Cowles

SIR

IBM 650 Soap Interpretive Routine

Programmer's Guide

International Business Machines Corporation
590 Madison Avenue, New York 22, N. Y.

Bruce Oldfield
New York Scientific Computing Center

William Hemmerle
Washington Scientific Computing Center

March 21, 1956

TABLE OF CONTENTS

Page

APPENDIX

Page

INTRODUCTION

The purpose of this report is to describe coding procedures and machine operating rules for "SIR", the Washington Soap Interpretive Routine for the IBM Type 650 Magnetic Drum Calculator.

The IBM Symbolic Optimal Assembly Program (SOAP) greatly simplifies programming for the 650 calculator. SIR is a relocatable library program which is used with the SOAP system to handle floating decimal* interpretive operations with ease and facility.

The main features of this system are:

1) Simplicity. SIR is an exceedingly simple system to learn. Very little information is needed in addition to that found in "SOAP Programmers Guide."

2) Ease and Speed of Programming. All interpretive and non-interpretive instructions are written in a symbolic system where most location and instruction addresses may be left blank and all operations can be written alphabetically.

3) Versatility. SIR contains important built-in features which greatly increase the usefulness of the system. These features include a fixed decimal to floating decimal routine, a floating decimal to fixed decimal routine, a flexible tracing routine for interpretive instructions and 6 interpretive floating point functions.

4) Efficiency. All interpretive instructions have been tightly optimized by hand. All non-interpretive instructions will be optimized by Soap. The fact that a large percentage of the instructions in any program are executed non-interpretively makes each program very efficient.

5) Compactness. SIR has been divided into 9 functional sections such that only the sections needed for a particular problem need be assembled.

6) Compatability. SIR is completely compatable with the SOAP system which means that no additional boards, card formats or operational procedures need be used.

* The floating point add, multiply and divide routines were written by Mrs. Leslie Ayres, Washington Scientific Computing Center.

DEFINITION OF TERMS

K = Psuedo accumulator used for floating point interpretive operations.

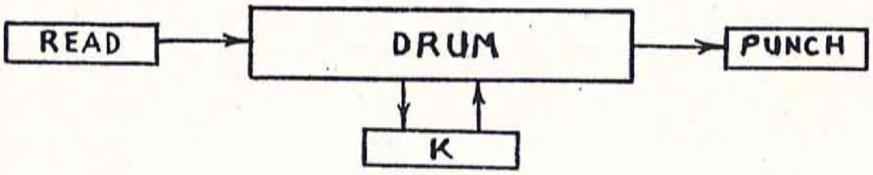
(K) = Contents of the psuedo accumulator K. A number such as 5.432101234 would be stored in K as 0000000050, 5432101234.

A = Any absolute or symbolic drum location.

(A) = Contents of drum location A. A number such as 5.432101234 would be stored in drum location A as 5054321012.

Δ = Amount of translation specified on a Library Translation Card.

When executing interpretive instructions, the 650 has the following schematic form:



It is important to understand that while in the interpretive mode the single psuedo accumulator K replaces the upper accumulator, the lower accumulator and the distributor. The location of K on the drum is 0119 + Δ for the characteristic and 0104 + Δ for the mantissa. The (K) will remain undisturbed after leaving the interpretive mode and thus, if desired, is available when the interpretive mode is next entered.

INTERPRETIVE OPERATIONS

The following table is a complete summary of all interpretive operation codes and the operation resulting from the use of each code.

650 SIR OPERATION CODES

Symbolic Op-Code	Numeric Op-Code	Resulting Operation
NOP	00	No-Operation
HLT	01	Stop
OUT	03	Exit
SQR	04	Sqr (K) to K
SIN	05	Sin (K) to K
COS	06	Cos (K) to K
LGN	07	Ln (K) to K
EXP	08	$e^{(K)}$ to K
ART	09	Tan^{-1}(K) to K
AUP	10	(K) + (A) to K
SUP	11	(K) − (A) to K
DIV	14	(K) ÷ (A) to K
ALO	15	(K) + (A) to K
SLO	16	(K) − (A) to K
AAB	17	(K) + $\|$(A)$\|$ to K
SAB	18	(K) − $\|$(A)$\|$ to K
MPY	19	(K) x (A) to K
STL	20	(K) to A
STU	21	(K) to A
STD	24	(K) to A
SRT	30	(K) x 10^{-b} to K
SLT	35	(K) x 10^{b} to K
NZU	44	Test (K) = 0
NZA	45	Test (K) = 0
BMI	46	Test (K) $<$ 0
RAU	60	(A) to K
RSU	61	−(A) to K
DVR	64	(K) ÷ (A) to K
RAL	65	(A) to K
RSL	66	−(A) to K
RAB	67	$\|$(A)$\|$ to K
RSB	68	−$\|$(A)$\|$ to K
LDD	69	(A) to K
RDS	70	Read a Card
PCH	71	Punch a Card

TABLE 1

The following are important features of the interpretive operations shown in Table 1:

1) THE NUMERIC OPERATION CODES 03 to 09 must be used, instead of the equivalent symbolic operations, unless a modified Soap deck is used for assembly. (The new Soap deck is identified 494.)

