Tightly-Coupled Memory IP – Lattice Propel Builder 2026.1

IP Version: v1.5.4

User Guide

FPGA-IPUG-02325-1.0

June 2026

Disclaimers

Lattice makes no warranty, representation, or guarantee regarding the accuracy of information contained in this document or the suitability of its products for any particular purpose. All information herein is provided AS IS, with all faults, and all associated risk is the responsibility entirely of the Buyer. The information provided herein is for informational purposes only and may contain technical inaccuracies or omissions, and may be otherwise rendered inaccurate for many reasons, and Lattice assumes no obligation to update or otherwise correct or revise this information. Products sold by Lattice have been subject to limited testing and it is the Buyer's responsibility to independently determine the suitability of any products and to test and verify the same. LATTICE PRODUCTS AND SERVICES ARE NOT DESIGNED, MANUFACTURED, OR TESTED FOR USE IN LIFE OR SAFETY CRITICAL SYSTEMS, HAZARDOUS ENVIRONMENTS, OR ANY OTHER ENVIRONMENTS REQUIRING FAIL-SAFE PERFORMANCE, INCLUDING ANY APPLICATION IN WHICH THE FAILURE OF THE PRODUCT OR SERVICE COULD LEAD TO DEATH, PERSONAL INJURY, SEVERE PROPERTY DAMAGE OR ENVIRONMENTAL HARM (COLLECTIVELY, "HIGH-RISK USES"). FURTHER, BUYER MUST TAKE PRUDENT STEPS TO PROTECT AGAINST PRODUCT AND SERVICE FAILURES, INCLUDING PROVIDING APPROPRIATE REDUNDANCIES, FAIL-SAFE FEATURES, AND/OR SHUT-DOWN MECHANISMS. LATTICE EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS OF THE PRODUCTS OR SERVICES FOR HIGH-RISK USES. The information provided in this document is proprietary to Lattice Semiconductor, and Lattice reserves the right to make any changes to the information in this document or to any products at any time without notice.

Inclusive Language

This document was created consistent with Lattice Semiconductor’s inclusive language policy. In some cases, the language in underlying tools and other items may not yet have been updated. Please refer to Lattice’s inclusive language FAQ 6878 for a cross reference of terms. Note in some cases such as register names and state names it has been necessary to continue to utilize older terminology for compatibility.

Contents

Figures

Tables

Abbreviations in This Document

A list of abbreviations used in this document.

Abbreviation	Definition
ASCII	American Standard Code for Information Interchange
CPU	Central Processing Unit
EBR	Embedded Block RAM
ECC	Error Correction Code
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
GSR	Global Set/Reset
GUI	Graphical User Interface
HDL	Hardware Description Language
LRAM	Large RAM
LUT	Lookup Table
RAM	Random Access Memory
RISC-V RX	Reduced Instruction Set Computer – V (Five) for Real-Time Operating System Applications
ROM	Read-Only Memory
TCM	Tightly-Coupled Memory

1. Introduction

This document provides technical information about the Tightly-Coupled Memory (TCM) IP and aims to provide information essential for IP and system development, verification, integration, testing, and validation.

The design is implemented in Verilog HDL. The IP can be configured and generated based on Table 1.1.

Table 1.1. TCM IP Quick Facts

IP Requirements	Supported FPGA Family	Certus™-N2, Lattice Avant™, MachXO5™-NX, CrossLinkU™-NX, CrossLink™-NX, CertusPro™-NX, and Certus-NX
Resource Utilization	Supported User Interfaces	Local Bus Interface, FIFO Interface
	Resources	See Table A.1 and Table A.2.
Design Tool Support	Lattice Implementation	IP v1.5.4 – Lattice Propel™ Builder 2026.1
	Simulation	For a list of supported simulators, see the Lattice Radiant™ software user guide.

1.1. What's New in This IP Release

Updated device support for MachXO5-NX, Certus-N2, and Lattice Avant families.

1.2. Features

- Configurable as a single-port or a dual-port memory
- The core memory can be implemented as EBR, LRAM, or Distributed RAM.
- Supports the ROM and the RAM mode.
- Supports configurable byte enables.
- Supports byte writes when used with compatible hardware.
- Configurable data width
- Configurable memory depth
- Configurable address range for every port
- Uses 32-bit data word transfers.
- Unaligned access port for every port
- Uses the little-endian bit structure.
- Burst write and read feature
- Supports the ATOMIC load or store.
- Optional FIFO Streaming Interface

1.3. Conventions

1.3.1. Nomenclature

The nomenclature used in this document is based on Verilog HDL.

1.3.2. Signal Names

Signal Names that end with:

- _n are active low signals.
- _i are input signals.
- _o are output signals.
- _io are bi-directional input/output signals.

1.4. Licensing and Ordering Information

The TCM IP is provided at no additional cost with the Lattice Propel design environment. The IP can be fully evaluated in hardware without requiring an IP license string.

