IGLOO2 HPMS

Embedded Nonvolatile Memory (eNVM) Configuration

Microsemi
Power Matters.™

Table of Contents

Introduction

The HPMS Embedded Nonvolatile Memory (eNVM) configurator enables you to create memory regions (clients) that need to be programmed in the IGLOO2 device eNVM block(s).

This document describes how to configure eNVM block(s). For more information about eNVM, refer to the Microsemi IGLOO2 User's Guide.

Important Information About eNVM Reserved Pages

A certain number of eNVM pages are reserved to store the HPMS configuration. These reserved pages are used by the HPMS to store the Certificate/Digest and the Peripheral Initialization Configuration Data for SERDES, FDDR and MDDR. These pages are located at the top of the eNVM address space. The total number of reserved pages in the eNVM is device-dependent, as shown in Table 1.

Table 1 • **Device Type and eNVM Reserved Pages**

IGLOO2 Device	Reserved Pages for Peripheral Initialization Configuration Data	Reserved Pages for Certificate/Digest	Total Reserved Pages
M2GL005, M2GL010, M2GL025,M2GL050	32	16	48
M2GL060,M2GL090, M2GL150	32	64	96

Note: You application should not write into these reserved pages, as it will most likely cause a runtime failure for your design.

The number of Available Pages displayed in the eNVM Configurator is the total number of pages available to you after the Reserved Pages have been taken into account. For example, the M2GL150S device data sheet shows a total of 4096 pages in the eNVM, but the eNVM Configurator (Figure 1-1) shows only 4000 Available Pages, because 96 pages have been reserved by the HPMS and made unavailable to the user.

1 – Creating Clients

The main page of the eNVM configurator enables you to add clients to your eNVM block (Figure 1-1).

- **Data Storage client** - Use the data storage client to define a generic memory region in the eNVM block. This region can be used to hold your application code or any other data content that your application may need.
- **Serialization Client** - A serialization client stores a value which is different for each device that is programmed.

The main grid also displays characteristics of any configured clients. These characteristics are:

- **Client Type** - Type of the client (Data Storage or Serialization) that is added to the system
- **Client Name** - Name of the client. Must be unique across the system.
- **Start Address** - Address in hex at which the client is located in eNVM. It must be on a page boundary. No overlapping addresses between different clients are allowed.
- **Depth x Width of Word** - Word size of the client in Depth x Width
- **Page Start** - Page on which the start address begins.
- **Page End** - Page on which the client memory region ends. It is automatically computed based on the start address, word size, and number of words for a client.
- **Initialization Order** - This field is not used by the IGLOO2 eNVM configurator.
- **Lock Start Address** - Specify this option if you do not want the eNVM configurator to change your start address when clicking the Optimize button. This applies to Data Storage clients only.

Usage statistics for the eNVM are also reported:

- **Available Pages** - Total number of pages available to create clients. The number of available pages may vary based on the selected die. This is the total number of pages available to you after the reserved pages have been taken into account.
- **Used Pages** - Total number of pages used by the configured clients.
- **Free Pages** - Total number of pages still available for configuring data storage and initialization clients.

Use the **Optimize** feature to resolve the conflicts on overlapping base addresses for clients. This operation does not modify the base addresses for any clients that have Lock Start Address checked.

