intercom 1000

DOUBLE PRECISION

Appendix Sub Routines • TECHNICAL MANUAL

THE BENDIX CORPORATION COMPUTER DIVISION
5630 Arbor Vitae Street, Los Angeles 45, California

APPLICATIONS SECTION

Project No. 84

TITLE: Intercom 1000 Double Precision Appendix Technical Manual

TYPE: General

EQUIPMENT AFFECTED: G-15D (Intercom 1000 D.P.)

SUBROUTINES USED: Refer to the contents of this project.

PREPARED BY: Bendix Computer Division

APPROVED BY: Tak Yamashita

DATE: 14 May 1959

PROJECT NO. 84

TABLE OF CONTENTS

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

TABLE OF CONTENTS (CONT.)

PROJECT NO. 84

INTERCOM 1000 DOUBLE PRECISION APPENDIX

TECHNICAL MANUAL

1. Introduction

In order to increase the versatility of the Intercom 1000 System and in order to be able to use auxiliary equipment with the Intercom 1000 System, a number of subroutines were developed. These subroutines are documented in this manual so that the internal structure is available to any user.

Some innovations have been introduced here which will not be found in the January, 1959 issue of Intercom 1000 manual. According to instructions contained in the manual, in order to enter a subroutine one has to use a command beginning with operation code 08. When this operation code is used, any mathematical subroutine specified in the address portion will be entered and will operate from line 19. All subroutines beginning with operation code 08 are grouped in the Appendix I to Intercom 1000. As an additional improvement it has been decided to develop Appendix II to Intercom 1000. Appendix II contains essentially the same mathematical subroutines and the only difference is that they are executed from line 05 instead of from line 19. In order to enter such subroutines, operation code 02 is used instead of 08. Appendix II does not contain input-output routines which are part of Appendix I.

Since these subroutines are tailor-made for the Intercom 1000 System, the specifications will not necessarily be of standard form but they will be consistently laid out in the manual. The following pattern will be used: each subroutine will consist of four parts; specifications, method, flow charts and coding sheets.

All subroutines are executed from line 19 if Appendix I is used and from line 05 if Appendix II is used.

Unless otherwise specified, the argument x (data input) must be placed in the "A" Register, and the result (data output) will be found also in the "A" Register.

1.1 Appendix I (Operation Code 08)

"Intercom 1000, Appendix I" is a tape containing subroutines which use channel 19 and whose operation code is 08.

Operation of the Appendix is fully explained in the Intercom 1000 Manual (pages 18, 19 of January, 1959 edition CB-029.)

1.2 Appendix II (Operation Code 02)

(Note: Operation Code 02 is not listed in the manual) Appendix II contains subroutines whose operation code is 02 and which do not involve channel 19 during their execution. They are executed, in the double precision system from channel 05.

In all other respects the operation of Appendix II is identical with that of Appendix I.

Note that Appendix II contains only the fraction length selector and the mathematical subroutines (square root, logarithms, exponential, sine, cosine, arc tangent.)

1.3 Comparison between the Appendices

Appendix II generally provides a faster program, for its subroutines can operate simultaneously with any input or output.

However, the use of Appendix II impedes the "listing of selected commands." (See manual page 16.) This consideration may make it preferable to use Appendix I exclusively, especially where there is little input or output during computation.

Nevertheless, even where it is planned to use Appendix II subroutines in the final version of a program, it may be advantageous to replace them temporarily with Appendix I subroutines and the corresponding 02 operation codes with 08. Thus, unimpeded listing could be resorted to whenever desired. Once the program is thoroughly "debugged," it will be a simple matter to restore the Appendix II subroutines and the 02 operation codes.

The Appendix II tape contains only mathematical subroutines, whereas Appendix I contains the same subroutines plus service routines such as flex-input, flex-output, magnetic tapes, etc.

2. General Description of Subroutines

The following pages contain some details of the Appendix Loader and the specifications, description of method, flow chart, coding sheets and memory allocation chart (sexadecimal type-out) of each subroutine, except the "fraction length selector," details of which are given in Applications Section Project No. 83.

The coding sheets and sexadecimal type-outs are those of Appendix I. The only difference between the two versions occur where "19" in the source or destination column has to be changed to "05", and where the characteristic "6", in the return commands, has to be changed to characteristic "5". These changes are listed at the end of the "specifications" for each subroutine.

The check sum or summation constant of each block of tape is in location u7 of the line (the first number read in), and is used by the loader to verify whether the tape has been correctly read in. This and some other auxiliary constants have no effect on the execution of the program. Differences in these constants between the two versions are not listed in the "specifications."

The error alarm in Intercom 1000 is five bell-rings and printed periods followed by a halt. The error exit, that is, the command in a subroutine which leads to the error alarm, is listed in the specifications.

Source 29 is used to obtain zero in most of the subroutines.

2.1 Check Sums

As of April 20, 1959 the following were the check sums of the subroutines in the two appendices:

	Appendix I	Appendix II
Loader	.w0yv40z	.w0yv40z
Fraction Selector	.1001601	.1001501
Square Root	.1002602	.1002502
Logarithms	.1000603	.1001503
Exponential	.1000604	.1001504
Sine-Cosine	.1001605	.1002505
Arc Tangent	.1000606	.1000506

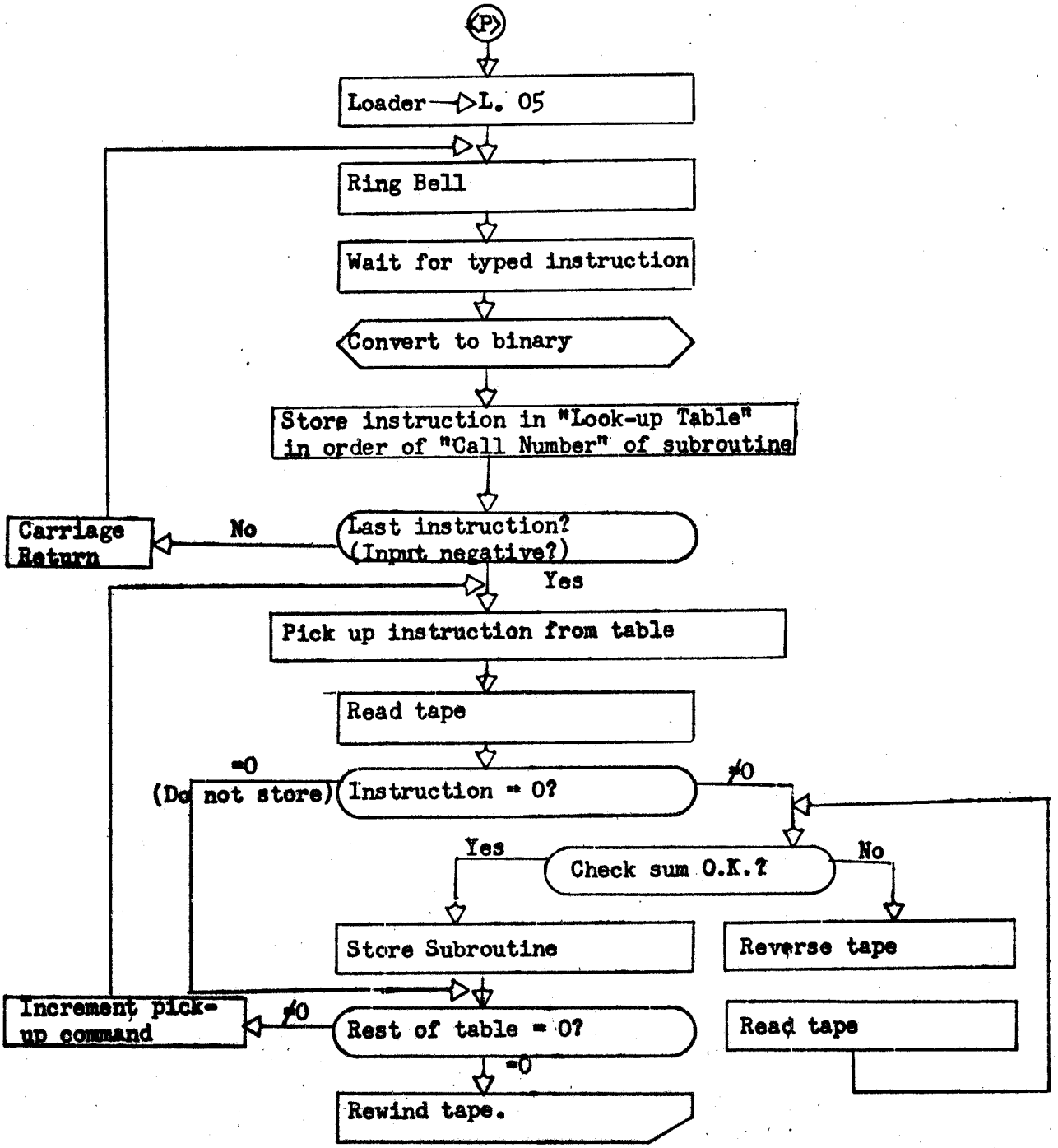
	Appendix I	Appendix II
Flex-Input	.1000607	
Flex-Output	.1000608	
Magnetic Tape	.1003189	
CA-1 Data	.100060 u	
CA-1 Commands	.100060 v	

3. The Loader

For instructions and codes required to call any Appendix subroutines, refer to page 18 of the Intercom 1000 manual.

3.1 INTERCOM 1000 (D. P.) APPENDIX LOADER

FLOW CHART



Bendix Computer — Los Angeles 45, California

Page 7 of 5

G-15D — Prepared by D. Hassell — Date:

PROGRAM PROBLEM: Appendix Loader — Line 05

L	P	T or Lk	N	C	S	D	BP	NOTES
								Enter from typing of ⟨P⟩
.00		u.01	.01	.0	.19	.05		Line 19 ——▷ Line 05
.01		.02	.02	.5	.21	.31		N.C. 05.02
.02		u4	u6	.4	.05	.25		Replace 00,01 with u4 and u5
.u6		00	00	.4	.25	.05		
.u4		04	04	.0	.23	.31		goes into 00 clear 2 wd registers
.04		u.09	.09	.0	.25	.23		Clear 23
.09		09	09	.0	.28	.31		Wait for ready
.10		10	11	.0	.17	.31		Ring bell
.11		13	13	.0	.12	.31		Gate input
.13		13	13	.0	.28	.31		Wait for ready
.14		16	17	.2	.23	.28		(23.00) ——▷ AR
.17		19	20	.0	.28	.21		(AR) ——▷ 21.03
.20		23	24	.0	.05	.20		(0ZZZZZZ) ——▷ 20.03
.24		27	28	.0	.31	.28		ONNCHWD ——▷ AR
.28		29	31	.2	.25	.25		(AR) ——▷ ID_1 0 ——▷ AR
.31		16	76	.1	.26	.31		ID_1 = OOONNCH
.76		63	65	.2	.25	.28		(ID_1) = OOONNCH ——▷ AR
.65		02	68	.1	.26	.31		ID_0 = WDOOOOO
.68		69	71	.1	.24	.25		0 ——▷ ID_1
.71		08	89	.1	.26	.31		(ID_0) = OWDOOOO
.89		90	18	.2	.25	.29		(ID_0) = OWDOOOO ——▷ AR+ = OWDNNCH
.18		u.39	.u0	.0	.00	.19		Dec/Bin command conversion
.u0		19	19	.6	.21	.31		N.C. 19.19 see 00.19

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 2 of 5

G-15D

Prepared by D. Hassell

Date:

PROGRAM PROBLEM: Appendix Loader

Line 05

L	P	T or LK	N	C	S	D	BP	NOTES
								Enter from command conv.
.40		.73	.73	.5	.21	.31		N.C. 05.73
.73		.08	.85	.1	.26	.31		Shift CH = (ID_0) to destination position
.85		.83	.84	.0	.26	.21		UDNNOOO ▷ 21.03
.84		.85	.86	.2	.25	.28		CH.2^{-28} ◁ AR
.86		.87	.88	.2	.26	.29		(WDNNOOO) ▷ AR+
.88		.91	.92	.3	.31	.29		(OONNOOO) ◁ AR+ = WDOOOCH
.92		.94	.96	.4	.28	.25		(AR) ▷ ID_0,1
.96		.99	u1	.0	.31	.28		(OONNOOO) ◁ AR
.u1	u	.02	.05	.0	.28	.29		OONNOOO x 2^{-8} = (NNOOOOO) ◁ AR
.05		.06	.34	.0	.05	.29		Dummy table store command + ▷ AR+
.34		.39	.39	.0	.31	.31		Obey AR
.06		.41	.66	.0	.25	.05		Store table entry
.66		.68	.69	.0	.23	.28		Input (sign) ◁ AR
.69		.74	.74	.0	.22	.31		AR < 0? Typewriter input complete?
								No
.74		.03	.07	.0	.05	.03		Type a carriage return
.07		.09	.00	.0	.08	.31		
								Yes
.75		.76	.77	.2	.24	.28		(MQ_0) = 0 ▷ AR
.77		.42	.36	.2	.05	.05		0 ▷ 05.42 + i 05.42 + i ▷ AR
.36		.36	.36	.0	.28	.31		Wait
.37	u	.38	.38	.1	.24	.19		0 ▷ line 19
.38		.40	.83	.0	.15	.31		Read tape
.83		.84	.90	.0	.28	.27		Store tape input?

FORM BCD 108-17.0

Los Angeles 45, California

Page 3 of 5

G-15D

PROGRAM PROBLEM: Appendix Loader

Prepared by D. Hassell

Date:

Line 05

L	P	T or LK	N	C	S	D	BP	NOTES
								No
.90		.91	.41	.0	.00	.00		N.C. 05:41
.41	u	.60	.63	.0	.05	.27		Through reading?
								Yes
.63		.65	.67	.0	.00	.31		Set ready
.67		.69	.15	.0	.15	.31		Read
.15		.17	.19	.0	.00	.31		Set ready
.19		.21	.35	.0	.06	.31		Reverse
.35		.36	.39	.1	.23	.28		23.00 ⟨> AR
.39		.40	.70	.3	.05	.29		(05.40) ⟨> AR+
.70		.71	.81	.0	.28	.27		AR = 0?
								=0
.81		.82	.97	.1	.24	.27		(MQ_0) = 0?
								=0
.97		.98	.99	.1	.05	.24		Non-zero ⟨> MQ_0
.99		.13	.16	.1	.08	.28		1W00000 ⟨> AR
.16		.40	.30	.0	.05	.23		(05.40) ⟨ 23.00
.30		.62	.62	.0	.22	.31		AR < 0?
								+
.62		.64	.95	.0	.00	.31		Set ready
.95		.97	.u5	.0	.06	.31		Reverse
.u5		.92	.30	.3	.07	.29		(0100000) ⟨> AR+
.98		.00	.u3	.0	.00	.31		Set ready

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 11 of 5

Prepared by D. Hassell

Date:

G-15D

PROGRAM PROBLEM: Appendix Loader

Line 05

L	P	T or LK	N	C	S	D	BP	NOTES
.u3		.u5.	u2.	0.	29.	31		Reset overflow
.u2		.00.	00.	0.	21.	31		Exit to Intercom
.82		.84.	99.	1.	29.	24		Zero ⋯ ◁ MQ_0
								Increment table lookup
.64		.77.	78.	0.	05.	28		(05.77) ▷ AR
.78		.79.	80.	0.	05.	29		(0100000) + ▷ AR+
.80		.77.	75.	0.	28.	05		Restore (AR) ⋯⋯ ▷ 05.77
.91		.93.	94.	0.	05.	29		Set to store input
.94		.u7.	26.	0.	28.	05		(AR)
.26		.26.	26.	0.	28.	31		
.27		.u7.	08.	1.	19.	28		
.08		u.09.	12.	3.	19.	29		
.12		.30.	32.	0.	28.	27		
.32		.34.	u7.	0.	00.	00		
.33		.35.	21.	0.	06.	31		
.21		.21.	21.	0.	28.	31		
.22		u.23.	25.	1.	24.	19		
.25		.27.	26.	0.	15.	31		

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 5 of 5

G-15D
PROGRAM PROBLEM: Appendix Loader

Prepared by D. Hassell

Date:

Line 05

L	P	T or LK	N	C	S	D	BP	NOTES
.93	u	20	84	0	19	00		
.64								
00-								
.03	u	68	00	0	00	00	.4	400000
.23		15	w7	3	31	31	-	.0zzzzzz
.79	u	01	00	0	00	00	.0	100000

FORM BCD 108-17.0

MEMORY ALLOCATION CHART

Problem { Intercom 1000 (D.P.) / Appendix Loader. }

Page 1 of 1.

Date 2 April '59

Prepared by Don Hassell.

Check Sum .w0yv40z

Line 05.

U7	U6	U5	U4
	-8000325	xw1ywzx	4042zz
U3	U2	U1	U0
69663vz	2vz	20539x	-1313uvz
99	98	97	96
8x1051w	6701z	63634v8	y3653zw
95	94	93	92
61690xz	yv1u385	1454260	-xy60399
91	90	89	88
xx5y0vx	5w29000	5v12v3x	xv5wzzx
87	86	85	84
	5858v5x	x354355	5656v3w
83	82	81	80
555u39v	x4637v8	536171v	wx4v385
79	78	77	76
100000	50500vx	uu248u5	vz41v3w
75	74	73	72
4x4xv1w	83070u3	85575z	
71	70	69	68
85975z	485139v	4u4u2xz	4647719
67	66	65	64
450z1zz	w4452zw	24475z	wx4y0vw
63	62	61	60
414301z	405z01z		
59	58	57	56
55	54	53	52
51	50	49	48
47	46	45	44
43	42	41	40
		3w3z0vv	-49496vz
39	38	37	36
2946wvx	28531zz	2626713	242439z
35	34	33	32
25276zw	27273zz	23150xz	u26v000
31	30	29	28
104w75z	3y3y2xz		1y1zv39
27	26	25	24
yv0867w	1ulu39z	1v1ulzz	9v1w3zw
23	22	21	20
zzzzzz	1719713	151539z	97180v4
19	18	17	16
15230xz	2764013	9314395	u81y0v7
15	14	13	12
111301z	9011uzw	x0x39z	9y2039v
11	10	09	08
x0x19z	u0v23z	90939z	90wy7x
07	06	05	04
90011z	u942325	7220vx	909337
03	02	01	00
4400000	-y86u0v9	- 2026vz	101265

Four-Word Lines

Line	3	2	1	0

4. Square Root

4.1 Specifications

ENTRY: Loc. 97

EXIT: The return from the subroutine will always be made to L. 00.01 unless a different exit is required. A return command can then be inserted either in loc. 50, 80 or 98 of this subroutine (See coding sheets).

ERROR EXIT: An error exit will be made to L. 04.26 unless a different exit is required. A return command can be inserted in loc. 89 of this subroutine (See coding sheets).

ERROR INDICATION: A negative argument will set up the error alarm.

DATA INPUT: Argument $\underline{x}$ must be placed in the "A" Register (21.00,01)

DATA OUTPUT: "A" Register (21.00,01)

RANGE OF ARGUMENT: Any non-negative number that can be expressed in Intercom 1000. The range is approximately from 2.9×10^{-38} to 1.07×10^{38}.

EXECUTION TIME: 5 drum cycles

ACCURACY: $\frac{\text{Error}}{\text{Answer}} \leq 2^{-49}$

SHORT LINE LOCATIONS USED: Line 20, words 0, 1
Line 21, words 0, 1, 2, 3
Line 22, word 0
Line 23, words 2, 3

LOCATIONS NOT USED IN LONG LINE: 00 through 45, and 55, 74, 80, 81, 84, u0 through u7.

CHECK SUM: 1002602 (Appendix I)
1002502 (Appendix II)

REMARKS: This tape contains an alternative square root subroutine with entry at 74 instead of 97 which saves one drum revolution, but which leaves only the following locations available to the programmer: 02, 04, 05, 07, 08, 10, 11, 15, 17, 18, 19, 21, 23, 24, 25, 27, 30, 31, 35, 36, 37, 38, 40, 41, 42, 43, 44, 45, 46, 48, 50, 51, 53, 55, 56, 58, 62, 68, 76, 78, 79, 83, 84, 86, 88, 89, 95, 97, u0-u7. (See page 3 of coding sheets)

APPENDIX II
VERSION: Change "19" to "05" in commands: 12, 32, 51, 59, 70, 88.

4.2 Method

Let $A = B \cdot 2^{b-128}$, where b is the binary exponent stored with the fraction B in d.p. floating point.

Let $\sqrt{A} = R \cdot 2^{r-128}$, in floating binary notation.

If b is even, then $r = \frac{b}{2} + 64$, $R = \sqrt{B}$

If b is odd, then $r = \frac{b+1}{2} + 64$, $R = \sqrt{B/2}$

Let $x = R^2$; then $1/4 \leq x < 1$.

If $R_0 = a + x(1-ax)$, where $a = 35 \cdot 2^{-7}$, then $|R-R_0| < 2^{-7}$.

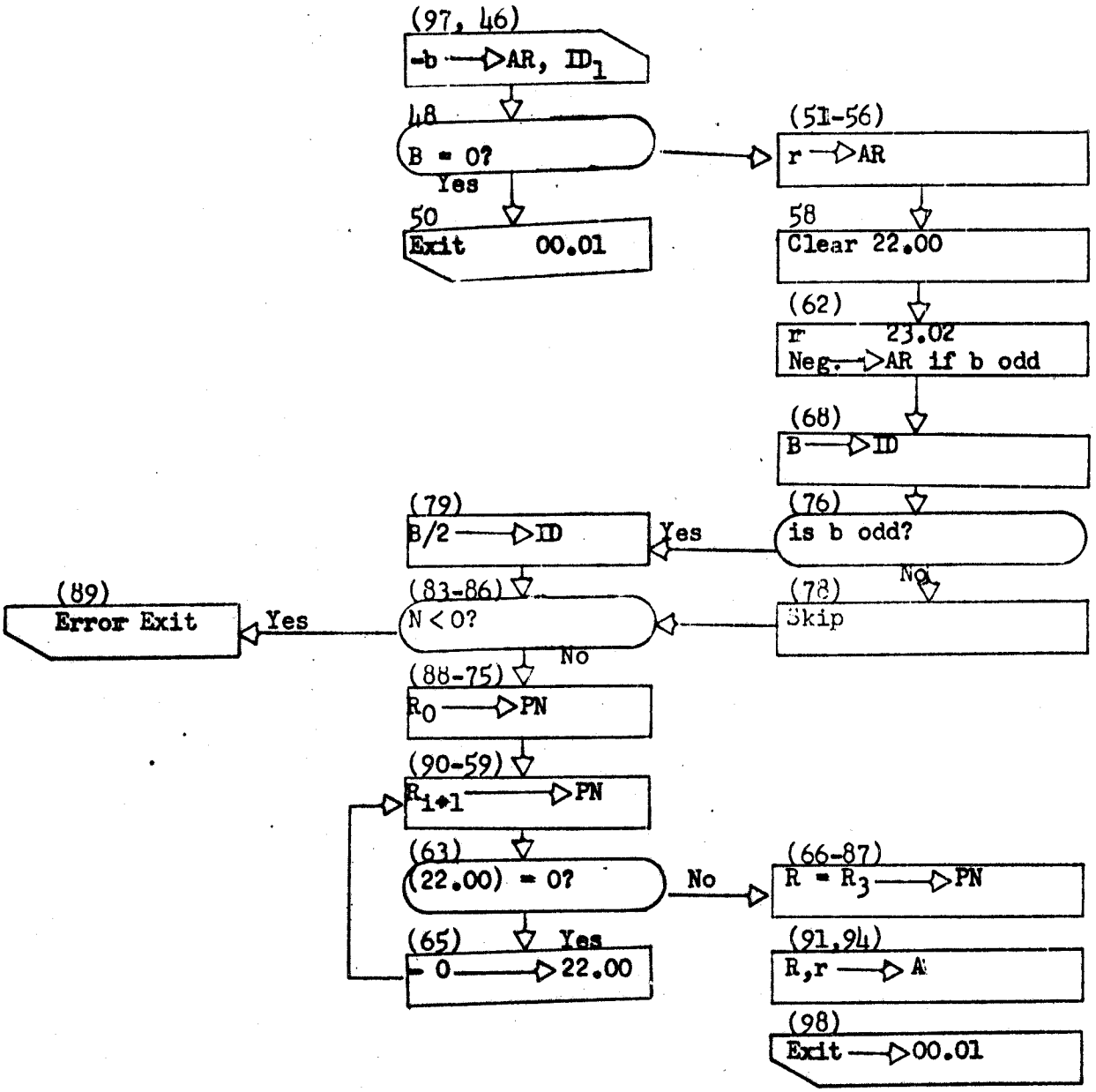
If $R_{i+1} = 1/2 \left(\frac{x}{R_i} + R_i\right) = 1/4 \left(x/\frac{Ri}{2}\right) + \frac{Ri}{2}$, then $|R-R_1| < 2^{-14}$; $|R-R_2| < 2^{-28}$; $|R-R_3| < 2^{-56}$.

