TB-7V-xxxT-PCIEXP
Configuration User Guide

Rev.1.02

Revision History

Version	Date	Description	Publisher
Rev.1.00	2013/11/07	Release version	Goto
Rev.1.01	2014/04/26	Add “onboard silk” information	Yoshioka
Rev.1.02	2014/10/28	Change ISE to VIVADO	Yoshioka

Table of Contents

List of Figures

List of Tables

1. Introduction

This document is described configuration method of TB-7VXxxxT-PCIEXP.

- The onboard FPGA (Virtex-7) allows the user to perform configuration using microSD Card and BPI Flash.
- Any of the following operations can be selected by configuring the MODE Select Rotary Switch (SW1).
  - ➢ Configuration from microSD Card to FPGA (Virtex-7)
  - ➢ Configuration from BPI Flash to FPGA (Virtex-7)
  - ➢ Writing data to BPI Flash or FPGA (Virtex-7) using a JTAG cable
- The microSD Card can store up to 16 configuration data sets. The user can perform configuration of the FPGA (Virtex-7) by selecting these data.
  The user also can perform an instant configuration by specifying desired data with the AREA Select Rotary Switch (SW2).

[Caution]
Before removing or inserting the microSD card, be sure to turn off the power switch of the board.

2. Switches and LEDs

2.1. Switches

2.1.1. SW3 (Virtex-7 Reconfiguration Push-SW) [Onboard silk: V7_RCFG]

- Press SW3 to execute the operation selected by SW1.
- The same operation is executed as the board is powered off and on.
- SW3 is located adjacent to the SD Card slot.

2.1.2. SW1 (MODE Select Rotary SW) [Onboard silk: S3_MODE]

- This switch is used for configuration mode selection.
- A predetermined operation can be executed by setting SW1 to a predetermined value and then pressing SW3 or powering the board again.

The table below shows the SW1 settings and operational behaviors.

Table 2-1 microSD Configuration Mode Setup

SW1	Mode Name	Description
0	BPI =>FPGA Transfer Mode (16bit transfer)	Configuration is executed from BPI to FPGA in 16bit Slave SelectMap.
1	microSD Card=>FPGA Transfer Mode (16bit transfer)	Configuration is executed from microSD Card to FPGA in 16bit Slave SelectMap.
2-E	No operation	No operation is executed.
F	JTAG Transfer Mode	This mode is selected to configure FPGA directly using a JTAG cable or write configuration data to BPI Flash.

2.1.3. SW2 (AREA Select Rotary SW) [Onboard silk: S3_AREA]

- In the microSD Card => FPGA Transfer Mode of operation (SW1=1), configuration is executed on the CONFIG.TXT file stored in the microSD Card by selecting any of 16 bit files which have been associated with #0 through #F commands.

(For information on how to describe the CONFIG.TXT file, refer to Section 3.2 and later).

Table 2-2 microSD Card Area Mapping

SW2	microSD Card
0	bit file #0
1	bit File #1
2	bit file #2
3	bit file #3
4	bit file #4
5	bit file #5
6	bit file #6
7	bit file #7
8	bit file #8
9	bit file #9
A	bit file #A(10)
B	bit file #B(11)
C	bit file #C(12)
D	bit file #D(13)
E	bit file #E(14)
F	bit file #F(15)

2.2. LEDs

2.2.1. LED1 (S3 CFG_DONE) LED [Onboard silk: S3_CFG_DONE]

- The LED provides the status information of DONE signal of the Spartan3AN that handles Virtex-7 configuration.
- The LED will light when the Spartan3AN successfully boots up prior to Virtex-7 configuration startup.

2.2.2. LED4 (V7 CFG_DONE) LED [Onboard silk: V7_CFG_DONE]

- The LED provides the station information on Virtex-7 DONE signal.
- The LED will light when the configuration to Virtex-7 is completed successfully.

