Packard Bell Computer

PB250 Program Library
Catalog Number 0130 A

IDENTIFICATION: SINGLE PRECISION FLOATING POINT PACKAGE
Subroutine Package

AUTHOR: R. Johnsen, PBC

ACCEPTED: 23 April 1962

PURPOSE: To perform the basic arithmetic floating point operations.

RESTRICTIONS: 1. If division by zero is attempted, FDV will exit with no operations performed and no registers changed. No error indication is given.

2. FSR makes no distinction between positive and negative numbers. It will attempt to take the square root in either case. No error indication is given.

STORAGE: 122 sectors of one long line between sectors 000 and 340 plus F01-F06, F10, F13, and F17. Not all the line 00 channels are used by each subroutine.

TIMING:

Float	(FLO)	396 μ sec
Float	(FLT)	588 μ sec
Fix	(FIX)	780 μ sec
Add	(FAD)	984 μ sec
Subtract	(FSB)	1188 μ sec
Multiply	(FMP)	672 μ sec
Divide	(FDV)	780 μ sec
Square Root	(FSR)	564 μ sec
Square	(FSQ)	816 μ sec

ACCURACY: The exponent and mantissa are each accurate to 21 bits plus sign.

USE: 1. Floating Point Format

A floating point number in this subroutine consists of two words in the following format:

a. The first (high order) word contains a normalized mantissa with the binary point between bits 0 and 1.

USE (cont.):

b. The second (low order) word consists of an exponent which represents the scaling of the mantissa.

Example: A floating point 1 in the AB registers would appear as follows (representation is in octal):

+4000000	+0000001
A Register	B Register

A floating point -1 in the AB registers would take this form:

-0000000	+0000000
A Register	B Register

2. Use of Subroutines

a. To float a Fixed Point Number (FLO):

Set Index with line location of subroutine.

```
LDA     argument
LDC     return command
TRU     entry (see Table 1)
```

The argument must be a single-precision fixed point number scaled at Q = 21. Upon exit, the AB registers will contain a floating point number in the format described above.

b. To float a fixed point number (FLT):
Set index with line location of subroutine.

```
LDP     argument
LDC     return command
TRU     entry (see Table 1)
```

The argument must be a single-precision fixed point number. The Q of the fixed point number (Q_1) must be in B scaled at Q = 21. The fixed point number must be in A scaled at Q_1.

c. To Fix a Floating Point Number (FIX):

Set Index with line location of subroutine.

```
LDP     argument
LDC     return command
TRU     entry (see Table 1)
```

The argument must be a floating point number as described above. Upon exit, the A register will con-

USE (cont.): tain a truncated fixed point integer with a fractional part in B, which will be the original fraction before truncation.

d. Floating Square (FSQ) and Square Root (FSR):

Set Index with line location of subroutine.

```
LDP         argument
LDC         return command
TRU         entry (see Table 1)
```

The argument must be a floating point number in the format described above.

e. Floating Add (FAD), Subtract (FSB), Multiply (FMP) and Divide (FDV):

These routines each require two arguments. The calling sequence is as follows:

Set Index with line location of subroutine.

Set one argument in F01-F02 (see below).

```
LDP         second argument
LDC         return command
TRU         entry (see Table 1)
```

Both arguments must be floating point numbers in the format described above. The argument in F01-F02 must have its exponent in F01 and its mantissa in F02; for FSB, this argument must be the minuend; whereas, for FDV this argument must be the dividend. The contents of F01-F02 will remain unchanged upon exit from the subroutine.

Table 1

Subroutine Entrances and Exits

Subroutine	Entrance Sector	Exit Sector
FLO	000	041
FLT	360	041
FIX	001	102
FMP	074	164
FSB	200	343
FAD	221	343
FDV	112	213
FSR	012	073
FSQ	060	164

METHOD:

1. FLO: The number is normalized and the exponent decremented from an initial value of 21. (The input comes in at $Q = 21$.

2. FLT: The number, scaled at $Q = Q_1$, is normalized and the exponent decremented from an initial value of Q_1, scaled at $Q = 21$.

3. FIX: The number is right shifted and the exponent incremented until $(\text{exp}-21) = 0$. Then tests for negative and fractional part determine if a correction must be added to the integer part to give a correct truncated result.

4. FSR: If the exponent was originally odd, the argument is shifted right one place and the exponent incremented. The exponent is then divided in two, and the square root of the mantissa taken. No normalization of the result is necessary.

5. FSQ: This routine leaves the argument both in A, B and in the pseudo-accumulator (F01-F02), and exits to FMP.

6. FMP: First the mantissas are multiplied, then a test for -0000000 is made. If the test fails (results correct) the exponents are added and the mantissa normalized. If not, the implication is that -0000000 has been multiplied by itself, which produces a result in the PB250 of -0000000, an incorrect result. In this case, the correct result (+4000000) is entered, and the two exponents are added and incremented by one.

7. FSB: The A, B registers are complemented and the routine exits to FAD. A special sequence is necessary to complement -0000000 since the one-sector multiply used to complement all other numbers does not work on this exception.

8. FAD: The two exponents are compared, the mantissa corresponding to the smaller exponent is shifted right until the binary points line up, and the common exponent is entered. The mantissas are then added. If overflow occurs, the mantissa is shifted right one place, the exponent incremented and the sign

METHOD (cont.): changed to correct the answer. If no overflow occurs, the result is normalized.

8. FDV: The exponent of the divisor is subtracted from the exponent of the dividend, and the result saved. The dividend is shifted right one place, and the quotient taken. The new exponent is then entered and incremented by one to take care of the initial shift of the dividend.

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM Single Precision Floating Point

IDENT NUMBER 0130 A

PAGE 1 OF 5

PROGRAMMER R. Johnsen

DATE 23 April 1962

LOCATION	INSTRUCTION	SYMBOLIC			REMARKS
		LOCATION	OP	ADDRESS	
00002$	001S1000;		STC		Enter FLO
001	002S1000;		STC		Enter FIX
002	003S4300;		CLB		Set to float
003	005S0300;		ROT		
004	005S04001		LDC		Set initial exponent
005	000 00051		Const		
006	035S2000;		NAD		Float argument
007	010S15001		SUB		Decrement exponent
010	000 00051		Const		
011	012S0100;		IAC		Set to fix
012	013S1000;		STC		Enter FSR
013	015S4300;		CLB		Clear for fraction
014	015S0300;		ROT		
01602$	045S2300;		SAI		Fix argument
017	021S2210;		SRT		Exp/2
02102$	022S0100;		IAC		Position for root
02302$	025 36001		TBN		Exponent odd?
024	030S1000;		STC		Save exp/2
025	027S2200;		RSI		Shift arg, incr exp
02702$	030S1000;		STC		Save exp/2 + 1
03102$	033S4400;		CLC		
03402$	035S4300;		CLB		Prepare for root
035	036S0200;		IBC		Position to exit
036	064S3000;		SQR		Take square root
037	041S3700;		TRU		Exit FLO
04502$	047 35001		TAN		Check for neg arg