R_1 and R_2 may be determined with sufficient accuracy by single precision divisions; only R_3 requires a final double precision division.

If was found during debugging that correct roots of certain numbers could be obtained only by filling bits to the right of single precision R_1 and R_2 with binary ones. This guarantees that $R_1 \geq R_2 \geq R$ in all cases.

4.3 INTERCOM 1000 (D. P.) SQUARE ROOT SUBROUTINE

FLOW CHART



Bendix Computer — Los Angeles 45, California

G-15D

PROGRAM PROBLEM: Intercom 1000 (D.P.) Square Root Subroutine

Prepared by S.H. Lewis

Page 1 of 3

Date: 31 Mar. 59

Line 19.

L	P	T or L_K	N	C	S	D	BP	NOTES
.97	s	.u0.	46.	3.	30.	28		b = (30.00) ▷ AR
.46	.	.47.	48.	1.	28.	25		(AR) ——+——▷ ID_1
.48	.	.49.	50.	0.	21.	27		(21.01) = 0?
.50	s	.01.	01.	0.	21.	31		=0 Exit ——▷ 00.01
.51	s	.52.	53.	3.	19.	29		≠0 [0000040] ——▷ AR+
.53	.	.02.	56.	0.	24.	31		Mult. (To shift)
.56	.	.57.	58.	2.	25.	29		|(ID_1)| ——▷ AR+
.58	.	.60.	62.	0.	29.	22		Clear (22.00)
.62	.	.66.	68.	6.	25.	23		A = (AR) ▷ 23.02 (ID_1) ▷ AR
.68	.	.72.	76.	4.	31.	25		B = (31.00,01) ——▷ $ID_{0,1}$
.76	.	.78.	78.	0.	22.	31		(AR)< 0?
.78	s	.83.	83.	0.	00.	00		+ Skip (b even)
.79	s	.02.	83.	0.	26.	31		− Shift (b odd)
.83	.	.84.	86.	0.	21.	28		(21.00) ——▷ AR
.86	.	.88.	88.	0.	22.	31		(AR) < 0?
.88	s	.92.	95.	6.	19.	24		+ a = [.4600000] ——▷ MQ_1
.89	s	.26.	26.	4.	21.	31		− Error exit
.95	su	.u0.	49.	4.	25.	21		x = ($ID_{0,1}$) ——▷ 21.00-03
.49	.	.14.	64.	0.	24.	31		Mult.
.64	.	.65.	67.	0.	26.	28		ax = (PN_1) ——▷ AR
.67	.	.68.	70.	4.	21.	25		x = (21.00,01) ——▷ $ID_{0,1}$
.70	.	.71.	72.	0.	19.	26		a = [.4600000] ——▷ PN_1
.72	.	.73.	75.	3.	28.	24		1-ax = -(AR) ——▷ MQ
.75	.	.14.	90.	0.	24.	31		Mult [y_0 = a+x (1-ax)]

FORM BCD 108-17.0

Los Angeles 45, California

G-15D

Prepared by S.H. Lewis

Date: 31 Mar. 59

PROGRAM PROBLEM: Intercom 1000 (D.P.) Square Root Subroutine

Line 19.

L	P	T or LK	N	C	S	D	BP	NOTES
.90	.u.	93.	93.	4.	26.	25		Ri = $(PN_{0,1})$ ⟨ $ID_{0,1}$
.93	. .	02,	96,	0,	26,	31		Shift
.96	.u.	99.	99.	4.	21.	26		x = (21.01,02) ⟩ $PN_{0,1}$
.99	. .	53.	47.	1.	25.	31		Divide (S.P.)
.47	. .	48.	54.	6.	24.	26		$1/4\ (x/\frac{Ri}{2}) = (MQ_0$ ⟩ PN_1
.54	. .	55.	57.	0.	25.	30		$\frac{Ri}{2} = (ID_1)$ ⟩ PN_1
.57	. .	59.	59.	0.	23.	31		Clear (even)
.59	. .	60.	63.	4.	19.	30		[-.0000001, zzzzzzz] ⟩ $PN_{0,1}$*
.63	. .	64.	65.	0.	22.	27		(22.00) = 0?
.65	s .	68.	90.	3.	29.	22		=0 ⟩ -0 ⟩ 22.00
.66	su.	69.	69.	4.	26.	25		≠0 $R_2 = (PN_{0,1})$ ⟨ $ID_{0,1}$
.69	. .	02.	73.	0.	26.	31		Shift
.73	. .	74.	77.	4.	21.	26		x = (21.02,03) ⟩ $PN_{0,1}$
.77	. .	v2.	82.	1.	25.	31		Divide (D.P.)
.82	.u.	85.	85.	4.	24.	26		$1/4\ (x/\frac{R_2}{2}) = (MQ_{0,1})$ ⟩ $PN_{0,1}$
.85	. .	86.	87.	1.	23.	28		r = (23.02) ⟨* ⟩ AR
.87	. .	88.	91.	4.	25.	30		$R_2/2 = (ID_{0,1})$ ⟨ $PN_{0,1}$*
.91	. .	92.	94.	4.	26.	21		R = $(PN_{0,1})$ ⟨ 21.00,01
.94	. .	96.	98.	0.	27.	21		(27.00) 21.00
.98	. .	01.	01.	0.	21.	31		Exit 00.01
.52								.0000040
.60								.zzzzzzz
.61								-.0000001
.71								.4600000
.92								.4600000

FORM BCD 108-17.0

Los Angeles 45, California

Page 3 of 3

G-15D

Prepared by S.H. Lewis

Date: 31 Mar. 59

PROGRAM PROBLEM: Intercom 1000 (D.P.) Square Root Subroutine

Line 19.

L	P	T or L_K	N	C	S	D	BP	NOTES
.74	s	.77	.80	.0	.21	.27		(21.01) = 0?
.80	s	.01	.01	.0	.21	.31		=0 Exit ⟨> 00.01
.81	s	.84	.00	.0	.21	.28		≠0 (21.00) ⟨> AR
.00	.	.02	.02	.0	.22	.31		(AR) < 0?
.03	s	.26	.26	.4	.21	.31		− Error exit
.02	s	.04	.06	.3	.30	.28		+ b = (30.00) ⟨> AR
.06	.	.07	.09	.1	.28	.25		(AR) ⟨> ID_1
.09	.	.02	.12	.0	.24	.31		Mult. (To shift)
.12	.	.13	.14	.3	.19	.29		[.0000040] ⟨> AR+
.14	.	.15	.16	.2	.25	.29		|(ID_1)| ⟨> AR+
.16	.	.18	.20	.0	.28	.23		r = (AR) ⟨> 23.02
.20	.u	.22	.22	.4	.25	.28		(ID_1) (a) ⟨> AR
.22	.	.24	.26	.4	.31	.25		b = (31.00,01) ⟨> $ID_{0,1}$
.26	.	.28	.28	.0	.22	.31		(AR) < 0?
.28	s	.32	.32	.0	.00	.00		Skip (b even)
.29	s	.02	.32	.0	.26	.31		Shift (b odd)
.32	.	.33	.34	.0	.19	.24		[.4600000] ⟨> MQ_1
.34	.	.36	.39	.0	.29	.22		Clear (22.00)
.39	.u	.44	.49	.4	.25	.21		x = ($ID_{0,1}$) ⟨> 21.00,01,02,03
								See P. 1
.13								.0000040
.33								.4600000
.u6								−.v0z9zzz4
.u7								.1002602

FORM BCD 108-17.0

Bendix Computer

MEMORY ALLOCATION CHART

Problem: Intercom 1000(D.P.) — SQUARE ROOT.

Page 1 of 1

Date L(19.

Prepared by SsH. Lewis

Check Sum 1002602

Line 19.

U7 1002602	U6 -v0z9zz4	U5	U4
U3	U2	U1	U0
99 352z73z	98 1012vz	97 y42yzxw	96 -63632vu
95 -6431335	94 y062375	93 26035z	92 4600000
91 -5y5y355	90 -5x5x359	89 -1ulu2vz	88 -xw5zu78
87 -5u5v33y	86 58582xz	85 57576zw	84
83 55562vw	82 -555531u	81 x4002vw	80 1012vz
79 25335z	78 x353000	77 705273z	76 4y4y2xz
75 y5u31z	74 wx502vv	73 -4w4x2vu	72 4u4vz98
71 4600000	70 484827u	69 24935z	68 -w84w3z9
67 -46462v9	66 -4545359	65 w45uzv6	64 424335w
63 41412xv	62 -w244v37	61 - 1	60 zzzzzzz
59 -3y3z27y	58 vw3y3v6	57 3v3v2zz	56 3u3uv3x
55	54 383933y	53 23831z	52 40
51 3535y7x	50 1012vz	49 y4031z	48 32322vv
47 -3236v1u	46 3030799	45	44
43	42	41	40
39 -2w31335	38	37	36
35	34 u4273v6	33 4600000	32 2222278
31	30	29 22035z	28 u020000
27	26 1w1w2xz	25	24
23	22 -981u3z9	21	20 -161633w
19	18	17	16 9214397
15	14 1010v3x	13 40	12 y0yy7x
11	10	09 20w31z	08
07	06 809799	05	04
03 -1ulu2vz	02 8406zxw	01 x4002vw	00 2022xz

Four-Word Lines

Line	3	2	1	0

5. Logarithm

5.1 Specifications

ENTRY: Loc. 71 for $\log_{10}x$ (word position to be used in "Perform Subroutine" Command, - 71)
Loc. 17 for $\log_{e}x$ (word position to be used in "Perform Subroutine" Command, - 17)
Loc. 08 for $\log_{2}x$ (word position to be used in "Perform Subroutine" Command, - 08)

EXIT: The return from the subroutine will always be made to L. 00.01 unless a different exit is required. A return command can then be inserted either in loc. 26 or 75 of this subroutine (See coding sheets).

ERROR EXIT: An error exit will be made to L. 04.26 unless a different exit is required. A return command can be inserted in loc. 82 of this subroutine (See coding sheets).

ERROR INDICATION: A negative or zero argument will be rejected and will set up the usual error indication.

DATA INPUT: Argument x must be placed in the "A" Register and $x > 0$.

DATA OUTPUT: "A" Register

RANGE OF ARGUMENT: Any positive non-zero number that can be expressed in Intercom 1000. The range is approximately from 2.9×10^{-38} to 1.07×10^{38}.

EXECUTION TIME: Approximately 22 drum cycles. No attempt has been made for optimizing this subroutine, rather it was written to accompany the 1000 Interpretive System.

ACCURACY: Correct to at least 11 significant figures.

SHORT LINE
LOCATIONS USED: Line 20, words 0, 1, 2, 3
Line 21, word 0, 1, 2
Line 22, words 0, 1

LOCATIONS NOT USED IN LONG LINE: 21, 32, u0-u7.
(Locations u5, u6, u7 contain summation and identification constants. These locations can be used for other purposes, since the constants are not required in the operation of this subroutine).

CHECK SUM: 1000603 (Appendix I)
1001503 (Appendix II)

REMARKS: Logarithms of arguments less than unity are expressed as negative numbers.

APPENDIX II
VERSION: Change source or destination 19 to 05 in the following commands: 08, 11, 17, 36, 42, 45, 50, 54, 61, 62, 64, 70, 72, 84, 85, 86, 90, 91, 96.

5.2 Method

The basic formula used is: $\log_2 u = 1/2 + C_1 v + C_2 v^3 + C_3 v^5 + C_4 v^7 + C_5 v^9$ where $v = \frac{u-\sqrt{2}}{u+\sqrt{2}}$ $\quad 1 \leq u < 2$

and where the constants have the following values (courtesy of Mr. E.T. Federighi of Bendix Radio Division)

	Decimal	Sexadec.	Scaled 2^{-3}
C_1 =	2.8853, 9008, 1790, 0618	5w551x9	95z6y40
C_2 =	.9617, 9667, 3368, 5217	1yw709x	2613xz0
C_3 =	.5770, 8358, 0209, 1610	-127777z	803w510
C_4 =	.4116, 7299, 2266, 0604	0x2w6wx	x9v77y8
C_5 =	.3407, 2519, 8333, 4278	0uy7388	zz0z8x8

The logarithms to all bases are just calculated to base 2, then converted to the required base by the multiplication of a constant, since: $\log_{10} x = \log_2 x \cdot \log_{10} 2$
$\log_e x = \log_2 x \cdot \log_e 2$

The argument is reduced to a fixed point number between 1/4 and 1/2. This represents u/4 in the formula

$$v = \frac{u/4 - \sqrt{2}/4}{u/4 + \sqrt{2}/4}$$

which gives the v of the polynomial above.

The polynomial is then evaluated. The result is complemented if $x<1$.

The values of logarithms for arguments outside of the basic range are found by adding the proper power of 2 for each bit in the binary exponent; for if

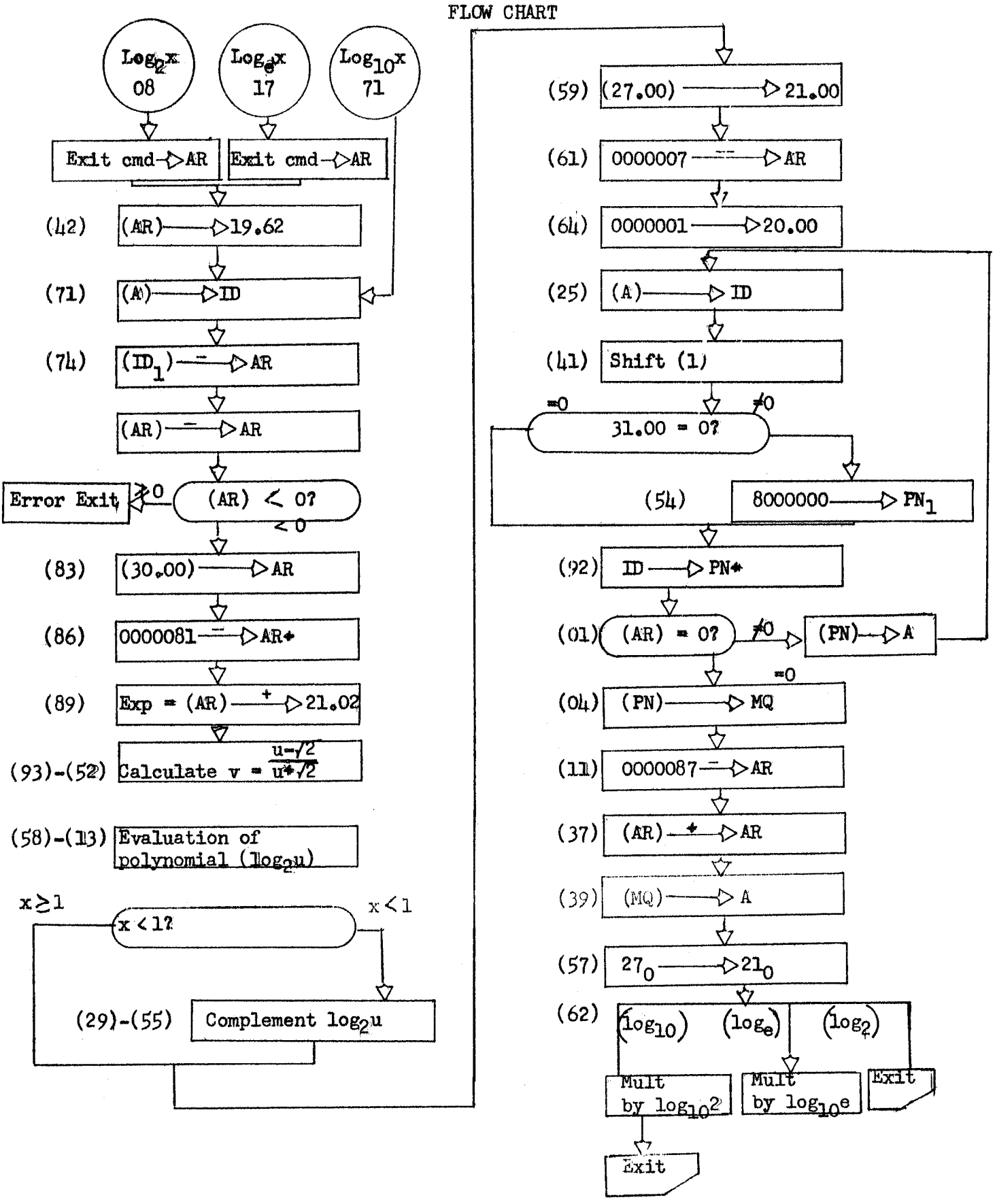
$$x = u \cdot 2^p$$

then: $\log_2 x = \log_2 u + p$

Logarithms to other bases are then found by multiplying by the proper constant.

5.3 INTERCOM 1000 (D. P.) LOGARITHMS

FLOW CHART



Bendix Computer — Los Angeles 45, California

G-15D — Prepared by Alfred J. da Costa. — Page 1 of 5 — Date: 6 Mar. 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Logarithms Subroutine, — Line 19.

L	P	T or L_K	N	C	S	D	BP	NOTES
								(Entry for $Log_2 x$.)
08	s	26	42	0	19	28		Sub-exit command = (19.26) ---▷ AR
								(Entry for $Log_e x$)
17	s	36	42	0	19	28		Sub-exit command = (19.36) ---▷ AR
42	.	62	71	0	28	19		Sub-exit command = (AR) —▷ 19.62
								(Entry for $Log_{10} x$.)
71	.	72	74	4	21	25		
74	.	75	78	0	25	28		Sign and most significant digits of mantissa —▷ AR
78	.	79	80	3	28	28		Reverse sign
80	.	82	82	0	22	31		Argument $x > 0$? {(AR) < 0 ?}
82	s	26	26	4	20	31		($x \leq 0$) Error Exit
83	s	84	86	0	30	28		($x > 0$)
86	.	88	89	3	19	29		
88								.0000081
89	.	90	93	1	28	21		Exponent —▷ 21.02
93	.	02	96	0	26	31		u/4
96	.	98	02	4	19	26		
98								.3z9xz40 } $\sqrt{2}/4$
99								-.5u82799 }
02	.	04	10	4	26	24		
10	.	12	16	4	25	30		
16	.	20	27	5	26	25		$(u + \sqrt{2}) / 4$
27	.	u 32	34	7	24	30		
34	.	36	38	5	26	21		
38	.	40	43	4	21	26		$(u - \sqrt{2}) / 4$

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

PROGRAM PROBLEM: Intercom 1000 (D.P.) LOGARITHMS SUBROUTINE.

Prepared by Alfred J. da Costa

Page 2 of 5

Date: 6 Mar. 59.

Line 19.

L	P	T or L_K	N	C	S	D	BP	NOTES
43	s	v6	52	5	25	31		$v = (u - \sqrt{2}) / (u + \sqrt{2})$
52		56	58	4	24	21		
58		60	65	4	21	25		
65		v4	76	0	24	31		
76		80	84	4	26	22		v^2
84		85	90	0	19	28		Pick-up command ——▷ AR
90		94	97	4	19	25		
94								.zz0z8x8 } C_5
95								.0uy7388 }
								(Loop I: Polynomial Evaluation.)
97	.	u0	00	4	22	24		v^2
00	.	02	02	0	31	31		
+85	s	30	33	4	19	26		Pick-up command executed from AR
33	.	v4	44	0	24	31		
44	.	46	50	4	26	25		
50	.	56	60	0	19	29		
56								.780000 0
60	.	72	72	0	22	31		(AR) $\leq$ 0?
72	s	81	97	3	19	29		($\geq$ 0) "T" of pick-up cmd. is reduced by 2.
81								.8000000
								(Exit from LOOP I.)
73	s	76	79	4	21	24		($<$ 0) v ——▷ MQ
79	.	v4	87	0	24	31		
87	.	88	91	4	26	26		

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15 D

Prepared by Alfred J. da Costa

Page 3 of 5

Date: 6 Mar. 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Logarithms Subroutine.

Line 19.

L	P	T or LK	N	C	S	D	BP	NOTES
91		03	09	0	19	30		Add 1/16
03								.1000000 = 1/16
09		10	13	4	26	24		
13		06	20	0	26	31		Multiply by 8
20		22	24	1	21	28		exponent
24		28	28	0	22	31		x<1?
28	s	32	59	5	24	21		(x>1) Skip to 59.
.29	s	.30	.35	.4	.25	.25		(x<1)
35		36	40	7	24	30		
40		42	47	5	26	26		
47		48	51	5	26	21		
51		02	55	0	26	31		
55		56	59	1	28	28		
59		60	61	0	27	21		Merge characteristic and mantissa
61		63	64	3	19	28		
63								.0000007
64		12	25	0	19	20		
12								.0000001
								(LOOP II: Addition of powers of 2.)
25		28	41	4	21	25		
41		02	49	0	26	31		
49		52	53	0	31	27		(31.00) ≠ 0?
53	s	54	92	0	28	28		(=0) Skip to 92
54	s	81	92	0	19	26		(≠0) Add 1/2.
81								.8000000

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by Alfred J. da Costa

Page 4 of 5

Date: 6 Mar. 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Logarithms Subroutine.

Line 19.

L	P	T or LK	N	C	S	D	BP	NOTES
92	b	94	01	4	25	30		
01	b	02	04	0	28	27		(AR) ≠ 07
05	s	08	25	4	26	21		(≠0)
								(Exit from Loop II.)
04	s	08	11	4	26	24		(=0)
11	b	18	19	3	19	28		
18								.0000087
19	b	98	37	0	27	31		
37	b	38	39	1	28	28		
39	b	40	45	4	24	21		
45	b	u 49	57	0	19	20		
46								-.zzzzz00
48								-.zzzzz00
57	b	60	62	0	27	21		
								(Exit for Log. to hthe base 10.)
62	b	66	70	4	19	25		$Log_{10}2$ ——▷ ID (This command may be replaced by command 26 or 36. (See cmds 08, 17)
66								.9z8007z } $Log_{10}2$
67								.9u209u8 }
								(Exit for Log. to the base e.)
36	s	68	70	4	19	25		
68								.zu3vx80 } $Log_e 2$
69								.v17217z }
70	b	75	77	0	19	23		
75	s	21	21	0	20	31		
77	s	80	80	4	20	31		

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by Alfred J. da Costa.

