TMS320 DSP
DESIGNER'S NOTEBOOK

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

APPLICATION BRIEF: SPRA236

Gerald Capwell and Rosemarie Piedra
Digital Signal Processing Products
Semiconductor Group

Texas Instruments
May 1994

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain application using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

TRADEMARKS

TI is a trademark of Texas Instruments Incorporated.

Other brands and names are the property of their respective owners.

CONTACT INFORMATION

US TMS320 HOTLINE	(281) 274-2320
US TMS320 FAX	(281) 274-2324
US TMS320 BBS	(281) 274-2323
US TMS320 email	dsph@ti.com

Contents

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

Abstract

The HEX30 utility takes a TMS320C3x/TMS320C4x COFF file and converts it into a PROM programmer (i.e. Intel, Motorola, and ASCII formats) file. This document describes how to link this output file to the rest of an assembly language application. It is suggested that using the Intel or ASCII formats when trying to link this data with your application will be the most effective path. The document includes brief code listings and command examples.

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 shown 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-extracting 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 that 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.

```
ASCI2ASM child.a0 child.asm tablename
where:   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 with label
      .word    yyyy
      .word    zzzz
       ...
_l2   .word    wwww  ;last data word
                     ;  of child.a0 with 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)