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM Single Precision Floating Point IDENT NUMBER 0130A

PAGE 2 OF 5

PROGRAMMER R. Johnsen DATE 23 April 1962

LOCATION	INSTRUCTION	SYMBOLIC			REMARKS
		LOCATION	OP	ADDRESS	
046	062S3700;		TRU		Exit FIX
047	050S0300;		ROT		
05202$	053S56001		CAM		Check for fraction
053	000 0000;		Const		
054	055S0100;		IAC		Reposition integer
05602$	064S0200;		IBC		Reposition fraction
06002$	061S1300;		SDP		Enter FSQ
06302$	075S0100;		IAC		Position for multiply
064	065S0300;		ROT		Root ——→ A
065	102 7500;		TOF		Exit FIX if no fraction
066	072S14001		ADD		Correct integer
067	070S0600;		LDB		Re-enter exponent
07102$	073S3700;		TRU		Exit FSR
072	000 00001		Const		
073	102S3700;		TRU		Exit FIX
074	075S0100;		IAC		Enter FMP
07602$	077S1200;		STB		Save exp β
10002$	102S0600;		LDB		Set to multiply
10302$	104S1100;		STA		Set exit
10502$	134S3200;		MUP		Multiply
11202$	113S1000;		STC		Enter FDV
11402$	115S56001		CAM		
115	000 0000;		Const		Exit if divide by 0
116	133 7500;		TOF		
117	120S0100;		IAC		Set divisor
12102$	122S0500;		LDA		Enter exp β

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM Single Precision Floating Point

IDENT NUMBER 0130 A

PAGE 3 OF 5

PROGRAMMER R. Johnsen

DATE 23 April 1962

LOCATION	INSTRUCTION	SYMBOLIC			REMARKS
		LOCATION	OP	ADDRESS	
12302$	124S1200;		STB		Save exp a
12502$	126S4300;		CLB		Clear for divide
12702$	131S2210;		SRT		Position for divide
13102$	160S3100;		DIV		Divide
13402$	135S0100;		IAC		Preserve product
13602$	137S0500;		LDA		Enter exp β
14002$	141S1400;		ADD		Add exp a
14202$	143S0100;		IAC		Set for normalize
14402$	145S56001		CAM		Test exception
145	200 0000;		Const		
146	150 75001		TOF		
147	151S2000;		NAD		Normalize product
150	152S2200;		RSI		Fix special case
151	155S0200;		IBC		Position for exit
152	153S14001		ADD		Clear sign
153	200 0000;		Const		
154	155S0200;		IBC		Position for exit
15602$	164S3700		TRU		Exit FMP
16002$	161S0500;		LDA		Enter exp a
16202$	164S1500;		SUB		Subtract exp β
16502$	166S14001		ADD		Increment exponent
166	000 00001		Const		
167	170S0300;		ROT		Set for normalize
17202$	174S2000;		NAD		Normalize quotient
17402$	175S0200;		IBC		Position for exit
17602$	213S3700;		TRU		Exit FDV

Packard Bell Computer
PB 250 PROGRAM LISTING

PROBLEM Single Precision Floating Point

IDENT NUMBER 0130 A

PAGE 4 OF 5

PROGRAMMER R. Johnsen

DATE 23 April 1962

LOCATION	INSTRUCTION	SYMBOLIC			REMARKS
		LOCATION	OP	ADDRESS	
20002$	201S56001		CAM		Enter FSB
201	200 0000;		Const		
202	203S1000;		STC		Save exit
20402$	206 75001		TOF		Check exception
205	206S0100;		IAC		Position for complement
206	210S16001		DPA		Increment exponent
207	212S4500;		CLA		Clear for complement
210	000 00001		Const		
211	000 0000;		Const		
212	223S05001		LDA		Enter correct complement
213	215S2110;		SLT		Ready for mult bit
21502$	216S16001		DPA		Load mult bit
216	000 00001		Const		
217	000 0000;		Const		
220	222S3237;		MUP		Complement
221	223S1000;		STC	F03	Enter FAD
222	225S1300;		SDP	F05	Save β
223	100 0000;		Const		
224	225S1300;		SDP		Save β
22702$	236S0300;		ROT		Position exponent
24002$	241S1500;		SUB		Exp α - exp β
24202$	243S0100;		IAC		Set to line up args
24402$	250 34001		TCN		Check relative magnitude
245	246S06001		LDB		Prepare to complement
246	000 00001		Const		
247	251S3200;		MUP		Complement difference

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM Single Precision Floating Point

IDENT NUMBER 0130 A

PAGE 5 OF 5

PROGRAMMER R. Johnsen

DATE 23 April 1962

LOCATION	INSTRUCTION	SYMBOLIC			REMARKS
		LOCATION	OP	ADDRESS	
250	276S2300;		SAI		Line up arguments
251	260S0100;		IAC		Set to line up args
26102$	262S0500;		LDA		Enter argument
26302$	311S2300;		SAI		Line up arguments
27602$	277S4300;		CLB		Clear unwanted portion
30002$	3 201S1600;		DPA	CHECK	add *a*, enter exp
30302$	330S0200;		IBC		Position for normalize
31102$	323S4300;		CLB		Clear unwanted portion
32402$	325S1600;		DPA		Add *β*, enter exp
32702$	330S0200;		IBC		Position for normalize
33102$	333 75001		TOF		Check overflow
332	334S2000;		NAD		Normalize sum
333	335S2200;		RSI		Reposition sum
334	340S0200;		IBC		Position for exit
335	336S15001		SUB		Change sign back
336	200 0000;		Const		
337	340S0200;		IBC		Position for exit
34102$	343S3700;		TRU		Exit FAD and FSB
36002$	361S1000;		STC		Enter FLT
36202$	363S0200;		IBC		
36402$	005S4300;		CLB		

Catalog Number 0500

IDENTIFICATION: SIN-COS I
Subroutine

AUTHOR: C. Yingst, PBCC

ACCEPTED: January 18, 1961

PURPOSE: To find the sine or cosine of a given argument.

SPACE: REQUIRED 71 words in one long line, plus channels F03, F10, and F15 of line 00.

TIMING: 3.072 milliseconds maximum

ACCURACY: Maximum error, ±.00001

USE:

1. Calling Sequence
Load index register with line number of subroutine

```
LDA   θ (see paragraph 3, below)
LDC   return command
TRU   subroutine
```

Transfer is to 350LL for sine θ and to 345LL for cosine θ, where LL is the line number of the subroutine.

2. Upon exit from the subroutine, the sin or cos will be in the A register scaled at Q = 0. For example:

A_{22}

Sin 30° = +4000000 = 0 100.000 000 → 000

Sin 90° = +7777777 = 0 111 111 111 → 111

3. The argument, θ, must be placed in the A register as an octal fraction of 360° scaled at Q = 0. For example:

A_0 A_{22}

0° = +0000000 = $0_\wedge$000 000 000 → 000 = 0

90° = +2000000 = $0_\wedge$010 000 000 → 000 = +.25

270° = +6000000 = $0_\wedge$110 000 000 → 000 = +.75

-90° = -6000000 = $1_\wedge$110 000 000 → 000 = -.25

USE (cont.):

4. The Sine-Cosine subroutine will normally be on the same tape as the Arctan subroutine, as both programs occupy the same memory line.

