EE 459/500 – HDL Based Digital Design with Programmable Logic

Lecture 1

Introduction

Course Info

- Times: TuTh 11am-12:20pm
- Room: Furnas 213/214
- Instructor – Cristinel Ababei
  - E-mail: cababei@buffalo.edu
  - Phone: 716-645-1607
  - Office hours (209 Davis Hall): TuTh 10-11am or by appointment
- Teaching assistant
  - Yi Cao, yicao@buffalo.edu
  - Labs: Wed. 2-4pm in Furnas 213/214

Course Requirements

- Textbook and references
  - Jr. Charles H. Roth and Lizy K. John, ***Digital Systems Design Using VHDL***, CL Engineering, 2nd edition, 2007.
  - Peter J. Ashenden, ***The Student's Guide to VHDL***, Morgan Kaufmann.
- Software
  - Xilinx ISE WebPack 14.1
  - Aldec's Active-HDL 9.1 Student Edition simulator
- Hardware
  - Digilent ATLYS FPGA (Spartan-6) development board

3

Grading

- Grade breakdown: "A"=[97-100] "A-"=[94-97) "B+"=[90-94) "B"=[87-90) "B-"=[84-87) "C+"=[80-84) "C"=[77-80) "C-"=[74-77) "D+"=[70-74) "D"=[60-70) "F"=[0-60).
- Final grade components:
  - Midterm exam: 20%
  - Final exam: 20%
  - Homework assignments: 10%
  - Labs: 20%
  - Course project: 30%

4

Course Outline

- Week 1: Introduction, overview of digital systems
- Week 2: Combinational logic circuits and design
- Week 3-5: VHDL modeling of digital systems
- Week 6-9: Sequential systems and design
- Week 10: Memory and timing
- Week 11-14: Computer design basics
- Week 15: Project demos

5

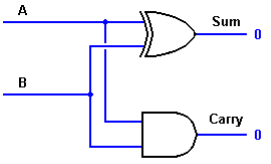
What is a Digital System?

- **Structure:** a collection of interconnected digital modules designed to perform a particular function.
- **Function:** takes a set of discrete information inputs and discrete internal information (system state) and generates a set of discrete information outputs.

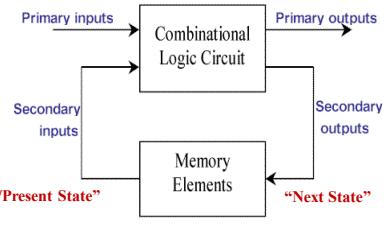
6

Types of Digital Systems

- No state present
  - Combinational Logic System
  - Output = Function (Input)



- State present
  - Sequential Logic System
    - State updated at discrete times
      - => Synchronous Sequential System
    - State updated at any time
      - =>Asynchronous Sequential System
  - Next State = Function (State, Input)
    - Output = Function (State, Input) – Mealy machine
    - Output = Function (State) – Moore machine



