECFM is FM or FBM. If FM is chosen, BLKSIZE must be 133. If FBM is chosen, BLKSIZE may be any multiple of 133 that will fit on the referenced device. If the device is a printer, BLKSIZE must be 133. Leaving the specification to the choice of the user permits an increase in efficiency in MVT, where SYSOUT data goes to intermediate direct access storage. By blocking to a full track on the direct access device, greater throughput and reduced I/O contention are achieved.

Examples:

1. //WRITE DD SYSOUT=A, DCB=(RECFM=FM, BLKSIZE=133)

2. //WRITE DD SYSOUT=A, DCB=(RECFM=FBM, BLKSIZE=7182) for full track blocking on a 2314.

B. CARDOUT - This is the ddname for the 1401 punch output. The usual operand is SYSOUT=B. No DCB parameters are required.

Example:

```
//CARDOUT   DD   SYSOUT=B
```

In installations with MVT, to avoid activating the output writer for the card punch when no punched output is desired, the DUMMY parameter may be substituted for SYSOUT=B.

Example:

```
//CARDOUT   DD   DUMMY
```

C. CARDIN - This is the ddname for the card input to the simulator, which can be the 1401 program, its data, or both.

It usually has the operand specification*.

Example:

```
//CARDIN   DD   *
```

4. Optional DD Statements

DD statements for tapes are required only if tapes are being used. They have the following specifications:

a. UNIT=device or class to specify the unit address or class of units to be assigned.

b. LABEL=(1, NL) or (1, BLP) to indicate the lack of labels.

c. DISP=OLD since all tapes are opened as input by the simulator, even if they are to be written.

d. VOLUME=SER=xxx is required because of the DISP=OLD parameter. Any serial number will do, since the tapes really aren't labelled. Since the volume serial number appears in the allocation request message, it would be helpful to use serial numbers that have meaning to the application, such as OUTPUT, INPUT 1, or SCRATCH.

e. A ddname must be supplied. Any ddname may be used, provided no two are alike for the same job step.

Example:

```
//TAPE1   DD   UNIT=2400, LABEL=(1, BLP), DISP=OLD,
        VOLUME=SER=ABC
```

```
//TAPE2  DD  UNIT=2400-2, LABEL=(1, BLP), DISP=OLD,
    VOLUME=SER=DEF
```

Examples of jobs that call the simulator:

1. 80-80 LIST

```
//JOB1 JOB MSGLEVEL=1

//STEP1 EXEC PGM=SIM1401

//WRITE  DD  SYSOUT=A, DCB=(RECFM=FM, BLKSIZE=133)

//CARDOUT  DD  DUMMY

//CARDIN  DD  *

            80-80 List Card
            (data cards to be listed)

/*
```

2. 80-80 Reproduce

```
//JOB2  JOB  MSGLEVEL=1

//STEP2  EXEC  PGM=SIM1401

//WRITE  DD  SYSOUT=A, DCB=(RECFM=FM, BLKSIZE=133)

//CARDOUT  DD  SYSOUT=B

//CARDIN  DD  *

            80-80 Reproduce card
            (data cards to be reproduced)

/*
```

3. Tape-to-print

```
//JOB3  JOB

//STEP3  EXEC  PGM=SIM1401

//WRITE  DD  SYSOUT=A, DCB=(RECFM=FM, BLKSIZE=133)
```

```
//CARDOUT  DD  DUMMY

//TAPE  DD  UNIT=2400-2, LABEL=(1, BLP), DISP=OLD,
      VOLUME=SER=XYZ

//CARDIN  DD  *

            1401 program deck

/*
```

COMMUNICATING WITH THE SYSTEM

When the scheduler has completed the allocation for the step and gives control to the simulator, the simulator performs some preliminary initialization, then issues a WTOR with the message "READY". It is then ready for operator commands. At this time the following initial conditions exist:

1. All sense switches are off.
2. No tape assignments are made.
3. 1401 core has not been cleared.

The operator commands are in the form of replies to the WTORs issued by the simulator and are in standard reply form. One exception to this is that letters in the reply message may be in either upper or lower case.

Following is a list of the possible operator entries. Only the message portion of the reply is given.

1. LDC - Load Program from Card Reader - Performs the same function as the LOAD button on the 1402.

2. LDT - Load Program from Tape - Performs the same function as the TAPE LOAD button on the 1401.

3. CLR - Clear all 1401 core. Clears all simulated 1401 core.

4. SRS - Start Reset - Performs the same function as START RESET on the 1401.

5. a. STT - Start - Performs the same function as START on the 1401.
   b. STTnnnnn - Start - Starts 1401 execution at the 1401 address nnnnn. Leading zeroes may be suppressed. Example - STT3107 will start 1401 execution at location 3107.