METHOD:

This subroutine essentially makes use of the following polynomial approximation:

$$\sin \frac{\pi}{2} X = C_1 X + C_3 X^3 + C_5 X^5 + C_7 X^7$$

Where: $-1 \leqslant X \leqslant 1$

$$C_1 = 1.570794852$$
$$C_3 = .645920978$$
$$C_5 = .079487663$$
$$C_7 = -.004362476$$

Reference: "Approximations for Digital Computers," C. Hastings, Princeton University Press, 1955, page 139.

PB 250 PROGRAM LISTING

PROBLEM SIN COS | PAGE 1 OF 3

PROGRAMMER C. YINGST | DATE 2-15-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
34503$	346S06031	LDB	COS θ ENTRANCE, θ AT 0
34603$	+2000000		90° SCALED AT 0 θ → F 15 COS 0 = SIN (90-θ)
34703$	355S1100;	STA	
35003$	352 35031	TAN	SIN θ ENTRANCE, θ AT 0
35103$	003S1000;	STC	CHECK IF NEGATIVE ANGLE
35203$	353S06031	LDB	ELIMINATE NEGATIVE ANGLE 180° SCALED AT 0.
35303$	+4000000		
35403$	355S1100;	STA	θ → F15
35603$	371S0200;	IBC	IF COS, COS θ = SIN (90-θ)
37203$	373S0100;	IAC	
37403$	375S1500;	SUB	IF SIN, SIN θ = SIN (180-θ)
37603$	000 35031	TAN	
37703$	002S37031	TRU	
00003$	001S14031	ADD	ELIMINATE CARRY INTO SIGN POSITION. OR MAKE COS (-θ) = COS θ
00103$	-0000000		
00203$	003S1200;	STB	EXIT IN F03.
00403$	005S14031	ADD	IS θ ⩾ 180° ?
00503$	+4000000		YES IF NEGATIVE.
00603$	021 35031	TAN	
00703$	010S14031	ADD	
01003$	+2000000		
01103$	012S56031	CAM	IS θ = 90°? YES IF OVERFLOW.
01203$	-0000000		
01303$	016 75031	TOF	
01403$	027 35031	TAN	IS θ >90° ? YES IF NEGATIVE
01503$	051S37031	TRU	
01603$	323S05031	LDA	SIN 90 = 1
02103$	022S14031	ADD	IS X ≥ 270 ?
02203$	-6000000		
02303$	024S56031	CAM	
02403$	-0000000		CHECK FOR X = 270°
02503$	324 75031	TOF	

PB 250 PROGRAM LISTING

PROBLEM SIN COS

PAGE 2 OF 3

PROGRAMMER C. YINGST

DATE 2-15-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
02603$	051 35031	TAN	TEST FOR X > 270 YES IF NEGATIVE
02703$	030S1100;	STA	
03103$	032S56031	CAM	
03203$	+6000000		CHECK FOR X = 180°
03303$	037 75031	TOF	
03403$	035S4500;	CLA	COMPLEMENT ARGUMENT
03603$	050S1500;	SUB	
03703$	326S4500;	CLA	SIN 180° = 0
05103$	054S2100;	LSD	X SCALED AT 0
05403$	055S1100;	STA	X → F15 SCALED AT 0
05603$	057S0100;	IAC	X → C REG. SCALED AT 0
06003$	075S0600;	LDB	X → B REG.
07603$	125S3200;	MUP	X^2 SCALED AT 0
12503$	126S0100;	IAC	X^2 → C REG.
12703$	130S1000;	STC	X^2 → F 10
13103$	132S06031	LDB	C_7 SCALED AT -3 INTO B REG.
13203$	-7560415		C_7 = - .004362476 AT -3.
13303$	163S3200;	MUP	S=24)8 X^2 C_7 SCALED AT -2.
16303$	164S14031	ADD	$C_5 + X^2 C_7$ SCALED AT -3
16403$	+2426247		C_5 = .079487663 SCALED AT -2
16503$	166S0200;	IBC	X^2 → B REG.
16703$	170S0100;	IAC	$C_5 + X^2 C_7$ SCALED AT 5
17103$	220S3200;	MUP	S=26)8 $X^2 C_5 + X^4 C_7$ → A REG. AT -2.
22003$	223S2200;	RSI	$(X^2 C_5 + X^4 C_7)$ SCALED AT 0.
22303$	224S14031	ADD	$(C_3 + X^2 C_5 + X^4 C_7)$ AT 0.
22403$	-2652235		C_3 = -.645920978 AT 0.
22503$	226S0100;	IAC	$(C_3 + X^2 C_5 + X^4 C_7)$ → C REG.
22703$	230S0600;	LDB	X^2 SCALED AT 0. → B REG.
23103$	261S3200;	MUP	S=27)8 $(X^2 C_3 + X^4 C_5 + X^6 C_7)$ AT 1.
26103$	262S14031	ADD	$(C_1 + X^2 C_3 + X^4 C_5 + X^6 C_7)$
26203$	+6220773		C_1 = 1.570794852 AT 1.
26303$	264S0100;	IAC	

PB 250 PROGRAM LISTING

PROBLEM SIN COS PAGE 3 OF 3

PROGRAMMER C. YINGST DATE 2-15-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
26503$	275S0600;	LDB	X SCALED AT ZERO
27603$	325S3200;	MUP	$S=26)_8$ $XC_1+X^3C_3+X^5C_5+X^7C_7$ AT 1.
32303$	+7777776		
32403$	326S14031		VALUE OF SIN 90
32503$	327S2100;	LSD	
32603$	+0000001		VALUE OF SIN 270
32703$	343S7500;		EXIT (TURN OFF OVERFLOW)
00001.			ANS. A REG. SCALED AT 0.

IDENTIFICATION: ARCTAN (X/Y)

AUTHOR: C. Yingst, PBCC

ACCEPTED: January 18, 1961

PURPOSE: To find the principal value, in radians, of $\tan^{-1}(X/Y)$ for a given pair of arguments.

SPACE: REQUIRED 102 words in one long line, plus F01, F03, F05, F11, F14, F15 of line 00. The storage requirements of the Arctan subroutine are compatible with those of the Sin-Cos subroutine (Cat. No. 500). That is, the Arctan and Sin-Cos subroutines can occupy the same line simultaneously.

TIMING: 4.608 milliseconds maximum execution time.

ACCURACY: ±.00001

USE: 1. Calling Sequence

Load Index register with line number of subroutine.

```
LDA     X
LDB     Y
LDC     return command
TRU     subroutine
```

Note: X and Y can be at any scaling, but the scaling of X must equal the scaling of Y.

The subroutine is entered at sector 040, line LL, where LL is the line number of the subroutine, and has been previously loaded into the Index register.

