LatticeMico On-Chip Memory Controller

The LatticeMico on-chip memory controller provides a slave interface to the WISHBONE bus master ports that allow them access to the Lattice Semiconductor FPGA embedded block RAMs (EBRs). The on-chip memory controller automatically instantiates the EBR using the parameterized module interface (PMI).

Version

This document describes the 3.4 version of the LatticeMico on-chip memory controller.

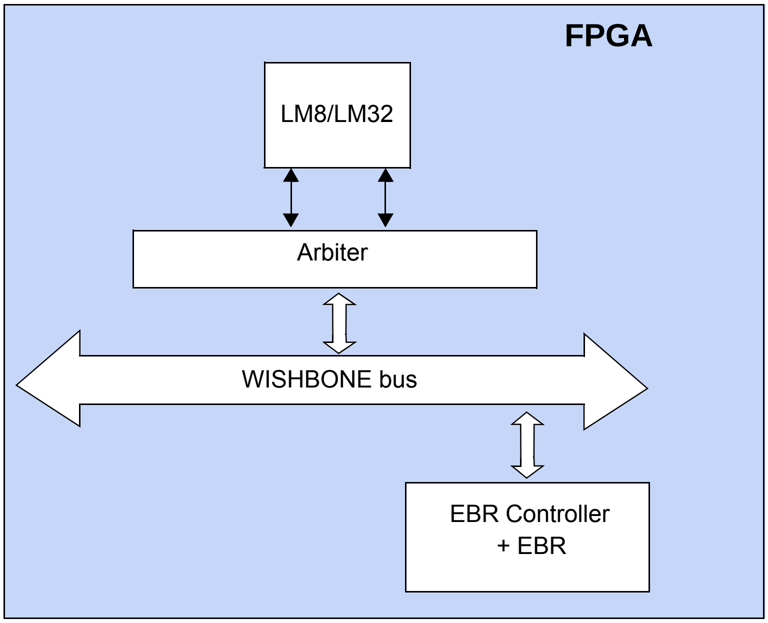
Features

The LatticeMico on-chip memory controller includes the following features:

- WISHBONE B.3 interface
- Support for Classic and Linear Incrementing Burst WISHBONE cycles.
- WISHBONE data bus width can be configured to be 8 or 32 bits wide.
- Support for 8-bit data transfer size for 8-bit and 32-bit WISHBONE data bus. Support for 16-bit and 32-bit data transfer size on 32-bit WISHBONE data bus.

As shown in Figure 1, the on-chip memory controller acts as a connector between the WISHBONE master ports and Lattice Semiconductor EBR memory. The master ports can write data to or read data from the EBR memory. When used, the memory controller instantiates the required EBRs.

Figure 1: LatticeMico On-Chip Memory Controller Slave Component



For additional details about the WISHBONE bus, refer to the *LatticeMico8 Processor Reference Manual* or the *LatticeMico32 Processor Reference Manual*.

