HEWLETT hp PACKARD

Service Manual

H/P Part No. 09100-90034

HEWLETT-PACKARD CALCULATOR

Models: 9100A
9100B

SERIALS PREFIXED
945-03675 – 9100A
938-02251 – 9100B

Page 73, Manual Backdating Changes adapts this manual to all prior serial prefixes.

P.O. Box 301, Loveland, Colorado, 80537 U.S.A.

PSEUDO-HARDWARE PROBLEMS

INTRODUCTION

Several operating characteristics of the 9100A/B Calculator could be interpreted as instrument malfunctions; therefore, the Field Service Engineer should ensure that an instrument malfunction exists prior to troubleshooting a calculator. This section will cover the most common pseudo-hardware of the A/B, A and B, respectively.

9100A/B

IMPROPER MAGNETIC CARD ENTRY

If the calculator's memory is unchanged except for core memory location 0-0 (or the starting address), which is changed to 77_8, check to be sure the customer is not putting the magnetic card in backwards, that is, the printed side of the card toward the display bezel rather than toward the keyboard. If the card is put in the cardreader backward and the ENTER key is pressed, the read heads of the cardreader will have no data input; therefore, the program counter of the calculator will not be advanced. When the card is ejected from the cardreader, the read heads pick up the electrical noise of the card-eject switch and load an octal code 77 into the memory location 0-0 (or the starting address). The same symptom will be observed if an unrecorded or accidentally erased magnetic card is used.

PROGRAM STEPS IN MEMORY CHANGED TO Y→( ) OR ACC+

If, after one pass through a program, a register (or registers) previously containing program steps now contains Y→( ) or ACC + instructions (octal codes 40 and 60 respectively), check to be sure that the program has not loaded data in registers that previously contained program steps.

The explanation of why octal code 40 and 60 are predominate in this situation begins with the core memory and ends with the display. Throughout the following discussion the 9100A will be used as an illustration, but the material applies equally to the 9100B (unless otherwise stated).

The core memory of the calculator (See Figure 1) contains 19 accessible registers (0-9, a-f, X, Y, Z). (The 9100B also has an additional negative page (−) containing registers (−)0-9, (−)a-f.) The registers contain 14 characters, each character contains 6 bits (F20-F25).