Page 5 of 5.

Date: 6 Mar. 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Logarithms Subroutine.

Line 19.

L	P	T or L_K	N	C	S	D	BP	NOTES
								(Exit for Logarithms to base 2.)
26	s	01	01	0	20	31		
								(Polynomial COnstants .)
06								.95z6y40 } C_1
07								.5w551x9 }
14								.2613xz0 } C_2
15								1yw709x }
22								803w510 } C_3
23								-127777z }
30								.x9v77y8 } C_4
31								.0x2w6wx. }

FORM BCD 108-17.8

MEMORY ALLOCATION CHART

Problem { INTERCOM 1000 (Double Precision) Page 1 of 1
LOGARITHMIC SUBROUTINE Date Mar.6, 59.

Prepared by Alfred J. da Costa. Check Sum .1000603 Line 19

U7	U6	U5	U4
1000603	3u9vvuu	5930619	
U3	U2	U1	U0
99	98	97	96
-5u82799	3z9xz40	-y4002x8	-y20227u
95	94	93	92
uy7388	zz0z8x8	26035z	-xy0133y
91	90	89	88
830927y	-xy61279	5v5x795	81
87	86	85	84
-5u5v35u	x859y7x	-9y2127u	565u27w
83	82	81	80
55563xw	-1u1u29z	8000000	52522xz
79	78	77	76
725731z	5050z9w	-505029z	-x054356
75	74	73	72
151529z	4w4y33w	-ww4z2v8	x161y7x
71	70	69	68
-4u4u2v9	wv4x277	v17217z	zu3vx80
67	66	65	64
9u209u8	9z8007z	724w31z	8w19274
63	62	61	60
7	-w246279	vz40y7w	48482xz
59	58	57	56
3x3x375	-vw412v9	vw3y375	7800000
55	54	53	52
393v79w	x15w27u	375w39w	-v83u315
51	50	49	48
23735z	v83w27x	v4353zv	-zzzzz00
47	46	45	44
-3233755	-zzzzz00	3139274	-uy32359
43	42	41	40
-743473z	vy47393	23135z	-uu2z75u
39	38	37	36
-2u2x315	-u82v2vu	272779w	-w446279
35	34	33	32
-2628z1y	-u426755	722w31z	
31	30	29	28
x2w6wx	x9v77y8	-2023339	-u03v715
27	26	25	24
-2022z1y	10129z	-9w292v9	1w1w2xz
23	22	21	20
-127777z	803w510		96186vw
19	18	17	16
622537z	87	u42u27w	-941v759
15	14	13	12
1yw709x	2613xz0	61435z	1
11	10	09	08
9213y7w	-8w1033y	- w0x358	9u2u27w
07	06	05	04
5w551x9	95z6y40	-8819355	-880v358
03	02	01	00
1000000	-840u358	30439v	2023zz

Four-Word Lines

Line	3	2	1	0

6. Exponential

6.1 Specifications

ENTRY: Loc. 72 for $\log_{10}x$ (word position to be used in "Perform Subroutine" Command, - 72)
Loc. 22 for $\log_{e}x$ (word position to be used in "Perform Subroutine" Command, - 22)
Loc. 08 for $\log_{2}x$ (word position to be used in "Perform Subroutine" Command, - 08)

EXIT: Transfer to L. 00.01. A return command can then be inserted either in loc. 89 or 79 of this subroutine. (See coding sheets).

ERROR EXIT: There is no error exit in the subroutine itself.

ERROR INDICATION: If $n > 10^{38}$, the multiplication subroutine will set up the error indication.

DATA INPUT: 21.00,01 ("A" Register)

DATA OUTPUT: 21.00,01 ("A" Register)

RANGE OF ARGUMENT: $-10^{-32} \times 10^{38}$ (approximately)
If x -10^{-32}, the output is made 0.
If x 10^{38}, error exit.

EXECUTION TIME: Approximately 25 drum cycles.

ACCURACY: Correct to at least 11 significant figures.

SHORT LINE
LOCATIONS USED: Line 20, words 0, 1, 2, 3
Line 21, words 0, 1
Line 22, word 1
Line 23, word 3

LOCATIONS NOT USED IN LONG LINE: 04, 26, 28, 39, 46, 51, 56, 65, 71, 96, u0-u7.

CHECK SUM: 1000604 (Appendix I)
1001504 (Appendix II)

REMARKS: Since locations u0-u7 are not used, this subroutine imposes no restriction on the use of the index registers, and it may be loaded by typing 4 CH u0 instead of 4 CH 00.

APPENDIX II
VERSION: (1) Change source or destination 19 to 05 in the following commands: 02, 03, 08, 09, 14, 22, 29, 40, 47, 53, 62, 64, 73, 74, 75, 84, 85, 90, 93.

(2) Change characteristic 6 to 5 in commands 59 and 67.

6.2 Method

Where e^x or 2^x is required, the argument x is first multiplied by $\log_{10}e$ or by $\log_{10}2$. The problem then becomes, in all cases, one of evaluating a power of 10.

The argument is reduced to a fixed point number u, and 10^u is evaluated by the following polynomial:

$$10^u = (C_0 + C_1u + C_2u^2 + \ldots + C_{10}u^{10})^2,$$

which is valid for $0 \leq u < 1$.

Since $x = u \cdot 2^p$, (p integral) we have $10^x = (10^u)^{2p}$.

Hence, for values of x outside of the basic range, 10^u is squared p times.

If x is negative, the reciprocal is calculated.

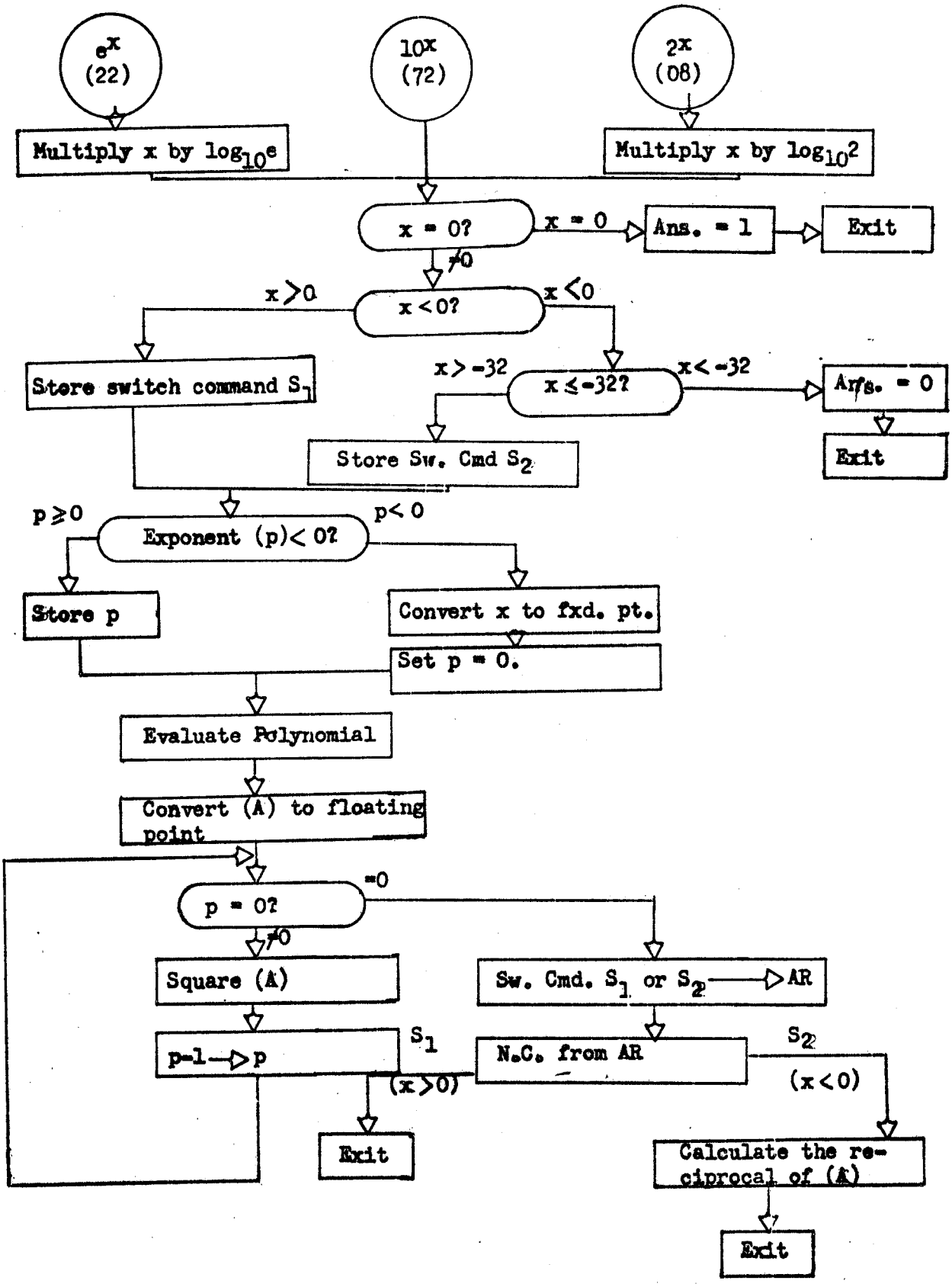
The following constants are used:

	Decimal			Sexadecimal	(Scaled 2^{-4})
C_0	1.0000	0000	00000	1000000	0000000
C_1	1.1512	9254	64923	-126vw1v	76u0430
C_2	.6627	3726	42412	0u9u926	77830x8
C_3	.2543	3481	46386	0411w16	3y3684w
C_4	.0732	0353	62839	-012vx77	1494z0z
C_5	.0168	5525	60848	00450u0	8v4ww27
C_6	.0032	3572	29645	000x40y	x89z738
C_7	.0005	2924	84557	00022uz	ul9y0wu
C_8	.0000	7978	19707	-000053u	90y8z17
C_9	.0000	0746	74589	000007x	90zvw7z
C_{10}	.0000	0202	15778	-0000021	x538983

(With acknowledgment to Mr. E.T. Fedrighi, Dept. 470, Bendix Radio Division)

6.3 INTERCOM 1000 (D. P.) EXPONENTIAL SUBROUTINE

FLOW CHART



Bendix Computer — Los Angeles 45, California

G-15 D

Prepared by Alfred J. da Costa.

Page 1 of 5.

Date: 30 Mar. 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Exponential Subroutine,

Line 19.

L	P	T or L_K	N	C	S	D	BP	NOTES
								Entry for 2^x
08	s	.32	.47	.4	.19	.25		
.32								.9z7v57z } $\log_{10}2$
.33								.9u209u8 }
								Entry for e^x
22	s	.44	.47	.4	.19	.25		
.44								.26x357z } $\log_{10}e$
.45								-.xy5vx8u }
47	s	.50	.62	.0	.19	.20		
.50								-.zzzzz00
.62	.	.67	.68	.0	.19	.23		Return command ⟶ 23.03
.68	.	.80	.80	.4	.20	.31		To floating point multiplication
.67	s	.92	.92	.6	.20	.31		(Ret. Command) N.C. 19.72 (read at T = u7)
								Entry for 10^x
.72	s	.73	.74	.0	.21	.27		$x \neq 0$?
.74	s	.76	.89	.4	.19	.21		$(x = 0)$ $1 \longrightarrow A$
.76								.0000081 } 1
.77								.8000000 }
.89	s	.01	.01	.0	.20	.31		Exit
.75	s	.76	.80	.0	.21	.28		$(x \neq 0)$
.80	.	.85	.85	.0	.22	.31		$x < 0$?
.85	s	.89	.11	.0	.19	.22		$(x > 0)$ Store switch command S_1 (= Exit Command)

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 2 of 5

G-15D

PROGRAM PROBLEM: Intercom 1000 (D.P.) Exponential Subroutine.

Prepared by Alfred J. da Costa.

Date: 30 Mar. 59.

Line 19.

L	P	T or L_K	N	C	S	D	DP	NOTES
36	s	.38	.90	.2	.28	.21		(x < 0)
90		.92	.94	.3	.19	.28		
92								.0000086
94		.96	.98	.0	.30	.29		
98		.02	.02	.0	.22	.31		Characteristic < 6 ?
02		.04	.26	.0	.19	.21		(≥ 6) 0 → A {x > 32}
04								.0000001
26		.29	.89	.0	.29	.21		
89		.01	.01	.0	.20	.31		Exit
03		.09	.11	.0	.19	.22		(< 6) Store Switch and S_2 = (19.09)
.11	.	.12	.14	.0	.30	.28		
.14	.	.15	.16	.3	.19	.29		
.15								.000007z
.16	.	.20	.20	.0	.22	.31		p = (AR) < 0 ?
.20	s	.23	.40	.0	.28	.20		(≥ 0) Store p → 20.03 (N.C. 40)
.21	s	.24	.27	.4	.21	.25		(< 00)
.27	.	v6	.35	.0	.26	.31		Convert to fixed-point fraction
.35	.	.36	.38	.5	.25	.21		
.38	.	.39	.40	.0	.29	.20		Set p = 0
.40	.	.60	.64	.4	.19	.25		C_{10} → ID
.64	.	.84	.87	.0	.19	.28		Coefficient selection command → AR
								LOOP TI Loop I
.87	.	.88	.91	.4	.21	.24		x → MQ
.91	.	.93	.93	.0	.31	.31		

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by Alfred J. da Costa.

Page 3 of 5.

Date: 30 Mar. 59.

PROGRAM PROBLEM: Intefcom 1000 (D.P.) Exponential Subroutine.

Line 19.

L	P	Ter LK	N	C	S	D	BP	NOTES
84	u	56	57	4	19	26		C_1 —— PN (executed from AR)
57		v4	69	0	24	31		
69		70	73	4	26	25		
73		81	83	3	19	29		
81								.0600000
83		87	87	0	29	31		Overflow ? (Exit from loop if set)
								Exit from Loop I
88	s	90	93	4	25	24		
93		95	97	3	19	28		
95								.0000084
97		98	05	0	27	31		Normalize
05		06	10	1	28	28		
10		12	17	5	24	21		
17		20	23	0	27	21		
								LOOP II v/ II
23		24	29	4	21	25		
29		34	41	0	19	20		-zzzzz00 = (19.34) 20.02
41		43	52	0	20	27		p ≠ 0 ?
52	s	53	70	0	22	28		(= 0) Switch command S_1 or S_2 ⟶ AR
70		72	72	0	31	31		N.C. from AR (Either exit or 09 below)
53	s	59	68	0	19	23		(≠ 0) Ret. command ⟶ 23.03
68		80	80	4	20	31		To floating point multiply
34								-.zzzzz00
59	s	78	78	6	20	31		(Ret command) N. C. 10.58 (read at T = u7)
58	s	59	63	3	20	28		-p ⟶ AR

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 4 of 5

G-15 D — Prepared by Alfred J da Costa. — Date: 30 Mar. 59.

PROGRAM PROBLEM: Intercom 1000 (Double Precision) Exponential Subroutine — Line 19.

L	P	T or LK	N	C	S	D	BP	NOTES	
63		.02.	66.	0.	26.	31		-p + 1 ⟶ AR (complemented)	
66		.68.	99.	1.	28.	28		Recomplement	
99		.u3.	23.	2.	28.	20		p = AR ⟶ 20.u3	
								(End of Loop II)	
								Final Commands for x < 0	×
09		.76.	78.	4.	19.	21		(Switch S_2) 1 ⟶ A	
78		.79.	82.	0.	19.	23		Exit command ⟶ 23.03	To obtain
82		.84.	84.	4.	20.	31		To floating point division	reciprocal
79		.21.	21.	0.	20.	31		(Exit command) (read at T = u7)	

FORM BCD 198-17.0

Bendix Computer

Los Angeles 45, California

Page 5 of 5

G-15D

Prepared by Alfred J. da Costa.

Date: 30 Mar. 59.

PROGRAM PROBLEM: Intercom 1000 (Double Precision) Exponential Subroutine Line 19.

L	P	T or Lk	N	C	S	D	BP	NOTES
								Polynomial Constants.
.00								.0000000
.01								.1000000
.06								.76u0430
.07								-.126vv1v
.12								.77830x8
.13								.0u9u926
.18								.3y3684w
.19								.0411w16
.24								.1494z0z
.25								-.012vx77
.30								.8v4ww27
.31								.00450u0
.36								.x89z738
.37								.000x40y
.42								.u19y0wu
.43								.00022uz
.48								.09y8z17
.49								-.000053u
.54								.90zvw7z
.55								.000007x
.60								.x538983
.61								-.0000021

FORM BCD 108-17.0

MEMORY ALLOCATION CHART

Problem { Intercom 1000 (Double Precision) Page 1 of 1
Exponential Subroutine. Date 3/14/59.

Prepared by Alfred J. da Costa. Check Sum .1000604 Line 19.

U7	U6	U5	U4
1000604	-7v896y4	5933019	
U3	U2	U1	U0
99	98	97	96
y717v94	2022xz	620537z	
95	94	93	92
84	y0623xx	xz61y7w	86
91	90	89	88
5x5x3zz	xw5yy7w	10129z	-xu5x338
87	86	85	84
-5u5v2v8	x85uv95	x90v276	-383927u
83	82	81	80
57573vz	-545429z	600000	55552xz
79	78	77	76
151529z	5052277	8000000	81
75	74	73	72
4x502vw	-ww59275	x153y7x	4u4u2vv
71	70	69	68
	48483zz	-4849359	-505029z
67	66	65	64
-5w5wu9z	w46379w		x45727w
63	62	61	60
24235z	w344277	- 21	x538983
59	58	57	56
-4y4yu9z	3w3zy9w	724531z	
55	54	53	52
7x	90zvw7z	vv44277	36462xw
51	50	49	48
	-zzzzz00	- 53u	9y8z17
47	46	45	44
v23y274		-xy5vx8u	26x357z
43	42	41	40
22uz	u19y0wu	uv3429v	-vw40279
39	38	37	36
	28283v4	x40y	x89z738
35	34	33	32
-2626735	-zzzzz00	9u209u8	9z7v57z
31	30	29	28
450u0	8v4ww27	u229274	
27	26	25	24
742335z	9x593v5	- 12vx77	1494z0z
23	22	21	20
-1u1x2v9	-uw2z279	-981v2v9	9728394
19	18	17	16
411w16	3y3684w	9417375	14142xz
15	14	13	12
7z	1010y7x	u9u926	77830x8
11	10	09	08
x0y3xw	-8w11715	-ww4y275	-u02z279
07	06	05	04
-126vv1v	76u0430	70u79w	1
03	02	01	00
890v276	841u275	1000000	

Four-Word Lines

Line	3	2	1	0

7. Sine, Cosine Subroutines

7.1 Specifications

ENTRY: Loc. 39 for sine, degrees.
Loc. 23 for cosine, degrees.
Loc 42, for sine, radians.
Loc. 26, for cosine, radians.

EXIT: (Loc. 93) to 00.01.
(Loc. 61) to 04.63 (Normalize) (See coding sheets)

ERROR EXIT: None

DATA INPUT: Angle (θ) in degrees or radians in 21.00,01

DATA OUTPUT: Sine θ or cosine 0 in 21.00,01

EXECUTION TIME: Approximately 18 drum cycles.

ACCURACY: Error $< 10^{-12}$

SHORT LINE
LOCATIONS USED: Line 20, words 00, 01, 02
Line 21, words 00, 01, 02, 3
Line 22, word 00.

LOCATIONS NOT
USED IN LONG
LINE: u0-u7 inclusive.

CHECK SUM: 1001605 (Appendix I)
1000505 (Appendix II)

REMARKS: Since these new versions do not use locations u0-u7, they may be entered into memory by typing 5 CH u0 instead of 5 CH 00, and any index register may be used without restriction.

APPENDIX II
VERSION: (1) Change source destination "19" to "05" in the commands: 13, 16, 20, 22, 23, 25, 26, 43, 49, 52, 59, 65, 68, 74, 83.

(2) Change characteristic "6" to "5" in command 71.

7.2 Method

The argument, A, given in degrees or radians, is first transformed to measurement in units of circles with a floating point division by either 360° or 2π radians, respectively. The result is then shifted to a fixed point fraction, eliminating integral numbers of circles which do not affect the sine or cosine function. If the fraction of a circle is negative, it is complemented. If cos A is desired, 1/4 (circle) is added, ignoring overflow. Let the result be called A'.

An argument, u, is determined such that the desired function (sine or cosine) of A equals sin u, where u is expressed as a positive or negative fraction of a quadrant. The most significant bit in A' represents the sign of u. It is brought into the sign position by end carry when |A'| is added to itself. The next most significant bit is then carried into the sign position by a second doubling of absolute value. If this result is +0, then sin u = 0; if the result is -0, then sin u = $\pm$1. (Note that a correct sine or cosine is obtained when the original argument, A, was exactly on the boundary between any two quadrants.)

The overflow indicator is turned off.