3. Configuration Sequence

3.1. microSD Card => FPA Transfer Mode (SW1 = 1)

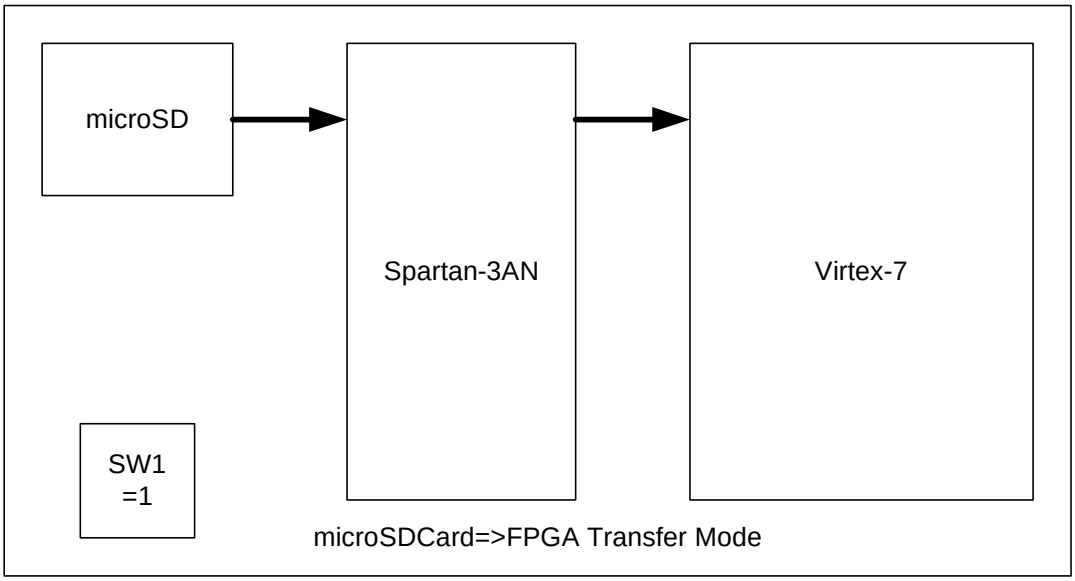

Figure 3-1 microSDCard=>FPGA Transfer Mode

- In this mode of operation, FPGA configuration is executed by reading a bit file from microSD Card.
- Write a CONFIG.TXT and a bit file (or “multiple”) in the root directory of the bit microSD Card.
  - ➢ Use Microsoft Windows for writing.
    (If written from Linux to microSD Card, the system cannot read correctly).
  - ➢ Any bit files generated by Linux version of ISE can be used. Using Microsoft Windows write files to microSD Card.
- List bit file names and various parameters in the CONFIG.TXT file. (Refer to Section 3-2 “CONFIG.TXT Sample”)
  If no CONFIG.TXT file is found, FPGA configuration is not executed.

3.1.1. CONFIG.TXT

- This text file is referred to when configuration data is loaded from microSD Card to FPGA.
- A fixed file name of “CONFIG.TXT” (lower-case, case-insensitive, one-byte ASCII codes) is used. This file must be placed on the root directory of microSD Card.
- List bit file names and various parameters in the CONFIG.TXT file.
- Only one-byte ASCII codes are allowed for use in the CONFIG.TXT file.

3.1.2. CONFIG.TXT Sample

It is required to prepare a file named CONFIG.TXT (lower case also acceptable) in the root directory of microSD Card.

```
// This file is config test          <= '/' this mark is comment out as Verilog-HDL.

#0 : TESTLED0.bit
#1 : TESTLED1.bit
#2 : TESTLED2.bit
#3 : TESTLED3.bit
#4 : TESTLED4.bit
#5 : TESTLED5.bit
#6 : TESTLED6.bit                   <= File name must be 8 characters or shorter.

//end
```

Figure 3-2 CONFIG.TXT Sample

3.1.3. Commands

(a) "/" ("slash")

- The slash ("/") is used to comment out.
- If a line contains a slash ("/"), a character string that begins with the slash ("/") and ends with a "CR" is skipped as a comment.

(b) bit file-name

- All lines that begin with a character not belonging to the above (a) or (b) are handled as a bit file by the Line-Processor.
- All bit file names must be less than or equal to 8 characters.
- These bit file names can contain a space or TAB before or after each bit file name (these space and TAB cannot be inserted in the file name).

