

# USING MULTIBOOT TECHNIQUE TO CREATE A MULTIPLE EMBEDDED DESIGNED SYSTEMS

<sup>1</sup>M. R. KHALIL, <sup>2</sup>S. A. MOHAMMED

<sup>1</sup>Asstt Prof., Department of Electrical Engineering, Technical College, Mosul, Iraq

<sup>2</sup> Msc. Student, Department of Computer Technical Engineering, Technical College, Mosul, Iraq

E-mail: [sreng86@gmail.com](mailto:sreng86@gmail.com)

## ABSTRACT

The research aims to save two configurations of two embedded designed systems in Intel Strata Flash Parallel NOR PROM available on Spartan-3E Starter kit in order to increase the capabilities of the XC3S500E FPGAs slice. Multiboot methodology is adopted to perform the goal of the work. Embedded development kit(EDK) prepared by Xilinx company is employed to develop two different embedded systems. Picoblaze microcontroller is used for the purpose of programming the Intel Strata Flash PROM. Integrated software environments, Xilinx ISE, is used to instantiate STARTUP\_SPARTAN3E component for the verification of Multiboot techniques.

**Keywords:** *Soft-Core Processor, Multiboot, Configuration, STARTUP\_SPARTAN3E.*

## 1. INTRODUCTION

Multi Boot describes the ability of the FPGA to load an image from one of several configuration images which are called "revisions". MultiBoot for the Spartan-3E family of devices works with the BPI configuration mode only and only allows two revisions. The first revision to load upon device power-up is the one specified by the FPGA's Mode Pin connections on the board. The second revision loads from the opposite end of the PROM and the addresses generated by the FPGA increment (or decrement) in the opposite direction of the board's Mode Pin connections. In order to use this capability for Spartan-3E, a design must be created with a connection to the Multi-Boot Trigger (MBT) pin on the Startup block. Then a second design can be created without MBT connection [1]. In this paper the revision is designed as an embedded processor system to take advantage of embedded system, In a variety of applications, an embedded processor is key to system flexibility, maintainability, and low cost.[2]

Embedded Development Kit (EDK) tools are used for this purpose to enables a design of a complete embedded processor system for implementation in a Xilinx FPGA device. A strataflash parallel NOR prom(16 Mbyte) is available on the Spartan3E starter Kit board whose connection to the FPGAs slice are shown in figure 1. The work in this paper aim to stores two different

FPGA configurations in the StrataFlash device and dynamically switch between the two using the Spartan-3E FPGA's MultiBoot feature. The suggested embedded design system is shown in figure 2.

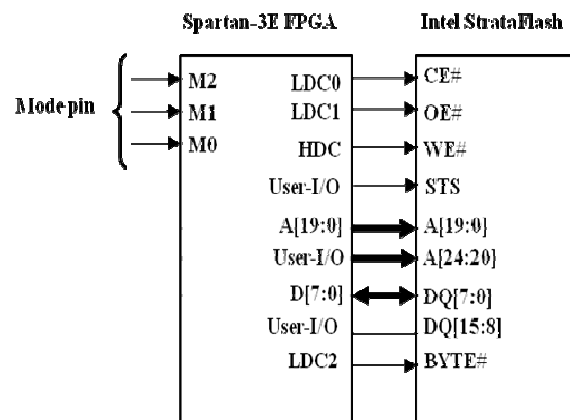


Figure 1. Connections of Intel Strataflash Flash Memory with Spartan-3E slices

The MicroBlaze embedded soft core processor is a reduced instruction set computer (RISC) optimized for implementation in Xilinx Field Programmable Gate Arrays (FPGAs) [4].

Processor Local Bus (PLB) v4.6 provides bus infrastructure for connecting an optional number of PLB masters and slaves into an overall PLB system [5].

Multi-Port Memory Controller MPMC is a fully parameterizable memory controller that supports SDRAM/DDR/DDR2 memory [6].

External Memory Controller (EMC) provides the control interface for external synchronous, asynchronous SRAM and Flash memory devices through the MCH or PLB interfaces[7].

Universal Asynchronous Receiver Transmitter (UART) Lite Interface connects to the PLB (Processor Local Bus) and provides the controller interface for asynchronous serial data transfer. This soft IP core is designed to interface with the PLBV46[8].

The BRAM Block is a configurable memory module that attaches to a variety of BRAM Interface Controllers[9].