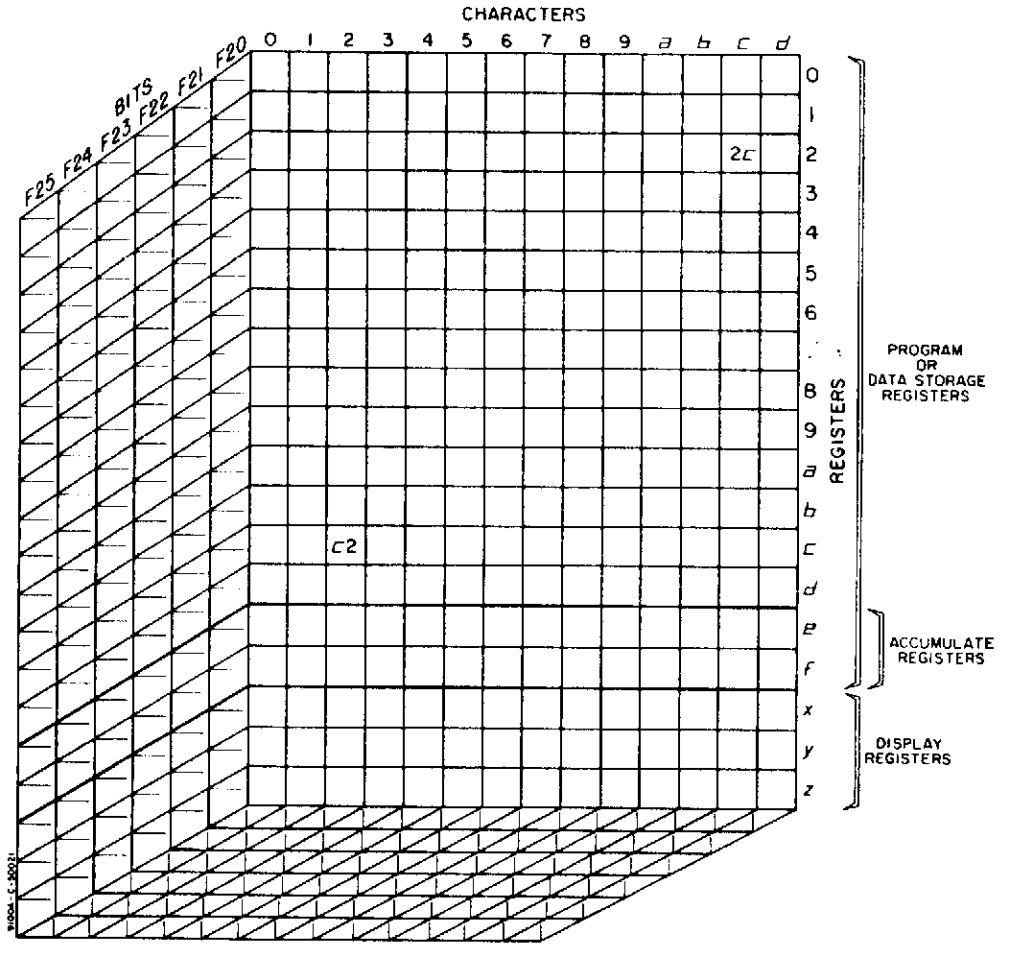

Figure 1. The 9100A's 19 Accessible Registers

Each character location will contain either an instruction or a digit. First, let's cover the situation of a character holding an instruction. The six bits are assigned the following octal weight:

BIT	F20	F21	F22	F23	F24	F25
OCTAL WEIGHT	1	2	4	10	20	40

A maximum of octal code 77 can be stored in a character, giving each character location the capability of storing any calculator instruction (see calculator pull-out card). The following chart contains examples of a character storing different calculator instructions.

9100A/B

PROGRAM STEPS IN MEMORY CHANGED TO Y→( ) OR ACC+ (Continued)

	F20	F21	F22	F23	F24	F25	BITS
	1	2	4	10	20	40	OCTAL WEIGHT
KEYS							TOTAL OCTAL WEIGHT
ACC+	0	0	0	0	1	1	60
STOP	1	0	0	0	0	1	41
$\sqrt{X}$	0	1	1	1	1	1	76
PAUSE	1	1	1	1	0	1	57

NOTE

The octal code 77 is not required for a keyboard instruction, but is used as an internal calculator instruction (In 10). (In the 9100B octal code 77 is the SUB RETURN instruction.)

When a character contains a digit rather than an instruction, only bits F20, F21, F22 and F23 are required to define that digit. Bits F24 and F25 contain the sign and blanking information. If bit F24 is a one, the digit is negative. If bit F25 is a one, the digit is blanked.

DIGIT	F20	F21	F22	F23	F24	F25	BITS	BLANKED	±
	1	2	4	10	20	40	OCTAL WEIGHT		
							TOTAL OCTAL WEIGHT		
1	1	0	0	0	1	0	21	NO	–
3	1	1	0	0	0	0	3	NO	+
5	1	0	1	0	1	0	25	NO	–
7	1	1	1	0	0	0	7	NO	+
9	1	0	0	1	1	0	31	NO	–

The CRT display is stored in the X, Y, and Z registers. The display of one of these registers in floating point is shown in Figure 2. It consists of ten digits (D_9 ---- D_0), two exponent digits (E_1 and E_0) and two guard digits (G_1 and G_0), which are always blanked. The two guard digits perform the function of rounding.

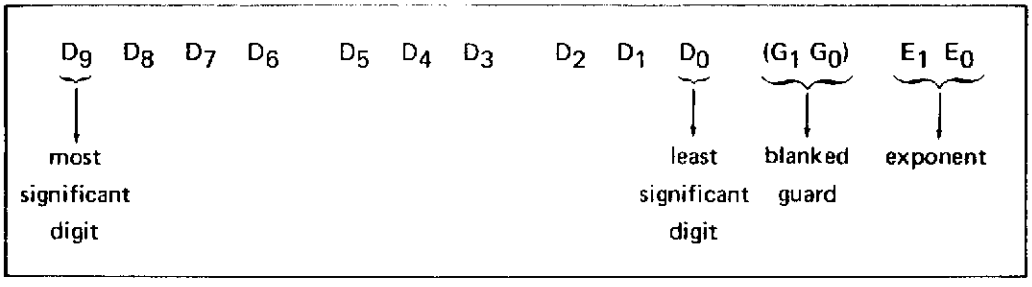

Figure 2. Display of Register X, Y or Z

If the CRT display in Figure 2 were being held in any register, it would be held in the following manner:

