A COMPILER FOR THE BAMA-BELL FLOATING POINT INTERPRETIVE SYSTEM

By

William J. Gray

A Thesis

Submitted in partial fulfillment of the requirements for the degree of Master of Arts in the Department of Mathematics in the Graduate School of the University of Alabama

UNIVERSITY, ALABAMA

1962

ACKNOWLEDGMENTS

The author expresses appreciation to Professor C. L. Seebeck, Jr. for his encouragement and many helpful suggestions.

TABLE OF CONTENTS

CHAPTER I

INTRODUCTION

The Bama-Bell Floating Point Interpretive System was first completed by the author in September, 1961. Since that time, the system has been used extensively at the University of Alabama Computer Center, both for research and for instructional purposes. The second version of Bama-Bell, called Bama-Bell II-80, was completed in July, 1962 and is now operational. The Bama-Bell II-80 language consists of 74 instructions and for this reason it is the largest Bell Interpretive System in existence.

The compiler for the Bama-Bell System interprets any program written in the Bama-Bell interpretive language and produces a basic machine language program. The machine language program, of course, operates considerably faster than the interpretive program.

So far as is known, the present compiler is the first of its type ever written.

CHAPTER II

TERMINOLOGY

A manual which fully explains the use of the Bama-Bell Interpretive System is included as an appendix. It will be assumed that the reader is familiar with this manual.[1]

A program written in Bama-Bell interpretive language is defined to be a <u>source program.</u> A basic machine language program produced by the Bama-Bell Compiler is an <u>object program.</u> The <u>Compiler Library</u> is a system of machine language sub-routines designed to function in conjunction with the object program to perform the operations called for in the source program.

The following notations will be used for brevity:

I P R - Instruction processing routine.

N B A R - Minimum latency routine.

---

[1] See Appendix.

CHAPTER III

TECHNICAL SUMMARY

The Compiler for the Bama-Bell Floating Point Interpretive System is a fixed library compiler. The compiler and library were written in X-6 relative coding for the Univac Solid-State 80.

The Compiler consists of approximately 2,000 instructions, and the Library consists of somewhat less than 1,600 instructions.

Compiler output is punched cards. Compilation speed is about 33 input instructions per minute, or approximately 150 output cards per minute.

The Compiler is applicable to mathematical and statistical problems.

CHAPTER IV

THE COMPILER

The Compiler consists of an initialization routine, the interpretive routine, the minimum latency routine, the punch routine, and a system of routines which produce the portion of the object program which correspond to a specific instruction in the source program.

The initialization program sets the punch routine to its initial conditions, and reads a header card provided by the programmer, and which contains the identification which is to be punched on each card of the object program. Control then proceeds to the interpretive routine.

The interpretive routine reads the next cards and determines if it is an instruction card, a data card, or a sentinel card; in the last two cases control is transferred to the appropriate section for processing a data card or a sentinel card; otherwise, the A, B, card L addresses of the instruction are extracted and stored for future reference. The operation code is then determined and control proceeds to an instruction processing routine.

The instruction processing routines work in conjunction with the minimum latency routine and punch routine to produce the object program. The IPR compiles a small machine language routine which corresponds to a Bama-Bell instruction. The IPR sequence begins at the L address assigned to the Bama-Bell instruction by the programmer. Furthermore, all locations used by the Bama-Bell programmer are used as is by the IPR. All other addresses are determined so as to provide the best possible latency. Address assignment will be discussed below. When the IPR finishes an instruction, control is returned to the interpretive routine.

NBAR, the minimum latency routine, is always entered from an IPR. The IPR provides NBAR with the next best address: NBAR, in turn, provides IPR with the next best available address. (The next best address is unique modulo 200.)

NBAR maintains a 400 word memory chart from which it may determine the availability of all memory locations in the Computer. NBAR seeks the next best address, modulo 200, and if the next best address is not available, the NBAR seeks the next best address plus one modulo 200. This process continues until the next best available address is determined. Then NBAR returns control to the IPR from which it was entered.

Upon demand by an IPR, the punch routine punches out an instruction or data word with the appropriate identification and machine location. The punch routine maintains a card count which is also punched out with each card.

CHAPTER V

THE LIBRARY

The Bama-Bell Compiler is a fixed library compiler, which means that the larger routines needed to execute a mathematical program are not compiled as part of the object program. These routines are assembled into one system called the Library. NBAR reserves the location used by the Library.