The LMB BRAM Interface Controller is the interface between the LMB and the bram\_block peripheral. A BRAM memory subsystem consists of the controller along with the bram\_block peripheral[10].

The LMB is a fast, local bus for connecting MicroBlaze instruction and data ports to high-speed peripherals, primarily on-chip block RAM (BRAM) [11].

The work must in FPGA based embedded designed system with two configurations that the user can switch between them.

## 2. ISE DESIGN FLOW

The ISE design flow is shown in the figure 3 and it implied the following stages [1].

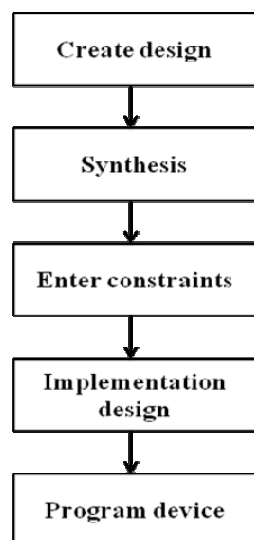


Figure 3. ISE design flow

### Create design

In this stage a VDL sources files are created, these file describe the functionality of the designed hardware system.

### Constraints Entry

In design constraints, the timing, placement, and other design requirements are specified. The ISE software provides editors to facilitate constraints entry for timing constraints as well as I/O pin and layout constraints

### Synthesis

After design entry and optional simulation, you run synthesis. During this step, VHDL, Verilog, or mixed language designs become netlist files that are accepted as input to the implementation step.

### Implementation

After synthesis, you run design implementation, which converts the logical design into a physical file format that can be downloaded to the selected target device. Using the Project Navigator Design Goals and Strategies, you can modify process properties to control the implementation and optimization of the design .

## 3. DEVICE CONFIGURATION AND PROGRAMMING

After generating a programming file, the device is configured. During configuration, configuration files are generated and the programming files from a host computer is downloaded to the Xilinx device.[1]

The Spartan-3E FPGA Starter Kit board supports a variety of FPGA configuration options:

1. Download FPGA designs directly to the Spartan-3E FPGA via JTAG, using the on board USB interface. The on-board USB-JTAG logic also provides in-system programming for the on-board Platform Flash PROM and the Xilinx XC2C64A CPLD. SPI serial Flash and StrataFlash programming are performed separately.
2. Program the on-board 4 Mbit Xilinx XCF04S serial Platform Flash PROM, then configure the FPGA from the image stored in the Platform Flash PROM using Master Serial mode.
3. Program the on-board 16 Mbit ST Microelectronics SPI serial Flash PROM, then configure the FPGA from the image stored in the SPI serial Flash PROM using SPI mode.

4. Program the on-board 128 Mbit Intel StrataFlash parallel NOR Flash PROM, then configure the FPGA from the image stored in the Flash PROM using BPI Up or BPI Down configuration modes. Further, an FPGA application can dynamically load two different FPGA configurations using the Spartan-3E FPGA's MultiBoot mode [3].

The configuration type are used in this paper is Spartan-3E FPGA's Multiboot mode.

## 5. CONFIGURATION: INITIALIZATION AND TIMING

Figure 4 shows the configuration process flow for a spartan3E FPGA.

Upon power-up, an FPGA's INIT pin is internally held Low while the FPGA initializes its internal circuitry and clears its configuration memory. Configuration will start as soon as the INIT pin transitions High, either after power-up or after the FPGA's PROG\_B pin is held Low and released.

The PROG\_B pin must be held low for a preset amount of time ( $T_{\text{PROG}}$ ). Figure 5 shows a representative example of Spartan3E power-up configuration timing, and Table 1 shows the associated timing characteristics [12].

Table 1. Power-Up Timing Characteristics

| Symbol             | Description  | Value    | Units         |
|--------------------|--|----------|---------------|
| $T_{\text{POR}}$   | Power-on Reset.  | 5 to 7   | ms            |
| $T_{\text{PL}}$    | Program Latency.   | 0.5 to 2 | ms            |
| $T_{\text{PROG}}$  | Program Pulse Width.   | 500      | ns            |
| $T_{\text{MINIT}}$ | Setup time on M[2:0] mode-select pins.                                     | 50       | ns            |
| $T_{\text{ICCK}}$  | the time from the rising edge of INIT_B until CCLK output begins toggling. | 0.5 to 4 | $\mu\text{s}$ |