If the last addition does not yield $\pm$0, the sign formed by end carry determines whether or not to complement the remaining bits (now shifted two places left from their original position in A') to form |u.|

1/2 sin u is approximated by a 15th order polynomial of the form $u \sum_{i=1}^{8} C_i (-u^2)^{i-1}$, where

$2C_1 = 1.57079\ 63267\ 941$

$2C_2 = .64596\ 40974\ 985$

$2C_3 = .07969\ 26262\ 378$

$2C_4 = .00468\ 17541\ 330$

$2C_5 = .00016\ 04411\ 748$

$2C_6 = .00000\ 35988\ 183$

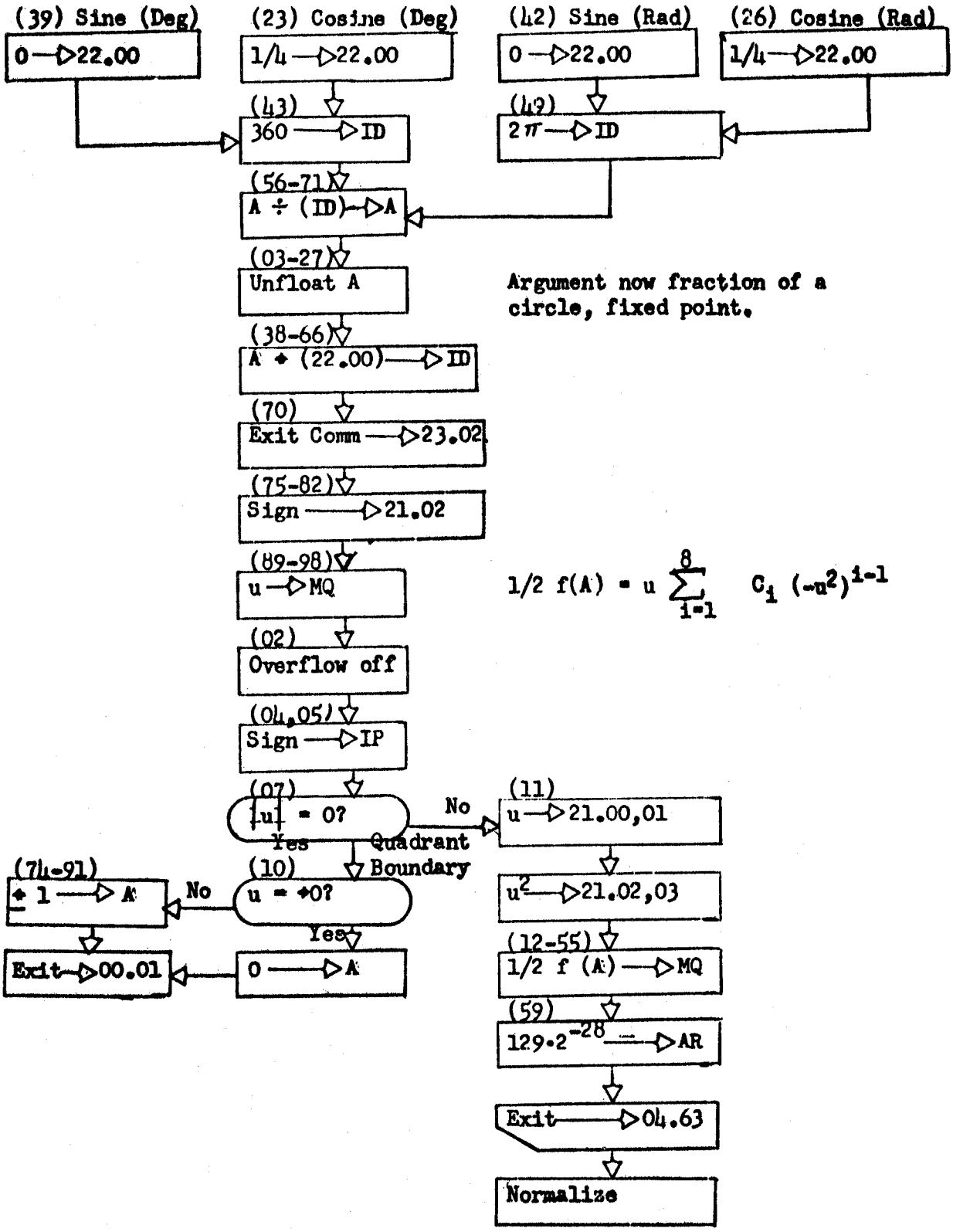
$2C_7 = .00000\ 00568\ 887$

$2C_8 = .00000\ 00006\ 456$

In evaluating the polynomial, time is saved by performing two double precision multiplications in a loop of three drum cycles.

7.3 INTERCOM 1000 (D. P.) SINE, COSINE SUBROUTINE

FLOW CHART



Bendix Computer — Los Angeles 45, California

G-15D

Prepared by S. H. Lewis.

Page 1 of 5

Date: 17 Mar. 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Sine-Cosine Subroutine.

Line 19.

L	P	T or LK	N	C	S	D	BP	NOTES
39	s	.40	43	0	29	22		(Sine, degrees) Clear 22.00
23	s	.40	43	0	19	22		(Cosine, degrees) 1/4 ——▷ 22.00
43	.	.46	56	4	19	25		360 = (19.46,47) ——▷ ID
42	s	.44	49	0	29	22		(Sine, radians) Clear 22.00
26	s	.40	49	0	19	22		(Cosine, radians) 1/4 ——▷ 22.00
49	.	.50	56	4	19	25		2π = (19.50,51) ——▷ ID
56	.	.58	65	0	08	20		Extractor -zzzzz00 = (08.58) ——▷ 20.02
65	.	.71	94	0	19	23		Return Cmd. = (19.71) ——▷ 23.03
94	.	u1	u1	4	21	31		Exit to floating-point division
71	s	.21	23	6	21	31		(Return Command) N.C. 19.03 (from 23.u7)
03	s	.08	13	0	30	28		(30.00) ——▷ AR
13	.	.14	17	3	19	29		.0000080 = (19.14) ——▷ AR+
17	.	.20	25	4	31	25		(31.00,01) ——▷ ID.
25	.	.28	29	0	19	26		Round-off = 2^{-49} = (19.28) ——▷ PN_0
29	.	.31	44	0	22	31		(AR) < 0? {Angle θ < 1/2 circle?}
44	s	.46	96	5	25	30		(≥ 0) {θ ≥ 1/2} (ID) ——▷ PN+
45	s	.98	88	0	26	31		(< 0) {θ < 1/2} Reduce to fixed-point fraction
88	.	.89	90	0	29	28		Clear AR
90	.	.92	96	6	25	30		|ID| ——▷ PN+
96	. w	.98	99	3	23	31		Signif. digits = (PN_0 02) PN_0 {(02.98) = .00001zz}
99	.	u0	00	0	28	27		(AR) ≠ 0?
00	s	.04	38	4	26	21		(=0) {θ < 1} (PN) ——▷ A {N.C. 19.38}
01	s	.04	06	4	26	24		(≠0) {θ ≥ 1} (PN) ——▷ MQ

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by S. H. Lewis.

Page 1 of 5

Date: 17 Mar. 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Sine-Cosine Subroutine.

Line 19.

L	P	T or LK	N	C	S	D	BP	NOTES
39	s	.40	43	0	29	22		(Sine, degrees) Clear 22.00
23	s	.40	43	0	19	22		(Cosine, degrees) 1/4 ——▷ 22.00
43	.	.46	56	4	19	25		360 = (19.46,47) ——▷ ID
42	s	.44	49	0	29	22		(Sine, radians) Clear 22.00
26	s	.40	49	0	19	22		(Cosine, radians) 1/4 ——▷ 22.00
49	.	.50	56	4	19	25		2π = (19.50,51) ——▷ ID
56	.	.58	65	0	08	20		Extractor -zzzzz00 = (08.58) ——▷ 20.02
65	.	.71	94	0	19	23		Return Cmd. = (19.71) ——▷ 23.03
94	.	u1	u1	4	21	31		Exit to floating-point division
71	s	.21	23	6	21	31		(Return Command) N.C. 19.03 (from 23.u7)
03	s	.08	13	0	30	28		(30.00) ——▷ AR
13	.	.14	17	3	19	29		.0000080 = (19.14) ——▷ AR+
17	.	.20	25	4	31	25		(31.00,01) ——▷ ID.
25	.	.28	29	0	19	26		Round-off = 2^{-49} = (19.28) ——▷ PN_0
29	.	.31	44	0	22	31		(AR) < 0? {Angle θ < 1/2 circle?}
44	s	.46	96	5	25	30		(≥ 0) {θ ≥ 1/2} (ID) ——▷ PN+
45	s	.98	88	0	26	31		(< 0) {θ < 1/2} Reduce to fixed-point fraction
88	.	.89	90	0	29	28		Clear AR
90	.	.92	96	6	25	30		|ID| ——▷ PN+
96	. w	.98	99	3	23	31		Signif. digits = (PN_0 02) PN_0 {(02.98) = .00001zz}
99	.	u0	00	0	28	27		(AR) ≠ 0?
00	s	.04	38	4	26	21		(=0) {θ < 1} (PN) ——▷ A {N.C. 19.38}
01	s	.04	06	4	26	24		(≠0) {θ ≥ 1} (PN) ——▷ MQ

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by S. H. Lewis.

Page 2 of 5

Date: 17n Mar. 59

PROGRAM PROBLEM: Intercom 1000 (D.F.) SINE-COSINE Subroutine.

Line 19.

L	P	T or LK	N	C	S	D	OP	NOTES
06	s	07	09	3	28	28		- (AR) ——▷ AR
09	.	96	27	0	26	31		Shift out whole circles
27	.	28	38	4	24	21		Fractional part of θ ——▷ A
38	.	40	63	6	22	26		(22.00) —V.A▷ PN_1 {0 if sin, 1/4 if cosine}
63	.	64	66	5	21	30		θ = (A) —+▷ PN+ {θ in circles}
66	.	68	70	4	26	25		θ = (PN)—▷ ID. {Clears PN}
70	.	71	75	0	08	23		Command (08.71) ——▷ 23.03
75	u	80	82	6	25	30		2θ (mod. 1) = 2(ID)——▷ PN+ {θ in half-circles}
82	.	86	89	1	26	21		Sign ——▷ 21.02
89	.	90	93	4	26	25		2θ (mod. 1) = (PN)——▷ ID {PN cleared}
93	u	98	98	6	25	30		4θ (mod. 1) = 2(ID)——▷ PN+ {θ scaled to quarter-circles}
98	.	u0	02	5	26	24		u (=θ in quarter-circles) = (PN) —+▷ MQ
02	.	04	04	0	29	31		Reset overflow.
04	s	06	07	0	21	25		(Set) (21.02)——▷ ID (Sign)
05	s	06	07	0	21	25		(Not Set)
07	.	08	10	4	24	27		u ≠ + 0? ——▷
10	s	12	73	1	24	27		(=+0) u ≠ 0 ?
73	s	80	95	4	07	21		(=0) 0 = (07.80,81) ——▷ A ——▷
74	s	86	91	4	19	24		(≠0) 1 = (19.86,87)——▷ MQ
91	.	92	95	4	24	21		+ 1 = (MQ)——▷ A ——▷
95	.	01	01	0	21	31		Exit to 00.01 ◁——
11	s	12	15	4	24	21		(≠ + 0) u = (MQ)——▷ A ◁——
15	.	16	21	4	21	25		u = (A)——▷ ID
21	.	92	08	0	24	31		Multiply

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

PROGRAM PROBLEM: Intercom 1000 (D.P.) SINE-COSINE Subroutine.

Prepared by S. H. Lewis.

Page 3 of 5.

Date: 17 Mar. 59.

Line 19.

L	P	T or L_K	N	C	S	D	BP	NOTES
08		.10.	12.	5.	26.	21		u^2 = (PN) ——▷ 21.02,03
12		.14.	16.	4.	21.	25		u^2 = (21.02,03) ——▷ ID
16		.18.	20.	4.	19.	24		C_8 = (19.18,19) ——▷ MQ
20		.22.	24.	0.	19.	28		Coefficient Selection Cmd. = (19.22) ——▷ AR
								{Polynomial Evaluation Loop}
24		.26.	26.	0.	31.	31		Obey AR
22	s	.36.	41.	5.	19.	26		C_{21} —♦—▷ PN
41		.v4.	48.	0.	24.	31		Multiply
48		.50.	52.	5.	26.	26		(PN) —♦—▷ PN
52		.53.	54.	0.	19.	29		.602u000 = (19.53) ——▷ AR♦
54		.56.	57.	0.	22.	31		(AR) < 0?
57	s	.58.	64.	4.	26.	24		(≥0) PN ——▷ MQ
64		.66.	68.	4.	21.	25		u^2 ——▷ ID
68		.69.	72.	3.	19.	29		.3400000 ——=——▷ AR♦
72		.74.	74.	0.	31.	31		Obey AR
(AR)		80	83	5	19	26		C_{21+1} —♦—▷ PN
83	s	.84.	85.	1.	19.	29		-2y2u000 —♦—▷ AR♦
85	.	.v4.	92.	0.	24.	31		Multiply
92	.	.94.	97.	5.	26.	24		PN —♦—▷ MQ
97	.	.98.	24.	4.	21.	25		u^2 ——▷ ID
.58	s	.60.	62.	4.	26.	25		(<0) PN ——▷ ID
.62	.	.64.	67.	4.	21.	24		u = (21.00,01) ——▷ MQ
.67	.	.92.	55.	0.	24.	31		Multiply
.55	.	.56.	59.	4.	26.	24		1/2 f (A) = (PN) ——▷ MQ

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by S. H. Lewis.

Page 4 of 5

Date: 17 Mar.59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) SINE-COSINE Subroutine.

Line 19.

L	P	T or LK	N	C	S	D	BP	NOTES
59		.60.	61.	3.	19.	28		.0000081 ——▷ AR
61		.63.	63.	4.	21.	31		Exit to 04.63 {To Normalise}
								CONSTANTS.
14								.0000080
18								.2w5x83z } C_8 = .00000 00003 228
19								.0000000 }
28								.0000100
30								-.441z14u } C_1 = .78539 81633 9705
31								.w90zxuu }
32								-.76w77w5 } C_3 = .03984 63131 189
33								.0u335y3 }
34								-.1994443 } C_5 = .00008 02205 874
35								.000541y }
36								-.455w982 } C_7 = .00000 00284 4435
37								-.0000007 }
40								.4000000 = 1/4
46								.0000089 } 360
47								.v400000 }
50								.442x183 } 2 π
51								.w90zxuu }
53								.602u000
60								.0000081
69								.3400000

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by S. H. Lewis.

Page 5 of [illegible]

Date: 17 [illegible]

PROGRAM PROBLEM: Intercom 1000 (D.P.) SINE-COSINE Subroutine.

Line 12[illegible]

L	P	T or L_K	N	C	S	D	BP	NOTES
.76								-.1256y36 } C_2 = .32298 20487 4925
.77								-.52uyz39 }
.78								-.wy39z31 } C_4 = .00234 08770 665
.79								.0099696 }
.80								-.0wy912y } C_6 = .00000 17994 0915
.81								.00001y3 }
.84								-.2y2u000
.86								.0000081 } 1
.87								.8000000 }
.u3								.9vv995y Sum balancer
.u7								.1001605 Check sum

FORM BCD 108-17.0

MEMORY ALLOCATION CHART

Bendix Computer

Problem { Intercom 1000 (D.P.) — Sine - COsine Subroutine

Page 1 of 1

Date

Prepared by S. H. Lewis.

Check Sum .1001605

Line 19.

U7 1001605	U6	U5	U4
U3 9vv995y	U2	U1	U0
99 650039v	98 -y402758	97 -64182v9	96 y263yzz
95 1012vz	94 -65652vz	93 -6262v3y	92 -xy61758
91 -5y5z315	90 -xw60v3y	89 -5w5x359	88 5u5u3vw
87 8000000	86 81	85 725w31z	84 -2y2u000
83 555567x	82 x659755	81 1y3	80 - wy912y
79 99696	78 -wy39z31	77 -52uyz39	76 -1256y36
75 -5052v3y	74 -x65v278	73 -x05z0z5	72 4u4u3zz
71 -1517uvz	70 484v117	69 3400000	68 4648y7x
67 5w3731z	66 -w446359	65 w75y277	64 -w2442v9
63 -42426vy	62 -w0432v8	61 -3z3z2vz	60 81
59 3x3xy7w	58 -vw3y359	57 -3w40358	56 vu41114
55 -3u3v358	54 38392xz	53 602u000	52 363627x
51 w90zxuu	50 442x183	49 -3438279	48 -v23475u
47 v400000	46 89	45 625835z	44 -uy6073y
43 -uy38279	42 uw313v6	41 723031z	40 4000000
39 292v3v6	38 -u83zuxu	37 - 7	36 -455w982
35 541y	34 -1994443	33 u335y3	32 -76w77w5
31 w90zxuu	30 -441z14u	29 1z2w2xz	28 100
27 -1y26315	26 u831276	25 9w1x27u	24 1u1u3zz
23 u82v276	22 -u42967u	21 5w0831z	20 961827w
19	18 2w5x83z	17 -94193z9	16 -9214278
15 -12152v9	14 80	13 z11y7x	12 -8y102v9
11 - y0z315	10 8w4971v	09 601v35z	08 -8u0w755
07 - u0u31v	06 809z9w	05 7072v9	04 86072v9
03 880x3xw	02 4043vz	01 -8406358	00 -84263[illegible]

Four-Word Lines

Line	3	2	1	0

8. Arc Tangents

8.1 Specifications

ENTRY: Loc. 24

EXIT: Transfer to L. 00.01. A return command can then be inserted in loc. 26 of this subroutine. (See coding sheets)

ERROR EXIT: None

DATA INPUT: Argument in "A" Regsiter (21.00,01)

DATA OUTPUT: Arc tangent x in radians in "A" Register (21.00,01)

RANGE OF ARGUMENT: Any number that can be expressed in Intercom 1000 notation.

RANGE OF OUTPUT: $-1/2\pi < \arctan x < 1/2\pi$

EXECUTION TIME: Approximately 30 drum cycles

SHORT LINE LOCATIONS USED: Line 20, words 0, 1, 2, 3
Line 21, words 0, 1
Line 22, word 1
Line 23, word 3

LOCATIONS NOT USED IN LONG LINE: 13.

CHECK SUM: 1000606 (Appendix I)
1000506 (Appendix II)

APPENDIX II VERSION: (1) Change source or destination "19" to "05" in the commands: 02, 06, 19, 20, 28, 31, 34, 42, 43, 45, 50, 51, 70, 74, 80, 81, 82, 86, 87, 89.

(2) Change characteristic "06" to "05" in the commahds: 03, 47, 90, u3.

(3) Do not change the constants in locations 57, 65. (Restore them if the "repositioner" is used.

8.2 Method

The approximating polynomial used is:

$$\text{Arctan } x = C_0x + C_1x^3 + C_2x^5 + C_3x^7 + C_4x^9 + C_5x^{11} + C_6x^{13} + C_7x^{15}$$

For $0 < x < 1/2$,

Where: $C_0 = .9999\ 9999\ 9988\ 54$

$C_1 = -.3333\ 3332\ 7389\ 69$

$C_2 = .1999\ 9949\ 1589\ 81$

$C_3 = -.1428\ 4047\ 0763\ 90$

$C_4 = .1108\ 3986\ 7038\ 74$

$C_5 = -.0884\ 6897\ 3459\ 21$

$C_6 = .0643\ 1533\ 6790\ 01$

$C_7 = -.0296\ 6739\ 0396\ 20$

The absolute value of the argument is used for the evaluation of the polynomial, the sign being appended later, since x and arctan x have the same sign in the range considered.

If $|x| > 1$, use is made of the identity:

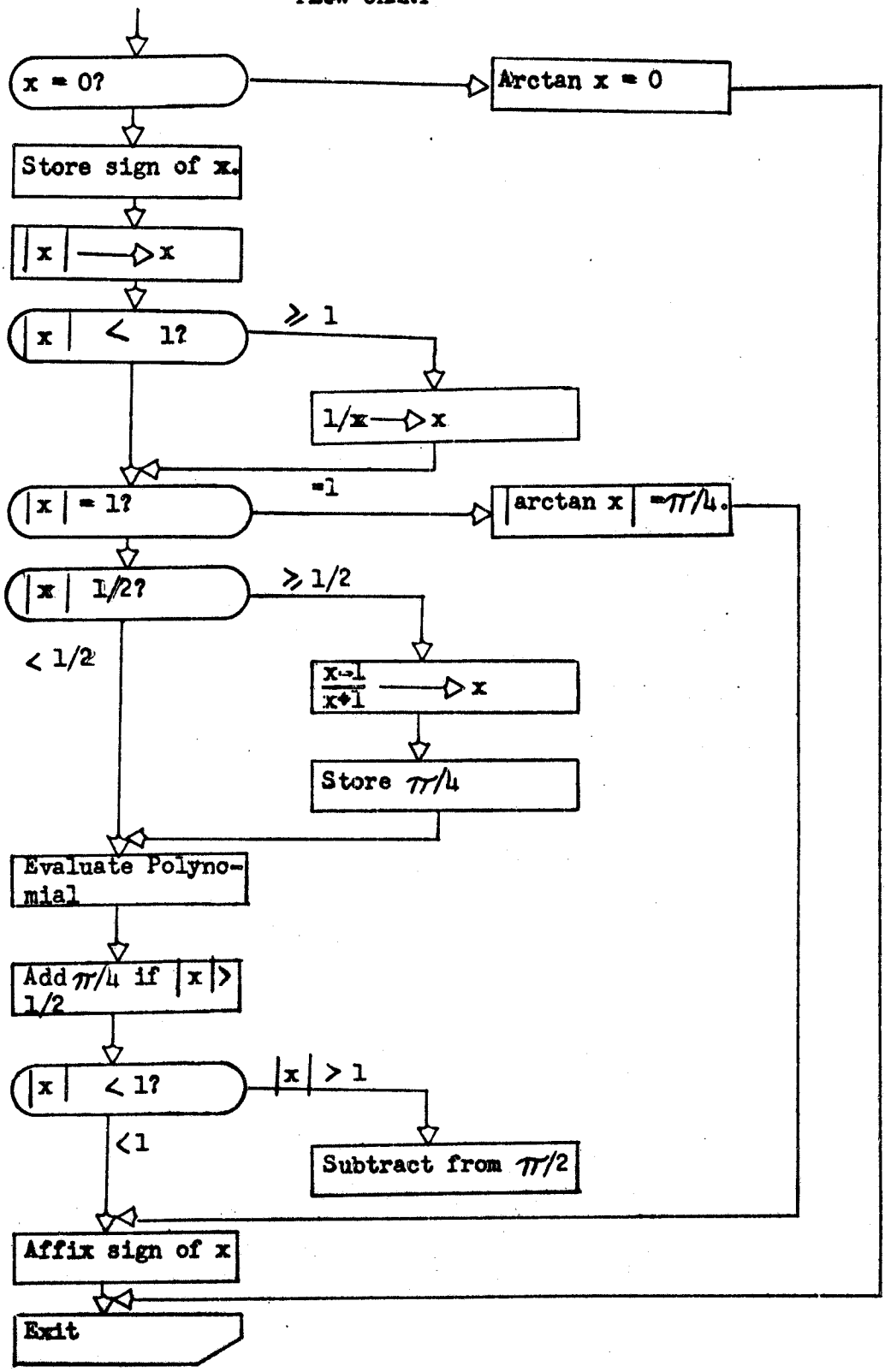
$$\text{Arctan } |x| = 1/2 - \arctan (1/|x|)$$

For values of x between 1/2 and 1, $\frac{x-1}{x+1}$ is substituted for x in the polynomial and $\frac{\pi}{4}$ added to the result.

(Acknowledgment is made to Mr. E.T. Federighi, Dept. 470, Bendix Radio Division, for the constants used)

8.3 INTERCOM 1000 (D. P.) ARC TANGENT X

FLOW CHART



Bendix Computer — Los Angeles 45, California

Page 1 of 5

G-15D

Prepared by Alfred J. da Costa Date: 7-17-58

PROGRAM PROBLEM: Intercom 1000 (Double Precision) Arctangent Subrout. Line 19.

L	P	T or LK	N	C	S	D	BP	NOTES
.24	s	.25.	26.	1.	21.	27		x = 0?
.26	s	.01.	01.	0.	21.	31		(=0) Exit (arctan x = 0)
.27	s	.28.	31.	4.	21.	25		(≠0) x —▷ ID
.31	.	.24.	33.	0.	25.	19		Store sign
.33	.	.36.	15.	4.	21.	24		To make sign positive (Reset IP)
.15	.	.16.	39.	4.	25.	21		|x| —▷ A
.39	.	.40.	45.	0.	30.	28		
.45	.	.76.	50.	3.	19.	29		
.76								.0000081
.50	.	.27.	52.	0.	28.	19		Store exponent -1
.52	.	.54.	70.	0.	22.	31		|x| < 1?
.70	s	.76.	80.	4.	19.	21		{|x| ≥ 1}
.76								.0000081 } 1
.77								.8000000 }
.80	.	u3	92	0.	19.	23		Return command = (19.u3) ——▷ 23.03
.92	.	.58.	12.	0.	08.	20		-zzzzz00 = (08.58) ——▷ 20.02
.12	.	.84.	84.	4.	21.	31		Exit to floating-point division
.u3	s	.91.	91.	6.	21.	31		Return command. (Executed at T = u7.)
								{|x| < 1}
.71	s	.72.	74.	0.	30.	28		
.74	.	.76.	79.	3.	19.	29		
.79	.	.80.	82.	0.	28.	27		|x| = 1?
.82	s	.84.	19.	4.	19.	21		|x| -1} π/4 —▷ A (N.C. p.4)
.84								.4487480 π/4.
.85								.w90zxuu

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D
PROGRAM PROBLEM: Intercom 1000 (D.P.) Arctangent Subroutine.

