Testing and testability analysis
Of Xilinx Virtex-4 FPGAs

A graduate project submitted in partial fulfillment of the requirements
For the degree of Master of Science in
Electrical Engineering
By
Srihitha Mogili
The project of Srihitha Mogili is approved:

__________________________________________________________________________________  
Ali Amini, Ph.D.                      Date

__________________________________________________________________________________  
Ileana Costea, Ph.D.                  Date

__________________________________________________________________________________  
Ramin Roosta, Ph.D., Chair           Date

California State University-Northridge
Dedication

I would like to thank California State University, Northridge for giving me an opportunity and facilities provided during my graduation.

I would like to thank Professor Ramin Roosta for introducing me to the world of VHDL and for his great guidance.

I also like to thank my Professor Ali Amini and Professor Ileana Costea for reviewing this project report.

Finally to mom, dad and sister, I would like to thank for your love, support, confidence and understanding me all my life which made me to achieve what I want so far in my life. Without all your support, this project would have been a blank page.
## Table of Contents

Signature page                  ii  
Dedication                      iii  
List of Figures                 vii  
Abstract                        viii  

Chapter 1: Introduction         1  

Chapter 2: Vertex-4 FPGA        2  
  2.1 General Description        2  
  2.2 Summary of Vertex-4 Family Features  4  
  2.3 System Blocks common to all Vertex-4 FPGA families  4  
    2.3.1 Xesium Clock Technology  4  
    2.3.2 Flexible Logic Resources  4  
    2.3.3 500 MHz Xtreme DSP slices  5  
    2.3.4 Integrated Block Memory  5  
    2.3.5 Select I/O technology  6  
    2.3.6 Chip sync Technology  6  
    2.3.7 Digitally Controlled Impedance  6  
    2.3.8 Active I/O Termination  6  
    2.3.9 Configuration  6  
    2.3.10 90nm copper CMOS process  7  
    2.3.11 1.2 Core voltage  7  
    2.3.12 Flip-chip Technology  7  
  2.4 System Blocks specific to the Virtex-4 FX family  7  
    2.4.1 RocketIO Multi-Gigabit Transceiver (MGT)  7  
    2.4.2 PowerPC 405 Processor RISC Core  7  
    2.4.3 PPC405 and Processor Local Bus  8  
    2.4.4 Tri-Mode Ethernet Media Access Controller  8  
  2.5 Architectural Description: Virtex-4 FPGA Array Overview  8  
  2.6 Virtex-4 FPGA Features  9  
    2.6.1 Input/output (SelectIO) Blocks  9  
    2.6.2 Configurable Logic Blocks (CLBs)  11  
    2.6.3 Block RAM  11  
    2.6.4 XtremeDSP Slices  11  
    2.6.5 Global Clocking  11  
    2.6.6 Routing Resources  12  
    2.6.7 Boundary-Scan  12  
    2.6.8 Configuration  12  
  2.7 Virtex-4 FX Family  12  
    2.7.1 RocketIO Multi-Gigabit Transceiver  12  
    2.7.2 One or Two PowerPC 405 Processor Cores  13
Chapter 3: ML410 Embedded Development Platform

3.1 Overview
3.2 Features
3.3 Block diagram
3.4 Detailed Description
  3.4.1 Configuration
  3.4.2 I/O Voltage Rails
  3.4.3 Clock Generation
  3.4.4 DDR and DDR2 Memory
    3.4.4.1 DDR Component Memory
  3.4.5 DDR2 SDRAM DIMM
    3.4.5.1 DDR2 Memory Expansion
  3.4.5.2 DDR2 Clock Signal
  3.4.5.3 DDR2 Signaling
  3.4.5.4 Tri-Mode (10/100/1000 Mb/s) Ethernet PHY
  3.4.5.5 RS-232 Ports
3.5 Introduction to JTAG
  3.5.1 Introduction to the System ACE Configuration Solution
  3.5.2 Board Bring-Up through the JTAG Interface
  3.5.3 CPU JTAG Header Pin out

Chapter 4:

4.1 Summary
4.2 Block RAM Introduction
4.3 Synchronous Dual-Port and Single-Port RAMs
  4.3.1 Data Flow
  4.3.2 Read Operation and Write Operation
  4.3.3 Operating Modes
    4.3.3.1 WRITE_FIRST or Transparent Mode (Default)
    4.3.3.2 READ_FIRST or READ-BEFORE-WRITE Mode
    4.3.3.3 NO_CHANGE Mode
4.4.1 Asynchronous Clocking
4.4.2 Synchronous Clocking
4.5 Additional Block RAM Features in Virtex-4 Devices
  4.5.1 Optional Output Registers
  4.5.2 Independent Read and Write Port Width Selection
  4.5.3 Cascadable Block RAM
4.6 Block RAM Address Mapping
List of Figures

1.1 Virtex-4 XC4VFX60 5
3.1: ML410 High level block diagram 17
3.2 ML410 Board and Front Panel Detail 18
3.3 FPGA UART and RS-232 connectivity 21
3.4 JTAG connections to the FPGA and system ACE CF controller 23
3.5 connector pin out (J9) 23
3.6 CPU JTAG header (J12) 23
4.1 WRITE_FIRST mode waveforms 26
4.2 READ_FIRST mode waveforms 26
4.3 NO_CHANGE mode waveforms 27
4.4 Block RAM logic diagram 28
4.5 Block RAM timing diagram 30
4.6 Block RAM timing model 32
4.7 Top-Level view of Block RAM ECC 33
ABSTRACT

Testing and testability analysis

Of Xilinx Virtex-4 FPGAs

BY

SRIHITHA MOGILI

MASTER OF SCIENCE IN

ELECTRICAL ENGINEERING

Virtex-4 is the current state-of-the-art radiation-tolerant SRAM-based (reconfigurable) FPGA, based on a 90 nm CMOS process. The objective of this project is to design and implement a dynamic burn in test platform that can test BRAM in Virtex-4. After creating the environmental file in Xilinx FPGA Vertex-4 FX family using EDK tool, a new project is created and a small C code is written to test the vertex-4 FPGA block RAM. In Xilinx FPGAs, a Block RAM is a dedicated two-port memory containing several kilobits of RAM. The BRAM Block is a configurable memory module that attaches to a variety of BRAM Interface Controllers. The BRAM Block structural HDL is generated by the EDK design tools based on the configuration of the BRAM interface controller IP. All BRAM Block parameters are automatically calculated and assigned by the EDK tools. The FPGA contains several or many of these blocks. Inside of each small logic block is a configurable lookup table. It is normally used for logic functions, but can reconfigure it as a few bits of RAM. By combining several of them it is called distributed RAM. The bit file is downloaded into ML410 board. BRAM is configured as instruction BRAM and data BRAM. The instruction BRAM is used to convert C code to HEX format where the processor could read it. It has one time readable memory. The data BRAM is used to store the temporary data. Such that processor releases the output of ‘C’ code and is displayed on HYPERTERMINAL and hence the BRAM is tested.
CHAPTER-1: INTRODUCTION

Enabled by the revolutionary Advanced Silicon Modular Block (ASMBL) architecture, the Virtex-4 product line is the world's first FPGA family with multiple domain optimized platforms. The initial Virtex-4 family includes three platforms; Virtex-4 LX for logic, Virtex-4 SX for very high performance signal processing, and Virtex-4 FX for embedded processing and high-speed serial connectivity. It has advanced 90nm triple-oxide technology; deliver more options, higher performance and lower power than any other FPGA family available today. With more than 100 technical innovations, the Virtex-4 family consists of 17 devices and three domain-optimized platforms each platform will offer a range of device options. With up to 200,000 logic cells and up to 500 MHz performance the Virtex-4 family delivers twice the density and up to twice the performance of any FPGA in the industry currently in production.

Xilinx ML410 is a RoHS-compliant embedded development platform based on the Xilinx Virtex™-4 XC4VFX60 FPGA. The advantage of the FPGA's dual PowerPC 405 processors, are the generous amount of FPGA fabric, and I/O capabilities that extend from the low bit-rate UARTs to the high-speed RocketIO™ Multi-Gigabit Transceivers (MGTs). When paired with the Xilinx Embedded Development Kit (EDK), and its catalog of IP peripherals, the ML410 can be used to rapidly prototype and verify system designs.