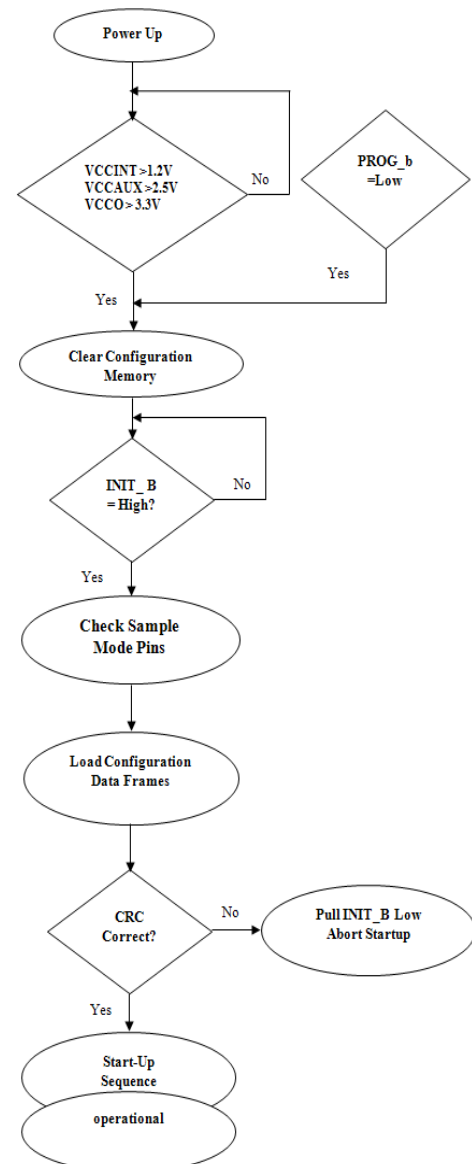


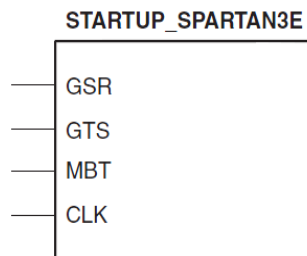
Figure 4. Configuration Process

## 6. MULTIBOOT CONFIGURATION

In order is to store more than a configuration in the PROM the following steps are adopted:

1. T transfer the Embedded system to the ISE
2. Using the STARTUP\_SPARTAN3E primitive, the Spartan-3E STARTUP primitive has an input pin to support MultiBoot functions. By default, the MultiBoot feature is disabled. To use MultiBoot in an application, the FPGA design must first include a STARTUP\_SPARTAN3E design

primitive. To trigger a MultiBoot event, assert a Low pulse lasting at least 300 ns on the MultiBoot Trigger (MBT) input to the primitive. When the MBT signal returns High after the Low pulse, the FPGA automatically reconfigures from the opposite end of the parallel Flash memory, startup block 711 may include additional pins for providing other functions such as a global set/reset pin GSR, a global tri-state pin GTS, and a clock pin CLK [3].



## 7. MULIBOOT CONFIGURATION SYSTEM

Figure 6 shows a system with a Programmable device (Spartan-3E) having multi-boot configuration capabilities; Programmable device may be coupled to a memory device for storing configuration data for configuring programmable device. Programmable device includes programmable resources, a startup block, and configuration block. Memory device may include at least two separate bitstreams, A and B, Corresponding two different configurations of programmable device. Programmable resources of programmable device may be initially configured with configuration data A stored in memory device, corresponding to a first design.

Thereafter, Programmable device may be triggered to reconfigure with a second bitstream (e.g., bitstream B) corresponding to a second design.

the initially loaded bitstream (e.g., bitstream A of figure. 6) is "armed" by setting a multi-boot flag. For instance, the multi-boot flag may be a bit in the bitstream that sets a register within programmable device. By arming the initial bitstream in this way, programmable device may enable the multi-boot feature Armed bitstream A may also configure a portion of configurable resources to connect to startup block. Startup block may include a MBT (multi-boot trigger) pin for triggering multi-boot and a reconfiguration operation. When a reconfiguration operation is desired, the MBT pin

may be asserted. multi-boot flag of alternate configuration bitstream B may not be set.