The following is a brief description of the library routine.

The floating point routine performs floating point arithmetic operations upon demand by the object program. The mathematical routines which may be called upon by the object program are the square root, exponential, log, sine-cosine, and arctangent routines. The Library also contains all input-output routines, the loop, and the loop-loop routines.

The locator routine is used in conjunction with the instruction modification routines. The locator routine determines the machine location of instructions to be modified.

The initialization routine, which must be entered before the execution of every object program, sets all library controls to their initial conditions and initializes input-output routines.

The program load routine loads the object program at 600 cards per minute.

It is hoped that the Compiler will contribute to the scientific and statistical research programs at the University of Alabama and elsewhere.

APPENDIX

BAMA-BELL II

Floating Point Mathematical Interpretative System for USS 80 SYSTEM

by

William J. Gray

BAMA-BELL II MANUAL

Foreword

Bama-Bell II is a reissue of Bama-Bell for the USS 80 Step system. There are two versions of Bama-Bell II: Bama-Bell II-80 for 5000 word memory, and Bama-Bell II-80S for 2600 word memory.

Bama-Bell II was programmed to increase the speed of computation and to simplify program tracing. It is believed that the programmer will find Bama-Bell II to be considerably faster than the original Bama-Bell. Several other changes have been made, and instructions have been added.

Introduction

Bama-Bell is a three address floating point interpretative system designed to simplify mathematical and scientific programming. Bama-Bell provides 999 locations for the programmer. The programmer may address the locations as 001-999. In addition the programmer may address the answer register as 000.

Instructions

A Bama-Bell instruction consists of 10 digits. The first digit of an instruction is called the operation code, (op. code). A valid op. code may be 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, I, R, or Z. Digits 2, 3, and 4 of an instruction form the A address of the instruction. Digits 5, 6, and 7 are the B address and digits 8, 9, and 10 are the C address.

op. code	A	B	C
x	aaa	bbb	ccc

If the op. code is I, R, or Z, the A address of the instruction will be part of the op. code. If the op. code is 0, the first digit of the A address of the instruction is part of the op. code. Depending on the type of instruction, A, B, and C addresses may or may not be a location in memory. If A, B, or C is a memory location, the contents of A, B, or C will be denoted $\bar{A}$, $\bar{B}$, $\bar{C}$, respectively. An instruction may be positive or negative.

Data

Bama-Bell II operates in the XS-50 floating point mode. A word of data consists of the XS-50 exponent (2 digits) and the XS-50 mantissa (8 digits). The format is as follows:

$$AAaaaaaaaa = a.aaaaaaa\ 10^{AA-50}$$

$$00 \leq AA \leq 99$$

For example in XS-50:

$$1 = 5010000000$$
$$2 = 5020000000$$
$$10 = 5110000000$$
$$200 = 5220000000$$
$$4000 = 5340000000$$
$$.6000 = 4960000000$$
$$.07965432 = 4878654320$$
$$7.0 \times 10^{-20} = 3070000000$$
$$7.0 \times 10^{20} = 7070000000$$

Numbers represented in the above form will be said to be normalized.

A word of data may be positive or negative.

Zero should always be represented by 0000000000

Arithemetic Instructions

1 A B C (Add)	The sum $\bar{A} + \bar{B}$ is stored in 000 and C

2 A B C (Subtract)	The difference $\bar{A} - \bar{B}$ is placed in 000 and C.
3 A B C (Multiply)	The product $\bar{A} \times \bar{B}$ is placed in 000 and C.
4. A B C (Divide)	The quotient $\bar{A} / \bar{B}$ is placed in 000 and C.
8 A 000 C (Convert)	$-\bar{A}$ is placed in 000 and C.
R 776 B C (Absolute)	$\lvert \bar{B} \rvert$ is placed in 000 and C.

Data Conversion Instructions

These instructions are not available for Bama-Bell II-80S.

0 1nn B C (Fix)

Convert $\bar{B}$ to fixed point with the decimal following the nnth digit, where the digits are counted left to right. Result to C and 000.

$00 \leq nn \leq 10.$

Example: Location 100 contains 50 250 000 000

Instruction	Result in ccc
0 101 100 ccc	25 0000 0000
0 102 100 ccc	02 5000 0000
0 103 100 ccc	00 2500 0000
0 109 100 ccc	00 0000 0025

0 2nn B C (Float)

Convert $\bar{B}$ to floating point. Result to C and 000. The decimal point is before the nnth digit counting from the right.