The Xilinx Embedded Development kit (EDK) is a suite of tools used to develop embedded system-on-chip designs. These designs can incorporate a wide range of soft and hard IP-cores such as microprocessors, interconnects, memories and an assortment of peripherals. The EDK tool suite provides a single environment for design, simulation, synthesis and compilation.

The BRAM Block is a configurable memory module that attaches to a variety of BRAM Interface Controllers. The BRAM Block structural HDL is generated by the EDK design tools based on the configuration of the BRAM interface controller IP. All BRAM Block parameters are automatically calculated and assigned by the EDK tools.

Report organization:

Chapter 2 presents the Xilinx virtex-4 FPGA architecture followed by ML410 driver board FPGA design. Chapter 3 explains DSP features on XILINX virtex-4 and its performance. In chapter 4 shows the practical design implementation of a sample design code and its simulation o/p result.
CHAPTER-2: VIRTEX-4 FPGA

2.1 General Description

Combining Advanced Silicon Modular Block (ASMBL™) architecture with a wide variety of flexible features, the Virtex®-4 family from Xilinx greatly enhances programmable logic design capabilities, making it a powerful alternative to ASIC technology. Virtex-4 FPGAs comprise three platform families-LX, FX, and SX—offering multiple feature choices and combinations to address all complex applications. The wide array of Virtex-4 FPGA hard-IP core blocks includes the PowerPC® processors (with a new APU interface), tri-mode Ethernet MACs, 622 Mb/s to 6.5 Gb/s serial transceivers, dedicated DSP slices, high-speed clock management circuitry, and source-synchronous interface blocks. The basic Virtex-4 FPGA building blocks are enhancements of those found in the popular Virtex, Virtex-II, Virtex-II Pro, and Virtex-II Pro X product families, so previous-generation designs are upward compatible. Virtex-4 devices are produced on a state-of-the-art 90 nm copper process using 300 mm (12-inch) wafer technology. (2)

2.2 Summary of Virtex-4 Family Features

• Three Families — LX/SX/FX
  - Virtex-4 LX: High-performance logic applications solution
  - Virtex-4 SX: High-performance solution for digital signal processing
    (DSP) applications
  - Virtex-4 FX: High-performance, full-featured solution for embedded platform
    applications

• Xesium™ Clock Technology
  - Digital clock manager (DCM) blocks
  - Additional phase-matched clock dividers (PMCD)
  - Differential global clocks

• XtremeDSP™ Slice
  - 18 x 18, two’s complement, signed Multiplier
  - Optional pipeline stages
- Built-in Accumulator (48-bit) and Adder/Subtracter

- Smart RAM Memory Hierarchy
  - Distributed RAM
  - Dual-port 18-Kbit RAM blocks
    - Optional pipeline stages
    - Optional programmable FIFO logic automatically remaps RAM signals as FIFO Signals
  - High-speed memory interface supports DDR and DDR-2 SDRAM, QDR-II, and RLDRAM-II.

- SelectIO™ Technology
  - 1.5V to 3.3V I/O operation
  - Built-in ChipSync™ source-synchronous technology
  - Digitally controlled impedance (DCI) active termination
  - Fine grained I/O banking (configuration in one bank)

- Flexible Logic Resources

- Secure Chip AES Bit stream Encryption

- 90 nm Copper CMOS Process

- 1.2V Core Voltage

- Flip-Chip Packaging including Pb-Free Package Choices

- RocketIO™ 622 Mb/s to 6.5 Gb/s Multi-Gigabit Transceiver (MGT) [FX only]

- IBM PowerPC RISC Processor Core [FX only]
  - PowerPC 405 (PPC405) Core
  - Auxiliary Processor Unit Interface (User Coprocessor)

- Multiple Tri-Mode Ethernet MACs [FX only]
2.3 System blocks common to all virtex-4 FPGA families:

2.3.1 Xesium Clock Technology

• Up to twenty Digital Clock Manager (DCM) modules
  - Precision clock deskew and phase shift
  - Flexible frequency synthesis
  - Dual operating modes to ease performance trade-off decisions
  - Improved maximum input/output frequency
  - Improved phase shifting resolution
  - Reduced output jitter
  - Low-power operation
  - Enhanced phase detectors
  - Wide phase shift range

• Companion Phase-Matched Clock Divider (PMCD) blocks

• Differential clocking structure for optimized low-jitter clocking and precise duty cycle

• 32 Global Clock networks

• Regional I/O and Local clocks

2.3.2 Flexible Logic Resources

• Up to 40% speed improvement over previous generation devices

• Up to 200,000 logic cells including:
  - Up to 178,176 internal registers with clock enable (XC4VLX200)
  - Up to 178,176 look-up tables (LUTs)
- Logic expanding multiplexers and I/O registers
  • Cascadable variable shift registers or distributed memory capability

2.3.3 500 MHz XtremeDSP Slices
  • Dedicated 18-bit x 18-bit multiplier, multiply-accumulator, or multiply-adder blocks
  • Optional pipeline stages for enhanced performance
  • Optional 48-bit accumulator for multiply accumulate (MACC) operation
  • Integrated adder for complex-multiply or multiply-add operation
  • Cascadeable Multiply or MACC
  • Up to 100% speed improvement over previous generation devices.

2.3.4 500 MHz Integrated Block Memory
  • Up to 10 Mb of integrated block memory
  • Optional pipeline stages for higher performance
  • Multi-rate FIFO support logic
    - Full and Empty Flag support
    - Fully programmable AF and AE Flags
    - Synchronous/ Asynchronous Operation
  • Dual-port architecture
  • Independent read and write port width selection (RAM only)
  • 18 Kbit blocks (memory and parity/sideband memory support)
  • Configurations from 16K x 1 to 512 x 36 (4K x 4 to 512 x 36 for FIFO operation)
  • Byte-write capability (connection to PPC405, etc.)
  • Dedicated cascade routing to form 32K x 1 memory without using FPGA routing
  • Up to 100% speed improvement over previous generation
2.3.5 SelectIO Technology

• Up to 960 user I/Os
• Wide selections of I/O standards from 1.5V to 3.3V
• Extremely high-performance
  - 600 Mb/s HSTL & SSTL (on all single-ended I/O)
  - 1 GB/s LVDS (on all differential I/O pairs)
• True differential termination
• Selected low-capacitance I/Os for improved signal integrity
• Same edge capture at input and output I/Os
• Memory interface support for DDR and DDR-2 SDRAM, QDR-II, and RLDRAM-II.

2.3.6 ChipSync Technology

• Integrated with SelectIO technology to simplify source-synchronous interfaces
• Per-bit deskew capability built in all I/O blocks (variable input delay line)
• Dedicated I/O and regional clocking resources (pin and trees)
• Built in data serializer/deserializer logic in all I/O and clock dividers
• Memory/Networking/Telecommunication interfaces up to 1 Gb/s+ DDR

2.3.7 Digitally Controlled Impedance (DCI)

2.3.8 Active I/O Termination

• Optional series or parallel termination
• Temperature compensation

2.3.9 Configuration

• 256-bit AES bit stream decryption provides intellectual property (IP) security
• Improved bit stream error detection/correction capability
• Fast SelectMAP configuration
• JTAG support
• Read back capability

2.3.10 90 nm Copper CMOS Process

2.3.11 1.2V Core Voltage

2.3.12 Flip-Chip Packaging
• Pb-Free packages available with production devices.

2.4 System Blocks Specific to the Virtex-4 FX Family

2.4.1 RocketIO Multi-Gigabit Transceiver (MGT)
• Full-duplex serial transceiver (MGT) capable of 622 Mb/s to 6.5 Gb/s baud rates
• 8B/10B, 64B/66B, user-defined FPGA logic, or no data encoding/decoding
• Channel bonding support
• CRC generation and checking
• Programmable TX pre-emphasis or pre-equalization
• Programmable RX continuous time equalization
• Programmable RX decision feedback equalization
• On-chip RX AC coupling
• RX signal detect and loss of signal indicator
• TX driver electrical idle mode
• User dynamic reconfiguration using secondary configuration bus

2.4.2 PowerPC 405 Processor RISC Core
• Embedded PowerPC 405 processor (PPC405) core
  - Up to 450 MHz operation
  - Five-stage data path pipeline
  - 16 KB instruction cache
  - 16 KB data cache
  - Enhanced instruction and data on-chip memory (OCM) controllers
- Additional frequency ratio options between