2. Functional Descriptions

2.1. Block Diagram

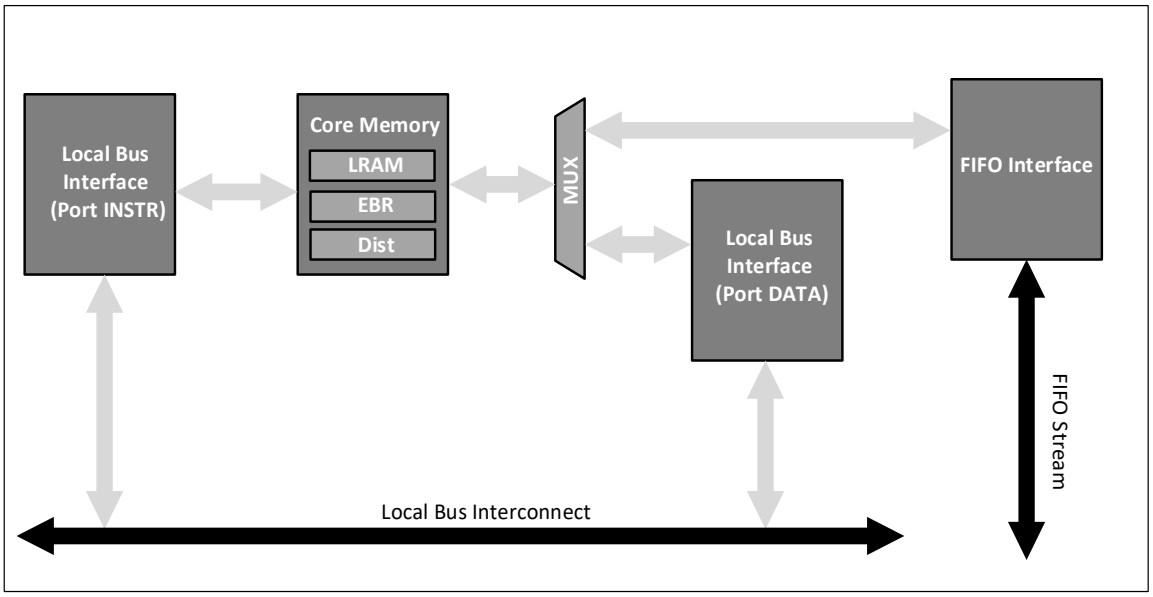

Figure 2.1. Generic TCM IP Block Diagram

2.1.1. Memory Implementation

The TCM memory module uses Embedded Block RAMs (EBR) or Distributed Memory in Certus-N2 and Lattice Avant family devices. As for CrossLink-NX, Certus-NX, CertusPro-NX, and MachXO5-NX family devices, Large RAMs can also be used. The memory can be configured as true dual-port or single-port. Each port can be configured for read/write or read-only access. The number of ports and read/write configuration of the TCM IP module automatically select the best type of memory for the user-selected application.

Table 2.1. System Core Memory Implementation

Memory Type	Configuration Used[1]	CrossLink-NX, Certus-NX, CertusPro-NX, and MachXO5-NX	Certus-N2 and Lattice Avant
LRAM	1 port : “W/R” 2 ports: “W/R” + “W/R”	Yes	No
EBR – ram_dp_true	2 ports: “W/R” + “W/R”	Yes	Yes
EBR – ram_sp	1 port : “W/R”	Yes	Yes
Distributed Memory	1 port : “W/R”	Yes	Yes

Note:

1. Byte-Enable cannot be used in conjunction with ECC. LRAM is not present in Certus-N2 and Lattice Avant family devices.

Table 2.2. Features Supported per Memory Block

Feature	LRAM	EBR	Distributed Memory
Memory Initialization	Yes	Yes	No
Registered Output	Yes	Yes	Yes
Dual-Port Configuration	Yes	Yes	No
Byte-Enable	Yes	Yes	No

2.2. Functional Overview

2.2.1. Local Bus Interface

The local bus interface is a lightweight data streaming interface connecting the RISC-V RX CPU IP to the TCM IP. The TCM memory module is designed to be fully compatible with the local bus interface. It can be configured as a single-port or a dual-port interface, depending on whether a single-port or a dual-port memory is needed along with the RAM or ROM configuration.

2.2.2. FIFO Interface

The TCM supports an optional FIFO interface to enable data to be streamed into the core memory. The FIFO interface is shared with Port DATA. This interface is used to inject the FIFO data from a FIFO stream. This is typically used to upload firmware values to the core memory. The FIFO starts writing at the designated first byte-addressable data and writes up to the maximum depth of the implemented memory.

Table 2.3. Allowable Combination of Features for TCM

Device	Byte-Enable	ECC	Unaligned Access	FIFO	Maximum Supported Port Count
LRAM	×	×	×	×	2
	×	×	×	✓	2
	×	×	✓	×	2
	×	×	✓	✓	2
	×	✓	×	×	2
	×	✓	×	✓	Not supported
	×	✓	✓	×	2
	×	✓	✓	✓	Not supported
	✓	×	×	×	2
	✓	×	×	✓	2
	✓	×	✓	×	2
	✓	×	✓	✓	2
	✓	✓	×	×	Not supported
	✓	✓	×	✓	Not supported
	✓	✓	✓	×	Not supported
	✓	✓	✓	✓	Not supported
EBR	×	×	N/A	×	2
	×	×	N/A	✓	2
	×	✓	N/A	×	2[1]
	×	✓	N/A	✓	Not supported
	✓	×	N/A	×	2
	✓	×	N/A	✓	2
	✓	✓	N/A	×	Not supported
	✓	✓	N/A	✓	Not supported

Note:

1. EBR ECC is limited to Single-Port W/R or Dual-Port W/O + R/O combinations only.

2.3. Attribute Summary

Table 2.4. TCM IP Attribute Summary

Parameter Name	Values	Default	Description
General			
Enable GSR	True, False	False	Enables global set/reset.
Memory Type	EBR, Distributed RAM, LRAM	EBR	Selects the type of memory implemented for this instance of TCM IP.
Port Count[1]	2	2	Determines if the TCM IP uses a single port or dual ports.
ECC Enable[2]	True, False	False	Determines if ECC is used. This applies to both ports.
Access Type for Port S0[3]	R/W, W/O, R/O	R/W	Determines the access for Port INSTR.
Access Type for Port S1[4]	R/W, W/O, R/O	R/W	Determines the access for Port DATA.
ATOMIC[5]	True, False	False	Indicates whether the TCM IP supports ATOMIC instructions or not.
Port S0/S1 Data Width	32, 64	32	Data width of the memory for Port INSTR and Port DATA
Port S0/S1 Address Depth	128 or 256 to device EBR or LRAM upper limit	512	Address depth of the memory for Port INSTR and Port DATA. The number should be a multiple of 256 if Data Width is 32 or a multiple of 128 if Data Width is 64. The maximum address depth depends on the data width, the memory type, and the device used.
FIFO Streamer			
Enable FIFO Streamer	True, False	False	Enables the FIFO Streamer interface. If this is used when Port Count is 2, the Streamer interface is muxed with Port DATA. Otherwise, it has its own dedicated port.
FIFO Clock Bypass	True, False	False	When enabled, the FIFO streamer uses the sys_clk as the clock source. Otherwise, it has its own dedicated clock. It is editable only when Enable FIFO Streamer is True.
FIFO Write Start Address	0 to ADDR_DEPTH-1	0	The start address where the FIFO streamer starts to write. Editable only when Enable FIFO Streamer is True.
Memory			
Memory Initialization	None, Memory File	None	Determines whether to enable initialization of the memory by providing the initialization file or not.
Memory File	<string>	None	Selects the initialization file for the TCM IP.
Memory File Format	Hex, Binary	Binary	Determines the file format of the initialization file.
Port S0 Settings[3]			
Port S0: Address Width	$\log_2$ ADDR_DEPTH	—	Maximum address to be used for Port INSTR
Read Port: Enable Output Register (S0)	True, False	False	Applies a registered output for Port INSTR.
Byte Enable for Port S0[6, 7]	True, False	True	Determines if Byte Enable for Port INSTR is used.
Reset Behavior for Port S0	Async, Sync	Sync	The reset mode for Port INSTR
Unaligned Access for Port S0[6, 8]	True, False	False	Enables the unaligned read functionality for Port INSTR.

Parameter Name	Values	Default	Description
Edit Address Range Port S0	True, False	False	Enables memory address offset for Port INSTR.
Start Address Port S0	0 to Address Depth–1	0	The starting memory address offset for Port INSTR
End Address Port S0	0 to Address Depth–1	Address Depth–1	Ending memory address offset for Port INSTR. It must be greater than Start Address Port S0.
Port S1 Settings[4]			
Port S1: Address Width	$\log_2$ ADDR_DEPTH	—	The maximum address to be used for Port DATA
Read Port: Enable Output Register (S1)	True, False	False	Applies a registered output for Port DATA.
Byte Enable for Port S1[6, 7]	True, False	True	Determines if Byte Enable for Port DATA is used.
Reset Behavior for Port S1	Async, Sync	Sync	The reset mode for Port DATA
Unaligned Access for Port S1[6, 8]	True, False	False	Enables the unaligned read functionality for Port DATA.
Edit Address Range Port S1	True, False	False	Enables memory address offset for Port DATA.
Start Address Port S1	0 to Address Depth–1	0	The starting memory address offset for Port DATA
End Address Port S1	0 to Address Depth–1	Address Depth–1	The ending memory address offset for Port DATA. It must be greater than Start Address Port S1.

Notes:

1. Port Count is fixed at 2.
2. Applies only to LRAM and EBR. Dual-port EBR ECC is limited to W/O and R/O for Access Type for Port S0 and Access Type for Port S1 respectively.
3. After the IP generation, you can see Port INSTR instead of Port S0.
4. After the IP generation, you can see Port DATA instead of Port S1.
5. ATOMIC should be enabled when connected to version 2.4.0 or above of the RX CPU IP core in the Balanced or the Advanced mode. It should be disabled in other cases. Refer to ATOMIC Instruction Support for more details.
6. Byte Enable and Unaligned Access use the lower address bits as control signals.
7. Byte Enable is not available when ECC is enabled.
8. Available for LRAM only.

2.3.1. Start Address and End Address Description and Implementation

The start address is hardwired to 0x0 and the end address is hardwired based on the address depth. The access to TCM should be in this range. Otherwise, the TCM IP responds with an out-of-bounds error on ibus_rsp_error/dbus_rsp_error.

2.4. Signal Descriptions

Table 2.5. TCM IP Ports

Pin Name	Direction	Width (bits)	Description
sys_clk_i	In	1	Clock source
sys_rst_i	In	1	Active low reset signal
Port INSTR			
ibus_cmd_valid_i	Input	1	Indicates that the manager is signaling valid write or read address and control information.
ibus_cmd_ready_o	Output	1	Indicates that the target is ready to accept an address and associated control signals.
ibus_cmd_payload_wr_i	Input	1	Indicates the transfer direction. When HIGH, this signal indicates a write transfer. When LOW, it indicates a read transfer. It has the same timing as that of the address signals. However, it does not remain constant throughout a burst transfer.
ibus_cmd_payload_uncached_i	Input	1	Indicates if the transfer is a cache operation. When HIGH, this signal indicates a cache transfer. When LOW, it indicates an uncached transfer.
ibus_cmd_payload_address_i	Input	32	Address
ibus_cmd_payload_data_i	Input	DATA_WIDTH_A	The write data bus transfers data from the manager to the target during write operations.
ibus_cmd_payload_mask_i	Input	DATA_WIDTH_A/8	Indicates the byte lanes that hold valid data. There is one write mask bit for every eight bits of the write data bus.
ibus_cmd_payload_size_i	Input	3	Indicates the size of the transfer, which is typically byte, halfword, or word. 3'b101 indicates an 8-word burst transfer.
ibus_cmd_payload_last_i	Input	1	Indicates the last transfer in a write burst.
ibus_rsp_valid_o	Output	1	The manager is signaling the required read data.
ibus_rsp_payload_last_o	Output	1	Indicates the last transfer in a read burst.
ibus_rsp_payload_data_o	Output	DATA_WIDTH_A	Read data
ibus_rsp_payload_error_o	Output	1	Indicates the status of the read transfer. When HIGH, this signal indicates a successful transfer. When LOW, it indicates a failed transfer.
Port DATA			
dbus_cmd_valid_i	Input	1	Indicates that the manager is signaling valid write/read address and control information.
dbus_cmd_ready_o	Output	1	Indicates that the target is ready to accept an address and associated control signals.
dbus_cmd_payload_wr_i	Input	1	Indicates the transfer direction. When HIGH, this signal indicates a write transfer. When LOW, it indicates a read transfer. It has the same timing as that of the address signals. However, it does not remain constant throughout a burst transfer.
dbus_cmd_payload_uncached_i	Input	1	Indicates if the transfer is a cache operation. When HIGH, this signal indicates a cache transfer. When LOW, it indicates an uncached transfer.
dbus_cmd_payload_address_i	Input	32	Address
dbus_cmd_payload_data_i	Input	DATA_WIDTH_B	The write data bus transfers data from the manager to the target during write operations.
dbus_cmd_payload_mask_i	Input	DATA_WIDTH_B/8	Indicates the byte lanes that hold valid data. There is one write mask bit for every eight bits of the write data bus.

