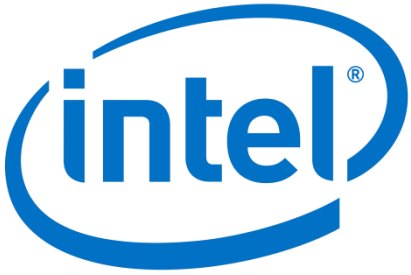

MAX 10 User Flash Memory(UFM) Write Operation

Author: Lee Lai Kum (Yvonne)

Date: March 2019

Revision: 1.0

Contents

1 Introduction

Intel MAX10 User Flash Memory (UFM) offers a single 32-bit program (write) operation. This design example demonstrates how to write the control register value through Avalon-MM control slave interface and write data into flash. Make sure to erase the flash before you perform a write operation.

2 Hardware and Software Requirements

2.1 Hardware Requirements

This design requires the following hardware:

- MAX10 FPGA Development Kit
- USB Blaster/Mini USB cable for programming the device

2.2 Software Requirements

This design requires the following software:

- Quartus Prime 17.1 Standard

3 Design Example Installation Files

File or Directory Name	Description
top.v	The top-level HDL file for this design example
Write_read.v	This file contains the state machine how to write data into UFM and read back the data from UFM after write to confirm the write is successful
pll1.v	The PLL that generates the system clocks for the top-level design
issp1.qsys	The generated in-system source and probe IP. The source of this IP is used to select write operation or read operation
ufm1.qsys	The generated on-chip flash IP core
top.sdc	The .**sdc** constraints file for the top-level design. Defines the top-level clocks in the design and sets timing and false paths to and from the top-level connections.
stp2.stp	The **.stp** file provides the signals of the control register and the flash

4 Functional Description

This design example is targeting on MAX10 device. The UFM offers a single 32-bit program (write) operation. This design example demonstrates how to write the control register value through Avalon-MM control slave interface and write data into flash up to address #256. The UFM will perform an incrementing burst read operation with burst count set to 8 after the write to confirm the write is successful.

A state machine has been defined to perform UFM write operation and read operation with the following steps:

1. Disable write protection mode
2. Write data into address up to address#256
3. Check the status register to confirm if the write is successful
4. Enable the write protection mode after write is successful
5. Start incrementing burst read operation, burst count set to 8 to confirm the read data is from previous write data.
6. Stop reading data if the address reached 256

5 Run the Design Example

5.1 Compiling the Design

1. Download the **ufm_write_test.par** from the design store
2. Launch the Quartus Prime software
3. Open the **.par** file from your working directory
4. In the Quartus Prime software, click **Processing > Start Compilation**

5.2 Running the Design on Hardware

1. In the Quartus Prime software, click **Tools > Programmer**
2. In the **Programmer** window, click on the **Hardware Setup**. A Hardware Setup dialog box will be opened
3. Select USB Blaster under **Currently selected hardware**. Click on close.
4. In the Programmer window, click **Add File**. Select the **.pof** file from your working directory
5. Check the **Erase** checkbox, then click **Start** to erase the flash before performing write operation