7

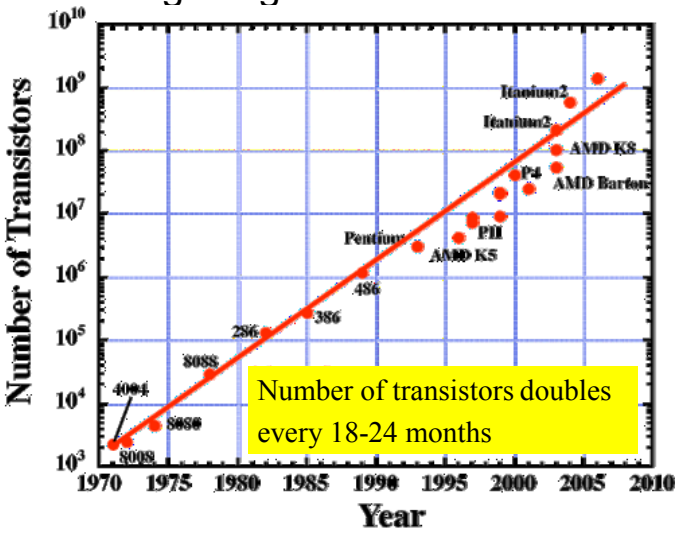
Digital Systems Modules

- Low level digital modules
  - Gates - AND, OR, NAND, NOR, XOR, etc.
  - Blocks - adder, subtractor, shifter, multiplier, etc.
- High level digital modules
  - ASICs (Application Specific Integrated Circuits)
  - Microprocessors/Microcontrollers
  - FPGAs/PLDs (Programmable Logic Device)
  - Memories

8

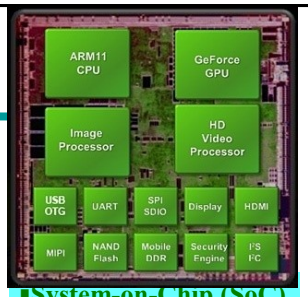
Digital Systems Trend

- Increasing integration: Moore's Law

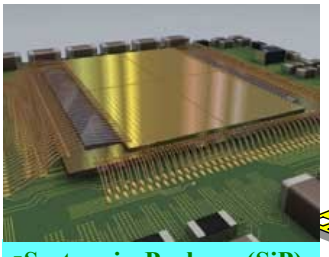


9

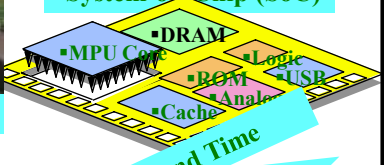
Digital Systems Trend



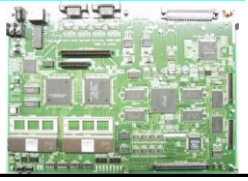
▪System-on-Chip (SoC)

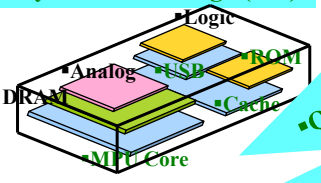


▪System-in-Package (SiP)



▪System-on-Board (SoB)





▪Quick Turn-around Time

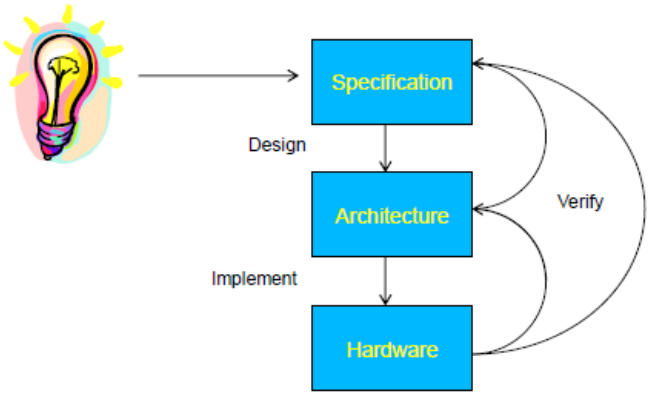
▪Smaller form factor

▪Lower power, cost

10

How is a Digital System Designed?

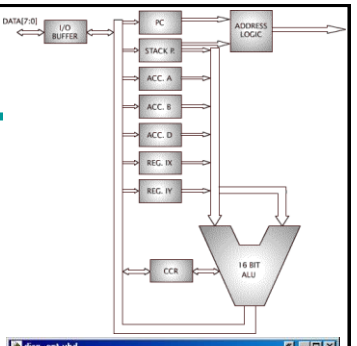
- **Digital Systems Design** - a process that entails a systematic development of an idea into an architecture that can be implemented digitally.