Pin Name	Direction	Width (bits)	Description
dbus_cmd_payload_size_i	Input	3	Indicates the size of the transfer, which is typically byte, halfword, or word. 3'b101 indicates an 8-word burst transfer.
dbus_cmd_payload_last_i	Input	1	Indicates the last transfer in a write burst.
dbus_cmd_payload_exclusive	Input	1	Indicates ATOMIC load or ATOMIC store.
dbus_cmd_payload_id	Input	4	Indicates the write ID.
dbus_rsp_valid_o	Output	1	The target signals the required read data. When ATOMIC is enabled, this also indicates a write is done.
dbus_rsp_payload_last_o	Output	1	Indicates the last transfer in a read burst. When ATOMIC is enabled, this also indicates the last write is done.
dbus_rsp_payload_data_o	Output	DATA_WIDTH_B	Read data
dbus_rsp_payload_error_o	Output	1	Indicates the status of the read transfer. When High, this signal indicates a successful transfer. When Low, it indicates a failed transfer.
dbus_rsp_payload_exclusive	Output	1	Indicates whether the ATOMIC store succeeds or not.
dbus_inv_payload_fragment_address	Output	32	The address might be modified by the FIFO interface.
dbus_inv_payload_fragment_enable	Output	1	Indicates there is a write from the FIFO interface.
dbus_inv_payload_last	Output	1	Indicates the last write from the FIFO interface.
dbus_inv_ready	Input	1	Indicates the CPU cache is ready to receive the invalidated information.
dbus_inv_valid	Output	1	Sends an invalidate request to the CPU cache.
dbus_ack_valid	Input	1	Indicates the CPU has flushed the cache for the previous invalidate request.
dbus_ack_ready	Output	1	Indicates the TCM IP is ready to receive ack signals.
dbus_ack_payload_fragment_hit	Input	1	Indicates the previous invalidate address hit in the cache.
dbus_ack_payload_last	Input	1	Indicates the last ack transfer.
dbus_sync_valid	Output	1	Indicates the previous write has been informed to all processors.
dbus_sync_ready	Input	1	Indicates the CPU is ready to receive sync signals.
dbus_sync_error	Output	1	Indicates a write error for the previous write.
dbus_sync_id	Output	4	Indicate the previous write ID.
FIFO Streamer			
fifo_clk_i	Input	1	FIFO clock source
fifo_wr_en_i	Input	1	FIFO write enable
fifo_wr_data_i	Input	8	FIFO write data
fifo_interface_en_i	Input	1	FIFO enable
fifo_address_rst_i	Input	1	FIFO address reset
fifo_full_o	Output	1	FIFO full status output flag

2.4.1. TCM Timing Diagrams

Figure 2.2 to Figure 2.4 show the timing diagrams for some configurations of the TCM. Target INSTR Port is given as an example, but DATA Port should have similar timing diagrams. The FIFO interface diagram is shown in Figure 2.5.