Configuration block of programmable device may control the configuration process, including providing addresses to memory device. a sequence of addresses may be provided by a counter of configuration block. Programmable device may be set to a programming mode (e.g., via mode pins of figure. 1) such that configuration When MBT is asserted with a pulse of sufficient duration, thereby triggering a multi-boot reconfiguration, counter of configuration block may be set to an all ones address for loading the second or alternate address. Counter may also be configured to count down. A status register within configuration block may be set to store the current state. That is, status register may store information indicating that programmable device was in configuration A when the MBT pin was triggered, and/or that configuration B is about to be loaded. This causes programmable device to ignore the state of its mode pins during the reconfiguration (which would have caused the programmable device to load configuration A), and provides information for subsequent multi-boot reconfigurations to load the proper bitstream. Programmable device may then start loading configuration B starting at the all ones memory address and counting down until configuration is Complete. Normal operation with configuration B may then commence. In some application, further multi-boot reconfiguration operations may be triggered by subsequent pulses on the MBT pin. For instance, triggering MBT may allow programmable device to switch back and forth between configurations A and B, or to switch among multiple other configurations [13].

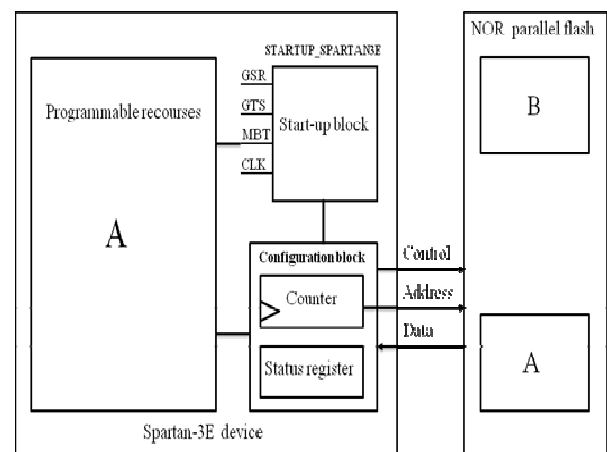


Figure 6. system with a Programmable device having multi-boot configuration capabilities

## 8. GENERATING A SPARTAN-3E MULTIBOOT PROM IMAGE USING IMPACT

The iMPACT programming software provides a graphical, step-by-step approach to create a MultiBoot PROM file. The steps outlined below are followed to create a MultiBoot PROM file using the iMPACT software [12].

1. Invoke the iMPACT programming software.
2. As shown in Figure 7, Prepare a PROM File is chosen.
3. Figure 8 shows, target a PROM Supporting Multiple Design Revisions, the Spartan3E MultiBoot method is chosen, and file format is MCS format, then a PROM File Name is entered.
5. Figure 9 displays, select the Initial Boot Direction. This is the location from where the first configuration image loads. The initial location depends on the BPI mode pin settings.
6. As shown in Figure 10, Parallel PROM Density is Selected, measured in bytes, in this case (16M) Add button is Clicked to use the PROM density.
7. In the File Generation Summary dialog box, Finish button is clicked.
8. FPGA configuration bitstream for the design is selected that initially loads at power-up or when the PROG\_B input is pulsed Low, then second MultiBoot configuration image is selected.
9. Operations → Generate File is Selected. In iMPACT to generate the MCS file. The iMPACT software successfully generates a PROM file.

The first MultiBoot image is loaded starting at PROM address 0 and ends at hexadecimal address 0x4547F. The second MultiBoot image is loaded starting at the highest PROM address, which is at hexadecimal 0xFFFFF for a 1Mbyte PROM. The image is loaded downward (decrementing address) and ends at hexadecimal address 0xFBAB80.

The Xilinx PicoBlaze NOR Flash Programmer is used to program the StrataFlash. The Xilinx PicoBlaze NOR Flash Programmer design is implemented using a single PicoBlaze processor and UART macros and is used as a configuration bit file for immediate programming of the Spartan XC3S500E on the Spartan-3E starter kit.

## 9. RESULTS AND DISCUSSION

Figure 12 shows Multiboot design results on the hyper terminal. Which at power up the first configuration loads from flash PROM address

0x000000 memory locations and increments through the PROM memory locations. After the FPGA completes configuration which contains embedded system application for test memory and peripherals of Spartan3E Starter Kit and the execution of this system as shown on figure 12. If the system memory and peripherals test is successful the multiboot event is triggered causing FPGA reconfigure from opposite end of flash PROM at starting address 0xFFFFF memory locations and decrements through the PROM memory locations this configuration contains embedded system application for simple matrix multiplication as and the result shown in figure 12.

