8N1 UART Tranceiver

Reference Design

FPGA-RD-02196-1.0

April 2020

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 and with all faults, and all risk associated with such information is entirely with Buyer. Buyer shall not rely on any data and performance specifications or parameters provided herein. 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. No Lattice products should be used in conjunction with mission- or safety-critical or any other application in which the failure of Lattice's product could create a situation where personal injury, death, severe property or environmental damage may occur. 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.

Contents

Figures

Tables

Acronyms in This Document

A list of acronyms used in this document.

Acronym	Definition
8N1	8 bit data, No parity, 1 stop bit
CPU	Central Processing Unit
DPRAM	Dual Port Random Access Memory
FIFO	First In, First Out
HDL	Hardware Description Language
IP	Intellectual Property
RTL	Register Transfer Level
Rx	Receiver
Tx	Transmitter
UART	Universal Asynchronous Receiver/Transmitter

1. Introduction

Universal Asynchronous Receiver/Transmitter (UART) performs serial-to-parallel and parallel to serial conversions on data characters received from a peripheral device or CPU. It operates at fixed 8-bit data, no parity, and 1 stop bit.

Both UART transmitter and receiver have a configurable FIFO option for better throughput. The FIFO buffer utilizes the Lattice Semiconductor DPRAM (Dual Port Random Access Memory) IP, whose depth can be configured using Lattice Radiant™ IP Catalog, Lattice Diamond® IPexpress, or Clarity Designer.

This reference design is implemented in Verilog. The Lattice Radiant and Lattice Diamond Place and Route tools integrated with the Synplify Pro® synthesis tool and Lattice Synthesis Engine are used for the implementation of the design. The design uses the iCE40 UltraPlus™ and MachXO2™ FPGAs and can be targeted to other Lattice FPGA devices.

2. Features

- 8-bit data, no parity, 1 stop bit (8N1)
- Optional FIFO buffer for both UART transmitter and receiver
- Configurable FIFO buffer depth
- Configurable baud rate
- Verilog RTL, testbench, and Aldec A-HDL script for simulation

3. Functional Description

3.1. Functional Block Diagram

Figure 3.1 shows an overview of the 8N1 UART Tranceiver reference design, which has a FIFO option for both transmitter and receiver. The UART Tranceiver Module connects to the User Module for byte data interface and to the Host Processer for serial data interface. The User Module may be internally configured (on the same FPGA device) or externally configured (on another device) connected to the UART Tranceiver Module.

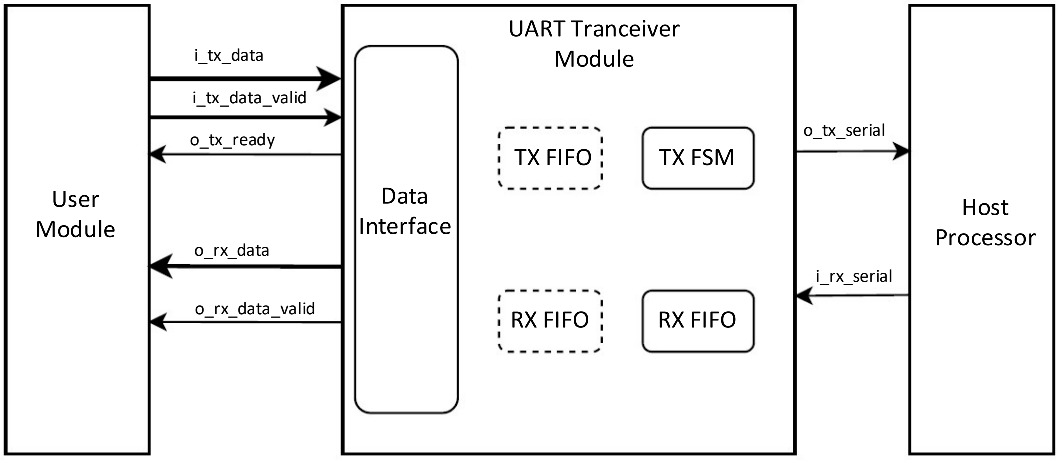


Figure 3.1. Functional Block Diagram

3.2. Signal Descriptions

Table 3.1 provides the description of each signal indicated in Figure 3.1 relative to the UART Tranceiver Module.

