Hardware Design

MicroBlaze 7.3

Objectives

After completing this module, you will be able to:

• List the MicroBlaze 7.3 Features
• List the functionality that defines an arbiter, a master, and a slave
• List the various buses available in the MicroBlaze™ processor
• List the various high performance links in the MicroBlaze processor
Outline

- **MicroBlaze Features**
  - Buses 101: Arbiter, Master, Slave
  - MicroBlaze System Interfaces
    - Processor Local Bus (PLB)
    - Local Memory Bus (LMB)
    - Fast Simplex Link (FSL)
    - Xilinx Cache Link (XCL)

MicroBlaze Block Diagram

Optional MMU for Linux2.6 and MPU block for ease of software use

Enhanced FSL (fast simplex link) for CPU to hw/sw accelerator connections

Note: The OPB bus interface on the core is no longer available in 7.30 version
MicroBlaze Processor

- Scalable 32-bit Core
  - Single-Issue pipeline
    - Supports either 3-stage (resource focused) or 5-stage pipeline (performance focused)
  - Configurable Instruction and Data Caches
    - Direct mapped (1-way associative)
  - Optional Memory Mgt or Memory Protection Unit
    - Required for Linux OS (Linux 2.6 is currently supported)
  - Floating-point unit (FPU)
    - Based upon IEEE 754 format
  - Barrel Shifter
  - Hardware multiplier
    - 32x32 multiplication to generate a 64-bit result
  - Hardware Divider
  - Fast Simplex Link FIFO Channels for Easy, Direct Access to Fabric and Hardware Acceleration
  - Hardware Debug and Trace Module

Multi-processor Capability

- EDK9.1/MBv6 introduced Processor ID
- EDK9.2 introduced two fundamental MP cores
  - Mailbox: allows for message passing between 2 CPUs
  - Mutex core: allows 2 or more CPUs to synchronize
- No need for customers to build their own cores
- Processor Version Register (PVR) support
  - Contains: Processor ID, configuration/user/processor info (e.g. cache size etc), version number and other internal information
- PVR options:
  - None. No PVR implemented at all
  - Basic. Only first PVR register implemented
    - Allow designers to assign unique ID for a particular MicroBlaze core
    - Important for multi-core application or debug
  - Full. All PVR registers implemented
    - Allow designers access to all PVR registers for individual MicroBlaze configuration information per core
MailBox Core

- Two processors can communicate with each other either using PLB or FSL interface
- Configurable parameters
  - Depth
    - 16 to 8192 words
  - Interrupt thresholds and maskable interrupts
    - Separate thresholds for receive and send
    - Maskable interrupts for error, receive, and send
  - Synchronous or asynchronous operations

Mutex Core

- The core provides mechanism for mutual exclusion to enable one process to gain exclusive access to a particular resource
- Contains a configurable number of mutexes (up to 32)
  - Each of these can be associated with a 32-bit User configuration register to store arbitrary data
- Configurable parameters
  - Priority based PLB interfaces from 0 to 7 (0 has the highest priority)
  - Asynchronous or synchronous interface operation
  - USER register
  - Number of mutexes
  - CPUID width
  - Enhanced security through hardware identification support
Mutex Core

- Mutex lock and release process
  - Write <CPUID & 1> to the MUTEX register. If the mutex is free the lock bit will be set to one and the CPUID field will be updated with the new CPUID. Should the mutex already be locked the access is ignored
  - Read back the MUTEX register to verify that the mutex has been locked by the current CPU by comparing the value with the written CPUID, if not retry previous step until ownership has been granted
  - Manipulate the shared resource that is protected by the mutex
  - Release mutex by writing <CPUID & 0> to the mutex register. The mutex will automatically set the MUTEX register to zero

Outline

- MicroBlaze features
- **Buses 101: Arbiter, Master, Slave**
- MicroBlaze System Interfaces
  - Processor Local Bus (PLB)
  - Local Memory Bus (LMB)
  - Fast Simplex Link (FSL)
  - Xilinx Cache Link (XCL)
Buses 101

- A bus is a multiwire path on which related information is delivered
  - Address, data, and control buses
- Processor and peripherals communicate through buses
- Peripherals may be classified as:
  - Arbiter, master, slave, or master/slave (bridge)

![Bus Diagram]