Prepared by Alfred J. da Costa.

Page 2 of 5
Date: 17 July59.
Line 19.

L	P	T or LK	N	C	S	D	DP	NOTES
83	s	43	u1	0	00	29		{|x| ≠ 1} 0000001 = (00.43) ——▷AR♦
u1	.	u4	88	4	21	25		x = (A) ——▷ID
88	.	89	94	0	28	27		|x| < 1/2? X
								{|x| ≥ 1/2} X
94	s	95	97	0	00	00		
97	.	02	u0	0	26	31		1/2 x
u0	.	u4	u6	4	25	23		1/2 x —▷23.02
u6	.	00	02	4	23	26		1/2 x —▷PN
02	.	77	04	0	19	30		1/2 = (19.77) ——▷PN1♦
04	.	06	10	4	26	25		1/2 (1 ♦ x) ——▷ID
10	.	12	14	7	21	30		Subtract x .·. (PN) = 1/2 (1-x)
14	.	16	22	4	26	26		
22	.	24	25	0	20	24		Set negative sign in IP.
25	.	v4	51	5	25	31		Divide: (MQ) = -(1-x)/(1+x)
51	.	84	08	4	19	23		π/4 23.00
08	.	10	13	4	24	25		(x-1)/(x+1) ——▷ID
								{|x| ≤ 1/2}
95	s	98	01	0	26	31		Reduce to fixed point fraction
01	.	04	07	4	29	23		0 ——▷23.00,01
07	.	08	13	4	25	24		x ——▷MQ
13	.	14	17	4	24	23		x ——▷23.02,03
17	.	v4	28	0	24	31		x^2
28	.	30	29	0	19	22		Counter ——▷22.02
30								.0y00000 (Counter = 14)

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by Alfred J. da Costa

Page 3 of 5

Date: 17 July 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) ARCTANGENT Subroutine.

Line 19.

L	P	T or LR	N	C	S	D	BP	NOTES
29		.32	.34	.5	.26	.21		x^2
34		.36	.38	.4	.19	.25		C_7 ——▷ ID
.36								.z04w8xw } C_7
.37								.0798483 }
								(LOOP)
38		.40	.42	.5	.21	.24		x^2 ——▷ MQ
42		.43	.44	.0	.19	.28		Coefficient sel cmd = (19.43) AR
44		.46	.46	.0	.22	.29		Counter ——▷ AR+
46		.48	.48	.0	.31	.31		
43		.52	.69	.5	.19	.26		(Selection cmd) C_i ——+——▷ PN
69		v4	.78	.0	.24	.31		
78		.80	.89	.5	.26	.25		
89		.96	.98	.3	.19	.28		
.96								.0200000
98		u2	u2	.0	.22	.29		
u2		u3	u4	.0	.28	.27		Counter = 0? (Exit when 0)
u3		.02	.38	.0	.28	.22		Modified counter ——▷ 22.02
								(Exit from loop.)
u4		u6	.05	.4	.23	.24		x ——▷ MQ
.05	u	.08	.09	.0	.29	.26		
.09		v4	.16	.0	.24	.31		
.16		.18	.20	.4	.26	.24		f (x) ——▷ MQ
.20		.21	.23	.3	.19	.28		
.21								.0000080
.23		.98	.53	.0	.27	.31		Normalize

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15 D

Prepared by Alfred J. da Costa.

Page 4 of 5.

Date: 17 July 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Arctangent Subroutine

Line 19.

L	P	T or LK	N	C	S	D	BP	NOTES
53		.54	.68	.1	.28	.28		
68		.72	.75	.4	.24	.21		f (x) ——▷A
75		.76	.87	.0	.27	.21		
87		.47	.91	.0	.19	.23		Ret. Cmd. ——▷23.03
91		.58	.93	.0	.08	.20		-zzzzz00 = (08.58) ——▷20.02
93		.96	.99	.4	.23	.25		0 or π/4 ——▷ID.
99		.79	.79	.4	.21	.31		To floating point addition
47	s	.26	.26	.6	.21	.31		Ret. Cmd. (Read at T = u7)
06	s	.27	.11	.0	.19	.28		Exponent ——▷AR
11		.13	.18	.0	.22	.31		{|x| < 1?} X
								{|x| > 1} X
8	s	.48	.81	.4	.21	.25		f (x) = (A) ——▷ ID
81		.48	.72	.4	.19	.21		1/2π ——▷A
.48								.4487481 } π 1/2
.49								.w90zxuu }
72		.58	.86	.0	.08	.20		-zzzzz00 = (08.58) ——▷20.02
86		.03	.32	.0	.19	.23		Ret. Cmd. ——▷23.03
32		.77	.77	.4	.21	.31		To floating point subtraction {1/2π - f (x)}
03	s	.39	.39	.6	.21	.31		Ret. Cmd. (Read at T = u7)
19	s	.24	.35	.0	.19	.28		Sign ——▷AR
.35		.37	.40	.0	.22	.31		Original x < 0?
.40	s u	.00	.26	.0	.00	.00		{x ≥ 0 (Exit)
.41	s	.44	.73	.3	.21	.28		{x < 0 - (A) ——▷AR
.73		.76	.26	.1	.28	.21		(AR) ——▷A (Exit)

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by Alfred J. da Costa.

Page 5 of 5.

Date: 17 July 59.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Arctangent Subroutine.

Line 19.

L	P	T or LK	N	C	S	D	DP	NOTES
								Constants
.54								-.zy7x980 } C_0
.55								-.zzzzzzs
.56								-.79x2440 } C_1
.57								-.5555553
.58								-.730zy20 } C_2
.59								-.33332uu
.60								-.wz40w60 } C_3
.61								-.2491316
.62								-.80x1590 } C_4
.63								.1w60006
.64								-.76z9830 } C_5
.65								.16u5y71
.66								-.849x 90 } C_6
.67								-.1076z84
.68								-.w84v315 } C_7
.69								.724y31z

FORM BCD 108-17.0

Bendix Computer

MEMORY ALLOCATION CHART

Problem: Intercom 1000 (D.P.) — ARCTAN A.

Page 1 of 1.

Date 4/2/59.

Prepared by Alfred J. da Costa.

Check Sum .1000606

Line 19.

U7: 1000606	U6: -80022zu	U5: 8226396	U4: -yu052z8
U3: -5v5vuvz	U2: 686839v	U1: -y8582v9	U0: -y86u337
99: -4z4z2vz	98: y6662xx	97: 26435z	96: 200000
95: 620135z	94: 6061000	93: -y0632z9	92: vu0w114
91: vu5x114	90: -3433u9z	89: y062y7w	88: 5u5y39v
87: uz5v277	86: 8320277	85: w90zxuu	84: 4487480
83: uv6501x	82: -x413275	81: -v048275	80: y75w277
79: 515239v	78: -x059759	77: 8000000	76: 81
75: 4x57375	74: ww4zy7x	73: ww1u795	72: vu56114
71: 494u3xw	70: -ww50275	69: 724y31z	68: -w84v315
67: -1076z84	66: -849xy90	65: 16u5y71	64: -76z9830
63: 1w60006	62: -80x1590	61: -2491316	60: -wz40w60
59: -33332uu	58: -730zy20	57: -5555553	56: -79x2440
55: -zzzzzzz	54: -zy7x980	53: 374479w	52: 36462xz
51: -x408277	50: 9v34393	49: w90zxuu	48: 4487481
47: -luluuvz	46: 30303zz	45: ww32y7x	44: uy2y2xx
43: -v44567u	42: 2w2w27w	41: uw49yvw	40: 1u000
39: 292x3xw	38: -u82u6v8	37: 798483	36: z04w8xw
35: 25282xz	34: -u426279	33: -u40z2v8	32: -4x4x2vz
31: 9821333	30: y00000	29: -u022755	28: 9y1x276
27: -1y1z2v9	26: 1012vz	25: -723373z	24: 1u1u6vv
23: 623537z	22: 9819298	21: 80	20: 1617y7w
19: 982327w	18: -v0512v9	17: 721w31z	16: -9214358
15: -1227335	14: -901635u	13: -1011317	12: -54542vz
11: x122xz	10: -8w0yyvy	09: 721031z	08: -8u0x319
07: 10x338	06: 9v0v27w	05: 8093vu	04: -860u359
03: - 727uvz	02: wx0427y	01: -84073v7	00: -3zw3163

Four-Word Lines

Line	3	2	1	0

9. Flex Input

9.1 Specifications

ENTRY: Loc. 17

EXIT: To 00.01 (Command 19.04)

ERROR EXIT: None

DATA INPUT: Commands or data in Intercom 1000 notation on paper tape prepared on off-line flexowriter.

DATA OUTPUT: Converted commands or data in 0-15D memory.

EXECUTION TIME: 15 to 20 seconds after read-in of photo-tape.

SHORT LINE
LOCATIONS USED: Line 20, words 0, 1, 2, 3.
Line 21, words 0, 1, 3
Line 23, word 3

LOCATIONS NOT
USED IN LONG
LINE: 14, u0-u7 inclusive.

CHECK SUM: 1000607

REMARKS: There is no Appendix II Version.

The decimal to binary subroutine of Intercom 1000 itself is used for the conversion of numbers.

9.2 Method

1. Two Intercom 1000 commands are required for the execution of this subroutine. The first is "081717," the second is either "k56ADDR" or "k57ADDR" (See manual p. 20).

2. The first command causes the subroutine to be transferred from line 17 to line 19, where it is entered at word 17.

3. Line 04 is copied into line 17, line 19 (the subroutine) into line 04. Control is then transferred to line 04. Line 19 is now free to receive data.

4. The address "ADDR" is converted to binary and stored for future use.

5. Depending upon whether commands or numbers are required, certain commands will be readied for execution as soon as input is read in.

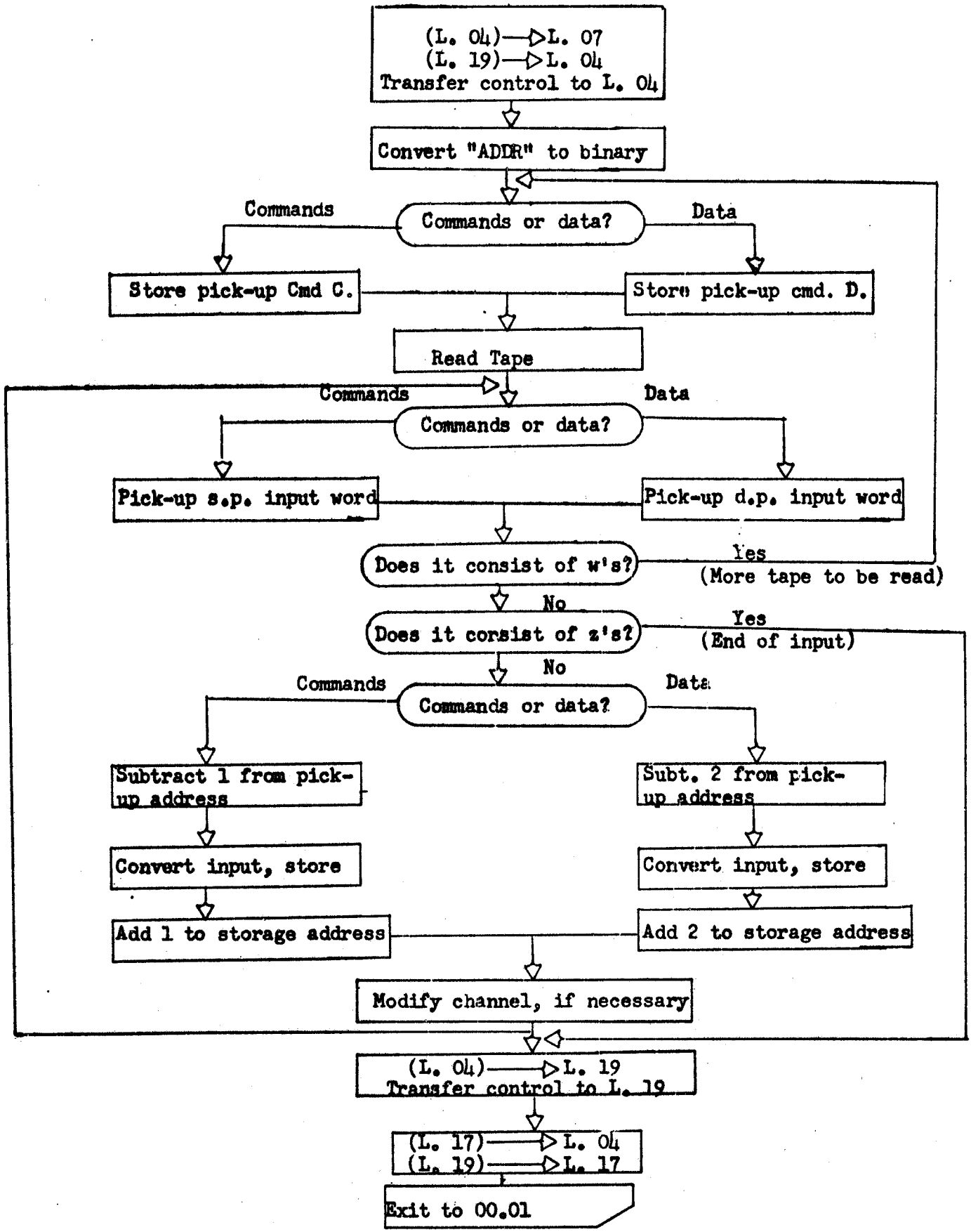
6. Input tape is read in. Each item is converted to binary notation and stored after being scrutinized for one of the codes marking the last entry on the block of tape read in.

7. If the last word consists of w's, another block of tape is read in as in step 6.

8. If the last word consists of z's, the subroutine is copied from line 04 into line 19 and control is transferred to line 19. The original contents of line 04 are copied back into line 04 from line 17, and the subroutine, which is now in line 19, is copied into line 17.

9.3 INTERCOM 1000 (D. P.) FLEXOWRITER INPUT SUBROUTINE

FLOW CHART



Bendix Computer — Los Angeles 45, California

Page 1 of 5.

G-15D — Prepared by Alfred J. da Costa. — Date: 29 Aug. 58

PROGRAM PROBLEM: Intercom 1000 (D.P.) Flexowriter Input Subroutine. — Line 19/04.

L	P	T or LK	N	C	S	D	BP	NOTES
								{Command Line 19}
.17	s	u	.18	.29	0	.04	.17	(L. 04) ——▷ L. 17
.29	.	u	.30	.32	0	.19	.04	Subroutine = (L. 19) ——▷ L. 04
.32	.		.34	.44	0	.19	.28	Return command = (19.34) ——▷ AR
.44	.		.46	.69	0	.28	.00	Return command = (AR) ——▷ 00.46
.69	.		.72	.99	4	.21	.04	(21.00,01) ——▷ 19.72,73
.72								} Temporary storage for
.73								} contents of "A" Register
.99	.		.01	.01	0	.21	.31	Transfer control to line 00.
.34	s		.49	.49	4	.21	.31	(Ret. Cmd) Transfer control to line 04
								{Command line 04}
.49	s		.51	.52	0	.30	.28	
.52	.		.54	.55	0	.28	.21	
.55	.		.58	.58	0	.23	.31	
.58	.		.60	.61	0	.28	.25	Conversion
.61	.		.10	.75	1	.26	.31	to
.75	.		.76	.78	0	.25	.21	binary of
.78	.		.80	.81	0	.30	.28	"ADDR"
.81	.		.82	.84	0	.04	.20	
.82								.7=00000
.84	.		.86	.88	0	.31	.29	
.88	.		.89	.90	0	.28	.21	Converted "ADDR" ——▷ 21.01

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 2 of 5

G-15D

Prepared by Alfred J. da Costa. Date: 29 Aug. 58.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Flexowriter Input Subroutine Line 19/04.

L	P	T or LK	N	C	S	D	BP	NOTES
90	s	91	94	0	31	28		
94	.	u3	u3	0	31	31		
(AR)		00	N	0	00	00		(N = 56 or 57)
								Command conversion (N = 56)
56	s	62	70	4	04	20		Extractors ---- ▷ 20.02,03
.62								.z000000 (→ 20.02)
.63								.0zz0000 (→ 20.03)
.70	.	76	87	0	04	21		Command C (04.76) ---- 21.00 (See p. 3)
								Data Conversion (N = 57)
.57	s	60	65	0	04	21		Command D (04.60) ---- ▷21.00 (See p. 3)
.65	.	68	74	0	04	28		5858000 ---- AR
.68								5858000
.74	.	86	87	0	28	02		5858000 ---→ 02.86
								Tape read-in and search for first word
.87	. u	88	89	0	29	19		Clear L. 19
.89	.	91	91	0	15	31		Read tape
.91	.	91	91	0	28	31		Ready? (Loop begins)
.92	.	93	94	0	04	28		Cmd. 04.93 AR
.94	.	u3	u3	0	31	31		N.C. from AR
.93	s u	00	00	0	19	27		(From AR with N = u3) (19.u4-u7) = 0
.00	s	02	03	0	08	31		(=0) Type AR Precess 4 words
.03	.	91	91	0	00	31		Set Ready
								(Return to 91)

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D
PROGRAM PROBLEM: Intercom 1000 (D.P.) Flexowriter Input Subroutine.

Prepared by Alfred J. da Costa.

Page 3 of 5
Date: 29 Aug. 58.
Line 19 / 04.

L	P	T or LK	N	C	S	D	BP	NOTES
								Exit from precessing loop
01	s	04	05	0	21	28		Cmd. C or D = (21.01) AR
05	.	08	08	0	23	31		Clear
08	.	10	10	0	31	31		N.C. from AR (Command 76 or 60)
								Command Conversion
76	s	u7	13	0	19	28		Cmd C First of next s.p. word = (19.u7) ——▷ AR
13	.	14	16	0	28	24		(AR) MQ_0
								Data Conversion
60	s	u6	11	4	19	24		Cmd. D First or next d.p. word = (19.u6,u7) ——▷ MQ
11	.	12	16	2	24	28		MQ_0 AR
								Check for "w" and "z" codes
16	.	18	19	3	04	29		wwwwwww AR+
18								wwwwwww
19	.	20	22	0	28	27		Word read in = wwwwwww?
22	s	24	50	0	21	28		Yes, more tape to be read Cmd. C or D ——▷ AR
50	.	56	56	0	22	31		(AR) 0? Cmd C is pos., D is neg.
								To 56 if positive; to 57 if neg. (See p. 2)
23	s	24	25	3	04	29		(No) 3333333 AR+
24								.3333333
25	.	27	27	0	29	31		Reset overflow
27	s	29	30	0	28	27		Word read in = zzzazzz?
28	s	29	30	0	28	27		" " " "
								(Yes) To final commands (p. 5)
31	s	32	33	0	21	28		(no) Cmd C or D AR
33	.	35	35	0	22	31		(AR) 0? Is (AR) = Cmd D?

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by Alfred J. da Costa.

Page 4 of 5

Date: 29 Aug. 58.

PROGRAM PROBLEM: Intercom 1000 (D.P.) Flexowriter Input Subroutine.

Line 19/04.

L	P	T or LK	N	C	S	D	DP	NOTES
								{Command Conversion}
.35	s	.37	.39	.3	.04	.29		(> 0) Subtract .0100000 from Cmd. C.
.37								.0100000
.39	.	.40	.41	.0	.28	.21		New Cmd C ——{> 21.00
.41	.	.42	.43	.0	.24	.28		MQ_0 ——{> AR
.43	.	.32	.77	.1	.26	.31		Shift 16 bits left
.77	. u	.80	.83	.2	.24	.21		(MQ_0) —VA—{> 21.03
.83	.	.10	.95	.1	.26	.31		Shift 5 bits left
.95	.	.97	.09	.2	.24	.28		$\|MQ_1\|$ ——{> AR
.09	. u	.12	.15	.0	.27	.28		27.02,03 ——{> AR
.15	. u	.18	.20	.0	.28	.25		Converted Cmd. ——{> ID.
.20	.	.21	.26	.0	.04	.28		Dummy store cmd = (04.21) ——{> AR
.26	.	.29	.94	.0	.21	.29		"ADDR" = (21.01) ——{> AR◆
.94	. .	.u3	.u3	.0	.31	.31		N.C. from AR
.21	s	.00	.12	.0	.25	.00		(Dummy store cmd) "ADDR" Executed from AR after addition of {(ID) ——{> "ADDR"}
.12	.	.37	.42	.0	.04	.28		Address increments ——{> AR {N.C. on p. 5}
								Data Conversion
.36	s	.38	.40	.3	.04	.29		Subtract .0200000 from cmd D
.38								.0200000
.40	.	.44	.45	.0	.28	.21		New cmd D ——{> 21.00
.45	.	.02	.51	.1	.26	.31		Shift input data 1 bit left 23.03
.51	.	.59	.67	.0	.04	.23		Return cmd = (04.59) —> 23.03 (Number Conversion Routine)
.67	.	.70	.70	.2	.21	.31		N.C. .02.70 (To Intercom Number Conversion Routine)
.59	s	.97	.97	.4	.21	.31		(Ret. Cmd) N.C. 04.97
.97	s	.u0	.02	.4	.22	.25		Converted number ——{> ID

Command Conversion

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 5 of 5

G-15D

Prepared by Alfred J. da Costa. Date: 29 Aug. 58

PROGRAM PROBLEM: Intercom 1000 (D.P.) Flexowriter Input Subroutine. Line 19/04.

L	P	T or LK	N	C	S	D	BP	NOTES
.02	.	.04	.06	.0	.04	.28		Dummy store cmd. = (04.04) ——▷AR
.06	.	.09	.10	.0	.21	.29		"ADDR" = (21.01) ——▷AR+
.10	.	.12	.12	.0	.31	.31		N.C. from AR
.04	s	.00	.07	.4	.25	.00		(Dummy Store Cmd) Executed from AR after additions of address.
.07	.	.38	.42	.0	.04	.28		Address increments (0200000) ——▷AR
								{Modification of "ADDR"}
.42	.	.45	.47	.0	.21	.29		"ADDR" = (21.01) ——▷AR+
.47	.	.49	.53	.0	.28	.21		New "ADDR" = (AR) ——▷21.01
.53	.	.54	.71	.3	.04	.29		63zzzzz ——▷AR+
.54								.63zzzzz {Reduces T to 00 and D by 1 when T has reached 100}
.71	.	.79	.79	.0	.22	.31		T = 100?
.79	s	.81	.01	.0	.28	.21		{T = 100} (AR) ——▷21.01
.80	s	.81	.01	.0	.28	.28		{T ≠ 100} Retain (21.01) Return to 01, p. 3
								{Final Commands}
.30	s	.46	.48	.0	.04	.00		Restore original contents of 00.46
.46								3030vzw
.48	u.	.49	.64	.0	.04	.19		Subroutine program = (L. 04) ——▷L. 19
.64	.	.66	.66	.6	.21	.31		N.C. from L. 19
								{Command Line 19}
.66	s	.72	.85	.4	.19	.21		Restore contents of 21.00,01
.85	.	.86	.96	.0	.19	.02		Restore (02.86)
.86								585833v
.96	. u.	.97	.98	.0	.17	.04		Original L. 04 = (L. 17) ——▷L. 04
.98	. u.	.99	.99	.0	.19	.17		Subroutine = (L. 19) ——▷L. 17
.99	.	.01	.01	.0	.21	.31		N.C. 00.01 (Exit)

FORM BCD 108-17.0

MEMORY ALLOCATION CHART

Problem { Intercom 1000 (D.P.) Appendix I Page 1 of 1
Flex Input. Date 4/2/59.