(c)#0 ~ #F: bit file name

- Each bit file name must be associated with 16 hex digits from 0 to F.
- Immediately after the # sign, 0~9 (30h~39h), A~F (41h~46h) or a~f (61h~66h) can be placed.
- Immediately after the ":", insert a bit file name you want to associate with.
- Each bit file name must be less than or equal to 8 characters.
- A space or TAB can be inserted before or after a file name (each file name cannot contain a space and TAB).

3.1.4. Restrictions

- Use the microSD Card that is formatted to less than or equal to 2GB FAT16.
  No support for SDHC that is formatted to more than or equal to 2GB FAT32.
  Even for the microSD Card that is formatted to less than or equal to 2GB, those that have been reformatted to FAT32 are not supported.
  To reformat your card, choose FAT16 (the word “FAT” appears in the Microsoft Windows volume property file system).
- bit file name must be less than or equal to 8 alphanumerical characters.
  Both uppercase and lowercase characters can be used (case-insensitive).
  ”_”(under bar) and ”-“(hyphen) can also be used as a character.
- If a file with filename length of more than or equal to 8 characters is copied to the microSD Card and then the filename is changed to less than 8 characters in length using the “rename” function, the file is still handled as a more than or equal to 8 character file and only the first 7 characters will be identified as a filename. So, change the file name first and then make a copy.

3.2. BPI Flash => FPGA Transfer Mode (SW1 = 0)

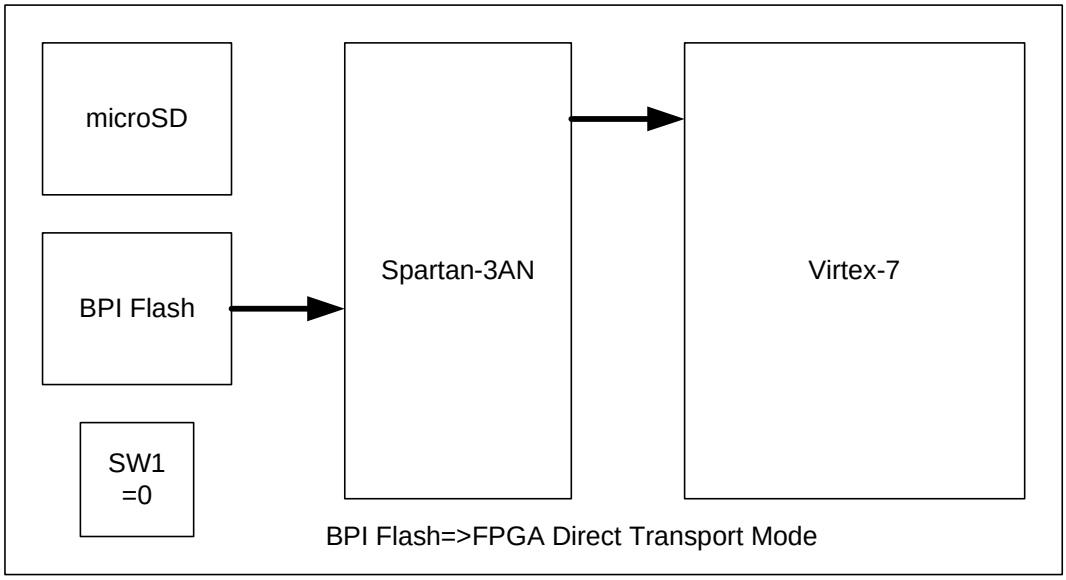

Figure 3-3 BPI Flash => FPGA Transfer Mode

- This mode is used to read a MCS file from the BPI Flash and configure FPGA.
  MCS file must be BPI Flash memory before operate this mode. Please refer JTAG Transfer Mode.