6. DISnnnoo - Display core - 100 characters of 1401 core, with their word marks, are display on the printer, starting with location nnnoo. The address must end in 00, but leading zeroes may be suppressed.

   Example - DIS3700 will display locations 3700-3799

PARM.60

7. SSSxxx... - Set sense switches - The sense switches represented by xxx will be set and all others will be cleared.

   Example - SSSAG will set sense switches A and G, and clear all others. SSS without any following letters will clear all sense switches.

PARM.60

8. TASnnn, m, nnn, m... - Tape Assignment - This entry assigns 1401 drive m to actual drive nnn. nnn must have been requested by the scheduler. Any previous assignments are cancelled.

   Example - TAS180, 4 will assign tape unit 180 as 1401 drive 4.
   TAS180, 2, 282, 5 will assign tape unit 180 as 1401 drive 2 and tape unit 282 as 1401 drive 5.

9. ALTnnnnn, c(M) Alter Core - Character c will be placed in core location nnnnn. Leading zeroes may be suppressed. If M is specified, a word mark will be included.

   Example - ALT 459, Q - A "Q" will be placed in location 459 without a word mark.

   ALT7312, 8M will place an 8 with a word mark in location 7312.

10. WTMm - Write Tape Mark - Writes a tape mark on the 1401 tape drive m.

11. RWDm - Rewind - Rewinds 1401 drive m.

12. TRM - Terminate the simulator - This entry causes the Simulator to terminate its operation and return to the control program.

The simulator should reply OK to all operator entries except LDC, LDT, and STT. LDC and LDT should not cause any replies. Neither should STT, unless no program has been loaded by LDC or LDT, in which case an error message is sent to the operator.

The operator enters LDC or LDT to load and execute a 1401 program. However, the program to be loaded may require tape and/or sense switch assignments, which mean a TAS and/or SSS entry before loading the 1401 program. Also, if the 1401 program does not have a clear storage routine in it, the operator may wish to enter CLR to clear all 1401 core.

When the 1401 program halts, the address of the halt is displayed on the typewriter. The operator may then make further entries. To terminate the simulator, the operator replies TRM.

The following messages may be sent to the operator by the simulator:

1. 00 READY jobname - The simulator has loaded and is ready to accept further operator entries. [illegible] r 00,'LDC'
2. 00 OK jobname - This is the normal response to any entry except LDC, LDT and STT. The simulator is then ready for further inputs.
3. HALT - This indicates that the 1401 program has halted and indicates the I address. If it is a 7-character halt, the A and B addresses are also displayed. The OK message follows immediately.
4. 00 ILLEGAL ENTRY - The operator has entered a command that the simulator does not recognize. It is then ready for another command.
5. READ TRIED AFTER LAST CARD - The 1401 program attempted to read a card, but no card was available. The simulator terminates with a User code of 0001.
6. 00 CANNOT START, NO PGM LOADED - A START command was given, but no 1401 program was loaded. The simulator is ready for further entries.
7. UNDEFINED TAPE followed by Panel Message - A 1401 tape operation was encountered but the operator has not assigned that 1401 drive number to an actual device through a TAS entry.
8. ILLEGAL OP CODE followed by Panel Message - The simulator found either an operation code or a D character that it could not recognize.
9. ILLEGAL LENGTH followed by Panel Message - The simulator encountered a 1401 instruction whose length is inconsistant with its operation code.
10. Panel Message - A Message following messages 7, 8, and 9 giving information about the instruction in error, such as its op code location, Length, and the instruction itself.

SAMPLE PROGRAM

The sample 1401 program that is included will read cards from the card reader and print them on the printer. If sense switch B is on, the program will double space between print lines. The sample deck consists of the 1401 program followed by 25 data cards labelled "CARD nn of 25 CARDS".

The format of the deck required to simulate this program is:

```
//LIST8080 JOB MSGLEVEL=1

//STEP1 EXEC PGM=SIM1401

//WRITE DD SYSOUT=A

//CARDOUT DD DUMMY

//CARDIN DD *

                Sample Deck, including data cards

/*
```

Also, if the simulator is in a private library, a JOBLIB card is required immediately after the JOB card.

Operating instructions for sample program job:

1. Schedule the job normally and enter it into the system.
2. The operator will receive the message 00 READY jobname.
3. The operator enters REPLY 00,'SSSA' to set sense switch A.
4. The simulator replies 00 OK jobname.
5. The operator enters REPLY 00,'LDC' to load the 1401 program.
6. The data cards should be read and printed with single spacing.

Rerun the job as before, but at step 3, enter REPLY 00, SSSAB to set sense switches A and B. The cards should read as before but should print with double spacing.

