



HP Forum Archive 18

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

HP 41 FORTH Manual

Message #1 Posted by [Egan Ford](#) on 8 Sept 2008, 3:28 a.m.

I'm looking for the HP41 FORTH Module Manual. Google, TOS, etc... has been of little help. Can any body help me?

Thanks.

Re: HP 41 FORTH Manual??

Message #2 Posted by [Walter B](#) on 8 Sept 2008, 5:06 a.m.,
in response to message #1 by Egan Ford

I know of a HP71 FORTH ROM module and manual being included in the Museum DVD, but didn't hear of something alike for the HP41 so far. Please check.

Edited: 8 Sept 2008, 5:09 a.m.

Re: HP 41 FORTH Manual??

Message #3 Posted by [Garth Wilson](#) on 8 Sept 2008, 5:43 a.m.,
in response to message #2 by Walter B

Yes there was one, and I thought I had some kind of manual here although I remember it was not very complete. I can't find it at the moment though.

As much as I like Forth, I have to wonder how well the 41 would do at it with its 56-bit registers, bit-serial bus, user RAM that is not wide enough to write your own primitives (8-bit when the 41 uses 10-bit for its machine language), a keyboard that's not particularly good for typing in much text source code, a very slow text editor, etc.. I'm curious though.

Edited: 8 Sept 2008, 5:45 a.m.

Re: HP 41 FORTH Manual??

Message #4 Posted by [Borja](#) on 8 Sept 2008, 9:41 a.m.,
in response to message #3 by Garth Wilson

Pick one and port ;)

The ancient fig-Forth listings are available

[online](#)

There was a time when you had to purchase a printed listing. Actually I have somewhere a 80% done fig-Forth port done to the Psion Organiser II (based on a Hitachi 6303 microcontroller, descendant of the Motorola 6800).

Re: HP 41 FORTH Manual??

Message #5 Posted by [Egan Ford](#) on 8 Sept 2008, 12:58 p.m.,
in response to message #3 by Garth Wilson

Quote:

As much as I like Forth, I have to wonder how well the 41 would do at it with its 56-bit registers, bit-serial bus, user RAM that is not wide enough to write your own primitives (8-bit when the 41 uses 10-bit for its machine language), a keyboard that's not particularly good for typing in much text source code, a very slow text editor, etc.. I'm curious though.

I am curious too, that is why I want to give it a go. A manual would help.

HP 41 FORTH, getting started.

Message #6 Posted by [Egan Ford](#) on 8 Sept 2008, 5:53 p.m.,
in response to message #3 by Garth Wilson

Garth, et al,

I spent a few minutes with Google and found no documentation and very little discussion of FORTH41. So, I thought I'd just try it. I didn't spend more that 15-20 minutes on this, but at least its a start.

The only emulators I was able to get FORTH41 to start up on were EMU41 and i41CXp (based on Nonpareil core). No luck with V41. I prefer EMU41 and i41CXp because they both have printer emulation.

If you want to play along, you can load up FORTH41 on i41CXp just by taking the default URL to load modules. For EMU41, I posted an example INI file and an EMU41 compatible FORTH41 ROM here: <http://sense.net/~egan/forth41>. Also posted at that URL is printer output of CAT 2 for the FORTH41 module and the overlay (very helpful).

The only test I did was to add two numbers together, here is how to do it:

1. XEQ FORTH, you should get an OK prompt. CLx/A to clear screen.
2. Do FORTH like any other FORTH interpreter, e.g. type without the quotes "1 2 + .", that will push 1 and 2 on the stack, add together, and the DOT will print the results. Alternately you can exit out of ALPHA and just type 1 R/S 2 R/S + R/S . and get the same results.
3. Press R/S to execute.
4. The screen should have "3 OK" on it.
5. Type "BYE" R/S to exit interpreter.

HP 41 FORTH, final thoughts

Message #7 Posted by [Egan Ford](#) on 11 Sept 2008, 1:45 a.m.,
in response to message #6 by Egan Ford

I finally had some time late tonight to explore the FORTH41 ROM a bit more. First of all I must state that I'm impressed with the effort. Kudos to Serge Vaudenay for this achievement.

What is FORTH41?