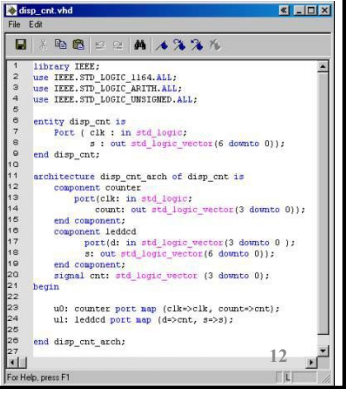


11

Specification

- Translation from idea into a formal description of behavior
- The highest level of abstraction/specification is a declarative statement or written expression that specifies the design idea
- Forms: text description; diagrams; specialized specification languages (VHDL, Verilog, etc.)





12

Hardware Description Languages

- Two primary choices
  - **VHDL** - VHSIC (very high-speed IC) Hardware Description Language
  - Verilog
- Can be used for behavioral specification, architectural definition, implementation, and verification
- Other specification languages specific areas include Harel's StateCharts, UML Statecharts, SystemC, SpecC, SystemVerilog, Simulink, C, C++, Java

13

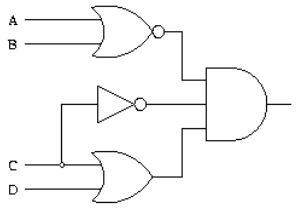
Architecture

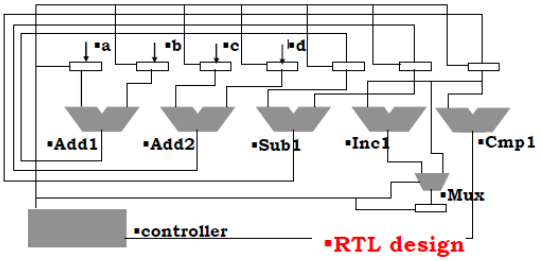
- High-level partitioning of system into functional blocks
- Can be expressed in a variety of forms - text, graphically, formal languages, etc.
- Difficult procedure without experience or tools to assist you
- Must choose between different possible architectures and weigh the costs and benefits of each choice
- Architecture design is often a matter of balancing tradeoffs

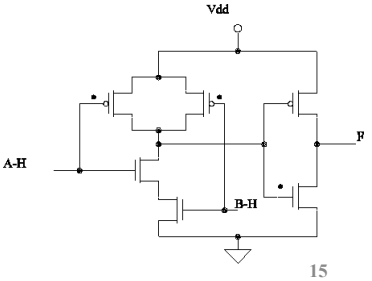
14

Implementation

- Transformation of architecture into hardware
  - Register Transfer Level (RTL)
  - Schematic Entry (logic)
  - Transistor Level