3.3. JTAG Transfer Mode (SW = F)

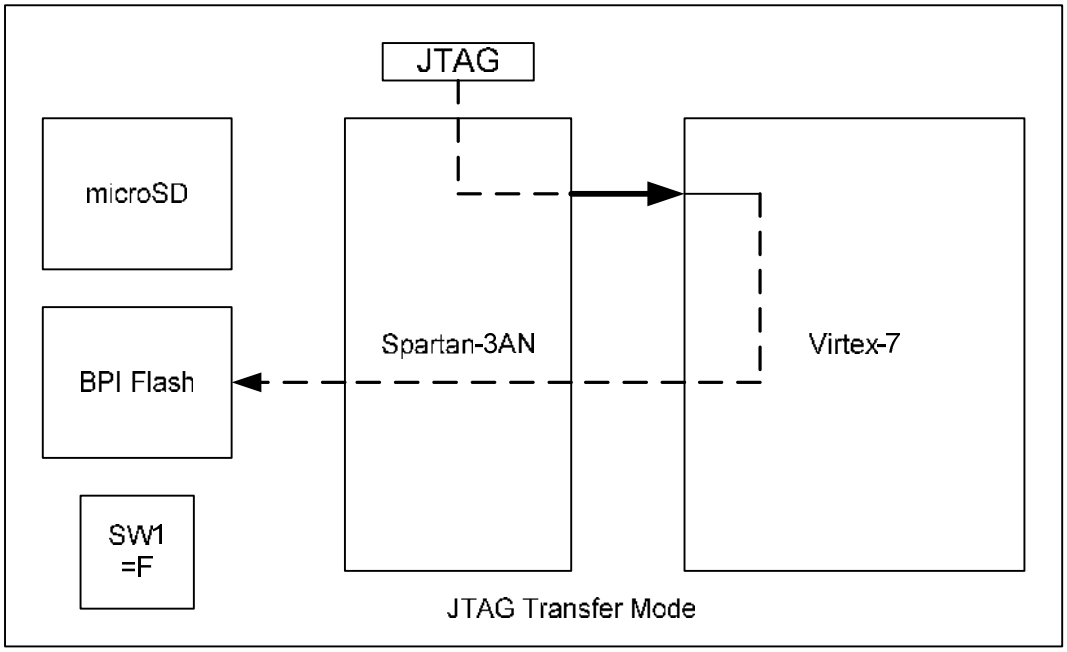

Figure 3-4 JTAG Transfer Mode

- This mode is used to configure a bit file directly to the FPGA using a Xilinx program tool and a JTAG cable or write a MCS file to the BPI Flash.
- When creating a MCS file for BPI Flash, insert the following descriptions into xxx.xdc for use by Xilinx Vivado Design Suite.

set_property BITSTREAM.CONFIG.BPI_SYNC_MODE TYPE1 [current_design]

set_property BITSTREAM.CONFIG.UNUSEDPIN PULLNONE [current_design]

set_property BITSTREAM.CONFIG.CONFIGRATE 40 [current_design]

3.3.1. Method to create a xxx.mcs file for BPI Flash

1. On the VIVADO GUI, using “cd” command on the Tcl Console and move to folder which is located bit file. ex, cd c:/my_work/test_mcs
2. Make a MCS file by following command
   [wirte_cfgmem –force –format MCS –seize 128 –interface BPIx16 –loadbit “up 0x0 xxx.bit” –file xxx.mcs]
   Ex, wirte_cfgmem –force –format MCS –seize 128 –interface BPIx16 –loadbit “up 0x0 test.bit” –file test.mcs