Example: Location 100 contains 25 0000 0000

Instruction	Result in ccc
0 209 100 ccc	50 250 000 00
0 207 100 ccc	52 250 000 00

0 3n0 B C
0 4n0 B C
0 5n0 B C
(Fix-Edit-Print)

Fix, edit, and print floating point numbers in locations B through C.

03 - space 1 line
04 - space 2 lines
05 - space 3 lines

5 words per line with decimal point and sign.

(a) $\bar{B}$ through $\bar{C}$ are fixed with the decimal point following the nth digit, $1 \leq n \leq 8$.

(b) Leading zeros are suppressed.

(c) $\bar{B}$ through $\bar{C}$ are printed with decimal point and sign.

(d) This instruction does not alter the contents of B through C in memory.

(e) In some cases the sign and decimal point may replace the two least significant digits of the fixed point number.

Note: For 01nn B C instruction, if the exponent of $\bar{B} > 59$, or if exponent $\bar{B}$ - nn > 49 the error routine is entered.

For 03n0 B C, 04n0 B C, 05n0 B C, if exponent $\bar{B} > 59$, or if exponent $\bar{B}$ - n > 49, error routine is entered.

If n = 1, 9, or 10 output will be in error. 03, 04, 05 instructions do not keep print line counters.

If 03, 04, 05, I 241, 242, 243 instructions are used in the same program and format control is desired, it is suggested that the counter for I 241, 242, 243 be suppressed and format be controlled by 08 instructions; to suppress print line counter: Store 99 9999 0000 in location 4141 or key into:

rA 99 9999 0000
rC 60 4141 4141,
Clear, one instruction run.

Caution: Give no I 260 instructions after print line counter is suppressed.

Input - Output

I 201 B C (Read Cards)

Numeric information is read from cards into consecutive locations beginning with B and ending with C. Either data or instructions may be read, but if instructions are to be read, operation codes I, R, and Z must be represented as follows:

I = 9/1
R = 9/2
Z = 9/3

If as many as 8 locations are to be loaded, 8 words per card are assumed. An "11" punch over the 10th digit of any word (Column 10, 20, 30, 40, 50, 60, 70, 80) will cause that word to be stored with a negative sign.

I 200 0nn 000 (Set Read Data Cards)

This instruction allows the programmer to vary the number of words per card read by the I 201 instruction. (I 201 normally reads 8 words per card).

nn	Words per card
84	8
72	7
60	6
48	5
36	4
24	3
12	2
00	1

First data word must be punched in cols. 1 - 10, second data word in 11 - 20, etc. This allows identification to be punched on data cards if desired. After I 200 is executed, each I 201 will read (nn/12) + 1 words per card until a new I 200 is executed.

I 220 B C (Punch cards)

Punch consecutive locations beginning with B and ending with C onto cards. 8 words per card are punched. Negative words are punched with an "11" punch over their 10th digits.

I 241 B C (Print, Single Space)

Print consecutive locations beginning with B and ending with C; 5 words per line, together with their signs, are printed at 50 lines per page.

I 242 B C (Print, Double Space)

Same as I 241 except for spacing.

I 243 B C (Print, Triple Space)

Same as I 241 except for spacing.

I 260 000 000 (Paper Eject)

Eject paper to next page. Reset print line counter.

Transfer

5 A B C (Copy)

The contents of A consecutive locations beginning with B into consecutive locations beginning with C.

6 A B 000 (Clear)

Place zeros (00 0000 0000) in consecutive locations beginning with A and ending with B.

Compare

9 A B C (Compare)	Compare $\overline{A}$ to $\overline{B}$. If $\overline{A} > \overline{B}$ go to C for next instruction. If $A \leq B$ continue in sequence.
R 401 B C (Test Greater)	If $\overline{000} \leq \overline{001}$ go to C. Otherwise go to B.
R 402 B C (Test Equal)	If $\overline{000} = \overline{001}$ go to B. Otherwise go to C.
R 778 B C (Transfer to Sign)	If $\overline{000} < 0$ go to C. Otherwise go to B.
R 406 B C	If $\lvert 000 \rvert > \lvert \overline{B} \rvert$ go to C. Otherwise continue in sequence.

