FPGA BASED SYSTEM DESIGN

Dr. Tayab Din Memon
tayabuddin.memon@faculty.muet.edu.pk
Lecture 1 & 2
Books

- Recommended Books:
  - Text Book: FPGA Based System Design by Wayne Wolf
  - Verilog HDL by Samir Palnitkar.
  - Advanced Digital Design with the Verilog HDL by Michael D. Ciletti
- Will tell you about other books in coming weeks.
Course Organization

• **Course material:**
  - All the course material will be provided in power point slides format

• **Course Instructor:**
  Dr. Tayab Din Memon
  Office: TQCIC Lab – II, IIT Building
  tayabuddin.memon@faculty.muet.edu.pk
Course Objectives & Scope

After taking this course, you will:

• Learn how to design digital circuits with HDL
• Have an understanding
  • VLSI: Fabrication, circuits, interconnects
  • FPGA based design techniques
  • FPGA fabrics
  • FPGA optimization for size, speed, and power consumption
  • VHDL
  • The structure of large digital circuits
  • Large scale platform and multi-FPGA systems
• Understand some of the important ideas for designing more complex systems.
Course Outline

• In this course most of the stuff is recommended from a book “FPGA Based System Design” that is given below:
  • FPGA Based Systems
  • VLSI Technology
  • FPGA Fabrics
  • Combinational Logic
  • Sequential Machines

• Some of the material will be covered from other books as:
  • Introduction to HDL and different modes of writing VHDL programming.
REVIEW OF VLSI AND EMBEDDED SYSTEMS

Lecture 1, 2 & 3
Introduction

- Integrated circuits: many transistors on one chip.
- **Very Large Scale Integration (VLSI)**: very many
- **Complementary Metal Oxide Semiconductor**
  - Fast, cheap, low power transistors
- **How to build your own simple CMOS chip**
  - CMOS transistors
  - Building logic gates from transistors
  - Transistor layout and fabrication
VLSI: Very Large Scale Integration

- Integration: Integrated Circuits
  - multiple devices on one substrate
- How large is Very Large?
  - SSI (small scale integration)
    - 7400 series, 10-100 transistors
  - MSI (medium scale)
    - 74000 series 100-1000
  - LSI 1,000-10,000 transistors
  - VLSI > 10,000 transistors
A Brief History

• 1958: First integrated circuit
  • Flip-flop using two transistors
  • Built by Jack Kilby at Texas Instruments

• 2003
  • Intel Pentium 4 processor (55 million transistors)
  • 512 Mbit DRAM (> 0.5 billion transistors)

• 53% compound annual growth rate over 45 years
  • No other technology has grown so fast so long

• Driven by miniaturization of transistors
  • Smaller is cheaper, faster, lower in power!
  • Revolutionary effects on society
Intel 4004 Micro-Processor
Evolution in Transistor Count
Moore’s Law

- 1965: Gordon Moore plotted transistor on each chip
  - Fit straight line on semilog scale
  - Transistor counts have doubled every 26 months

Integration Levels

SSI: 10 gates
MSI: 1000 gates
LSI: 10,000 gates
VLSI: > 10k gates
VLSI Design

• But the real issue is that VLSI is about designing systems on chips.
• The designs are complex, and we need to use structured design techniques and sophisticated design tools to manage the complexity of the design.
• We also accept the fact that any technology we learn the details of will be out of date soon.
The Process of VLSI Design:

Consists of many different representations/Abstractions of the system (chip) that is being designed.

- System Level Design
- Architecture / Algorithm Level Design
- Digital System Level Design
- Logical Level Design
- Electrical Level Design
- Layout Level Design
- Semiconductor Level Design (possibly more)

Each abstraction/view is itself a Design Hierarchy of refinements which decompose the design.
Design Abstraction Levels
Help from Computer Aided Design tools

- **Tools**
  - Editors
  - Simulators
  - Libraries
  - Module Synthesis
  - Place/Route
  - Chip Assemblers
  - Silicon Compilers

- **Experts**
  - Logic design
  - Electronic/circuit design
  - Device physics
  - Artwork
  - Applications - system design
  - Architectures
New Design Methodologies