2.4.3 PPC405 and Processor Local Bus

- Auxiliary Processor Unit (APU) Interface for direct connection from PPC405 to Coprocessors in fabric
- APU can run at different clock rates
- Supports autonomous instructions: no pipeline stalls
- 32-bit instruction and 64-bit data
- 4-cycle cache line transfer

2.4.4 Tri-Mode Ethernet Media Access Controller

- IEEE 802.3 compliant
- Operates at 10, 100, and 1,000 Mb/s
- Supports tri-mode auto-detect
- Receive address filter
- Fully monolithic 1000Base-X solution with RocketIO MGT
- Implements SGMII through RocketIO MGT to external PHY device
- Supports multiple PHY (MII, GMII, etc.) interfaces through an I/O resource
- Receive and transmit statistics available through separate interfaces
- Separate host and client interfaces
- Support for jumbo frames
- Flexible, user-configurable host interface

2.5 Architectural Description: Virtex-4 FPGA Array Overview

Virtex-4 devices are user-programmable gate arrays with various configurable elements and embedded cores optimized for high-density and high-performance system designs. Virtex-4 devices implement the following functionality:

- I/O blocks provide the interface between package pins and the internal configurable logic. Most popular and leading-edge I/O standards are supported by programmable I/O blocks (IOBs). The IOBs are enhanced for source-
synchronous applications. Source-synchronous optimizations include per-bit deskew, data serializer/deserializer, clock dividers, and dedicated local clocking resources.

- Configurable Logic Blocks (CLBs), the basic logic elements for Xilinx FPGAs, provide combinatorial and synchronous logic as well as distributed memory and SRL16 shift register capability.
- Block RAM modules provide flexible 18Kbit true dual-port RAM, that are cascadable to form larger memory blocks. In addition, Virtex-4 FPGA block RAMs contain optional programmable FIFO logic for increased device utilization.
- Cascadable embedded XtremeDSP slices with 18-bit x 18-bit dedicated multipliers, integrated Adder, and 48-bit accumulator.
- Digital Clock Manager (DCM) blocks provide self-calibrating, fully digital solutions for clock distribution delay compensation, clock multiplication/division, and coarse-/fine grained clock phase shifting. Additionally, FX devices support the following embedded system functionality:
  - Integrated high-speed serial transceivers enable data rates up to 6.5 Gb/s per channel.
  - Embedded IBM PowerPC 405 Processor RISC CPU (up to 450 MHz) with the auxiliary processor unit interface
  - 10/100/1000 Ethernet media-access control (EMAC) cores.

The general routing matrix (GRM) provides an array of routing switches between each component. Each programmable element is tied to a switch matrix, allowing multiple connections to the general routing matrix. The overall programmable interconnection is hierarchical and designed to support high-speed designs.(2)

All programmable elements, including the routing resources, are controlled by values stored in static memory cells. These values are loaded in the memory cells during configuration and can be reloaded to change the functions of the programmable elements.

2.6 Virtex-4 FPGA Features

This section briefly describes the features of the Virtex-4 family of FPGAs.

2.6.1 Input/output (SelectIO) Blocks

IOBs are programmable and can be categorized as follows:

- Programmable single-ended or differential (LVDS) operation
- Input block with an optional single data rate (SDR) or double data rate (DDR) register
- Output block with an optional SDR or DDR register
- Bidirectional block
- Per-bit deskew circuitry
- Dedicated I/O and regional clocking resources
- Built in data serializer/deserializer

The IOB registers are either edge-triggered D-type flip-flops or level-sensitive latches.

IOBs support the following single-ended standards:

- LVTTL
- LVCMOS (3.3V, 2.5V, 1.8V, and 1.5V)
- PCI (33 and 66 MHz)
- PCI-X
- GTL and GTLP
- HSTL 1.5V and 1.8V (Class I, II, III, and IV)
- SSTL 1.8V and 2.5V (Class I and II)

The DCI I/O feature can be configured to provide on-chip termination for each single-ended I/O standard and some differential I/O standards.

The IOB elements also support the following differential signaling I/O standards:

- LVDS and Extended LVDS (2.5V only)
- BLVDS (Bus LVDS)
- ULVDS
- Hypertransport
- Differential HSTL 1.5V and 1.8V (Class II)
- Differential SSTL 1.8V and 2.5V (Class II)

Two adjacent pads are used for each differential pair. Two or four IOB blocks connect to one switch matrix to access the routing resources. Per-bit deskew circuitry allows for programmable signal delay internal to the FPGA. Per-bit deskew flexibly provides fine-grained increments of delay to carefully produce a range of signal delays. This is especially useful for synchronizing signal edges in source synchronous interfaces.

General purpose I/O in select locations (four per bank) is designed to be “regional clock capable” I/O by adding special hardware connections for I/O in the same locality. These regional clock inputs are distributed within a limited region to minimize clock skew between IOBs. Regional I/O clocking supplements the global clocking resources. Data serializer/deserializer capability is added to every I/O to support source synchronous interfaces. A serial-to-parallel converter with associated clock divider is included in the input path, and a parallel-to-serial converter in the output path.
2.6.2 Configurable Logic Blocks (CLBs)

A CLB resource is made up of four slices. Each slice is equivalent and contains:

- Two function generators (F & G)
- Two storage elements
- Arithmetic logic gates
- Large multiplexers
- Fast carry look-ahead chain

The function generators F & G are configurable as 4-input look-up tables (LUTs). Two slices in a CLB can have their LUTs configured as 16-bit shift registers, or as 16-bit distributed RAM. In addition, the two storage elements are either edge-triggered D-type flip-flops or level sensitive latches. Each CLB has internal fast interconnect and connects to a switch matrix to access general routing resources. (2)

2.6.3 Block RAM

The block RAM resources are 18 Kb true dual-port RAM blocks, programmable from 16K x 1 to 512 x 36, in various depth and width configurations. Each port is totally synchronous and independent, offering three “read-during-write “modes. Block RAM is cascadable to implement large embedded storage blocks. Additionally, back-end pipeline registers, clock control circuitry, built-in FIFO support, and byte write enable are new features supported in the Virtex-4 FPGA. (2)

2.6.4 XtremeDSP Slices

The XtremeDSP slices contain a dedicated 18 x 18-bit 2’s complement signed multiplier, adder logic, and a 48-bit accumulator. Each multiplier or accumulator can be used independently. These blocks are designed to implement extremely efficient and high-speed DSP applications.

2.6.5 Global Clocking

The DCM and global-clock multiplexer buffers provide a complete solution for designing high-speed clock networks. Up to twenty DCM blocks are available. To generate deskewed internal or external clocks, each DCM can be used to eliminate clock distribution delay. The DCM also provides 90°, 180°, and 270° phase-shifted versions of the output clocks. Fine-grained phase shifting offers higher resolution phase adjustment with fraction of the clock period increments. Flexible frequency synthesis provides a clock output frequency equal to a fractional or integer multiple of the input clock
frequency. Virtex-4 devices have 32 global-clock MUX buffers. The clock tree is designed to be differential. Differential clocking helps reduce jitter and duty cycle distortion.

### 2.6.6 Routing Resources

All components in Virtex-4 devices use the same interconnect scheme and the same access to the global routing matrix. Timing models are shared, greatly improving the predictability of the performance for high-speed designs.

### 2.6.7 Boundary-Scan

Boundary-Scan instructions and associated data registers support a standard methodology for accessing and configuring Virtex-4 devices, complying with IEEE standards 1149.1 and 1532.

### 2.6.8 Configuration

Virtex-4 devices are configured by loading the bit stream into internal configuration memory using one of the following modes:

- Slave-serial mode
- Master-serial mode
- Slave SelectMAP mode
- Master SelectMAP mode
- Boundary-Scan mode (IEEE-1532)

Optional 256-bit AES decryption is supported on-chip (with software bit stream encryption) providing Intellectual Property security.

### 2.7 Virtex-4 FX Family

This section briefly describes blocks available only in FX devices.

