CS411
Digital System Design

Dr. Arshad Aziz

Introduction

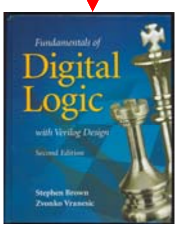
Course Descriptions

- A digital design course for undergraduate students in EE/EL
- Language-based design method: a new approach to digital circuits/systems design, compared to methods introduced in DLD.
- Focus on Verilog HDL modeling, simulation and synthesis of digital logic circuits
- Architecture of Modern FPGAs
- Extensive use of CAD tools and Verilog (Xilinx ISE 10 or 12)
- Also include testing of logic circuits
- Any experience of computer language will be helpful.

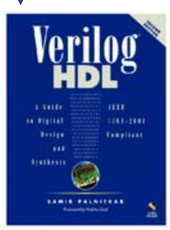
why this course ?

- ✓ Most of modern digital circuits are big
- ✓ It is hard to design big circuits using paper-and-pencil.
- ✓ We ask computer for help if a good automatic design software is available.
- ✓ Finally, with computer's help, we can design big digital systems quickly

Textbook and Reference book



Fundamentals of Digital Logic with Verilog Design
Stephen Brown and
Zvonko Vranesic,
McGraw-Hill, 2nd or 1st edition.



Verilog HDL,
a guide to digital design and synthesis
Samir Palnitkar,
Prentice Hall, 2003.
ISBN 0-13-044911-3

Learning outcomes

- Students will be able to design digital circuits/systems through a hardware descriptive language.
- Students will understand the implementation of digital systems by programmable devices, such as FPGA and CPLD.
- Students will able to use CAD tools to design and analyze digital systems.
- Students will be familiar with the synthesis of digital circuits.
- Students will be familiar with the testing issues of digital circuits.
- Student will be able to design any digital system by themselves.

Information Delivery

- Presentation in class is the main channel of information, including knowledge, schedule announcements.
- Lecture notes will be posted on course website.
- Tutorials for Xilinx ISE, Verilog and Ebooks will be posted on course website.

Introduction to Logic Design

Learning objectives

- Be aware where logic circuits are in our daily life.
- To know that how logic circuits are implemented in real world.
- To get familiar with the concept of design process in industry.

Logic circuits are everywhere

- Computers
- Digital watches
- CD players
- Electronic games
- Telephone and television networks
- Missile guidance systems
- Airplanes and space shuttles

Invention of the Transistor

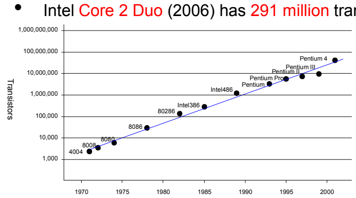
- Vacuum tubes ruled in first half of 20th century Large, expensive, power-hungry, unreliable
- 1947: first point contact transistor at Bell Labs
  - John Bardeen and Walter Brattain at Bell Labs
  - Read *Crystal Fire* by Riordan, Hoddeson

Transistor Types

- Bipolar transistors
  - npn or pnp silicon structure
  - Small current into very thin base layer controls large currents between emitter and collector
  - Base currents limit integration density
- Metal Oxide Semiconductor Field Effect Transistors
  - nMOS and pMOS MOSFETS
  - Voltage applied to insulated gate controls current between source and drain
  - Low power allows very high integration
  - Simpler fabrication process

Moore's Law

- **Moore's Law** , is the empirical observation made in 1965 that the number of transistors on an IC for minimum component cost doubles every 24 months
- Dr. Gordon E. Moors (born 1929), a co-founder of Intel. Moore has since retracted this law, claiming it to be
- Intel 4004 (1971) has 3500 transistors.
- Intel Core 2 Duo (2006) has 291 million transistors.