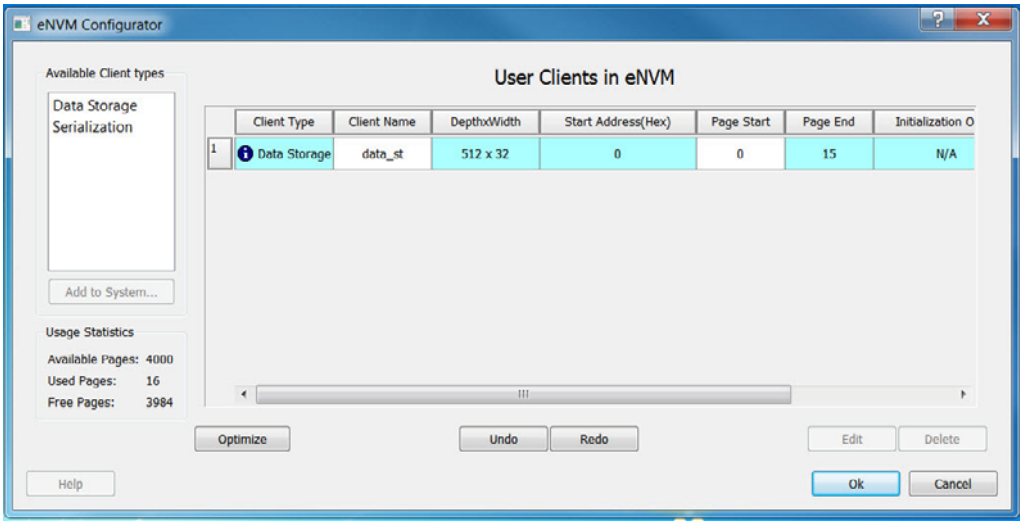

***Figure 1-1* • eNVM Configurator**

2 – Configuring a Data Storage Client

To add a Data Storage client, open the eNVM Configurator. For available client types, select Data Storage and click **Add to System**.

The Add Data Storage Client dialog box appears.

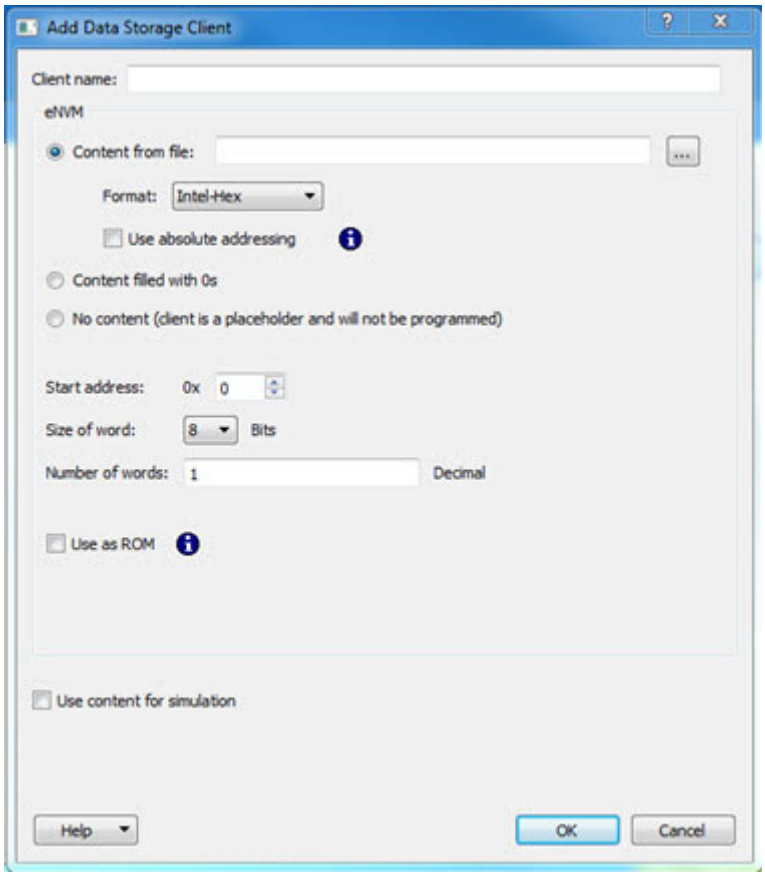

***Figure 2-1* • Adda Data Storage Client**

In the Add Data Storage Client dialog box, you must specify the following fields/options in the Client Configuration dialog box.

Client name – Enter a name for the Data Storage. Each client name must be unique across the system.

Content from file – Specify the memory file you want to program into eNVM. Click the Browse button. The Import Memory File dialog box appears.

Import Memory File

The Import Memory File dialog box allows you to navigate to a disk location and import a Memory File for a Data Storage Client or a serialization client.