Routines

R 600 B C (Square Root_	$\sqrt{\overline{B}}$ is placed in C and 000. $\overline{B} \geq 0$.
R 601 B C (Exponential)	$e^{\overline{B}}$ is placed in C and 000. $\lvert \overline{B} \log_{10} e \rvert < 50$.
R 602 B C (Sin)	Sin $\overline{B}$ is placed in C and 000. $\lvert \overline{B} \rvert < 10^8$
R 603 B C (Cos)	Cos $\overline{B}$ is placed in C and 000. $\lvert \overline{B} \rvert < 10^8$
R 604 B C (Log_e)	$\mathrm{Log}_e \overline{B}$ or $\log_{10} \overline{B}$ is placed in C and 000. $\overline{B} > 0$.
R 605 B C (Log_{10})	
R 606 B C	Arctangent $\overline{B}$ is placed in C and 000. $\overline{B} \geq 0$.
R 608 B C (Cos, Degrees)	Sin $\overline{B}$, Cos $\overline{B}$ is placed in C and 000. $\lvert (\pi \overline{B})/180 \rvert < 10^8$
R 609 B C (Sin, Degrees)	

Note: Not available for Bama-Bell II-80 Step

R 607 B C (10^X)	$10^{\overline{B}}$ is placed in 000 and C, $0 \leq \lvert \overline{B} \rvert \leq 50$.
R 610 B C (Arc Sin)	Arc Sin $\overline{B}$ or Arc Cos $\overline{B}$ (in radians) is placed in 000 and C. $-1 \leq \overline{B} \leq 1$

R 611 B C
(Arc Cos)

R 612 B C
(Arctan, Degrees) — Arctan $\bar{B}$, Arcsin $\bar{B}$, or Arccos $\bar{B}$ is placed in 000 and C. $B \geq 0$ for R 612.

R 613 B C
(Arc Sin, Degrees) — $-1 \leq B \leq 1$ for R 613, R 614

R 614 B C
(Arc Cos, Degrees)

Note: If $\bar{B}$ is not in range indicated, the error routine is automatically entered. See <u>Error Routine</u>.

<u>Instruction Modification</u>

Z 080 B C
(Increment A) — The A address of the instruction located in B is incremented by C. The result is stored in B.

Z 081 B C
(Increment B) — The B address of the instruction located in B is incremented by C. The result is stored in B.

Z 082 B C
(Increment C)

Example: 101 contains

1 040 200 201

The instruction Z 080 101 012 is executed. Location 101 then contains the new instruction:

1 052 200 201

Z 070 B C
(Decrementa) — The A (B,C) address of the instruction located in B is decremented by C. The result is stored in B.

Z 071 B C
(Decrement B)

Z 072 B C
(Decrement C)

Example: 101 contains

1 040 200 201

The instruction Z 071 101 075 is executed. Location 101 then contains the new instruction:

1 040 125 201

Z 090 B C
(Set A) — The A (B,C) address of the instruction in B is set to C. The result is stored in B.

Z 091 BC
(Set B)

Z 092 B C
(Set C)

Example: 101 contains

1 040 200 201

Z 092 101 300 is executed. Location 101 then contains the new instruction:

1 040 200 300

Example: 101 contains

-1 040 200 201

Z 080 101 012 is executed. Location 101 then contains

-1 052 200 201

then Z 071 101 075 is executed. Location 101 then contains

-1 052 125 201.

All the Increment-Decrement instructions handle negative instructions in this manner - i.e. If the instruction to be modified is negative, the C portion of the modification instruction is set negative before modification.

Control Instructions

An instruction may be placed in memory at any location from 001 to 999. The location of an instruction in memory is called the L ADDRESS of that instruction. Ordinarily, when Bama-Bell executes an instruction at location L, the next instruction will be taken from L + 1. The compare and control instructions interrupt this sequence.

7 A B C
(Tally)

If A + 1 ≤ B, increase A by 1 and go to C. If A + 1 > B, set A = 000 and continue in sequence. Tally instruction may not be negative.

Z 100 B C
(Loop A)

Loop B + 1 times. The loop begins at C.

Z 010 B C
(Loop B)

The A and/or B and/or C addresses of all negative instructions within the loop are increased by 1 each time the loop is repeated after the first time. When the loop is executed B + 1 times, go to L + 1 for next instruction. (Continue in sequence)

Instructions are not modified in memory by Loop or Loop-Loop instructions.

Z 001 B C
(Loop C)

Note: A loop inside a loop is not permitted. Tally orders may be used to accomplish this.

