ELEC 271

Digital Systems

Fall, 2025

Queen's UNIVERSITY

Course Administration

- Instructors:
- Ryan E. Grant

  Email: ryan.grant@queensu.ca

  Office hours: by appointment

- Kleber Cabral

  Email: kleber.cabral@queensu.ca

  Office hours: by appointment

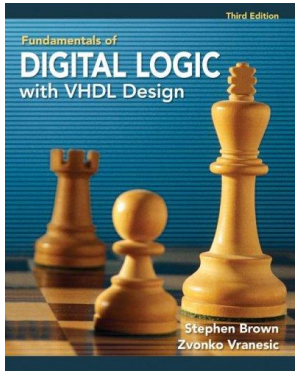
- TA (coordinators):
  - Jose Gonzales
  - pmt4@queensu.ca
- Course Web site: **OnQ**
- Prerequisites: APSC-171, APSC-172, APSC-174
- Lectures & tutorials: See your schedule!
- Problem Sets: 7 sets; no marking – solutions will be posted after the tutorial

Course Administration (Cont'd)

- Midterm: **in class – October 28th – room assignments TBD**
  - Covers all material up to near the midterm. Information on exactly what is fair game for the midterm will be provided closer to the date.
- Lecture slides:
  - Available on OnQ as the term progresses. Slides are used to teach the fundamental concepts, and for quick reference to code, block diagrams, timing diagrams, etc.
  - There is a **significant amount of design material that will be covered in class but not necessarily included in the slides**. You're highly encouraged to come to class as much of the material in this course is best covered using in-person blackboard solutions.

Course Administration (Cont'd)

- Grading:
  - Midterm: 30%
  - Labs: 30%
  - Final exam: 40%



- Textbook (Required): available in campus bookstore
  - Fundamentals of Digital Logic with Verilog Design ISBN: 9780073380544 Stephen Brown and Zvonko Vranesic, McGraw-Hill

Course Objectives

- Learn the basic principles and limitations behind digital logic systems
- Understand the current technology being used in this area
- Understand the fundamental electronic aspects of digital circuits
- Analyze, synthesize, and optimize multilevel digital systems
- Design arithmetic, combinational, and sequential digital systems
- Understand how digital circuits are designed in industry using hardware description languages (VHDL) and CAD tools
- Understand the real-world applications of what you are learning
- Develop the ability to creatively design new systems using your knowledge gained in this course

Course Outline

- This is a tentative course outline. Some topics and sections may be added or removed.
- Section 1: Design Concepts
  - Design process, logic circuits, binary numbers
- Section 2: Introduction to Logic Circuits
  - Logic gates, functions, truth tables, Boolean algebra, analysis and synthesis of simple circuits, sum-of-products, product-of-sums, CAD tools, introduction to VHDL
- Section 3: Implementation Technology
  - Transistors operating as switches, NMOS and CMOS logic gates, ICs, PLAs, PALs, CPLDs, FPGAs, cell and gate arrays, voltage levels, noise margins, fan-in and fan-out, tri-state buffers
- Section 4: Optimized Implementation of Logic Functions
  - Karnaugh map, minimum-cost implementation techniques, multilevel synthesis, more VHDL

Course Outline (Cont'd)

- Section 5: Number Representation and Arithmetic Circuits
  - Representation of numbers, simple adder/subtractor circuits, carry-lookahead adder, more VHDL
- Section 6: Combinational-Circuit Building Blocks
  - Multiplexers, decoders, demultiplexers, encoders, more VHDL
- Section 7: Flip-Flops, Registers, Counters, and a Simple Processor
  - Latches, D flip-flop, T flip-flop, JK flip-flop, registers, shift registers, counters, bus structure, timing analysis, more VHDL
- Section 8: Synchronous Sequential Circuits
  - Moore/Mealy synchronous sequential circuits, state minimization, finite state machines, more VHDL

The lab component of this course is essential to understanding digital systems. There will be labs on basic digital logic, VHDL, and Altera Quartus II CAD environment using Altera DE2 evaluation board with many features including a Cyclone FPGA.

A little bit about me – Dr. Grant

- Joined Queen's in 2021
- Long history at Queen's arrived here in 2000 originally
  - I get it, I've been exactly where you are now, it's not easy.
- Spent a decade in US at major research lab, working with all major players in the tech industry (AMD, Intel, IBM, HPE/Cray, Amazon, Google, Nvidia, Fujitsu, Cisco, Broadcom, etc.)
  - Plenty of experience seeing how different companies do things and how teams deliver
- Invented lots of stuff and wrote/contributed to several standards
- Every supercomputer in the world has something I designed in it
  - Networks, communicators middleware
  - Power measurement and control systems
- Industry-focused – Course designed like what you'll be expected to do in your future careers (like in labs), with design examples from real-world