#### 2.7.1 RocketIO Multi-Gigabit Transceiver

8 – 24 Channels RocketIO Multi-Gigabit Serial Transceivers (MGTs) capable of running 622 Mb/s – 6.5 Gb/s

- Full Clock and Data Recovery
- 32-bit or 40-bit data path support
• Optional 8B/10B, 64B/66B, or FPGA-based encode/decode

• Integrated FIFO/Elastic Buffer

• Support for Channel Bonding

• Embedded 32-bit CRC generation/checking

• Integrated Comma-detect or programmable A1/A2, A1A1/A2A2 detection

• Programmable pre-emphasis (AKA transmitter equalization)

• Programmable receiver equalization

• Embedded support for:
  - Out of Band (OOB) Signaling: Serial ATA
  - Beaconing and Electrical Idle: PCI-Express™

• On-chip bypassable AC coupling for receiver

**2.7.2 One or Two PowerPC 405 Processor Cores**

• 32-bit Harvard Architecture

• 5-Stage Execution Pipeline

• Integrated 16KB Level 1 Instruction Cache and 16KB Level 1 Data Cache

• Integrated Level 1 Cache Parity Generation and Checking

• Core Connect Bus Architecture

• Efficient, high-performance on-chip memory (OCM) interface to block RAM

• PLB Synchronization Logic (Enables Non-Integer CPU-to-PLB Clock Ratios)

• Auxiliary Processor Unit (APU) Interface and Integrated APU Controller
  
  - Optimized FPGA-based Coprocessor connection
  
  - Automatic decode of PowerPC floating-point instructions and allows custom instructions (decode for up to eight instructions)
  
  - Extremely efficient microcontroller-style interfacing
2.7.3 Intellectual Property Cores

Xilinx offers IP cores for commonly used complex functions including DSP, bus interfaces, processors, and processor peripherals. Using Xilinx LogiCORE products and cores from third party AllianceCORE participants, customers can shorten development time, reduce design risk, and obtain superior performance for their designs. Additionally, our CORE Generator system allows customers to implement IP cores into Virtex-4 FPGAs with predictable and repeatable performance. It offers a simple user interface to generate parameter-based cores optimized for our FPGAs. The System Generator for DSP tool allows system architects to quickly model and implement DSP functions using handcrafted IP, and features an interface to third-party system level DSP design tools. System Generator for DSP implements many of the high-performance DSP cores supporting Virtex-4 FPGAs including the Xilinx Forward Error Correction Solution with Interleaver/De-interleaver, Reed-Solomon encoder/decoders, and Viterbi decoders. These are ideal for creating highly-flexible, concatenated codecs to support the communications market. Industry leading connectivity and networking IP cores include the electronics industry's first Advanced Switching product, leading-edge PCI Express, Serial RapidIO, Fibre Channel, and 10Gb Ethernet cores that include Virtex-4 FPGA RocketIO multi-gigabit serial interfaces. The Xilinx SPI-4.2 IP core utilizes the Virtex-4 FPGA embedded ChipSync technology to implement dynamic phase alignment for high-performance source-synchronous operation. MicroBlaze processor 32-bit core provides the industry's fastest soft processing solution for building complex systems for the networking, telecommunication, data communication, embedded and consumer markets. The MicroBlaze processor features a RISC architecture with Harvard-style separate 32-bit instruction and data busses running at full speed to execute programs and access data from both on-chip and external memory. A standard set of peripherals are also CoreConnect enabled to offer MicroBlaze processor designers compatibility and reuse. All IP cores for Virtex-4 FPGAs are found on the Xilinx IP Center Internet portal presenting the latest intellectual property cores and reference designs via Smart Search for faster access. (2)
CHAPTER-3: ML410 EMBEDDED DEVELOPMENT PLATFORM

3.1 Overview

The ML410 series of Embedded Development Platforms offer designers a versatile Virtex-4 FX platform for rapid prototyping and system verification. In addition to the more than 30,000 logic cells, over 2,400 kb of block RAM, dual IBM PowerPC 405 (PPC405) processors, and RocketIO transceivers available in the FPGA, the ML410 provides an onboard Ethernet MAC PHY, DDR memory, multiple PCI bus slots, and standard front panel interface ports within an ATX form factor motherboard. An integrated System ACE Compact Flash (CF) controller is deployed to perform board bring-up and to load applications from the Compact Flash card.(3)

3.2 Features

The list below is a superset of features in the ML410 series. Appendix A, “Board Revisions” describes the differences between the ML410-P and ML410 model boards.

- ATX form factor motherboard and ATX-compliant power supply
- 32-bit component DDR memory and 64-bit DDR2 DIMM
- 512 MB Compact Flash (CF) card and System ACE CF controller for configuration
- Two onboard 10/100/1000 Ethernet PHYs with RJ-45 connectors
- PCI Express interface and MIC2592B PCI Express power controller
- Two UARTs with RS-232 connectors
- VGA graphics interface
- LEDs, LCD*, and switches
- 32/33 PCI subsystem
  - Two 3.3V slots and two 5V slots
  - ALi South Bridge SuperIO controller
    - PS/2 mouse and keyboard connectors
    - 3.5mm headphone and microphone connectors
    - Two USB peripheral ports and one parallel port
    - General purpose I/O (GPIO)
    - Flash memory interface
- Two serial ATA connectors
- Xilinx Personality Module (XPM) interface for access to:
  - RocketIO transceivers
  - SPI4.2
  - GPIO
- Power
- JTAG and trace debug ports
- Encryption battery
- Fan controller
- Onboard power regulators for all necessary voltages
- IIC/SMBus interface
  - LTC1694 SMBus accelerator
  - RTC8566 Real Time Clock (RTC)
  - 64 kb 24LC64 EEPROM
  - LM87 voltage/temp monitor
  - DDR2 DIMM SPD EEPROM
  - SPI EEPROM*
- High-speed I/O through RocketIO transceivers

3.3 BLOCK DIAGRAM

Figure 3.1: ML410 High level block diagram
Figure 3.2 ML410 Board and Front Panel Detail
3.4 DETAILED DESCRIPTION

3.4.1 Configuration

ML410 platforms support configuration in JTAG mode only. Configuration can be accomplished by using a Xilinx download cable (such as Parallel Cable IV or Platform Cable USB) or by using the onboard System ACE Compact Flash solution.

3.4.2 I/O Voltage Rails

The FPGA requires different banking voltages that are set based on the I/O voltage interface requirements of each device directly connected to the FPGA. The Virtex-4 FPGA I/O can be configured to use different I/O standards such as SSTL18 as required on the DDR2 DIMM interface. (3)

3.4.3 Clock Generation

ML410 boards are equipped with two crystal oscillator sockets (X6 and X10) each wired for standard LVTTL-type oscillators. Both sockets accept half- and full-size oscillators.

X6 is populated with a 100 MHz oscillator that provides the system clock. This system clock is typically used to generate multiple other clocks with varying frequencies and phases within the FPGA fabric by using the Virtex-4 DCMs. The FPGA also generates and drives clocks required by the DDR memory, DDR2 DIMM memory, and PCI bus interfaces. If required, a second user clock can be brought into the FPGA by installing a second oscillator in the X10 socket. (3)

High-precision clock signals can be supplied to the FPGA using differential clock signals brought in through 50Ω SMA connectors. A single-ended clock can be connected to USER_SMA_CLK_P. Two additional single-ended clocks can be supplied through the XPM connectors. Furthermore, ML410 boards are equipped with several high-precision clocks for driving the high-speed RocketIO transceivers (not available on ML410-P model boards). These clocks can also be used to drive the global clock nets of the FPGA.

3.4.4 DDR and DDR2 Memory

ML410 platforms have two types of double data rate (DDR) memory, two component DDRs and a DDR2 SDRAM DIMM. The two memory systems are independent and enable users to build independent systems.(3)

3.4.4.1 DDR Component Memory

The board contains 64 MB of DDR SDRAM (U42 and U43). Each chip is 16 bits wide and together form a 32-bit data bus. All DDR SDRAM signals are terminated through
47Ω resistors to a 1.25V VTT reference voltage. The board is designed for matched length traces across all DDR control and data signals except clocks

a) DDR Component Clock Signal

The DDR component clock signals are broadcast from the FPGA as a single differential pair that drives both DDR chips. The delay on the clock trace is designed to match the delay of the other DDR control and data signals. The DDR component clock is also fed back to the FPGA to allow for clock deskews using Virtex-4 DCMs. The board is designed so that the DDR clock signal reaches the FPGA clock feedback pin at the same time it arrives at the DDR components.

b) DDR Component Signaling

The FPGA DDR interface uses SSTL2 signaling. All signals are controlled impedance and are SSTL2 terminated. Note that the DDR1_DQ signal names do not correlate as the FPGA uses IBM notation, big endian, while the DDR components use Intel notation, little endian.

