One-Time Programmable (ALTOTP) Megafunction User Guide

© November 2009 UG-01059-1.0

This user guide describes the features and behavior of the ALTOTP megafunction. In addition, this user guide briefly describes the top-level OTP fuse block in relation to the fuse programming, fuse pre-loading, and fuse reading processes.

Altera provides a one-time programmable (OTP) feature through the dedicated OTP fuse block of the HardCopy® III and HardCopy IV devices. The OTP fuse block is a 128-bit non-volatile memory that can only be programmed once, and data is stored permanently. The OTP feature is commonly used for customizing a device with unique codes. Some examples of OTP applications are the storing of manufacturing codes and unique user serial numbers. The OTP fuse block can be read through the ALTOTP megafunction. The ALTOTP megafunction provides an interface to the OTP fuse block of the HardCopy III and HardCopy IV devices.

Stratix® III and Stratix IV devices do not have dedicated OTP fuse block. However, you can compile and simulate the ALTOTP megafunction on these devices for the purpose of prototyping a HardCopy device migration.

For an introduction to megafunctions and more information about how to create them, refer to *Megafunction Overview User Guide* on the Altera website.

This user guide contains the following sections:

Fuse Processes

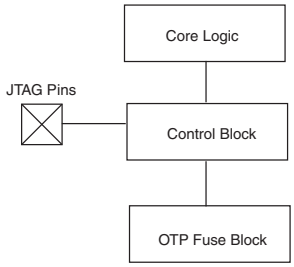
There are three phases to fuse processing:

- Phase 1: Fuse Programming
- Phase 2: Fuse Pre-Loading
- Phase 3: Fuse Reading

This section describes the interactions between the top-level OTP fuse block and the block components that are related to each phase of the fuse processes.

Figure 1 shows the top-level OTP fuse blocks and the relevant component blocks.

Figure 1. Top-Level OTP Fuse Block and Related Component Blocks



Phase 1: Fuse Programming

Fuse programming is a process of writing data into the OTP fuse block, and it involves interactions between the JTAG Pins, control block, and OTP fuse block. Data is programmed through the JTAG Pins and loaded into the OTP fuse block using the JTAG TAP controller that resides in the control block. This phase can only be executed during user mode.

☞ Data is programmed into the OTP fuse block through the JTAG interface only and not through the ALTOTP megafunction.

Phase 2: Fuse Pre-Loading

Fuse pre-loading involves interactions between the control block and the OTP fuse block. In this phase, data is loaded from the OTP fuse block into the storage register that resides in the control block. This phase is carried out automatically during initialization, and the content of the OTP fuse block is only available after you have reset the device.

Phase 3: Fuse Reading

This section describes the fuse reading through the core logic that is accomplished using the ALTOTP megafunction.

Fuse reading involves the following processes:

- Loading Operation
  The pre-loaded data is transferred parallelly from the storage register to the access register in the control block.
- Shifting Operation
  The JTAG pins or core logic read the loaded data serially from the access register.

When the loading operation is completed, the data is retained in the access register even after the shifting operation is activated to read out the data. For more information about loading and shifting operations, refer to `otp_shiftnld` port description of Table 3 on page 6.

The access register serves as a communication interface between the control block and core logic, as well as the JTAG interface. When accessing the access register, the JTAG interface has a higher priority over the core logic.

Loading and Shifting Operations: An Example of a Complete 128-bit Fuse Reading

The waveforms in this section illustrate the OTP loading and shifting operations, and the behavior of the `otp_dout` signal in response to the `otp_clken` and `otp_shiftnld` signals. In this example, the OTP data is initialized to 128'hF55…551.

The frequency that is used to illustrate the OTP loading and shifting operations does not reflect the supported frequency. The HardCopy III and HardCopy IV devices only support up to approximately 100 MHz.

The OTP loading and shifting operations are illustrated in Figure 2 and Figure 3 on page 4, respectively.