- Bus masters have the ability to initiate a bus transaction
- Bus slaves can only respond to a request
- Bus arbitration is a three-step process:
  - A device requesting to become a bus master asserts a bus request signal
  - The arbiter continuously monitors the request and outputs an individual
    grant signal to each master according to the master’s priority scheme and
    the state of the other master requests at that time
  - The requesting device samples its grant signal until the master is granted
    access. The master then initiates a data transfer between the master and a
    slave when the current bus master releases the bus
- Arbitration mechanisms
  - Fixed priority, round-robin, hybrid
CoreConnect Bus Architecture

- The IBM CoreConnect bus architecture standard provides three buses for interconnecting cores, library macros, and custom logic:
  - Processor Local Bus (PLB)
  - On-Chip Peripheral Bus (OPB)*
  - Device Control Register (DCR)** bus
- IBM offers a no-fee, royalty-free CoreConnect bus architecture license
  - Licenses receive the PLB arbiter, OPB arbiter, and PLB/OPB bridge designs along with bus-model toolkits and bus-functional compilers for the PLB, OPB, and DCR buses
  - Required only if you create your own CoreConnect bus architecture peripheral or you are using the Bus Functional Model (BFM)

*OPB bus is deprecated, hence won’t be discussed
**DCR bus is PowerPC specific, hence won’t be discussed

Busses 101
The MicroBlaze processor core is organized as a Harvard architecture
Outline

- MicroBlaze Introduction
- Buses 101: Arbiter, Master, Slave
- **MicroBlaze System Interfaces**
  - Processor Local Bus (PLB)
  - Local Memory Bus (LMB)
  - Fast Simplex Links (FSL)
  - Xilinx Cache Link (XCL)

---

**MicroBlaze System**

- MicroBlaze™ 32-Bit RISC Core
- Processor Local Bus (PLB)
- Fast Simplex Link
- Arbiters
- Configurable Sizes
- D-Cache BRAM
- I-Cache BRAM
- Off-Chip Memory
- Flash/SRAM
- Custom Functions
- On-Chip Peripheral
- UART
- GPIO
- 10/100 E-Net
- Memory Controller
- Bus Bridge
- Local Memory Bus (LMB)
- SDRAM
- Another segment of PLB necessary when slow devices to be operated at slower bus speed enabling higher-performance system
- Arbiter required only when a peripheral is master capable and wants to write to peripheral on the other segment
Outline

- MicroBlaze Introduction
- Buses 101: Arbiter, Master, Slave
- **MicroBlaze System Interfaces**
  - Processor Local Bus (PLB)
  - Local Memory Bus (LMB)
  - Fast Simplex Links (FSL)
  - Xilinx Cache Link (XCL)

MicroBlaze System

Processor Local Bus (v4.6 PLB)

- MicroBlaze™ 32-Bit RISC Core
- Configurable Sizes
- BRAM
- Local Memory Bus
- Fast Simplex Link
- Custom Functions
- CacheLink
- I-Cache BRAM
- D-Cache BRAM
- Arbiter
- PLB
- Bus Bridge
- On-Chip Peripheral
- SDRAM
- Off-Chip Memory
- FLASH/SRAM

Another segment of PLB necessary when slow devices to be operated at slower bus speed enabling higher-performance system

Arbiter required only when a peripheral is master capable and wants to write to peripheral on the other segment
PLB Bus

- Connection infrastructure for high-bandwidth master and slave devices
- Fully synchronous to one clock
- Centralized bus arbitration—PLB arbiter
- 32 or 64-bit address (upper 32-bit are connected to GND by default)
- 32, 64, or 128-bit data bus
- Selectable shared bus or point-to-point interconnect topology
  - Point-to-point optimization available for 1 master, 1 slave configuration
  - Point-to-point topology supports 0 cycle latency via arbitration removal
- Selectable address pipelining support (2-level only)
- Dynamic master request priority based arbitration
- Vectored resets and address/qualifier registers

PLB Interconnect / Architecture

- One to 16 PLB masters, each connect all of their signals to the PLB arbiter
- The PLB arbiter multiplexes signals from masters onto a shared bus to which all the inputs of the slaves are connected
- One to n PLB slaves OR together their outputs to drive a shared bus back to the PLB arbiter
- The PLB arbiter handles bus arbitration and the movement of data and control signals between masters and slaves
PLB Bridge

