SmartFusion 2 Two-Port Large SRAM Configuration User Guide

Introduction (Ask a Question)

A Two-Port Large SRAM enables write access on one port and read access on the other port as shown in the following figure.

The core configurator automatically cascades Large SRAM blocks to create wider and deeper memories by choosing the most efficient aspect ratio. It also handles the grounding of unused bits. The core configurator supports the generation of memories that have different Read and Write aspect ratios.

Two-Port Large SRAM is synchronous for read and write operations, setting up the addresses as well as writing and reading the data. The memory write and read operations are triggered at the rising edge of the clock.

An optional pipeline register is available at the read data port to improve the clock-to-out delay.

In this document, we describe how you can configure a Two-Port Large SRAM instance and define how the signals are connected. For more information about the Two-Port Large SRAM, see UG0445: SmartFusion2 SoC and IGLOO2 Fabric User Guide.

Figure 1. Two-Port Large SRAM Configurator

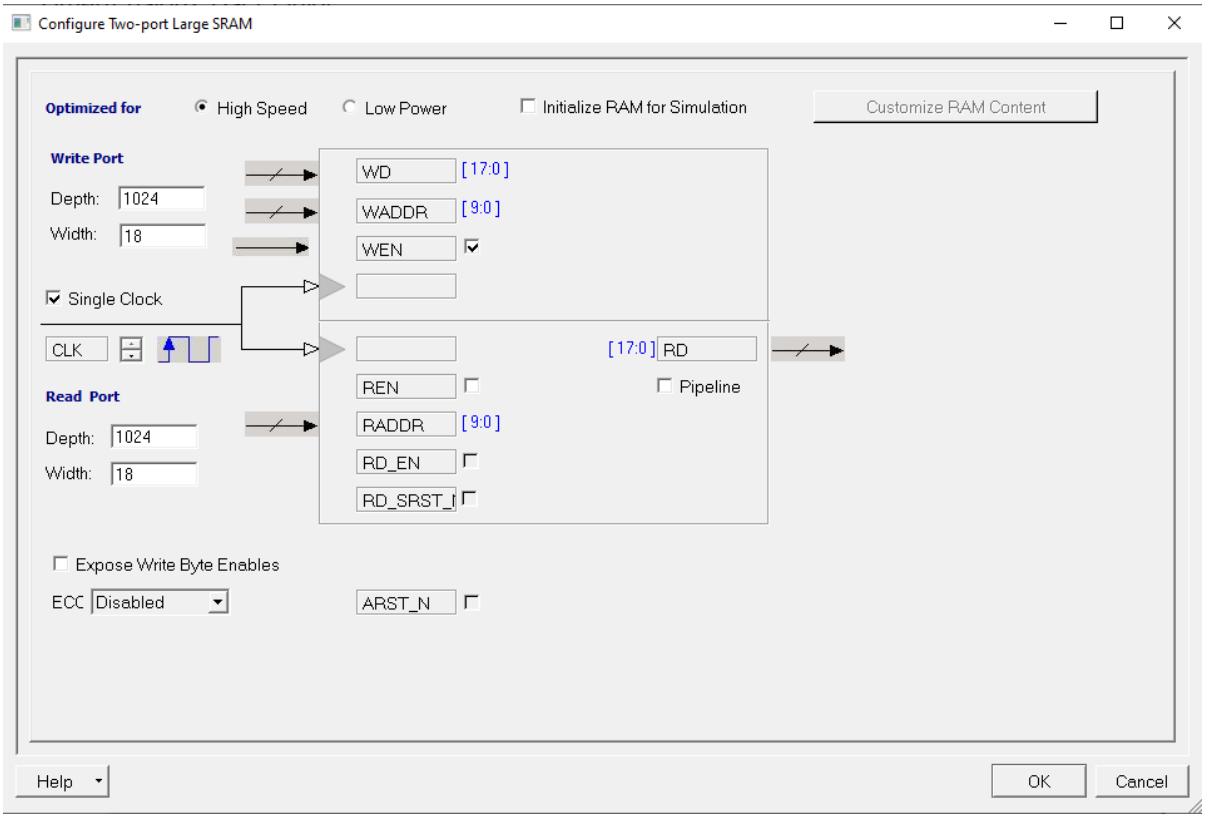


Table of Contents

1. Functionality (Ask a Question)

This section describes all the functionalities of the Two-Port LSRAM configurator.

1.1 Optimization for High Speed or Low Power (Ask a Question)

Selecting High Speed results in a macro optimized for speed and area (width cascading).

Selecting Low Power results in a macro optimized for low power, but uses additional logic at the input and output (depth cascading). Performance for a low power optimized macro may be inferior to that of a macro optimized for speed.

1.2 Write Depth/Width and Read Depth/Width (Ask a Question)

The depth range for each port is 1-524288. The width range for each port is 1-2484.

The two ports can be independently configured for any depth and width. (Write Depth × Write Width) must equal (Read Depth × Read Width)

1.3 Single Clock (CLK) or Independent Write and Read Clocks (WCLK and RCLK) (Ask a Question)

The default configuration for Two-Port Large SRAM is a Single clock (CLK) to drive WCLK and RCLK with the same clock. Uncheck the Single clock checkbox to drive independent clocks (one each for Write and Read).

Click the waveform next to any of the clock signals to toggle its active edge.