2. Upon exit from the subroutine, θ = arctan (X/Y) will be in the A register in radian units scaled at Q = 1.

3. The Arctan subroutine will normally be on the same tape as the Sin-Cos subroutine, since both programs occupy the same memory line.

METHOD:

Polynomial approximation for $0 < (X/Y) < \tan 7.5^{\circ}$ and continued fractions for $\tan 7.5^{\circ} \leqslant (X/Y) \leqslant 1$.

Should $(X/Y) > 1$, the cotan^{-1} is found of (Y/X) for $\tan^{-1} (X/Y) = \text{cotan}^{-1} (Y/X)$.

The polynomial approximation used is:

$$\tan^{-1} N \approx N \left[C_0 - N^2 (C_1 - C_2 N_2) \right]$$

$$\text{with} \quad \begin{array}{l} C_0 = 0.9999999 \\ C_1 = 0.3332966 \\ C_2 = 0.1957408 \end{array}$$

The continued fractions are:

$$\tan^{-1} N \approx N \left\{ A_0 + \frac{A_1 |}{|N^2 + B_1} - \frac{A_2}{|N^2 + B_2} \right\}$$

$$\text{with} \quad \begin{array}{ll} A_0 = 0.2388229 & \\ A_1 = 2.4452053 & B_1 = 3.9435297 \\ A_2 = 1.3147472 & B_2 = 1.7982496 \end{array}$$

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM ARCTAN

PROGRAMMER C. YINGST

PAGE 1 OF 4

DATE 2-21-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
04007$	041S1000;	STC	ENTRANCE. EXIT INTO F01
04207$	043S1200;	STB	Y INTO F03
04407$	045S1100;	STA	X INTO F05
04607$	052 35071	TAN	IS X NEG. ?
04707$	055 36071	TBN	IS Y NEG. ?
05007$	063S1500;	SUB	X - Y
05507$	063S1400;	ADD	X + Y
06407$	105 35071	TAN	IS Y > X ? YES IF NEG.
06507$	112S37071	TRU	X > Y
05207$	057 36071	TBN	IS Y NEG ?
05307$	103S1400;	ADD	X + Y
05707$	103S1500;	SUB	X - Y
10407$	112 35071	TAN	IS X> Y ? YES IF NEG.
10507$	106S0200;	IBC	GET X/Y.
10707$	110S06071	LDB	SET INDICATOR WORD, θ< 45°.
11007$	-0000000		Y INTO C REG.
11107$	125S0500;	LDA	X INTO A REG.
11207$	113S4400;	CLC	GET Y/X.
11407$	115S0300;	ROT	SET INDICATOR WORD, θ > 45°.
11707$	125S0400;	LDC	Y INTO A REG.
12607$	135S1200;	STB	X INTO B REG.
13607$	140S4300;	CLB	
14107$	170S3100;	DIV	N = X/Y OR
17007$	172S1200;	STB	N = Y/X → F12
17307$	174S1200;	STB	N → F14 SCALED AT O.
17507$	176S0200;	IBC	CHECK FOR AN N OF UNITY.
17707$	201S2200;	RSI	
20107$	204 36001	TBN	
20207$	210 34001	TCN	N = 1
20307$	213S37071	TRU	

PROBLEM ARCTAN PAGE 2 OF 4

PROGRAMMER C. YINGST DATE 2-21-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
20407$	213 3407I	TCN	
20507$	206S4500;	CLA	N =1
20707$	225S5600;	CAM	TEST FOR N= 0
22607$	232S4300;	CLB	
23307$	235S2100;	LSD	
23507$	261 7500;	TOF	
23607$	316S0507I	LDA	N = 0, ARCTAN O = 0
31607$	-4667403	-45	-45° → A REG.
31707$	070 3407I	TCN	CHECK IF N = -∞
32007$	361S3700;	TRU	N = -1, ARCTAN (-1) = -45°
07007$	072S2100;	LSD	N = -∞, ARCTAN -∞= -90
07207$	101S3700;	TRU	
21007$	211S0507I	LDA	ARCTAN 1 = 45°
21107$	+3110375	+45°	
21207$	221S3700;	TRU	
21307$	214S0600;	LDB	GET N^2 SCALED AT 2.
21507$	244S3200;	MUL	
24407$	245S4400;	CLC	
24607$	247S0000;	MRG	
25007$	251S1100;	STA	
25207$	253S1507I	SUB	
25307$	+0106773		IS N < TAN 7.5°
25407$	256 3500I	TAN	
25507$	263S0100;	IAC	COMPUTE ARCTAN N > 7.5°
26407$	267S2200;	RSI	
26707$	270S1407I	ADD	$N^2 + B_2^*$ SCALED AT 2.
27007$	+3461320	B^*_2	
27107$	272S0100;	IAC	
27307$	274S0507I	LDA	A^*_2 SCALED AT 2.
27407$	+2502232	A^*_2	

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM ARCTAN PAGE 3 OF 4

PROGRAMMER C. YINGST DATE 2-21-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
27507$	303S4300;	CLB	$A^*_2 \ / \ (N^2 + B^*_2)$ SCALED AT 2.
30407$	330S3100;	DIV	
33007$	331S0500;	LDA	N^2 INTO A REG AT 0.
33207$	334S1200;	STB	$P = A^*_2 \ /(N^2 + B^*_2) \longrightarrow$ F14
33507$	341S2200;	RSI	N^2 SCALED AT 2.
34107$	342S14071	ADD	$(N^2 + B^*_1)$ SCALED AT 3.
34207$	+3743054	B^*_1	B^*_1 SCALED AT 3.
34307$	354S1500;	SUB	$(N^2 + B^*_1 - P)$ SCALED AT 2.
35507$	356S0100;	IAC	
35707$	365S05071	LDA	A^*_1 , SCALED AT 3.
36507$	+1161762	A^*_1	
36607$	367S4300;	CLB	
37007$	017S3100;	DIV	$A^*_1 \ /(N^2 + B_1^* - P)$ SCALED AT 1.
01707$	026S0300;	ROT	
03007$	077S14071	ADD	$A^*_0 + A^*_1 \ /(N^2 + B^*_1 - P)$ SCALED AT 1.
07707$	+0751055	A^*_0	
10007$	101S0100;	IAC	
10207$	112S0600;	LDB	N SCALED AT 0. INTO B REG.
11307$	120S37071	TRU	
12007$	147S3200;	MUP	$N[A^*_0 + A^*_1 \ /(N^2 + B^*_1 - P)$
14707$	151S1100;	STA	SCALED AT 1.
15207$	155S0600;	LDB	CHECK IF ARCTAN N ∠ 45°
15607$	161 3600;	TBN	YES IF NEG. EXIT.
15707$	166 35071	TAN	
16007$	161S05071	LDA	ARCTAN N $>$ 45.°
16107$	+6220772	π/2	ADJUST ANSWER BY π/2.
16207$	217S37071	TRU	
16607$	216S05071	LDA	
21607$	-1557006	-π/2	
21707$	231S1500;	SUB	

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM ARCTAN PAGE 4 OF 4

PROGRAMMER C. YINGST DATE 2-21-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
23207$	241S3700;	TRU	EXIT COMPUTE ARCTAN N < 7.5°
25607$	257S06071	LDB	
25707$	+1441602	C_2	
26007$	307S3200;	MUP	$N^2 \quad C_2$
30707$	310S15071	SUB	$N^2 \ C_2 - C_1$
31007$	+2525136	C_1	
31107$	313S0300;	ROT	
31507$	344S3200;	MUP	$N^2 \ [N^2 C_2 - C_1]$
34407$	361S14071	ADD	$C_0 + N_2[N^2 C_2 - C_1]$
36107$	+7777777	C_0	
36207$	364S2200;	RSI	
36407$	100S37071	TRU	
00001.			

