

HP RPN Evolves

Richard J. Nelson

Introduction

When HP introduced RPL as a user interface it utilized a new method of dealing with the automatic stack. RPL⁽¹⁾ operates very much like classical RPN in that it executed an operation such as + on the first two levels of the stack – X & Y registers in Classical RPN stack terms. In general terms this is called postfix logic.

If you press the digit 5 on an RPL calculator the “5” is entered into level 1 of the stack. The same thing happens with a classical RPN calculator. If, however, you then press ENTER the two user interfaces operate differently.

RPL will terminate the digit 5, and leave it on level 1 of the stack.

Classical RPN will terminate the digit 5, AND it will copy it to level 2 (the Classical RPN Y stack register) - implements a stack lift - AND it will leave the level 1 (the Classical RPN X register) ready to receive digits. If a second digit is pressed, say a 2, there will be a 5 on level 2 (Y register) and an unterminated 2 on level 1 (X register).

The Automatic Stack

There are two basic versions of RPN⁽²⁾. One is the classical RPN and the other is what I will call Entry RPN. The difference between the two “RPN” versions is making the extra copy when the ENTER key is pressed⁽³⁾.

In classical RPN terms the automatic stack is lifted when the ENTER key is pressed. Table 1 provides a specific perspective on how HP RPN machines have subtly changed in terms of how Classical RPN works. A primary requirement for defining Classical RPN is the automatic stack, and the rules that govern it. See [HP Solve](#) issue 4 page 3 for a detailed description. Here is a problem. Add -5 to 2.

Table 1 - RPN Stack Lift Test: 5, ENTER, CHS, 2, +

Case	Models (a few examples)	Result	Period	Comments
1	HP-35A	3	1/4/72	The first RPN calculator. One machine only.
2	HP-45A, 80, 65, 70, 55, 21, 22, 25, 25C, 27,	7	1973 - 1977	All 7 Classic, most, 5, Woodstock Series.
3	HP-29C, 91, 97, 19C, 41C, 34C, 15C, 35s, 20b, 30b	-3	Mid 1977 - 2012	Last “C” Woodstock, Top Cat Series, etc. to present.

Case 1 Stack diagram. ~ is don't care. ↑ is ENTER. The CHS is retained with a stack lift and a new digit entry.
Case 1 (2 stack levels) = +3.

Step	1	2	3	4	5
T	~	~	~	~	~
Z	~	~	~	~	~
Y	~	5	5	5	~
X	5	5	-5	-2	3
Press	5	↑	CHS	2	+

Fig. 1 – Case 1 HP-35A automatic stack operation.

This doesn't work because the user clearly did not enter a -2.

Case 2 Stack diagram. ~ is don't care. ↑ is ENTER. The CHS is not retained with a stack lift and new digit entry. **Case 2 (2 stack levels) = +7.**

Step	1	2	3	4	5
T	~	~	~	~	~
Z	~	~	~	~	~
Y	~	5	5	5	~
X	5	5	-5	2	7
Press	5	↑	CHS	2	+

Fig. 2 – Case 2 Classic & Woodstock automatic stack operation.

This doesn't work because the User wanted a -5.

Case 3 Stack diagram. ~ is don't care. ↑ is ENTER. The CHS terminates digit entry, and a new digit entry causes a stack lift. **Case 3 (3 stack levels) = -3.**

Step	1	2	3	4	5
T	~	~	~	~	~
Z	~	~	~	5	~
Y	~	5	5	-5	5
X	5	5	-5	2	-3
Press	5	↑	CHS	2	+

Fig. 3 –Automatic stack operation Mid-1977 to present.

The automatic stack logic illustrated by case 3 is by far the most commonly used in terms of the large number of HP calculators that use it. Note, however, that for this automatic stack operation sequence (that includes CHS) that three stack registers are used to add two numbers. If the sequence of operations were ordered differently only two stack levels are required. This, of course, is the proper way to solve the problem. In all of these variations the number of keystrokes is the same.

Step	1	2	3	4	5
T	~	~	~	~	~
Z	~	~	~	~	~
Y	~	~	-5	-5	~
X	5	-5	-5	2	-3
Press	5	CHS	↑	2	+

Fig. 4 –Alternate Fig. 3 keystrokes to only use two stack levels.

When RPL appeared with the introduction of the HP-28C in January of 1987, the “automatic stack” radically changed. One of the distinctive differences is the action of the ENTER key. In Classic RPN the ENTER key copies the X register into the Y register. In an RPL machine ENTER terminates the digit entry and does not make a copy (or provide stack lift). It appears that the RPL operating system logic has

impacted the design of recently developed calculators, which have a 4-level high stack, but which use Entry RPN.

