ECE 351
Digital Systems Design

Introduction

Wei Gao

THE UNIVERSITY of TENNESSEE KNOXVILLE

Spring 2017

Course Information

- Class time: 9:40am – 10:55am TTR, Min Kao 405
- Instructor: Wei Gao, weigao@utk.edu
  - Office: Min Kao 353
  - Office hour: 11:00am – 12:00pm TTR
- TA: Sagarvarma Sayyaparaju, ssayyapa@vols.utk.edu
  - TA Office Hour: TBD
- Slides, schedule, announcement posted at http://web.eecs.utk.edu/~weigao/ece351/spring2017/schedule.html

Examples of Digital Systems

- Cellphone, Personal Digital Assistant (PDA)
- Printer.
- GPS.
- Automobile: engine, brakes, dash, etc.
- Digital camera.
- iPod.
- Household appliances: microwave, air conditioning
- Wrist watch.
- and a lot more …

- Fact: > 95% of daily appliances are having digital system components

Design Choices of Digital Systems

- Application-Specific Integrated Circuits (ASICs)
- Microprocessors
- Field-Programmable Gate Arrays (FPGAs)

ASIC

Example: Digital baseband processing for cell phones

- ✓ Performance: Fast!
- ✓ Power: Fewer logic elements → low power
- ✗ Development cost: Very high
  - 2 million $ for starting production of a new ASIC
  - Needs a long time and a large team
- ✗ Reprogrammability: None!
  - → Single-purpose devices
  - → Difficult to upgrade systems

Microprocessors

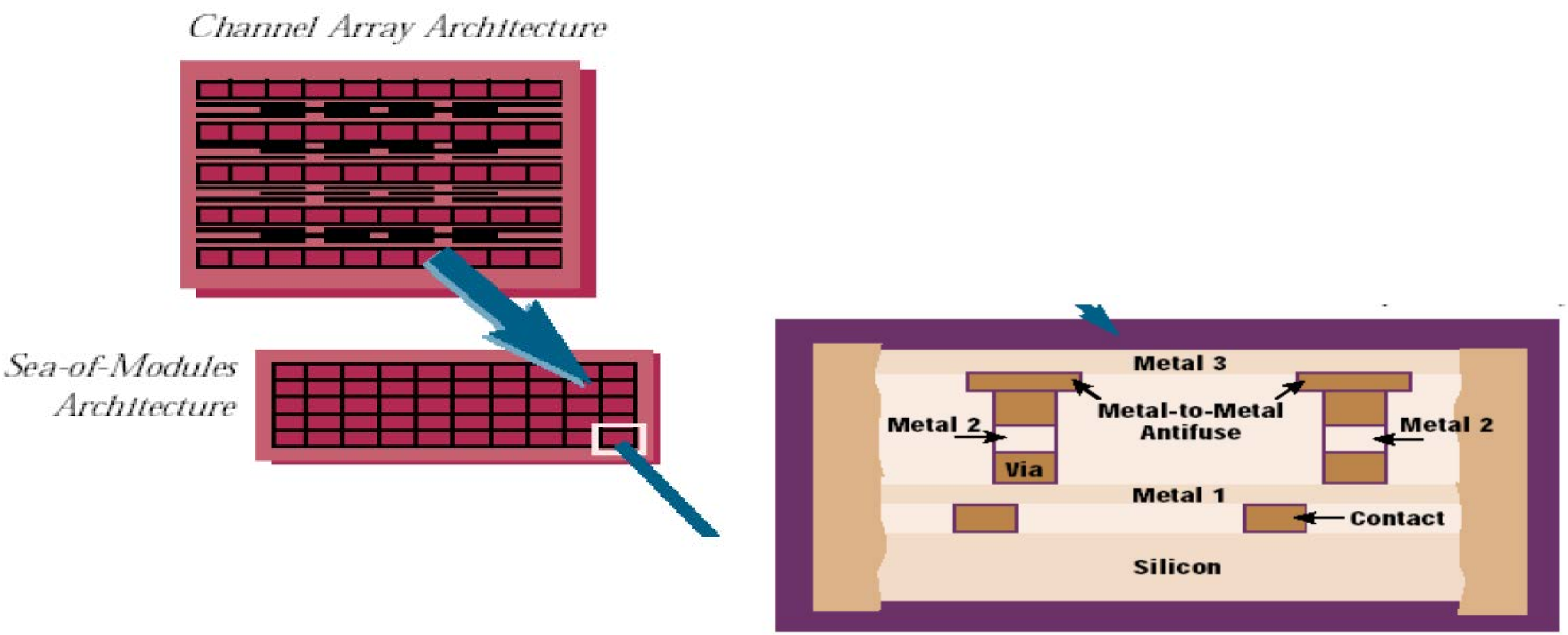
- Von Neumann (or Harvard) architecture is fundamentally slow!
  - Fetch, decode instructions
- Improve performance at the cost of power!
  - Performance/watt remains low
- Let software do the work
  - Flexibility and low development cost
  - Microprocessor + ASIC is common