IDENTIFICATION: LOGARITHM, BASE 2, e, 10
Fixed Point Subroutine

AUTHORS: P. Jarvie, PBCC
F. Valadez, PBCC

ACCEPTED: April 15, 1961

PURPOSE: To calculate the logarithm, base 2, e, or 10 of a single precision, binary-scaled, fixed point number.

RESTRICTIONS:

1. The argument, X, must be greater than zero.

2. The binary scale, Q, of the argument must be no greater, in magnitude, than 21.

SPACE: The routine is line relocatable and requires 86 sectors of one long line plus F01, F04 through F06, and F10 through F17. The routine is designed to merge in the same line with k^X.

TIMING:

Base 10 - - 2.62 ms
Base e - - 2.81 ms
Base 2 - - 3.00 ms

ACCURACY:

$|\varepsilon| < 10^{-6}$

$\varepsilon = A - T$

A = computed value
T = true value

USE:

1. Calling Sequence

Set Index register with line number of subroutine

```
LDA     argument @Q
LDB     Q of argument @21
LDC     return command
TRU     entry
```

Entry is 005LL for $\log_2$, 025LL for $\log_e$, or 045LL for $\log_{10}$, where LL is the line location of the subroutine.

2. Output

The subroutine exists with the logarithm of the argument in the combined AB registers scaled @Q = 6, and with overflow off.

3. Error Alarms

If the argument in the A register is not greater than zero, the subroutine exists with all one's in AB and with overflow on.

4. Program Tape

The logarithm subroutine is on the same tape as the exponential subroutine (catalog number 0113), since both have compatible storage requirements.

METHOD:

X is reduced to a range where a continued fraction approximation is valid.

We have $$\log_k X \equiv \log_k 2 \log_2 X$$

and $$\log_2 X \equiv n + \log_2 2^{-n} X$$

where n is an integer and $2^{-n}X \in [1/2, 1)$

Now, letting $X_n = 2^{-n}X$

$$\log_2 X_n = \log_2 e \log_e X_n$$

so that $$\log_k X = \log_k 2 \, (n + \log_2 e \log_e X_n)$$

$$= (\log_k 2)n + \log_k e \log_e X_n$$

If X_n is far away from 1, a multiplier, 1.5, is introduced for further reduction, so that, finally,

$$\log_k X = M_k n + K_k + A_k(t)$$

where $M_k = \log_k 2$

$$K_k = \begin{cases} -\log_k 1.5, & X_n < .875 \\ 0, & X_n \geqslant .875 \end{cases}$$

and

$$A_k(t) = t \left[A_{k_o} + \frac{A_{k_1}|}{|t + A_2} + \frac{A_3|}{|t + A_4} \right]$$