1.4 Write Enable (WEN) (Ask a Question)

Asserting WEN writes the data WD into the RAM at the address WADDR on the next rising edge of WCLK. Un-checking the WEN option ties the signal to the active state and removes it from the generated macro; click the signal arrow to toggle its polarity.

1.5 Read Enable (REN) (Ask a Question)

De-asserting REN forces the Read data (RD) to zero.

Asserting the REN reads the RAM at the read address RADDR onto the input of the RD register on the next rising edge of RCLK.

The default configuration for REN is unchecked, which ties the signal to the active state and removes it from the generated macro. Click the checkbox to insert that signal on the generated macro; click the signal arrow to toggle its polarity.

1.6 Pipeline for Read Data Output (Ask a Question)

Click the Pipeline checkbox to enable pipelining for RD. This is a static selection and cannot be changed dynamically by driving it with a signal.

Turning off pipelining of RD also disables the configuration options of the RD_EN, RD_SRST_N, and ARST_N signals.

1.7 Register Enable (RD_EN) (Ask a Question)

The pipeline register for RD has an active high, enable input. The default configuration is to tie this signal to the active state and remove it from the generated macro. Click the signal's checkbox to insert that signal on the generated macro; click the signal arrow to toggle its polarity.

1.8 Synchronous Reset (RD_SRST_N) (Ask a Question)

The pipeline register for RD has an active low, synchronous reset input. The default configuration is to tie this signal to the inactive state and remove it from the generated macro. Click the signal's checkbox to insert that signal on the generated macro; click the signal arrow to toggle its polarity.

1.9 Asynchronous Reset (ARST_N) (Ask a Question)

The pipeline register for RD has an active low, asynchronous reset input. The default configuration is to tie this signal to the inactive state and remove it from the generated macro. Click the signal's checkbox to insert that signal on the generated macro; click the signal arrow to toggle its polarity.

1.10 RD Register Truth Table (Ask a Question)

The following table describes the functionality of the control signals on the RD register.

Table 1-1. RD Register Truth Table

ARST_N	Pipeline	RCLK	RD_EN	RD_SRST_N	d	q
0	x	x	x	x	x	0
1	T	Not rising	x	x	x	q
1	T	Rising	0	x	x	q
1	T	Rising	1	0	x	0
1	T	Rising	1	1	x	d
1	F	x	0	x	x	q
1	F	x	1	0	x	0
1	F	x	1	1	x	d

2. Implementation Rules (Ask a Question)

This section describes the implementation rules.

2.1 Caveats for Two-Port Large SRAM Generation (Ask a Question)

- If you use a word width of 9, 18, or 36 for one port, then the width of the other port cannot be 1, 2, or 4. Configurations that do not use the 9th bit (for example, a Read width of 8192 × 4 and a Write width of 1024 × 32) are supported.
- The core configurator only supports depth cascading up to 32 blocks.
- The core configurator does not generate RAM based on a specific device. Refer to the datasheet to check for the available RAM1K × 18 modules in the device.
- The software returns a configuration error for unsupported configurations.

Important:

- All unused inputs must be grounded.
- ARST_N does not reset the memory contents. It resets only the RD.
- Writing to and reading from the same address is undefined and should be avoided. There is no collision prevention or detection.

3. RAM Content Manager (Ask a Question)

The RAM Content Manager enables you to specify the contents of your memory so that you can avoid the simulation cycles required for initializing the memory, which reduces simulation runtime.

The RAM core generator takes away much of the complexity required in the generation of large memory that utilizes one or more RAM blocks on the device. The configurator uses one or more memory blocks to generate a RAM matching your configuration. In addition, it also creates the surrounding cascading logic.

- The configurator cascades RAM blocks in three different ways.
- Cascaded deep (for example, 2 blocks of 16384 × 1 to create a 32768 × 1)
- Cascaded wide (for example, 2 blocks of 16384 × 1 to create a 16384 × 2)
- Cascaded wide and deep (for example, 4 blocks of 16384 × 1 to create a 32768 × 2, in 2 blocks width-wise by 2 blocks depth-wise configuration)

Specify memory content in terms of your total memory size. The configurator must partition your memory file appropriately such that the right content goes to the right block RAM when multiple blocks are cascaded.

3.1 Supported Formats (Ask a Question)

Intel-Hex and Motorola-S file formats are supported. The Microchip implementation of these formats interprets data sets in bytes. This means that if the memory width is 7 bits, every 8th bit in the data set is ignored. Or, if the data width is 9, two bytes are assigned to each memory address, and the upper 7-bits of each 2-byte pair are ignored.

The following examples illustrate 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)
```

- For 9-bit word size:

```
0x11FF -> 0x01FF (address 0)
0x22EE -> 0x00EE (address 1)
0x33DD -> 0x01DD (address 2)
0x44CC -> 0x00CC (address 3)
0x55BB -> 0x01BB (address 4)
```

Notice that for 9-bit, the upper 7-bits of the 2-bytes are ignored.

3.1.1 INTEL-HEX (Ask a Question)

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 a new line, '\n', characters and each record starts with a ':' character. For more information regarding this format, see 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 Microchip SoC tools)
  - 04 extended linear address record
  - 05 start linear address record (ignored by Microchip 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:

```
:0300300002337A1E
```

3.1.2 MOTOROLA S-record (Ask a Question)

Industry standard file. The 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 the 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
```

3.1.3 Write Port Width Alignment (Ask a Question)

The Microchip implementation of these formats interprets data sets in bytes. The implementation used is the same for all memory formats. The following examples show how data in a memory file is interpreted for different write port widths of memory.