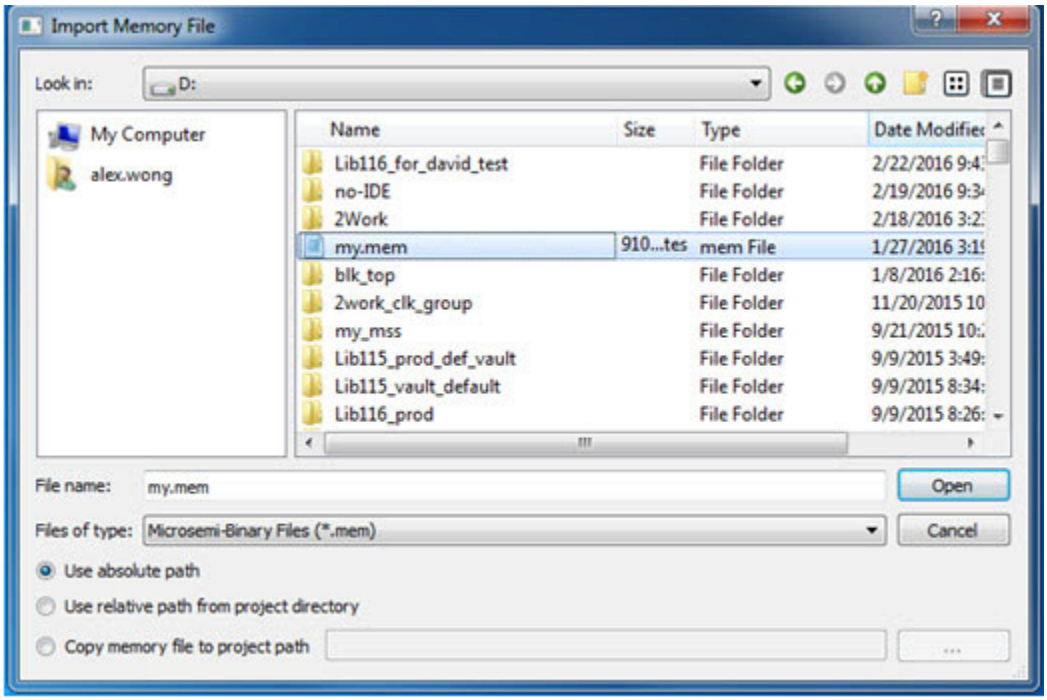

Figure 2-2 **• Import Memory File Dialog Box**

Use absolute path – When this radio button is checked, the Absolute Path of the Memory File appears in the Content from File field.

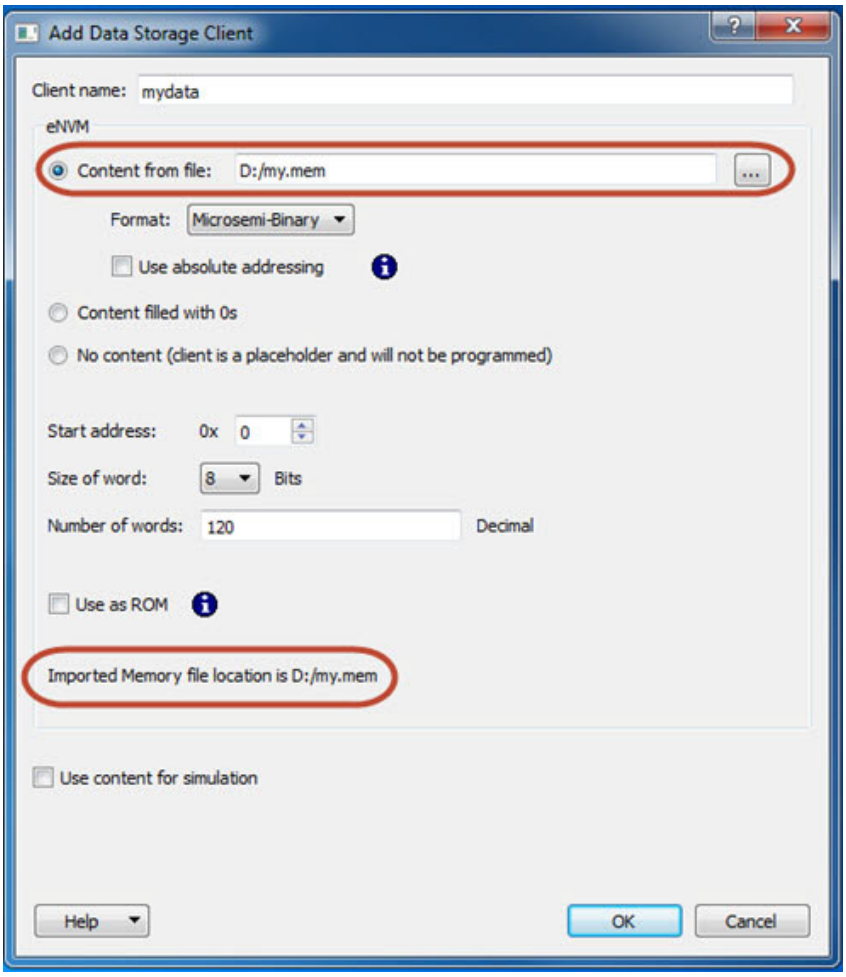

***Figure 2-3* • Absolute Path of Memory File**

Use relative path from project directory – When this radio button is checked, the Relative Path of the Memory File (relative to the Project location) you import appears in the Content from File field.