Prepared by Alfred J. da Costa. Check Sum .1000607 Line 19/17/04

U7 1000607	U6 -420x047	U5	U4
U3	U2	U1	U0
99 1012vz	98 6363271	97 -y4022x9	96 6162224
95 y109v1w	94 67673zz	93 27y	92 5y5y09w
91 5v5v39z	90 5w5y3zw	89 5v5v1zz	88 5u5u395
87 58593v3	86 585833v	85 5760262	84 x6583zx
83 u5z75z	82 7z00000	81 5354094	80 520139w
79 x101395	78 x0513xw	77 5053v15	76 yv0x27w
75 4x4y335	74 x657382	73	72
71 4z4z2xz	70 ww57095	69 -w8632u4	68 5858000
67 4646uvz	66 -w855275	65 w44u09w	64 -4242uvz
63 zz0000	62 z000000	61 u4v75z	60 -yu0v278
59 -61612vz	58 vw3x399	57 vw41095	56 -vy46094
55 3u3u2zz	54 63zzzzz	53 3747w9x	52 v637395
51 vv43097	50 38382xz	49 v3343xw	48 3140093
47 v135395	46 3030vzw	45 23375z	44 uy45380
43 204x75z	42 ux2z2vx	41 2v2v31w	40 uw2x395
39 2929395	38 200000	37 100000	36 u628w9x
35 u527w9x	34 -31312vz	33 23232xz	32 u22w27w
31 21212vw	30 uy30080	29 1y20264	28 1y1y39v
27 9x1y39v	26 9x5y2vx	25 1v1v3vz	24 3333333
23 1919w9x	22 98322vw	21 800w320	20 161u09w
19 151639v	18 wwwwwww	17 121x091	16 9213w9x
15 1214399	14	13 z10398	12 u52u09w
11 x10v1w	10 w0w3zz	09 w0z37w	08 u0u3zz
07 u62u09w	06 890u2vx	05 8082zz	04 -8007320
03 5v5v01z	02 840609w	01 84052vw	00 20311z

Four-Word Lines

Line	3	2	1	0

10. Flex Output

10.1 Specifications

ENTRY: The Intercom command 08 1675 causes the routine to be transferred to line 19, and control transferred to word 75.

ERROR EXT: None in this routine.

DATA INPUT: In N/2 consecutive Intercom double precision locations in memory as specified by the second command of the two which is K N ADDR.

DATA OUTPUT: Up to 50 double precision floating point numbers punched on tape in a form to be tabulated by flexowriter.

EXECUTION TIME: Each number requires 3 drum cycles in addition to conversion time for processing. Tape punching time is additional.

SHORT LINE LOCATIONS USED: A format is placed in 02.00-03. Line 04 is used for storage of the routine during execution. Line 19 is used as output storage. In addition to the short line storage used by Intercom, and the Binary to Decimal Conversion routine, the following are used by this routine: 21.02,03, 22.00, 22.01, 22.02.

CHECK SUM: (u7) 1000608

REMARKS: The output format is as follows: sign, two digits, period, five digits, tab, seven digits, tab; sign, two digits, period, five digits, tab, seven digits, carriage return, stop or reload.

Index Register 8 may not be used with the command to use this routine.

10.2 Method

Since this routine requires the use of line 19 for output, it cannot be executed from line 19 while the output information is stored there. It must be executed from some other command line. This is accomplished by interchanging it with line 04 of Intercom using line 19 as intermediate storage. When the output is completed, the lines are switched back again.

This routine uses various parts of Intercom to do some of its work. A pair of Intercom commands must be written to use the routine. The second command contains a number to tell how many numbers are to be punched on tape, and the location of the first one. The

command interpreting, part of Intercom is used to convert this command to a form usable by this routine.

The binary to decimal conversion routine of Intercom itself is used by this routine for the actual conversion of the numbers.

10.3 Detailed Description

1. The Intercom Command 08 1675 causes this routine to be transferred from line 16 to line 19, with the next command taken from word 75 of the routine.

2. Part of Intercom (Line 04) is transferred into line 16, and this routine transfers itself from line 19 to line 04. A special exit to this routine is inserted in the command interpreting part of Intercom at 00.46, and control is transferred back to Intercom.

3. On return to this routine from the Intercom command interpreting part, the command K N ADDR has been converted to binary form giving WD, N and CH. (See page 11 of the Intercom 1000 Double Precision Technical Manual.)

4. Line 19 is cleared. N is extracted from the converted command and shifted (in AR) to the T position (2^{-8}) to be used as a tally. Part of the format is placed in line 02.

5. WD and CH are extracted from the converted command. A dummy is added to form the initial pick-up command. The contents of 22.02, which are used by Intercom are stored temporarily in 04.30, with the initial pick-up command going into 22.02. The balance of the format is placed in line 02, and the initial store command is set up.

6. The tally is reduced by $2 \cdot 2^{-8}$ and its sign tested. If negative, the looping is completed (see 10 below). If positive, the loop is continued.

7. The two-word registers are cleared. The pick-up command is executed bringing a number into the ID. A return command is placed in the AR and control transferred to the binary-to-decimal conversion routine (01.41).

8. The converted number is moved from PN to ID also. The part in the ID is shifted right 1 bit, and the low order part copied into PN_0. The two parts of the PN are copied (single precision) into temporary storage, putting the sign on each part. The number is then picked up in ID and the store command executed. The store command is then modified by subtracting $2 \cdot 2^{-8}$.

9. The pick-up command is modified by adding $2 \cdot 2^{-8}$. A constant of $28 \cdot 2^{-8}$ is added and the sign tested. A positive indicates that the last number picked up had been stored in words 98 and 99 of a channel. The next number should be picked up from

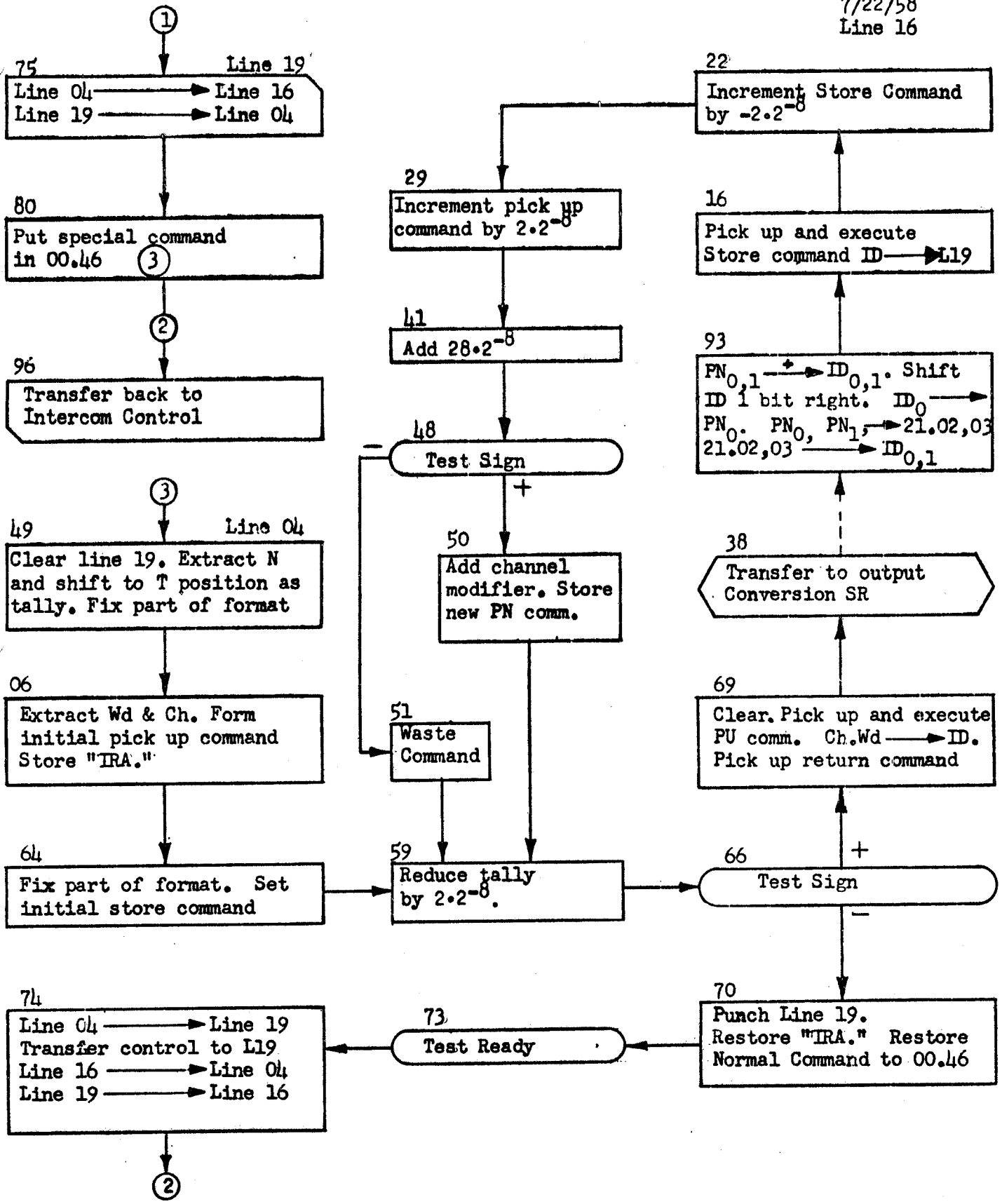
words 00 and 01 of the next channel, so a channel modifier is added to form the new pick-up command. The loop goes back to step 6 (above).

10. Line 19 is punched. The contents of 22.02 is restored, and the normal command is replaced in 00.46.

11. When punching is completed this routine is transferred from line 04 to line 19, and control transferred to line 19. Line 16 (the original line 04 of intercom) is transferred back to line 04. This routine is transferred from line 19 back to line 16, and control is returned to Intercom.

10.4 Intercom 1000 - Flexowriter Number Output Routine

7/22/58
Line 16



Bendix Computer — Los Angeles 45, California

G-15D

PROGRAM PROBLEM: Intercom 1000 - Flexowriter Number Output Routine

Prepared by R.J. Margolin

Page 1 of 3

Date: 7/22/58

Line 16

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
U0	U1	U2	U3
U4	U5	U6	

L	P	T or L_k	N	C	S	D	BP	NOTES
75	U	76	77	0	04	16		Line 04 ——▷ Line 16
77	U	78	80	0	19	04		Line 19 ——▷ Line 04
80		81	83	0	04	28		Spec. Comm. = 04.81 ——▷ AR
(81)	W	48	49	4	21	31		[NC from 04.49]
83		46	96	0	28	00		Spec. Comm. = AR ——▷ 00.46
96	W	98	01	0	21	31		Trans to 00.01
49	U	50	54	0	29	19		Clear L. 19
54		55	57	0	31	28		N = 20.21.03 ——▷ AR
57	U	66	67	2	28	29		AR —┤└—▷ AR◆ 8w.t.
67		68	94	0	28	22		Tally = AR ——▷ 22.00
94		02	06	4	04	02		Format I = 04.02,03 ——▷ 02.02,03
(03)								[8030001–
(02)								0000034◆]
06		07	10	0	30	28		Wd & Ch = 20.21.03 ——▷ AR
10		11	21	0	04	29		Dummy PU Comm. = 04.11 ——▷ AR◆
(11)	W	00	26	4	00	25		[00.00,01 ——▷ $ID_{0,1}$]
21		30	31	0	22	04		IRA = 22.02 ——▷ 04.30
31		34	64	0	28	22		Init. PU Comm. = AR ——▷ 22.02
64		00	08	4	04	02		Format II = 04.00,01 ——▷ 02.00,01
(01)								[0300018◆
(00)								0000110◆]
08		13	59	0	04	22		Initial St. Comm. = 04.13 ——▷ 22.01
(13)	W	U6	22	4	25	19		[$ID_{0,1}$ ——▷ 19.U6.U7]

FORM BCD 108-5.0

Bendix Computer — Los Angeles 45, California

Page 2 of 3

G-15D

Prepared by R.J. Margolin

Date: 7/22/58

PROGRAM PROBLEM: Intercom 1000 - Flexowriter Number Output Routine

Line 16

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
U0	U1	U2	U3
U4	U5	U6	

L	P	T or L_k	N	C	S	D	BP	NOTES
59		60	61	0	22	28		Tally 2^{-8} = 22.00 ⟶ AR ⟵
61		62	63	3	04	29		$2 \cdot 2^{-8}$ = 04.62 ⟹ AR+
(62)								[0200000+]
63		64	66	0	28	22		New Tally = AR ⟶ 22.00
66		68	69	0	22	31		T = AR ⟶ Test
70		72	72	0	10	31		- [Set] Punch L 19
69		72	76	0	23	31		+ [Not Set] Clear
76		14	18	0	22	28		PU Comm = 22.02 ⟶ AR
								NC from AR
(AR)	W	(00)	26	4	(00)	25		D.P. No. = Ch, Wd, Wd+1 ⟶ $ID_{0,1}$
26		27	38	0	04	28		Ret. Comm. = 04.27 ⟶ AR
(27)		93	93	4	21	31		[NC from 04.93]
38		41	41	1	21	31		Transfer to 01.41 (29296VZ)
93		94	97	5	26	25		Conv. No. = $PN_{0,1}$ + ⟶ $ID_{0,1}$
97		02	05	0	26	31		Shift T = 2
05		06	09	0	25	26		Low order = ID_0 ⟶ PN_0
09	U	12	12	0	26	21		PN_0, PN_1 ⟶ 21.02, 21.03
12		14	16	4	21	25		21.02, 03 ⟶ $ID_{0,1}$
16		17	18	0	22	28		St. Comm. = 22.01 ⟶ AR
18		20	20	0	31	31		NC from AR
(AR)	W	(00)	22	4	25	19		$ID_{0,1}$ ⟶ 19.TT
22		23	24	3	04	29		$2 \cdot 2^{-8}$ = 04.23 ⟹ AR+
(23)								[0200000]
24		25	29	0	28	22		New St. Comm = AR ⟶ 22.01
29		30	32	0	22	28		PU Comm. = 22.02 ⟶ AR

FORM BCD 108-5.0

Bendix Computer — Los Angeles 45, California

Page 3 of 3

G-15D

Prepared by R.J. Margolin

Date: 7/22/58

PROGRAM PROBLEM: Intercom 1000 - Flexowriter Number Output Routine

Line 16

0	1	2	3	L	P	T or L_k	N	C	S	D	BP	NOTES
4	5	6	7	32		33	36	0	04	29		$2 \cdot 2^{-8}$ = 04.33 ——▷AR+
8	9	10	11	(33)								[0200000+]
12	13	14	15	36		38	41	0	28	22		New PU Comm. = AR ——▷22.02
16	17	18	19	41		42	48	0	04	29		$28 \cdot 2^{-8}$ = 04.42 ——▷AR+
20	21	22	23	(42)								[1W00000+]
24	25	26	27	48		50	50	0	22	31		T_1.AR ——▷Test - [Set]
28	29	30	31	51		53	59	0	00	00		Waste Comm.
32	33	34	35	50		53	56	0	04	29		Channel Modifier + [Not Set] = 04.53——▷AR+
36	37	38	39	(53)								[8000020-]
40	41	42	43	56		58	59	0	28	22		New PU Comm. = AR——▷22.02 ——▷
44	45	46	47									
48	49	50	51									
52	53	54	55	72		30	45	0	04	22		IRA = 04.30 ——▷22.02
56	57	58	59	45		46	73	0	04	00		Normal Comm. = 04.46——▷00.46
60	61	62	63	(46)		47	48	2	31	28		[Extract]
64	65	66	67	73		73	73	0	28	31		Test Ready
68	69	70	71	74	U	75	78	0	04	19		Line 04 ——▷Line 19
72	73	74	75	78	W	80	82	6	21	31		Transfer control to this routine in line 19
76	77	78	79	82	U	83	84	0	16	04		Line 16 ——▷Line 04
80	81	82	83	84	U	85	96	0	19	16		Line 19 ——▷Line 16
84	85	86	87									
88	89	90	91									
92	93	94	95									
96	97	98	99									
U0	U1	U2	U3	(U6)								[4UW2755+]
U4	U5	U6		(U7)								[1000608+]

FORM BCD 108-5.0

MEMORY ALLOCATION CHART

Problem { Intercom 1000 (D.P.) / Flex Output.

Page 1 of 1.

Date 4/2/59.

Prepared by Robert J. Margolin.

Check Sum .1000608

Line 19/04.

U7	U6	U5	U4
1000608	4uw2755		
U3	U2	U1	U0
99	98	97	96
		20535z	y2012vz
95	94	93	92
	-8206082	-6061759	
91	90	89	88
87	86	85	84
			5560270
83	82	81	80
uy60380	5354204	-v0312vz	525309w
79	78	77	76
	-x052uvz	4y50264	8y122xw
75	74	73	72
4w4x090	4v4y093	494939z	9y2x096
71	70	69	68
	484815z	484w2zz	
67	66	65	64
455y396	44452xz		-8008082
63	62	61	60
4142396	200000	3z3zw9x	
59	58	57	56
3x3x2xw		4243v9x	vu3v396
55	54	53	52
	38393zw	-8000020	
51	50	49	48
v53v000	v53809x	32363v3	32322xz
47	46	45	44
	3030vzw	2z49080	
43	42	41	40
	1w00000	2v3009x	
39	38	37	36
	29296vz		u629396
35	34	33	32
		200000	222409x
31	30	29	28
u240396		1z202xw	
27	26	25	24
-5x5x2vz	1w2609w		1u1x396
23	22	21	20
200000	1818w9x	9y1z2w4	
19	18	17	16
	14143zz		12122xw
15	14	13	12
		-yu16333	-8y102v9
11	10	09	08
-801u019	1509x	w0w355	8x3v096
07	06	05	04
	80u3xw	70933u	
03	02	01	00
-8030001	34	300018	110

Four-Word Lines

Line	3	2	1	0
20				
21	Temporary Storage			
22		Pickup Command	Store Command	Tally
23				

11. Intercom 1000 Card Input-Output Routine

11.1 Specifications - Data Card Input Routine

ENTRY: The Intercom command 08 1605 causes the routine to be transferred to line 19, and control transferred to word 05.

EXIT: None in this routine.

DATA INPUT: $N \leq 4$ double precision numbers on a card.

DATA OUTPUT: The numbers stored in consecutive Intercom double precision locations in memory as specified by the second command of the two which is K N ADDR.

EXECUTION TIME: The major part of the time is required for card reading. Each number requires 2 drum cycles in addition to conversion time for processing.

SHORT LINE
LOCATIONS USED: Line 04 is used for storage of the routine during execution. Line 19 is used as input storage. In addition to the short line storage used by Intercom, and the decimal-to-binary conversion routine, the following are used by this routine: 20.02,03; 21.02; 22.02; 22.03; 23.03.

CHECK SUM: (u7) 100060u

REMARKS: Index Register 8 may not be used with the command to use this routine.

11.2 Specifications - Data Card Output Routine

ENTRY: The Intercom command 08 1610 causes the routine to be transferred to line 19, and control transferred to word 10.

EXIT: None in this routine.

DATA INPUT: A floating-binary number in the Intercom AA register (21.00,01)

DATA OUTPUT: The converted number punched in a card.

EXECUTION TIME: Six drum cycles plus conversion time.

SHORT LINE
LOCATIONS USED: Line 19. 05.76-.97. 02.02,03 for format. This routine uses no short line storage beyond that used by Intercom, and the binary-to-decimal conversion routine.

CHECK SUM: (u7) 100060u

REMARKS: Since it is possible for Intercom computation to proceed while output occurs, the whole punching time is not included in the above time.

11.3 Method - Data Card Input Routine

Since this routine requires the use of line 19 for input, it cannot be executed from line 19 while input information is stored there. It must be executed from some other command line. This is acoomplished by interchanging it with line 04 of Intercom, using line 19 as intermediate storage. When the input and conversion are completed, the lines are switched back again.

This routine uses various parts of Intercom to do some of its work. A pair of Intercom commands must be written to use the routine. The second command contains a number to tell how many numbers are punched on the card and the location where the first number is to be stored. The command interpreting part of Intercom is used to convert this command to a form usable by this routine.

The decimal-to-binary conversion routine of Intercom itself is used by this routine for the actual conversion of the numbers.

11.4 Method - Data Card Output Routine

Since this routine requires the use of line 19 for output, it cannot be executed from line 19 during output. During the output part of the routine, it is executed from line 05.

11.5 Detailed Description - Data Card Input Routine

1. The Intercom command 08 1605 causes this routine to be transferred from line 16 to line 19, with the next command taken from word 05 of the routine.

2. Part of Intercom (Line 04) is transferred into line 16, and this routine transfers inself from line 19 to line 04. A special exit to this routine is inserted in the command interpreting part of Intercom at 00.46, and control is transferred back to Intercom.

3. On return to this routine from the Intercom command interpreting part, the command K N ADDR has been converted to binary form giving WD, N and CH (See page 11 of the Intercom 1000 D.P. Technical Manual.)

4. A card read command is given. Special extractors are placed in 20.02,03 and the contents of 22.02,03 are stored temporarily in 04.70,71. The two-word registers are cleared. N is extracted from the converted command and shifted (in AR) to give 2N in the T position (2^{-8}) to be used as a tally.

5. WD and CH are extracted from the converted command. CH is separated from WD and shifted to the D position. It is then recombined with WD and a dummy to form the initial store command. The normal command is restored in 00.46.

6. A special command is placed in 02.86. The routine waits until the card is finished being read.

7. The tally is reduced by $2 \cdot 2^{-8}$ and its sign tested. If negative, looping is completed (See10 below). A dummy is added to the tally to form a pick-up command, which is executed after the two-word registers are cleared.

8. The number is shifted one bit left. Then control is transferred to the decimal-to-binary conversion routine (02.70), with a return command in 23.03.

9. After conversion, the store command is executed and then modified by adding $2 \cdot 2^{-8}$. The constant $28 \cdot 2^{-8}$ is then added and the sign tested. A positive indicates that the last number converted had been stored in words 98 and 99 of a channel. The next number should be stored in words 00 and 01 of the next channel, so a channel modifier is added to form the new store command. The loop goes back to step 7, above.

10. This routine is transferred from line 04 to line 19 and control transferred to line 19. Line 16 (the original line 04 of Intercom) is transferred back to line 04, while this routine is transferred from line 19 back to line 16. The original contents of 22.02,03 are restored, as well as the normal command in 02.86. Control is returned to Intercom.