The following figure shows data in Intel HEX memory file format. The same data is assumed to be used to initialize RAM in all examples.

Figure 3-1. Memory File Data—Intel HEX Memory File Format

```
:04000000FF11EE22DC
:04000400DD33CC44D8
:04000800BB550012D2
```

The Hex data from the memory file is converted into binary and read by the tool as a stream of bits. Based on the memory port width, the *required number of bits* for each address location in RAM is taken from the stream of bits.

Important: The *required number of bits* taken from the stream of bits for each address location is always a multiple of 8 (byte).

3.1.3.1 Write Port Widths Aligned on Byte Boundary (Ask a Question)

The following examples show how data from a memory file is stored in RAM when memory port widths are a multiple of a byte (aligned on a byte boundary).

3.1.3.1.1 32-Bit Write Port Width (Ask a Question)

When the memory write port width is 32-bits, which is aligned on a byte boundary, the tool uses 32-bits from the binary stream for each 32-bit word in the RAM. The resulting data stored in RAM is shown in the following figure.

Figure 3-2. Data in RAM—32-Bit Write Port Width Aligned on Byte Boundary

Memory File Data
:04000000FF11EE22DC
:04000400DD33CC44D8
:04000800BB550012D2

Address	Data Stored in RAM
0	0x22EE11FF
1	0x44CC33DD
2	0x120055BB

3.1.3.1.2 16-bit Write Port Width (Ask a Question)

When the memory write port width is 16-bits, which is aligned on a byte boundary, the tool uses 16-bits from the binary stream for each 16-bit word in the RAM. The resulting data stored in RAM is shown in the following figure.

Figure 3-3. Data in RAM - 16-bit Write Port Width Aligned on Byte Boundary

Memory File Data
:04000000FF11EE22DC
:04000400DD33CC44D8
:04000800BB550012D2

Address	Data Stored in RAM
0	0x11FF
1	0x22EE
2	0x33DD
3	0x44CC
4	0x55BB
5	0x1200

3.1.3.1.3 8-bit Write Port Width (Ask a Question)

When the memory write port width is 8-bits, which is aligned on a byte boundary, the tool uses 8-bits from the binary stream for each 8-bit word in the RAM. The resulting data stored in RAM is shown in the following figure.

Figure 3-4. Data in RAM - 8-bit Write Port Width Aligned on Byte Boundary

Memory File Data
:04000000FF11EE22DC
:04000400DD33CC44D8
:04000800BB550012D2

Address	Data Stored in RAM
0	0xFF
1	0x11
2	0xEE
3	0x22
4	0xDD
5	0x33
6	0xCC
7	0x44
8	0xBB
9	0x55
A	0x00
B	0x12

3.1.3.2 Write Port Widths Not Aligned on Byte Boundary (Ask a Question)

The following examples show how data from a memory file is stored in RAM when memory port widths are not a multiple of a byte.

3.1.3.2.1 9-bit Write Port Width (Ask a Question)

When the memory write port width is 9-bits, which is not aligned on a byte boundary, the tool uses 16-bits from the binary stream for each 9-bit word. The tool ignores the upper 7-bits of each 16-bits

when processing the memory file data. The resulting data stored in RAM is shown in the following figure.

Figure 3-5. Data in RAM - 9-bit Write Port Width Not Aligned on Byte Boundary

Memory File Data
:04000000FF11EE22DC
:04000400DD33CC44D8
:04000800BB550012D2

Address	Data Stored in RAM
0	0x1FF
1	0x0EE
2	0x1DD
3	0x0CC
4	0x1BB
5	0x000

3.1.3.2.2 4-Bit Write Port Width (Ask a Question)

When the memory write port width is 4-bits, which is not aligned on a byte boundary, the tool uses 8-bits from the binary stream for each 4-bit word. The tool ignores the upper 4-bits of each 8-bits when processing the memory file data. The resulting data stored in RAM is shown in the following figure.

Figure 3-6. Data in RAM—4-Bit Write Port Width Not Aligned on Byte Boundary

Memory File Data
:04000000FF11EE22DC
:04000400DD33CC44D8
:04000800BB550012D2

Address	Data Stored in RAM
0	0xF
1	0x1
2	0xE
3	0x2
4	0xD
5	0x3
6	0xC
7	0x4
8	0xB
9	0x5
A	0x0
B	0x2

3.1.3.3 Specifying Data in Memory File (Ask a Question)

If all the bits in the memory file are relevant, take steps to avoid bits from being ignored when the write port width is not aligned on a byte boundary. The following examples describe how to specify data in memory file so that the tool does not ignore bits. To avoid bits from being ignored, convert the hexadecimal data intended to initialize RAM into binary stream of bits, and then pad (insert) zeros based on the port width so that resulting data is byte-aligned, as described in the following sections.

3.1.3.3.1 9-bit Write Port Width (Ask a Question)

Consider the following Intel HEX memory file.