Note: On a Windows system, if the memory file and the Project location are on different drives, the Absolute Path is used even when you select Relative Path.

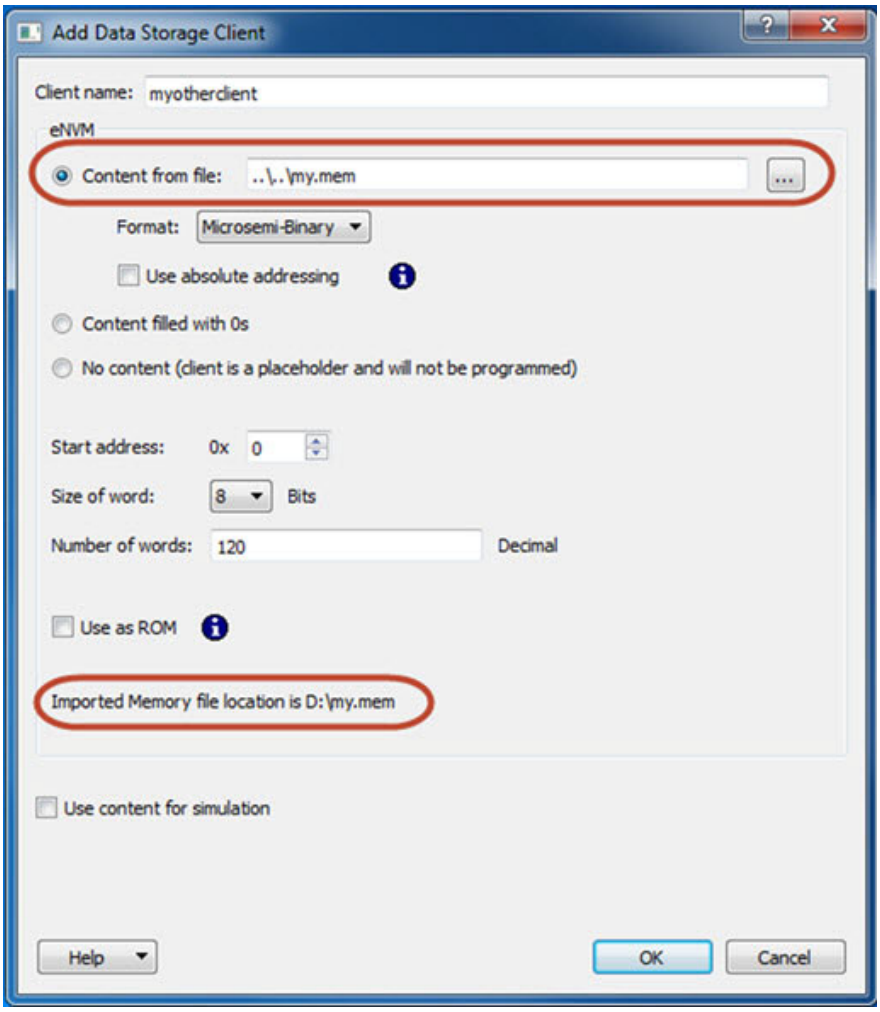

***Figure 2-4* • Relative Path of Memory File**

Copy memory file to project path – Click this radio button and click the Browse button at the far right to navigate to the location of the memory file to copy from.