• Methodologies which are based on:
  • System Level Abstractions v.s. Device Characteristic Abstractions
    • Logic structures and circuitry change slowly over time
      • trade-offs do change, but the choices do not
  • Scalable Designs
    • Layout techniques also change slowly.
      • But the minimum feature size steadily decreases with time (also Voltage, Die Size, etc.)
Design Approaches

- **Custom**
  - full control of design
  - best results, slowest design time.
- **Semi-custom (std cell)**
  - use Cell libraries from vendor
  - cad tools, faster design time
- **Gate Array**
  - fastest design time
  - worst speed/power/density
  - best low volume (worst high volume)
- **EPLA/EPLD - FPGA - electrically programmable (in the field)**
Evolution in Speed/Performance
Embedded Systems Overview

Embedded Systems :-
Application-specific systems which contain hardware and software tailored for a particular task and are generally part of a larger system (e.g., industrial controllers)

- Characteristics
  - Are dedicated to a particular application
  - Include processors dedicated to specific functions
  - Represent a subset of reactive (responsive to external inputs) systems
  - Contain real-time constraints
  - Include requirements that span:
    - Performance
    - Reliability
    - Form factor
Embedded systems overview

- **Embedded computing systems**
  - Computing systems embedded within electronic devices
  - Hard to define. Nearly any computing system other than a desktop computer
  - Billions of units produced yearly, versus millions of desktop units
  - Perhaps 50 per household and per automobile

Computers are in here...
and here...
and even here...

Lots more of these, though they cost a lot less each.
Examples: Refrigerator

Compressor control
Alarm
Display

The embedded computer
Actual temperature
Required temperature

Human interaction
Networked interaction (maybe!)

Dr. Gheith Abandah
Examples: Car Door

Window stall sensor

Window control buttons

Window motor

Lock control

Lock actuator

Open door sensor
A “short list” of embedded systems

- Anti-lock brakes
- Auto-focus cameras
- Automatic teller machines
- Automatic toll systems
- Automatic transmission
- Avionic systems
- Battery chargers
- Camcorders
- Cell phones
- Cell-phone base stations
- Cordless phones
- Cruise control
- Curbside check-in systems
- Digital cameras
- Disk drives
- Electronic card readers
- Electronic instruments
- Electronic toys/games
- Factory control
- Fax machines
- Fingerprint identifiers
- Home security systems
- Life-support systems
- Medical testing systems
- Modems
- MPEG decoders
- Network cards
- Network switches/routers
- On-board navigation
- Pagers
- Photocopiers
- Point-of-sale systems
- Portable video games
- Printers
- Satellite phones
- Scanners
- Smart ovens/dishwashers
- Speech recognizers
- Stereo systems
- Teleconferencing systems
- Televisions
- Temperature controllers
- Theft tracking systems
- TV set-top boxes
- VCR’s, DVD players
- Video game consoles
- Video phones
- Washers and dryers

And the list goes on and on
Concepts of co-design

• **Codesign**
  • The meeting of system-level objectives by exploiting the trade-offs between hardware and software in a system through their concurrent design

• **Key concepts**
  • **Concurrent**: hardware and software developed at the same time on parallel paths
  • **Integrated**: interaction between hardware and software developments to produce designs that meet performance criteria and functional specifications
Essential components and considerations

• Essential components :-
  - Microprocessor / DSP core
  - Sensors
  - Converter ( A-D and D-A )
  - Actuators
  - Memory (on-chip and off-chip )
  - Communication path with interfacing environment

• Essential considerations :-
  - Response time :- ( Real time system )
  - Area, Cost, Power, Portability, Fault-tolerance
Design-flow in ES Design

Major Tasks in ES Design

- Modelling
  - the system to be designed, and experimenting with algorithms involved;

- Refining (or “partitioning”)
  - the function to be implemented into smaller, interacting pieces;

- HW-SW partitioning: Allocating
  - elements in the refined model to either HW units, or SW running on custom hardware or a uP.

- Scheduling
  - the times at which the functions are executed. This is important when several modules in the partition share a single hardware unit.

- Mapping (Implementing)
  - a functional description into (1) software that runs on a processor or (2) a collection of custom, semi-custom, or commodity HW.
THANKS FOR YOUR PATIENCE

End of Lecture 1 & 2