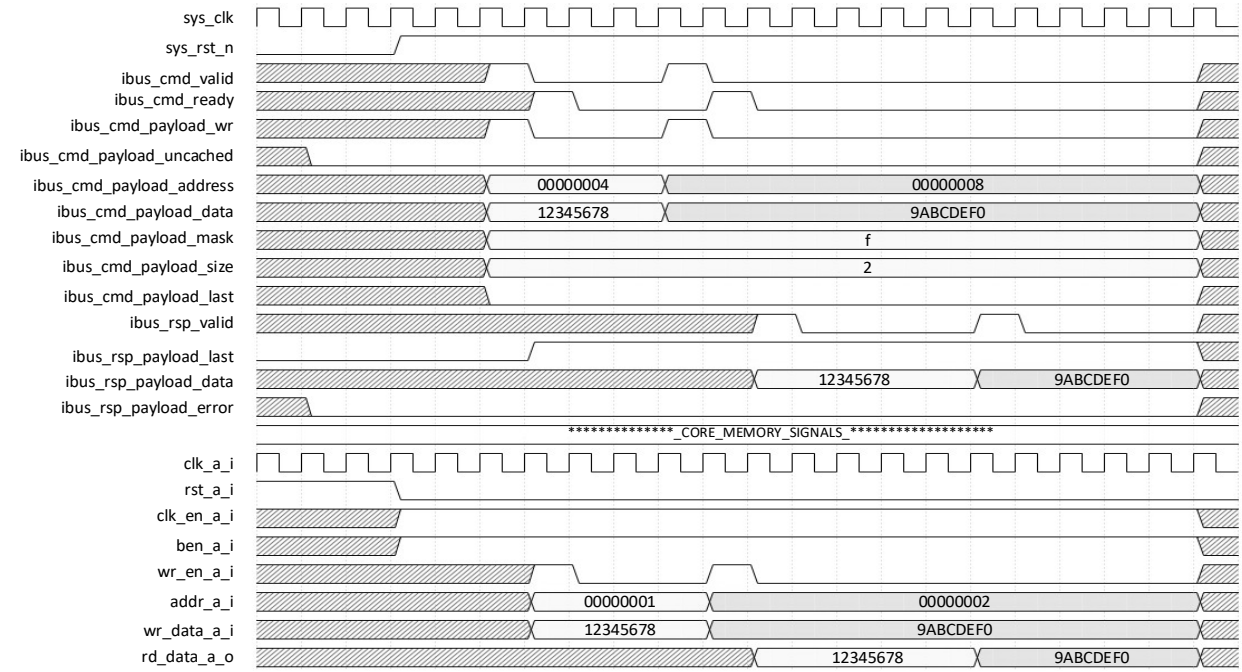

Figure 2.2. Write-Read Operation

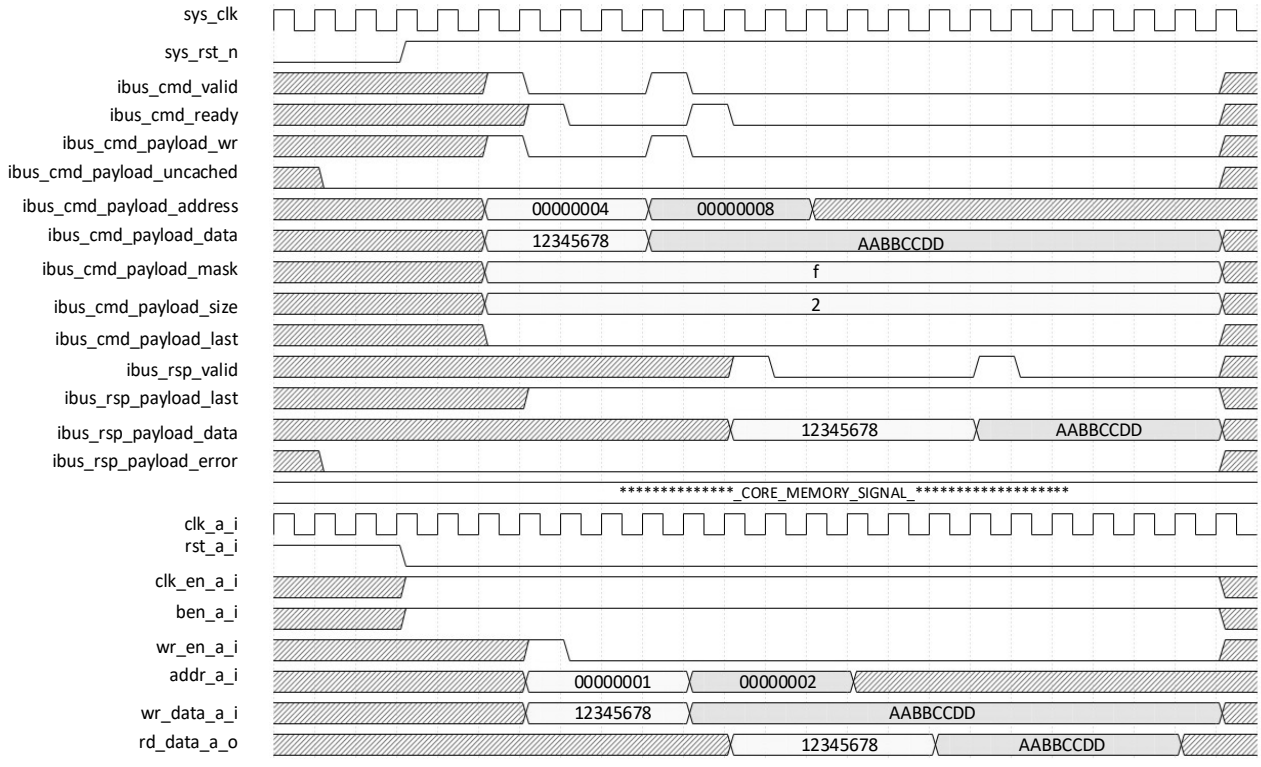

Figure 2.3. Sequential Write/Read

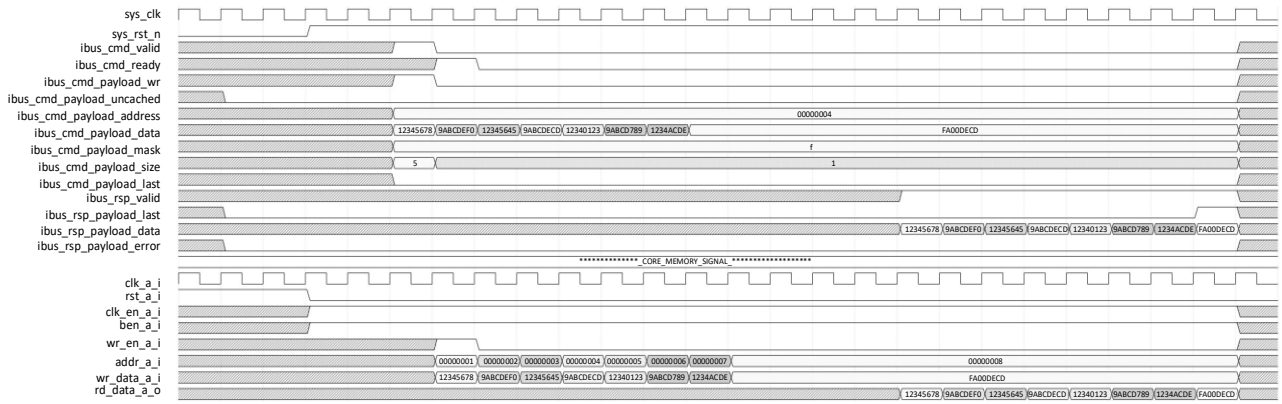

Figure 2.4. Bulk Write-Bulk Read Operation

Figure 2.5 shows the FIFO interface timing diagram. When using the FIFO, you can safely extend the fifo_wr_en_i and fifo_wr_data_i signals even when the fifo_full_o signal has been asserted, but those signals are ignored. Further write transactions are also ignored when fifo_full_o is asserted.