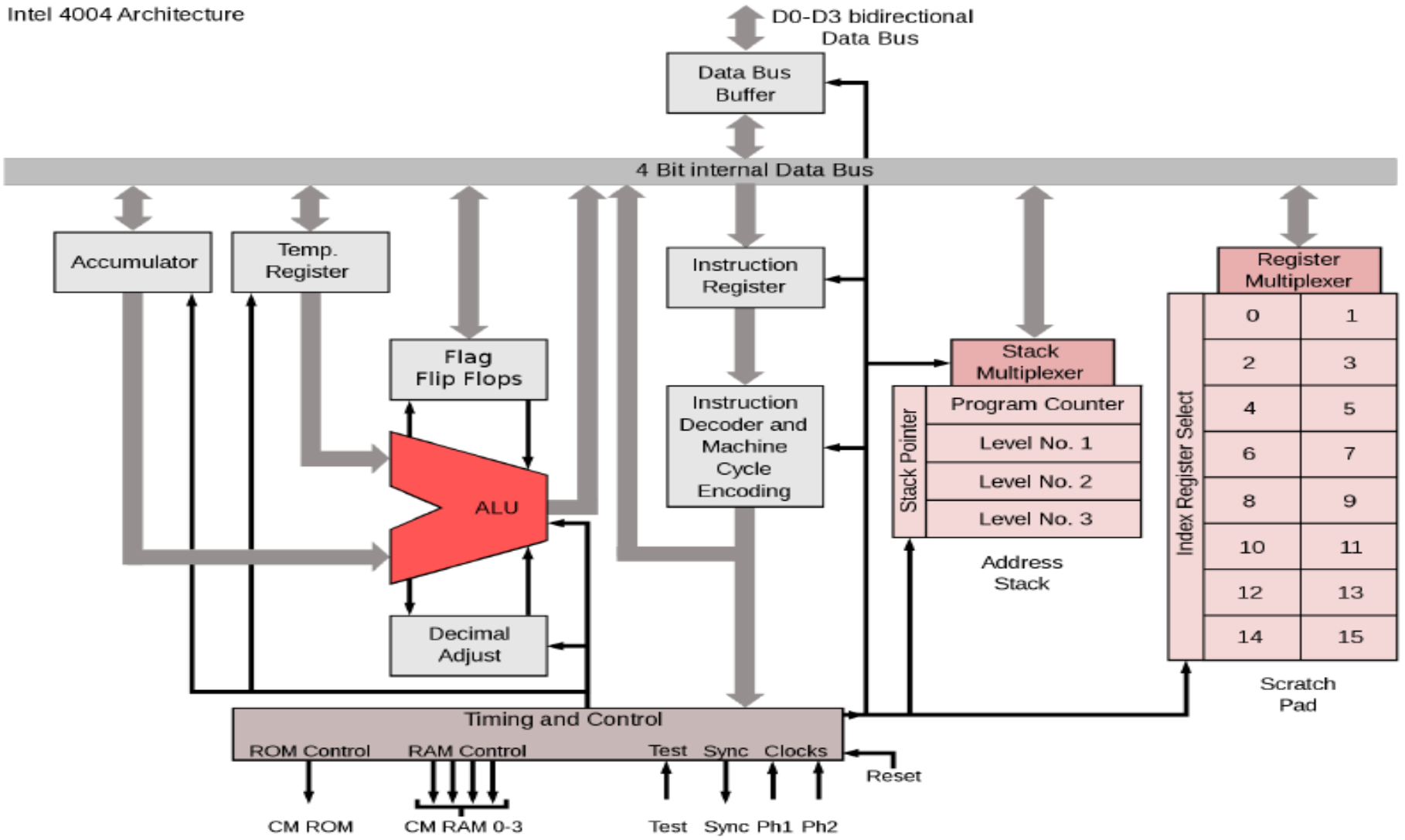
A little bit about me – Dr. Cabral

- Joined Queen's in 2022
- Bachelor in Computer Eng.
- Masters and Ph.D. in Electrical and Comp. Engineering.
- Research focus on Robotics and Artificial Intelligence.
  - Interested in multi-agent interaction for large scale.
  - Focusing on safety and intelligent interaction between humans and robots in crowded spaces.
- Teaching philosophy is application-focused – I will focus my teaching on your future careers in mind, trying to bring real-world experience with design examples from real-world

New Year New Experiences

- It’s a first day for a lot of things
  - First second year digital course!
- ELEC271 is fundamental, so you’ll use this stuff throughout your career
  - First time thinking about this sort of subject
  - Can be difficult as it’s the first time you’re exposed to these kind of ideas
  - Allocate time to think about these ideas yourself and take the time to read other materials as well if it’s not ”clicking”
- Some of the material in this course you need to learn even if it’s not on the exam
  - Important for your future careers, hard to test
  - Probably some of the most important material long-term
- Don’t forget to enjoy Frosh week!

Common Theme in the Course



ELEC 271

Digital Systems

Lecture Outline

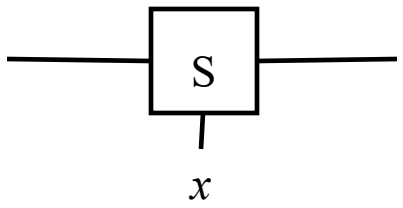
- Digital Hardware
- Moore's Law, Technology Trends, Chip Types
- Layers of Abstraction
- Design Process
- Scope of this Course
- Design Flow for Logic Circuits
- Fixed Point Numbers and Positional Number Representation

Read Chapter 1 of the textbook.

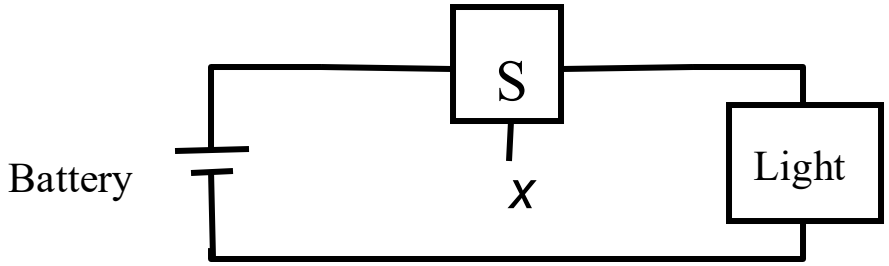
Digital Hardware

- An analog signal does not rely on discrete values, it can be an essentially infinite number of values (or levels) ex. A radio signal.
- A digital signal is one where there are agreed upon discrete values; for example, a **binary** system where levels are “1” or “0”. In practice an example is a voltage < 0.5V is considered “0”, while a voltage > 2.8V is considered a “1”.
- A switch is a basic element in implementing a digital system.