Table 3.1. Signal Descriptions

Signal Name	I/O Type	Width	Description
i_tx_data	Input	8	Byte data to transmit
i_tx_data_valid	Input	1	Transmit byte data valid
o_tx_serial	Output	1	Serial data to transmit
o_tx_ready	Output	1	Transmit ready signal: Busy [0], Ready [1]
i_rx_serial	Input	1	Serial data received
o_rx_data	Output	8	Byte data received
o_rx_data_valid	Output	1	Received byte data valid
clk	Input	1	System clock
resetn	Input	1	Active low signal to reset

3.3. 8N1 UART Tranceiver Parameters

Table 3.2 shows the default value and description of each 8N1 UART Tranceiver parameter.

Table 3.2. 8N1 UART Tranceiver Parameters

Parameter	Default Value	Description
TX_FIFO_MODE	1	FIFO buffer for UART Tx: On [1] , Off [0]
RX_FIFO_MODE	0	FIFO buffer for UART Rx: On [1] , Off [0]
FIFO_DEPTH	512	Set the FIFO buffer depth equal to the DPRAM IP address depth
BAUDRATE	B9600	BAUDRATE parameter represents the divisor value computed as: Divisor = Clock Input/Baud Rate

3.4. Data Format

Figure 3.2 shows the serial data format for both Tx and Rx. It includes start bit, 8-bit data, and stop bit.

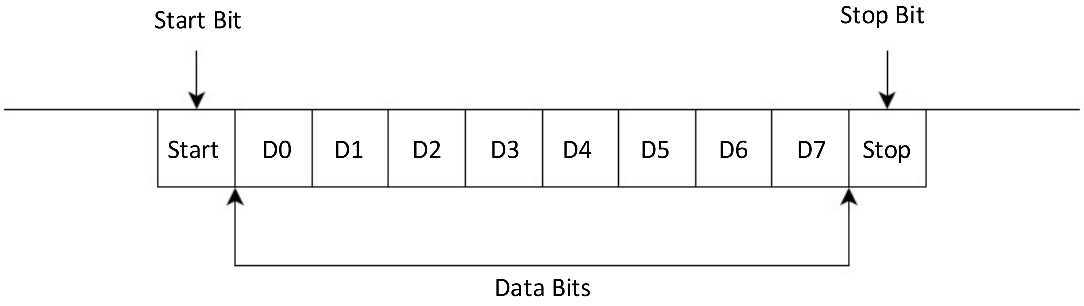


Figure 3.2. UART Data Format

3.5. UART Transmitter Timing

Figure 3.4 shows the timing diagram of byte write to an internal register (that is, DPRAM FIFO) and serial data output.

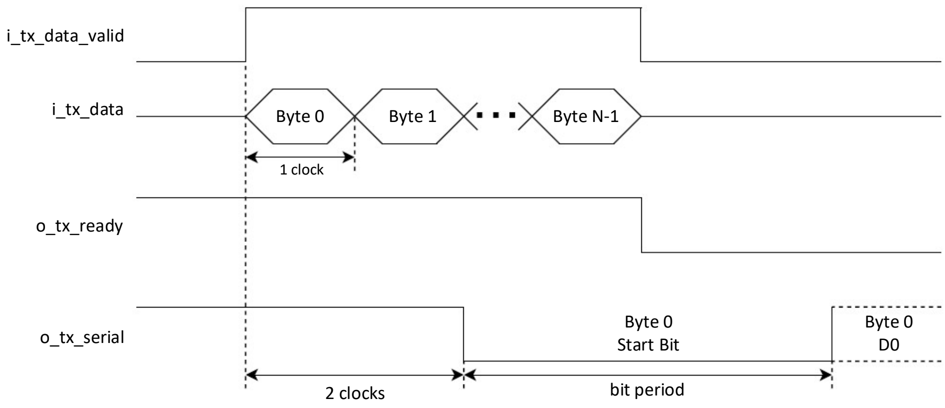


Figure 3.3. Transmitter Write Timing Diagram

3.6. UART Receiver Timing

Figure 3.4 shows the timing diagram for receiving serial data.