An example of a current model Entry RPN machine is the HP 30b. If you place the machine in RPN mode, and have a clear stack, you may compare the stack operation with a Classical RPN machine such as the HP 15C. Press 5, ENTER. Next press x. On the 30b you will see 0 because the Y register is zero and $0 \times 5 = 0$. On the 15C you will see 25 because the ENTER, \uparrow , raised the stack as shown in the three cases above.

This basic difference between Entry RPN, and Classical RPN was described by Gene Wright in *HP Solve* RPN Tip 20 in Issue 20 page 10, (**I20,P10**). See the article list number 4 which is reproduced below. "I" is the Input key. Note the extra ENTER (I) required as mentioned above.

"4. The I command (=works just as well) does not copy X to Y if pressed while entering a value, but simply terminates digit entry, similar to the behavior of the HP graphing calculators. On the HP 12c, pressing 4 ENTER x shows a result of 16. To achieve this on the HP 30b, 4II*is required."

RPN Terms

Digit Entry is the process of keying digits into the calculator. The machine has to have some means to know when the number is complete. If you key 12 into an HP 35s, for example, and turn the machine off and back on again, the display will show 12 with an underscore to indicate that digit entry is not complete.

Digit Termination is the process of telling the calculator that digit entry is complete. The example given in digit entry (12_) may accomplish digit termination entry by pressing $x \leftarrow y$ twice. The stack is not disturbed. Pressing a function key will terminate digit entry. Specific details of all the situations (functions and operations) that terminate digit entry are usually provided in the User's Guide.

Stack Lift Disable is activated under a very few conditions of the operation of the automatic stack. See your user's guide for details (list). Stack lift is either enabled, disabled, or neutral.

Observations and Conclusions

There are two basic forms of HP RPN. Classical RPN and Entry RPN. While they are very similar overall in that they are both a postscript user interface there are subtle differences, a few of which are described. A test to determine which one is operational on your HP calculator is provided. With a clear stack press 5, ENTER, x. On an Entry RPN machine the result is zero. On a Classical RPN calculator the number is squared.

The current machine examples of an Entry RPN calculator are the HP 20b and HP 30b.

RPN calculators have subtle differences, and the Table 1 CHS digit entry example shows that HP RPN machines have changed twice since the first Classical handheld RPN calculator, the HP-35A. These subtle differences will often impact RPN programs that may run on one RPN machine and not on another. It is highly doubtful that HP will return to Classical RPN except, perhaps, for a legacy machine such as the HP-15C. In all cases of an RPN (and RPL) calculator the best way to describe the user interface is that it is a post script logic user interface rather than one with a rigid set of automatic stack rules.

HP RPN Evolves - Notes

- (1) *RPL is an acronym for Reverse Polish Lisp. LISP is an artificial intelligence, AI, high level computer programming language. A popular programmer meaning given to LISP is "Lots of Insidious and Silly Parentheses." The programming power of RPL is astounding. See Wikipedia description of LISP at: <http://en.wikipedia.org/wiki/Lisp>. RPL is not mentioned, however. The Wikipedia entry for RPL also gives RPL as meaning ROM-based procedural Language*

A quote from one of the 1984 RPL Developers at HP Corvallis Oregon, Dr. William Wickes, says RPL could stand for Reverse Polish Lisp. "In the early days of RPL development, we got tired of calling the unnamed system 'the new system,' and one of the development team came up with "RPL," both as a play on "RPN" which has been the loved/hated hallmark of HP calcs forever, and as an accurate indication of the derivation of the language from Forth and Lisp. RPL was never particularly intended to be a public term; at the time of the HP Journal article (August 1987) on the HP 28C there was an attempt to create a less whimsical name--hence 'ROM-based procedural language,' which preserved the initials but had a more dignified sound.

The development team never calls it anything but (the initials) RPL. You can choose either of the two full-word versions that you prefer. Or how about 'Rich People's Language?'" See: <http://www.fags.org/fags/hp/hp48-faq/part2/>

Dr. Wickes recently retired from HP and he has the distinction of having been a development manager for HP's smallest consumer product (calculators) and HP's largest commercial product (Halo).

- (2) *Classical RPN reigned during the 70's and 80's. In the 90's, however, HP (internally) decided that the advantages of RPN should be supported, but that it needed to be modernized. The PC and other handhelds used by students used an ENTER key that behaved differently, and it was no longer an HP exclusive. This ENTER key, and how it worked became so widespread that HP felt that RPN was better "defined" as being a post script user interface rather than a rigid set of rules of how the stack must operate. This change is what I am calling Entry RPN.*
- (3) *Converting RPL based programs to run on RPN machines and vice versa will have to take this into account. If an RPN program is to run on an RPL machine an extra ENTER may be required. RPL programs are not good candidates for RPN machines because there are too many program structural differences. Converting an RPN program to run on an RPL calculator is a bit easier. RPN programs may have a GOTO instruction that is meaningless in an RPL program.*