• The PLB-to-PLB is required when two PLB segments are connected
  – Different bus speed
  – Different bus width
• The bridge translates PLB transactions on one side into the PLB transactions of the other side
• The bridge functions as a slave on one PLB side and a master on the other PLB side
• For a typical system with two PLB segments, one bridge is necessary for transactions originating from processor
  – A second bridge is required if a peripheral on the other side is master capable and wants to address a peripheral on the processor side

Outline

• MicroBlaze Introduction
• Buses 101: Arbiter, Master, Slave
• MicroBlaze System interfaces
  – Processor Local Bus (PLB)
  – Local Memory Bus (LMB)
  – Fast Simplex Links (FSL)
  – Xilinx Cache Links (XCL)
Local Memory Bus (LMB)

- The Local Memory Bus (LMB) provides single-cycle access to on-chip dual-port block RAM for MicroBlaze™ processors.
- The LMB provides simple synchronous protocol for efficient block RAM transfers.
- DLMB: Data interface, local memory bus (block RAM only).
- ILMB: Instruction interface, local memory bus (block RAM only).
Outline

- MicroBlaze Introduction
- Buses 101: Arbiter, Master, Slave
- MicroBlaze System Interfaces
  - Processor Local Bus (PLB)
  - Local Memory Bus (LMB)
  - Fast Simplex Link (FSL)
  - Xilinx Cache Link (XCL)

MicroBlaze System

Fast Simplex Links

- Another segment of PLB necessary when slow devices to be operated at slower bus speed enabling higher-performance system
- Arbiter required only when a peripheral is master capable and wants to write to peripheral on the other segment

- Fast Simplex Link
- Local Memory Bus
- Processor Local Bus
- Bus Bridge
- Processor Local Bus
- Arbiter
- SDRAM
- Off-Chip Memory
- FLASH/SRAM
The Software Streaming Data Challenge

- Suppose you want to move data through a hardware/software processing application with following characteristics
  - Data may be of a streaming or burst nature
  - Deterministic latency between hardware and software
- Possible solutions include
  - Bus peripheral, maybe PLB
    - Multiple clock-cycle overhead
    - Address decode time
    - Arbitration, loss of hardware/software coherency
  - Custom microprocessor instruction access to peripheral hardware
    - May require processor to be stalled
    - Complex logic can slow overall processor speed
    - May require assembly language to access special instruction
  - Fast Simplex Links

Another Alternative: Fast Simplex Links (FSL)

- Unidirectional point-to-point FIFO-based communication
- Dedicated (unshared) and nonarbitrated architecture
- Dedicated MicroBlaze™ C and ASM instructions for easy access
- High speed, access in as little as two clocks on processor side, 600 MHz at hardware interface
- Available in Xilinx Platform Studio (XPS) as a bus interface library core from Hardware → Create or Import Peripheral Wizard
FSL Features

- 32-bit wide interface
- Configurable FIFO depths – 1 to 8192 using SRL16 or block RAM
- Synchronous or asynchronous FIFO clocking with respect to the MicroBlaze™ system clock
- Selectable use of control bit
- Simple software interface using predefined C instructions; Automatically generated C drivers
- Blocking and nonblocking software instructions for data and control (**get** and **put**)
  - Blocking FSL instruction made interruptable
    - Return from interrupt will resume the FSL instruction
- Exception from FSL can be generated
- Disable interrupt while FSL executing
- Addition of dynamic assignment of FSL channel (**getd** and **putd**)
  - Channel number from register rather than immediate

Outline

- MicroBlaze Introduction
- Buses 101: Arbiter, Master, Slave
  - MicroBlaze System Interfaces
    - Processor Local Bus (PLB)
    - Local Memory Bus (LMB)
    - Fast Simplex Link (FSL)
    - Xilinx Cache Link (XCL)
Xilinx Cache Link

Xilinx Cache Link is a high-performance solution for memory accesses.

- The MicroBlaze CacheLink interface is designed to connect directly to a memory controller with integrated FSL Buffers.
  - i.e. MicroBlaze can connect directly to data ports of EDK supported multi-port memory controllers.
- The CacheLink Interface is only available on MicroBlaze when caches are enabled.
- The CacheLink cache controllers handle 4 or 8-word cache lines.
- All individual CacheLink accesses follow the FSL FIFO based transaction protocol.