Z 110 B C
(Loop A and B)

Z 101 B C
(Loop A and C)

Z 011 B C
(Loop B and C)

Z 111 B C
(Loop A and B and C)

Z 176 B C
(Loop-Loop)

This is a generalization of the Loop instruction. A Z 176 order must be preceded by a loop instruction and immediately followed by a transfer instruction. Z 176 causes the loop controlled by the last loop order to be repeated C times. The first time the loop is repeated the addresses specified by the loop order will be incremented by B plus the amount they are normally incremented by the loop order. In general, the nth time the loop is repeated, the addresses specified by the Loop order will be incremented by n x B plus the amount they are normally incremented by the Loop order. If the B address of the Loop order is 000, Z 176 operates exactly like a Loop order except designated addresses of negative instructions are incremented by the amount B, instead of by 1. The transfer instruction, R 400 B 000 specifies the beginning of the loop which is to be repeated:

Example:	L	op	A	B	C
	200	Z	111	010	150
		. . . . .			
	210	Z	176	020	004
	211	R	400	150	000

If there are no other loop instructions between 200 and

and 210, the Z 176 instruction in 210 will cause the loop beginning at 150 and controlled by the Loop order Z 111 in 200 to be repeated 4 times with an increment Of 020. When the Loop-Loop is completed, the next instruction is taken from L + 2. (In the above example L + 2 is 212).
(See Loop-Loop Section).

R 400 B 000 — Transfer to B.

R 403 000 000 — Halt Computation.

R 405 000 000 — Take the next instruction, from L + 1, i.e. Continue in sequence, no operation.

Logical

Z 120 B C (Erase) — Erase $\bar{B}$ with $\bar{C}$. Result with sign of $\bar{B}$ is placed in 000.

Z 140 B C (Buff) — Buff $\bar{B}$ onto $\bar{C}$. Result with sign of $\bar{C}$ is placed in 000.

Exit

R 404 B 000 (Exit) — Exit from Bama-Bell. Go to B for a machine language instruction.

To re-enter from machine language place 00 0ccc 0000 in rA, where ccc is the L address of the next Bama-Bell instruction to be executed; transfer control to 1016. When Bama-Bell is re-entered, two loop counters are set to zero. If it is not desired to set the loop counters to zero, re-enter at:

4050 Bama-Bell II-80

1200 Bama-Bell II-80S

Example: To re-enter and execute an instruction in location 528.

A	op	M	C
0812	25	0814	1016
0814	00	0528	0000

Format Instructions

Note 1: These instructions are not availabel for Bama-Bell II-80Step

Note 2: These are internal instructions. They do not affect working memory; Locations 000-999.

0 600 000 000 (Read Header Card) — Read an alphanumeric header card containing 5 column titles. This header card will be available for print-out over the five Bama-Bell data print columns as follows:

Card Columns	Print Columns
1 - 10	1
11 - 20	2
21 - 30	3
31 - 40	4
41 - 50	5

Columns 51 - 80 of a header card are ignored. Any or all of columns 1 - 50 may be blank, in which case they are printed as blanks. A header card remains in Bama-Bell internal memory until a new 06 instruction is executed.

0 700 000 000 (Read Line Titles Card) — Read a card containing 5 alphanumeric line titles in card columns 1 - 50. These line titles are assigned a number in Bama-Bell internal memory as follows:

Card Columns	Line Title Number
1 - 10	0
11 - 20	1
21 - 30	2
31 - 40	3
41 - 50	4

Line titles will be destroyed only by execution of a new 07 instruction.

0 800 000 0xx (Print Header Card) — Space xx lines, $00 \leq xx \leq 79$, and print header card After an 06 instruction is executed, the programmer may give as many 08 instructions as desired without giving a new 06 instruction. This instruction will not print a line title, but will not destroy a line title if one is present. (See Note 3.)

0 900 00x 000 (Set Line Title) $0 \leq x \leq 4$ — Line title x will be printed in the right hand margin of each line of print arising from an I 241, I 242, I 243, 03, 04, 05 instruction until:

1. A read instruction, I 201, 06, 07 is executed.
2. Bama-Bell initialization routine is entered at 1016.
3. A clear line title instruction is executed.
4. A new 09 instruction is executed.

R 615 000 000 (Clear Line Title)	After this instruction is executed, line titles will not be printed until a new 09 instruction is executed.