Figure 3-7. Memory File Data - Intel HEX Memory File

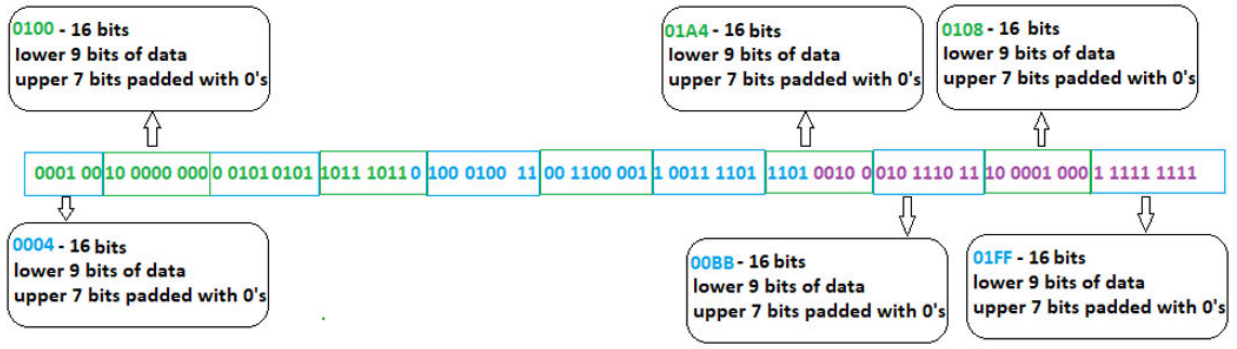
```
:04000000FF11EE22DC
:04000400DD33CC44D8
:04000800BB550012D2
```

The binary stream of bits for the preceding memory file data is:

```
0001 0010 0000 0000 0101 0101 1011 1011 0100 0100 1100 1100 0011 0011 1101 1101 0010 0010 1110 1110 0001 0001 1111 1111
```

If the memory port width is 9-bits, you must pad seven zeros to every 9-bits of data from the binary stream to create 16-bits (byte-aligned), as shown in the following figure.

Figure 3-8. Padding Zeros to Create 16 Bits



The following figure shows the equivalent memory file data padded with zeros to achieve a 9-bit write port width.

Figure 3-9. Equivalent Memory File Data Padded with Zeros (9-bit Write Port Width)

```
:04000000FF010801F3
:04000400BB00A40198
:040008003D01610055
:04000C001301760165
:0400100055000001 96
:04001400040000E4
```

When the tool parses the above memory file data (padded with zeros), the tool converts the data to binary and reads it as a stream of bits. If the port width is 9 bits, the tool reads 16 bits (byte-aligned), ignores the upper 7 bits, and stores the lower 9 bits of actual data in RAM, as shown in the following table.

Table 3-1. 16-bit Write Port Width

Address	Data
0	0x1FF
1	0x108

.....continued	
Address	**Data**
2	0x0BB
3	0x1A4
4	0x13D
5	0x061
6	0x113
7	0x176
8	0x055
9	0x100
A	0x004
B	0x000

3.1.3.3.2 4-bit Write Port Width (Ask a Question)

Consider the following Intel HEX memory file.

Figure 3-10. Memory File Data - Intel HEX Memory File

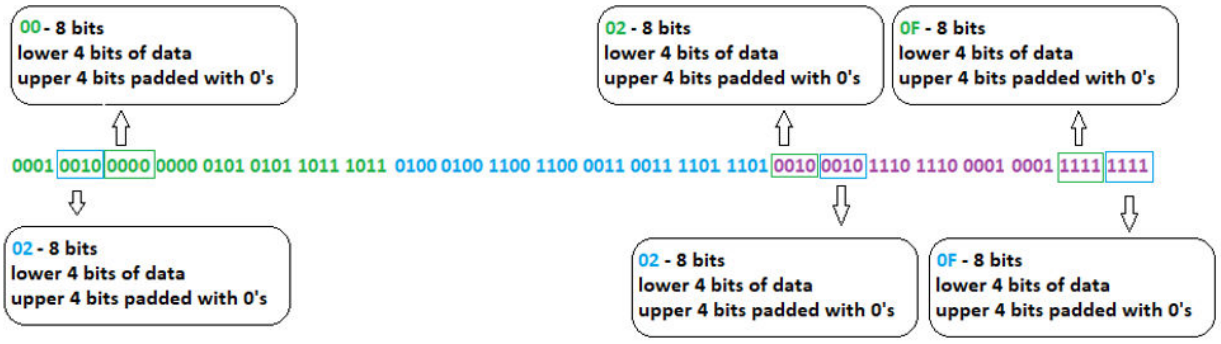
```
:04000000FF11EE22DC
:04000400DD33CC44D8
:04000800BB550012D2
```

The binary stream of bits for the preceding memory file data is:

```
0001 0010 0000 0000 0101 0101 1011 1011 0100 0100 1100 1100 0011 0011 1101 1101 0010 0010 1110 1110 0001 0001 1111 1111
```

If the memory port width is 4-bits, the tool reads 8-bits at a time from the binary stream above. For the 8-bits, you must pad zeros for the upper 4-bits and specify the actual data in the lower 4-bits, as shown in the following figure.

Figure 3-11. Padding Zeros for the Upper 4 Bits and Specifying Data in the Lower 4 Bits



The following figure shows the equivalent memory file data padded with zeros to achieve a 4-bit write port width.