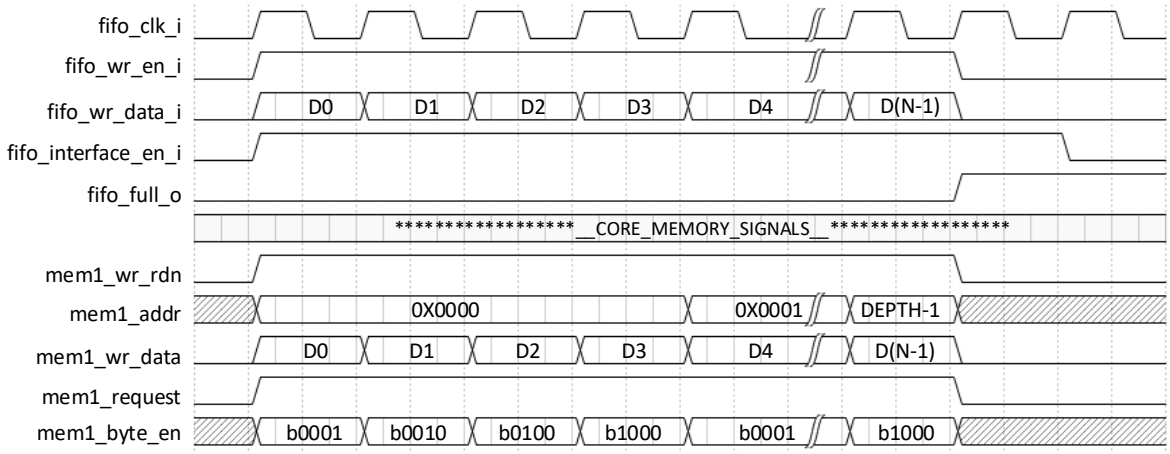

Figure 2.5. FIFO Interface Timing Diagram

2.5. User Accessible Parameters

- **Supported Families**
  - Certus-N2
  - Lattice Avant
  - MachXO5-NX
  - CrossLinkU-NX
  - CrossLink-NX
  - CertusPro-NX
  - Certus-NX
- **Supported Platforms**
  - Lattice Nexus 2
  - Lattice Avant
  - Lattice Nexus
- **Supported Bus Interfaces**
  - Single-Port Memory
  - Dual-Port Memory

- **Supported Primitives**
  - EBR
  - LRAM
  - Distributed RAM

2.6. Unaligned Access Feature

The Large RAM Module supports RISC-V compressed instruction chunks of data and shifts them. If RISC-V needs to read the upper 16 bits of data in some address, it is helpful to add support for shifting the upper 16 bits of output into the lower 16 bits, padding the upper bits with 0. Unaligned Read enables this feature. Table 2.6 shows possible combinations for unaligned read shifting. The unaligned read pins are shared with the ben_i ports for single-port large RAM. Given this, change the value of these signals carefully. The port functions as byte-enable during write-access and unaligned read during read-access.

Table 2.6. Unaligned Read Shift Function, Single-Port LRAM

ben_i[1:0]	ben_i[2] = 0	ben_i[2] = 1
00	DOx=x[31:0] (no shift)	DOx=x[31:0] (no shift)
01	DOx={8'b0,x[31:8]}	DOx={x[23:0],8'b0}
10	DOx={16'b0,x[31:16]}	DOx={x[15:0],16'b0}
11	DOx={24'b0,x[31:24]}	DOx={x[7:0],24'b0}

2.7. Initialization Format

The initialization file is an ASCII file, which you can create or edit using any ASCII editor. The Module/IP Block Wizard supports the following memory file formats:

- Binary File
- Hex File

The file name for the memory initialization file is <file_name>.mem. Each row includes the value to be stored in a particular memory location. The number of characters represents the number of bits for each address and the number of columns represents the width of the memory module.

The memory initialization can be static or dynamic. In case of static initialization, the memory values are stored in the bitstream. Dynamic initialization of memories involves memory values stored in the external flash and can be updated by user logic, knowing the EBR address locations. The size of the bitstream, either as bit or rbt file, is larger due to static values stored in them.

The initialization file can be used to preload memory contents. The first line of the initialization file corresponds to the start address of the TCM IP, and the subsequent lines of the initialization file correspond to the sequential addresses in the memory. There is no mechanism to skip over address ranges. If the address ranges are expected to be skipped, fill those lines with zeros. The initialization file does not have to be as large as the TCM, in which case the values of uninitialized regions are zeros.

If the amount of data in a memory file is larger than the address depth, there is a warning indicating this issue. Note that the program can still overflow even if there are no warnings at this stage. Always unify the address depth and memory space in the linker script in a C project and check the project compilation report.

Binary File

The binary file is a text file of 0s and 1s. The rows indicate the number of words, and the columns indicate the width of the memory.

Memory Size 20 x 32

```
00100000010000000010000001000000
00000001000000010000000100000001
00000010000000100000001000000010
00000011000000110000001100000011
00000100000001000000010000000100
00000101000001010000010100000101
00000110000001100000011000000110
00000111000001110000011100000111
00001000010010000000100001001000
00001001010010010000100101001001
00001010010010100000101001001010
00001011010010110000101101001011
00001100000011000000110000001100
00001101001011010000110100101101
00001110001111100000111000111110
00001111001111110000111100111111
00010000000100000001000000010000
00010001000100010001000100010001
00010010000100100001001000010010
00010011000100110001001100010011
```

Hex File

The hex file is a text file of hexadecimal characters arranged in a similar row-column arrangement. The number of rows in the file is the same as the number of address locations, with each row indicating the content of the memory location.

Memory Size 8 x 16

```
A001
0B03
1004
CE06
0007
040A
0017
02A4
```

2.8. ATOMIC Instruction Support

The TCM IP supports ATOMIC instructions, a feature designed for the RISC-V A extension. This support is available only in version 2.4.0 or later of the RX CPU IP core and is implemented in the Balanced and Advanced modes. With this feature, the previous Port S0 and Port S1 do not work in the same way. Therefore, they are renamed as Port INSTR and Port DATA respectively to avoid mixed usage of the two ports, as shown in Table 2.7. This ATOMIC feature is implemented on Port DATA, with extra signals that indicate whether the transaction is ATOMIC or not, and if the ATOMIC store succeeds or not. When ATOMIC is enabled, Port DATA should expose those extra signals to connect to the DATA port of the RX core.