Information about this module is very sparse, the only proper documentation was written about 6 months ago and is difficult to obtain. This is what I was able to find:

1. "FORTH FOR THE HP-41", Datafile V3N45 Page 29. This article was published in 1984. I've posted a scan of this page here: http://sense.net/~egan/forth41/forth41_v3n45_p29.pdf.
2. Emmanuel Compes (the author of the aforementioned FORTH41 manual) has a bit more information here: <http://www.emmanuel.hp41.eu/database/forth41.html>.
3. According to sources in the previous two points, FORTH41 is based on the Forth-79 standard. You can obtain a copy of the standard here: <http://www.dartmouth.edu/~lmaurer/forth/Forth-79.pdf>.

Observations

1. While FORTH41 is based on Forth-79, it is not complete, however there are substitutes for what is missing. E.g. no 2/ or 2* for right and left shifts, no problem, just use 2 / and 2 *.
2. Performance is poor. I created two similar programs (one Forth, one RPN); given X, get the next prime number. I chose this because it is purely integer work (Forth does not have native FP support), and I had an ML version to compare with as well. **NOTE:** Neither program is well optimized, the purpose was to have the same method running under two different environments.
3. Programming Forth on the 41 is not as cumbersome as I thought it would have been.

Benchmark Results

To benchmark I set X to 1000 and generated the next 5 primes. The ML program was hands down the fastest. The RPN version was about 10x faster than the Forth version. BTW, Forth on the 71B is measurably faster than BASIC (about 3x).

Summary

Despite lesser performance and an incomplete implementation, it was still the most fun I've had with my 41 in a while.

If you are interested in learning Forth, the book, "Starting FORTH", is freely available from here: <http://home.iae.nl/users/mhx/sf.html>.

NOTE: All of this testing was done with i41CXp--the iPhone 41C emulator.

NOTE: I have V41 working now. FORTH41 does not play well with the CX (uses pages 4 and 5). I've been lucky with EMU41 and i41CXp. For V41 just use the CV ROM image with FORTH41.

Forth Code

```

DECIMAL
VARIABLE R
VARIABLE M
VARIABLE S

: ISQRT
M ! 0 R ! 16384 DUP S !
BEGIN 0> WHILE
  M @ S @ R @ OR < IF
    R @ 2 / R !

```

```

      ELSE
        S @ R @ OR NEGATE M +!
        R @ 2 / S @ OR R !
      THEN
        S @ 4 / DUP S !
      REPEAT
        R @
    ;

: NP
  1 + DUP
  2 MOD 0 = IF 1 + THEN
  BEGIN
    DUP ISQRT 1 + 3 DO
      DUP I MOD 0 = IF
        1 LEAVE THEN
    2 +LOOP
    DUP 1 = IF
      DROP 2 + 0
    ELSE DUP THEN
  UNTIL
;

: BM
  1000 NP NP NP NP NP
;

```

NOTE: If you think this looks a bit like RPL, it does. RPL was heavily influenced by Forth. Replace ! with STO and @ with RCL, quote your variables, and you are almost there.

RPN Code

```

01 LBL "NP"
02 1
03 +
04 STO 00
05 2
06 MOD
07 X#0?
08 GTO 01
09 1
10 ST+ 00
11 LBL 01
12 RCL 00
13 SQRT
14 INT
15 1
16 +
17 STO 01
18 3
19 STO 02
20 LBL 02
21 RCL 00
22 RCL 02
23 MOD
24 X=0?
25 GTO 03
26 2
27 ST+ 02
28 RCL 01
29 RCL 02
30 X<=Y?
31 GTO 02
32 RCL 00
33 RTN
34 LBL 03

```

```

35 2
36 ST+ 00
37 GTO 01
38 LBL "BM"
39 1000
40 XEQ "NP"
41 XEQ "NP"
42 XEQ "NP"
43 XEQ "NP"
44 XEQ "NP"
45 END

```

Edited: 11 Sept 2008, 12:02 p.m.

Re: HP 41 FORTH, final thoughts

*Message #8 Posted by [Garth Wilson](#) on 11 Sept 2008, 2:50 p.m.,
in response to message #7 by Egan Ford*

Quote:

1. "FORTH FOR THE HP-41", Datafile V3N45 Page 29. This article was published in 1984. I've posted a scan of this page here:
http://sense.net/~egan/forth41/forth41_v3n45_p29.pdf.
2. Emmanuel Compes (the author of the aforementioned FORTH41 manual) has a bit more information here:
<http://www.emmanuel.hp41.eu/database/forth41.html>.
3. According to sources in the previous two points, FORTH41 is based on the Forth-79 standard.