Two states of a switch



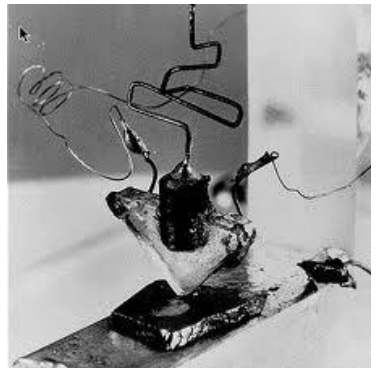
Symbol for a switch



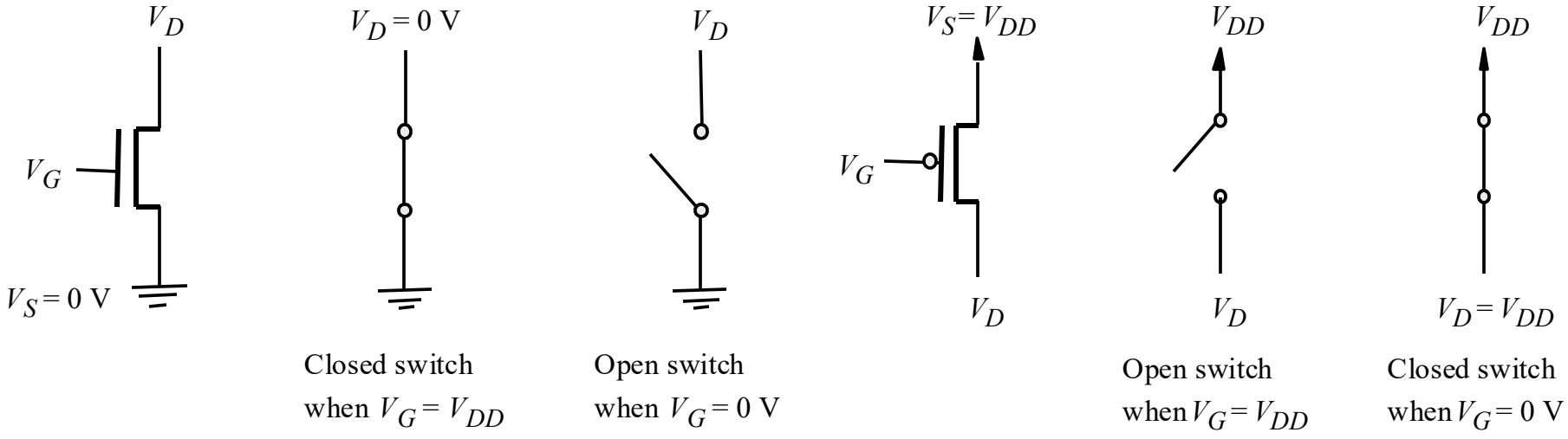
Simple connection to a battery

Digital Hardware (Cont'd)

- Early digital systems used electromechanical relays (vacuum tubes). Modern digital systems use transistors as the switching element.
- Up to 1960's, discrete transistors (invented in 1947 at Bell Laboratories) were used to implement digital systems.
- There are two major "types" of transistors, NMOS and