2) Wherever possible, each normal 650 operation code is analogous to an interpretive operation code. This enables a programmer to consider the psuedo accumulator as either the upper accumulator, the lower accumulator or the distributor of the 650 calculator. Consequently, several different operation codes will produce the same operation. For example: RAL, RAU and LDD all result in (A) to K, while STL, STU and STD all result in (K) to A.

3) A data or instruction address of 800X cannot be given while in the interpretive mode.

4) Address modificiation CANNOT be done while in the interpretive mode.

5) Any invalid 650 data or instruction address in an interpretive instruction will result in an error stop. For example, the instruction to exit from the interpretive mode should be given as OUT 0000 XXXX and must not be given with an invalid data address.

6) The READ instruction given interpretively MUST BE USED WITH CARE since a load card will cause the machine to obtain its next instruction from the location specified by the data address.

7) DO NOT leave both the data and instruction address blank on any interpretive or non-interpretive instruction.

EXAMPLE 1

Consider the following sequence of instructions to be interpreted. These instructions have been coded symbolically with the location and instruction address left blank.

COL.	41	42	43 44 —— 47	48 49,50	51 52 —— 55	56 57 —— 60	61 —— 72
	TYPE	SIGN	LOCATION	OPERATION CODE	DATA ADDRESS	INSTRUCTION ADDRESS	REMARKS
			F IRST	R AL	X 1		(X1) TO K
				A LŌ	X 2		(X2) + (K) TO K
				M PY	X 3		(X3) × (K) TO K
				S LT	0003		(K) × 10^3 TO K
				S TL	X 4		(K) TO X4

The contents of the psuedo accumulator and the contents of the data location after the execution of each instruction is shown below.

Oper.	Data Address	Contents of Data Location	Psuedo Accumulator Characteristic	Mantissa
R AL	X 1	50 42631746	00000000 50	4263174600
A LO	X 2	47 81176338	50	4271292234
M PY	X 3	49 20000000 -	49	8542584468 -
S LT	0003		52	8542584468 -
S TL	X 4	52 85425844 -	52	8542584468 -

FLOAT ROUTINE

It is very important to be able to change numbers easily from a fixed decimal to a floating decimal form. A FLOAT routine is included in SIR to provide this flexibility.

Define: PP = The number of positions to the left of the decimal point in a ten digit field. For example, the number 0012.936421 would have PP = 04.

FWA = The location of the first word to FLOAT or to FIX.

LWA = The location of the last word to FLOAT or to FIX.

Numbers to be floated must be in consecutive storage locations with aligned decimal points. As the numbers are floated, they are returned to their original storage locations.

COL. 41	42	43 44 — 47	48 49,50	51 52 — 55	56 57 — 60	61 — 72
TYPE	SIGN	LOCATION	OPERATION CODE	DATA ADDRESS	INSTRUCTION ADDRESS	REMARKS
		FLŌAT	RAL	CWD		
			LDD	NEXT	FLŌAT	Basic Linkage
		CWD	PP	FWA	LWA	Control WORD
		NEXT				Next Instruction

These instructions will float all of the numbers from FWA to LWA locations and then return by basic linkage to execute the NEXT instruction.

To Float one number,	LWA = FWA
To Float N numbers,	LWA = (FWA + N - 1)

The FLOAT ROUTINE cannot be executed while in the interpretive mode.

FIX ROUTINE

It is very important to be able to change numbers easily from a floating decimal to a fixed decimal form. A FIX routine is included in SIR to provide this flexibility.

Numbers to be FIXED must be in consecutive storage locations. As the numbers are fixed, they are returned to their original storage locations with decimal points aligned at PP.