Thanks. I had written down a list of questions, and these answered many of them. It looks like a great work considering a 16-year-old did it, but I'm sure he could do far better now some 20 years later.

They list the primitives, but do you have a list of the secondaries? If it runs in main RAM (which is 8-bit), it wouldn't have any way for the user to write his own primitives (since machine language is 10-bit), but I hope it would at least have CREATE and DOES>

Although you say the 71 is typically three times as fast in Forth as in BASIC, I would comment that the 71's Forth is a very poor implementation, and I've been able to dramatically speed up many words by just redefining them in Forth, not even in assembly. The most extreme case I remember was CHR\$ which I made 14 times as fast. I suspect they must have just referenced some of the BASIC routines that were already there, and the interface between the two languages was very inefficient. I wonder if the 41 Forth was done the same way. There's no excuse for the 41 running 10 times as slow in Forth as in RPN. It should be 10 times as fast. I suppose the lack of the 2/ and 2* shifts probably accounts for quite a bit of that, since a divide or multiply takes much, much longer than a shift. Apparently it does not even have a SHIFT primitive. When FIG was still publishing Forth Dimensions, there were frequent discussion on how few primitives you could get by with. I think someone got it down to 12, but the performance was ridiculously slow-- which is not what Forth is all about.

The 71 Forth uses a 41-style 4-level stack for floating-point. Some Forth FP

implementations don't use a separate stack for it. When I first came to Forth, I viewed as a big limitation the fact that floating-point is discouraged. But as I've gained more experience, I've found I can do all the log and trig functions and so on very efficiently in scaled-integer.

Re: HP 41 FORTH, final thoughts

*Message #9 Posted by [Egan Ford](#) on 11 Sept 2008, 4:02 p.m.,
in response to message #8 by Garth Wilson*

Quote:

They list the primitives, but do you have a list of the secondaries?

Nope. But it would not be too hard to try them one-by-one to find them all.

Quote:

If it runs in main RAM (which is 8-bit), it wouldn't have any way for the user to write his own primitives (since machine language is 10-bit), but I hope it would at least have CREATE and DOES>

CREATE A 9 ALLOT worked for me. I didn't test DOES> since I have no idea what it *does*. I've only been playing with Forth for about a year, mostly to get better performance than BASIC for challenges posted here, and because I have a hard time with BASIC.

Quote:

Although you say the 71 is typically three times as fast in Forth as in BASIC, I would comment that the 71's Forth is a very poor implementation, and I've been able to dramatically speed up many words by just redefining them in Forth, not even in assembly. The most extreme case I remember was CHR\$ which I made 14 times as fast. I suspect they must have just referenced some of the BASIC routines that were already there, and the interface between the two languages was very inefficient. I wonder if the 41 Forth was done the same way. There's no excuse for the 41 running 10 times as slow in Forth as in RPN. It should be 10 times as fast.

I think you may be correct. I was listing random blocks and see a lot of RPN in there.

Quote:

I suppose the lack of the 2/ and 2* shifts probably accounts for quite a bit of that, since a divide or multiply takes much, much longer than a shift. Apparently it does not even have a SHIFT primitive. When FIG was still publishing Forth Dimensions, there were frequent discussion on how few primitives you could get by with. I think someone got it down to 12, but the performance was ridiculously slow-- which is not what Forth is all about.

My integer square root is very fast if you can shift. As it stands now it takes a

few seconds.

Quote:

The 71 Forth uses a 41-style 4-level stack for floating-point. Some Forth FP implementations don't use a separate stack for it. When I first came to Forth, I viewed as a big limitation the fact that floating-point is discouraged. But as I've gained more experience, I've found I can do all the log and trig functions and so on very efficiently in scaled-integer.

Something that I have not got around to yet. I've leveraged the 71 Forth FP stack in almost every challenge that I used Forth.

Forth is great fun, especially since I've been using RPL since 1993. I may be the only person with 41+Forth emulation on their iPhone that actually programs it. The emulator QWERTY support makes quick entry of Forth code with just my thumb.