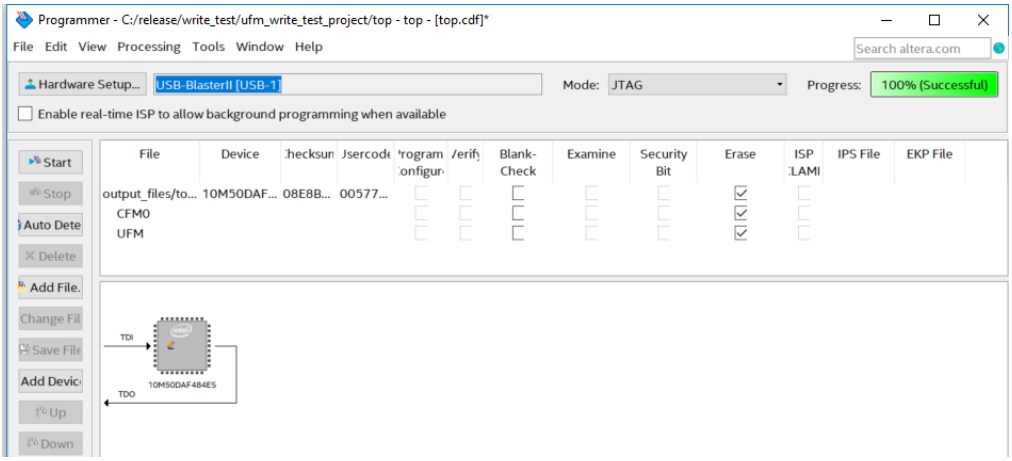

Figure 1: Quartus Prime Programmer – Erasing the flash

6. Check the **Program/Configure** checkbox, then click **Start** to start programming.

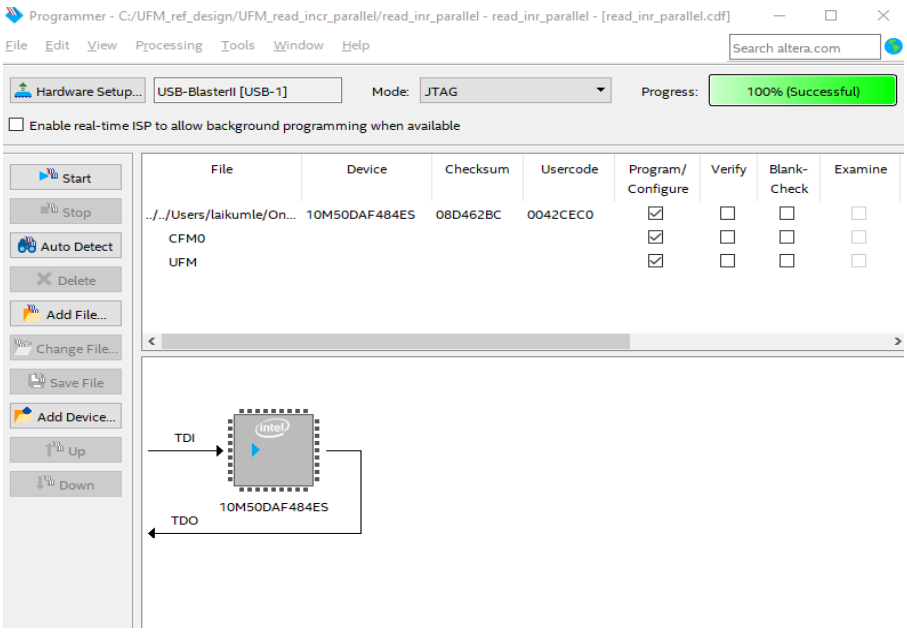

Figure 2: Quartus Prime Programmer – Programming the flash

7. In the Quartus Prime software, click **Tools > Signal Tap Logic Analyzer**
8. In the Quartus Prime software, click **Tools > In-System Sources and Probes Editor**
9. In the **Signal Tap Logic Analyzer** window, at the **SOF Manager** section, browse the **.sof** file from your working directly and click the **program device** button to program the sof file
10. In the **Signal Tap Logic Analyzer** window, Click **Processing > Autorun Analysis**
11. In the **In-System Sources and Probes Editor** window, click **Processing > Read Source Data**
12. Click on the **Data** for **source[1]** to change the value for 0 to 1 to enable write operation
13. After a while, change back the **source[1]** value to 0 to disable write operation
14. Click on the **Data** for **source[0]** to change the value for 0 to 1 to enable read operation

