

HP Forum Archive 13

[[Return to Index](#) | [Top of Index](#)]

Speed

Message #1 Posted by [bill platt](#) on 1 July 2003, 1:33 p.m.

Hi all voyager and pioneer fans,

Just got my copy of a 15c. Just to start the fires, I thought I would make a post about speed.

1.st: My 15c is about 10-15% slower than my 11-C. There is variation from machine to machine (old posts talk about this).

Now, given a simple loop: (pioneer syntax)

```
LBL A
```

```
-1
```

```
+
```

```
x>0?
```

```
GTO A
```

```
1.E40
```

```
RTN
```

start by putting a number in the stack, say 50, and when done cycling, display wil say "1 E 40"

Difference in speed:

15c: 50 loops = 24.0 sec

32sii 200 loops = 5.0 sec

or about 19:1

(incidentally I used my 48gx as a stopwatch :-})

So, what is the difference in time between release of these products? 1981 for voyager, 1988 for pioneer, so 7 years.

Extrapolating linearly, that would mean that today, we should be able to build a calculator with a speed of 63 times that of the Voyager. I bet it is even faster (but we have to be fair on comparing at a similar price point).

Re: Speed

*Message #2 Posted by [Patrick](#) on 1 July 2003, 2:10 p.m.,
in response to message #1 by [bill platt](#)*

The famous Moore's law says that speed (and memory capacity) approximately doubles every 18 months, meaning there is exponential growth. Those of you with a calculator handy (!) can compute an equivalent of about a 58 3/4% growth rate, compounded continuously.

Using this rate, one would expect an approximate 25 fold speedup in speed in 7 years, very much in line with the factor of 19 in your experiment.

Extrapolating from 1981 to 2003, we might expect a speedup factor of 26,000! (No, that isn't factorial)

This is where reality sets in and we have to look at whether we are comparing apples to apples. The Voyager series was specifically designed for long battery life. The technology at the time might have supported making a faster processor at about the same cost, but the HP engineers might have sacrificed speed for extra battery life.

A famous scientist once said "Prediction is very difficult, especially when it is about the future."

Re: Speed

*Message #3 Posted by [bill platt](#) on 1 July 2003, 5:01 p.m.,
in response to message #2 by [Patrick](#)*

Patrick,

Thanks for your reply,

so, at the moore's law worked backwards, starting from a 500 MHz Palmtop, I work backwards to find a 19.2 kHz speed for the old 15C. Question is, "what was the processor speed of the old 15C?"

regards,

Bill

Re: Speed

Message #4 Posted by [David Ramsey](#) on 2 July 2003, 1:06 a.m.,
in response to message #2 by Patrick

Actually, Moore's law says the number of transistors on a chip doubles every 18 months. But it's been extended before...

Re: Speed

Message #5 Posted by [Trent Moseley](#) on 1 July 2003, 3:11 p.m.,
in response to message #1 by bill platt

Same results for a 1988 16C and a 1999 32SII. For the record a 1980 67 did 50 loops in 20.5 seconds.

tm

The Old Time Speedy: HP-25C

Message #6 Posted by [Trent Moseley](#) on 2 July 2003, 12:53 a.m.,
in response to message #5 by Trent Moseley

Fifty loops in 14 seconds! 1978 vintage. No LBL A of course.

tm

Misleading !

Message #7 Posted by [Valentin Albillo](#) on 2 July 2003, 5:29 a.m.,
in response to message #1 by bill platt

Hi Bill,

Bill posted:

"Just got my copy of a 15C"

First of all, congratulations and welcome to the club ! But let me point out that you didn't get a "copy" of a 15C, but a real, genuine *original*, right ? There are no *copies* of 15C's whatsoever, that is until and unless CFKAHP (Company Formerly Known As HP) releases an HP-15C Platinum, Iridium, Praseodimium, or something, though they'll probably release instead a "32SII Platinum" aka 33S.

Now, for the speed issue, this is a very thorny affair, because measuring speed isn't as trivial as setting up a loop and timing the seconds it takes to run. There are *serious* problems with that simple approach, namely:

- The coding of the loop can make *all* the difference, because a finely tuned loop can run many times faster than a sloppy approach. For instance, you can code a loop in the HP-15C like you did, with that `-1, +, x>0?` construct, but why not `1, -` instead of `-1, +`, which seems much more natural, or even using the powerful **ISG** or **DSE** looping instructions which are provided in the instruction set precisely for that very purpose, or use **GTO (i)** for rapid reverse branching, thus avoiding the label and the time-consuming label search, or ...
- related to the point above, the best way to code a loop in a particular machine (say a 32S) usually is *not* the way to code same loop in another one, and here the coder can't usually avoid to incur in *language chauvinism* or the constructs he/she is used to in another language and/or machine. Only a user absolutely knowledgeable about a number of different machines and who puts aside any preferences on language or coding techniques (say structured vs. goto) can actually produce optimal tests for a given machine or set of machines.
- finally, the very concept of testing the speed using a simple loop is as *misleading* a concept as one can be. There are machines that will run a 1000-cycle loop blindingly fast, yet will choke to death upon encountering a floating point sine computation, not to mention doing it properly (range, accuracy, etc).