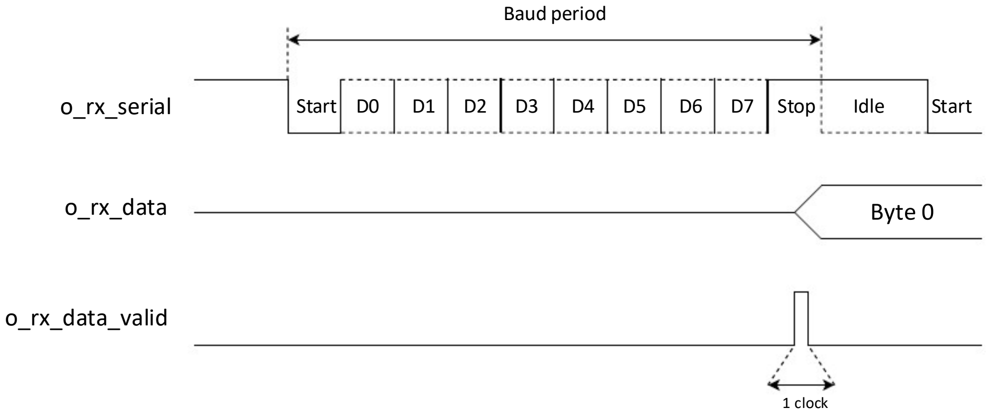


Figure 3.4. Receiver Read Timing Diagram

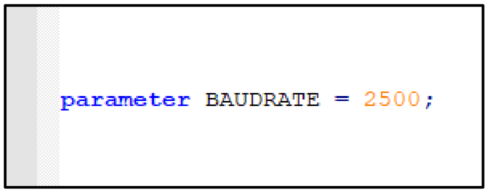
4. Baud Rate Configuration

The BAUDRATE parameter represents the value of the divisor to set the required actual baud rate. Table 4.1 shows BAUDRATE parameter string names and their equivalent divisor values at 12 MHz reference clock.

Table 4.1. Defined BAURATE Parameters

BAUDRATE	Divisor Value
B115200	104
B57600	208
B38400	313
B19200	625
B9600	1250
B4800	2500
B2400	5000
B1200	10000
B600	20000
B300	40000

To set the required baud rate with a different reference clock, use Divisor Value = Reference Clock/Baud Rate. For example, if the required baud rate is 9600 at 24 MHz reference clock, the Divisor Value is 2500.