Integration Levels

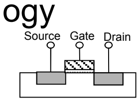
SSI: 10 gates

MSI: 1000 gates

LSI: 10,000 gates

VLSI: > 10k gates

Facts: Process Technology



- Gate Length: Intel's 65-nanometer (nm) process technology features transistors whose gate length is just 35nm (a gate is the switch that turns a transistor on and off). Approximately 1,400 of these gates could fit inside the pore diameter of human skin.*

- Gate Height: Intel's 65nm process technology has a gate height of 1.2nm. More than 90,000 of these gate oxide layers would need to be stacked to achieve the thickness of one U.S. $1 bill.**

- Line Width: Intel's 65nm process technology prints individual lines 54,000 times thinner than a silk thread.***

Facts: Transistor Density

- There are more transistors (291 M) in an Intel® Core™2 Duo processor than there are minutes in 552 years.
- An Intel® Core™2 Duo processor has over 291 M transistors on a die size of 143 mm^2. This averages to nearly 2 M transistors per square mm or 2 M transistors on the size of the tip of a ball point pen.
- If you had a penny for every transistor in an Intel® Core™2 Duo processor, you could get a stack of pennies 284 miles high. If you spread the pennies out, side by side, the span would be longer than the distance between New York and San Francisco. **
- An Intel® Core™2 Duo processor would be approximately the size of two compact parking stalls if its transistors were the same size as those on Intel's first microprocessor, the 4004. The 4004 microprocessor incorporated just 2,300 transistors, compared to the 291 million on the Intel® Core™2 Duo processor.***
- If you had a grain of rice for each transistor in the Intel® Core™2 Duo processor you could serve rice to the entire population of Samoa, which has population of 177K. ****

Facts: Transistor Speed

- The transistors on the Intel® Core™2 Duo processor, which act like switches controlling the flow of electrons inside a microchip, turn on and off more than a trillion times per second. This helps enable the Intel® Core™2 Duo processor to complete close to a billion calculations in the blink of an eye or finish four million calculations in the time it takes a speeding bullet to travel one inch.**
- It would take a person 25,000 years to turn a light switch on and off 1.5 trillion times, but Intel has developed transistors such as those found in the Intel® Core™2 Duo processor that can switch on and off that many times each second.
- Intel's 4004 microprocessor, introduced in 1971, ran at 108 kilohertz (108,000 hertz). Intel® Core™2 Duo processors exceed speeds >2 gigahertz (2,000,000,000 hertz). If the speed of an automobile had increased since 1971 at the same pace, you would now be able to drive from San Francisco to New York City in less than 10 seconds.***

Semiconductor manufacturing processes

10 µm — 1971
3 µm — 1975
1.5 µm — 1982
1 µm — 1985
800 nm (0.80 µm) — 1989
600 nm (0.60 µm) — 1994
350 nm (0.35 µm) — 1995
250 nm (0.25 µm) — 1998
180 nm (0.18 µm) — 1999
130 nm (0.13 µm) — 2000
90 nm — 2002
65 nm — 2006
45 nm — 2008
32 nm (double patterning) — 2010 **Intel's Core i3 ,i5 and i7**
22 nm (end of planar bulk CMOS) — 2011 (per ITRS)
16 nm (transition to nanoelectronics) — approx. 2013
11 nm (nanoelectronics) — approx. 2015
7.9 nm (hypothesized end to scaling in silicon)

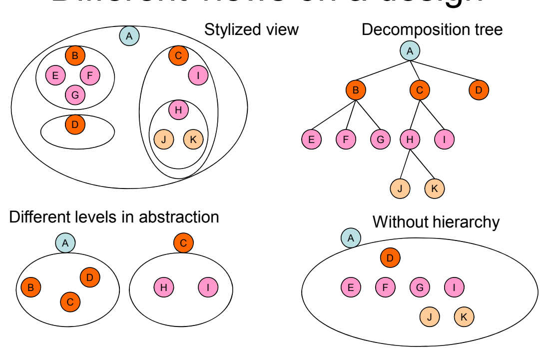
1 nanometre =

SI units 1×10^{-9} m 1×10^{-3} µm
US customary / Imperial units $3.281\times10^{\Delta-9}$ ft $39.37\times10^{\Delta-9}$ in