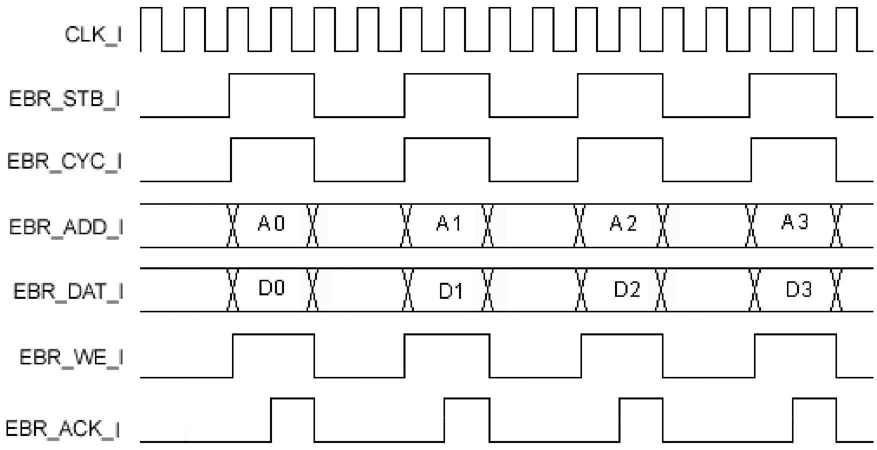
Functional Description

The LatticeMico on-chip memory controller provides Classic and Linear Incrementing Burst WISHBONE write and read bus operations, each of which supports 8-bit, 16-bit, and 32-bit read and write operations. (When the controller is configured for an 8-bit WISHBONE data bus, only 8-bit read/write operations are allowed.) EBRs are included with the memory controller. The maximum memory size or address range is limited by the FPGA's EBR block. The controller always completes the read and write in two cycles.

Figure 2 shows the port names and timing diagram of the on-chip memory controller in write mode.

The memory write occurs across two clock cycles.

Figure 2: On-Chip Memory Controller Timing Diagram for Write Mode



When the CTI signal value is 3’b000 and the WE signal is disabled, the master port reads one piece of data at a time, as shown in Figure 3. The width of the data transaction can be 8, 16, or 32 bits.

The memory read occurs across two clock cycles.

Figure 3: On-Chip Memory Controller Timing Diagram for Read Mode

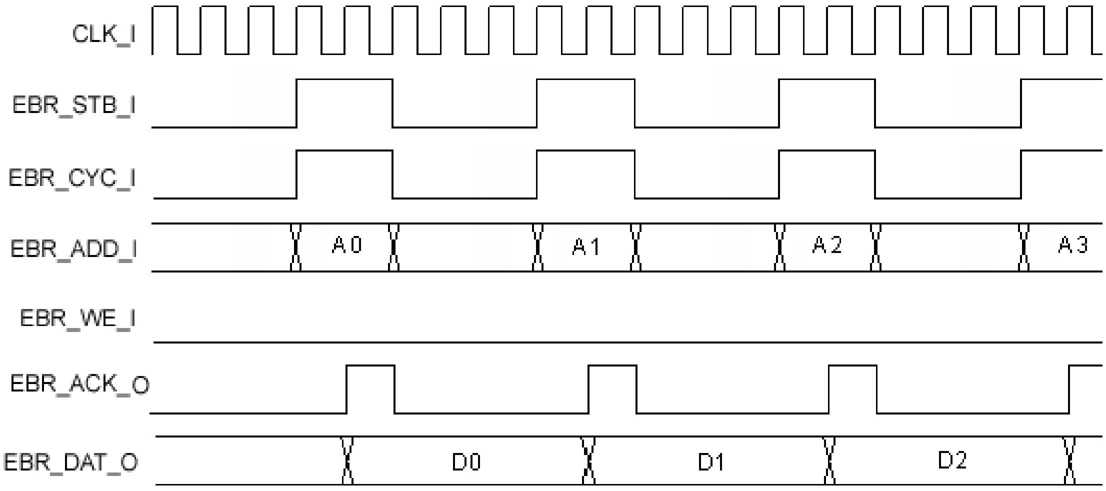


Figure 4 shows the port names and timing diagram of the on-chip memory controller in burst 4 write mode.