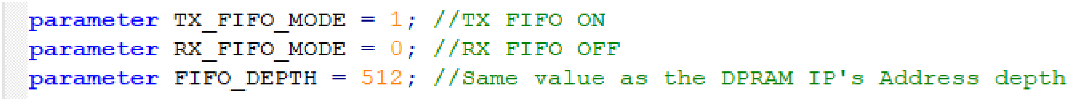
```
parameter BAUDRATE = 2500;
```

Figure 4.1. Parameter Settings for 9600 Baud Rate at 24 MHz Reference Clock

5. FIFO Buffer Configuration

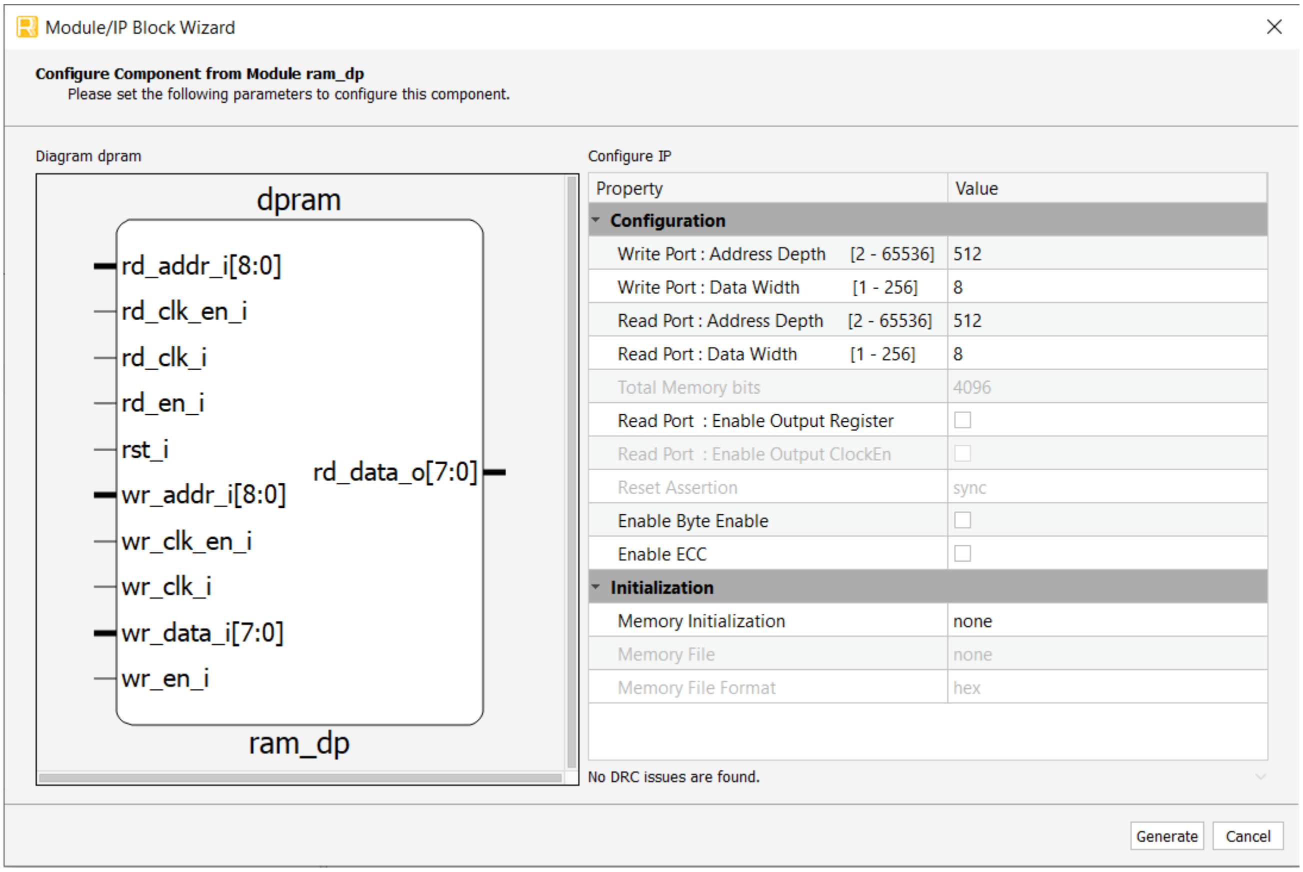
TX_FIFO_MODE, RX_FIFO_MODE, and FIFO_DEPTH are the parameters used for FIFO buffer configuration. TX_FIFO_MODE is used to enable or disable the synthesis of the FIFO buffer for the UART transmitter. RX_FIFO_MODE is used to enable or disable the synthesis of the FIFO buffer for the UART receiver. FIFO_DEPTH is used to set the buffer's depth.

Note: To properly set the buffer's depth, configure the DPRAM IP's address depth as well using IP Catalog, IPexpress, or Clarity Designer.



```
parameter TX_FIFO_MODE = 1; //TX FIFO ON
parameter RX_FIFO_MODE = 0; //RX FIFO OFF
parameter FIFO_DEPTH = 512; //Same value as the DPRAM IP's Address depth
```

Figure 5.1. Sample Verilog FIFO Buffer Configuration



Configure Component from Module ram_dp
Please set the following parameters to configure this component.

