

DESIGNER'S NOTEBOOK



How to Convert a HEX30 Output File Into a Linkable Assembly File

Contributed by Gerald Capwell and Rosemarie Piedra

Design Problem

I used HEX30 to generate a 1-section output file in PROM programmer format. I want to link the data contained in that file with the rest of my application.

Solution

HEX30 takes a 'C3x/'C4x COFF file and converts it into a PROM programmer (i.e., Intel, Motorola, and ASCII formats) file. Please use the Intel or ASCII formats when trying to link this data with your application. The conversion process is below:

1. Generate a single HEX30 output file in Intel or ASCII formats. As an example, we use ASCII format (the HEX30 output file is called child.a0 in this description). After doing so, the output will be a PROM programmer file listing the data of the COFF file and other control characters.
2. Now you must use the Hex-to-Assembly utilities to convert the programmer file to an Assembly file. The HEX2ASM.EXE is a self-unarchiving executable containing two utilities called ASCI2ASM.EXE and INTL2ASM.EXE. These utilities extract the data from the PROM programmer file and create an .asm file which contains a .sect table listing each 32-bit word of code. For more information about the utilities, please read the documentation located in the HEX2ASM archive file. An example using ASCI2ASM.EXE is shown below.

Example: `ASCI2ASM child.a0 child.asm tablename`

Diagram illustrating the command line arguments for the example:

- `child.a0`: HEX30 child output file
- `child.asm`: name of the .asm file that the utility creates
- `tablename`: section name to be assigned to the boot table

The `child.asm` file that the utility creates contains the raw data extracted from the HEX30 output file as follows:

```
.sect "tablename";
.global _l1, _l2
_l1      .word   xxxx      ;first data word of child.a0 w/ label
         .word   yyyy
         .word   zzzz
         ...
_l2      .word   wwwwww    ;last data word of child.a0 w/ label
         .end
```

The ASCI2ASM utility also allows you to set labels at the beginning and/or at the end of the section (see `_l1` and `_l2`). The labels can be used externally and referred to by the code to which it is linked.

The 'C3x/C4x linker also offers what is called "linker variables" that can be used in your linker command file to create labels pointing to the beginning and/or the end of the section. Refer to assembler/linker Users Guide. If using the "linker variables" option, the linker command file should include the following text:

```
SECTIONS
{
    .child: {_l1 = .;
            *(.child)
            _l2 =.-1;}> RAM1
}
```

* Note: For conversion from Intel PROM format, the utility called "INTL2ASM.EXE" must be used (located in archive file called HEX2ASM.EXE)

3. Now you have a regular `.asm` file that you may assemble and link with your main program.

For further clarification, the following application notes illustrate the use of the HEX30 and HEX2ASM utilities:

- "Bootloading 'C4x Networks" (BBS filename = C4XNETB.EXE)
- "Exploring 'C4x Networks" (BBS filename = EXPLORE.EXE)
- "Hex-to-Assembly Conversion Utilities" (BBS filename = HEX2ASM.EXE)