3.4.5 DDR2 SDRAM DIMM

The DDR2 DIMM is a standard 240-pin DIMM socket, supporting standard computer DDR2 memory. ML410 platforms are shipped with a single-rank registered 256 MB PC2-3200 DDR2-400 Dual Inline Memory Module (DIMM). The DDR2 DIMM uses nine 32M x 8 DDR2 SDRAM devices with 14-row address lines, 10-column address lines, and two bank address lines. Read and write access is programmable in burst lengths of 4 or 8. The memory module inputs and outputs are compatible with SSTL18 signaling. Serial Presence Detect (SPD) using a SMBus interface to the DDR DIMM is also supported.

The DDR2 DIMM memory interface includes a 64-bit wide data path to the DDR2 DIMM, thus ECC is not supported.

3.4.5.1 DDR2 Memory Expansion

The DDR2 interface is very flexible and can accommodate different DDR2 memory requirements, such as increased memory size. The DDR2 interface core delivered with EDK supports registered DDR2 memory interfaces.

3.4.5.2 DDR2 Clock Signal

The DDR2 clock signal is broadcast from the FPGA as a single differential pair that drives a clock fan-out chip, which then drives the DDR2 DIMM. The clock fan-out chip provides support for registered DIMMs. The delay on the clock trace is designed to match the delay of the other DDR2 control and data signals. The DDR2 clock is also fed
back to the FPGA to allow for clock deskew using Virtex-4 DCMs. The board is designed so that the DDR2 clock signal reaches the FPGA clock feedback pin at the same time as it arrives at the DDR2 DIMM.

### 3.4.5.3 DDR2 Signaling

All DDR2 SDRAM signals are terminated through 47Ω resistors to a 0.9V VTT reference voltage. The board is designed for matched length traces across all DDR2 control and data signals, except clocks. The FPGA DDR2 interface supports SSTL18 signaling. All DDR2 signals are controlled impedance and are SSTL18 terminated.

### 3.4.5.4 Tri-Mode (10/100/1000 Mb/s) Ethernet PHY

ML410 platforms feature two Ethernet PHYs that support MII, RGMII, and SGMII interfaces. The PHY devices have independent MDIO and MDC controls and individual 25 MHz clock crystals.(3)

### 3.4.5.5 RS-232 Ports

Two RS-232 ports are connected to the FPGA (U37) through independent MAX3232 transceivers (U7 and U46)

![Figure 3.3 FPGA UART and RS-232 connectivity](image-url)
The FPGA RS-232 ports are wired as DTE and meet the EIA/TIA 574 standard. The ports are accessible via two DB9M connectors integrated on the P1 connector assembly.

3.5 Introduction to JTAG

JTAG (Joint Test Action Group) is a simple interface that provides for many uses. On ML410 platforms, the primary uses include configuration of the FPGA, debugging software (similar to the CPU debug interface), and debugging hardware using the Chip Scope Integrated Logic Analyzer (ILA). The Virtex-4 family is fully compliant with the IEEE Standard 1149.1 Test Access Port and Boundary-Scan Architecture and includes all mandatory elements defined in the standard.

These elements include the Test Access Port (TAP), the TAP controller, the instruction register, the instruction decoder, the Boundary-Scan register, and the bypass register. The Virtex-4 family also supports some optional instructions; the 32-bit identification register, and a configuration register in full compliance with the standard.(3)

3.5.1 Introduction to the System ACE Configuration Solution

The Xilinx System ACE Compact Flash (CF) configuration controller allows a Type I or Type II Compact Flash card to program the FPGA through the JTAG port. Both hardware and software data can be downloaded through the JTAG port. The System ACE controller supports up to eight configuration images on a single Compact Flash card. The configuration address switches allow the user to choose which of the eight N Configuration images to use. System ACE error and status LEDs indicate the operational state of the System ACE controller:

- A blinking red error LED indicates that no Compact Flash card is present
- A solid red error LED indicates an error condition during configuration
- A blinking green status LED indicates a configuration operation is ongoing
- A solid green status LED indicates a successful download

3.5.2 Board Bring-Up through the JTAG Interface

The System ACE CF controller is located between the JTAG connector and the FPGA, and Passes the JTAG signals back and forth between the two. During configuration, the System ACE CF controller has full control of the JTAG signals. Figure 2.4 shows the connections between the JTAG connector, System ACE CF controller, and the FPGA. The CPU JTAG header (J12) is used to access the JTAG interface when J27 is jumpered. The pin out shown in Figure 2.5 is compatible with the Parallel Cable IV (PC4) JTAG programming solution. The J9 header is used when programming the FPGA by way of the PC4 download cable.
3.5.3 CPU JTAG Header Pin out

Figure 2.6 shows J12, the 16-pin header that can be used to debug the software operating in the CPU with debug tools such as Parallel Cable IV or third party tools.
CHAPTER-4: BLOCK RAM

4.1 Summary

The Virtex®-4 FPGA block RAMs are similar to the Virtex-II and Spartan™-3 FPGA block RAMs. Each block RAM stores 18 Kbits of data. Write and Read are synchronous operations; The two ports are symmetrical and totally independent, sharing only the stored data. Each port can be configured in any “aspect ratio” from 16Kx1, 8Kx2, to 512x36, and the two ports are independent even in this regard. The memory content can be defined or cleared by the configuration bit stream. During a write operation the data output can either reflect the new data being written, or the previous data now being overwritten, or the output can remain unchanged. (1)

Virtex-4 FPGA enhancements of the basic block RAM include:

- The user can invoke a pipeline register at the data read output, still inside the block RAM. This allows a higher clock rate, at the cost of one additional clock period latency.
- Two adjacent block RAMs can be combined to one deeper 32Kx1 memory without any external logic or speed loss.
- Ports 18 or 36 bits wide can have individual write enable per byte. This feature is used for interfacing to an on-chip (PPC405) microprocessor.
- Each block RAM contains optional address sequencing and control circuitry to operate as a built-in Multi-rate FIFO memory. The FIFO can be 4K deep and 4 bits wide, or 2Kx9, 1Kx18, or 512x36. Write and read ports have identical width. The two free-running clocks can have completely unrelated frequencies (asynchronous relative to each other). Operation is controlled by the read and writes enable inputs. FULL and EMPTY outputs signal the extreme conditions, without a possibility of errors or glitches. Programmable ALMOSTFULL and ALMOSTEMPTY outputs can be used for warning to simplify the external control of the write and read operation, especially the maximum clock rate. (1)

Additional Virtex-4 FPGA block RAM features include:

- All output ports are latched. The state of the output port does not change until the port executes another read or write operation.
- All inputs are registered with the port clock and have a setup-to-clock timing Specification.
- All outputs have a read function or a read-during-write function, depending on the state of the WE pin. The outputs are available after the clock-to-out timing interval. The read-during-write outputs have one of three operating modes: WRITE_FIRST, READ_FIRST, and NO_CHANGE.
A write operation requires one clock edge.
A read operation requires one clock edge.
DO has an optional internal pipeline register.
Data input and output signals are always described as buses; that is, in a 1-bit width configuration, the data input signal is DI [0] and the data output signal is DO [0].

4.2 Block RAM Introduction

In addition to distributed RAM, Virtex-4 devices feature a large number of 18 Kb block RAM memories. True Dual-Port™ RAM offers fast blocks of memory in the device. Block RAMs are placed in columns, and the total number of block RAM memory depends on the size of the Virtex-4 device. The 18 Kb blocks are cascadable to enable a deeper and wider memory implementation, with a minimal timing penalty. Embedded dual- or single-port RAM modules, ROM modules, synchronous FIFOs, and data width converters are easily implemented using the Xilinx CORE Generator software memory modules. Asynchronous FIFOs can be generated using the CORE Generator tool FIFO Generator module. The synchronous or asynchronous FIFO implementation does not require additional CLB resources for the FIFO control logic since it uses dedicated hardware resources. (1)

4.3 Synchronous Dual-Port and Single-Port RAMs

4.3.1 Data Flow

The 18 Kb block RAM dual-port memory consists of an 18 Kb storage area and two completely independent access ports, A and B. The structure is fully symmetrical, and both ports are interchangeable. Data can be written to either or both ports and can be read from either or both ports. Each write operation is synchronous, each port has its own address, data in, data out, clock, clock enable, and write enable. The read operation is synchronous and requires a clock edge. There is no dedicated monitor to arbitrate the effect of identical addresses on both ports. It is up to the user to time the two clocks appropriately. When a block RAM port is enabled, all address transitions must meet the setup/hold time of the ADDR inputs with respect to the port clock.

