AN 024: Using the DDR Read/Write Example Design (T120 BGA576)

AN024-v1.0
July 2020
www.efinixinc.com

Contents

Introduction 3

Required Hardware 3

Set Up the Hardware 4

Program the Trion® T120 BGA576 Development Board 5

View the Design in the Debugger 6

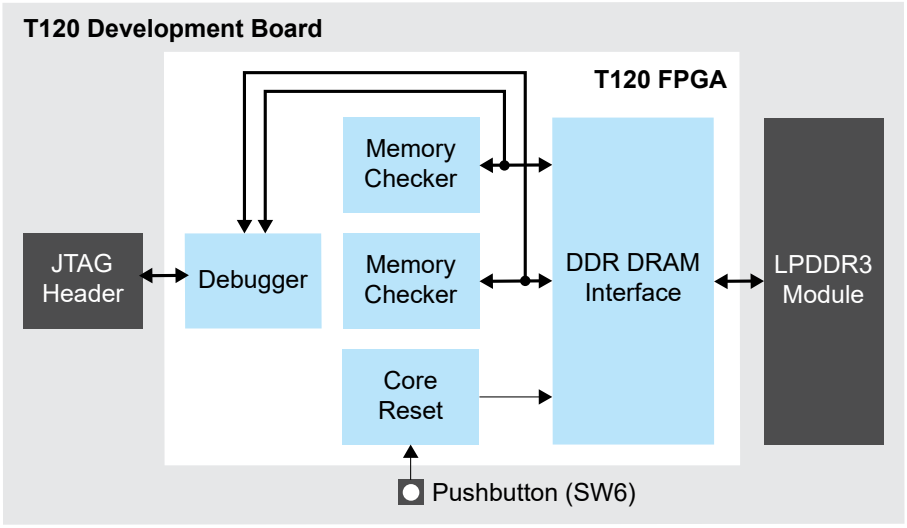
Revision History 7

Introduction

The Trion® T120 BGA576 Development Kit features a T120 FPGA in a 576-ball BGA package. The kit is ideal for exploring design and protoptyping with the T120 FPGA's MIPI CSI-2 and DDR DRAM interfaces.

This application note describes a simple design that demonstrates a project using the T120F576I4 FPGA's DDR interface. It performs a simple read/write test for the LPDDR3 memory chip on the Trion® T120 BGA576 Development Board.

Figure 1: DDR DRAM Read/Write Example Design



Required Hardware

The example design uses the following hardware from the Trion® T120 BGA576 Development Kit:

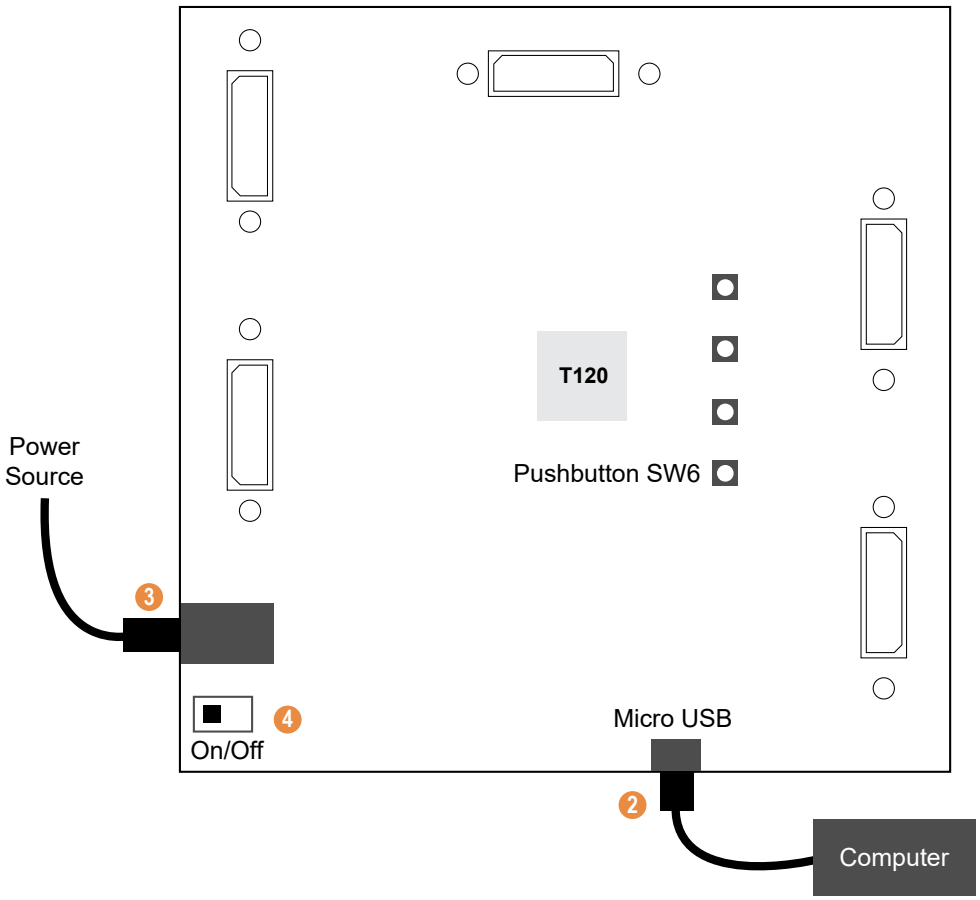
- Trion® T120 BGA576 Development Board
- Micro-USB cable
- 12-V power supply