Note 3: On 08 (Print Header Cards) instruction (0 800 000 0xx) where xx is spacing: If xx > 49 use undigit 5, 6, 7, in MSD of xx, e.g. if a spacing of 58 lines is desired put xx = 4/1 8; for 62 lines, xx = 4/2 2; for 75 lines, xx = 4/3 5. (This is the usual convention for USS 80 as regards 11 and 16 instructions).

Service Instructions

Not available for Bama-Bell II-80 Step

I 221 B C (Punch Program Cards)	Punch locations B through C onto Bama-Bell instruction cards, one location per card with L address in cols. 4 - 6, key in col. 7, contents of location in cols. 11 - 20. These cards are to be loaded by the Bama-Bell program load routine.
I 202 B C (Memory print)	Print locations B through C, one location per line, double spacing. The L address of the location is printed in col. 1, the contents of the location in col. 2, with sign if negative.

Tape Instructions

I 250 Onn Omm (00 IIII 0000) (Write Tape) Logical Uniservo 1.	Write bands nn through mm onto tape, where mm ≥ nn. L + 1 contains 00 IIII 0000 where IIII is the identification for this set of numeric information. This information is to be read back into memory by an I 258 instruction, henceIIII must be unique on the tape used. Next instruction is L + 2.

Example:

L	Instruction
200	I 250 002 004
201	0 011 110 000

Reads Bands 2 and 4 onto tape with identification 1111. Next instruction is in 202. Writes 2 blocks of tape per band.

I 258 Onn Oyy (00 IIII 0000) (Read Tape) Logical Uniservo 1.	Read tape prepared by I 250 instruction. Read yy blocks into yy/2 bands beginning with band nn. Identification is IIII in L + 1. Next instruction in L + 2. Reads 2 blocks per band. yy is even.
I 261 000 000 (Rewind Tape) Logical Uniservo 1.	Rewind tape w/o interlock.

Note: Band numbers for I 250, I 258 are 00, 02, 04, 06, 08.

Programming and Program Cards

Unless otherwise directed, Bama-Bell II will execute an instruction in location L and seek the next instruction in L + 1. This sequence may be interrupted by a control or compare instruction.

The Bama-Bell Program Load Section loads cards containing data or instructions arranged in the following format:

Cols. 4,5,6	L address of instruction or word of data; i.e. the location into which the data word or instruction is to be loaded.
Col. 7	Blank if the word is positive. 1 if the word is negative.
Cols. 11 - 20	The Bama-Bell instruction or word of data.
Cols. 70 - 75	Program name.
Cols. 78 - 80	Card number.

The last card in the program deck must be a sentinel card which contains:

Cols. 4,5,6	a) 9/8 punches, i.e. each of cols. 4,5,6 contains both a 9 and an 8 punch, if the program is not to be traced. b) 9/1 punches if the program is to be traced.
Cols. 14,15,16	The L address of the first Bama-Bell to be executed when the program is begun.

Instructions may be loaded from the keyboard of the console by using:

9/3 or 9/8 for Z

9/2 for R

9/1 for I

An A, B, or C address of a Bama-Bell instruction should always consist of digits 0 - 9, and must never be left blank.

Operating

Load Bama-Bell II into memory from cards or tape. THEN, BUT NOT BEFORE, PLACE 9800 INHIBIT SWITCH TO UP POSITION.

When Bama-Bell II is in memory, a program may be loaded by keying

00 0000 1900

into rC, general clear, select c, continuous, and run. Each program deck must have a sentinel card.

The computer stops with: 67 HHHH 1016 (regular sentinel card)

67 TTTT 1016 (trace sentinel card)

in rC; place data cards, if any, in HSR input hopper, general clear, select c, continuous, run. CAUTION: BE SURE PUNCH MOTOR IS ON AND CARDS IN ALL STATIONS.

Bama-Bell may be entered from the key-board by keying into:

rC 00 0000 1016 (Bama-Bell entrance)(If it is not desired to set loop counter back, use 4050)

rA 00 Occc 0000 (your program's entrance)

where ccc is the address of the next Bama-Bell II instruction to be executed.

General clear, select c, continuous, and run.

Loop-Loop

The Z 176 B C instruction must be preceded by a loop instruction and

IMMEDIATELY followed by a transfer instruction (R 400 B 000). The B address of the transfer instruction must be the same as the C address of the Loop instruction.