4.3.2 Read Operation and Write Operation

The read operation uses one clock edge. The read address is registered on the read port, and the stored data is loaded into the output latches after the RAM access time.

A write operation is a single clock-edge operation. The write address is registered on the write port, and the data input is stored in memory.

4.3.3 Operating Modes
There are three modes of a write operation. The “read during write” mode offers the flexibility of using the data output bus during a write operation on the same port. Output mode is set during device configuration. These choices increase the efficiency of block RAM memory at each clock cycle. Three different modes are used to determine data available on the output latches after a write clock edge: WRITE_FIRST, READ_FIRST, and NO_CHANGE. Mode selection is set by configuration. One of these three modes is set individually for each port by an attribute. The default mode is WRITE_FIRST.

4.3.3.1 WRITE_FIRST or Transparent Mode (Default)

In WRITE_FIRST mode, the input data is simultaneously written into memory and stored in the data output (transparent write), as shown in Figure 4-1.

4.3.3.2 READ_FIRST or READ-BEFORE-WRITE Mode

In READ_FIRST mode, data previously stored at the write address appears on the output latches, while the input data is being stored in memory (read before write), as shown in Figure 4-2.
4.3.3.3 NO_CHANGE Mode

In NO_CHANGE mode, the output latches remain unchanged during a write operation. As shown in Figure 4-4, data output is still the last read data and is unaffected by a write operation on the same port. NO_CHANGE mode is not supported in 32K x 1 RAM configuration.

![Figure 4.3 NO_CHANGE mode waveforms](image)

4.3.3.3 NO_CHANGE Mode

Virtex-4 FPGA block RAM is a true dual-port RAM, where both ports can access any memory location at any time. When accessing the same memory location from both ports, the user must, however, observe certain restrictions, specified by the clock-to-clock setup time window. There are two fundamentally different situations: The two ports either have a common clock (“synchronous clocking”), or the clock frequency or phase is different for the two ports (“asynchronous clocking”).

4.4.1 Asynchronous Clocking

Asynchronous clocking is the more general case, where the active edges of both clocks do not occur simultaneously:

- There are no timing constraints when both ports perform a read operation.
- When one port performs a write operation, the other port must not read- or writes access the same memory location by using a clock edge that falls within the specified forbidden clock-to-clock setup time window. If this restriction is ignored, a read operation could read unreliable data, perhaps a mixture of old and new data in this location; a write operation could result in wrong data stored in this location. There is, however, no risk of physical damage to the device. If a read and write operation is performed, the write will store valid data at the write location. The clock-to-clock setup timing parameter is specified together with other Block RAM switching characteristics.
4.4.2 Synchronous Clocking

Synchronous clocking is the special case, where the active edges of both port clocks occur simultaneously:

- There are no timing constraints when both ports perform a read operation.
- When one port performs a write operation, the other port must not write into the same location, unless both ports write identical data.
- When one port performs a write operation, the write operation succeeds; the other port can reliably read data from the same location if the write port is in READ_FIRST mode. DATA_OUT will then reflect the previously stored data. If the write port is in either WRITE_FIRST or in NO_CHANGE mode, then the DATAOUT on the read port would become invalid (unreliable). Obviously, the mode setting of the read-port does not affect this operation.

4.5 Additional Block RAM Features in Virtex-4 Devices

![Block RAM logic diagram](image)

**Figure 4.4 Block RAM logic diagram**

4.5.1 Optional Output Registers

The optional output registers improve design performance by eliminating routing delay to the CLB flip-flops for pipelined operation. These output registers have programmable clock inversion as in CLB flip-flops. An independent clock enable input is provided for these output registers. As a result the output data registers hold the value independent of the input register operation.
4.5.2 Independent Read and Write Port Width Selection

All block RAM ports have control over data width and address depth. Virtex-4 devices extend this flexibility to each individual port where Read and Write can be configured with different data widths. If the Read port width differs from the Write port width, and is configured in WRITE_FIRST mode, then DO show valid new data only if all the write bytes are enabled. Independent Read and Write port width selection increases the efficiency of implementing a content addressable memory (CAM) in block RAM. Excluding the built-in FIFO, this option is available for all RAM port sizes and modes.

4.5.3 Cascadable Block RAM

Combining two 16K x 1 RAMs to form one 32K x 1 RAM is possible in the Virtex-4 block RAM architecture without using local interconnect or additional CLB logic resources. NO_CHANGE mode is not supported in 32K x 1 RAM configuration. Any two adjacent block RAMs can be cascaded to generate a 32K x 1 block RAM. Increasing the depth of the block RAM by cascading two block RAMs is available only in the 32K x 1 mode.

4.6 Block RAM Address Mapping

Each port accesses the same set of 18,432 memory cells using an addressing scheme dependent on the width of the port. The physical RAM locations addressed for a particular width are determined using the following formula

\[
\text{END} = ((\text{ADDR} + 1) \times \text{Width}) - 1
\]

\[
\text{START} = \text{ADDR} \times \text{Width}
\]

4.7 Block RAM Applications

4.7.1 Creating Larger RAM Structures

Block RAM columns have special routing to create wider/deeper blocks with minimal routing delays. Wider or deeper RAM structures are achieved with a smaller timing penalty than is encountered when using normal routing resources. The CORE Generator software offers the designer an easy way to generate wider and deeper memory structures using multiple block RAM instances. This program outputs VHDL or Verilog instantiation templates and simulation models, along with an EDIF file for inclusion in a design. (1)
4.8 Block RAM Timing Model

4.8.1 Block RAM Timing Characteristics

The timing diagram in Figure 4.5 describes a single-port block RAM in write-first mode without the optional output register. The timing for read-first and no-change modes are similar. For timing using the optional output register, additional clock latency appears at the DO pin. At time 0, the block RAM is disabled; EN (enable) is Low.

![Block RAM timing diagram](image)

Figure 4.5 Block RAM timing diagram

4.8.1.1 Clock Event 1

**Read Operation:** During a read operation, the contents of the memory at the address on the ADDR inputs are unchanged.

- TRCCK_ADDR before clock event 1, address 00 becomes valid at the ADDR inputs of the block RAM.
- At time TRCCK_EN before clock event 1, enable is asserted High at the EN input of the block RAM, enabling the memory for the READ operation that follows.
- At time TRCKO_DO after clock event 1, the contents of the memory at address 00 become stable at the DO pins of the block RAM.

4.8.1.2 Clock Event 2

**Write Operation:** During a write operation, the content of the memory at the location specified by the address on the ADDR inputs is replaced by the value on the DI pins and
is immediately reflected on the output latches (in WRITE-FIRST mode); EN (enable) is High.

- At time TRCCK_ADDR before clock event 2, address 0F becomes valid at the ADDR inputs of the block RAM.
- At time TRDCK_DI before clock event 2, data CCCC becomes valid at the DI inputs of the block RAM.
- At time TRCCK_WEN before clock event 2, write enable becomes valid at the WEN following the block RAM.
- At time TRCKO_DO after clock event 2, data CCCC becomes valid at the DO outputs of the block RAM

4.8.1.3 Clock Event 4

**SSR (Synchronous Set/Reset) Operation:** During an SSR operation, initialization parameter value SRVAL is loaded into the output latches of the block RAM. The SSR operation does NOT change the contents of the memory and is independent of the ADDR and DI inputs.

- At time TRCCK_SSR before clock event 4, the synchronous set/reset signal becomes valid (High) at the SSR input of the block RAM.
- At time TRCKO_DO after clock event 4, the SRVAL 0101 becomes valid at the DO outputs of the block RAM.

4.8.1.4 Clock Event 5

**Disable Operation:** Deasserting the enable signal EN disables any write, read, or SSR operation. The disable operation does NOT change the contents of the memory or the values of the output latches.

- At time TRCCK_EN before clock event 5, the enable signal becomes valid (Low) at the EN input of the block RAM.
- After clock event 5, the data on the DO outputs of the block RAM is unchanged.

4.9 Block RAM Timing Model

Figure 4.6 illustrates the delay paths associated with the implementation of block RAM. This example takes the simplest paths on and off chip. This timing model demonstrates how and where the block RAM timing parameters are used.