Register	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	Xa	Xb	Xc	Xd
Display	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	E_0	E_1	G_0	G_1

This can be demonstrated in the following manner:

SET:	RUN		
PRESS:	1	PRESS:	6
"	2	"	7
"	3	"	X→( )
"	4	"	9
"	5	"	GO TO ( ) ( )
"	6	"	9
"	7	"	0
"	8	SET:	PROGRAM
"	9		
"	0		
"	7		
"	8		
"	ENT EXP		

NOTE
For a 9100B, use the (+) page for all examples.

Pressing the STEP PRGM key will cause the contents of the 9 register to be displayed one character at a time in octal form.

Character	9-0	9-1	9-2	9-3	9-4	9-5	9-6	9-7	9-8	9-9	9-a	9-b	9-c	9-d
Octal	00	11	10	07	06	05	04	03	02	01	07	06	10	07
Decimal	0	9	8	7	6	5	4	3	2	1	7	6	8	7
Display	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	E_0	E_1	G_0	G_1

This procedure loaded 12 significant digits into the 9 register. The demonstration for a partial display produces quite different results:

SET:	RUN
PRESS:	1
"	2
"	3
"	X→( )
"	9
"	GO TO ( ) ( )
"	9
"	0
SET:	PROGRAM

9100A/B

PROGRAM STEPS IN MEMORY CHANGED TO Y→( ) OR ACC + (Continued)

The STEP PRGM key will display the contents of the 9 register in octal form one character at a time.

Register Location	9-0	9-1	9-2	9-3	9-4	9-5	9-6	9-7	9-8	9-9	9-a	9-b	9-c	9-d
Octal	40	40	40	40	40	40	40	03	02	01	02	00	40	40
Decimal	-	-	-	-	-	-	-	3	2	1	2	0	-	-
Display	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	E_0	E_1	G_0	G_1

In this example the octal code 40 is predominant, meaning an insignificant zero which is blanked. Storing a negative number produces slightly different results.

SET:	RUN
PRESS:	1
"	2
"	3
"	CHG SIGN
"	X→( )
"	9
"	GO TO ( ) ( )
"	9
"	0
SET:	PROGRAM

The STEP PRGM key will display the contents of the 9 register in octal form, one character location at a time.

Register Location	9-0	9-1	9-2	9-3	9-4	9-5	9-6	9-7	9-8	9-9	9-a	9-b	9-c	9-d
Octal	40	40	40	40	40	40	40	03	02	21	02	00	40	40
Decimal	-	-	-	-	-	-	-	3	2	1	2	0	-	-
Display	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	E_0	E_1	G_0	G_1

In this example the octal code in location 9-9 contains a twenty bit, which means that the number is negative. If the CHG SIGN had been pressed first, still another example of a negative entry would have occurred.

SET:	RUN		
PRESS:	CHG SIGN	PRESS:	9
"	1	"	0
"	2	SET:	PROGRAM
"	3		
"	X→( )		
"	9		
"	GO TO ( ) ( )		

9100A/B

The STEP PRGM key will display the contents of the 9 register in octal form, one character location at a time.

Register Location	9-0	9-1	9-2	9-3	9-4	9-5	9-6	9-7	9-8	9-9	9-a	9-b	9-c	9-d
Octal	60	60	60	60	60	60	60	23	22	21	02	00	60	60
Decimal	-	-	-	-	-	-	-	3	2	1	2	0	-	-
Display	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	E_0	E_1	G_0	G_1

Pressing CHG SIGN first has loaded octal code 20's in all of the digits. If the digit was a blanked zero, it is an octal code 40; the two taken together result in octal code 60 which is a negative, blanked decimal zero.