## 10. CONCLUSION

Multiboot embedded designed system is accomplished in this paper. The system included two different configurations of embedded systems, the user can switch between the two configurations. This doubles the capability of the FPGA based embedded system without expanding the available resources.

## 11. FUTURE WORK

Multibooting technique can be envisaged to store a number of configurations more than two by using the capability of MicroBlaze processor to control the reconfiguration process.

## REFERENCE

- [1]. Xilinx Company, "ISE help"  
[http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx10\\_1/isehelp\\_start.htm](http://www.xilinx.com/support/documentation/sw_manuals/xilinx10_1/isehelp_start.htm)
- [2]. Xilinx Company, "Spartan-3 Generation FPGA User Guide", User Guide UG331 (v1.4), June 25, 2008  
[http://www.xilinx.com/support/documentation/user\\_guides/ug331.pdf](http://www.xilinx.com/support/documentation/user_guides/ug331.pdf)
- [3]. Xilinx Company, "Spartan-3E FPGA Starter Kit Board User Guide", User Guide UG230 (v1.1), January-20-2011.  
[http://www.xilinx.com/support/documentation/boards\\_and\\_kits/ug230.pdf](http://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf)
- [4]. Xilinx Company, "MicroBlaze Processor Reference Guide, " UG081 (v9.0).  
[http://www.xilinx.com/support/documentation/sw\\_manuals/mb\\_ref\\_guide.pdf](http://www.xilinx.com/support/documentation/sw_manuals/mb_ref_guide.pdf)
- [5]. Xilinx Company, "Processor Local Bus (PLB) v4.6", data sheet DS531, April 24, 2009.



- [http://www.xilinx.com/support/documentation/ip\\_documentation/ds531.pdf](http://www.xilinx.com/support/documentation/ip_documentation/ds531.pdf).
- [6]. Xilinx Company , "Multi-Port Memory Controller (MPMC) (v6.00.a) ", data sheet DS643, April 19, 2010.  
[http://www.xilinx.com/support/documentation/ip\\_documentation/mpmc.pdf](http://www.xilinx.com/support/documentation/ip_documentation/mpmc.pdf).
- [7]. Xilinx Company , "XPS Multi-CHannel External Memory Controller (v3.01a) ", data sheet DS575 Apr 19, 2010 .  
[http://www.xilinx.com/support/documentation/ip\\_documentation/xps\\_mch\\_emc.pdf](http://www.xilinx.com/support/documentation/ip_documentation/xps_mch_emc.pdf)
- [8]. Xilinx Company , "XPS UART Lite (v1.01a) ", data sheet DS643, April 19,2010.  
[http://www.xilinx.com/support/documentation/ip\\_documentation/xps\\_uartlite.pdf](http://www.xilinx.com/support/documentation/ip_documentation/xps_uartlite.pdf)
- [9]. Xilinx Company , "Block RAM (BRAM) Block (v1.00a) ",data sheet DS444 , March 12, 2007.  
[http://www.xilinx.com/support/documentation/ip\\_documentation/bram\\_block.pdf](http://www.xilinx.com/support/documentation/ip_documentation/bram_block.pdf)
- [10]. Xilinx Company , " LMB BRAM Interface Controller (v2.10b) ", data sheet DS452, March 2, 2010.  
[http://www.xilinx.com/support/documentation/ip\\_documentation/lmb\\_bram\\_if\\_cntlr.pdf](http://www.xilinx.com/support/documentation/ip_documentation/lmb_bram_if_cntlr.pdf)
- [11]. Xilinx Company, " Local Memory Bus (LMB) (v1.00a) ", data sheet DS445 , April 24. 2009  
[http://www.xilinx.com/support/documentation/ip\\_documentation/lmb\\_v10.pdf](http://www.xilinx.com/support/documentation/ip_documentation/lmb_v10.pdf).
- [12]. Xilinx Company, "Spartan-3 Generation Configuration ",User Guide UG332 (v1.5), March 16, 2009  
[http://www.xilinx.com/support/documentation/user\\_guides/ug332.pdf](http://www.xilinx.com/support/documentation/user_guides/ug332.pdf)
- [13]. United States Patent, "MULTI-BOOT CONFIGURATION OF PROGRAMMABLE DEVICES", US 7,301,822 B1,,Nov. 27, 2007.  
<http://www.freepatentsonline.com/7301822.pdf>