I hope I can track down the manual, I'd like to get the editor, printer, data transfer from RPN to Forth, and file I/O working. Fortunately state is maintained, if I exit Forth and start it up again all my words are there. FEX works as well, just put your stack and words in ALPHA and run FEX. I think I can harvest ALPHA to get the results, but suddenly I lost all my Forth work. Unsure what I did wrong, I need to start over to continue exploring.

I've dumped all my research here: <http://sense.net/~egan/forth41> if you want to play.

Edited: 11 Sept 2008, 4:04 p.m.

Re: HP 41 FORTH, final thoughts

*Message #10 Posted by [Garth Wilson](#) on 11 Sept 2008, 6:06 p.m.,
in response to message #9 by Egan Ford*

Quote:

Nope. But it would not be too hard to try them one-by-one to find them all.

WORDS or VLIST should list them all at once so you don't have to try one by one.

Quote:

I didn't test DOES> since I have no idea what it does.

CREATE and DOES> allow you to make your own defining words, essentially extending the compiler. This is a very powerful feature of even very minimalist, tiny Forths. You can specify what action is taken when the child word is created, and then, following DOES> , specify what action is taken by the child word when it runs. ("Child" words are the words created by your new defining word.) This is the "Pearl of Forth" which lets you define for example new structure types. With STATE , you can have the

word do one thing when encountered during compilation and something else when encountered other times. If you want to define the child's run-time action in assembly instead of high-level Forth, you would use ;CODE instead of DOES> . Going along with that, COMPILE and [COMPILE] and IMMEDIATE also let you for example add new program structures such as CASE or any other one you may want to invent to meet your need, and it will become part of the language.

I don't think I've ever come across FEX . What is it?

Re: HP 41 FORTH, final thoughts

Message #11 Posted by [Egan Ford](#) on 11 Sept 2008, 8:16 p.m.,
in response to message #10 by Garth Wilson

Quote:

WORDS or VLIST should list them all at once so you don't have to try one by one.

Yes, they would if FORTH41 had those words. :-(

Quote:

I don't think I've ever come across FEX . What is it?

FEX is part of FORTH41, it allows you to send via ALPHA a string to Forth, e.g.:

```
ALPHA 1 2 . + ALPHA
XEQ FEX
```

Returns:

3

to the screen, any key will exit, nothing left in ALPHA.

Re: HP 41 FORTH, final thoughts

Message #12 Posted by [Garth Wilson](#) on 11 Sept 2008, 10:27 p.m.,
in response to message #11 by Egan Ford

so it's short for "Forth EXecute," similar to FORTHX on the 71 which allows you to execute Forth routines from the BASIC environment?

The 71 Forth manual has VLIST on page 32. It would have to be modified a bit to run on the 41, but it would be easy to write once you know how the dictionary headers are put together.

Re: HP 41 FORTH, final thoughts

Message #13 Posted by [Egan Ford](#) on 11 Sept 2008, 11:50 p.m.,
in response to message #12 by Garth Wilson

Quote:

so it's short for "Forth EXecute," similar to FORTHX on the 71 which allows you to execute Forth routines from the BASIC environment?

Yes.

Re: HP 41 FORTH Manual

*Message #14 Posted by **PeterP** on 8 Sept 2008, 12:05 p.m.,
in response to message #1 by Egan Ford*

emmanuel compes talks about the hp-41 forth module on his webpage.

<http://www.emmanuel.hp41.eu/database/forth41.html>

It seems to be a module that has been created by Sergey Vaudeney. Its image (and overlay) is available on TOS.

Emmanuel sold a HP41-Forth a while ago on eBay and included a manual that he wrote. (see also here: http://www.emmanuel.hp41.eu/ebay/ebay_uk.html)

Regular readers here will know about my personal 'luck' with Emmanuel so I don't know if this route is a way to actually get a manual. But it does exist...

Cheers

Peter

Re: HP 41 FORTH Manual

*Message #15 Posted by **Egan Ford** on 8 Sept 2008, 12:57 p.m.,
in response to message #14 by PeterP*

Quote:

Regular readers here will know about my personal 'luck' with Emmanuel so I don't know if this route is a way to actually get a manual. But it does exist...

Yeah, I know, so what happened? You can PM me. Thanks.

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



[Go back to the main exhibit hall](#)