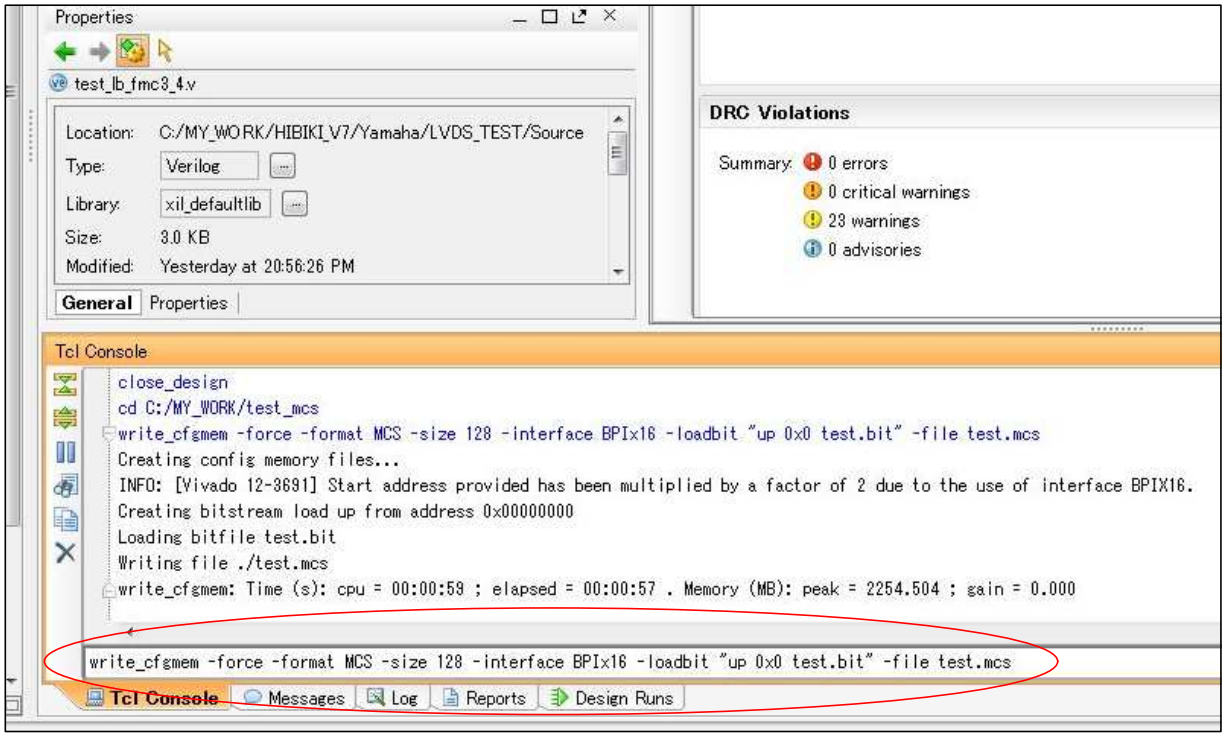

Figure 3-5 VIVADO Tcl Console GUI

3.3.2. Method to write a xxx.mcs file to BPI Flash

1. Before power up, set SW1 to “F” and Connect JTAG Cable then power up.
2. On the VIVADO GUI, Clock to “Open Hardware Manager”