Field-Programmable Gate Arrays (FPGA)

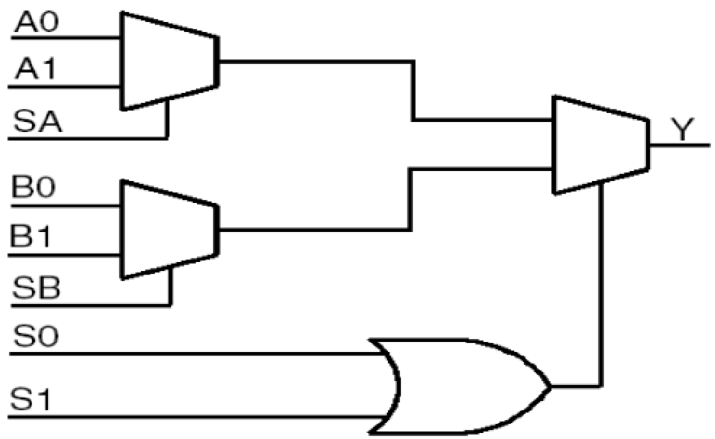
- Programmable hardware
- Combine the benefits of ASIC and microprocessor
  - Hardware implementation → good performance/watt
  - Reprogammable → lower development cost
- Implement something in hardware, but change it as easily as changing software running on a microprocessor
  - The ideal hardware analogy would be the ability to reconfigure the connection between millions of transistors to compute a new function for a new application

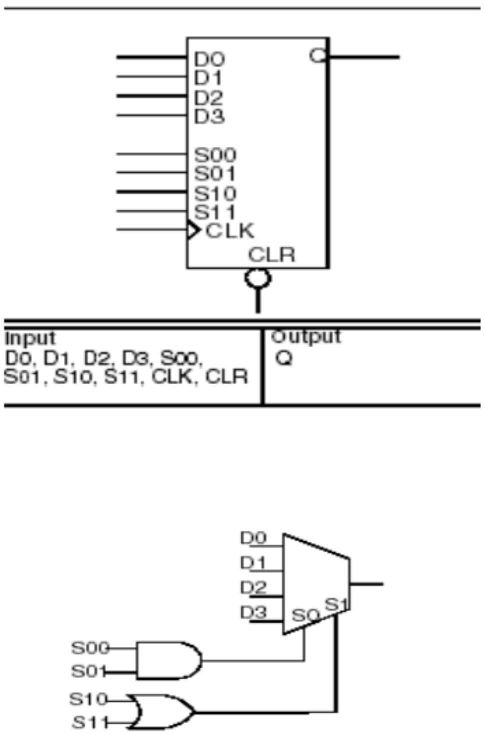
Traditional FPGA Architecture

- Modules are connected by a “programmable” link.
  - One-time-programmable only
  - Analogy: Early-stage Flash ROM storage devices