The HP-15C is optimized for *mathematical computations*, not branching and looping, and at that it excels. Its instruction set includes functions to do almost all things mathematical with as little looping as possible. If you study my past HP-15C Quizzes/Challenges, you will see that the matrix operations, for instance, are so powerful that you can do *without loops* many computations that seemed to require them, and even when loops are needed, they can be kept to a bare minimum.

In other words, if you want to make a *fair* speed comparison and learn something about the relative speed of machines, avoid coming to premature conclusions by resorting to the use of an inadequate, quick & dirty test. I'd suggest the following instead:

Run a test which does significant computation, using a lot of mathematical functions [which is what a calculator such as the HP-15C is optimized to do], operators, storage, etc, and little branching. You should strive for a 90% computing, 10% branching or less. See what figures you get then. I'll bet you *won't* get a 19:1, right ?

If really trying to assess the speed of the HP-15C in particular, try to solve a 7x7 linear equations system in both your 15C and your 32S, and compare the times. See ?

Of course the 32S doesn't have matrix operations but so what ? That's not the 15C's fault, and we're testing speed in solving real-life math problems, aren't we ? ;-)

Matter of fact, I've done computations on my HP-15C, in 40+ steps, in a few seconds, that you wouldn't even dream of doing on a 32S, in 200+ steps, in many minutes [see my Datafile article next month], that is, if you could attempt them, which you can't because they won't fit. A comprehensive, well thought, awesomely powerful instruction set more than makes for lack of speed in other areas.

That said, welcome again and enjoy to the max your HP-15C. And I fully **expect** that you'll post a solution to my next HP-15C Quiz/Challenge, to be posted in a few days, you have that much time to learn all the capabilities of your wonderful machine, lest it becomes painfully obvious you simply don't deserve to own one. Do not try it with your 32S, though, it won't work :-)

Best regards.

Edited: 2 July 2003, 5:31 a.m.

Re: Misleading !

*Message #8 Posted by **R Lion (España)** on 2 July 2003, 11:55 a.m.,
in response to message #7 by Valentin Albillo*

Quote:

or use GTO (i) for rapid reverse branching

What is GTO (i)?

Thanks in advance

Raul

Rapid Reverse Branching

*Message #9 Posted by **Valentin Albillo** on 2 July 2003, 1:54 p.m.,
in response to message #8 by R Lion (España)*

Hi, Raul:

Actually, GTO (i) is the correct form for such machines as the HP-67, HP-34C, HP-11C, etc. For the HP-15C, the correct form is GTO I, with a capital "I" instead of "(i)".

GTO I, when register I does contain a negative integer (say -15), branches execution to that program step number (step 15 in this case), instead of searching for a label.

As no label search is performed whatsoever, it executes much faster, and besides it doesn't require a label at all at the destination step. It's a rarely seen technique, but available and powerful nevertheless.

Best regards.

Re: Rapid Reverse Branching

*Message #10 Posted by [gene](#) on 2 July 2003, 5:00 p.m.,
in response to message #9 by Valentin Albillo*

My guess about why this is a rarely seen technique (GTO I in particular) is because of the overhead required to store the step number into I, particularly when one wanted to change the destination in a program.

And, of course, the difficulty of debugging and maintaining a program using absolute addressing. Yuck! I know it RUNS faster, but the pain involved in getting there is nasty. :-)

Speed is Misleading

*Message #11 Posted by [bill platt](#) on 7 July 2003, 10:30 a.m.,
in response to message #7 by Valentin Albillo*

Valentin,

Thanks for your excellent response--it is just the sort of dialog I hoped to start!

Just now back from a short vacation---so looking forward to your next 15c quiz.

Regards,

Bill

Re: Speed -- a few more data points . . .

*Message #12 Posted by [Paul Brogger](#) on 2 July 2003, 11:21 a.m.,
in response to message #1 by bill platt*

HP-97 -- 50 loops in 21.5 seconds

HP-34c -- 50 loops in 21.9 seconds

HP-32s -- 200 loops in 4.0 seconds

HP-28C -- didn't care to figure out RPL (again)

HP-48G -- too lazy to figure out RPL

(I used my watch as a stopwatch . . .)

Lesson: RPL -- Use it or Lose it!

Re: Speed -- a few more data points . . .

*Message #13 Posted by [bill platt](#) on 7 July 2003, 11:46 a.m.,
in response to message #12 by Paul Brogger*

Paul,

looks good. Notice the correlation between your 97 and Trent's 67---this should be, as they are the same in their guts, yes?

Take a look at Valentin's post---he has great points about the validity of speed tests. I am going to play with some other loop structures and see what happens.

Regarding RPL---I am a newbie to RPL---but yes it seems a bit like "use ot or lose it." But I did successfully figure out how to do an IF-THEN-ELSE recently on RPN, so maybe I will be able to make a REPEAT-WHILE.....

regards,

Bill