

A04 4004 4040 cross-assembler

A04 4004 4040 cross-assembler

Last updated Aug 6 2020. Edited by Herb Johnson, (c) Herb Johnson, except for content written by others. [Contact Herb at www.retrotechnology.com](http://www.retrotechnology.com), an email address is on that page..

Introduction:

This Web page is about a 2014 adaptation of the A85 cross-assembler to support 4004 and 4040 assembly for various of my vintage computing projects. [I restore vintage computers of the 1970's](#) and [I support S-100 computers of the 70's and 80's](#) and some others including this [Intel MCS 4/40 computer](#). - Herb

[Here's the A04 cross assembler, supporting the Intel 4004 and 4040 mnemonics](#). There's example 4004 and 4040 C source files which are assembled, and there's an A04.DOC manual with details of the assembler. The executable is a 32-bit MS-DOS program, which can be run in any Windows "command prompt" MS-DOS "window". What does that mean? Read this Web page, then read my [A18 cross assembler Web page about Windows and MS-DOS](#).

And I really mean, "read the fine manuals". OK? - Herb Johnson

A04 history

I wanted a 4004 and 4040 crossassembler, under Windows, with sources I could modify to suit various situations. One cross assembler in C, is the A85, A68, A18 etc. series of cross assemblers by William Colley. I first revised his 1802 version, then his many others. [See my A18 Web page for details about all that](#). I was able to modify A85 to support the 4004, and then add 4040 mnemonics.

How and where to run A04

The program runs under Windows (XP, 2000) as an MS-DOS 32-bit executable, which runs under the "DOS box" command line. I provide the C sources with my changes clearly marked. There are docs and sample assembly code. For those who download it, please advise me of any errors and issues. I make zero guarantees, offer zero warranties. I am not responsible for any loss, injury or damage to person or property of any sort. Use entirely at your own risk.

To run under Linux: [The A18 1802 cross assembler Web page](#) has more information about compiling this and related cross-assemblers. Also, it points to other Web pages, where I have those 8-bit processor's cross assemblers from Colley I've adapted. **The A18 Web page, also has some software tools.**

future: With small changes to the source, it should compile under Borland's old (16-bit) MS-DOS based Turbo C. If I do that, I'll add the 16-bit MSDOS executable to the ZIP file.

A04 features, fixes

"Read the fine manual", A04.DOC, for details of features. Keep in mind that very old assembly languages like those original to the 4004, 4040, 8008, etc. used programming conventions not followed in later 8-bit assemblers. Your favorite assembler may vary from A04. Let me know if you miss some feature, or if there's some apparent error. However "RTFM" to see if there's another way. Meanwhile, here's some features that may be of interest. I'd like to thank **Dwight Elvey** for his correspondence with me in Oct 2019, for his attention to some of these issues and fixes, from his work with the 4004 and 4040. - Herb

Register-pair descriptions: Note: Intel's 4004 assembler requires even-values for register-pair descriptions in instructions like SRC, FIN, FIM. 0 2 4 6 8 10 12 14 refer to R0R1, R2R3, etc. Some other assemblers, required ODD values, and some required values 0 through 7. My A04 assembler has two options, check the "RPAIRS" pseudoop for "EVENS" or "ZERO7". You can use EQUated mnemonics for register-pairs as in the INCLUDE.ASM file. Let me know your experiences with other 4004 4040 assemblers or with hand-coding. - Herb

This version assumes and produces hexadecimal values in its listings of codes from the ASM file; changes in the C source would produce octal listings. You can create binary, hex, decimal or octal constants in your source, by adding a prefix or suffix to those constants: read the "A40.DOC" file for details. Most of the 4004 and 4040 code I've seen, assumes decimal values; much 8008 code and some 8080 code seems to be in octal. The default radix for the A04 assembler is in one line of C code.

Many early Intel assemblers, accepted **multiple assembly statements per line**. I've added the '!' character as a statement-seperator. So single lines like " DB 20 ! DB 25 ! DB 40" were assembled as three seperate statements. In A04, there must be a space after the '!'; the statements are seperate lines in the listing. It's clumsy, but it works I believe.

Most modern assemblers want operands to be seperated by commas. I tried to have A04 accept a space between operands, like "FIM 0 2220". I could not make that work. Use "FIM 0, 220".

The 4004 only permits branches within the same page (256 decimal bytes, FFH hex bytes) of code. Branch address are represented by a byte value. If you use a lable as a branch value, no problem. If you use a RELATIVE address (the lower 8 bits) as a branch value, A04 will flag it as an error; but generate the correct byte value. If you use a 16 bit address value, no error; again an in-page byte address will be generated. If this gets annoying, let me know and I'll see if I can fix it; or fix the C source yourself.

Similarly: the FIM instruction generates an 8-bit value from its data 2nd operand, without an error flag. If you use or generate a 12-bit or 16-bit value, or a label, the truncated lower 8-bit value is used and no error message is generated.

- Herb Johnson

Contact information:

*Herb Johnson
New Jersey, USA*

This page and edited content is copyright Herb Johnson (c) 2020. Copyright of other contents beyond brief quotes, is held by those authors. [Contact Herb at www.retrotechnology.com](http://www.retrotechnology.com), [an email address is available on that page.](#)