Now, back to the problem. The operator complained that after one pass through the program, steps previously containing program steps now contain Y → ( ) instructions (octal code 40) or ACC + instructions (octal code 60). If he has stored a positive number with insignificant zeros over the register containing program steps, he has octal code 40's in the character locations that contain the insignificant zeros. If he has loaded a negative partial display over a register containing program steps, he has octal code 60's in the characters not containing digits.

INCORRECT POLAR CONVERSION

With zeros in the X and Y registers and the DEGREES - RADIANS switch in the DEGREES position, pressing TO POLAR will cause 90 and zero, respectively, to appear in the X and Y registers. Performing the same operation with the DEGREES - RADIANS switch in the RADIANS position will cause 1.570 796 326 00 and zero to appear in the Y and X registers. This occurs because of the algorithm used in the TO POLAR conversion, which is:

$$\text{ARC TAN } \frac{Y}{X} = \emptyset$$

If the X and Y registers contained zeros, the calculator would divide zero into zero and obtain the instrument's interpretation of infinity (9.999 999 999 99). The ARC TAN of 9.999 999 999 99 is 90 degrees. 90 degrees in radians is 1.570 796 326 00.

PRESSING TWO KEYS SIMULTANEOUSLY

Pressing two keys simultaneously will result in any of three occurrences:

1. The number in the X register will not change.
2. Either key may be entered in the X register.
3. Zeros will be entered in the X register.

This is normal operation and requires no service action.

9100A/B

AMBIGUOUS EQUALITY

The "IF" instructions compare all twelve digits (i.e. ten displayed digits and two guard digits) and the two-digit exponent, of the numbers in the X and Y registers when testing for the condition indicated. However, if a number consists of digit nine (9) in the first eleven places and (one of digits) five (5) through nine (9) in the twelfth place, then that number is considered to be equal to the next higher (more positive) power of ten.

For example,
9.999 999 999 9(5 thru 9) x $10^2 = 10^3$
similarly,
9.999 999 999 9(5 thru 9) x $10^{-2} = 10^{-1}$

Despite their equality to the next higher power of ten, those numbers which have unequal digits only in the twelfth place are not considered to be equal to each other.

For example:
9.999 999 999 95 x 10^2 is not equal to
9.999 999 999 96 x 10^2.

INCORRECT INT X KEY OPERATION

The rounding performed on the calculator display when the FLOATING - - FIXED POINT switch is in the FIXED POINT position could lead an operator to believe the INT X key is not functioning properly. For example, placing the number 5.9 in the X register and setting the DECIMAL DIGITS to 0 will cause the calculator to display six (6). If the INT X key is then pressed, the display will change to five (5). This occurs because the FLOATING - - FIXED POINT switch affects only the calculator display, not the number stored in the calculator's internal register (which is always in floating notation). Even though the calculator was displaying six (6), the internal registers were carrying the number as 5.9. When INT X was pressed the decimal part of the number (stored in the internal registers) was dropped, causing the calculator to display five (5).

DISPLAY CHANGES IF AN "IF" KEY IS PRESSED

Pressing an "IF" key when a calculator is not running a program may cause the program counter to change location and, occasionally, may result also in a change of the display. This will occur because the calculator will process the key as though it were in a program and (depending on which "IF" key was pressed and the program steps in the memory) process the next two instructions in the calculator's memory or branch to the third following step and process it. The rule of thumb is: any "IF" key executed in this manner, will be executed as though the instruction were in the memory and encountered using the STEP PRGM key.

INCORRECT PROGRAM STEP LOADED INTO MEMORY FROM MAGNETIC CARD

If incorrect program steps (classically, octal code 77) are being loaded into the core memory (say, in the area of steps 2-1 thru 2-7) from a magnetic card, the program on the card may not have an END instruction. Some knowledge of the magnetic card reader operation is required to explain how this can happen.