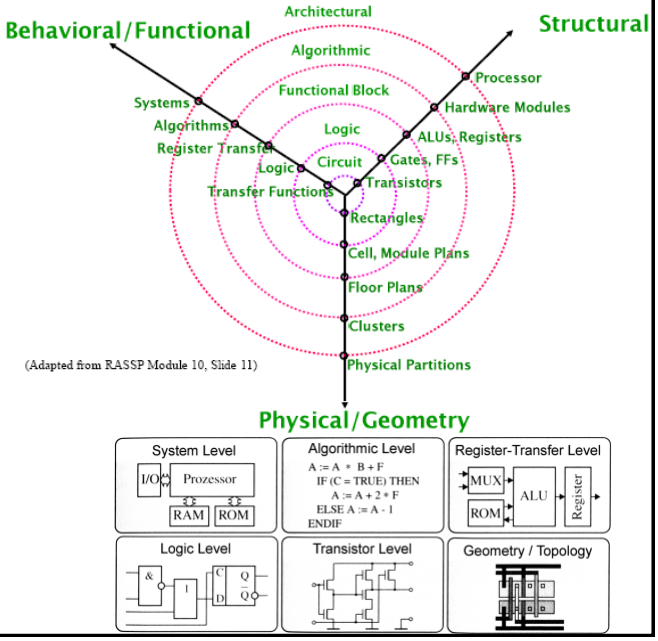




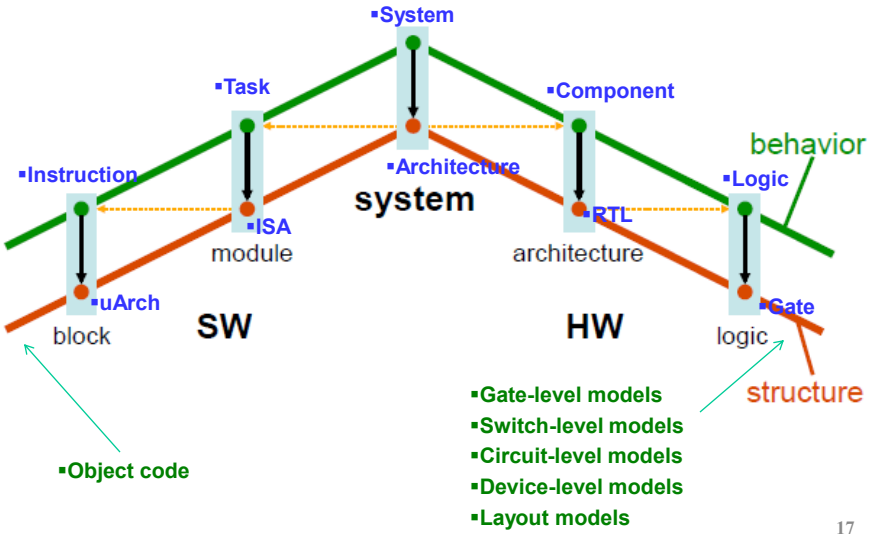
15

Digital Systems Modeling

- Gajski and Kuhn Y chart (1983)



Digital Systems Modeling – Abstraction Levels



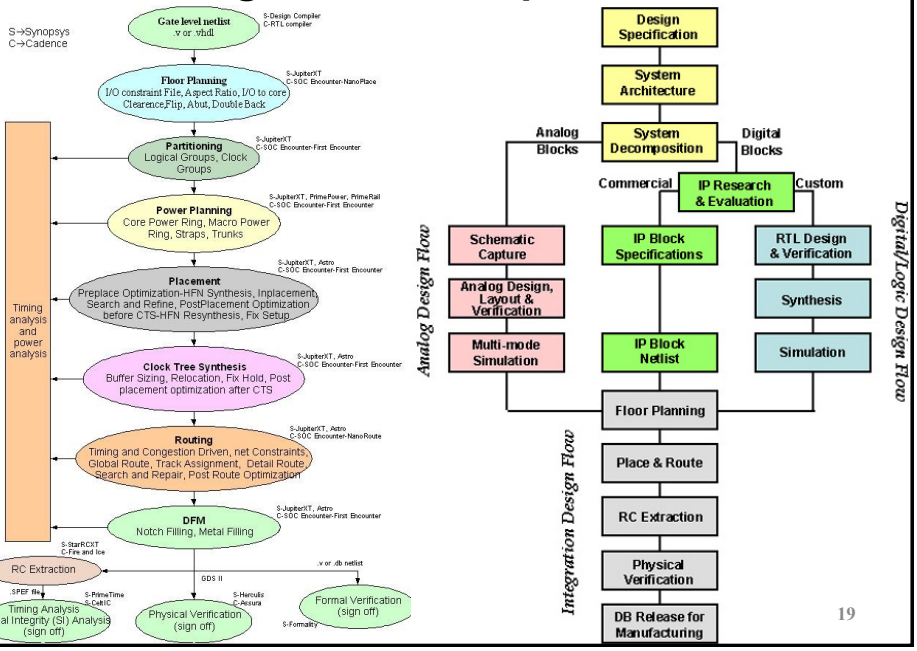
17

Synthesis Tools

- **Logic Synthesis**: generate a circuit netlist from design descriptions in VHDL or Verilog
  - Logic Synthesis Tools: Cadence, Synopsys (includes Magma now ☺), Mentor Graphics, etc.
  - The final phase is technology mapping
    - Optimization: optimize area, delay, etc.
    - Technology-specific: ASIC, FPGA, CPLD
- **Physical Synthesis/Design**: place lower level blocks (e.g., logic gates) and route interconnections
- **FPGA synthesis tools**: Xilinx ISE WebPack, Altera Quartus, Mentor Graphics, Synopsys