Figure 4: On-Chip Memory Controller Timing Diagram for Burst 4 Write Mode

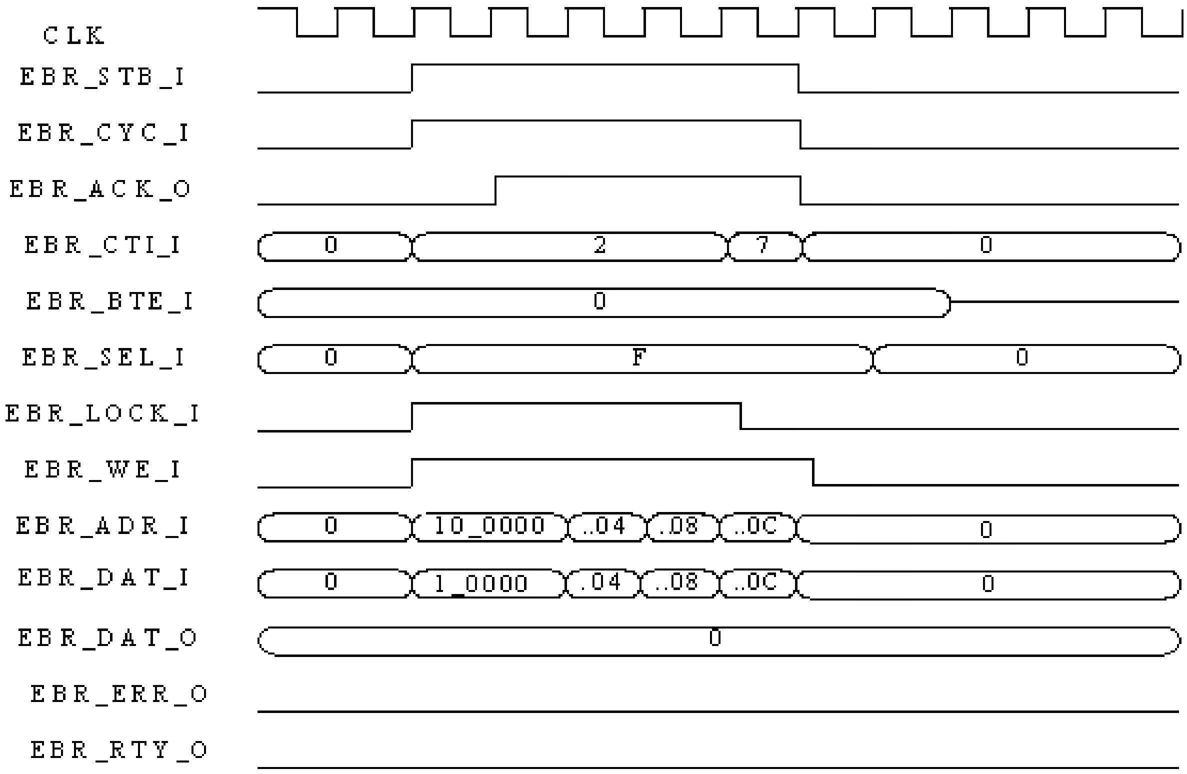
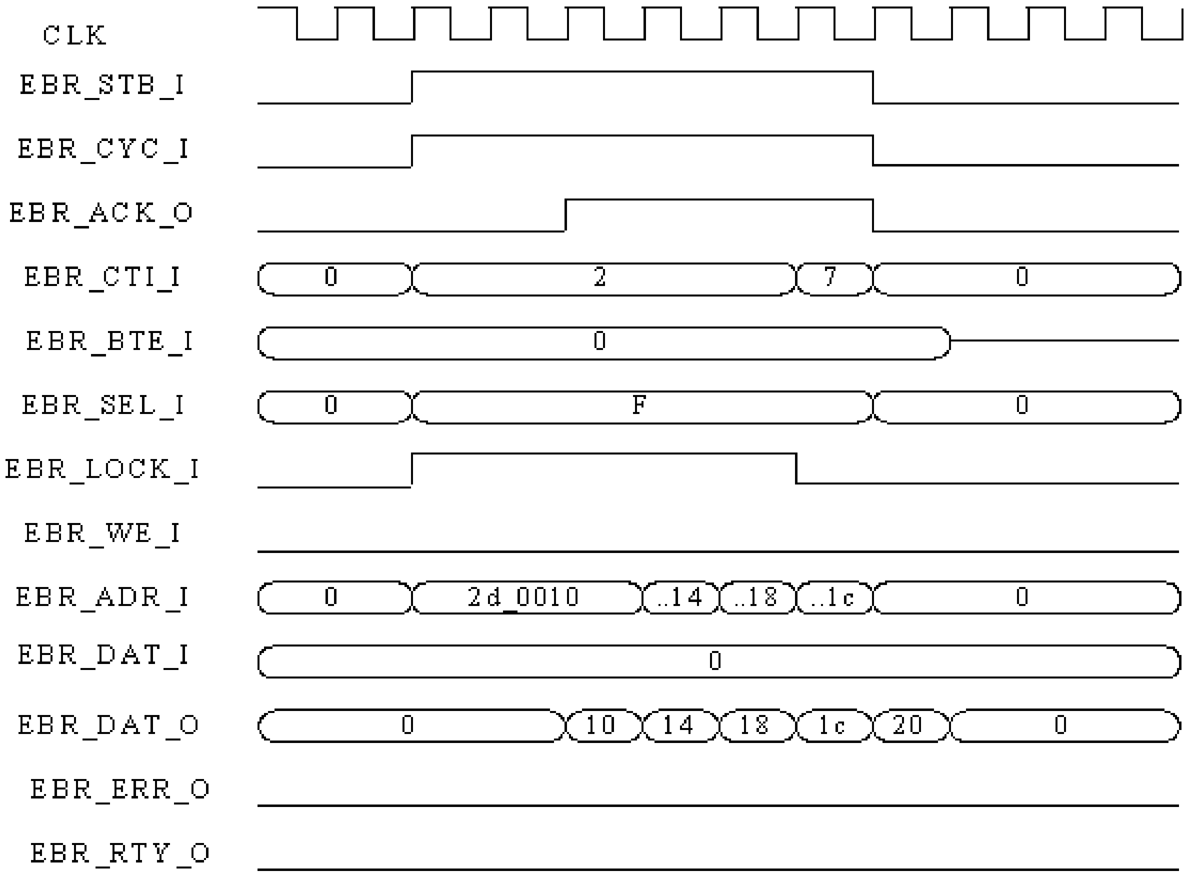