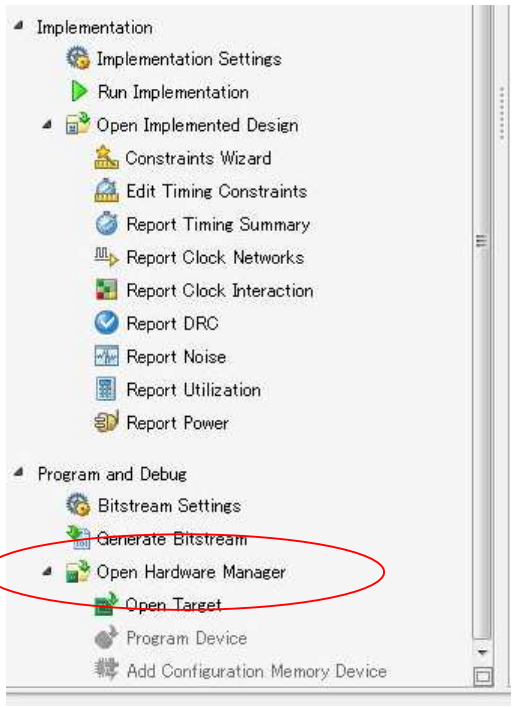

Figure 3-6 Open Hardware Manager

3. Clock “Open recent target” or “Open a new hardware target” then check JTAG Chain.

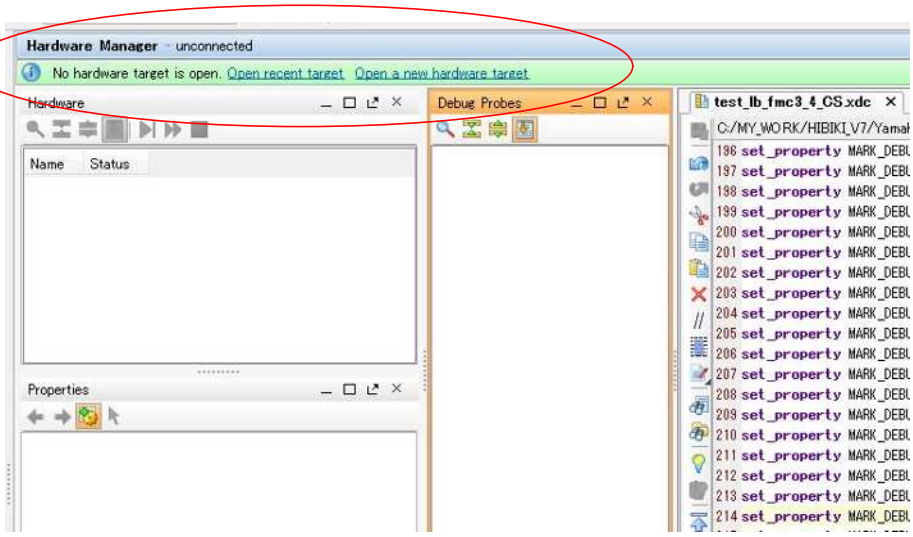

Figure 3-7 Activate JATAG Chain

4. Right click to the device model# and select “Add Configuration Memory Device”.

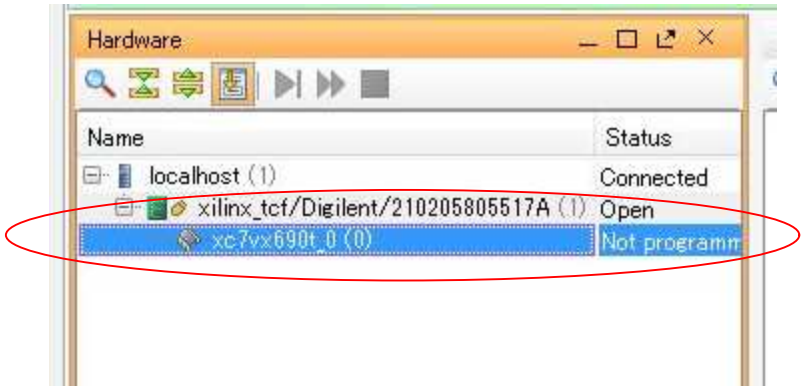

Figure 3-8 Add Configuration Memory Device

5. Following setting windows will be open. Set to following then clock to OK.
   Manufacturer: Micron
   Type: bpi
   Density(Mb): 1024
   Width: x16
   Memory Part: mt28gu01gaax1e-bpi-x16