Figure 3-12. Equivalent Memory File Data Padded with Zeros (4-bit Write Port Width)

```
:040000000F0F0101DC
:040004000E0E0202D8
:040008000D0D0303D4
:04000C000C0C0404D0
:040010000B0B0505CC
:0400140000000201E5
```

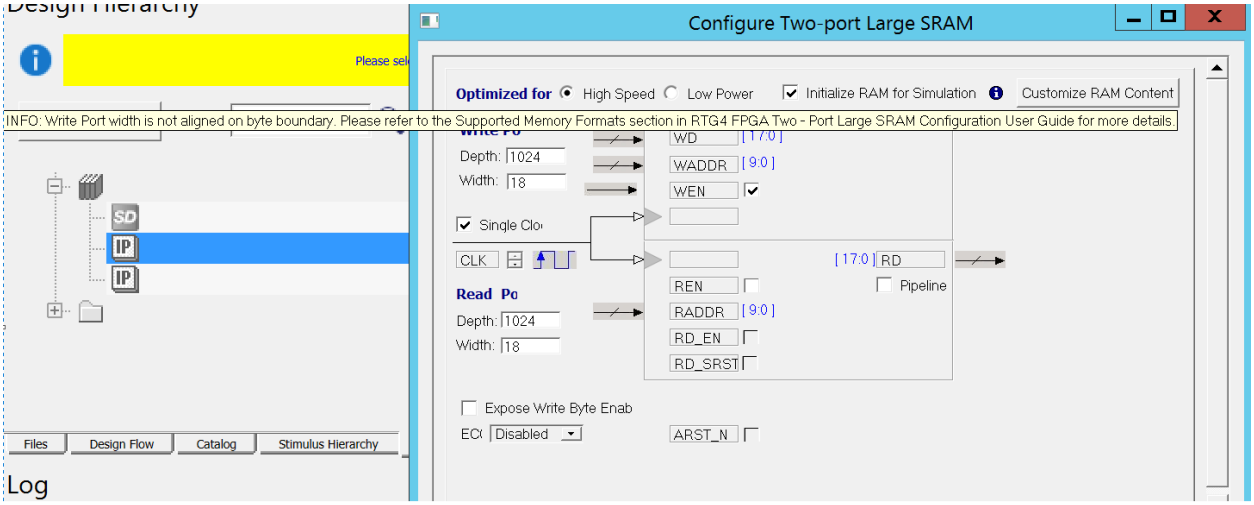
When the tool parses the above memory file data (padded with zeros), it converts the data to binary and reads it as a stream of bits. If the port width is 4 bits, the tool reads 8 bits (byte-aligned), ignores the upper 4 bits of actual data, and stores the lower 4 bits of actual data in RAM, as shown in the following table.

Table 3-2. 16-bit Write Port Width

Address	Data
0	0xF
1	0xF
2	0x1
3	0x1
4	0xE
5	0xE
6	0x2
7	0x2
8	0xD
9	0xD
A	0x3
B	0x3
C	0xC
D	0xC
E	0x4
F	0x4
10	0xB
11	0xB
12	0x5
13	0x5
14	0x0
15	0x0
16	0x2
17	0x1

Note: x1 and x2 port widths are handled using the same technique of padding zeros. You always zero pad to the next 8-bit-width increment (8, 16, 24, and so on). If a write port width is not aligned on a byte boundary, the following message appears.

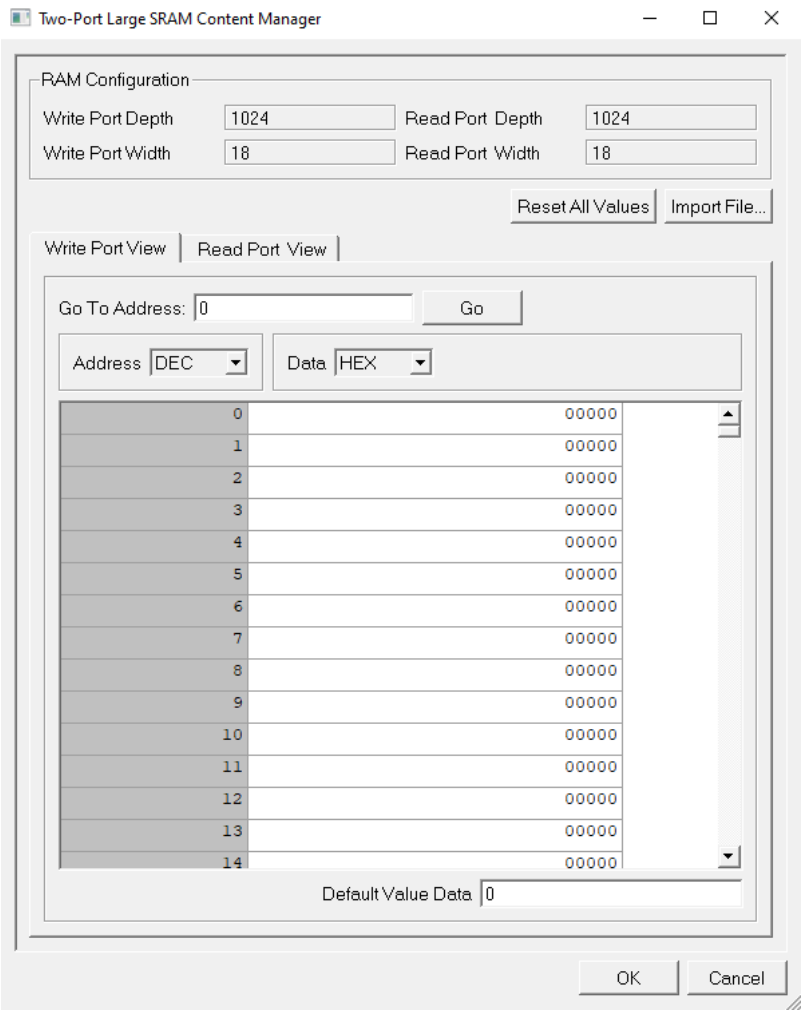
Figure 3-13. Message when a Write Port Width is Not Aligned on a Byte Boundary



3.2 RAM Content Manager Functionality (Ask a Question)

To open the RAM Content Manager, after specifying your RAM configuration (set your Read and Write Depth and Width), select the Initialize RAM for Simulation checkbox, and then click Customize RAM Content. The RAM Content Manager appears as shown in the following figure.