where $A_{ko} = 1/9 \ (\log_k e)$

$A_{k1} = 17/9 \ (\log_k e)$

$A_2 = 124/51$

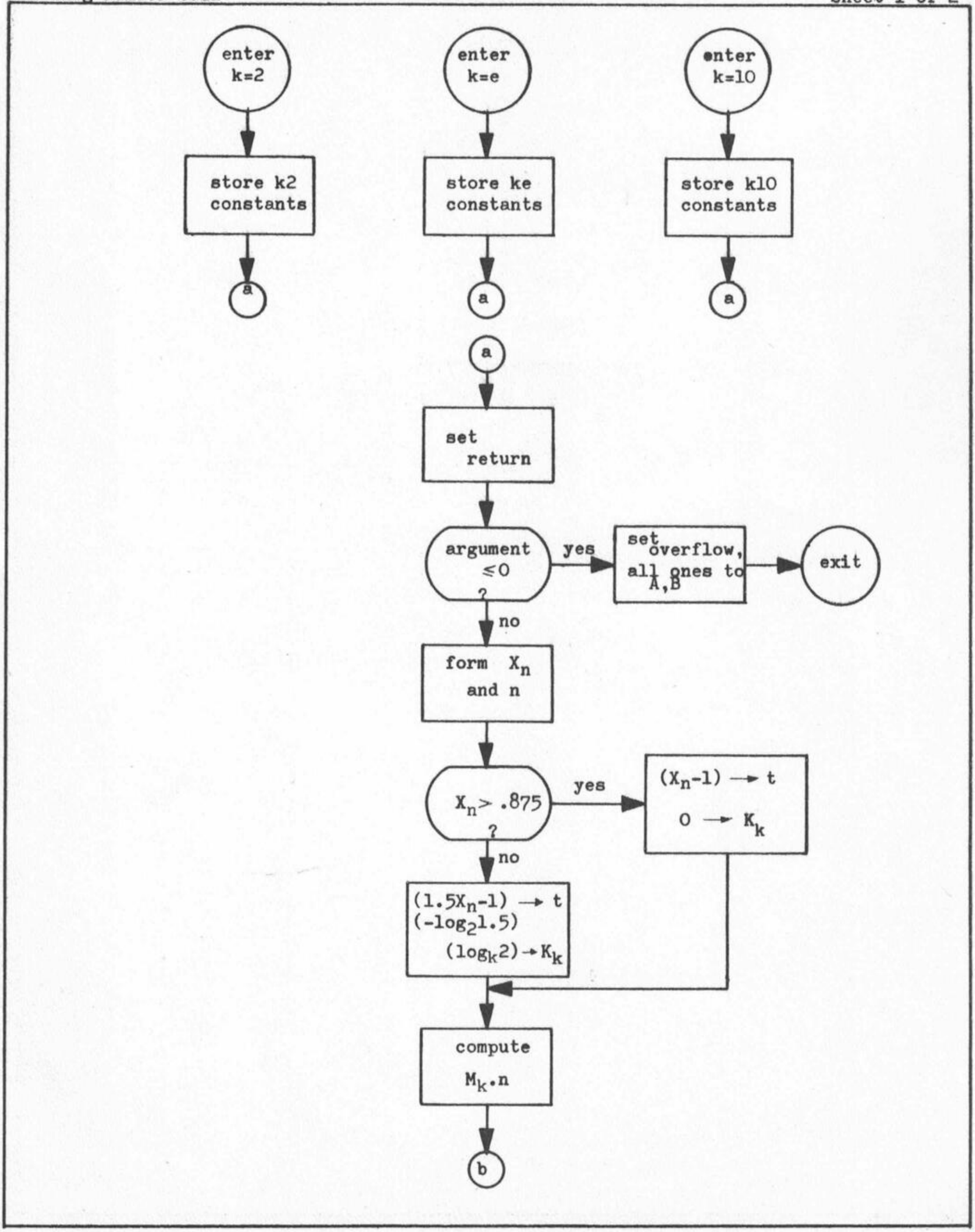
$A_3 = -1250/(51)^2$

$A_4 = 80/51$

Flow Diagram

LOGARITHM, BASE 2, e, 10

Catalog Number 0112 Sheet 1 of 2

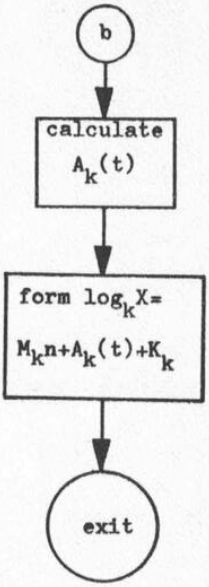


Flow Diagram

LOGARITHM, BASE 2, e, 10

Catalog Number 0112

Sheet 2 of 2



pb Packard Bell Computer

PB 250 PROGRAM LISTING

CATALOG NUMBER 0112

PROBLEM LOGARITHM, BASE 2, ε, 10

PAGE 1 OF 4

PROGRAMMER F. VALADEZ

DATE 5-5-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
00506$	015S7100;	MCL	
00606$	+7777777	CONST	
01006$	+0510226	CONST	
01106$	+2256002	CONST	STORE CONSTANTS FOR κ = 2
01206$	+2426072	CONST	
01406$	+2563176	CONST	
01506$	056S1000;	STC	SET RETURN
02506$	035S7100;	MCL	
02606$	+5427103	CONST	
03006$	+0343434	CONST	STORE CONSTANTS FOR κ = ε
03106$	+1476311	CONST	
03206$	-6161616	CONST	
03406$	+1707070	CONST	
03506$	056S1000;	STC	SET RETURN
04506$	055S7100;	MCL	
04606$	+2321012	CONST	
05006$	+0142647	CONST	STORE CONSTANTS FOR κ = 10
05106$	+0550505	CONST	
05206$	-2210664	CONST	
05406$	+0644005	CONST	

pb Packard Bell Computer

PB 250 PROGRAM LISTING

CATALOG NUMBER 0112

PROBLEM LOGARITHM, BASE 2, E, 10

PAGE 2 OF 4

PROGRAMMER F. VALADEZ

DATE 5-5-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
05506$	056S1000;	STC	SET RETURN
05706$	076 3506I	TAN	CHECK FOR ARGUMENT ≦ 0
06006$	063S5606I	CAM	
06306$	+0000000	CONST.	
06406$	076 7506I	TOF	
06506$	066S0200;	IBC	FORM X_N AND N
06706$	070S4300;	CLB	
07106$	116S2000;	NAD	
07606$	126S4500;	CLA	ERROR RETURN
11606$	117S1100;	STA	TEST FOR RANGE OF X_N
12006$	121S1506I	SUB	
12106$	+7000000	CONST	
12206$	124 3506I	TAN	
12306$	131S4500;	CLA	
12406$	125S1406I	ADD	
12506$	+7000000	CONST	
12606$	130S2210;	SRT	
12706$	163S5606I	CAM	ERROR RETURN
13006$	131S0600;	LDB	SET K_K
13206$	133S1200;	STB	
13406$	135S1406I	ADD	FORM T
13506$	-0000000	CONST	
13606$	137S1400;	ADD	
14006$	141S1100;	STA	
14206$	144S0200;	IBC	COMPUTE $M_K \cdot N$
14506$	146S0400;	LDC	
14706$	157S3200;	MUP	
15706$	161S0400;	LDC	
16206$	164S1300;	STD	

pb Packard Bell Computer

PB 250 PROGRAM LISTING

CATALOG NUMBER 0112

PROBLEM LOGARITHM, BASE 2, E, 10

PAGE 3 OF 4

PROGRAMMER E. VALADEZ

DATE 5-5-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
16306$	+0000000	CONST	ERROR RETURN
16406$	203S0506I	LDA	
16606$	167S0100;	IAC	COMPUTE A_{IK} (T)
17006$	172S2210;	SRT	
17206$	173S1606I	DPA	
17306$	-0000000	CONST	
17406$	+6214431	CONST	
17506$	176S0100;	IAC	
17706$	200S0706I	LDP	
20006$	-3676637	CONST	
20106$	-6047606	CONST	
20206$	231S3100;	DIV	
20306$	-7777777	CONST	ERROR RETURN
20406$	205S0606I	LDB	
20506$	-7777777	CONST	
20606$	376S3700;	TRU	
23106$	234S0500;	LDA	COMPUTE A_{K} (T)
23506$	236S0300;	ROT	
24006$	241S1400;	ADD	
24206$	245S2210;	SRT	
24506$	246S1606I	DPA	
24606$	-4000000	CONST	
24706$	+4671563	CONST	
25006$	252S0600;	LDB	
25306$	254S0100;	IAC	
25506$	304S3100;	DIV	
30406$	305S0300;	ROT	
30706$	310S1400;	ADD	
31106$	313S0100;	IAC	
31406$	321S0600;	LDB	

pb Packard Bell Computer

PB 250 PROGRAM LISTING

CATALOG NUMBER 0112

PROBLEM LOGARITHM, BASE 2, E, 10

PAGE 4 OF 4

PROGRAMMER F. VALADEZ

DATE 5-5-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
32206$	351S3200;	MUP	
35106$	353S1500;	SUB	FORM $\Lambda (T) + K_K$
35406$	362S2210;	SRT	
36206$	364S1600;	DPA	FORM $A (T) + K_K + M_K N$
36606$	367S47061	EXF	
36706$	+0077777	CONST	
37006$	376S3700;	TRU	EXIT

Catalog Number 0113

IDENTIFICATION: EXPONENTIAL, BASE 2, e, 10
Fixed Point Subroutine

AUTHORS: P. Jarvie, PBCC
F. Valadez, PBCC

ACCEPTED: April 28, 1961

PURPOSE: To calculate 2, e, or 10 to a given power, X.

RESTRICTIONS: 1. The argument, X, must be limited as follows:

a) Base 2 : $X < 20.4$

b) Base e : $X < 14.2$

c) Base 10 : $X < 6.15$

2. The binary scale, Q, of the argument must lie within the range:

$$-6 \leqslant Q \leqslant 19$$

STORAGE: 81 sectors in one long line, plus F01, F05, F06, F12, F13, and F15. This routine is designed to merge in the same line with Logarithm, Base 2, e, 10 (catalog number 0112). Note: Sector 376 of the long line is used.