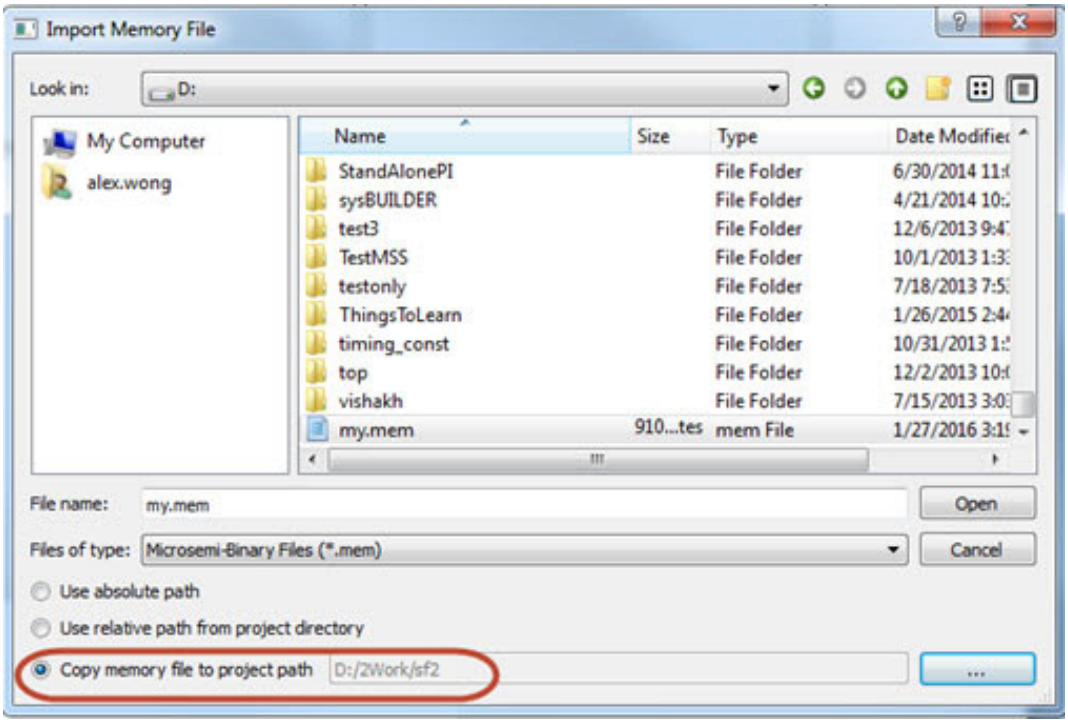

Figure 2-5 • **Copy Memory File to Project Path**

Notes

The memory file cannot be copied to and stored in the project's subfolders: component, smartgen, synthesis, designer, simulation, stimulus, tooldata, and constraint. To prevent users from inadvertently copying the memory file into these subfolders, these project subfolders are hidden from view when you select the project folder.

The copied Memory File path is internally stored as relative path.

If the Memory File is copied to the project, updating the content of the Memory File is the user's responsibility.