Figure 2. OTP Loading

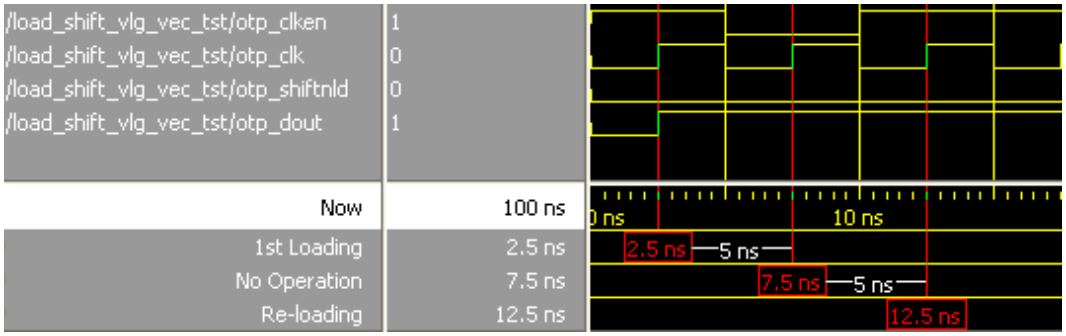


Table 1. Description of the OTP Loading in Figure 2

Time (ns)	Description
2.5	First loading operation is activated. The 128-bit OTP data from the storage register is parallelly loaded into the access register. The loading operation requires only one clock cycle to complete. The LSB of the OTP data is shown at the `otp_dout` port even before the shifting operation is activated.
7.5	No operation (`otp_clken` is **LOW**). The `otp_dout` signal remains unchanged.
12.5	Reloading operation is activated. The `otp_dout` signal keeps showing the LSB of the OTP data if the shifting operation is not activated.

Figure 3. OTP Shifting (Reading)

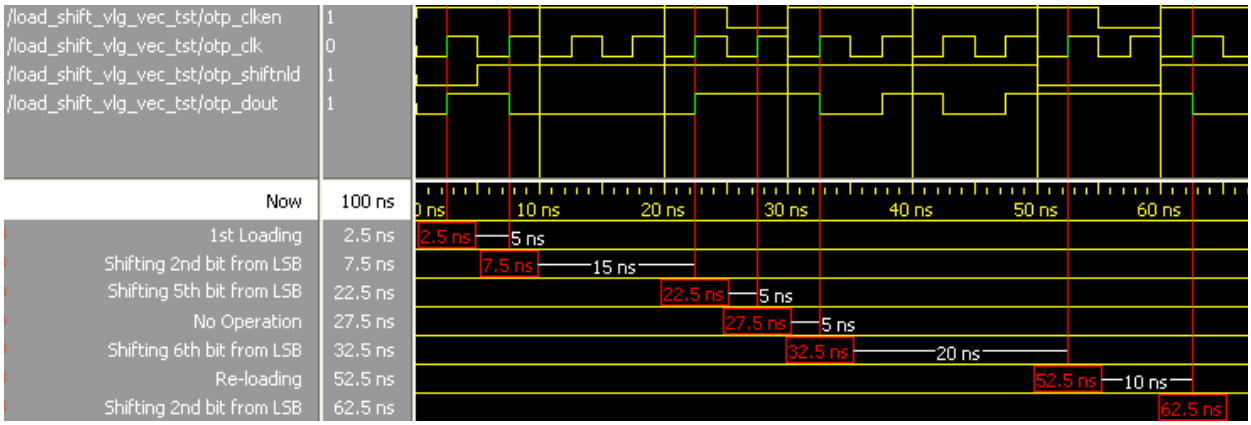


Table 2. Description of the OTP Shifting Operation in Figure 3

Time (ns)	Description
2.5	First loading is activated. The LSB of the OTP data (1) is shown at the otp_dout port.
7.5	Shifting operation is activated. The otp_dout port shows the second bit of the OTP data (0) from the LSB. Note that the first shifting operation always shifts out the second bit of the OTP data because the LSB of the OTP data (1) is shifted out during the loading operation.
7.5-22.5	Shifting operation continues.
27.5	Shifting operation stops at the fifth bit of the OTP data (1), as no operation is executed. (otp_clken port is **LOW**). The output of the otp_dout port remains unchanged.
32.5	Shifting operation continues. The otp_dout port shows the sixth bit of the OTP data (0).
52.5	Reloading operation occurs. The otp_dout port shows the LSB of the OTP data. When re-loading occurs, data is shifted from the beginning again.
62.5	Shifting operation is activated. The otp_dout port shows the second bit of the OTP data from the LSB.

If otp_shiftnld is held **HIGH** beyond 128 clock cycles (after all 128-bit OTP data is read), the otp_dout signal shows 0 and the value remains until the reloading operation occurs.

Design Example: Parallel Read of 128-bit OTP data

The ALTOTP megafunction only supports serially read OTP data. In applications that require the OTP data to be presented as a complete 128-bit data, you must incorporate additional logic to achieve this function.

This design example illustrates how you can implement a parallel read. The example contains additional logic that governs the reading process from the ALTOTP megafunction block, and stores the serially read data in a shift register. When the process is complete, a signal is asserted to indicate that the 128-bit OTP data is ready to be read.