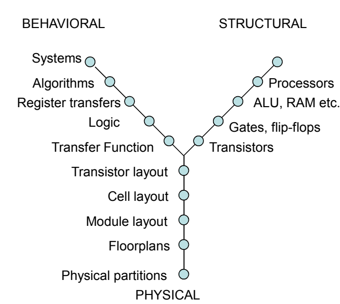
The Digital Design Problem

- Realization of a specification
- Optimization of
  - Area
  - Speed
  - Power dissipation
  - Design time
  - Testability

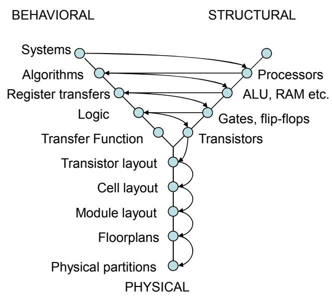
Different views on a design



Y-CHART



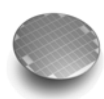
Top-down structural decomposition / bottom-up layout reconstruction



Design methods

- Full custom
  - Maximal freedom
  - High performance blocks
  - Slow
- Semi-custom
  - gate arrays (mask vs field programmable)
  - standard cells
  - parametrizable modules (adder, multiplier, memories)

Digital hardware



- Technology evolution
  - Before 1960s, transistors and resistors as individual parts
  - Integrated circuits: a number of transistors on a single chip (SSI, MSI)
  - By 1970, a microprocessor on a single chip
  - In the early 1990s, a few million transistors on a chip (VLSI)
  - By the late 1990s, 10 million transistors on a chip
- Moore's law
  - The number of transistors on a chip will double every 1.5 to 2 years. (from 1980 to 2010)
- Question ?
  - What is the main reason that the # of transistors on a chip grows so dramatically?

Digital hardware

- Implemented:
  - On a single chip
  - Multiple chips on a PCB (printed circuit board)
- Available chips that may be used
  - Standard chips: <100 transistors, simple function, fixed functionality.
  - PLD (programmable logic devices): larger size, general structure, user defines the function by programming PLD.
  - Custom design (no chip available): manufacturing a chip on semiconductor material directly based on logic function, area efficient, high speed.

Digital hardware

- comparison

	Advantages	Drawbacks	examples
Standard chips	Simple, cheap	Area inefficient, fixed function	74LS04
PLD	Programmable, A wide range of sizes, short time to market	Area inefficient, not high speed	FPGA
Custom design	Area efficient, High speed,	Expensive, Long time to market	Some image processing chips

How to design a digital circuit with millions of transistors

- A good designer should
  - ✓Understand the specifications of product
  - ✓Convert the specifications to logic problems
  - ✓Grasp principles of logic design
  - ✓Skillfully use CAD tools.

CAD Tools for Automation

- Algorithmic and system design: C, DFG (Design Flow Graph), HDL (Hardware Description Language) / Simulator
- Structural and logic design: Schematic editor / Simulator
- Transistor level design: Schematic editor / Simulator
- Layout design: Layout editor

SYNTHESIS

- High level: synthesis from algorithmic behavioral level to structural logic level
- Logic: synthesis from structural level to gate level

HARDWARE DESCRIPTION LANGUAGES

- HDL are used to describe the hardware for the purpose of modeling, simulation, testing, design, and documentation.
  - Modeling: behavior, flow of data, structure
  - Simulation: verification and test
  - Design: synthesis

Purpose of Verilog

- **Problem**
  - Need a method to quickly design, implement, test, and document increasingly complex digital systems
  - Schematics and Boolean equations inadequate for million-gate IC
- **Solution**
  - A hardware description language (HDL) to express the design
  - Associated computer-aided design (CAD) or electronic design automation (EDA) tools for synthesis and simulation
  - Programmable logic devices for rapid implementation of hardware
  - Custom VLSI application specific integrated circuit (ASIC) devices for low-cost mass production

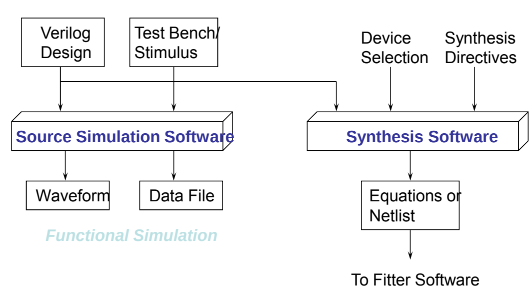
Verilog: Why to use?