Actel Logic Elements



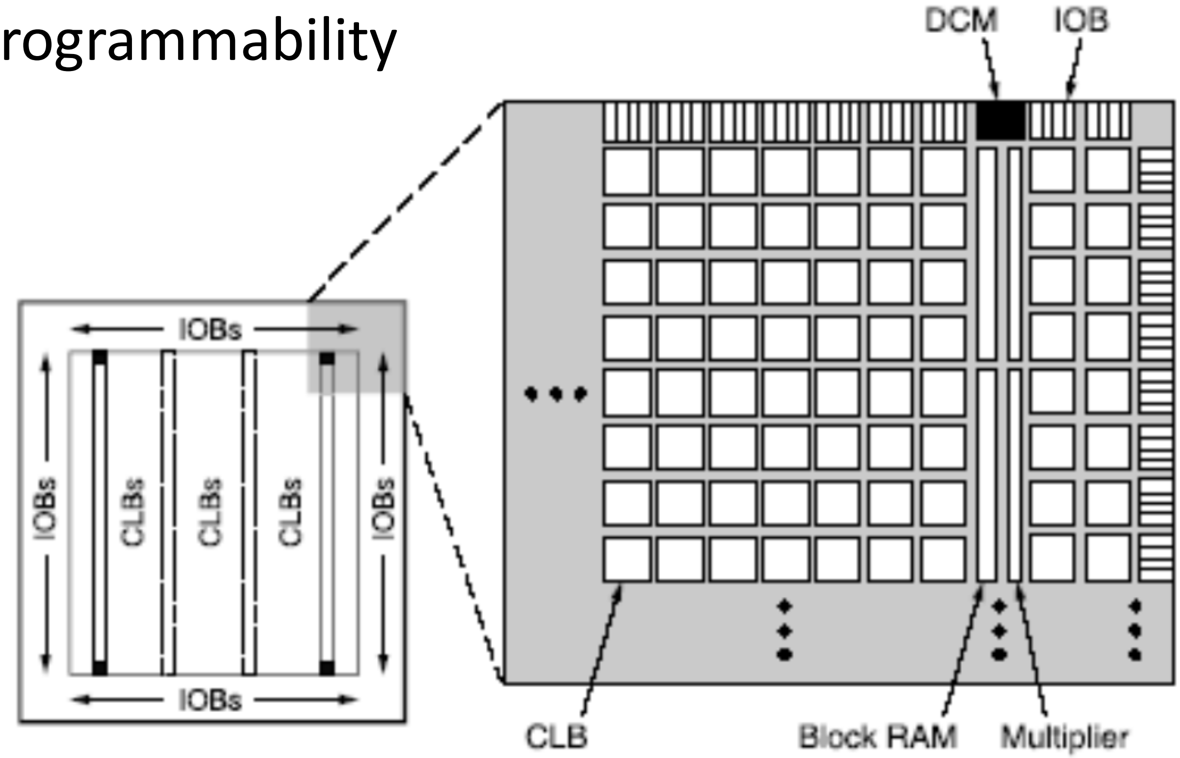


Combinatorial module

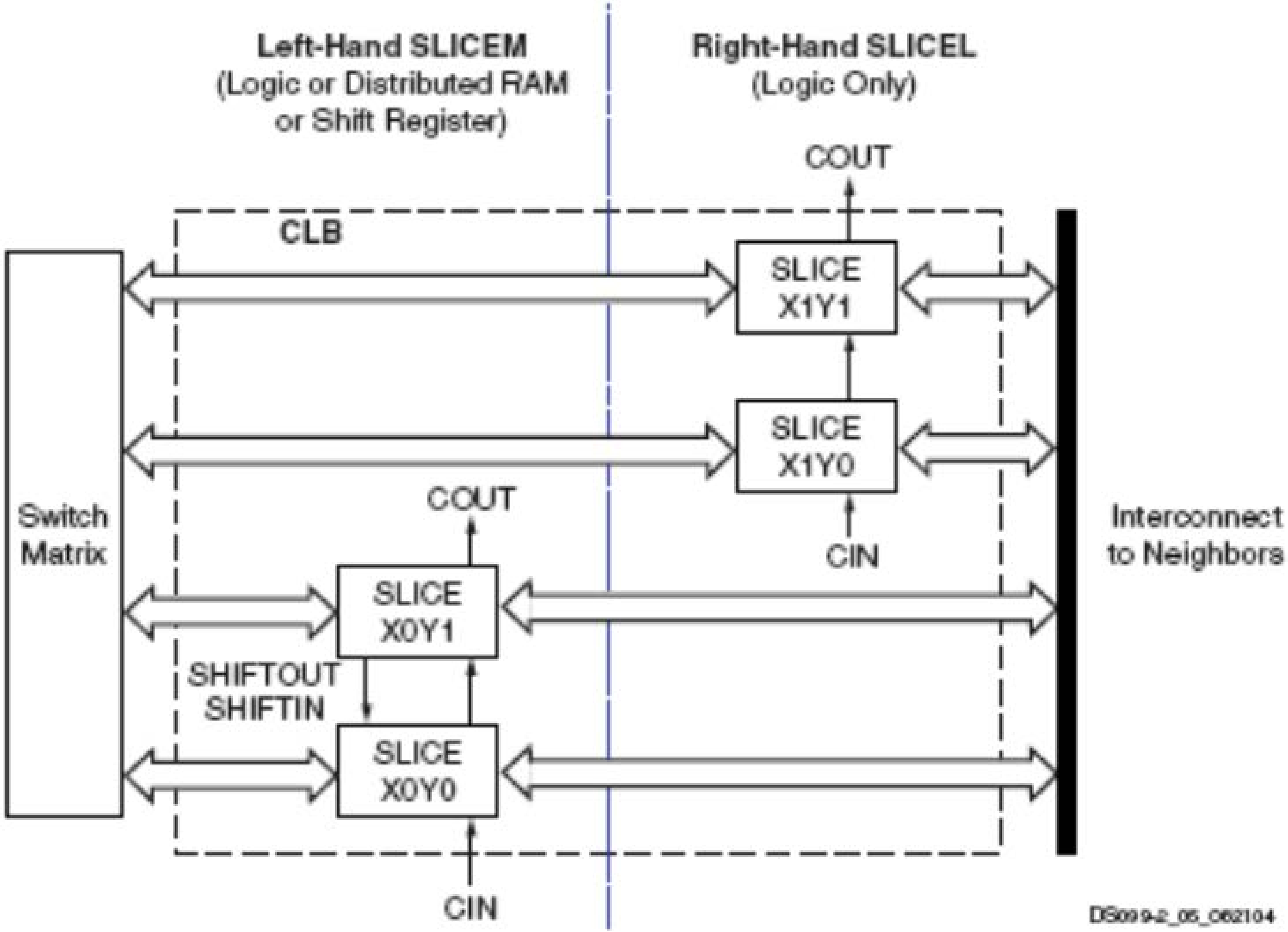
Sequential module

Xilinx Spartan FPGA Implemenation

- Configurable Logic Blocks are connected by programmable interconnect, configuration of which is controlled by SRAM
  - Unlimited reprogrammability



Configurable Logic Block



Programmable Logic Design Flow

- **Capture design:** Draw Schematic or use a Hardware Description Language (HDL)
- **Synthesis/Map:** Maps and optimizes design to fit into FPGA architecture primitives
- **Place / Route:** figures out where on the FPGA/CPLD each primitive block goes, and how to connect them together.
- **Programming:** Using the output from the FPGA design software, the connections on the FPGA are physically created on the actual chip.

VHSIC Hardware Description Language (VHDL)

- Programming models operating FPGA
- VHSIC: Very-High-Speed Integrated Circuits
  - DoD project in 1980s – military use
  - Control given to IEEE in 1986 for standardization
- Functional description of a digital circuit
  - VHDL ’87: first VHDL standard
  - VHDL ’93: first revision among lots of software

VHDL Modeling

- Circuit complexity is so high that designs can't be completely analyzed
- Digital circuits are highly parallel and not easily modelled by traditional software.
- Simulation of our system is how we verify.
- VHDL provides a convenient, universally supported means to model circuit behavior

Examples of VHDL Use

- **Early Project Verification:** VHDL system model can be used to verify that a proposed system design will work
- **Subsystem design requirements specification:** Detailed requirements for each piece of the system can be formed with the aid of the system model
- **Synthesis:** models of some pieces can be refined to be synthesizable and thus easily built.
- **Verification:** post-synthesis VHDL models with timing can be inserted into the system model and verified.