DETAILED PROGRAM DESCRIPTION

The 1401 Simulator can be divided logically into four parts: simulator core, I/O areas, instruction processing routines, and control routines. Simulated core consists of 16,000 bytes of core, obtained through a Getmain, each byte representing one character of 1401 core. The bits of each byte are used as follows:

Bit	Use
0	unused
1	word mark
2	B
3	A
4	8
5	4
6	2
7	1

Information in simulated core is maintained exactly as it would be in actual 1401 core. An area of about 25K is obtained through a Getmain to be used for a tape I/O area. Tape cannot be read directly into or written directly from simulated core. Tape is written so that it will exactly match that written by a 1401. Therefore, the data is moved to the I/O area, word marks are removed, blanks are replaced by A-bits if the writing is to be in BCD mode, and finally the tape is written without translation or data conversion. The reverse process occurs in reading tape. In doing this, a 7-track tape is in exact 1401 format. Also, the simulator need not care whether a tape is 7 or 9-track. The same process is used in reading and writing tape with word marks, with the added job of forming and combining the word separator characters, which causes the I/O area to be larger than the equivalent simulated core area. Card reading, card punching, and printing are handled in a similar manner, except for the EBCDIC translation.

The control routines consist of I/O subroutines, operator communication routines and 1401 operation code selection routines. The operation code selection routine examines the operation code of the next 1401 instruction and branches to the proper instruction processing routine. There is one instruction processing routine for each 1401 instruction. Each routine simulates the execution of the 1401 instruction that it represents, then returns to the operation code selection routine. Most instruction routines that process multiple characters work with one character at a time. This is probably the slowest method of processing, but it was the simplest.

Certain registers have particular uses in the simulator.

They are:

Reg.	Use
15	Base
14	Base
13	Base
12	B Address
11	A Address
10	I Address
9	Instruction Length
8	Return for Branch & Link
7	Base of simulated core

The other registers are used within individual routines, and do not convey useful information between routines.

This simulator is a revision of a previously written version that runs stand-alone. The base registers used are incompatable with the register usage required for OS/360. Therefore, before any OS/360 macros, registers 13-15 are saved, then restored after the macro execution is complete.

The QSAM access method is used for card and print operations, but EXCP is used for tape operations for better control of modes of operation.

The amount of core required for the simulator may be reduced somewhat if the user determines that he doesn't require 25K for a tape I/O area. The area required is equal to the largest core size to be read or written. If the tape is to be read or written with word marks, the maximum number of word marks must be added to the size. The resulting size is the amount of core to be obtained through the Getmain macro.

Sample Problem Input/Output Listing

```
,006015,019026,030,034041,045,053,0570571026                        0001
L068112,102106,113/101099/I99.D27A07002B<027B0010270B0261,001/00111310 0002
,008015,022029,036040,047054,061068,072/061039            ,0010011040 0003
,001F18367A1M0802802B362B              L025357,337339,344345,3523531040 0004
B339F339JF1.3330                       L016373,362367,369373,04004010400005
                                       /333080                          0006
      CARD 01 OF 25 CARDS
      CARD 02 OF 25 CARDS
      CARD 03 OF 25 CARDS
      CARD 04 CF 25 CARDS
      CARD 05 OF 25 CARDS
      CARD 06 OF 25 CARDS
      CARD 07 OF 25 CARDS
      CARD 08 OF 25 CARDS
      CARD 09 OF 25 CARDS
      CARD 10 OF 25 CARDS
      CARD 11 OF 25 CARDS
      CARD 12 OF 25 CARDS
      CARD 13 OF 25 CARDS
      CARD 14 OF 25 CARDS
      CARD 15 OF 25 CARDS
      CARD 16 OF 25 CARDS
      CARD 17 OF 25 CARDS
      CARD 18 OF 25 CARDS
      CARD 19 OF 25 CARDS
      CARD 20 OF 25 CARDS
      CARD 21 OF 25 CARDS
      CARD 22 CF 25 CARDS
      CARD 23 OF 25 CARDS
      CARD 24 OF 25 CARDS
      CARD 25 OF 25 CARDS
```

[illegible]

```
//     EXEC LINKEXEC,P1=SIM14Ø1,PARM.GO='SSSACG,
//      TAS180,1,181,2'


//GO.SYSUDUMP   DD   SYSOUT=A
//GO.WRITE    DD   SYSOUT=A,DCB=(RECFM=FM,BLKSIZE=133)
//GO.CARDOUT   DD   DUMMY  (unless you want the punch)
//GO.TAPE1   DD   UNIT=18Ø,LABEL=(,NL),DISP=OLD,VOL SER=INP
//GO.TAPE2
           {
//GO.CARDIN   DD   *,DCB=BLKSIZE=8Ø
```