COL. 41	42	43 44 — 47	48 49,50	51 52 — 55	56 57 — 60	61 — 72
TYPE	SIGN	LOCATION	OPERATION CODE	DATA ADDRESS	INSTRUCTION ADDRESS	REMARKS
		FIX	RAL	CWD		
			LDD	NEXT	FIX	Basic Linkage
		CWD	PP	FWA	LWA	Control Word
		NEXT				Next Instruction

These instructions will convert the floating decimal numbers stored in FWA to LWA locations to fixed decimal numbers and then return by basic linkage to execute the NEXT instruction.

Notice that numbers can be altered or lost due to FIXING them. If the floating decimal number (54)12345678 were fixed with PP = 02, the result would be 45.678000, the digits 123 being lost.

The FIX routine cannot be executed while in the interpretive mode.

ENTER SIR - EXIT SIR

The Interpretive Routine is ENTERED by giving a single instruction.

COL.	41	42	43	44 —— 47	48	49,50	51	52 —— 55	56	57 —— 60	61 —— 72
	TYPE	SIGN	LOCATION		OPERATION CODE		DATA ADDRESS		INSTRUCTION ADDRESS		REMARKS
					R	AL	F	IRST	I	NT	ENTER SIR

"FIRST" is the location of the first instruction to be executed interpretively and can be left blank if the first instruction immediately follows the enter instruction.

Once the interpretive routine has been entered, all further instructions will be interpreted until an OUT instruction is given.

COL.	41	42	43	44 —— 47	48	49,50	51	52 —— 55	56	57 —— 60	61 —— 72
	TYPE	SIGN	LOCATION		OPERATION CODE		DATA ADDRESS		INSTRUCTION ADDRESS		REMARKS
					O	UT		0000	N	EXT	EXIT From SIR

"NEXT" is the location of the next instruction to be executed non-interpretively and can be left blank if this instruction immediately follows the OUT instruction.

LIBRARY SYMBOL CARDS

When entering the Interpretive Routine, the Float Routine and the Fix Routine, a reference is made to a symbolic address. Each symbolic address is assigned an absolute address by use of a type 9 library symbol card. The library symbol cards are shown below and are cards 2, 3 and 4 of the SIR deck.

COL.	41	42	43	44 —— 47	48	49,50	51	52 —— 55	56	57 —— 60	61 —— 72
	TYPE	SIGN	LOCATION		OPERATION CODE		DATA ADDRESS		INSTRUCTION ADDRESS		REMARKS
	9						I	NT		0280	SIR symbolic Entry
	9						F	LŌAT		0229	FLOAT Symbolic Entry
	9						F	IX		0200	FIX Symbolic Entry

INTERPRETIVE TRACE ROUTINE

A flexible tracing routine is incorporated in SIR and can be of great assistance in the debugging of any program. This routine will trace interpretive instructions only. The punched card output of the trace routine contains the following information:

WORD	
1	xxYYYYxxxx where YYYY is the location of the instruction.
2	The interpreted instruction.
3	Characteristic of (K) before the operation.
4	Mantissa of (K) before the operation.
5	Contents of the data address of the interpreted instruction.
6	Characteristic of (K) after the operation.
7	Mantissa of (K) after the operation.

This routine has 3 tracing modes available: 1) Trace ALL SIR instructions, 2) Trace NEGATIVE SIR instructions, 3) Trace NO SIR instructions. The particular mode of tracing is controlled by Trace Control Cards which are described in the APPENDIX.

The ability to trace a particular interpretive instruction by making that instruction negative is sometimes referred to as "Break Point Operation" and is an exceedingly useful debugging technique.

Interpretive branch instructions will not be traced when the branch is taken. The first instruction when entering the interpretive mode will not be traced. DO NOT put a negative sign on an interpretive instruction.that will be modified unless this negative condition is programmed for in advance.

FUNCTIONS

Table 1 shows the 6 functions (Square Root, Sine, Cosine, Logarithm, Exponential and Arctangent) which are included in SIR. Each function is assembled only when necessary, but once included it will be executed by an interpretive instruction as indicated in the operation codes. A discussion of the limitations and a list of program stops for these subroutines are given in the APPENDIX.

EXAMPLE 2

The following program illustrates the use of the interpretive routine.