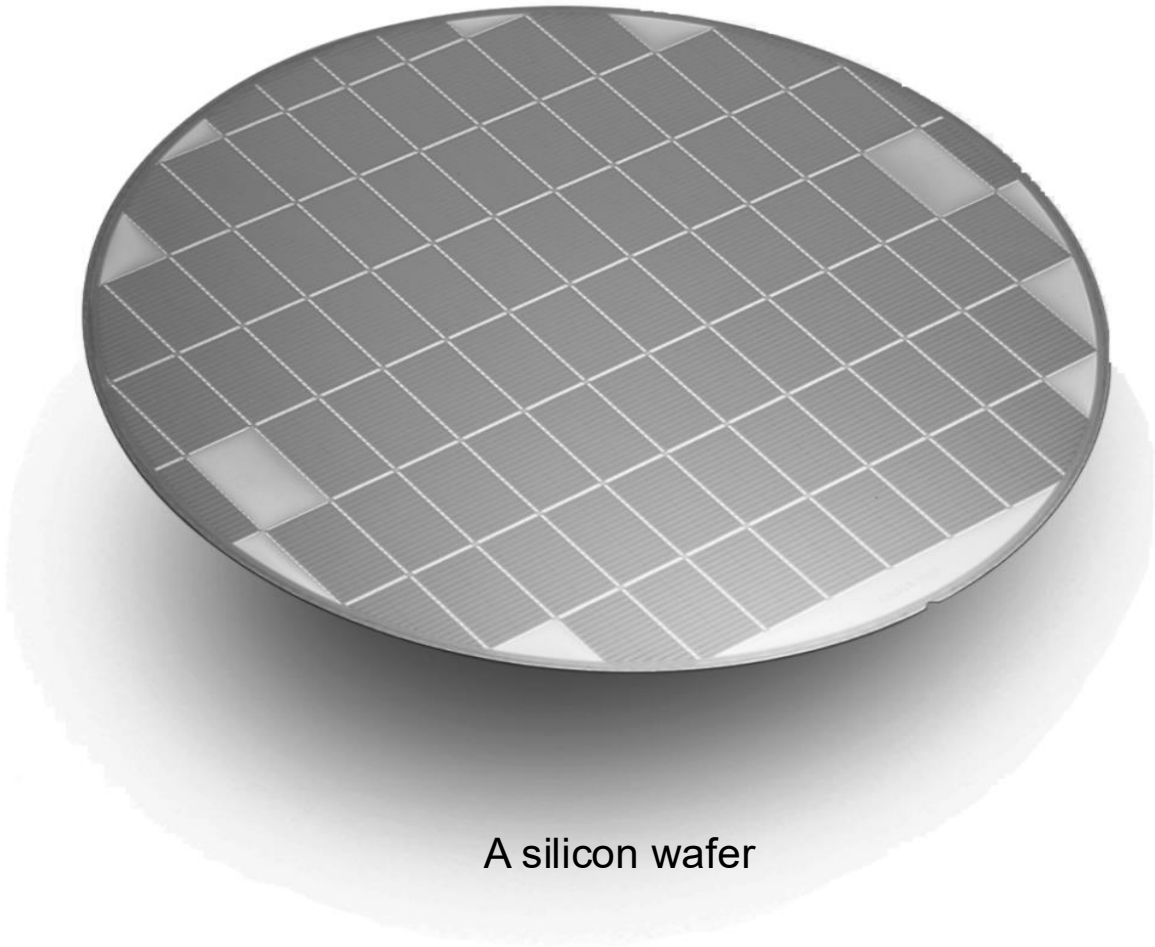
PMOS



Simple use of NMOS and PMOS transistors in digital systems

Digital Hardware (Cont'd)

- Integrated Circuit (IC): IC technology was born in early 1960's (in fact, in 1958 by Texas Instruments), and the first microprocessor or