This simplifies the reading process in which you only need to execute the reading, and wait for the indicator to show when the parallel data is ready to be read, without considering the OTP shifting and loading processes.

This design example illustrates the concept of how the 128 bit OTP data is read paralelly. However, this design example might not represent an optimized design or the methodology of the implementation.

The Quartus® II simulator does not support an OTP fuse block simulation. This design example uses the ModelSim®-Altera software to run the functional simulation. Refer to the **readme.txt** file that comes with the design example for the steps on running the functional simulation.

For more information about the ModelSim-Altera software, refer to the *Simulation* section in volume 3 of the *Quartus II Handbook*.

Design Files

The design examples in this user guide are available on the Literature and Technical Documentation page of the Altera website. The files are located under the following sections:

- On the Quartus II Development Software Literature page, expand the **Using Megafunctions** section and then expand the **Memory Compiler** section
- Literature: User Guides section

The following design files can be found in **DesignExample_otpfuse.zip**:

- **myotp.v**
- **shiftreg.v**
- **control_blk.v**
- **otp_autoread_ser2par.v**
- **otp_autoread_ser2par.vt**
- **script.do**

Configuration Settings

The **myotp.v** file is a design variation file for the ALTOTP megafunction that is preconfigured with the settings shown in Table 3.

Table 3. ALTOTP Megafunction Settings

MegaWizard Page	Available Options	Configured Settings
3	Currently selected device family.	HardCopy III
	Create a clock enable port.	Selected
	Initialization data (hex).	123456789abcdef0fedcba9876543210

To simulate the ALTOTP megafunction in a third-party EDA simulation tool, you need to use the HardCopy III atom simulation library (**hardcopyiii_atoms.v**), which is located at *<quartus installation path>*/eda/sim_lib.

The **shiftreg.v** file is a design variation file for the LPM_SHIFTREG megafunction that is preconfigured with the settings shown in Table 4.

Table 4. LPM_SHIFTREG Megafunction Settings

MegaWizard Page	Available Options	Configured Settings
1	Currently selected device family.	HardCopy III
	How wide should the 'q' output bits be?	128 bits
	Which direction do you want the registers to shift?	Right
	Which output do you want (select at least one)?	Data output
	Do you want any optional inputs?	■ Clock Enable input ■ Serial shift data input
2	Do you want any optional inputs?	Asynchronous inputs - Clear

To simulate the LPM_SHIFTREG megafunction in a third-party EDA simulation tool, you need to use the LPM simulation library (**220model.v**), which is located at `<quartus installation path>/eda/sim_lib`.

The **control_blk.v** file contains the control logic that automatically performs the loading and shifting operations of the `myotp` block when you execute reading. During the shifting operation, the serially read data is transferred from the `myotp` block to the `shiftreg` block. When the last bit of the OTP data is read, the shifting operation ends and asserts the indicator to indicate that the parallel data on the `shiftreg` block is valid and ready to be read.

The **otp_autoread_ser2par.v** is a top-level design variation file that instantiates `myotp`, `shiftreg`, and `control_blk`. To simulate the design, a test bench file, **otp_autoread_ser2par.vt** is created and you can use the **script.do** file to automate the simulation in the ModelSim-Altera software.

Understanding the Simulation Results

Table 5 shows the input and output ports of the top-level block. These ports are used in "Design Example: Parallel Read of 128-bit OTP data" on page 5.

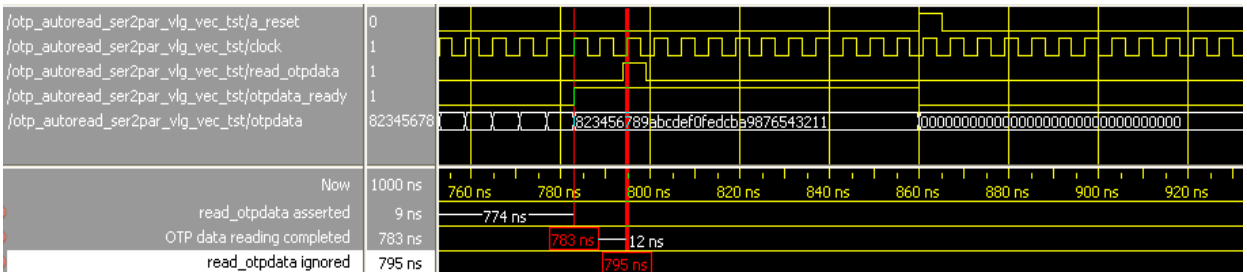
Table 5. Top-Level Input and Output Port Descriptions