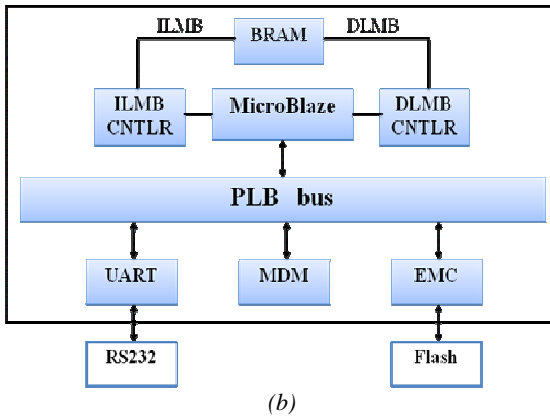
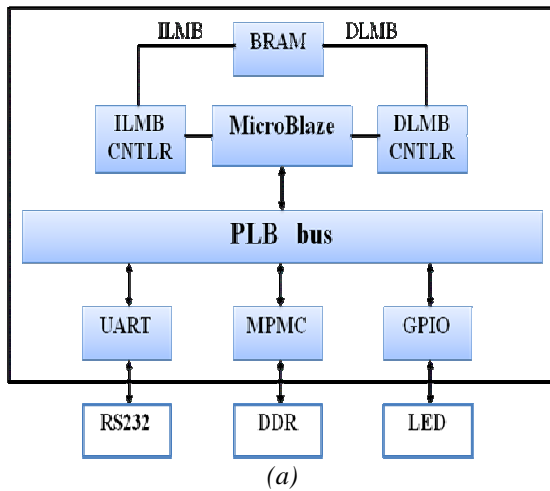