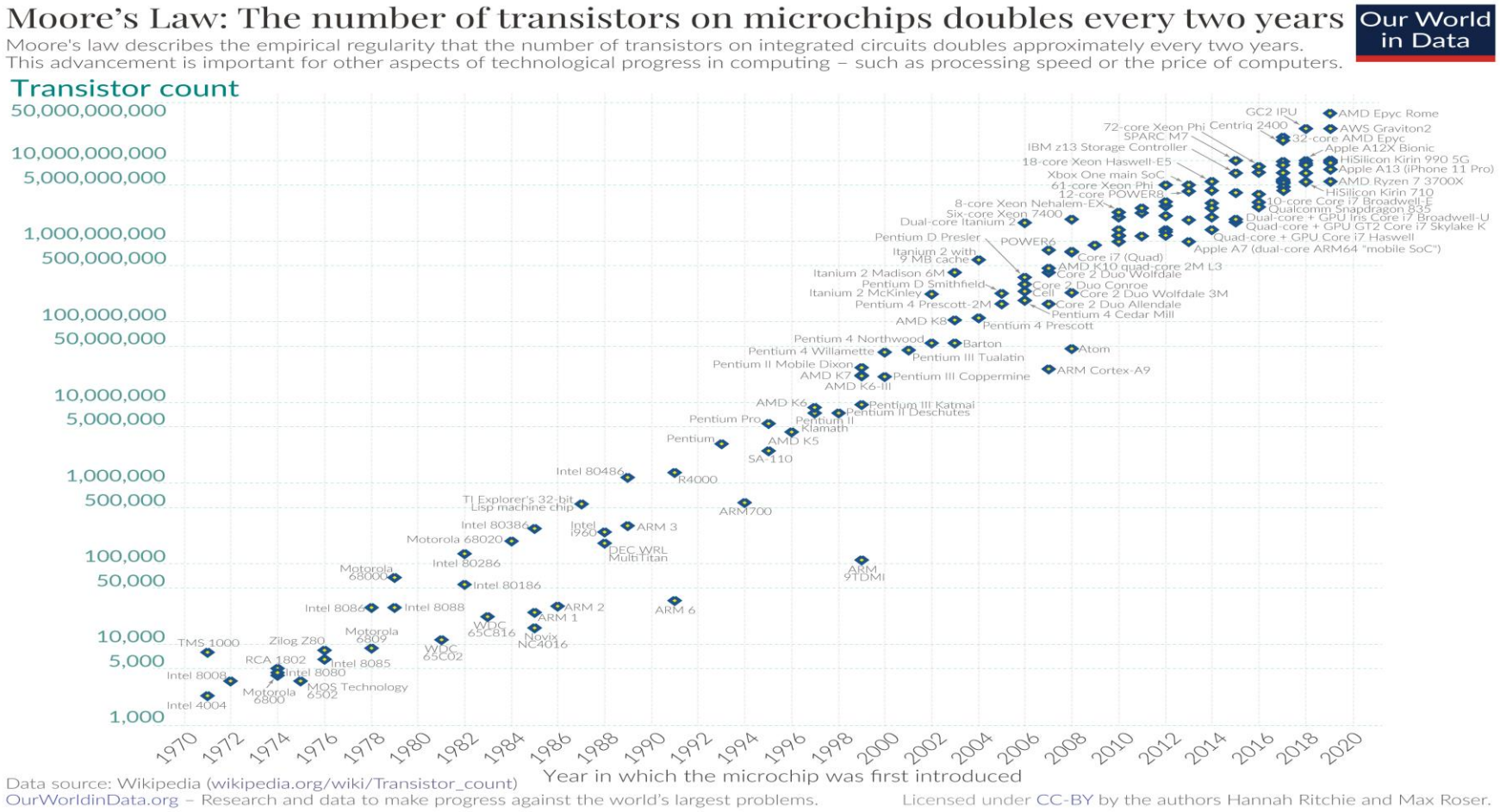
A silicon wafer

Zooming in a CPU

- https://www.youtube.com/watch?v=xZIZ3LWyhvc

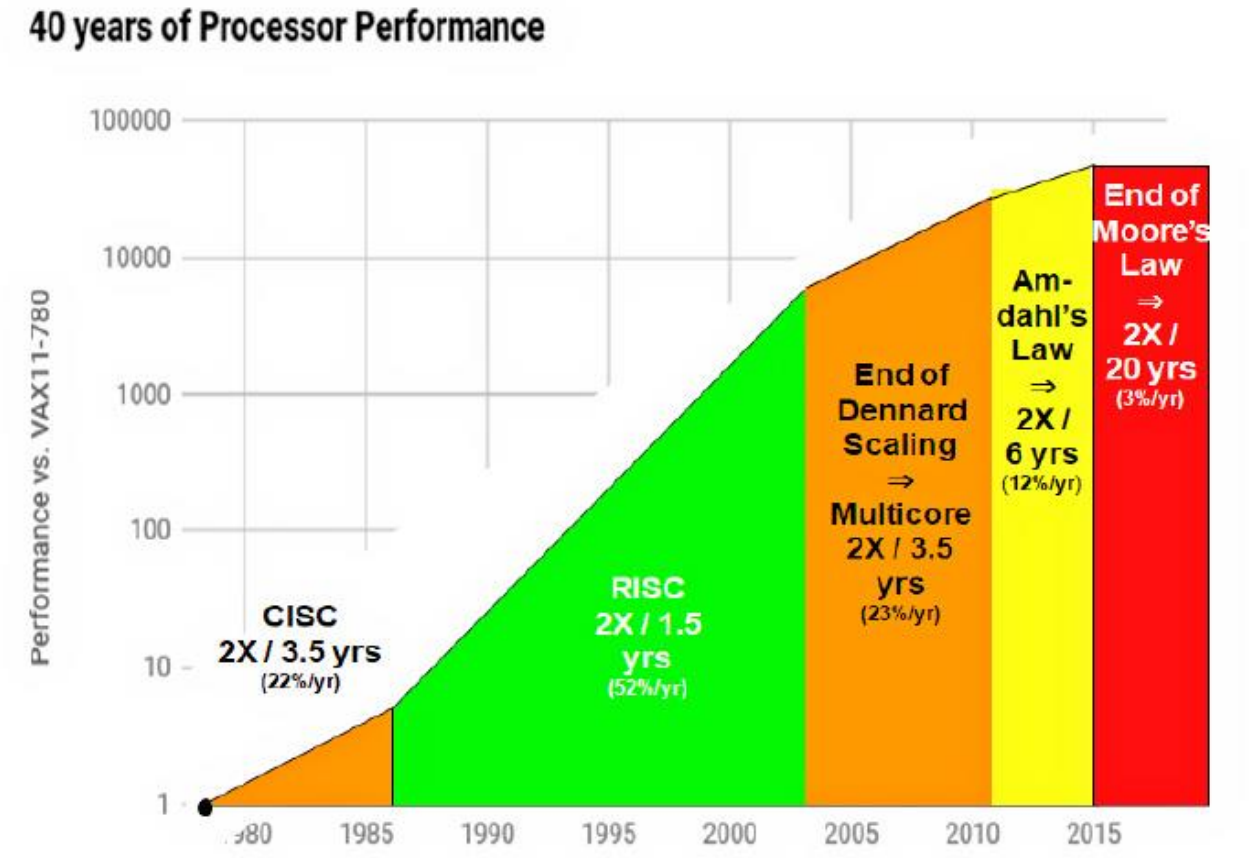
Moore’s Law

- Moore’s Law – chip density: an empirical observation that the number of transistors on a chip is doubling every 18 months. Intel expects new transistor tech to lead to trillion transistor chips by 2030!



Technology Trends

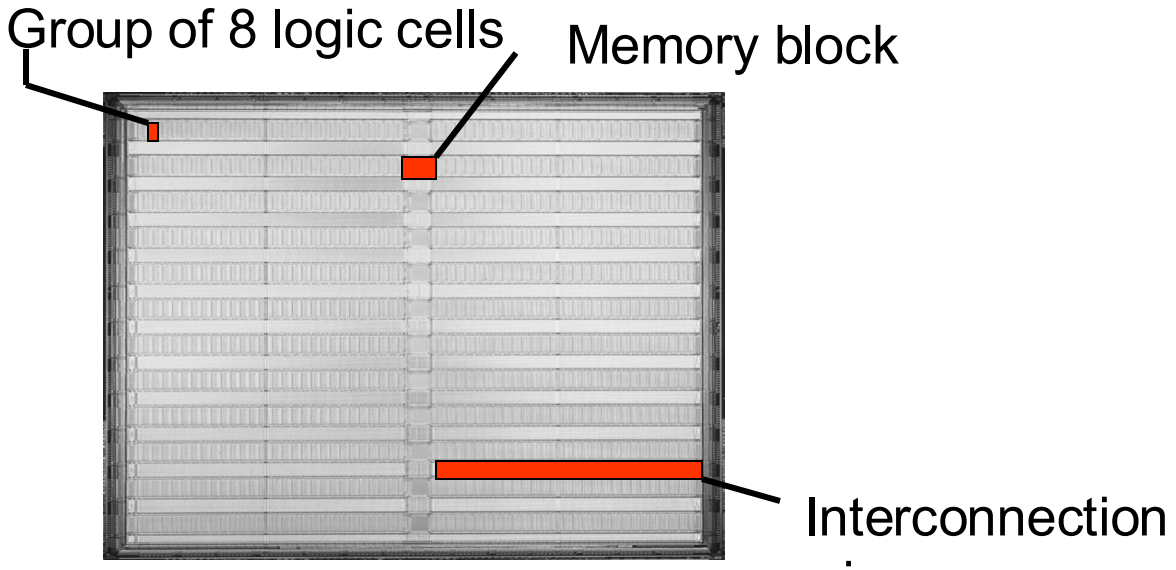
- Everyone predicts the end of Moore's Law, and yet we keep going.
- Patterson even thinks so, he's one of the inventors of RISC!