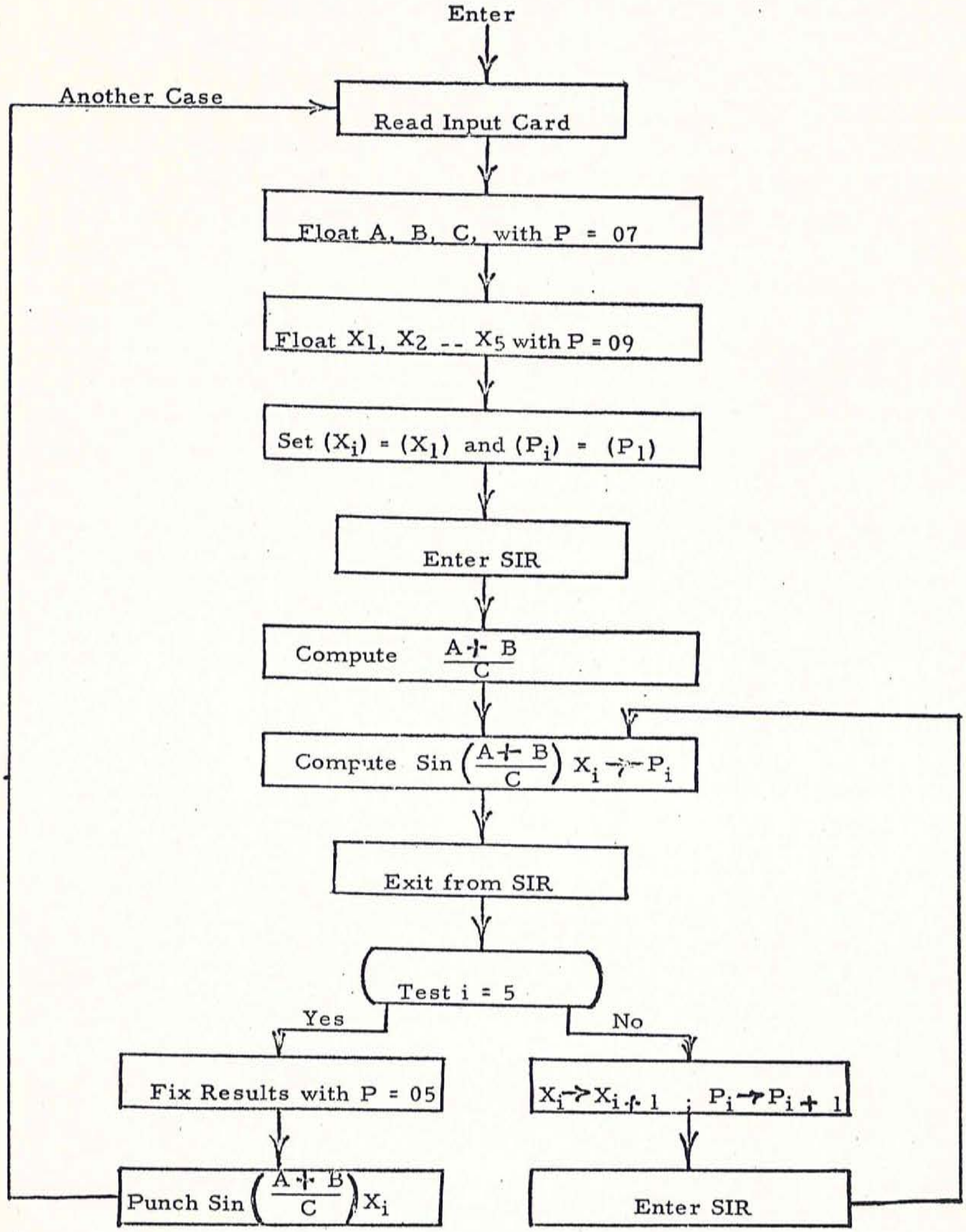
Compute $\text{Sin}\left(\frac{A+B}{C}\right)X_i$ for $i = 1, 2, 3, 4, 5$

The input is in a fixed point form, the computations are done interpretively in floating point and the results are punched in a fixed point form. The Sine routine must be relocated by SOAP along with the main library program.

The data is read into locations R 1 to R 8 with the position of the decimal point indicated below.

Number	Location	Decimal Form
A	R 1	000000X. XXX
B	R 2	000000X. XXX
C	R 3	0000XXX. XXX
X_1	R 4	0000000XX. X
X_2	R 5	0000000XX. X
X_3	R 6	0000000XX. X
X_4	R 7	0000000XX. X
X_5	R 8	0000000XX. X

The results are punched in a fixed form 00000. XXXXX


Enter
Another Case
Read Input Card
Float A, B, C, with P = 07
Float X_1, X_2 -- X_5 with P = 09
Set $(X_i) = (X_1)$ and $(P_i) = (P_1)$
Enter SIR
Compute $\frac{A + B}{C}$
Compute Sin $\left(\frac{A + B}{C}\right) X_i \rightarrow P_i$
Exit from SIR
Test i = 5
Yes
No
Fix Results with P = 05
$X_i \rightarrow X_{i+1}$; $P_i \rightarrow P_{i+1}$
Punch Sin $\left(\frac{A + B}{C}\right) X_i$
Enter SIR