Figure 2.  
(a) First Embedded Design System  
(b) second Embedded Design System

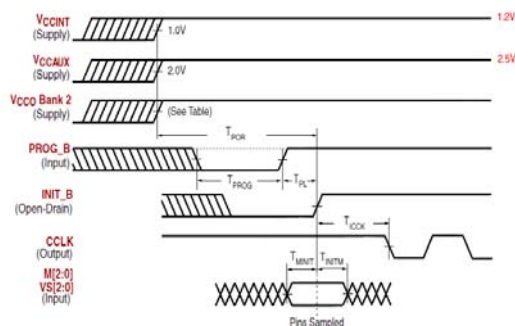


Figure 5. Power-Up Timing Configuration Signals

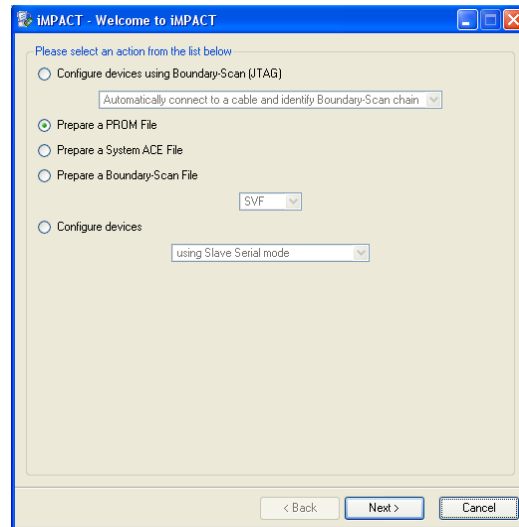


Figure 7: Prepare a MultiBoot PROM Image

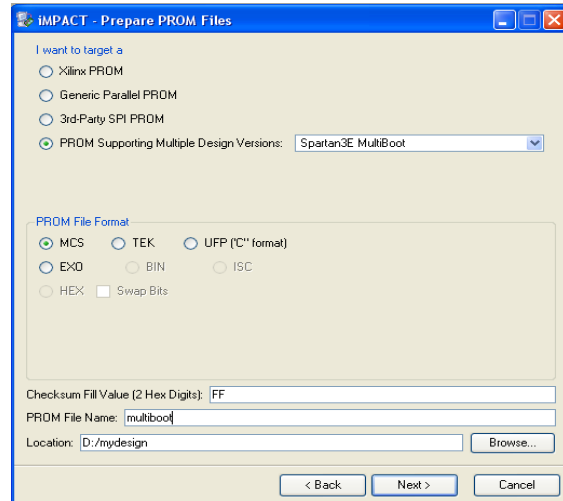


Figure 8. Select a PROM Supporting MultiBoot for Spartan-3E FPGAs

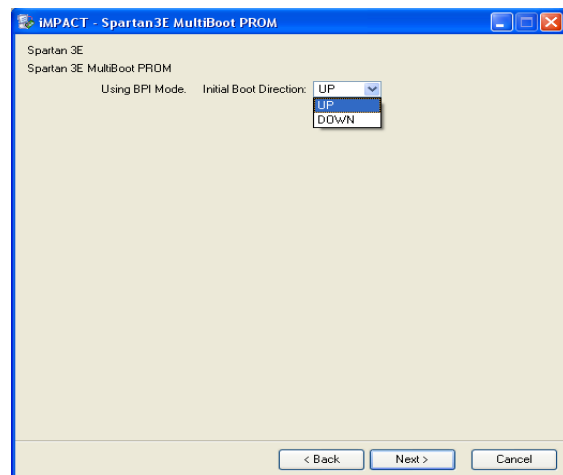


Figure 9. Select the Configuration Direction of the First MultiBoot Image

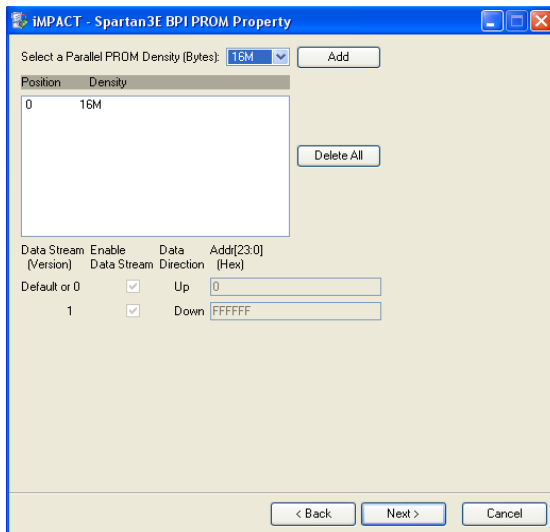


Figure 10. Select a PROM Size and Add It to the Design

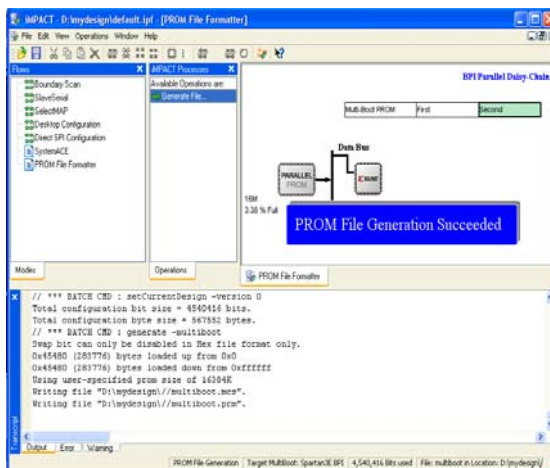


Figure 11. Generate the PROM File Using the Specified Parameters

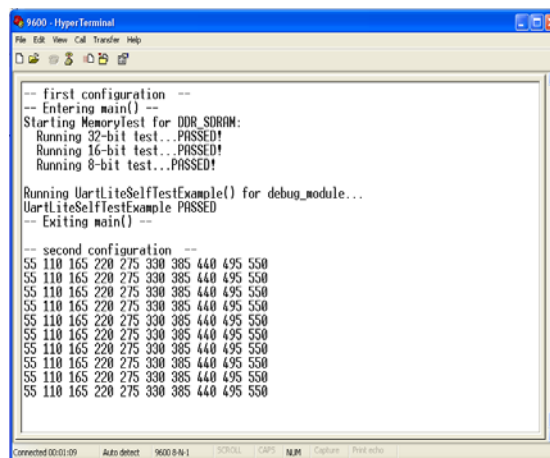


Figure 12. Multiboot design result

## AUTHOR PROFILES:

**Mazin R. Khalil** received the master degree in Electronic Engineering from university of Mosul, Iraq, in 1985 and did his Ph.D in 2011 from Aleppo university. He is an Associate Professor at Technical College of Mosul, in Iraq. He published 20 scientific researches and two books. His interests are in FPGA based control system design.

**Sarah A. Mohamed** received the Bsc. degree in Technical Computer Engineering from Technical College of Mosul, Iraq, in 2008. Currently, she is a research student of master degree in Technical College of Mosul, Iraq. Her interests in subjects of Microprocessor Architecture and FPGA embedded system.