Port Name	Port Type	Description
a_reset	Input	Asynchronously reset otpdata_ready and otpdata.
clock	Input	System clock for the control_blk, myotp, and shiftreg blocks.
read_otpdata	Input	Set to **HIGH** for one clock cycle to perform OTP reading. Setting the signal **HIGH** for more than one clock cycle or intermittent setting on the signal has no effect when the OTP reading is performed.
otpdata_ready	Output	When asserted, it indicates that the parallel data at the otpdata port is valid and ready to be read.
otpdata	Output	Represents the 128-bit OTP data when otpdata_ready is asserted. Ignore any data on the otpdata port if otpdata_ready is **LOW**.

Figure 4 shows the expected simulation waveform with the OTP data shown in parallel form of 128-bits when the OTP reading is completed.

The frequency that is used to illustrate the OTP loading and shifting operations does not reflect the supported frequency. HardCopy III devices only support up to approximately 100 MHz.

Figure 4. 128-Bit Parallel OTP Data



The read_otpdata port is sampled **HIGH** at 9 ns (not shown in the waveform) to perform OTP reading. At 783 ns, the otpdata_ready port is asserted, indicating that the OTP reading is complete and all serially read OTP data are converted to parallel form. The entire process from loading the data to the time when the OTP data is ready to be read in parallel form takes approximately 129 clock cycles.

At 795 ns, read_otpdata is ignored because it requires only one assertion. Asserting the read_otpdata port during the OTP reading or after the OTP reading is complete has no effect on the OTP data. If you want to clear the otpdata_ready and otpdata ports or repeat the OTP reading, set a_reset to **HIGH** before asserting read_otpdata.

Ports and Parameters

Figure 5 shows the input and output ports of the ALTOTP megafunction block.

Figure 5. Input and Output Ports of the ALTOTP Megafunction Block

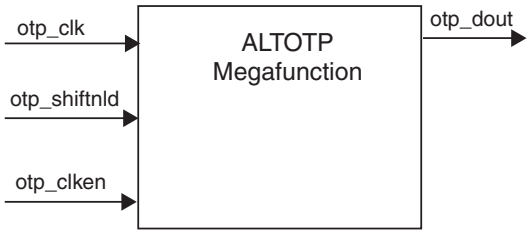


Table 6 describes the ports and parameters of the ALTOTP megafunction.

Table 6. ALTOTP Megafunction Ports and Parameters

Name	Type	Description
otp_clk	Input	Source clock to the ALTOTP megafunction block.
otp_shiftnld	Input	Controls the shifting and loading operations. Set the otp_shiftnld port to **LOW** or **HIGH** to activate the loading or shifting operation, respectively. (1) The loading operation loads the 128-bit OTP data parallel from the storage register to the access register in one clock cycle. After the loading operation is performed, the OTP data is always retained in the access register regardless of the shifting operation performed. (2) The shifting operation serially reads the OTP data from the access register to the otp_doutport. To read all the OTP data completely, set the otp_shiftnld port to **HIGH** and hold it for 127 clock cycles. (3)
otp_clken	Input	This is an optional port. The default value is **HIGH** when it is not used for the ALTOTP megafunction. When it is used, assert the port to enable the operation of the OTP block. (4)
otp_dout	Output	The OTP data is serially read from the access register. The data is read from the LSB to the MSB of the 128-bit OTP data.
init_data	Parameter	A 128-bit initialization data for the storage register. The 128-bit initialization data can be set in hexadecimal format through the ALTOTP MegaWizard™ interface. If the value of init_data is less than 128 bits, the Quartus II software appends '0' from the MSB to the rest of the uninitialized data bits. (5)

Notes to Table 6:

(1) Before performing the shifting operation or the OTP reading, the loading operation must be performed by setting the otp_shiftnld port to **LOW** for at least one clock cycle.

(2) For the loading or reloading operation, the otp_dout port always show the LSB of the OTP data.

(3) When the loading operation occurs, the otp_dout port shows the LSB of the OTP data. This causes the shifting operation to start reading from the 2nd bit of the OTP data.

(4) To use the otp_clken port, select **Create a clock enable port** from the ALTOTP MegaWizard interface.

(5) The initialization data is only meant for simulation purposes and is not programmed to the OTP fuse block because the ALTOTP megafunction does not have OTP programming capability.

Document Revision History

Table 7 table shows the revision history for this user guide.

Table 7. Document Revision History

Date	Version	Changes Made
November 2009	1.0	Initial release

101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support