Flow Chart for Example II

CODING FORM

650 S.O.A.P.

COL. 41	42	43 44 — 47	48 49,50	51 52 — 55	56 57 — 60	61 — 72			
TYPE	SIGN	LOCATION	OPER.	DATA ADDRESS	INSTRUCTION ADDRESS	REMARKS	ACCUMULATOR		DISTRIBUTOR
							UPPER	LOWER	
3				R 1951	1960	Reserve RDS.			
3				P 1027	1036	Reserve PCH.			
		START	RDS	R 0001		Read Data R1 to R8			
			RAL	C 1		Float R1 to R3 (A, C, C)			
			LDD		FLŌAT				
			RAL	C 2		Float R4 to R8 (Xi)			
			LDD		FLŌAT				
			LDD	C 3					
			STD	MŌD1		Initialize the Loop			
			LDD	C 4					
			STD	MŌD2					
			RAL		INT	Enter SIR			
			RAL	R 0001		[Interpreted Instructions]			
			ALŌ	R 0002		For			
			DIV	R 0003		$\frac{A\pm B}{C} \rightarrow T1$			
			STL	T 1	MŌD1				
		MŌD1	RAL	R 0004	K 1	To be Modified			
		K1	MPY	T 1		[Interpreted Instructions]			
			SIN	0000	MŌD2	$\sin(\frac{A\pm B}{C})X_i \rightarrow F_i$			
		MŌD2	STL	P 0001	K 2	To be Modified			
		K2	ŌUT	0000	TEST				
		C1	07	R 0001	R 0003				
		C2	09	R 0004	R 0008	Constants			
		C3	RAL	R 0004	K 1				
		C4	STL	P 0001	K 2				

Compute $\sin(\frac{A\pm E}{C})X_i$

Page 1 of 2

CODING FORM

650 S.O.A.P.

COL.	41	42	43 44 —— 47	48 49,50	51 52 —— 55	56 57 —— 60	61 —— 72			
	TYPE	SIGN	LOCATION	OPER.	DATA ADDRESS	INSTRUCTION ADDRESS	REMARKS	ACCUMULATOR		DISTRIBUTOR
								UPPER	LOWER	
			TEST	RAL	MOD1					
				SLO	C5		Test $i = 5$			
				EMI		EXIT				
				ALO	C6					
				STL	MOD1					
				RAL	MOD2		steps i to $i+1$			
				ALO	C7					
				STL	MOD2					
				RAL	MOD1	INT	Enter SIR			
			EXIT	RAL	C8		Fix P1 to P5			
				LDD		FIX	Form ooxxx.xxxxx			
				PCH	P0001	START	Punch Fixed Results			
			C5	RAL	R0008	K1				
			C6	RAL	R0009	K1	CONSTANTS			
			C7	00	0001	0000				
			C8	05	P0001	P0005				

APPENDIX

SECTIONS

SIR has been separated into 9 sections on a functional basis. Only the sections of SIR required for a particular problem need be assembled. The idea of separate functional sections is most important because it minimizes the assembly time and the storage locations required for each problem. The instruction cards for each section are identified by name and by sequence number. This identification is in columns 61 to 72, columns ordinarily used for REMARKS. The following table gives the identification, number of cards and number of 650 storage locations used by each section.

Identification	Number of Cards	Number of 650 Loc.
MAIN SIR	185	184
TRACE	26	31
FLOAT	38	38
FIX	46	45
SQR	93	97
SIN. -COS.	83	88
LGN	105	111
EXP	97	105
ARC TAN	66	69

TRANSLATION

A single Type 7 Library Translation Card is placed in front of the main library program when assembling. This card should have 0799 as its LWA. An EVEN △ must be used in order to preserve optimization.

SIR can be split into parts for relocation if necessary. However, the operation table occupies locations 0 to 71 and CANNOT be split.

Operation	Estimated Time	
Float 1 Word	38.6 ms	Not Interpreted
Float 100 Words	3.0 sec.	" "
Fix 1 Word	41.6 ms	" "
Fix 100 Words	3.4 sec.	" "
Interpretive Loop	18.5 ms	Without Trace
Interpretive Loop	117.5 ms	With Trace, Includes Punch Time

Interpretive operations referring to the LOWER ACCUMULATOR will result in slightly more optimum programming than operations referring to the upper accumulator or the distributor.