Figure 3-14. Customize RAM Content for Simulation



Important:

1. Clearing and selecting the **Initialize RAM for Simulation** check box in the same dialog box does not discard the memory file data in the generated files. To discard the memory file data, clear the **Initialize RAM for Simulation** check box and click **OK**.
2. After selecting the **Initialize RAM for Simulation** check box, use the **Customize RAM Content** option to initialize memory content to zeros.

3.2.1 RAM Configuration (Ask a Question)

Write Depth and Write Width - As specified in the RAM core generator dialog box (not editable).

Read Depth and Read Width - As specified in the RAM core generator dialog box (not editable).

3.2.2 Write Port View/Read Port View (Ask a Question)

Go To Address - Enables you to go to a specific address in the manager. Each memory block has many addresses; it is often difficult to scroll through and find a specific one. This task is simplified by enabling you to type in a specific address. The number display format (Hex, Bin, and Dec) is controlled by the value you set in the drop-down menu above the Address column.

Address - The Address column lists the address of a memory location. The drop-down menu specifies the number format for your address list (hexadecimal, binary, or decimal).

Data - Enables you to control the data format and data value in the manager. Click the value to change it.

Note that the dialogs show all data with the MSB down to LSB. For example, if the row showed 0xAABB for a 16-bit word size, the AA would be the MSB and the BB would be LSB.

Default Data Value - The value given to memory addresses that have not been explicitly initialized (by importing content or editing manually). When changed, all default values in the manager are updated to match the new value. The number display format (Hex, Bin, Dec) is controlled by the value you set in the drop-down menu above the Data column.

Reset All Values - Resets the Data values.

Import from File - Opens the Import Memory Content dialog box; enables you to select a memory content file (Intel-Hex) to load. Intel-Hex file extensions are set to `*.hex` during import.

OK - Closes the manager and saves all the changes made to the memory and its contents.

Cancel - Closes the manager, cancels all your changes in this instance of the manager, and returns the memory to the state it held before the manager was opened.

3.3 MEMFILE (RAM Content Manager output file) (Ask a Question)

Transfer of RAM data (from the RAM Content Manager) to test equipment is accomplished via MEM files. The contents of your RAM are first organized into the logical layer and then reorganized to fit the hardware layer. Then it is stored in MEM files that are read by other systems and used for testing.

The MEM files are named according to the logical structure of RAM elements created by the configurator. In this scheme, the highest order RAM blocks are named CORE_R0C0.mem, where "R" stands for row and "C" stands for column. For multiple RAM blocks, the naming continues with CORE_R0C1, CORE_R0C2, CORE_R1C0, etc.

The data intended for the RAM is stored as ASCII 1s and 0s within the file. Each memory address occupies one line. Words from logical layer blocks are concatenated or split to make them fit efficiently within the hardware blocks. If the logical layer width is less than the hardware layer, two or more logical layer words are concatenated to form one hardware layer word. In this case, the lowest bits of the hardware word are made up of the lower address data bits from the logical layer. If the logical layer width is more than the hardware layer, the words are split, and placing the lower bits in lower addresses.

If the logical layer words do not fit cleanly into the hardware layer words, the most significant bit of the hardware layer words is not used and defaulted to zero. This is also done when the logical layer width is 1 to avoid having leftover memory at the end of the hardware block.

4. Port Description (Ask a Question)

The following table lists the Two-Port Large SRAM signals in the generated macro.

Table 4-1. Two-Port Large SRAM Signals

Port	Direction	Default Polarity	Description
CLK	In	Rising Edge	Single clock to drive both WCLK and RCLK
WD[]	In	—	Write data
WADDR[]	In	—	Write address
WEN	In	Active-high	Write port enable
WCLK	In	Rising-edge	Write clock
RCLK	In	Rising-edge	Read clock
REN	In	Active-high	Read port enable
RADDR[]	In	—	Read address
RD[]	Out	—	Read data
RD_EN	In	Active-high	Read data register enable
RD_SRST_N	In	Active-low	Read data register Synchronous reset
ARST_N	In	Active-low	Read data register Asynchronous reset

5. Parameters (Ask a Question)

The following table lists the Micro SRAM parameters in the generated macro.