- Reasons to use Verilog
  - Power and flexibility
  - Device-independent design
  - Portability among tools and devices
  - Device and tool benchmarking capability
  - VLSI ASIC migration
  - Quick time-to-market and low cost (with programmable logic)
- Problems with Verilog
  - Loss of control with gate-level implementation (so what?)
  - Inefficient logic implementations via synthesis (engineer-dependent)
  - Variations in synthesis quality among tools (always improving)

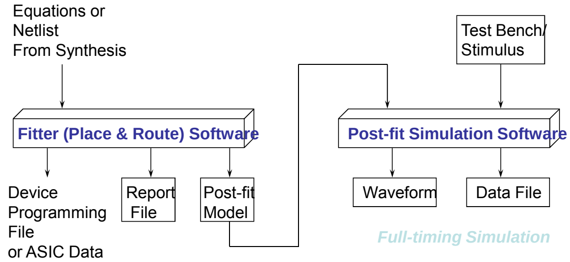
Design Flow in Verilog

- Define the design requirements
- Describe the design in Verilog
  - Top-down, hierarchical design approach
  - Code optimized for synthesis or simulation
- Simulate the Verilog source code
  - Early problem detection before synthesis
- Synthesize, optimize, and fit (place and route) the design for a device
  - Synthesize to equations and/or netlist
  - Optimize equations and logic blocks subject to constraints
  - Fit into the components blocks of a given device
- Simulate the post-layout design model
  - Check final functionality and worst-case timing
- Program the device (if PLD) or send data to ASIC vendor

Design Tool Flow (1)



Design Tool Flow (2)



How to design a digital circuit with millions of transistors

- Basic design loop

1) generate initial design, manual efforts
2) simulation, extensively use CAD tools.
3) If errors, redesign

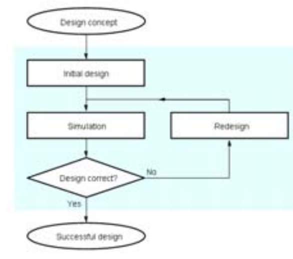
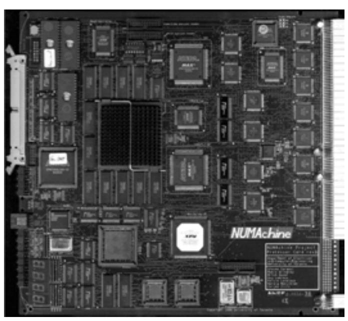


Figure 1.4. The basic design loop.

A PCB in a computer system



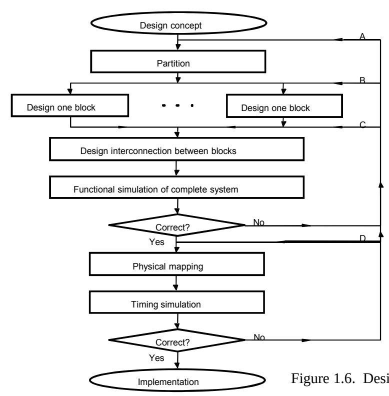


Figure 1.6. Design flow for logic circuits.

Summary

- Introduction to logic design
  - Life needs digital
  - History of digital circuits
  - Design methodology
- The Digital digital design problem is described.
- Y-Chart and the different levels of abstraction are explained.
- VLSI design automation and CAD tools are mentioned.
- Purpose and background of Verilog are explained.
- Verilog and programmable logic are the best current solution for rapid design, implementation, testing, and documenting of complex digital systems.
- A standard 6-step design synthesis process is used with Verilog.
- The general flow of information through standard Verilog synthesis CAD tools was described.