TIMING:

Base 2 - - 3.32 ms

Base e - - 3.56 ms

Base 10 - - 3.80 ms

ACCURACY:

$$\varepsilon < 2 \cdot 10^{-6}$$

For $X < 0$, $\varepsilon = |\varepsilon_A|$

For $X \geqslant 0$, $\varepsilon = |\varepsilon_R|$

ACCURACY (cont.): where $\varepsilon_A = A - T$

$\varepsilon_R = \varepsilon_A / T$

and A = computed value

T = true value

USE:

1. Calling Sequence

Set Index register with line number of subroutine.

LDA X@Q

LDB Q_X@21

LDC return command

TRU entry

Entry is 000LL for 10^X, 020LL for e^X, and 040LL for 2^X, where LL is the line location of the subroutine.

2. Output

The subroutine exits with the value of the exponential in the combined AB registers, scaled at 21.

3. Error Alarms

There are no error alarms. Specifying an X or Q outside the allowable range will cause an erroneous answer to be computed.

METHOD:

A rational function approximation, A, was found to e^X which is very accurate for $X \in \sim (-.13, +.13)$:

$$A = 1 + \frac{12X}{(X - 3)^2 + 3}$$

METHOD (cont.): Now, since $K^X \equiv 2^{(\log_2 K)X} = 2^{n+F}$ (K = 2, e, or 10)

where n an integer

and $F \in \left[-\frac{1}{2}, \frac{1}{2}\right)$

and $2^F \equiv e^{(\log_e 2)F} = e^f$

where $f \in \sim(-.35, 35)$

and, also, since

$$e^f = B_j e^{f + 1j} \quad j = 1, 2, 3$$

for $f \in I_j$

where $1_j = -\log_e B_j$

and
$$\begin{aligned} B_1 &= .75 \text{ for } I_1 = (-.35, -.12) \\ B_2 &= 1.00 \text{ for } I_2 = [-.12, .12] \\ B_3 &= 1.25 \text{ for } I_3 = (.12, .35) \end{aligned}$$