Table 5-1. Two-Port Large SRAM Parameters

GENFILE Parameter	Configurator Parameter	Valid Range	Default	Description
DESIGN	—	—	—	Name of the generated macro
FAM	—	SmartFusion2	—	Target family
OUTFORMAT	—	Verilog, VHDL	—	Netlist format
LPMTYPE	—	LPM_RAM	—	Macro category
DEVICE	—	500 - 5000	5000	Target device
PTYPE	PTYPE	1, 2	1	1: Two-port
INIT_RAM	INIT_RAM	F, T	F	Initialize RAM for simulation
CASCADE	CASCADE	0, 1	0	• 0: Cascading for WIDTH or Speed • 1: Cascading for DEPTH or Power
CLKS	CLKS	1, 2	1	• 1: Single Read/Write Clock • 2: Independent Read and Write Clocks
WCLK_EDGE	CLK_EDGE	CLKS=1 RISE, FALL	RISE	• RISE: Rising edge Single clock • FALL: Falling edge Single clock
WWIDTH	WWIDTH	1-2484	18	Write data width
WDEPTH	WDEPTH	1-524288	1024	Write address depth
RWIDTH	RWIDTH	1-2484	18	Read data output width
RDEPTH	RDEPTH	1- 524288	1024	Read address depth
WE_POLARITY	WE_POLARITY	0, 1, 2	1	• 0: Active-low Write port enable • 1: Active-high Write port enable • 2: Write port enable tied-off to be always active
WCLK_EDGE	WCLK_EDGE	CLKS=2 RISE, FALL	RISE	• RISE: Rising edge Write clock • FALL: Falling edge Write clock
RCLK_EDGE	RCLK_EDGE	CLKS=2 RISE, FALL	RISE	• RISE: Rising edge Read clock • FALL: Falling edge Read clock

.....continued

GENFILE Parameter	Configurator Parameter	Valid Range	Default	Description
RE_POLARITY	RE_POLARITY	0, 1, 2	2	• 0: Active-low Read port enable • 1: Active-high Read port enable • 2: Read port enable tied-off to be always active
PMODE2	RPMODE	0, 1	0	• 0: Bypass Read data register • 1: Pipeline Read data
A_DOUT_EN_POLARITY	A_DOUT_EN_POLARITY	PMODE2=1 0, 1, 2	2	• 0: Active-low Read data register enable • 1: Active-high Read data register enable • 2: Read data register enable tied-off to be always active
A_DOUT_SRST_POLARITY	A_DOUT_SRST_POLARITY	PMODE2=1 0, 1, 2	2	• 0: Active-low Read data register Sync-reset • 1: Active-high Read data register Sync-reset • 2: Read data register Sync-reset tied-off to be always inactive
RESET_POLARITY	ARST_N_POLARITY	PMODE2=1 0, 1, 2	2	• 0: Active-low Read data register Async-reset • 1: Active-high Read data register Async-reset • 2: Read data register Async-reset tied-off to be always inactive
CLOCK_PN	CLOCK_PN	CLKS=1	CLK	Single clock Port name
DATA_IN_PN	DATA_IN_PN	—	WD	Write data Port name
WADDRESS_PN	WADDRESS_PN	—	WADDR	Write address Port name
WE_PN	WE_PN	WE_POLARITY<2	WEN	Write port enable Port name
WCLOCK_PN	WCLOCK_PN	CLKS=2	WCLK	Write clock Port name
RCLOCK_PN	RCLOCK_PN	CLKS=2	RCLK	Read clock Port name
RE_PN	RE_PN	RE_POLARITY<2	REN	Read port enable Port name

.....continued

GENFILE Parameter	Configurator Parameter	Valid Range	Default	Description
RADDRESS_PN	RADDRESS_PN	—	RADDR	Read address Port name
DATA_OUT_PN	DATA_OUT_PN	—	RD	Read data Port name
A_DOUT_EN_PN	A_DOUT_EN_PN	PMODE2=1	RD_EN	Read data register enable Port name
A_DOUT_SRST_PN	A_DOUT_SRST_PN	PMODE2=1	RD_SRST_N	Read data register Sync-reset Port name
RESET_PN	RESET_PN	PMODE2=1	ARST_N	Read data register Async-reset Port name
COLLISION_WARN_MSGS	COLLISION_WARN_MSGS	-1, 0, >0	-1	• -1: All warning messages related to collisions will appear in the simulation log. • 0: One warning message related to collisions will appear in the simulation log. • >0: More than one warning message equal to the integer value passed to the parameter related to collisions will appear in the simulation log.

6. Revision History (Ask a Question)

The revision history describes the changes that were implemented in the document. The changes are listed by revision, starting with the most current publication.

Revision	Date	Description
A	01/2024	The following is the list of changes in revision A of this document: • Converted the document to Microchip template • The document number was changed to DS50003646 from 50200349 • Added a note about the **Initialize RAM for Simulation** option. See 3.2. RAM Content Manager Functionality. • Added a section 3.1.3. Write Port Width Alignment.

Microchip FPGA Support

Microchip FPGA products group backs its products with various support services, including Customer Service, Customer Technical Support Center, a website, and worldwide sales offices. Customers are suggested to visit Microchip online resources prior to contacting support as it is very likely that their queries have been already answered.

Contact Technical Support Center through the website at www.microchip.com/support. Mention the FPGA Device Part number, select appropriate case category, and upload design files while creating a technical support case.

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**

Microchip Information

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip products:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner, within operating specifications, and under normal conditions.
- Microchip values and aggressively protects its intellectual property rights. Attempts to breach the code protection features of Microchip product is strictly prohibited and may violate the Digital Millennium Copyright Act.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is “unbreakable”. Code protection is constantly evolving. Microchip is committed to continuously improving the code protection features of our products.

Legal Notice

This publication and the information herein may be used only with Microchip products, including to design, test, and integrate Microchip products with your application. Use of this information in any other manner violates these terms. Information regarding device applications is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Contact your local Microchip sales office for additional support or, obtain additional support at www.microchip.com/en-us/support/design-help/client-support-services.

THIS INFORMATION IS PROVIDED BY MICROCHIP "AS IS". MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL, OR CONSEQUENTIAL LOSS, DAMAGE, COST, OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION.