18

VLSI Design Flow Examples



Verification

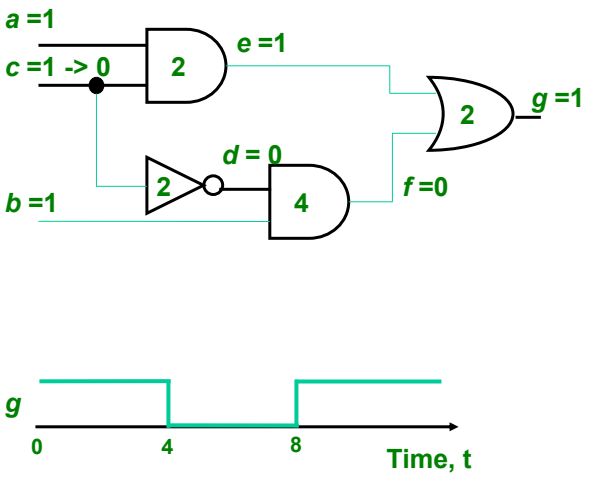
- Check whether implementation matches specification
  - Simulation: EDA companies offer simulators (Aldec, ModelSim, Synopsys, Cadence, etc.)
  - Formal equivalency checking
- Iterative process
  - Simulate
  - Refine specification or architecture if necessary
  - Repeat

How Logic Simulation Works

- VHDL simulator: **event-driven** simulator
  - When a circuit node changes in value, the time, the node and the new value are collectively known as an event
  - When a specified time is reached, the logic value of the node is changed
  - Changes are detected and executed in parallel using concurrent VHDL statements

21

Event-driven Algorithm



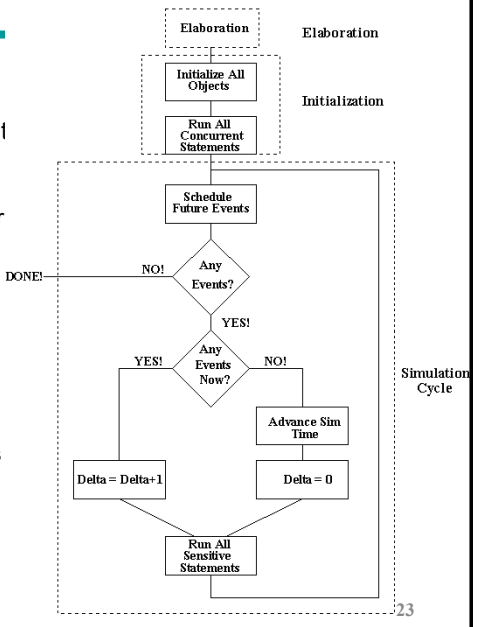
Time stack

	Scheduled events	Activity list
t = 0	c = 0	d, e
1		
2	d = 1, e = 0	f, g
3		
4	g = 0	
5		
6	f = 1	g
7		
8	g = 1	

22

VHDL Simulation

- VHDL simulation process can be broken into:
  - Elaboration: Design hierarchy is first elaborated. All the pieces of the model code (entities, architecture and configurations) are put together
  - Initialization: The nets (correspond to wires) in the model are initialized just before simulation starts
  - Simulation cycle:
    - Simulation cycle is continuously repeated during which processes are executed and signals are updated via "propagation" of events
- Advantage: top-down design methodology, technology independent



23

Summary

- Moore's Law → Increased circuit complexity and integration → SoCs
- However, focus of this course: block (digital module) level and FPGA implementation/prototyping
- Emphasis on VHDL; goal is to introduce you to VHDL; better learning comes with practice, practice, practice…
- Individual course project, the most important: 30% of final grade
- Completion of labs is mandatory
- Please read textbook before attending class
- Take ownership of class
- Emphasis on learning by examples

24