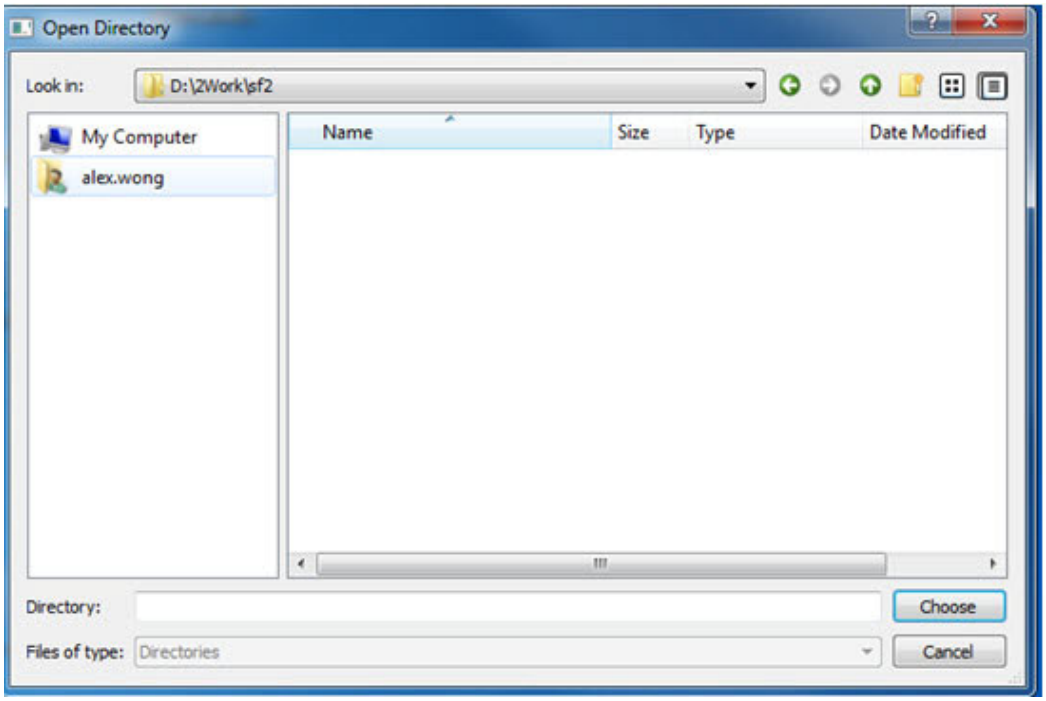

***Figure 2-6* • Project Subfolders Hidden from View**

Choose one of the following options for the Data Storage client:

Format – Select a file on disk that matches one of the following memory file formats—Intel-Hex, Motorola-S, Microsemi-Hex, or Microsemi-Binary

Use absolute addressing – Lets the memory content file dictate where the client is placed in the eNVM block. The addressing in the memory content file for the client becomes absolute to the whole eNVM block. Once you choose the absolute addressing option, the software extracts the smallest address from the memory content file and uses that address as the start address for the client.

Content filled with 0's – Client content is all 0's.

No content – The client is a place holder. You can load a memory file using FlashPro/FlashPoint at programming time without having to go back to this configurator.

Start Address – The eNVM address where the content is programmed.

Size of Word – Word size, in bits, of the initialized client; can be either 8, 16 or 32.

Number of words – Number of words (in Decimal) of the client.

Use as ROM – When checked (ON), all masters have read-only access to this eNVM client. The client content is protected and its content cannnot be overwritten. When unchecked (OFF), access to this eNVM client is controlled by settings in the MSS security policies configurator.

Use Content for Simulation – When checked, the client content is passed to the Simulator.

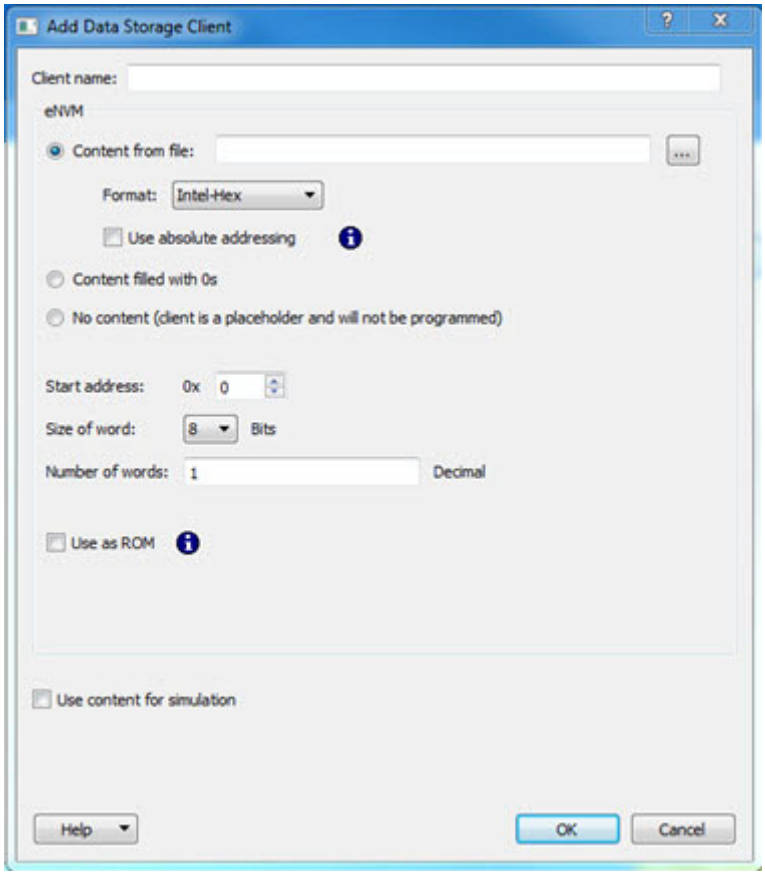

Figure 2-7 • **Add Data Storage Client Dialog Box**

3 – Memory File Formats

The following memory file formats are available as input files into the eNVM Configurator:

- INTEL-HEX
- MOTOROLA S-record
- Actel BINARY
- ACTEL-HEX

An example of how to interpret the memory content is shown below.