Diagram dpram

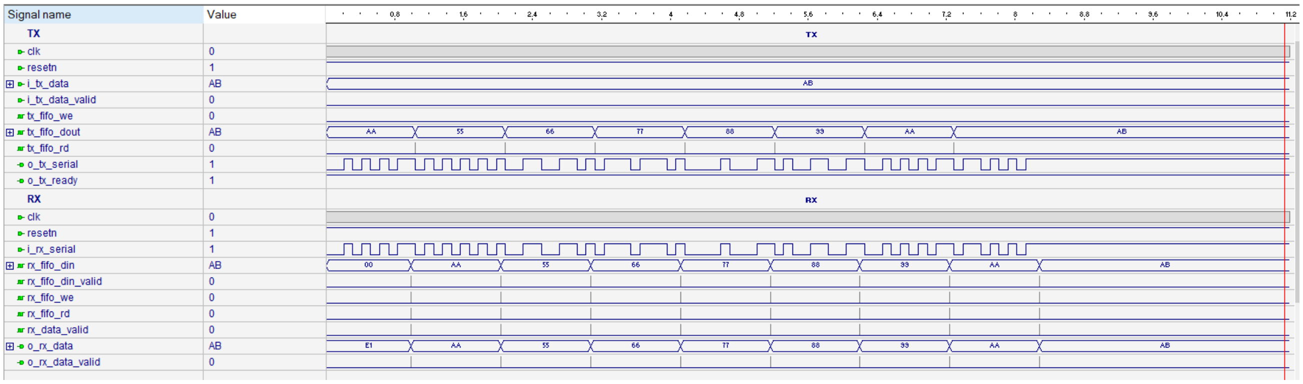
Configure IP

Property		Value
Configuration		
Write Port : Address Depth	[2 - 65536]	512
Write Port : Data Width	[1 - 256]	8
Read Port : Address Depth	[2 - 65536]	512
Read Port : Data Width	[1 - 256]	8
Total Memory bits		4096
Read Port : Enable Output Register		□
Read Port : Enable Output ClockEn		□
Reset Assertion		sync
Enable Byte Enable		□
Enable ECC		□
Initialization		
Memory Initialization		none
Memory File		none
Memory File Format		hex

No DRC issues are found.

Figure 5.2. DPRAM IP Configuration Window (Lattice Radiant)

6. HDL Simulation



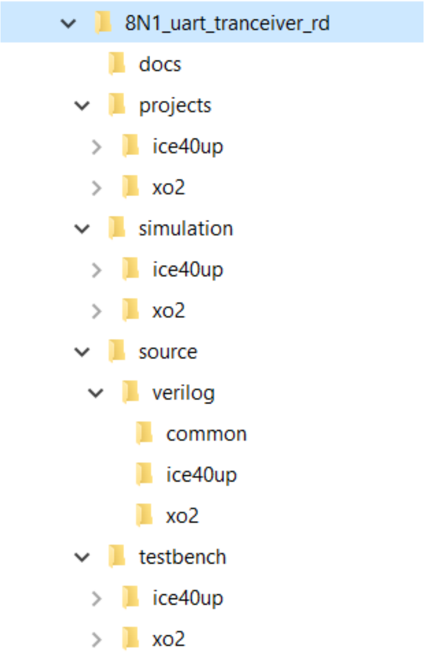
Signal name	Value
TX	
clk	0
resetn	1
i_tx_data	AB
i_tx_data_valid	0
tx_fifo_we	0
tx_fifo_dout	AB
tx_fifo_rd	0
o_tx_serial	1
o_tx_ready	1
RX	
clk	0
resetn	1
i_rx_serial	1
rx_fifo_din	AB
rx_fifo_din_valid	0
rx_fifo_we	0
rx_fifo_rd	0
rx_data_valid	0
o_rx_data	AB
o_rx_data_valid	0

Figure 6.1. Tx FIFO = On and Rx FIFO = On

7. Packaged Design

The reference design folder (*8N1_uart_tranceiver_rd*) contains five subfolders:

- Project – contains the Lattice Radiant project file (.rdf), project stategy file (.sty), IP instantiation file (.ipx); Diamond project file (.ldf), project strategy file (.sty); and IPexpress file (.ipx).
- Simulation – contains the simulation file (.do), which is used to run the RTL simulation on Aldec Active-HDL.
- Source – contains the UART Tranceiver RTL file (*8N1_uart_tranceiver.v*) along with other source files.
- Testbench – contains the top-level testbench file (uart_tb.v).



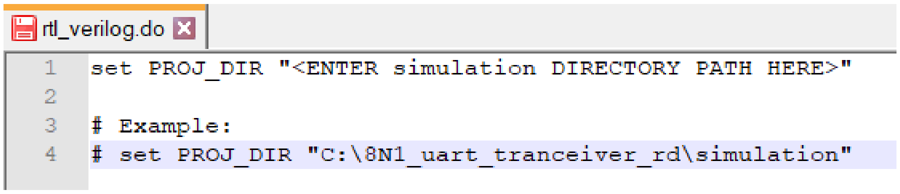
```
8N1_uart_tranceiver_rd
  docs
  projects
    ice40up
    xo2
  simulation
    ice40up
    xo2
  source
    verilog
      common
      ice40up
      xo2
  testbench
    ice40up
    xo2
```

Figure 7.1. Packaged Design Directory Structure

7.1. Using the Simulation File (.do)

To use the simulation file:

1. Open the .do file on a text editor and replace the text **<ENTER simulation DIRECTORY PATH HERE>** from Line 1 with the directory path of the simulation file. An example is shown on Line 4 of the file in Figure 7.2.



```
1  set PROJ_DIR "<ENTER simulation DIRECTORY PATH HERE>"
2
3  # Example:
4  # set PROJ_DIR "C:\8N1_uart_tranceiver_rd\simulation"
```

Figure 7.2. Changing the Simulation Directory

2. Run the file on Aldec Active-HDL by selecting **Tools > Execute macro**.

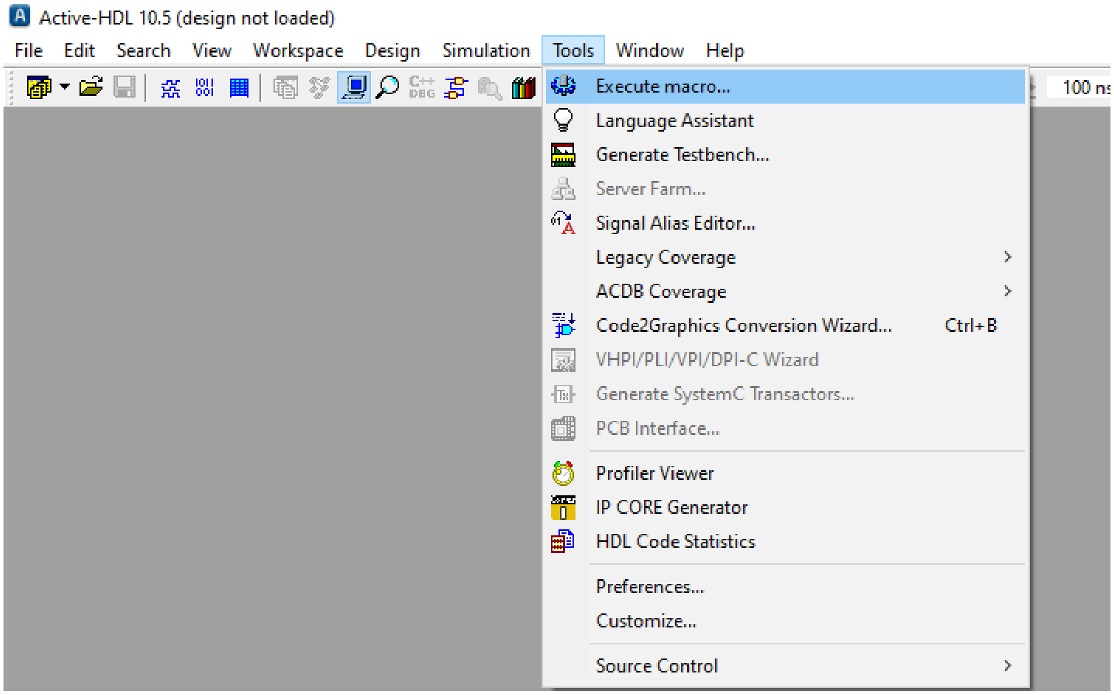


Figure 7.3. Running the Simulation File

8. Implementation

This design is implemented in Verilog. When using this design in a different device, density, speed or grade, performance and utilization may vary.

Table 8.1. Performance Resource Utilization

Family	Language	Tx FIFO (512 depth)	Rx FIFO (512 depth)	Utilization (LUTs)	I/O
iCE40 UltraPlus	Verilog - Synplify	Off	Off	135	23
		On	On	231	23
	Verilog - LSE	Off	Off	142	23
		On	On	223	23
MachXO2	Verilog - Synplify	Off	Off	83	23
		On	On	165	23
	Verilog - LSE	Off	Off	101	23
		On	On	186	23

References

For more information, refer to the following documents:

- iCE40 UltraPlus Family Data Sheet (FPGA-DS-02008)
- Memory Usage Guide for iCE40 Devices (FPGA-TN-02002)
- MachXO2 Family Data Sheet (FPGA-DS-02056)
- Memory Usage Guide for MachXO2 Devices (FPGA-TN-02159)

Technical Support Assistance

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

Revision History

Revision 1.0, April 2020

Section	Change Summary
All	Initial release.

LATTICE
SEMICONDUCTOR®

www.latticesemi.com