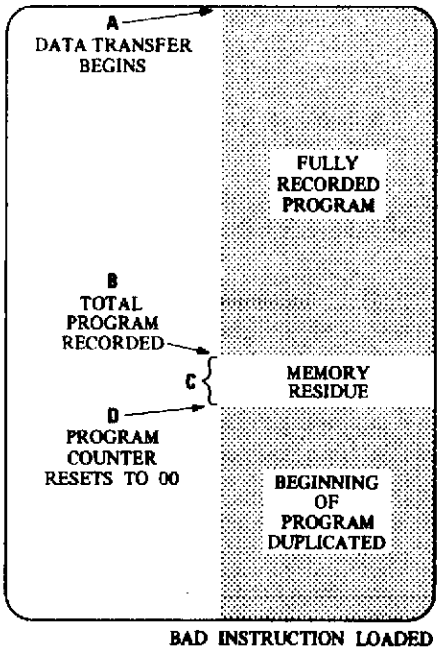

Figure 3. A Program Recorded On A Magnetic Card

When a magnetic card is recorded, the data transfer begins at point A in fig. 3. The program is fully recorded on the card at point B in fig. 3, but the recording will continue until the card is ejected. At C the unused memory is being recorded. At D the program counter reaches d-d and resets to 0-0. The recording process is uninterrupted and the beginning of the program is recorded again.

When a magnetic card is entered in the calculator, the data transfer begins at point A (see Figure 3). The program is fully entered in the calculator memory at point B. At this time, had there been an END instruction on the magnetic card, the cardreader heads would stop reading the magnetic card. If, however, there were no END instruction, the cardreader heads would continue reading the balance of the card. At D, the program counter reaches d-d and resets to 0-0 and the cardreader heads continue reading the card. At the end of the magnetic card, as it is ejected from the cardreader, circuit switches, which sense the presence of the magnetic card in the cardreader, close. The cardreader heads, which are still sensitive, pick up the electrical noise of the circuit switches' closure and transfer a bad instruction to the calculator memory.

9100A

KEYBOARD DIGIT ENTRY BECOMES A PART OF THE NUMBER IN THE X REGISTER

The following instructions do not terminate a preceding digit entry in the 9100A:

IF FLAG
SET FLAG
PAUSE
END
CHG SIGN
•
ENTER EXP

This characteristic of the 9100A operation can yield several different pseudo-hardware problems. For example, here is one:

> During diagnostic program execution, the PAUSE key is held pressed. When the program stops, there will be a three (3) in the X register. If the digit key five (5) is pressed the X register will contain thirty-five (35).

The way around this "problem" is, prior to making a digit entry, press any key that will terminate digit entry, such as CLEAR X.

PROGRAM EXECUTION STARTS WHEN STEP PRGM IS PRESSED

Any of the following instructions will initiate automatic program execution if they are encountered in the calculator memory with the STEP PRGM key when the calculator is in the RUN mode:

CONTINUE
PRINT/SPACE
FMT

CONTINUE - When STEP PRGM is pressed at this instruction the CONTINUE will override the STEP PRGM condition and automatically continue execution of the program.

FMT and PRINT/SPACE - Both of these instructions are used to control peripheral equipment. When a program is running, the program stops at these instructions and waits for a "continue" signal from the peripheral equipment before continuing with the program. In the RUN mode, when STEP PRGM is pressed at either of these instructions, the "continue" signal from the peripheral equipment will cause the program to continue running.

To overcome the above situations:

1. Using the STEP PRGM key, step to (not through) the particular instruction.
2. Manually branch (using the GO TO) around the instruction (be sure to also branch around the instruction associated with a FMT).
3. Manually key the instructions that were skipped.

TURNING CALCULATOR ON LIGHTS ERROR LAMP

The error lamp being lit when the calculator is switched ON is a normal operation and should be of no concern; provided, the light can be reset by pressing any key on the calculator keyboard.

9100B

UNUSUAL Z REGISTER DISPLAY

Occasionally, when the PROGRAM - RUN switch is set to PROGRAM, an unusual display may appear in the Z register. This may consist of a random number between the address and the octal code; alternatively, an improper address or octal code, consisting of more than two digits, may appear. This display is the transfer vectors stored in the instrument when a subroutine is called. To clear it, switch to RUN and press the SUB/RETURN key; then, readdress the program counter to the required address and switch back to PROGRAM.