INTEL-HEX

Industry standard file. Extensions are HEX and IHX. For example, file2.hex or file3.ihx.

A standard format created by Intel. Memory contents are stored in ASCII files using hexadecimal characters. Each file contains a series of records (lines of text) delimited by new line, '\n', characters and each record starts with a ':' character. For more information on this format, refer to the Intel-Hex Record Format Specification document available on the web (search Intel Hexadecimal Object File for several examples).

The Intel Hex Record is composed of five fields and arranged as follows:

```
:llaaaatt[dd...]cc
```

Where:

- : is the start code of every Intel Hex record
- ll is the byte count of the data field
- aaaa is the 16-bit address of the beginning of the memory position for the data. Address is big endian.
- tt is record type, defines the data field:
  - 00 data record
  - 01 end of file record
  - 02 extended segment address record
  - 03 start segment address record (ignored by Microsemi SoC tools)
  - 04 extended linear address record
  - 05 start linear address record (ignored by Microsemi SoC tools)
- [dd...] is a sequence of n bytes of the data; n is equivalent to what was specified in the ll field
- cc is a checksum of count, address, and data

Example Intel Hex Record:

```
:10000000112233445566778899FFFA
```

Where 11 is the LSB and FF is the MSB.

MOTOROLA S-record

Industry standard file. File extension is s, such as file4.s

This format uses ASCII files, hex characters, and records to specify memory content in much the same way that Intel-Hex does. Refer to the Motorola S-record description document for more information on this format (search Motorola S-record description for several examples). The RAM Content Manager uses only the S1 through S3 record types; the others are ignored.

The major difference between Intel-Hex and Motorola S-record is the record formats, and some extra error checking features that are incorporated into Motorola S.

In both formats, memory content is specified by providing a starting address and a data set. The upper bits of the data set are loaded into the starting address and leftovers overflow into the adjacent addresses until the entire data set has been used.

The Motorola S-record is composed of 6 fields and arranged as follows:

```
Stllaaaa[dd...]cc
```

Where:

- S is the start code of every Motorola S-record
- t is record type, defines the data field
- ll is the byte count of the data field
- aaaa is a 16-bit address of the beginning of the memory position for the data. Address is big endian.
- [dd...] is a sequence of n bytes of the data; n is equivalent to what was specified in the ll field
- cc is the checksum of count, address and data

Example Motorola S-Record:

```
S10a0000112233445566778899FFFA
```

Where 11 is the LSB and FF is the MSB.

Actel BINARY

The simplest memory format. Each memfile contains as many rows as there are words. Each row is one word, where the number of binary digits equals the word size in bits. This format has a very strict syntax. The word size and number of rows must match exactly. The file extension is MEM; for example, file1.mem.

Example: Depth 6, Width is 8

```
01010011
11111111
01010101
11100010
10101010
11110000
```

Actel-HEX

A simple address/data pair format. All the addresses that have content are specified. Addresses with no content specified will be initialized to zeroes. The file extension is AHX, such as filex.ahx. The format is:

```
AA:D0D1D2
```

Where AA is the address location in hex. D0 is the MSB and D2 is the LSB.

```
The data size must match the word size. Example: Depth 6, Width is 8
00:FF
01:AB
02:CD
03:EF
04:12
05:BB
```

All other addresses will be zeroes.

4 – Interpreting Memory Content

Absolute vs. Relative Addressing

In relative addressing, the addresses in the memory content file did not determine where the client was placed in memory. You specify the location of the client by entering the start address. This becomes the 0 address from the memory content file perspective and the client is populated accordingly.

For example, if we place a client at 0x80 and the content of the memory file is as follows:

```
Address: 0x0000 data: 0102030405060708
Address: 0x0008 data: 090A0B0C0D0E0F10
```

Then the first set of bytes of this data is written to address 0x80 + 0000 in the eNVM block. The second set of bytes is written to address 0x80 + 0008 = 0x88, and so on.

Thus the addresses in the memory content file are relative to the client itself. Where the client is placed in memory is secondary.

In absolute addressing, the memory content file dictates where the client is placed in the eNVM block. So the addressing in the memory content file for the client becomes absolute to the whole eNVM block. Once you enable the absolute addressing option, the software extracts the smallest address from the memory content file and uses that address as the start address for the client.