Example:

L	Sgn	op	A	B	C	
400	-	1	500	600	700	
401	+	Z	111	009	400	
402	+	Z	176	020	003	
403	+	R	400	400	000	
404	+	I	241	700	709	etc.

(When the Loop-Loop instruction in 402 has been fully executed, the nest instruction will be taken from 404).

The results of the above sequence are as follows:

a

$\overline{700} = \overline{500} + \overline{600}$

$\overline{701} = \overline{501} + \overline{601}$

$\overline{702} = \overline{502} + \overline{602}$

. . . .

$\overline{709} + \overline{509} + \overline{609}$

b

$\overline{720} = \overline{520} + \overline{620}$

$\overline{721} + \overline{521} + \overline{621}$

. . . .

$\overline{729} + \overline{529} + \overline{629}$

c

$\overline{740} = \overline{540} + \overline{640}$

. . . .

$\overline{749} = \overline{549} + \overline{649}$

d

$\overline{760} = \overline{560} + \overline{660}$

. . . .

When a Loop-Loop instruction in location L is executed, the next instruction is taken from L + 2, so the transfer instruction in L + 1 is ignored.

The computations in the above example are performed and the results are printed out by the following program:

L	Sgn	op	A	B	C
400	-	1	500	600	700
401	+	Z	111	009	400
402	+	Z	176	020	003
403	+	R	400	400	000
404	-	I	241	700	709
405	+	Z	011	000	404
406	+	Z	176	020	003
407	+	R	400	404	000
408	+	R	403	000	000

Stops

67 7200 cccc	Reader off normal. Correct error. Clear, run.
67 1100 cccc	Printer off normal. Correct error. Clear, run.
67 8100 cccc	Punch off normal. Correct error. Clear, run.
67 1016 1900	PROGRAMMED HALT. To load new program, select c and run. To begin computation at location ccc, Key 00 0ccc 0000 in rA, continuous, select m, run.
67 9800 1016	9800 inhibit switch is down. Place switch to up position, Key into rA: 00 0ccc 0000 where ccc is entrance to your program. General clear, continuous, select c, run.
67 F600 cccc	Tape error. Clear, run.

Note: If computer should ever hang up on a 22 test, select m, continuous, run. This can happen after an overflow in certain cases.

67 H42H H66H (67 NO) — Program error. See Error Routine.

Error Routine

If a program error such as division by zero, Log $\bar{B}$, $\bar{B} \leq 0$, Arctan $\bar{B}$, $\bar{B} < 0$, or other condition occurs, the error routine is entered. A trace line is printed:

1st word: "L" address of instruction causing error

2nd word: $\bar{A}$

3rd word: $\bar{B}$

4th word: $\overline{000}$

5th word: Instruction causing error; the instruction is preceded by a 1 if the instruction is negative.

6th word: Loop counter contents.

Then the paper in the printer is spaced 79 lines, and a 67 NO stop appears in rC.

Example of Error Trace Line

400	5213700000	0000000000	5010000000	1	4500600700	22022022

Bama-Bell attempted to execute the instruction 4 522 622 722. Location 522 contains 137.00 and 622 contains zero. Division by zero has been attempted.

Trace Routine

If the program deck is followed by a valid trace sentinel card, the program deck will load and stop with 67 TTTT 1016 in rC, (see operating section). Data cards are placed in HSR hopper, general clear, continuous, run. The program just loaded will be executed, and a trace line printed after each instruction is executed:

1st word:	"L" address
2nd word:	$\bar{A}$
3rd word:	$\bar{B}$
4th word:	$\overline{000}$
5th word:	Instruction preceded by "1" if the instruction is negative.
6th word:	Current contents of the loop counter.

If a program error occurs during tracing, the error routine is entered, (see Error Routine section).

Table of Important Locations

Location	Function
1016	Enter initialization section
4050	Enter interpretative routine
4056	Current "L" address
4014	Loop Counter 1.
4141	Print Line Counter, I 241, 242, 243
0000 - 0999	Programmer's working memory
2200 - 2399	Print, Punch, Read interlaces
4000 + n	Enter routine for n A B C instruction $1 \leq n \leq 9$.
4aaa	Enter routine for Iaaa B C, Raaa B C instruction
4800 + aaa	Enter routine for Zaaa B C instruction
3000 + n	Enter routine for On instruction $1 \leq n \leq 9$
1900	Enter program load.
4239	Enter trace routine.