Figure 5 shows the port names and timing diagram of the on-chip memory controller in burst 4 read mode.

Figure 5: On-Chip Memory Controller Timing Diagram for Burst 4 Read Mode



Configuration

The following sections describe the graphical user interface (UI) parameters, the hardware description language (HDL) parameters, and the I/O ports that you can use to configure and operate the LatticeMico on-chip memory controller.

UI Parameters

Table 1 shows the UI parameters available for configuring the LatticeMico on-chip memory controller through the Mico System Builder (MSB) interface.

Table 1: On-Chip Memory Controller UI Parameters

Dialog Box Option	Description	Allowable Values	Default Value
Instance Name	Specifies the name of the on-chip memory controller instance.	Alphanumeric and underscores	ebr
Base Address	Specifies the base address for the device. The minimum boundary alignment is 0X00. The base address can be configured automatically in the LatticeMico Mico System builder.	0X00000000–0XFFFFFFFF If other components are included in the platform, the range of allowable values will vary.	0X00000000
Size of Memory (in Bytes)	Specifies the total size (depth x width) of the memory implemented using embedded block RAMs, in bytes (EBRs, in bits). The size is the limit of the EBR block in the selected FPGA.	0 – 72K	2048
Initialization File Name	Specifies the name of the memory initialization file. **Note**: The memory initialization file can be created from the software development environment. For a detailed description of the steps, refer to the “Deploying Your Software to LatticeMico8 Platform” section of the *LatticeMico8 Developer User Guide*” or the “Deploying to On-Chip Memory” section of the *LatticeMico32 Software Developer User Guide*.	Valid file name	None
File Format	Specifies the formatof the memory initialization file.	Hexadecimal, binary	Hexadecimal
WISHBONE Data Bus Width	Specifies the size of the WISHBONE data bus	8, 32	32

Note: For LatticeEC/ECP devices, 1 EBR = 1 kb

Note

You can use the ispLEVER Memory Initialization Tool to change the content of initialized RAM (EBR) after synthesis, placement, and routing. For more information on the ispLEVER Memory Initialization Tool, choose **Help > Project Navigator > User Interface > Memory Initialization Dialog Box**.

HDL Parameters

Table 2 lists the parameters that appear in the HDL.