Data Interpretation Example

The following examples show how the data is interpreted for various word sizes:

For the given data:

```
FF 11 EE 22 DD 33 CC 44 BB 55
```

(where 55 is the MSB and FF is the LSB)

For 32-bit word size:

```
0x22EE11FF (address 0)
0x44CC33DD (address 1)
0x000055BB (address 2)
```

For 16-bit word size:

```
0x11FF (address 0)
0x22EE (address 1)
0x33DD (address 2)
0x44CC (address 3)
0x55BB (address 4)
```

For 8-bit word size:

```
0xFF (address 0)
0x11 (address 1)
0xEE (address 2)
0x22 (address 3)
0xDD (address 4)
0x33 (address 5)
0xCC (address 6)
0x44 (address 7)
0xBB (address 8)
0x55 (address 9)
```

A – Product Support

Microsemi SoC Products Group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, electronic mail, and worldwide sales offices. This appendix contains information about contacting Microsemi SoC Products Group and using these support services.

Customer Service

Contact Customer Service for non-technical product support, such as product pricing, product upgrades, update information, order status, and authorization.

From North America, call **800.262.1060**
From the rest of the world, call **650.318.4460**
Fax, from anywhere in the world, **650.318.8044**

Customer Technical Support Center

Microsemi SoC Products Group staffs its Customer Technical Support Center with highly skilled engineers who can help answer your hardware, software, and design questions about Microsemi SoC Products. The Customer Technical Support Center spends a great deal of time creating application notes, answers to common design cycle questions, documentation of known issues, and various FAQs. So, before you contact us, please visit our online resources. It is very likely we have already answered your questions.

Technical Support

For Microsemi SoC Products Support, visit http://www.microsemi.com/products/fpga-soc/design-support/fpga-soc-support.

Website

You can browse a variety of technical and non-technical information on the Microsemi SoC Products Group home page, at www.microsemi.com/soc.

Contacting the Customer Technical Support Center

Highly skilled engineers staff the Technical Support Center. The Technical Support Center can be contacted by email or through the Microsemi SoC Products Group website.

Email

You can communicate your technical questions to our email address and receive answers back by email, fax, or phone. Also, if you have design problems, you can email your design files to receive assistance. We constantly monitor the email account throughout the day. When sending your request to us, please be sure to include your full name, company name, and your contact information for efficient processing of your request.

The technical support email address is soc_tech@microsemi.com.

My Cases

Microsemi SoC Products Group customers may submit and track technical cases online by going to My Cases.

Outside the U.S.

Customers needing assistance outside the US time zones can either contact technical support via email (soc_tech@microsemi.com) or contact a local sales office.

Visit About Us for sales office listings and corporate contacts.

Sales office listings can be found at www.microsemi.com/soc/company/contact/default.aspx.

ITAR Technical Support

For technical support on RH and RT FPGAs that are regulated by International Traffic in Arms Regulations (ITAR), contact us via soc_tech_itar@microsemi.com. Alternatively, within My Cases, select **Yes** in the ITAR drop-down list. For a complete list of ITAR-regulated Microsemi FPGAs, visit the ITAR web page.

Microsemi Corporate Headquarters
One Enterprise, Aliso Viejo,
CA 92656 USA

Within the USA: +1 (800) 713-4113
Outside the USA: +1 (949) 380-6100
Sales: +1 (949) 380-6136
Fax: +1 (949) 215-4996

E-mail: sales.support@microsemi.com

About Microsemi

Microsemi Corporation (Nasdaq: MSCC) offers a comprehensive portfolio of semiconductor and system solutions for communications, defense & security, aerospace and industrial markets. Products include high-performance and radiation-hardened analog mixed-signal integrated circuits, FPGAs, SoCs and ASICs; power management products; timing and synchronization devices and precise time solutions, setting the world's standard for time; voice processing devices; RF solutions; discrete components; Enterprise Storage and Communication solutions, security technologies and scalable anti-tamper products; Ethernet solutions; Power-over-Ethernet ICs and midspans; as well as custom design capabilities and services. Microsemi is headquartered in Aliso Viejo, Calif. and has approximately 4,800 employees globally. Learn more at **www.microsemi.com**.

5-02-00482-3/02.17