- **NET** = Varying interconnect delays
- **TIOP1** = Pad to I-output of IOB delay
- **TIOOP** = O-input of IOB to pad delay
- **TBCCKO_O** = BUFGCTRL delay
4.10 Built-in Block RAM Error Correction Code

Two vertically adjacent block RAMs can be configured as a single 512 x 64 RAM with built in hamming error correction, using the extra eight bits in the 72-bit wide RAM. The operation is transparent to the user. The eight protection bits are generated during each write operation, and are used during each read operation to correct any single error, or to detect (but not correct) any double error. Two status outputs indicate the three possible read results: No error, single error corrected, double error detected. The read operation does not correct the error in the memory array; it only presents corrected data on DO. This error correction code (ECC) configuration option is available with almost all block RAM pairs as long as the lower RAM is instantiated in an even numbered row. However, the ECC configuration cannot use the one block RAM immediately above or below the PowerPC® 405 blocks in Virtex-4 devices.

The functionality of the block RAM is changed when using the ECC mode.

- The two block RAM ports still have independent address, clocks, and enable inputs, but one port is a dedicated write port, and the other is a dedicated read port.
- DO represents the read data after correction.
- DO stays valid until the next active read operation.
- Simultaneous reading and writing, even with asynchronous clocks, is allowed, but requires careful clock timing if read and write addresses are identical.
- The READ_FIRST or WRITE_FIRST modes of the normal block RAM operation are not applicable to the ECC configuration. (1)
4.11 Top-Level View of the Block RAM ECC Architecture

Figure 4.7 shows the top-level view of a Virtex-4 FPGA block RAM in ECC mode.

Figure 4.7 Top-Level view of Block RAM ECC
REFERENCES

1) Virtex-4 FPGA user guide UG070(v2.6) DEC 1,2008
2) Virtex-4 FPGA user guide DS112(v3.1) AUG30,2010
3) ML 410 Embedded development platform user guide (UG085-V1.7.2) DEC 11, 2008
4) Block RAM user guide from Xilinx DS 444 March2,2010
5) Xilinx EDK tutorial and notes by Jonathan W. Donaldson, Sandia National Laboratories
6) Qualification & Screening Issues of Xilinx Virtex-4 FPGAs for Space Applications, Ramin Roosta, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California.
CREATING AN EDK PROJECT

The project creating will ultimately be configured to display “Hello World!” over the board’s standard RS-232 serial port using a very simple C application.

- When EDK opens select “Blank XPS Project”.
- Pick a location for Xilinx Microprocessor Project (XMP)
- Choose the correct “Target device” options for the FPGA
- Leave the Microprocessor Hardware Specification (MHS) file and Repository paths blank. Some Xilinx boards (e.g. XUPV2P/ML310/ML410) will come with own additional Intellectual Property (IP) core libraries. These libraries contain IP cores that are not available in the Xilinx “standard” library that comes with the EDK installation. Click OK as shown in Fig A.1

![Create New XPS Project](image)

**Figure A.1 Creating a new XPS project**

- Then it will be presented with the default EDK interface. Let’s follow EDK’s instructions presented in the console log window and start adding some useful IPs to our design.
- Expand the “Processor” list under the “IP Catalog” tab in the left panel, right click on the “ppc405” and select “Add IP”. It is shown in image A.2. Since the Vertex-
4FX FPGA contains two PowerPC 405 (PPC405) processors, EDK will require to add IP cores2 for both of them even if do not intend to use both in the design.

- Add the following IP cores in the same fashion as you did for the PPC405:
  a) **proc sys reset** - This is a reset control block which is very useful. It has a port for an external reset input which, when toggled, will perform resets to the internal PPC components, the processor bus structures, and the peripherals in a very specific order. It also has internal reset ports which connect to the processor itself. The PPC has its own set of “reset request” signals which can tell the reset control block to reset the system as well.

![Figure A.2 Adding the required IP](image)

File” under the “Project” tab in the left panel and add the following lines.
BEGIN proc_sys_reset
PARAMETER INSTANCE = reset_block
PARAMETER HW_VER = 1.00.a
END

b) **dcm module** - Digital Clock Manager (DCM) module is required for two reasons: the proc sys reset module requires a DCM Locked output signal for its logic and It’s good design practice to buffer the external input clock and running external clock through a DCM automatically does this and makes the design more versatile if different clock rates are required in the future.
c) **plb v34** - The Processor Local Bus (PLB) is part of the International Business Machines (IBM) Core Connect Bus Architecture specification and is the high speed data interface to the PPC core. All of our peripherals and system memory will communicate with the processor using this bus.

d) **opb v20** - The On-Chip Peripheral Bus (OPB) is another piece of the IBM CoreConnect Bus Architecture specification.

e) **plb2opb bridge** - The PLB2OPB bridge will allow low-speed peripherals to interact with the processor over the PLB (the PLB is the only physical interface to the PPC for peripherals). This bridge will act as a “slave” on the PLB and a “master” on the OPB.

f) **bram block** - Block Random Access Memory (BRAM) is used as the processor’s main memory in this design. However, in a larger design with larger applications most likely want to use Double Data Rate (DDR) memory (if the board has a DDR slot available). Xilinx provides DDR controller cores as well.

g) **plb bram if cntrl** - This module will be the interface BRAM. If the BRAM is used as main system memory or cache space one of these controllers are needed.

h) **opb uartlite** - This is an RS-232 UART core that will use to transfer characters to a standard VT100 terminal (e.g. HyperTerminal, TeraTerm Pro, etc.). Compared to the Xilinx 16550 compliant UART the UARTlite is very simplistic.

- First, connect the JTAG PPC Controller to both of the PPC cores. By using only one PPC core, EDK will throw an error if the JTAG controller is not connected to the one that is not being used. Nothing else will have to be connected to the second PPC core. Do this by connecting the respective JTAG controller ports under the bus connection column to each processor. See image 2.4.

- Now connect main system buses. In this design both data and instructions will pass over the same Processor Local Bus (PLB) bus. So select the PLB instance name under the Bus Connection column for the Data Processor Local Bus (DPLB) and Instruction Processor Local Bus (IPLB) bus interface’s for ppc405 as shown in Fig A.4.

- Now, connect the PLB2OPB Bridge as a slave on the PLB and a master on the On Chip Peripheral Bus (OPB).
Now attach the RS232 UARTlite as a slave on the OPB as shown in Fig 2.4.
- Connect the PLB Block Random Access Memory (BRAM) Interface (IF) Controller as a slave on the PLB. And also connect the BRAM block itself up to the BRAM IF controller on PORTA.

  Also right-click on the PLB BRAM IF controller, select “Configure IP. . . ”, select the “System” tab, select “PLB” from the configuration list, and change PLB clock period to match the period of whatever speed clock connected it to.

- Change the tab near the top of the window from “Bus Interface” to ”Ports”.

  Expand the Digital Clock Manager (DCM) instance block and attach the RST input to “net gnd” Now, for the “CLKin” port select “Make External” for the connection - this is the input from the onboard oscillator. For the CLK0 output choose “New Connection” from the drop down list. For the DCM’s clock feedback input (CLKFB) selects the signal name that is being used for the CLK0 output. Finally, make a new connection for the LOCKED output as well as shown in Fig A.5. The DCM port list should look like image 2.5. Right click on the DCM instance and choose “Configure IP...”. In the “DCM” section, change the “Configuration Startup Wait” option to “True”. This will cause the design to be held in reset until the DCM has locked. Additionally, change the “Input Clock Period” parameter to match input clock period (in nanoseconds).

![Figure A.5 DCM port Connections](image_url)
Next, connect up the system reset block to the PPC. For the “Slowest sync clk” ports just choose the slowest clock frequency that will actually be used by any device in the entire system. For this design, that frequency will be the CLK0 output of the DCM. For “Ext Reset In” port select “Make External”.

Under the PPC instance, create new connections for the three “C405RSTCHIP/CORE/ SYSSYSRESETREQ” ports and connect them to the reset controller’s “Core/Chip/System Reset Req” input ports as mentioned earlier. Finally, connect the “RSTC405RESETCHIP/CORE/SYS” input ports of the PPC to the “Rstc405resetcore/chip/sys” output ports of the reset block. Also, connect “CPMC405CLOCK” (i.e. the PPC’s main clock) to the DCM’s CLK0 output signal.

Now, connect the PLB and OPB clock inputs to the DCM’s CLK0 output. Additionally, connect the PLB and OPB’s “SYS Rst” input port to the “Bus Struct Reset” output of the system reset controller as shown in Fig A.6. Also, right-click on the PLB instance and uncheck the box for “Include DCR Interface and Error Registers”.