Use of Microchip devices in life support and/or safety applications is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, CryptoMemory, CryptoRF, dsPIC, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AgileSwitch, ClockWorks, The Embedded Control Solutions Company, EtherSynch, Flashtec, Hyper Speed Control, HyperLight Load, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, TimeCesium, TimeHub, TimePictra, TimeProvider, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, Augmented Switching, BlueSky, BodyCom, Clockstudio, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic

ISBN:

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office	**Australia - Sydney**	**India - Bangalore**	**Austria - Wels**
2355 West Chandler Blvd.	Tel: 61-2-9868-6733	Tel: 91-80-3090-4444	Tel: 43-7242-2244-39
Chandler, AZ 85224-6199	**China - Beijing**	**India - New Delhi**	Fax: 43-7242-2244-393
Tel: 480-792-7200	Tel: 86-10-8569-7000	Tel: 91-11-4160-8631	**Denmark - Copenhagen**
Fax: 480-792-7277	**China - Chengdu**	**India - Pune**	Tel: 45-4485-5910
Technical Support:	Tel: 86-28-8665-5511	Tel: 91-20-4121-0141	Fax: 45-4485-2829
www.microchip.com/support	**China - Chongqing**	**Japan - Osaka**	**Finland - Espoo**
Web Address:	Tel: 86-23-8980-9588	Tel: 81-6-6152-7160	Tel: 358-9-4520-820
www.microchip.com	**China - Dongguan**	**Japan - Tokyo**	**France - Paris**
Atlanta	Tel: 86-769-8702-9880	Tel: 81-3-6880- 3770	Tel: 33-1-69-53-63-20
Duluth, GA	**China - Guangzhou**	**Korea - Daegu**	Fax: 33-1-69-30-90-79
Tel: 678-957-9614	Tel: 86-20-8755-8029	Tel: 82-53-744-4301	**Germany - Garching**
Fax: 678-957-1455	**China - Hangzhou**	**Korea - Seoul**	Tel: 49-8931-9700
Austin, TX	Tel: 86-571-8792-8115	Tel: 82-2-554-7200	**Germany - Haan**
Tel: 512-257-3370	**China - Hong Kong SAR**	**Malaysia - Kuala Lumpur**	Tel: 49-2129-3766400
Boston	Tel: 852-2943-5100	Tel: 60-3-7651-7906	**Germany - Heilbronn**
Westborough, MA	**China - Nanjing**	**Malaysia - Penang**	Tel: 49-7131-72400
Tel: 774-760-0087	Tel: 86-25-8473-2460	Tel: 60-4-227-8870	**Germany - Karlsruhe**
Fax: 774-760-0088	**China - Qingdao**	**Philippines - Manila**	Tel: 49-721-625370
Chicago	Tel: 86-532-8502-7355	Tel: 63-2-634-9065	**Germany - Munich**
Itasca, IL	**China - Shanghai**	**Singapore**	Tel: 49-89-627-144-0
Tel: 630-285-0071	Tel: 86-21-3326-8000	Tel: 65-6334-8870	Fax: 49-89-627-144-44
Fax: 630-285-0075	**China - Shenyang**	**Taiwan - Hsin Chu**	**Germany - Rosenheim**
Dallas	Tel: 86-24-2334-2829	Tel: 886-3-577-8366	Tel: 49-8031-354-560
Addison, TX	**China - Shenzhen**	**Taiwan - Kaohsiung**	**Israel - Ra'anana**
Tel: 972-818-7423	Tel: 86-755-8864-2200	Tel: 886-7-213-7830	Tel: 972-9-744-7705
Fax: 972-818-2924	**China - Suzhou**	**Taiwan - Taipei**	**Italy - Milan**
Detroit	Tel: 86-186-6233-1526	Tel: 886-2-2508-8600	Tel: 39-0331-742611
Novi, MI	**China - Wuhan**	**Thailand - Bangkok**	Fax: 39-0331-466781
Tel: 248-848-4000	Tel: 86-27-5980-5300	Tel: 66-2-694-1351	**Italy - Padova**
Houston, TX	**China - Xian**	**Vietnam - Ho Chi Minh**	Tel: 39-049-7625286
Tel: 281-894-5983	Tel: 86-29-8833-7252	Tel: 84-28-5448-2100	**Netherlands - Drunen**
Indianapolis	**China - Xiamen**		Tel: 31-416-690399
Noblesville, IN	Tel: 86-592-2388138		Fax: 31-416-690340
Tel: 317-773-8323	**China - Zhuhai**		**Norway - Trondheim**
Fax: 317-773-5453	Tel: 86-756-3210040		Tel: 47-72884388
Tel: 317-536-2380			**Poland - Warsaw**
Los Angeles			Tel: 48-22-3325737
Mission Viejo, CA			**Romania - Bucharest**
Tel: 949-462-9523			Tel: 40-21-407-87-50
Fax: 949-462-9608			**Spain - Madrid**
Tel: 951-273-7800			Tel: 34-91-708-08-90
Raleigh, NC			Fax: 34-91-708-08-91
Tel: 919-844-7510			**Sweden - Gothenberg**
New York, NY			Tel: 46-31-704-60-40
Tel: 631-435-6000			**Sweden - Stockholm**
San Jose, CA			Tel: 46-8-5090-4654
Tel: 408-735-9110			**UK - Wokingham**
Tel: 408-436-4270			Tel: 44-118-921-5800
Canada - Toronto			Fax: 44-118-921-5820
Tel: 905-695-1980			
Fax: 905-695-2078			