UNEXPECTED RESULTS OBTAINED FROM OPERATIONS FOLLOWING "IF" INSTRUCTIONS

The four conditional keys (IF FLAG, IF X < Y, IF X = Y, IF X > Y) are specifically intended to be followed by a branching address; therefore, in a few cases, unexpected results will be obtained if operational keys are substituted for the branching address. The following is a list of precautions that must be observed when using certain operational keys following "IF" keys.

Applicable to all four "IF" keys when the condition is met.

1. IF X = Y
   CHG SIGN

   Regardless of the steps preceding the "IF", changes the sign of the exponent of the number in X. The sign of the number remains unchanged.

2. IF X = Y
   ●
   1

   If the program steps are on the (+) page: .1 → X

   If the program steps are on the (−) page: −.1 → X

3. IF X = Y
   RCL

   The contents of (+) F are recalled to X, but the Y register remains unchanged; also, the program counter branches to a random address.

Pages cannot be crossed without care. Any of the instructions listed below, if contained in the step following the "IF", constitute the start of an address so that the next two steps must contain the remainder of the address:

SUB/RETURN

+

−

any alphameric

A CONTINUE cannot be used as a "no operation" in the step after an "IF" instruction, if it is followed by any one of the following:

SUB/RETURN

+

—

any alphameric

This is because the CONTINUE does not clear the "GO TO" condition set up when an "IF" instruction is met. For example, assume the following program steps are executed and the IF X = Y instruction is met:

STEP	KEY
2	IF X=Y
3	CONT
4	—
5	3
6	7

The program counter will branch to (—)3-7. Had the IF X = Y instruction been failed, the number 37 would have been entered in the X register.

Applicable only to the IF FLAG instruction when the flag is not set.

STEP	KEY
2	ENTER EXP
3	6
4	IF FLAG
5	3
6	4
7	any alphameric

The "not met" IF FLAG does not reset the ENTER EXP instruction; this results in the alphameric character being entered as a digit of the exponent. If RCL is used in place of the alphameric, then the contents of F are recalled to X but nonsense appears in the Y register.

UNEXPECTED RESULTS OBTAINED FROM A GO TO ( ) ( ) INSTRUCTION

STEP	KEY
7	GO TO ( ) ( )
8	CHG SIGN
9	2
10	3

Will send the program counter to (+)2-3 if the program counter is on the (—) page and to (—)2-3 if the program counter is on the (+) page.

9100B

KEYSTROKES DO NOT APPEAR IN X

A 9100A program cannot be stopped (with the STOP key) during a GO TO ( ) ( ) instruction; the 9100B can. To see how this can create a pseudo-hardware problem, consider the following example:

STEP	KEY
1	CLEAR
2	1
3	+
4	PAUSE
5	GO TO ( ) ( )
6	0
7	2
8	END

If the program is started and STOP is pressed to stop the program, the following two manual keyboard entries may not appear in the X register. This is because the STOP key will not work during PAUSE (where this program spends most of its time). The GO TO ( ) ( ) requires the second longest execution time and since the STOP will work during the GO TO ( ) ( ), the odds are good that the STOP key will take effect there. If that is the case, the next two digit entry keystrokes are used to fill the "GO TO" address. If the STOP takes effect during any of the other program steps, the following digit entry keys will be entered in the X register. To avoid this situation, PRESS: STOP, STOP.

SUBROUTINE CALLED FROM KEYBOARD WILL NOT RETURN CONTROL TO MAIN PROGRAM

If the program counter is addressed from the keyboard to the starting address of a subroutine and the CONTINUE key is pressed, then after the subroutine is completed, the program will not return to the correct address. This occurs because the transfer vector (return address) was never stored. Normally, a subroutine is called during program execution, which is when the transfer vector is stored; subroutines addressed from the keyboard have no way of returning to the main program.

PRESSING STEP PRGM CAUSES MULTIPLE KEY EXECUTION

Occasionally, when in the RUN mode, pressing STEP PRGM once will cause several keys to be executed.

When STEP PRGM is pressed at a FMT instruction both the FMT and the next instruction will be executed as one instruction. The next time STEP PRGM is pressed, the second instruction after the FMT will be executed.

The IF FLAG, $X < Y$, $X = Y$, $X > Y$ will cause the program counter to branch immediately if the condition is not met and execute the third instruction following the qualifier.