Table 2.7. Port Name in GUI

Port Name in GUI	
During IP Configuration	**After IP Generation**
Port S0	Port INSTR
Port S1	Port DATA

Moreover, to receive the valid exclusive response, the rsp_valid signal is also asserted after a write transaction. To avoid unexpected sequential errors, the ATOMIC feature should only be enabled when connecting to version 2.4.0 or later of the RX CPU IP core in the Balanced or Advanced mode. It should be disabled when connecting to the RX core in Lite mode, or to an RX core version lower than 2.4.0.

3. Tightly-Coupled Memory IP Generation

This section provides information on how to generate the TCM IP using the Lattice Propel Builder software.

To generate the TCM IP module:

1. In the Lattice Propel Builder software, create a new design. Select the TCM package.
2. Enter the component name. Click **Next**, as shown in Figure 3.1.

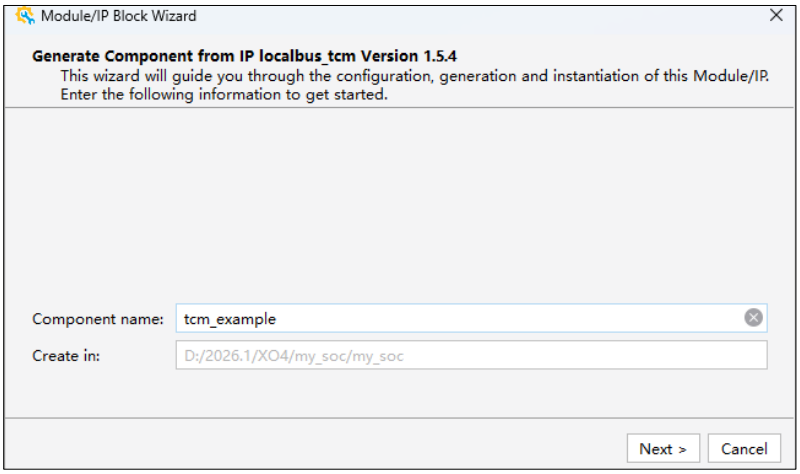

Figure 3.1. Entering Component Name

3. Configure the parameters as needed. Click **Generate** (Figure 3.2).

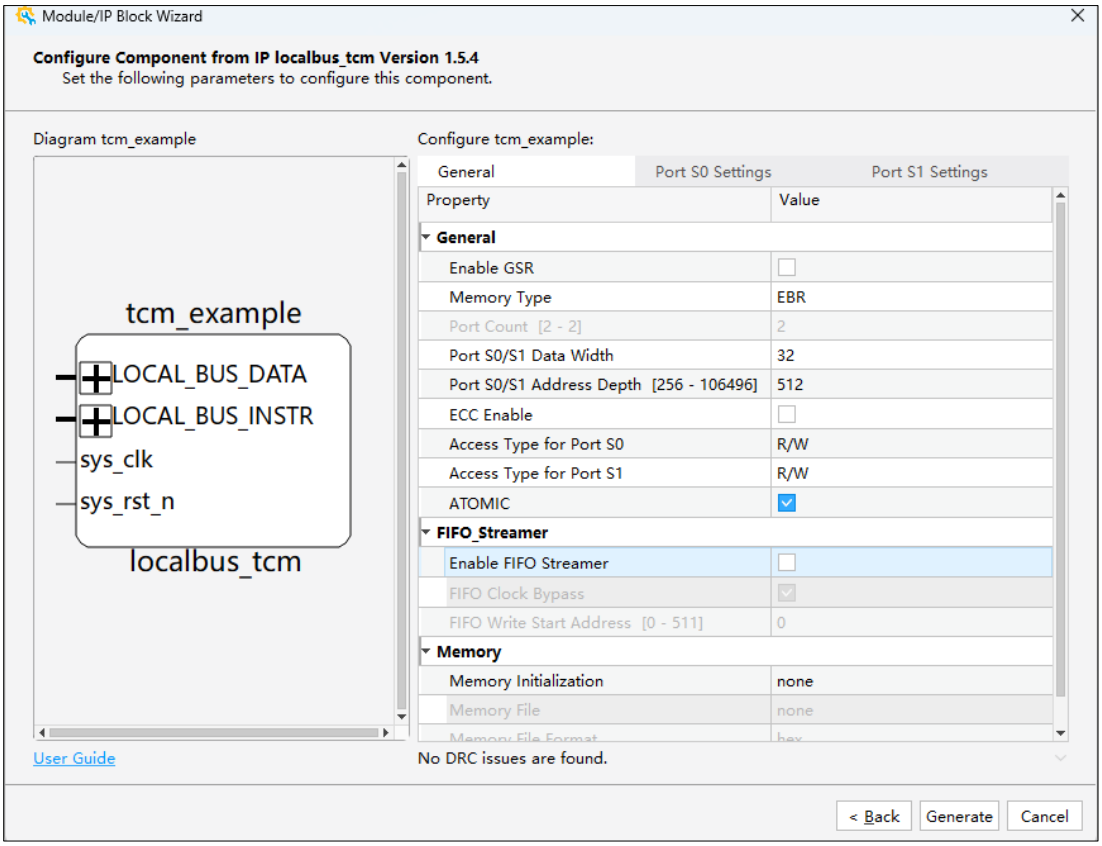

Figure 3.2. Configuring Parameters

4. Verify the information. Click **Finish** (Figure 3.3).

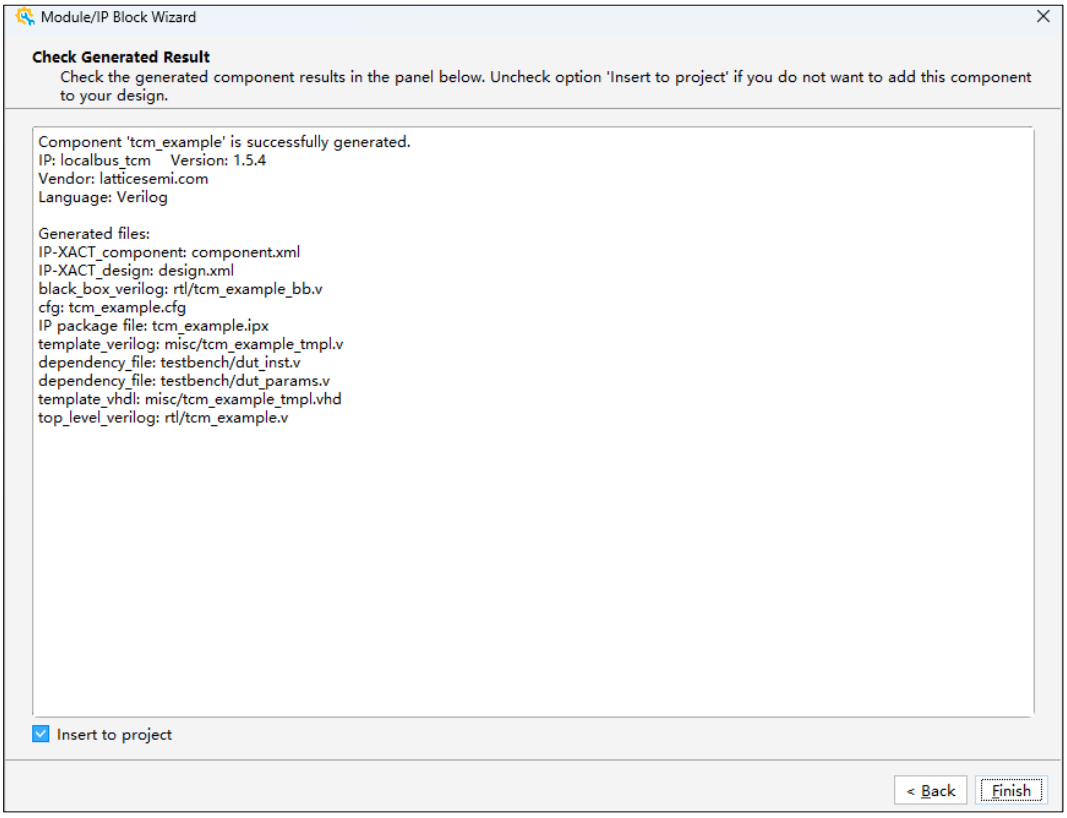

Figure 3.3. Verifying Results

5. Confirm or modify the module instance name. Click **OK** (Figure 3.4).

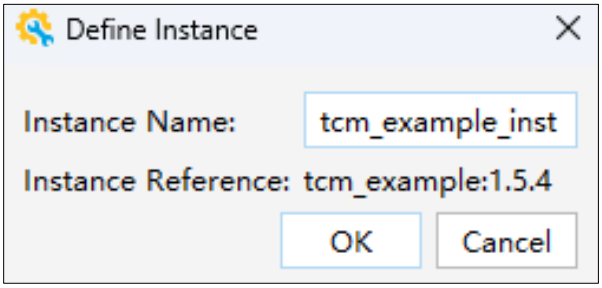

Figure 3.4. Specifying Instance Name

6. The TCM IP instance is successfully generated, as shown in Figure 3.5.

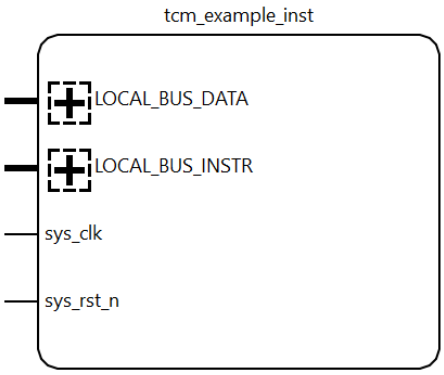

Figure 3.5. Generating TCM IP Instance