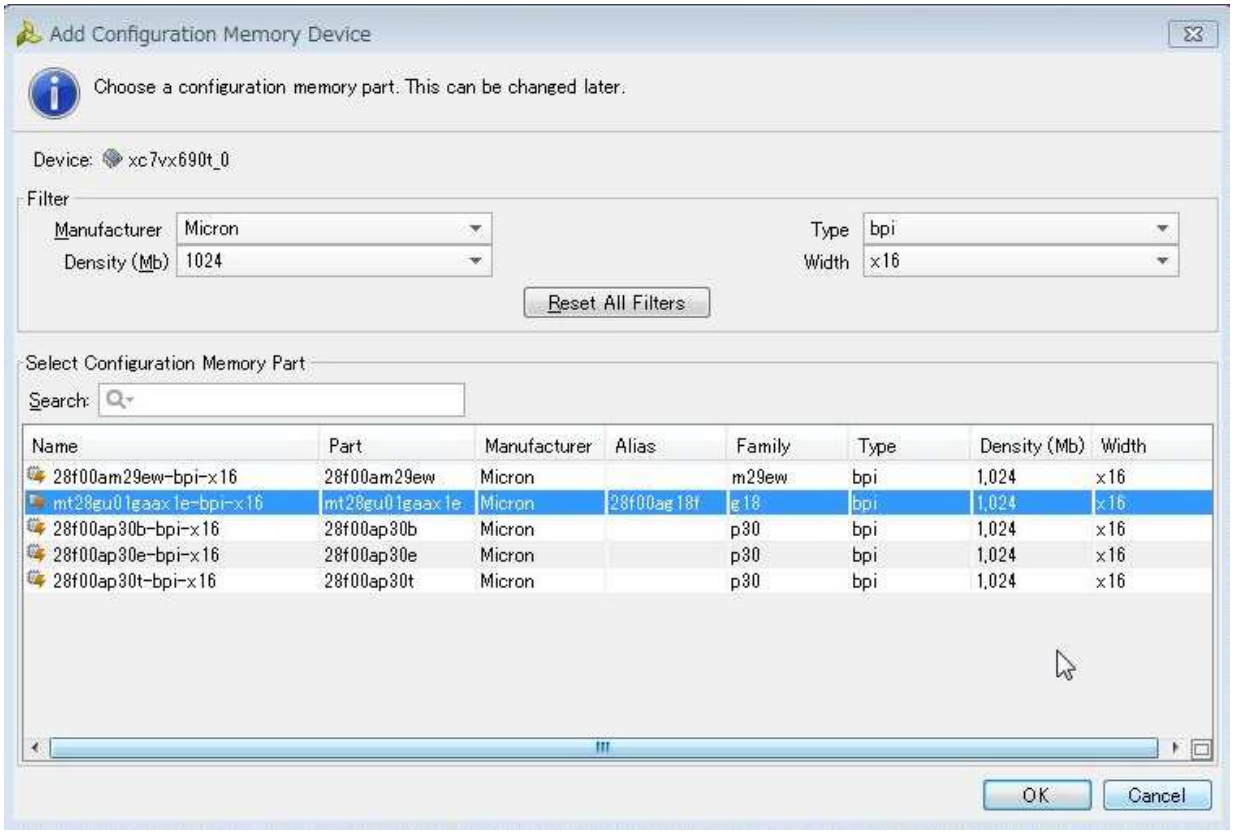

Figure 3-9 Set memory device

6. The following window will appear. Click "**OK**".

Figure 3-10 Confirmation window of Add Device

7. The following window will appear. Select xxx.mcs file to "Configuration file" then clock "OK". Starting to write mcs file to BPI flash memory.

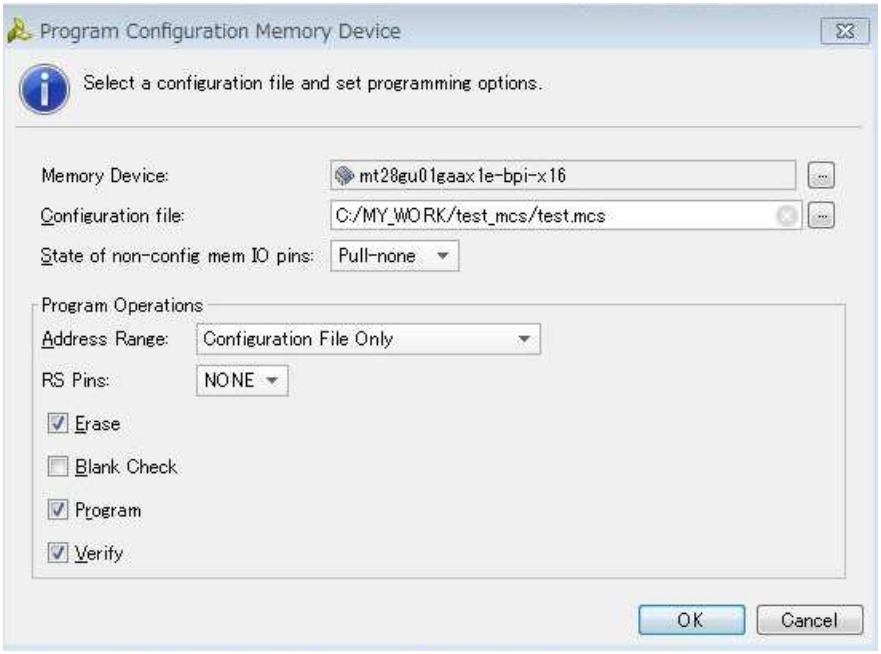

Figure 3-11Starting BPI flash write

8. The following window will appear. Operation is successfully finished. Power down and set SW1 to "0" then power up again. It will start BPI configuration.
   If LED4 is light up, Virtex-7 device configuration is finished.

Figure 3-12 Flash Programing Succeeded

TOKYO ELECTRON DEVICE

PLD Solution Dept. PLD Division
URL: **http://solutions.inrevium.com/**
E-mail: psd-support@teldevice.co.jp

HEAD Quarter: Yokohama East Square, 1-4 Kinko-cho, Kanagawa-ku, Yokohama City,
Kanagawa, Japan 221-0056
TEL: +81-45-443-4016 FAX: +81-45-443-4058