Additionally, the design uses requires a laptop or personal computer with the Efinity® software installed.

Set Up the Hardware

The following figure shows the hardware setup steps:

Figure 2: Hardware Setup



Important: Make sure that the Trion® T120 BGA576 Development Board is turned off before connecting any cards or cables.

1. If you have not already done so, attach standoffs to the Trion® T120 BGA576 Development Board
2. Connect a USB cable to the board and to your computer.
3. Connect the power supply to the board and a power source.
4. Turn on the board using the power switch.

Learn more: Refer to the T120 BGA576 Development Kit User Guide for information on the components available on the Trion® T120 BGA576 Development Board.

Program the Trion® T120 BGA576 Development Board

The Trion® T120 BGA576 Development Board ships pre-loaded with an HDMI video streaming design. To use the DDR read/write example design, you must program the design into the board.

Note: You can use active, passive, or JTAG programming.

1. Download the example design file, **ddr_example_design_t120f576_devkit-v<*version*>.zip** from the Support Center.
2. Unzip the file into your working directory.
3. Open the project (**t120f576_lpddr3_x32.xml**) in the Efinity software and review it.

   Although the project targets the C4 FPGA while the board has an I4, the speed grades are the same, so the project works for both.
4. Use the Efinity® Programmer to download the bitstream file to your board. The example includes a bitstream file, **t120f576_lpddr3_x32.bit**

Learn more: Documentation on how to use the Efinity® software is available in the Support Center.

View the Design in the Debugger

The example design includes a Virtual I/O debug core so that you can monitor the signals from the DDR DRAM interface. To use the Debugger:

1. Run the Efinity software.
2. Open the Debugger. The tool opens to the Debug perspective. The `VIO_0` debug core sources and probes displays.
3. Click Connect Debugger. The `VIO_0` core captures data and displays it.
4. To introduce errors into the system to see error messages in the Debugger, reset the PLL by changing the `pll_rstni` source's value to 0.
5. To reset the system, change the `check_rstn` source's value to 0 or press pushbutton SW6.

Table 1: Virtual I/O Debug Core Sources and Probes.

Signal Name	Type	Description
rdata_exp0	Probe	Expected value written to the DDR module through the AXI target 0 interface.
rdata_det0	Probe	Value read from the DDR module through the AXI target 0 interface.
rdata_exp1	Probe	Expected value written to the DDR module through the AXI target 1 interface.
rdata_det1	Probe	Value read from the DDR module through the AXI target 1 interface.
axi0_states	Probe	Shows the state machine status for user logic connected to AXI target 0.
axi1_states	Probe	Shows the state machine status for user logic connected to AXI target 1.
check_rstn	Source	1: Design operating normally; 0: Reset the system and start the test again.
pll_rstni	Source	1: Design operating normally. 0: Reset the PLL.

The `rdata_exp0`, `rdata_exp1`, `rdata_det0`, and `rdata_det1` signals display 0s if the values read from the memory match the expected value. Otherwise, they show the error value read back and the expected value.

Revision History

Table 2: Document Revision History

Date	Version	Description
July 2020	1.0	Initial release.