Table 2: On-Chip Memory Controller HDL Parameters

Parameter Name	Description	Allowable Values
BASE_ADDRESS	Specifies the base address for the device.	0X00000000–0XFFFFFFFF
SIZE	Specifies the total size of the memory, in bits (depth x width), implemented using embedded block RAMs (EBRs). The size is the limit of the EBR block in the selected FPGA.	0X00000000–0XFFFFFFFF
INIT_FILE_NAME	Specifies the name of the memory initialization file.	Valid file name
INIT_FILE_FORMAT	Specifies the format of the memory initialization file.	Hexadecimal, binary
EBR_WB_DAT_WIDTH	Specifies the width of the WISHBONE data bus	8, 32

I/O Ports

Table 3 describes the input and output ports of the LatticeMico on-chip memory controller.

Table 3: On-Chip Memory Controller I/O Ports

Signal Name	Active	Direction	Initial State	Description
CLK_I	HIGH	I	0	Input clock signal. The clock is rising-edge active.
RST_I	HIGH	I	0	System reset signal
WISHBONE Slave Interface Signals				
EBR_ADDR_I [31:0]	—	I	0	Address input array. Address is generated by the master.
EBR_DAT_I[31:0]	—	I	0	Data input array, which is valid for a write request.
EBR_WE_I	—	I	0	Write enable signal. Value is 1 for a write and 0 for a read.
EBR_CYC_I	HIGH	I	0	Cycle input. When asserted, it indicates that a bus cycle is in progress.
EBR_STB_I	HIGH	I	0	Strobe input. When asserted, it indicates that the slave is selected.
EBR_SEL_I[3:0]	HIGH	I	0	Select input array, which indicates where the valid data is expected on the data bus. The read operation reads the bytes that are EBR_SEL high. The write operation writes only the selected bytes, leaving the others unchanged.

Table 3: On-Chip Memory Controller I/O Ports (Continued)

Signal Name	Active	Direction	Initial State	Description
EBR_CTI_I[2:0]	—	I	0	Cycle-type indication signal
EBR_BTE_I[1:0]	—	I	0	Burst-type extension signal
EBR_LOCK_I	HIGH	I	0	Slave bus locked
EBR_ACK_O	HIGH	O	0	Acknowledge output. When asserted, it indicates normal cycle termination.
EBR_DAT_O[31:0]	—	O	0	Data output array
EBR_ERR_O	HIGH	O	0	Slave error, which is always 0.
EBR_RTY_O	HIGH	O	0	Slave retry, which is always 0.

EBR Resource Utilization

The number of EBRs in the LatticeMico on-chip memory controller depends on the user-selected memory size.

- For LatticeECP2, LatticeXP2, LatticeECP2M, LatticeSC, and LatticeSCM devices, the number of EBRs is equal to:

$$\frac{\text{ceil (size in bytes)}}{2048}$$

- For LatticeEC, LatticeECP, and LatticeXP devices, the number of EBRs is equal to:

$$\frac{\text{ceil (size in bytes)}}{1024}$$

Software Support

The LatticeMico on-chip memory controller does not require associated software support. It can access a memory's location by treating it as a general-purpose read/write memory.

Revision History

Component Version	Description
1.0	Initial release.
3.0 (7.0 SP2)	Added support for an EBR size that is not a power of 2. Fixed the PMI memory's reset mode to synchronous mode.
3.1	Changed single read/write command from 3 cycles to 2 cycles. The first transfer of the write burst is also changed to 2 cycles.

Revision History (Continued)

Component Version	Description
3.2	Support added for sub-word (byte and halfword) Wishbone Burst Cycles
3.3 (8.1 SP1)	Support added for 8-bit data transfer size for 8-bit and 32-bit WISHBONE data bus. Support added for 16-bit and 32-bit data transfer size on 32-bit WISHBONE data bus.
3.4	Fixes address increment in byte burst transfers when WISHBONE bus data width is 8 bits.
3.4	Updated document with new corporate logo.