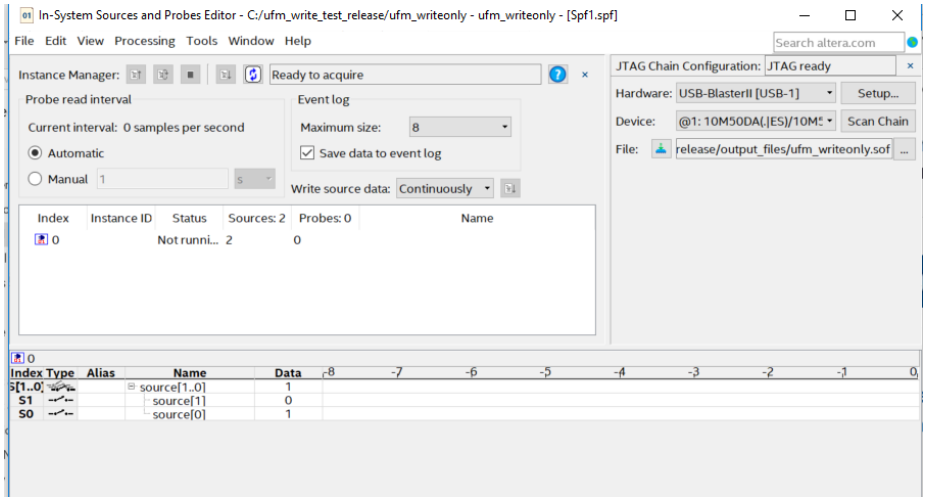

Figure 3: Sources from In-system Sources and Probes editor

15. Burst count is set to 1 during write operation. Figure 4 shows the signals from Signal Tap during write operation.

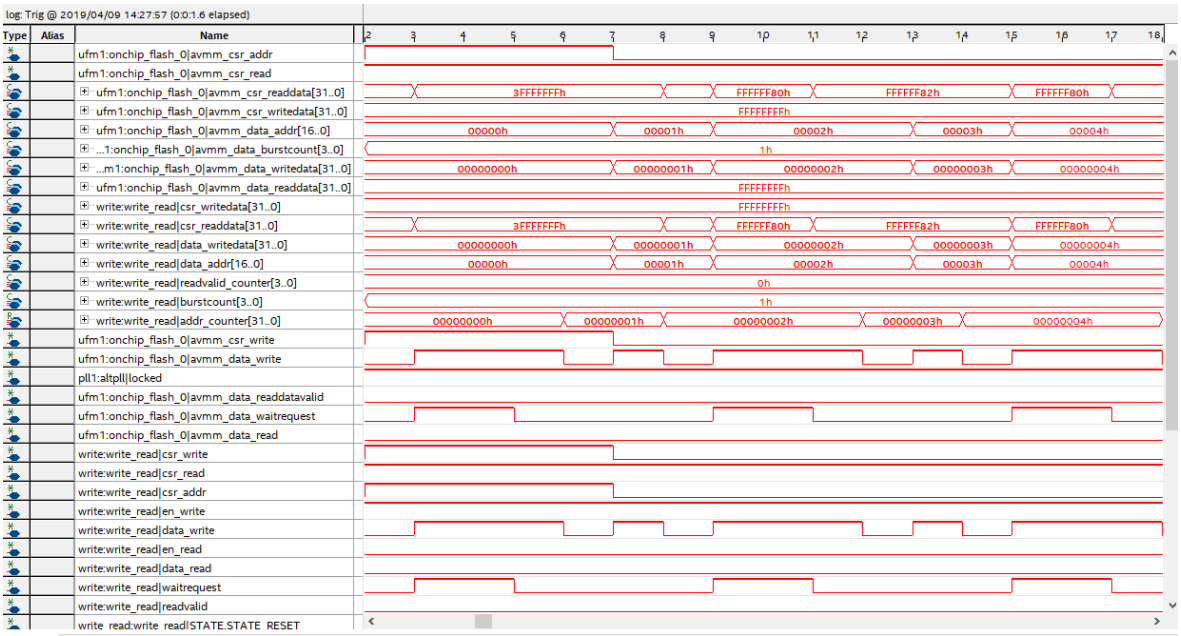

Figure 4: data_writedata and data_addr signals from Signal Tap during write operation

16. Burst count is set to 8 during incrementing burst read operation. Read operation will stop after address reach 256. data_readdata value is the data that was write into the flash IP core previously as shown in the Figure 5.

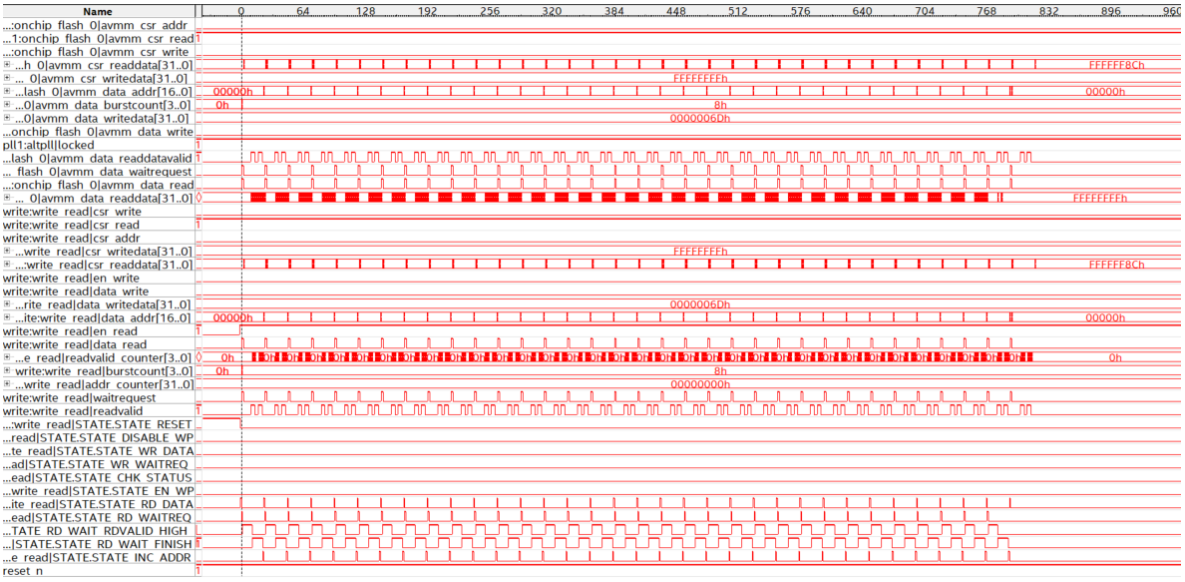

Figure 5: data_readdata and data_addr signals from Signal Tap during read operation

6 Revision History

Revision	Description
1.0	Initial Release