11.6 Detailed Description - Data Card Output Routine

1. The Intercom command 08 1610 causes this routine to be transferred from line 16 to line 19, with the next command taken from word 10 of the routine.

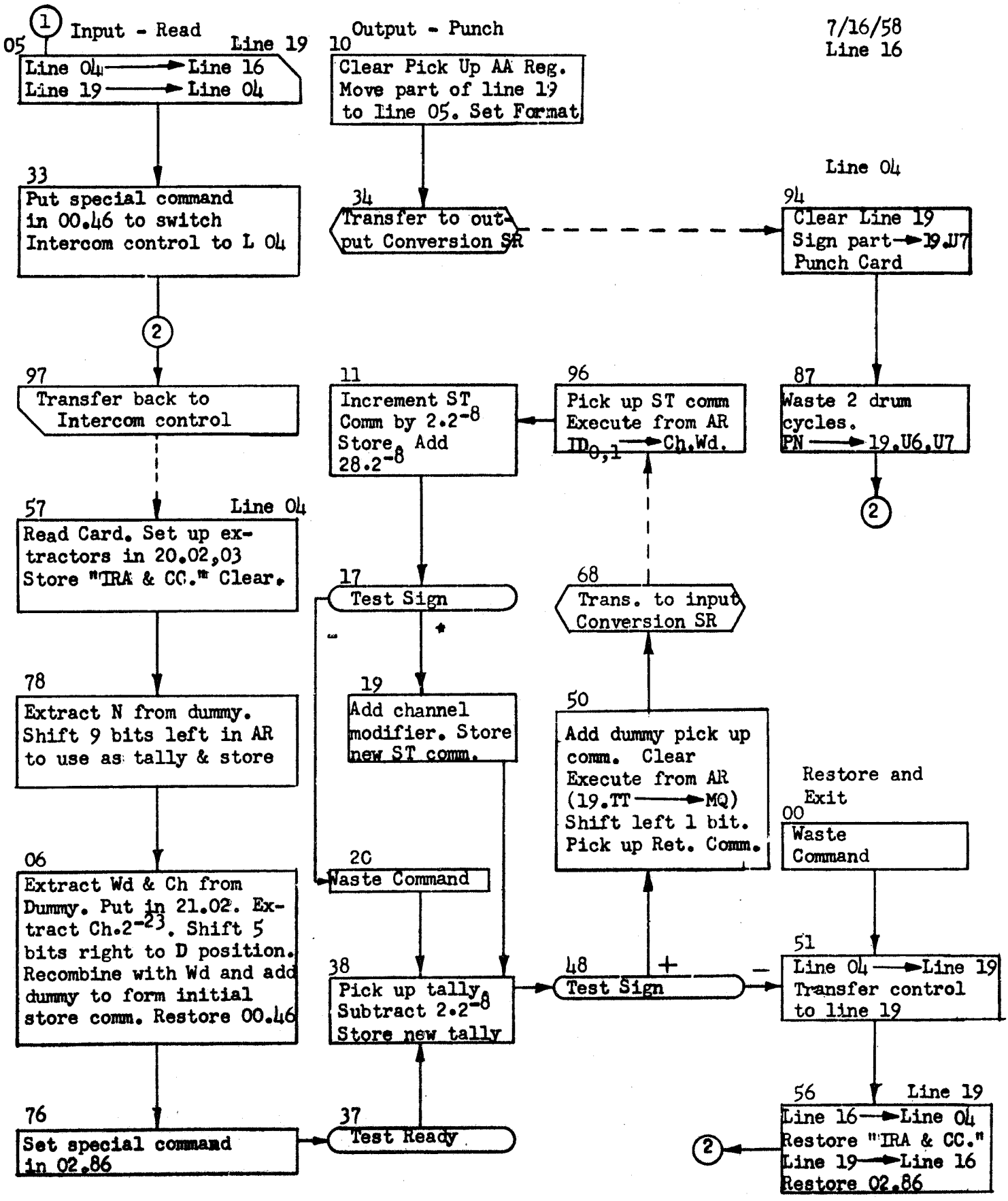
2. With the two-word registers cleared the contents of the Intercom AA register are picked up in the ID. A return command is picked up in the AR.

3. Part of the commands are transferred to line 05 for operation after the conversion. The format for punching is set in line 02, and control is transferred to the binary-to-decimal conversion routine.

4. The return command causes control to be transferred to line 05. Line 19 is cleared, then part of the converted number with the sign is placed in 19.u7. The card punch command is given, and after wasting 2 drum cycles to allow the sign to be punched, the whole converted number is placed in 19.u6,u7.

5. Control is then returned to Intercom.

11.7 Intercom 1000 - Card Input-Output Routine



Bendix Computer — Los Angeles 45, California

G-15D

Prepared by R.J. Margolin

Page 1 of 4

Date: 7/16/58

PROGRAM PROBLEM: Intercom 1000 D.P. - Card Input Routine

Line 16

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
U0	U1	U2	U3
U4	U5	U6	

L	P	T or L_k	N	C	S	D	BP	NOTES
05	U	06	30	0	04	16		Line 04 ——▷ Line 16
30	U	31	33	0	19	04		Line 19 ——▷ Line 04
33		36	44	0	19	28		Spec. Comm. = 19.36 ——▷ AR
(36)	W	48	57	4	21	31		[Transfer to 04.57]
44		46	97	0	28	00		Spec. Comm. = AR ——▷ 00.46
97	W	99	01	0	21	31		Return to Intercom Control
57		58	61	0	14	31		Read Card
61		62	69	4	04	20		Extractors = 04.62,63 ——▷ 20.02,03
(63)								[7Z003Y0◆
(62)								0000BY0◆]
69		70	73	4	22	04		"IRA & CC" = 22.02,03 ——▷ 04.70,71
73		76	78	0	23	31		Clear
78		79	83	0	30	28		$N \cdot 2^{-16} = \overline{20}.21.03$ ——▷ AR
83	U	93	93	2	28	29		AR ——▷ AR◆ 9 w.t.
93		95	06	2	28	22		$2N \cdot 2^{-8}$ = AR ——▷ 22.03
06		07	08	0	31	28		Wd & Ch = 20.21.03 ——▷ AR
08		10	21	0	28	21		Wd & Ch = AR ——▷ 21.02
21		22	23	0	31	25		$Ch \cdot 2^{-23}$ = 20.21.02 ——▷ ID_0
23		10	35	0	26	31		Shift T = 10
35		38	42	2	30	28		$Wd \cdot 2^{-8} = \overline{20}.21.02$ ——▷ AR
42		46	53	0	04	00		Normal Comm. = 04.46 ——▷ 00.46
(46)		47	48	2	31	28		[Extract
53		54	59	2	25	29		$Ch \cdot 2^{-28} = ID_C$ ——▷ AR◆]
59		60	64	0	04	29		Dummy St. Comm. = 04.60 ——▷ AR◆
(60)	W	(00)	11	4	25	(00)		[$ID_{0,1}$ ——▷ 00.00,01]
64		66	76	0	28	22		Init St. Comm. = AR ——▷ 22.02

FORM BCD 108-5.0

Bendix Computer — Los Angeles 45, California

Page 2 of 4

G-15D

Prepared by R.J. Margolin

Date: 7/16/58

PROGRAM PROBLEM: Intercom 1000 D.P. - Card Input Routine

Line 16

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
U0	U1	U2	U3
U4	U5	U6	

L	P	T or L_k	N	C	S	D	BP	NOTES
76		77	79	0	04	28		Special Comm. = 04.77 ⟶ AR
(77)								5858000*
79		86	37	0	28	02		Special Comm. = AR ⟶ 02.86
37		37	37	0	28	31		Test Ready
38		39	40	0	22	28		$2N \cdot 2^{-8}$ = 22.03 ⟶ AR ⟵
40		41	45	3	04	29		$2 \cdot 2^{-8}$ = 04.41 ⟶ AR*
(41)								0200000*
45		47	48	0	28	22		New Tally = AR ⟶ 22.03
48		50	50	0	22	31		T_1AR ⟶ Test
51	U	52	54	0	04	19		Line 04 ⟶ Line 19 to p.3 — Set
50		52	55	0	04	29	*	Dummy PU Comm. = 04.52 ⟶ AR* — Not Set
(52)	W	(00)	09	4	19	24		19.(00.01) ⟶ $MQ_{0,1}$
55		58	99	0	23	31		Clear
99		U1	U1	0	31	31		NC from AR
09		02	65	0	26	31		Shift T = 2
65		67	68	0	04	23		Ret. Comm. = 04.67 ⟶ 23.03
(67)	W	93	96	4	21	31		NC from 04.96
68		70	70	2	21	31		Transfer to Input Conversion SR
96		98	99	0	22	28		Store Comm. = 22.02 ⟶ AR
(AR)	W	(Wd)	11	4	25	(Ch)		Conv No. = $ID_{0,1}$ ⟶ Ch.Wd — NC from AR
11		12	13	0	04	29		$2 \cdot 2^{-8}$ = 04.12 ⟶ AR*
(12)								0Z00000*
13		14	15	0	28	22		New St. Comm. = AR ⟶ 22.02
15		16	17	0	04	29		$28 \cdot 2^{-8}$ = 04.16 ⟶ AR*
(16)								1W00000*
17		19	19	0	22	31		T_1.AR ⟶ Test

FORM BCD 108-5-0

Bendix Computer — Los Angeles 45, California

Page 3 of 4

G-15D

Prepared by R.J. Margolin

Date: 7/16/58

PROGRAM PROBLEM: Intercom 1000 D.P. - Card Input Routine

Line 16

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
u0	u1	u2	u3
u4	u5	u6	

L	P	T or L_k	N	C	S	D	BP	NOTES
20		22	38	0	00	00		Waste Comm. - Set to p.2
19		22	25	0	04	29		Channel Modifier = 04.22 → AR ← Not Set
(22)								[8000001-]
25		26	38	0	28	22		New St Comm. = AR → 22.02
54		56	56	6	21	31		Transfer control to this routine in line 19 ← from p. 2
56	U	57	66	0	16	04		Line 16 → Line 04
66		70	72	4	19	22		"IRA & CC" = 19.70,71 → 22.02,03
72	U	73	74	0	19	16		Line 19 → Line 16
74		86	97	0	19	02		Normal Comm. = 19.86 → 02.86 → to p. 1
(86)								[5858337 ←]
00		02	51	0	00	00		Restore & Exit

FORM BCD 108-5.0

Bendix Computer

Los Angeles 45, California

Page 4 of 4

Date: 7/16/58

G-15D

PROGRAM PROBLEM: Intercom 1000 D.P. - Card Output Routine

Prepared by R.J. Margolin

Line 16

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
U0	U1	U2	U3
U4	U5	U6	

L	P	T or L_k	N	C	S	D	BP	NOTES
10		13	14	0	23	31		Clear
14		16	28	4	21	25		AA Reg. = 21.00,01 ⟶ $ID_{0,1}$
28		29	75	0	19	28		Exit Comm. = 19.29 ⟶ AR
(29)	W	93	94	5	21	31		[NC from 05.94]
75	U	98	98	0	19	05		Part of line 19 ⟶ Line 05
98		02	34	4	19	02		Format = 19.02,03 ⟶ 02.02,03
(03)								[8000000+
(02)								0000Y40+]
34	W	36	41	1	21	31		Trans to output conversion SR
94	U	95	95	0	29	19		Clear Line 19
95		U7	88	0	26	19		Sign Part = PN_1 ⟶ 19.U7
88		89	87	0	11	31		Punch Card
87	W	88	80	0	28	28		Waste 2 drum cycles
80		U6	97	4	26	19		Conv. No. = $PN_{0,1}$ ⟶ 19.U6,U7 to p.1 ⟶
U6								[814Y668+
U7								100060U+]

FORM BCD 108-5.0

MEMORY ALLOCATION CHART

Problem { Intercom 1000 (D.P.) / CA-1 Data Input-Output. }

Page 1 of 1.

Date 4/2/59.

Prepared by Robert J. Margolin.

Check Sum .100060u

Line 19/04/05

U7	U6	U5	U4
100060u	814v668		
U3	U2	U1	U0
99	98	97	96
65653zz	-8222262	y3012vz	y2632xw
95	94	93	92
yv58353	5z5z3v3	xz06v96	
91	90	89	88
			595717z
87	86	85	84
x85039w	585833v		
83	82	81	80
5x5xv9x			-yu61353
79	78	77	76
x625382	50533xw	5858000	4y4z09w
75	74	73	72
6262265	x661262	4w4y2zz	494u270
71	70	69	68
		-48492w4	4646uvz
67	66	65	64
-xx602vz	-w648276	w344097	w24w396
63	62	61	60
7z003y0	3y0	-4045094	-800v320
59	58	57	56
3x400?x	3w3w3zz	3u3x1xz	3942204
55	54	53	52
3u632zz	-3838uvz	373vv3x	-8009278
51	50	49	48
3436093	v43709x		32322xz
47	46	45	44
	3030vzw	uz30396	uy61380
43	42	41	40
	uy35080	200000	2u2xw9x
39	38	37	36
	28282xw	252539z	-v0392vz
35	34	33	32
u62uvxw	u4296vz	u42w27w	
31	30	29	28
	1z21264	-xx5y6vz	1y4v27w
27	26	25	24
		1v26396	
23	22	21	20
u2335z	-8000001	17173z9	9626000
19	18	17	16
961909x		13132xz	1w00000
15	14	13	12
111109x	-901w2v9	z0z396	200000
11	10	09	08
x0x09x	x0y2zz	24135z	8u15395
07	06	05	04
	8083zw	61y090	
03	02	01	00
8000000	y40		8233000

Four-Word Lines

Line	3	2	1	0
20	7Z003Y0	00003Y0		
21	Converted Command	Wd & Ch	AA Register	
22	Tally	Store Command		
23	Return Command			

12. Intercom 1000 Command Card Routine

12.1 Specifications

ENTRY: The Intercom command 08 1775 causes the routine to be transferred to line 19, and control transferred to word 75.

EXIT: None in this routine.

DATA INPUT: Intercom commands punched on cards in the 026, N commands per card.

DATA OUTPUT: Commands in consecutive Intercom locations beginning with the location specified by the second command of the two, which is K N ADDR.

EXECUTION TIME: Basically, the time is determined by card reading time, during which conversion and storage of commands may occur. Each command requires 3 drum cycles.

SHORT LINE LOCATIONS USED: Line 04 is used for storage of the routine during execution. Line 19 is used for input. Words u0 through u7 of line 05 are used for temporary storage. Line 23 is used for input. In addition to the short line storage used by Intercom, this routine uses: 20.02,03; 21.00; 21.01; 21.02,03; 22.01; 23.00-03.

CHECK SUM: (u6) -4809005
(u7) 100060v

REMARKS: Index register 9 may not be used with the command to use this routine.

12.2 Method

Since this routine requires the use of line 19 for input, it cannot be executed from line 19 while the line is being used for input. It must be executed from some other command line. This is accomplished by interchanging it with line 04 of Intercom, using line 19 as intermediate storage. When the input is completed, the lines are switched back again.

The routine must re-arrange the command from the card, which is in the form K OP CH WD to the form for storage K WD OP CH. By shifting, the first form is made into three versions, each containing some of the characters in the positions desired for the final form. Then, by extraction and combination, the final form is made up.

12.3 Detailed Description

1. The Intercom command 08 1775 causes this routine to be transferred from line 17 to line 19 with the next command taken from word 75 of the routine.

2. Part of Intercom (line 04) is transferred into line 17 and this routine transfers itself from line 19 to line 04. A special exit to this routine is inserted in the command interpreting part of Intercom at 00.46, and control is transferred back to Intercom.

3. On return to this routine from the Intercom command interpreting part, the command K N ADDR has been converted to binary form giving WD, N and CH. (See page 11 of the Intercom 1000 Double Precision Technical Manual)

4. A card read command is given. The two-word registers are cleared. N is extracted from the converted command and shifted (in AR) to the T position (2^{-8}) to be used as a tally setter. A special extractor is placed in 20.02.

5. WD and CH are extracted from the converted command. The CH is then extracted and shifted right to the D position, recombined with WD and a dummy to form the initial store command. The contents of the AA register (20.00,01) are temporarily stored in line 04. Special extractors are placed in 20.02,03. The normal command is restored to 00.46.

6. Ready is tested to find whether reading of card is completed. The card information is moved from lines 23 and 19 to 05.u0-u7. The last command from the card is tested against all z's. If not equal another card read command is given. If equal, the card read command is skipped. The tally setter is picked up.

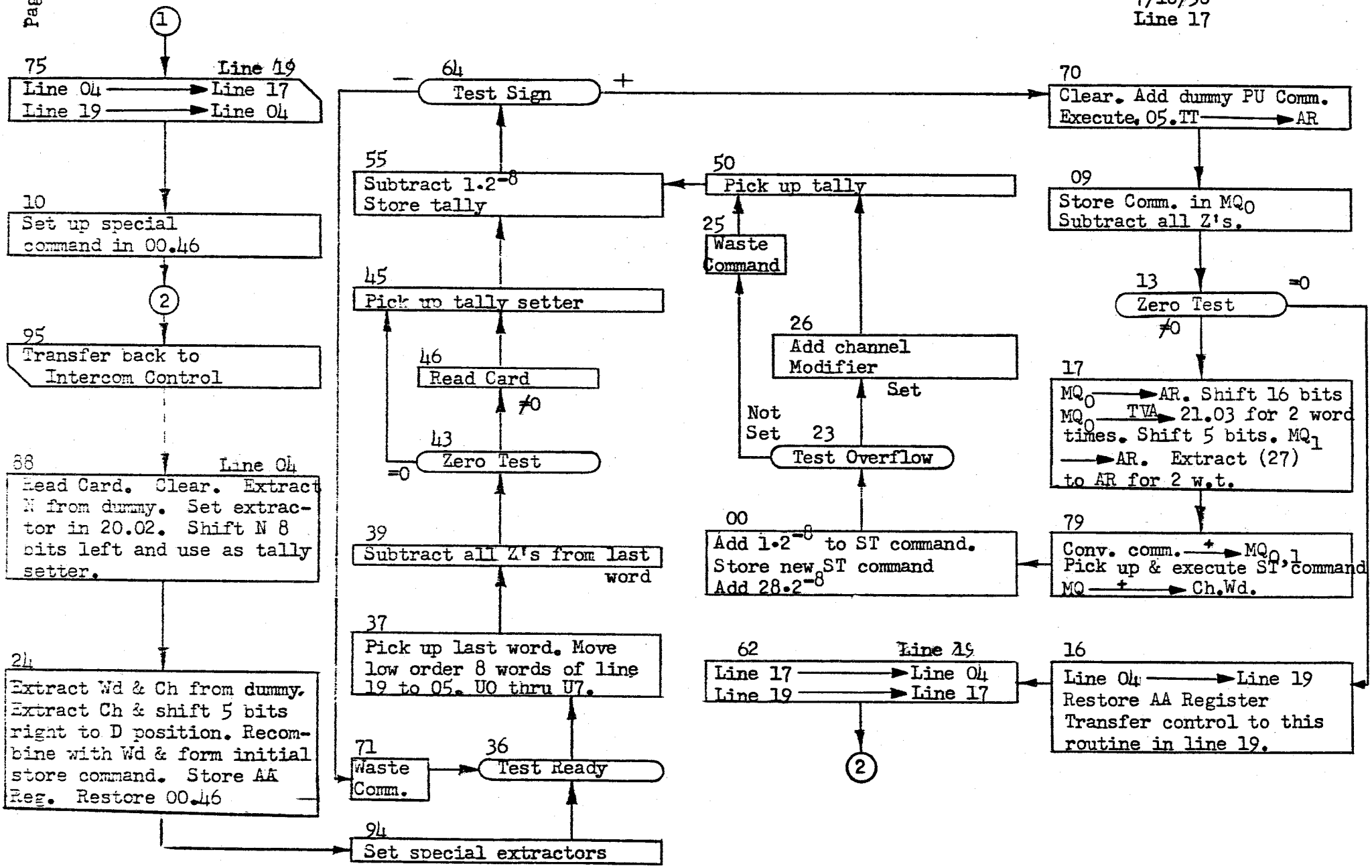
7. 1.2^{-8} is subtracted and the tally stored, then its sign tested. If negative, the program goes back to step 6 above. If not, the two-word registers are cleared, and a dummy added to the tally to form a pick-up command, which is executed. The command from card is tested against all z's. If equal, the looping is ended (see 9 below). If not, the command is re-arranged into the form for internal storage. (See method)

8. The store command is executed, and then modified by having 1.2^{-8} added to it. A constant of 28.2^{-8} is then added and the overflow tested. If the overflow is set, the last command from cards had been stored in word 99 of a channel. The next command should be stored in word 00 of the next channel, so a channel modifier is added to form the new store command. The tally is then picked up and the loop goes back to step 7 above.

9. This routine is transferred, from line 04 to line 19, and the contents of the AA register (21.00,01) is restored. Control is transferred to this routine in line 19. Line 17 (the original line 04 of Intercom) is transferred back to line 04. This routine is transferred from line 19 back to line 17, and control is returned to Intercom.