Select “Make External” for both the TX and RX lines of the UART. For any device as slow as a UART an interrupt is really the only efficient way for information exchange between it and the processor. Finally right-click on the UART to configure it, and change the baud rate to 115200, and change ”Use Parity” to False.

Now, open up the “system.mhs” file by changing tabs in the left panel of the EDK window from “IP Catalog” to “Project” and double-click on “MHS File” to open it. A new file is created prior to even creating an EDK project, add the file to a new design, and it will update the Graphical User Interface (GUI) port and bus interface display appropriately. In other words, instead of using the GUI to generate the MHS file we could have used the MHS file to generate the GUI connections. First, locate the “opb uartlite” instance and change the C CLK FREQ parameter to whatever the OPB’s clock rate is.
Navigate back to the “System Assembly View” window and change filters from “Ports” to “Addresses”. A list of buses, controllers, and peripherals are seen. (5)

*Figure A.5 PLB and OPB port connections*
APPENDIX-B

Simulating EDK design and testing BRAM

- To do this, select the “Applications” tab in the left EDK panel. Double-click on “Add Software Application Project...”, name your project and be sure it is set to be executed on “ppc405 0”. See image 2.10 below. Say OK to that window and then right-click on the new project name in the left panel and click “Mark to Initialize BRAMs”. This tells the Xilinx EDK flow tools to embed this project compiled application binary code into the FPGA’s configuration bit stream so that it can be loaded into the BRAM along with the FPGA physical design itself.

- Now, right-click on “Sources” under the project name, select “Add New File...”, and select a location for the program’s main file. Double click on new source file to open it and let’s add some code. For BRAM test app the regular C code is written which multiplies the two numbers as shown in the following Fig B.1

```c
#include "xparameters.h"
#include "stdio.h"
#include "util.h"

int main (void) {

    printf("--MULTIPLICATION --\r\n");

    int a = 14, b = 22, c;

    c = a + b;

    printf("a = %d b = %d c = %d\n",a,b,c);

    /*
     * MemoryTest routine will not be run for the memory at
     * 0x00000000 |plp_bram_if_entire|
     * because it is being used to hold a part of this application program
     */

    printf("-- Exiting main() --\r\n");

    return 0;
}
```

Figure B.1 General ‘C’ code to test BRAM
• There is actually one VERY important thing to note here and that is the fact that we are calling print() and not the ‘C’ standard library printf() function. It seems like a very small difference but it is actually a HUGE difference. The print() function is a Xilinx function and it uses the bare minimum number of instructions to write characters out to the UART.

• Go to Project: Testapp_memory and right click to build the project and the simulation report is as follows.

At Local date and time: Thu Nov 04 22:37:09 2010
xbash -q -c "cd /cygdrive/c/EDK/EDKexamples/BRAM1/; /usr/bin/make -f system.make TestApp_Memory_program; exit;" started...
make: Nothing to be done for `TestApp_Memory_program'.
Done!

• Open HyperTerminal using on your PC/Linux box and configure it. Connect the serial cable from the board to your PC/Linux box as well. And also connect up the Xilinx programming cable to the board and power on the board.

• After the previous step was successful let’s integrate our test application’s binary into the hardware design’s bitstream. Do this by clicking the “Device Configuration” menu and selecting “Update Bitstream”. The simulation report is obtained.

At Local date and time: Thu Nov 04 22:39:16 2010
xbash -q -c "cd /cygdrive/c/EDK/EDKexamples/BRAM1/; /usr/bin/make -f system.make init_bram; exit;" started...
make: Nothing to be done for `init_bram'.
Done!

• Then click the “Device Configuration” menu again and select “Download Bitstream”. This will target the board and download your design to the FPGA. With a bit of luck you should see “Hello World!” displayed in your serial port application’s terminal window. The simulation report is as follows:

At Local date and time: Thu Nov 04 22:40:21 2010
xbash -q -c "cd /cygdrive/c/EDK/EDKexamples/BRAM1/; /usr/bin/make -f system.make download; exit;" started...

Downloading Bitstream onto the target board

******************************************************************************
impact -batch etc/download.cmd
Release 8.2i - iMPACT I.31
Copyright (c) 1995-2006 Xilinx, Inc. All rights reserved.
// *** BATCH CMD : setMode -bs

// *** BATCH CMD : setCable -port auto

AutoDetecting cable. Please wait.

Connecting to cable (Parallel Port - LPT1).

Checking cable driver.

Driver windrvr6.sys version = 7.0.0.0. LPT base address = 0378h.

ECP base address = 0778h.

Cable connection failed.

Connecting to cable (Parallel Port - LPT2).

Checking cable driver.

Driver windrvr6.sys version = 7.0.0.0.Cable connection failed.

Connecting to cable (Parallel Port - LPT3).

Checking cable driver.

Driver windrvr6.sys version = 7.0.0.0.Cable connection failed.

Connecting to cable (Parallel Port - LPT4).

Checking cable driver.

Driver windrvr6.sys version = 7.0.0.0.Cable connection failed.

Connecting to cable (Usb Port - USB21).

Checking cable driver.

Driver xusbdfwu.sys version: 1021 (1021).

Driver windrvr6.sys version = 7.0.0.0.Calling setinterface num=0, alternate=0.

DeviceAttach: received and accepted attach for:

    vendor id 0x3fd, product id 0x8, device handle 0x18e0038

Cable PID = 0008.
Max current requested during enumeration is 280 mA.

Cable Type = 3, Revision = 0.

Setting cable speed to 6 MHz.

Cable connection established.

Firmware version = 1021.

CPLD file version = 0012h.

CPLD version = 0012h.

// *** BATCH CMD : identify

Identifying chain contents ....Version is 0010

'1': Manufacturer's ID = Xilinx xc4vfx60, Version : 2

PMSPEC -- Overriding Xilinx file <c:/Xilinx/virtex4/data/virtex4.acd> with local file <c:/Xilinx/virtex4/data/virtex4.acd>

INFO:iMPACT:1777 -

Reading c:/Xilinx/virtex4/data/xc4vfx60.bsd...

INFO:iMPACT:501 - '1': Added Device xc4vfx60 successfully.

----------------------------------------------------------------------

Version is 0000

'2': Manufacturer's ID = Xilinx xccace, Version : 0

INFO:iMPACT:1777 -

Reading c:/Xilinx/acecf/data/xccace.bsd...

INFO:iMPACT:501 - '1': Added Device xccace successfully.

----------------------------------------------------------------------

done.
Chain TCK freq = 16700000.
Validating chain...
Boundary-scan chain validated successfully.
Elapsed time = 1 sec.

// *** BATCH CMD : identifyMPM
Elapsed time = 0 sec.

// *** BATCH CMD : assignFile -p 2 -file "implementation/download.bit"

'2': Loading file 'implementation/download.bit' ...
done.
INFO:iMPACT:501 - '2': Added Device xc4vfx60 successfully.

----------------------------------------------------------------------
----------------------------------------------------------------------
----------------------------------------------------------------------

// *** BATCH CMD : program -p 2

Chain TCK freq = 16700000.
Validating chain...
Boundary-scan chain validated successfully.

'2': Programming device...
done.

'2': Reading status register contents...
   CRC error : 0
   Decryptor security set : 0
   DCM locked : 1
   DCI matched : 1
End of startup signal from Startup block : 1
status of GTS_CFG_B : 1
status of GWE : 1
status of GHIGH : 1
value of MODE pin M0 : 1
value of MODE pin M1 : 1
Value of MODE pin M2 : 1
Internal signal indicates when housecleaning is completed: 1
Value driver in from INIT pad : 1
Internal signal indicates that chip is configured : 1
Value of DONE pin : 1
Indicates when ID value written does not match chip ID: 0
Decryptor error Signal : 0
System Monitor Over-Temperature Alarm : 0
INFO:iMPACT:2219 - Status register values:
INFO:iMPACT - 0011 1111 1111 1110 0000 0000 0000 0000
INFO:iMPACT:579 - '2': Completed downloading bit file to device.
INFO:iMPACT:580 - '2': Checking done pin ....done.
'2': Programmed successfully.
Elapsed time = 6 sec.
// *** BATCH CMD : quit

DONE!
Output is seen on HYPERETERMINAL as shown in the following Figure and hence BRAM is tested.

![HyperTerminal Output]

**Figure B.2 Output on HYPERETERMINAL**