Source: John Hennessy and David Patterson, Computer Architecture: A Quantitative Approach, 6/e. 2018

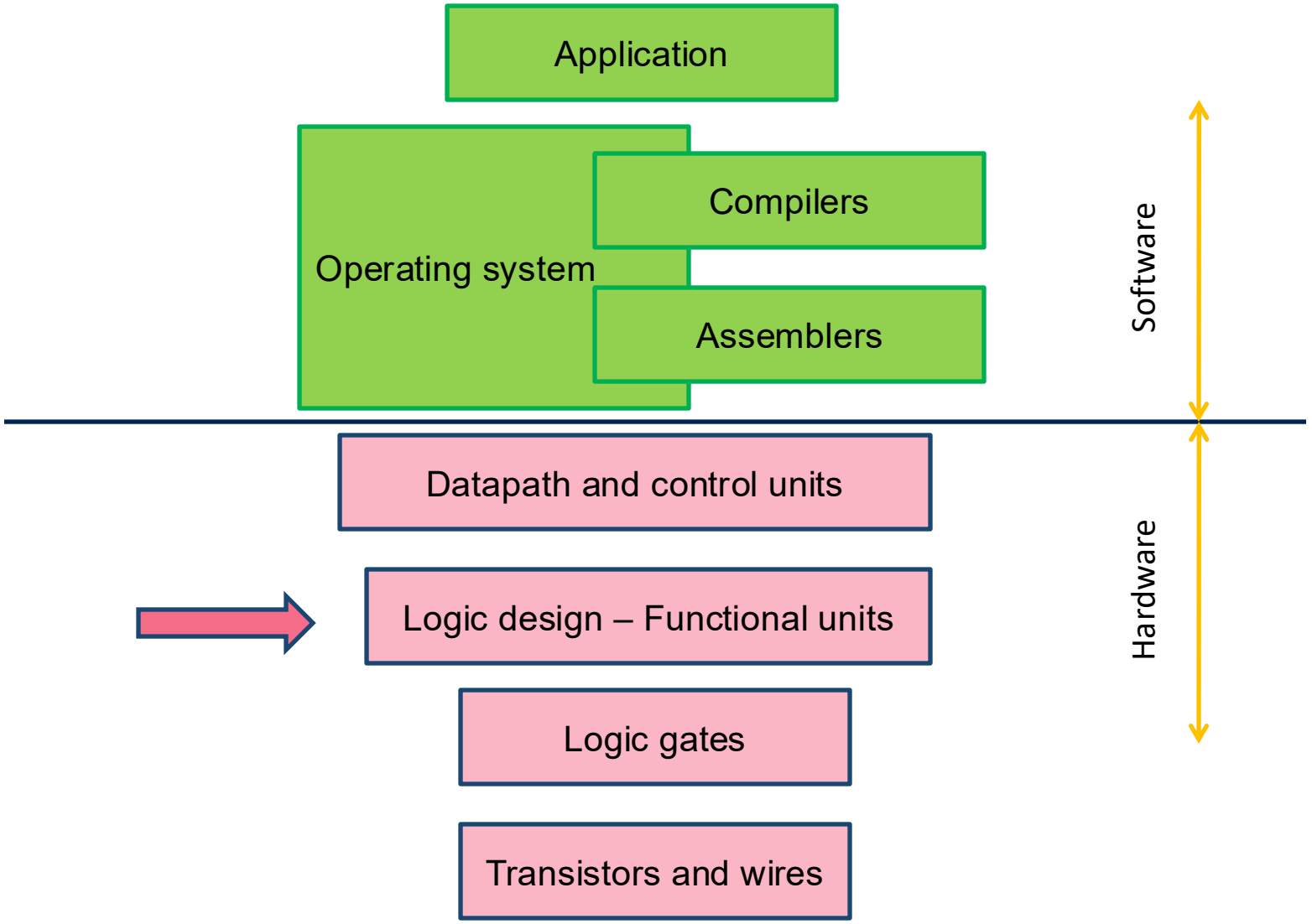
Chip Types

- Standard chips: 74 series, 40/45 series (small and fixed functionality)
- Programmable logic devices: Programmable Logic Devices (PLDs), complex PLDs (CPLDs), field-programmable gate arrays (FPGA) – programmable switches consume valuable space.
- Custom-designed chips: Application-specific ICs (ASICs), Gate Array, Standard Cells – design optimized for a particular task; high time-to-market.



A field-programmable gate array chip

Layers of Abstraction



Design Process

- Logic design: given a desired system specification, design a digital system that meets some criteria, such as speed, cost, power, etc.
  - Choose the logic components
  - Choose implementation technology
- Goals:
  - Design should be correct and meet the specification.
  - It should be cost-effective; costs can be classified as time-to-market, power dissipation, area, etc.
  - Design should be easy to upgrade or enhance.
  - Design should be reliable, and work without error.
    - For these, designers may need to maximize speed, optimize logic, minimize power, lower cost of manufacturing, lower time-to-market, use CAD tools, etc.