12.4 Intercom 1000 – Command Card Input Routine

7/18/58
Line 17



Bendix Computer — Los Angeles 45, California

Page 1 of 4

G-15D

Prepared by R.J. Margolin

Date: 7/17/58

PROGRAM PROBLEM: Intercom 1000 - Command Card Input Routine

Line 17

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
U0	U1	U2	U3
U4	U5	U6	

L	P	T or L_k	N	C	S	D	BP	NOTES
75	U	76	76	0	04	17		Line 04 ——▷ Line 17
76	U	77	10	0	19	04		This Routine Line 19 ——▷ Line 04
10		27	34	0	04	28		Special Comm = 04.24 ——▷ AR
(27)	W	49	88	4	21	31		[NC from 04.88]
34		46	95	0	28	00		Special Comm = AR ——▷ 00.46
95	W	U0	01	0	21	31		Transfer to 00.01
88		89	89	0	14	31		Read Card
89		92	92	0	23	31		Clear
92		95	97	0	31	28		N = 20.21.03 ——▷ AR
97		98	04	0	04	20		Extractor = 04.98 ——▷ 20.02
(98)								[0000 3Y0 ◆]
04	U	13	18	2	28	29		AR ——▷ AR◆ (8w.t.)
18		21	24	0	28	22		Tally Setter 2^{-8} = AR ——▷ 22.01
24		27	29	0	30	28		Wd & Ch = 20.21.03 ——▷ AR
29		30	33	0	28	21		Wd & Ch = AR ——▷ 21.02
33		34	38	0	31	25		Ch.2^{-23} = 20.21.02 ——▷ ID_0
38		40	47	4	21	04		AA reg. = 21.00.01 ——▷ 04.40.41
47		10	72	0	26	31		Shift T = 10
72		74	44	0	04	28		Normal Command = 04.74 ——▷ AR
(74)	U	48	48	2	31	28		[Extract]
44		46	48	0	28	00		Normal Command = AR ——▷ 00.46
48		50	51	0	30	28		Wd.2^{-8} = 20.21.02 ——▷ AR
51		52	66	2	25	29		Ch.2^{-28} = ID_0 ——▷ AR◆
66		67	86	0	04	29		Dummy St Comm = 04.67 ——▷ AR◆
(67)	W	00	00	1	24	00		[MQ ——◆——▷ 00.00]
86		88	94	0	28	21		Init St Comm = AR ——▷ 21.00

FORM BCD 108-5.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by R.J. Margolin

Page 2 of 4

Date: 7/17/58

PROGRAM PROBLEM: Intercom 1000 - Command Card Input Routine

Line 17

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
U0	U1	U2	U3
U4	U5	U6	

L	P	T or L_k	N	C	S	D	BP	NOTES
94		14	36	4	04	20		Special Extractors = 04.14.15 ——▷ 20.02.03
(14)								[Z000000+
(15)								0ZZ0000+]
36		36	36	0	28	31		Test Ready
37		40	99	1	23	24		Last Comm = 23.00 ——▷ MQ_0
99	U	U4	01	0	23	05		L23 ——▷ 05.U0 ——▷ U3
01	U	08	80	0	19	23		19.04-07 ——▷ L23
80		81	96	0	04	28		Special Comm = 04.81 ——▷ AR
(81)	U	00	22	0	23	05		[L23 ——▷ 05.U4-7]
96		98	U3	0	31	31		NC from AR
22		24	39	1	24	28		Last Comm = MQ_0 —+—▷ AR
39		42	43	3	04	29		All Z's = 04.42 —=—▷ AR+
(42)								[ZZZZZZZ+]
43		45	45	0	28	27		AR ——▷ Test
46		45	45	0	14	31		Set Read Card
45		49	55	0	22	28		Tally Test Set Setter = 22.01 ——▷ AR
55		58	60	3	04	29		1.2^{-8} = 04.58 —=—▷ AR+ ◁——
(58)								[0100000]
60		61	64	0	28	21		New Tally = AR ——▷ 21.01
64		66	70	0	22	31		T_1·AR ——▷ Test
71		72	36	0	00	00		- Set Waste Comm.
70		73	78	0	23	31		+ Not Set Clear
78		84	91	0	04	28		Dummy PU Comm. = 04.84 ——▷ AR+
(84)	W	U0	09	0	05	28		[05.(U0) ——▷ AR]
91		93	93	0	31	31		NC from AR
09		10	11	0	28	24		Comm. = AR ——▷ MQ_0

FORM BCD 108-5.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by R.J. Margolin

Page 3 of 4

Date: 7/17/58

PROGRAM PROBLEM: Intercom 1000 - Command Card Input Routine

Line 17

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
U0	U1	U2	U3
U4	U5	U6	

L	P	T or L_k	N	C	S	D	BP	NOTES
11		12	13	3	04	29		All Z's = 04.12 ⟶ AR+
(12)								ZZZZZZZ+
13		15	16	0	28	27		AR ⟶ Test
16	U	17	35	0	04	19		=0 Not Set; Line 04 ⟶ Line 19; ≠0 Set
17		18	19	0	24	28		MQ_0 ⟶ AR
19		32	53	0	26	31		Shift T = 32
53		56	57	2	24	21		MQ_0 —TVA→ 21.03 (2w.t.)
57		10	68	0	26	31		Shift T = 10
68		69	73	2	24	28		MQ ⟶ AR
73	U	76	79	0	27	28		20.21+20.AR.02.03 ⟶ AR
79	U	82	83	1	28	24		Conv. Comm. = AR ⟶ MQ_0, MQ_1
83		84	91	0	21	28		Store Comm. = 21.00 ⟶ AR
(AR)		(Wd)	00	1	24	(Ch)		Conv. NC from AR; Comm. = MQ ⟶ Ch.Wd
00		05	06	0	04	29		1.2^{-8} = 04.05 ⟶ AR+
(05)								0100000+
06		08	20	0	28	21		New St. Comm. = AR ⟶ 21.00
20		21	23	0	04	29		28.2^{-8} = 04.21 ⟶ AR+
(21)								W00000+
23		25	25	0	29	31		Test Overflow
25		26	50	0	00	00		Not Set; Waste Command
26		30	31	0	04	29		Channel Modifier = 04.30 ⟶ AR+
(30)								8000001-
31		32	50	0	28	21		New St Comm. = AR ⟶ 21.00
50		53	55	0	21	28		Tally = 21.01 ⟶ AR

FORM BCD 108-5.0

Bendix Computer — Los Angeles 45, California

Page 4 of 4

G-15D

Prepared by R.J. Margolin

Date: 7/17/58

PROGRAM PROBLEM: Intercom 1000 - Command Card Input Routine

Line 17

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
28	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
72	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99
u0	u1	u2	u3
u4	u5	u6	

L	P	T or L_k	N	C	S	D	BP	NOTES
35		40	59	4	04	21		Restore AA register
59	W	61	62	6	21	31		Transfer control to this routine in line 19
62	U	63	63	0	17	04		Line 17 ——▷ Line 04
63	U	64	95	0	19	17		Line 19 ——▷ Line 17
(U6)								4809005-
(U7)								100060V

FORM BCD 108-5.0

MEMORY ALLOCATION CHART

Problem { Intercom 1000 (D.P.) / CA-1 Commands. }

Date 4/2/59.

Prepared by Robert J. Margolin.

Check Sum .100060v

Line 19/04.

Four-Word Lines

Line	3	2	1	0
20	Intercom Extractor	00003Y0		
21	Interpreted Command	Wd & Ch	AA Register	
22			Tally Setter	
23				
20	0ZZ0000	Z000000		
21	Temporary Storage		Tally	Store Command
22			Tally Setter	
23		Input		

U7 100060v	U6 -4809005	U5	U4
U3	U2	U1	U0
99 68012y5	98 3y0	97 6304094	96 62673zz
95 y4012vz	94 -8y24094	93	92 xz613zw
91 5x5x3zz	90	89 5w5w2zz	88 59591xz
87	86 x85y395	85	84 y4090vw
83 55v2vw	82	81 162y5	80 526009w
79 5253798	78 x45v09x	77	76 4x0u264
75 4w4w091	74 3030vzw	73 4w4z37w	72 wu2w09w
71 4924000	70 494y2zz	69	68 4649v1w
67 8000700	66 445609x	65	64 42462xz
63 405z271	62 3z3z224	61	60 3y40395
59 -vx3yuvz	58 100000	57 u4435z	56
55 vu3ww9x	54	53 3839v15	52
51 3542v3x	50 v5372vw	49	48 v2333xw
47 u4835z	46 2x2x1xz	45 v1372xw	44 uy30380
43 ux2x39v	42 zzzzzzz	41 2v2vw9x	40
39 uu2vw9x	38 -u82z2u4	37 u8636z8	36 242439z
35 -u83v095	34 uy5z380	33 23263z9	32
31 2132395	30 -8000001	29 1z21395	28
27 -v1582vz	26 9y1z09x	25 1v32000	24 9v1x3xw
23 19193vz	22 98277lw	21 1w00000	20 161709x
19 203535z	18 9518396	17 131331w	16 1123093
15 zz0000	14 z000000	13 8z1039v	12 zzzzzzz
11 x0xw9x	10 9v2209w	09 v0v398.	08
07	06 8814395	05 100000	04 x12v9x
03	02	01 850277	00 850609x

13. MTA Service Routine

13.1 Specifications

ENTRY: Command 081870 enters at loc. 18.70. MTA operations enter at word 88. (At this entry, the MTA command characteristic is formed and a dummy command is formed in AR to exit to the different operations.) Operation entries follow:

Write on Magnetic Tape	Word 02
Read Magnetic Tape	Word 01
Write File Number on Magnetic Tape	Word 03
Reverse Magnetic Tape	Word 04
Search Magnetic Tape	Word 05

ERROR STOPS: Incorrect reading of a block of information will be detected. Upon detection, the incorrectly read block is reversed and read again. This procedure will repeat until stopped by the operator. To regain control, follow the instructions listed below:

(1) While tape is moving in the reverse direction, place compute switch in BP position.

(2) Set compute switch to "Idle."

(3) When reading halts, with enable on, type "i".

(4) Set compute to GO to execute the next Intercom command, or to BP to return to manual.

SHORT LINE STORAGE USED: 20.02, 21.02, 23.00-03.

SUMMATION CONSTANT: Balancer in word u6 and check sum in word u7.

REMARKS:

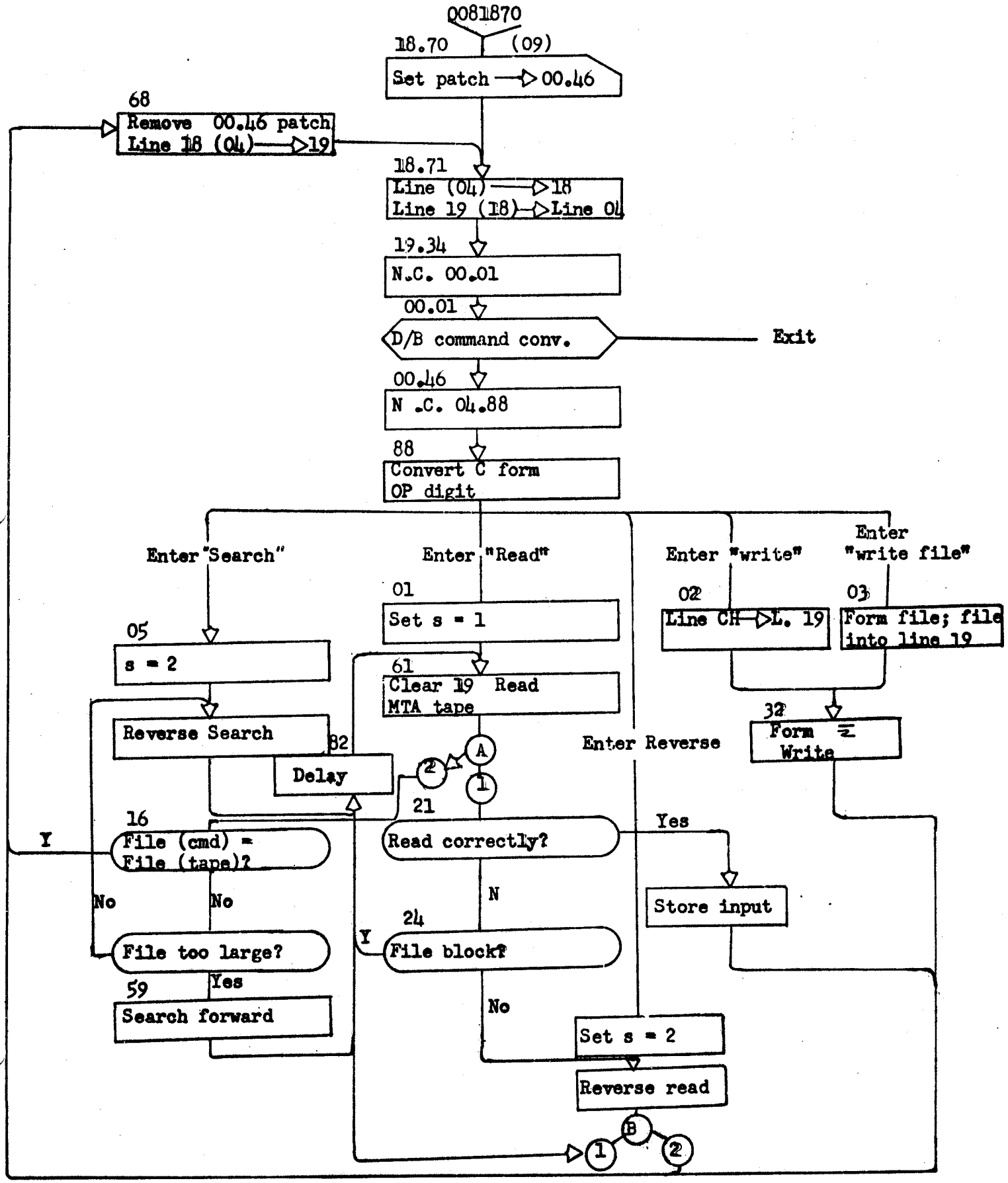
(1) A block of information written by this subroutine will have words u0 and u1 destroyed. These locations are used for the storage of a balancer and check sum.

(2) A channel of information will not be written on tape correctly if words 00-03 are zero.

(3) Timing delay between operations is set at nine drum revolutions.

13.2 MTA Service Routine (Intercom 1000 D.P.)

FLOW CHART



Bendix Computer — Los Angeles 45, California

G-15D

Prepared by D. Hassell

Page 1 of 7

Date: 18/04

PROGRAM PROBLEM: MTA Service Routine (INTERCOM 1000 D.P.)

Line 10/04

L	P	T or LK	N	C	S	D	BP	NOTES
								ENTER AFTER EXECUTION OF "081870"
70		14	25	0	19	28		(19.14) = patch #1 ⟶ AR
14		88	88	4	21	31		
25		46	71	0	28	00		(AR) = patch ⟶ 00.46
71	u	72	73	0	04	18		Line Interchange
73		75	72	0	29	31		Reset Overflow
72	u	73	34	0	19	04		Line Interchange
34		01	01	0	21	31		N.C. 00.01
								ENTER TO EXECUTED "OP" CODE
88	u	51	09	1	29	26		Clear PN
09		11	97	0	30	28		ADDR ⟶ AR
97		98	07	0	28	21		(AR) = ADDR ⟶ 21.02
07		26	93	0	04	20		
26								00003y0
93		95	91	0	31	28		OP ⟶ AR
91		u1	95	3	04	29		$u \cdot 2^{-16}$ ⟶ AR
u1								000u000
95		98	98	0	22	31		AR < 07
								NO
98		08	91	1	04	30		FORM MTA #(characteristic)
08								0000400
								YES
99		u1	u3	1	04	29		$u \cdot 2^{-16}$ ⟶ AR

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 2 of 7

G-15D

Prepared by D. Hassell

Date:

PROGRAM PROBLEM: MTA Service Routine (INTERCOM 1000 D.P.)

Line 18/04

L	P	T or Lk	N	C	S	D	BP	NOTES
u3		04	13	1	26	22		MTA # ——▷ 22.00
13		08	29	0	24	31		Shift ID = (CH). = Dest.
29	u	30	35	0	29	19		(0) ——▷ 19.
35		37	37	0	31	31		
AR		00	0P	0	00	00		
								ENTER FROM 04.33 BELOW
AR		39	52	C	13	31		Read tape unit C NC 52 below
								ENTER "READ"
01	w	61	18	4	21	31		Mark 61 N.C. 04.18
18		10	33	0	04	28		Read command ——▷ AR
10		39	52	0	13	31		
33		36	35	0	22	29		Characteristic —+—▷ AR
								See 35 above
52		54	55	0	04	28		Read command ——▷ AR
54		80	80	0	13	31		
55		48	57	0	22	29		(delay) and MTA # —+—▷ AR

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 3 of 7

G-15D

Prepared by D. Hassell

Date:

PROGRAM PROBLEM: MTA Service Routine (INTERCOM 1000 D.P.)

Line 18/04

L	P	T or L_K	N	C	S	D	BP	NOTES
57		u7	u7	0	31	31		
u7		60	60					
AR		80	80	C	13	31		
u7		20+	68					
AR	u	WD	88	0	19	CH		Store input N.C. 04.68 See page 5
60	w	47	00	4	20	31		Return to mark > 47
								ENTER "READ" (CONT.)
61		61	61	0	28	31		Wait for ready
62		u1	20	1	19	28		Σ —+→ AR
20	u	21	21	3	19	29		Σ 19 —−→ AR
21		40	23	2	04	27		AR = 0^1 (04.40) ——→ AR
40	u	20	88	0	19	00		Dummy store com.
								= 0
23		21	41	2	25	29		Destination —+→ AR
41		46	57	0	30	29		wd —+→ AR N.C. 04.57 above
								≠ 0
24	u	00	74	0	19	27		File block
								YES
74	w	01	82	4	21	31		Set to read next block after delay
								NO
75	w	77	82	4	21	31		Set to reverse after delay

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 4 of 7

G-15 D

Prepared by D. Hassell

Date:

PROGRAM PROBLEM: MTA Service Routine (INTERCOM 1000 D.P.)

Line 18/04

L	P	T or L_K	N	C	S	D	BP	NOTES
								ENTER TO DELAY
02		81	u5	0	01	01		delay 1 drum rev.
u5		u4	u4	0	04	04		delay 2 drum rev.
u4		u2	u2	0	06	06		delay 2 drum rev.
u2		u0	u0	0	08	08		delay 2 drum rev.
u0		u2	60	0	09	09		delay 2 drum rev.
								Return to mark.
								ENTER AFTER INVALID "READ"
77	w	37	19	4	21	31		Mark 37 to stop on B.P. otherwise mark return to 00.
19		31	55	0	04	28	}	
31		57	58	0	04	31		
								Move off file code
AR		57	58	C	04	31		
38	w	00	35	0	00	31		
AR		57	58	C	04	31		Reverse tape motion
58		60	60	0	12	31		sets up to read in reverse direction
								RETURN TO MARK
								ENTER HERE AFTER INVALID READ REVERSE
00		0000		0	28	31		Wait for ready then began re-read.

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 5 of 7

G-15D — Prepared by D. Hassell — Date:

PROGRAM PROBLEM: MTA Service Routine (INTERCOM 1000 D.P.) — Line 18/04

L	P	T or LK	N	C	S	D	BP	NOTES
								ENTER "WRITE"
02		94	55	0	04	28		Dummy forward search ——▷ AR
94		59	63	0	05	31		
AR		59	43 63	C	05	31		Skip first file code
43	w	00	85	0	00	31		Set ready.
85		36	39	0	04	28		Dummy line pickup ——▷ AR
36	u	20	52	0	00	19		
39		42	57	0	21	29		ADDR —+—▷ AR
AR	u	20+ wd	32 52	0	CH	19		CH.00-wd ——▷ 19
32		u0	76	5	29	19		0 ——▷ 19.u0.u1
76		78	79	1	29	28		0 ——▷ AR
79	u	80	80	1	19	29		Σ 19 —+—▷ AR
80		u1	69	1	28	19		Σ 19 ——▷ 19.u1
69		70	81	1	28	28	}	
81		82	83	3	28	28	}	Balancer ——▷ 19.u0
83		u0	84	1	28	19	}	
84		86	33	0	04	28		form write command
86	w	00	67	0	01	31		
AR		00	67	C	01	31		write 19 on tape
67		67	67	0	28	31		
68	u	69	11	0	18	19		Line 18(04) ——▷ 19
11		12	25	0	04	28		Patch #2 ——▷ AR
12		47	48	2	31	28		Patch #2
								NC 04.25 page 4

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

G-15D

Prepared by D. Hassell

Page 6 of 7

Date:

PROGRAM PROBLEM: MTA Service Routine (INTERCOM 1000 D.P.)

Line 18/04

L	P	T or LK	N	C	S	D	BP	NOTES
								ENTER "WRITE FILE"
03		28	33	0	04	28		Read command ——▷ AR
28		57	51	0	13	31		
AR		57	51	C	13	31		Move off stop code
51	w	00	56	0	00	31		Set ready
56		15	33	0	04	28		Write file com. ——▷ AR
15		44	46	0	30	31		
AR		44	46	C	30	31		Write file
46		50	90	0	21	28		Add (file #) ——▷ AR
90	u	00	32	0	28	19		#0 ——▷ 19 91-00
								N.C. 04.32 page 5
								"ENTER REVERSE"
04	w	67	19	4	21	31		Mark 67 N.C. 04.19 reverse
								ENTER SEARCH
05		05	05	0	28	31		Wait for ready
06		96	33	0	04	28		Reverse search ——▷ AR
96		59	63	0	04	31		
AR		59	63	C	04	31		Reverse search
63		63	63	0	28	31		
64	w	53	82	4	21	31		Mark to delay routine
53	w	49	18	4	21	31		Mark transfer to read routine
49		49	49	0	28	31		Wait for ready
50		54	65	6	21	25		File # ——▷ ID$_1$
65		22	92	1	26	31		Shift 11 bits

FORM BCD 108-17.0

Bendix Computer — Los Angeles 45, California

Page 7 of 7

G-15D

Prepared by D. Hassell

Date:

PROGRAM PROBLEM: MTA Service Routine (INTERCOM 1000 D.P.)

Line 18/04

L	P	T or LR	N	C	S	D	BP	NOTES
92		03	17	0	19	28		Tape file ——▷ AR
17		20	22	0	28	26		Tape file ——▷ PNO
22		25	27	1	26	28		0 ——▷ AR
27	u	64	66	5	26	30		Shift
66	u	69	78	1	25	29		File +——▷ AR
78	u	81	16	3	26	29		Tape file —m—▷ AR
16		17	47	0	28	27		AR = 0
								(Correct file?)
47		48	67	0	00	00		Delay and exit
48		44	44	0	21	31		Tape file to large?
44	w	59	82	4	21	31		N.C. delay routine
								Search forward
59		94	33	0	04	28		Search forward ——▷ AR
AR		59	63	C	05	31		N.C. 63 page 6
45	w	87	82	4	21	31		Mark 87 go to delay
30								Empty
87		89	33	0	04	28		
89		59	05	0	04	31		
AR		59	05	C	04	31		Search reverse
								N.C. reverse again
37		67	67	0	17	17		B.P. Exit
u6								Balancer
u7								Σ 1002189

FORM BCD 108-17.0

Bendix Computer

MEMORY ALLOCATION CHART

Problem: Intercom 1000 (D.P.) / MTA Subroutine.

Page 1. of 1

Date 4/2/59.

Prepared by Don Hassell.

Check Sum .1003189

Line 18.

U7 1003189	U6 2677544	U5 y868084	U4 y6660w6
U3 840x756	U2 y464108	U1 u000	U0 y63w129
99 y56749x	98 885v49y	97 6307395	96 3v3z09z
95 62622xz	94 3v3z0vz	93 xz5v3zw	92 831127w
91 y55zw9x	90 20393	89 3v0509z	88 33097vu
87 x92109w	86 804303z	85 u42709w	84 x62109w
83 y454793	82 x169021	81 5353z9w	80 y545793
79 505067x	78 5110z5x	77 -u5132vz	76 wy4z7vw
75 -wx522vz	74 -81522vz	73 4v483vz	72 4922264
71 4849092	70 8y1927w	69 475179w	68 450v253
67 434339z	66 454y73x	65 165w75z	64 -v5522vz
63 3z3z39z	62 y51467w	61 3x3x39z	60 -uz0029z
59 xy2109w	58 3wvw19z	57 6v6v3zz	56 8z2109w
55 v0392xx	54 50501vz	53 -v1122vz	52 v63709w
51 803801z	50 -v641uv9	49 313139z	48 2w2w2xz
47 3143000	46 v25u2vw	45 -x7522vz	44 -vv522vz
43 805501z	42 uy352vw	41 uy393xx	40 1458260
39 uu392vx	38 802301z	37 w243231	36 1434013
35 25253zz	34 1012vz	33 u4232xx	32 -y44w7v3
31 393u09z	30 u143000	29 1y233v3	28 39331vz
27 -404275y	26 3y0	25 uy47380	24 4u27v
23 9529v3x	22 991v75w	21 u81789v	20 1515y7x
19 9z3709w	18 8u2109w	17 941639u	16 122z39v
15 2w2y3xz	14 -58582vz	13 81x31z	12 uz30vzw
11 x1909w	10 27341vz	09 8v613xw	08 400
07 9u5x094	06 y02109w	05 50539z	04 -w3132vz
03 9w2109w	02 xy3709w	01 -vx122vz	00 39z

Four-Word Lines

Line	3	2	1	0