Special Topics in Advanced Digital System Design

by
Dr. Lesley Shannon
Email: lshannon@ensc.sfu.ca
Course Website: http://www.ensc.sfu.ca/~lshannon/

Simon Fraser University

Slide Set: 1
Date: January 16, 2007

Slide Set Overview

- Things to keep in mind for your project
- The XUP Virtex II Pro Development board
- The Virtex II Pro
- The MicroBlaze Processor

Design Methodologies

- Things to keep in mind for your project
  - Design Considerations
  - Design Methodologies
  - Design Specification
  - Design for Testability
- However, the most important thing is …

HAVE A PLAN!

Design Considerations: "Techy" Terms

- Area
- Performance
  - Max Latency
  - Max Throughput
- Power/Energy Consumption
- Environment
- Scalability
- Security
- Reliability

Design Considerations: "Real World" Terms

- What does it need to do?
- Who is going to be using it?
- What's the environment it will be used in?
- What are the cost constraints?
- What type of scalability is required for future use?
- Are there security concerns?
- How reliable does the system have to be?
  - Can it fail without drastic consequences?

Design Methodologies

- Top-Down
- Bottom-Up
- Embedded Systems
  - Fully Software to Software + Hardware
  - Fully Hardware to Software + Hardware
  - Data Flow Graph & FSMs to Software + Hardware

Design Specification

- A Design Specification should be:
  - Complete
  - Unambiguous
  - Traceable
  - Modifiable
- There are:
  - Functional Requirements
    - "Use-Case" List
  - Non-Functional Requirements
    - Performance
    - Real Time requirements: Hard versus Soft Deadlines

Design Specification

- Unified Modelling Language (UML): Use-Case Diagram
  - Provide an overview of the usage requirements for a system
  - Provide an understandable representation to both technical and non-technical people
  - We're interested in:
    - Use Cases
    - Actors
    - Associations
    - System Boundary Boxes (optional)

Design Specification: Use Case Diagram

- Use Cases
  - Describes a sequence of actions that provide something of measurable value to an actor
  - Drawn as a horizontal ellipse
- Actors
  - A person, organization, or external system that plays a role in one or more interactions with your system
  - Drawn as stick figures
- Associations
  - Exist between actors and use cases
  - Occurs whenever an actor is involved with an interaction described by a use case
  - Indicated in by solid lines between actors and use cases

Design Specification: Use Case Diagram

- Associations (cont'd)
  - May have an optional arrowhead on one end of the line
    - The arrowhead is used to indicate
      - the direction of the initial invocation of the relationship or
      - the primary actor within the use case.
    - Arrowheads do ***not*** indicate data flow (can be confusing)
- System Boundary Boxes (optional)
  - Indicates the scope of your system
  - Anything within the box represents functionality that is in scope and anything outside the box is not.
  - Drawn as a rectangle around the use cases
  - Rarely used, but can be used to identify which use cases will be delivered in each major release of a system

Design Specification: Use Case Diagram

- Other Use Case Diagram Concepts include:
  - Packages (optional)
  - Extend
  - Include
  - Inheritance
- Won't be covered here
  - For more information, goto:
    - http://www.agilemodeling.com/artifacts/useCaseDiagram.htm
    - The Object Primer (3rd edition) by Scott W. Ambler, Chapter 5

Design Specification

- Don't forget:
  - Accuracy (fixed point/floating point);
  - Design Cost includes fabrication and design time
  - Power/Energy Usage generates heat (battery life)
  - Maximum available area (product dimensions)
  - Flexibility/Adaptability (upgrading functionality)
  - Reliability/Fault tolerance (serviceability and safety)
  - Security

Design for Testability

- BIST: Built-In Self Test
  - The technique of designing additional hardware and software features into integrated circuits to allow them to perform self-testing
    - Allowing them to test their own operation
      - functionally,
      - parametrically,
      - or both
  - By using their own circuits, it reduces the dependence on an external automated test equipment (ATE).

BIST- A DFT technique

- BIST is a Design-for-Testability (DFT) technique
  - It makes the electrical testing of a chip
    - easier,
    - faster,
    - more efficient, and
    - less costly.
- The concept of BIST is applicable to just about any kind of circuit
- DRAM is an example application
  - They incorporate additional circuits for:
    - pattern generation,
    - timing,
    - mode selection, and
    - go-/no-go diagnostic tests.

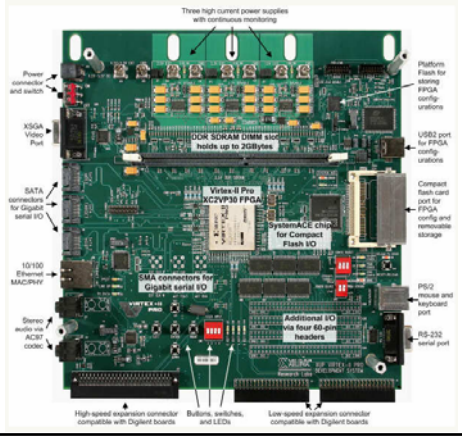
BIST- A DFT technique

- Also can be used to test the functionally of diverse blocks built using different technologies on one chip.
  - This would normally require high-end mixed-signal testers that possess special digital and analog testing capabilities
    - ***VERY EXPENSIVE***: 6-7 figures
  - BIST can be used to perform these special tests with additional on-chip test circuits eliminates the need to acquire such high-end testers.

Design For Testability

- In your case, you are using an FPGA
  - No post-fabrication design concerns
- You can use:
  - Simulation
  - Build ***visibility*** into your circuits
  - You can create on-chip Testbeds

XUP Virtex II Pro Development System



Virtex II Pro 30

Virtex II Pro 30

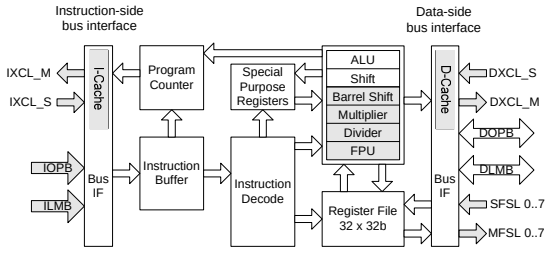
- Logic Cells: 30,816
  - Logic Cell = 1LUT + 1Flip flop
- BRAM: 2,448 **Kbits** (~306 KB)
- 18x18 Multipliers: 136
- Digital Clock Managers: 8
- PowerPCs: 2
  - Max Clock Frequency: 300MHz
- Available SERDES Transceivers: 8
  - Max Frequency: 3.123 Gbps
- Max Available User I/O: 644

Xilinx's MicroBlaze

Xilinx's MicroBlaze

- Harvard Architecture
  - Separate Instruction and Datapath
  - Used in:
    - DSP processor architectures
      - E.g. Blackfin from Analog Devices
    - Microcontrollers
      - E.g. PIC from Microchip Technology
- Resource Usage (Assuming lightweight)
  - ~1000 LUTs
  - ~800 Flipflops
- Max Frequency: ~250 MHz
- Xilinx also has PicoBlaze (look it up)

MicroBlaze Processor



Let's spec our own ipod

Creating an ipod Design Specification

- Let's make:
  - A Use-Case Diagram
  - A System-Block Diagram
  - An Algorithm Flow Chart