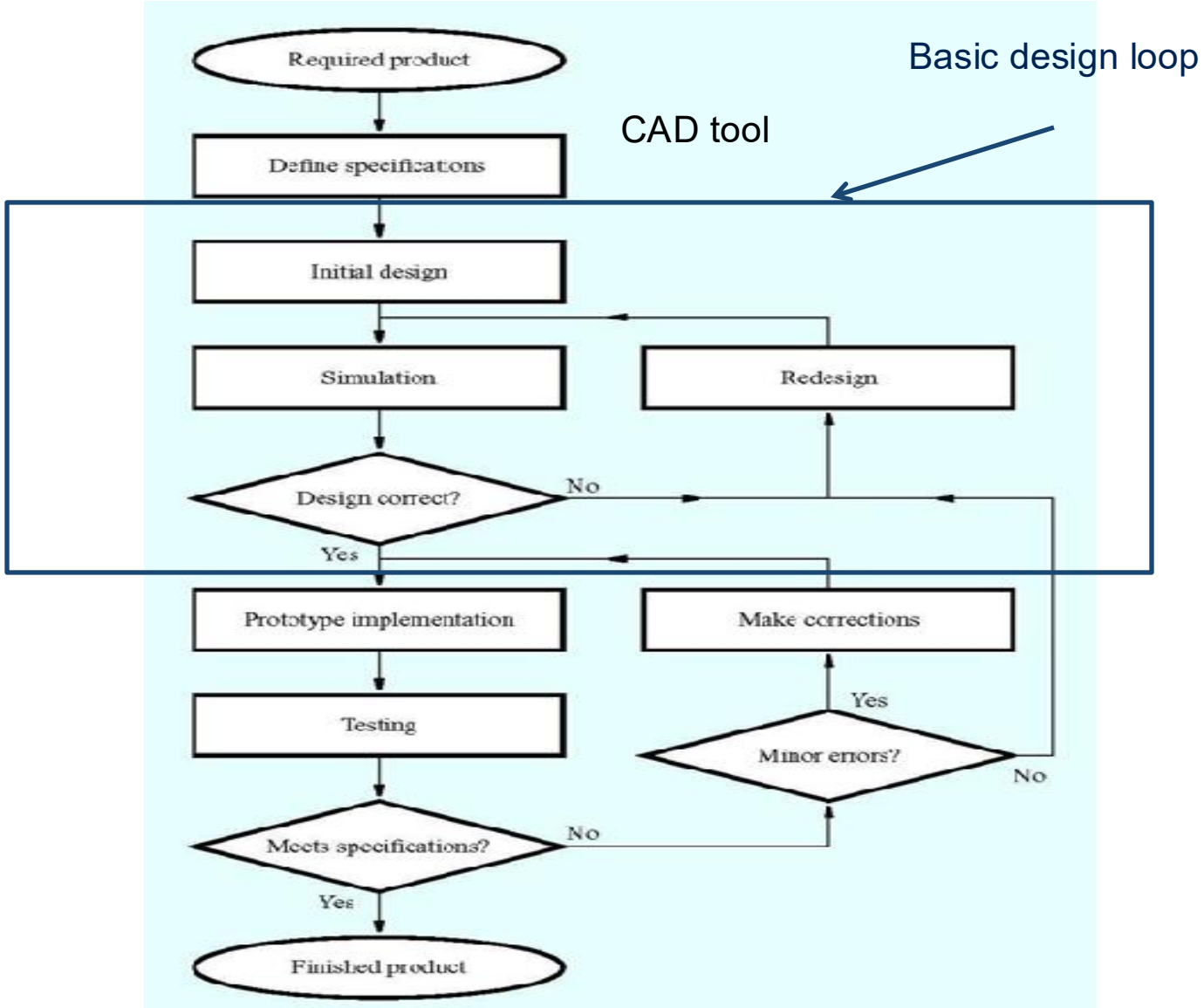
Design Process (Cont'd)

- Significant changes such as access to VLSI and ULSI technology, cheaper and faster time-to-market designs are happening right now in the hardware design phase.

- This is only possible because of significant progress in:
  - Utility and acceptability of CAD environment over traditional manual methods
  - Simulation techniques to verify the large, complex logic design
  - Automatic synthesis methods
  - Using programmable chips such as FPGA

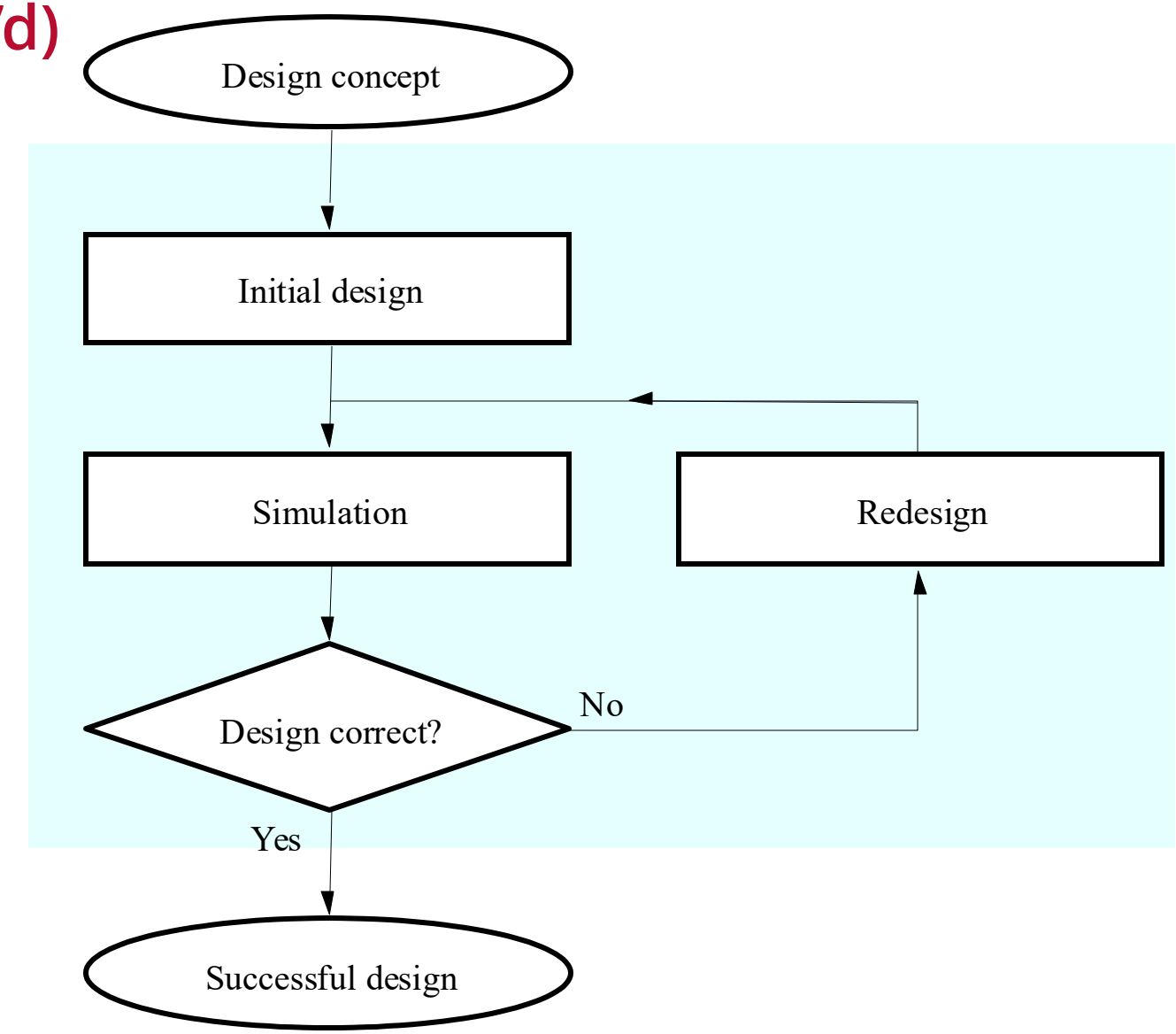
Design Process (Cont'd)