Generalization: Cyber-Physical System

- A physical system that tightly interacts with a digital system.
  - Digital circuits replace mechanical controllers
  - Use the sensed data for feedback control and optimization
- Automobile systems:
  - Engine controllers replace distributor, carburetor, etc
  - Complex algorithms allow both greater fuel efficiency and lower emissions
- More examples
  - Urban sensing, smart healthcare, etc

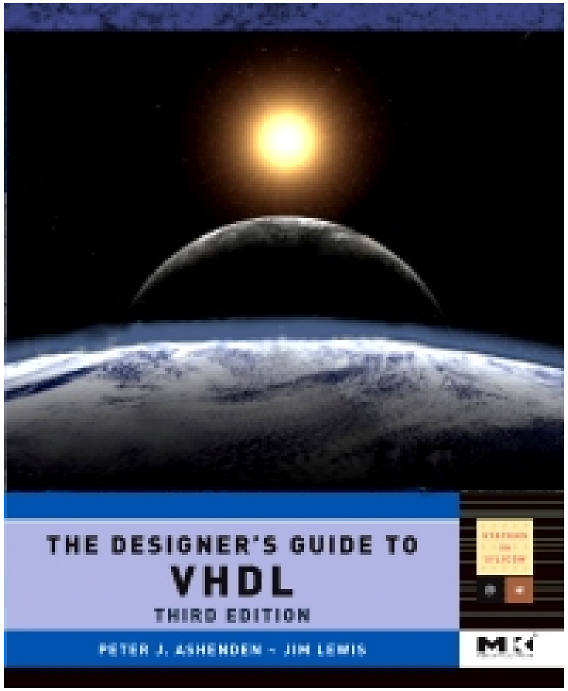
Why are those digital systems special?

- Application specific
  - Specialize and optimize the design for specific application
  - *Not* a general-purpose computer.
    - Don't need all the bells and whistles, e.g., hard drive, monitor, keyboard…
- Have to worry about both hardware and software
- Have to worry about non-functional constraints
  - Real-time
  - Memory footprint
  - Power
  - Reliability and safety
  - Cost

Just functionally working is NOT enough!

Goal of This Course

- The common principles and hands-on skills of digital system design
  - Instead of a specific type of FPGA device
  - Emphasize aspects that are distinct to reprogrammable digital systems
- Textbook:
  - *The Designer's Guide to VHDL*, 3rd edition
  - Peter J. Ashenden
  - Morgan Kaufmann Publishers, 2008
  - Not required but recommended
- Experimentation platform
  - Diligent Basys 2 FPGA board
  - Embedded system components



Goal of This Course

- References
  - The Student's Guide to VHDL
    - Peter J. Ashenden
    - Morgan Kaufmann, 2^{nd} edition, 2008.
  - Digital Design: An Embedded Systems Approach Using VHDL
    - Peter J. Ashenden
    - Morgan Kaufmann, 2008.
  - FPGA-Based System Design
    - Wayne Wolf
    - Prentice Hall, 2004.

What will you learn from this course?

- Hardware
  - FPGA, reprogrammable circuits, etc.
- VHDL programming
  - Synthesis of digital hardware
  - Modeling of complex digital systems
  - Testbench development
- FPGA development
  - FPGA design concepts
  - Hybrid hardware/software systems-on-chip using the soft-core microprocessors inside the FPGA
- Digital system development
  - Using the FPGA to control peripheral components and realize system functionality

What will you NOT learn from this course?

- Computer logic
  - ECE 255: Introduction to Logic Design of Digital Systems
- Computer architecture
  - ECE 451: Computer System Architecture
- Embedded system design
  - ECE 455: Embedded System design
- System programming
  - COSC 360: Systems programming

Grading (Tentative)

- Labs (4) 25%
- Group project 35%
  - Proposal presentation 5%
  - Midterm presentation/demo 7%
  - Final presentation/demo 8%
  - Final report 15%
  - 2 students per group
- Midterm exam 15%
- Final exam 20%
- Participation 5%

Course policy

- Academic integrity
  - Must be your OWN work
  - No collaboration for homework/lab assignment
- Lab policy
  - Results must be checked by TA
  - Printed copy of source code needs to be submitted
- Exam policy
  - Closed-book, No discussion, No make-up exams
- Project policy
  - Clearly identify the contribution of each group member
- Class policy
  - No laptops in class
  - Attend each lecture

Discussion

- TA office hours vs. lab hours
- Choice 1: 2 1-hour office hours + 2 1-hour lab hours
- Choice 2: 3-4 1-hour lab hours but no office hours
- What specific help do you expect to need from the TA?

Next Class

- Introduction of course projects