Noting that $f + 1_j = Z \in \sim(-.13, 13)$

we have $K^X = 2^n B_j e^Z$

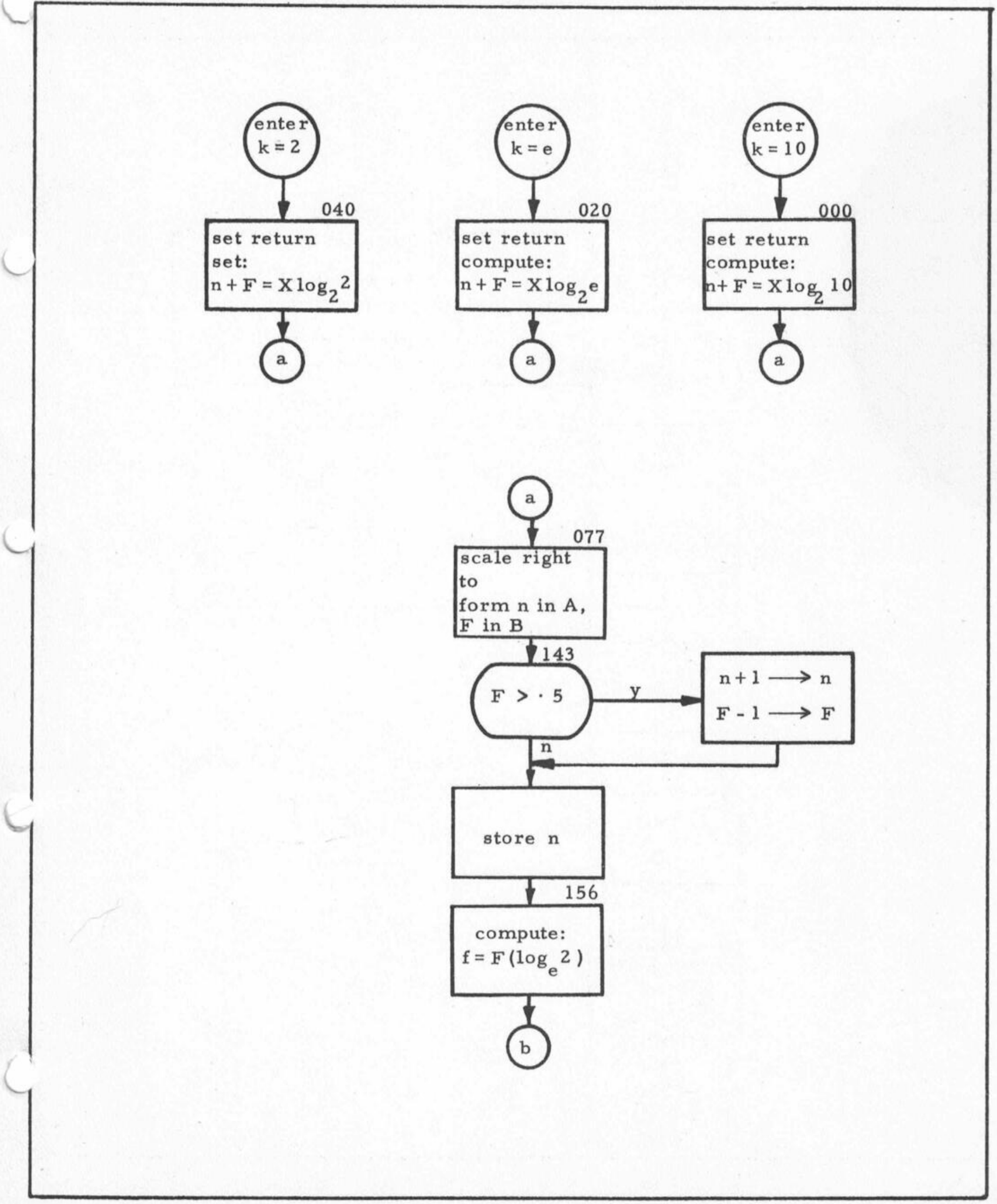
or, finally, $K^X = 2^n B_j A(Z)$

Flow Diagram

EXPONENTIAL, BASE 2, e, 10

Catalog Number 0113

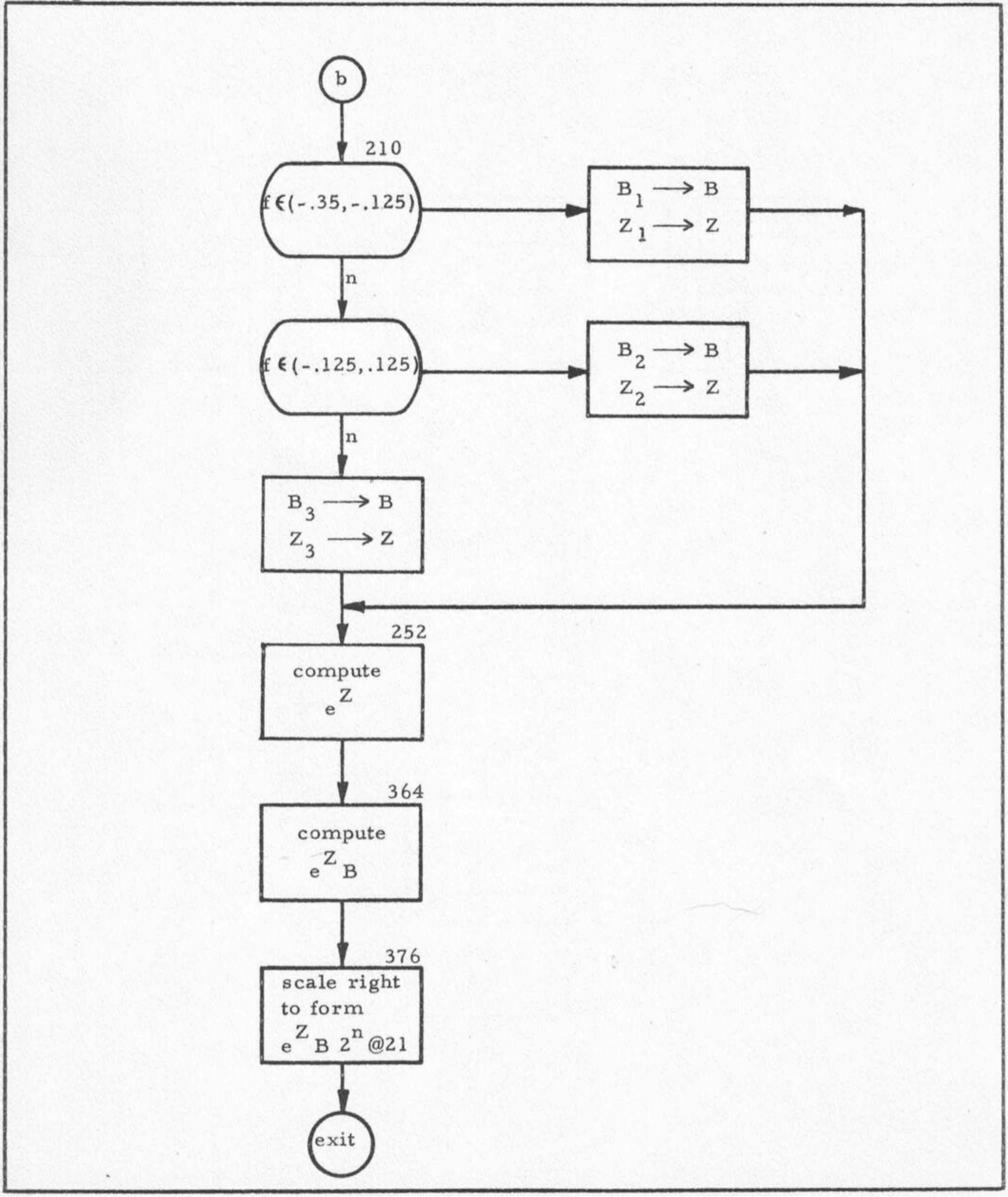
Sheet 1 of 2



Flow Diagram

EXPONENTIAL, BASE 2, e, 10

Catalog Number 0113 Sheet 2 of 2



pb Packard Bell Computer

PB 250 PROGRAM LISTING

CATALOG NUMBER 0113

PROBLEM EXPONENTIAL, BASE 2, E, 10

PAGE 1 OF 3

PROGRAMMER F. VALADEZ

DATE 5-8-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
00006$	001S1000;	STC	
00206$	016S0406I	LDC	
01606$	+6511517	CONST	ENTRY FOR K = 10
01706$	025S1300;	STD	
02006$	021S1000;	STC	
02206$	023S0406I	LDC	
02306$	+2705244	CONST	ENTRY FOR K = E
02406$	025S1300;	STD	
02706$	031S0300;	ROT	COMPUTE N + F IF K = 10, E
03306$	062S3200;	MUP	
04006$	041S1000;	STC	
04206$	045S1300;	STD	
04706$	055S4300;	CLB	ENTRY FOR K = 2
05606$	061S2210;	SRT	
06106$	065S0400;	LDC	
06206$	065S0400;	LDC	
06606$	067S0100;	IAC	
07006$	073S7100;	MCL	
07206$	+0000024	CONST	
07306$	074S1506I	SUB	FORM : N IN A REGISTER F IN B REGISTER
07406$	+0000023	CONST	
07506$	076S0100;	IAC	
07706$	131S2300;	SAI	
13106$	132S1500;	SUB	
13306$	136S0100;	IAC	
13706$	142S4500;	CLA	
14306$	150 3606I	TBN	TEST FOR F > .5
14406$	146S2210;	SRT	
14606$	155S1000;	STC	
15006$	151S0506I	LDA	

pb Packard Bell Computer

PB 250 PROGRAM LISTING

CATALOG NUMBER 0113

PROBLEM EXPONENTIAL BASE 2, E, 10

PAGE 2 OF 3

PROGRAMMER F. VALADEZ

DATE 5-8-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
15106$	-7777777	CONST	
15206$	154S2200;	RSI	
15406$	155S1000;	STC	
15606$	160S0406I	LDC	
16006$	+1305621	CONST	COMPUTE F = F (LOG $_e$ 2)
16106$	210S3200;	MUP	
21006$	223 3506I	TAN	
21106$	212S1506I	SUB	
21206$	+0200000	CONST	
21306$	221 3506I	TAN	
21406$	215S1406I	ADD	
21506$	-1633401	CONST	
21606$	217S0406I	LDC	
21706$	+0000017	CONST	
22006$	251S1000;	STC	
22106$	222S1406I	ADD	
22206$	+0200000	CONST	
22306$	224S1406I	ADD	
22406$	+0200000	CONST	DETERMINE RANGE OF F AND SELECT CORRESPONDING VALUES OF B_j AND Z
22506$	234 3506I	TAN	
22606$	227S1506I	SUB	
22706$	+6200000	CONST	
23006$	232S0406I	LDC	
23206$	+0000014	CONST	
23306$	251S1000;	STC	
23406$	236S1506I	SUB	
23606$	+5531324	CONST	
23706$	243S0406I	LDC	
24306$	+0000011	CONST	
24406$	251S1000;	STC	

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM EXPONENTIAL BASE 2, ε, 10

PROGRAMMER F. VALADEZ

CATALOG NUMBER 0113

PAGE 3 OF 3

DATE 5-8-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
25206$	253S0100;	IAC	
25406$	256S4606I	AOC	
25606$	-7777777	CONST	
25706$	306S3200;	MUP	
30606$	307S4300;	CLB	
31006$	312S1406I	ADD	
31206$	+1400000	CONST	COMPUTE A $(Z) \equiv \varepsilon^{Z}$
31306$	314S0100;	IAC	
31506$	316S1406I	ADD	
31606$	+6000000	CONST	
31706$	346S3100;	DIV	
34606$	351S0400;	LDC	
35206$	353S1606I	DPA	
35306$	+2525253	CONST	
35506$	363S0200;	IBC	
36406$	372S3200;	MUP	COMPUTE B_j A (Z)
37206$	374S2110;	SLT	
37406$	375S0400;	LDC	
37606$	053S2300;	SAI	FORM 2^N B_j A(Z)
05306$	061S3700;	TRU	EXIT