Appendix A. Resource Utilization

Table A.1. Resource Utilization in LFCPNX-100 Device

Configuration	LUTs	Registers	sysMEM EBRs	LRAM
MEMORY_TYPE = EBR, ADDR_DEPTH = 8192, DATA_WIDTH = 32, ATOMIC = 0	77	40	16	—
MEMORY_TYPE = EBR, ADDR_DEPTH = 8192, DATA_WIDTH = 32, ATOMIC = 1	78	41	16	—
MEMORY_TYPE = LRAM, ADDR_DEPTH = 8192, DATA_WIDTH = 32, ATOMIC = 1	82	41	—	1

Note: Resource utilization characteristics are generated using Radiant 2024.2 software.

Table A.2. Resource Utilization in LAV-AT-E70ES1 Device

Configuration	LUTs	Registers	sysMEM EBRs	LRAM
MEMORY_TYPE = EBR, ADDR_DEPTH = 8192, DATA_WIDTH = 32, ATOMIC = 0	74	40	8	—
MEMORY_TYPE = EBR, ADDR_DEPTH = 8192, DATA_WIDTH = 32, ATOMIC = 1	75	41	8	—

Note: Resource utilization characteristics are generated using Radiant 2024.2 software.

References

- Lattice Propel Design Environment web page
- Certus-N2 Family Devices web page
- Avant-E Family Devices web page
- Avant-G Family Devices web page
- Avant-X Family Devices web page
- MachXO5-NX Family Devices web page
- CrossLink-NX Family Devices web page
- CertusPro-NX Family Devices web page
- Certus-NX Family Devices web page
- Lattice Insights for Lattice Semiconductor Training Series and Learning Plans

Technical Support Assistance

Submit a technical support case through www.latticesemi.com/techsupport.

For frequently asked questions, refer to the Lattice Answer Database at www.latticesemi.com/Support/AnswerDatabase.

Revision History

Revision 1.0, IP v1.5.4, June 2026

Section	Change Summary
All	Production release.

www.latticesemi.com