- A typical development process

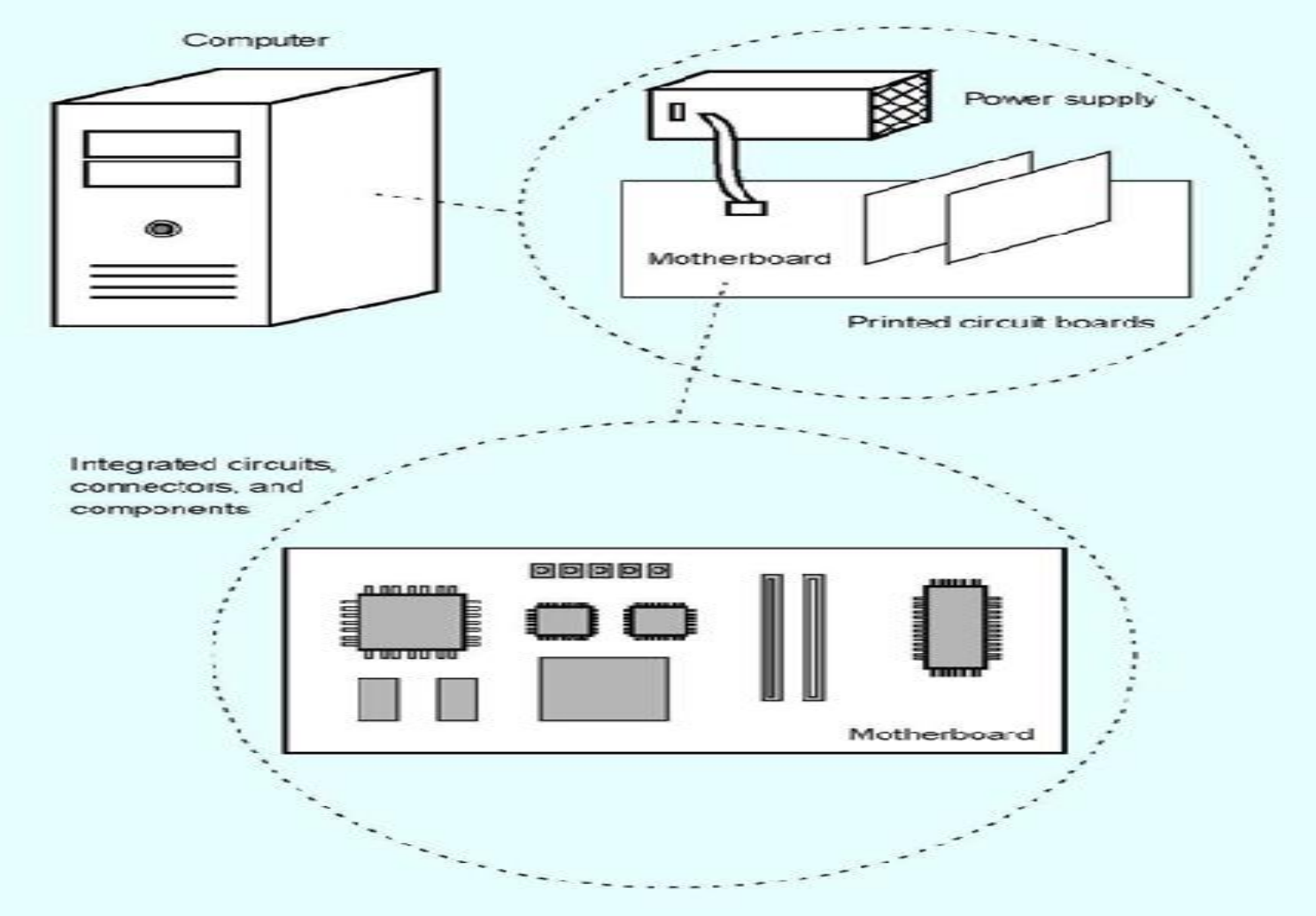


Design Process (Cont'd)

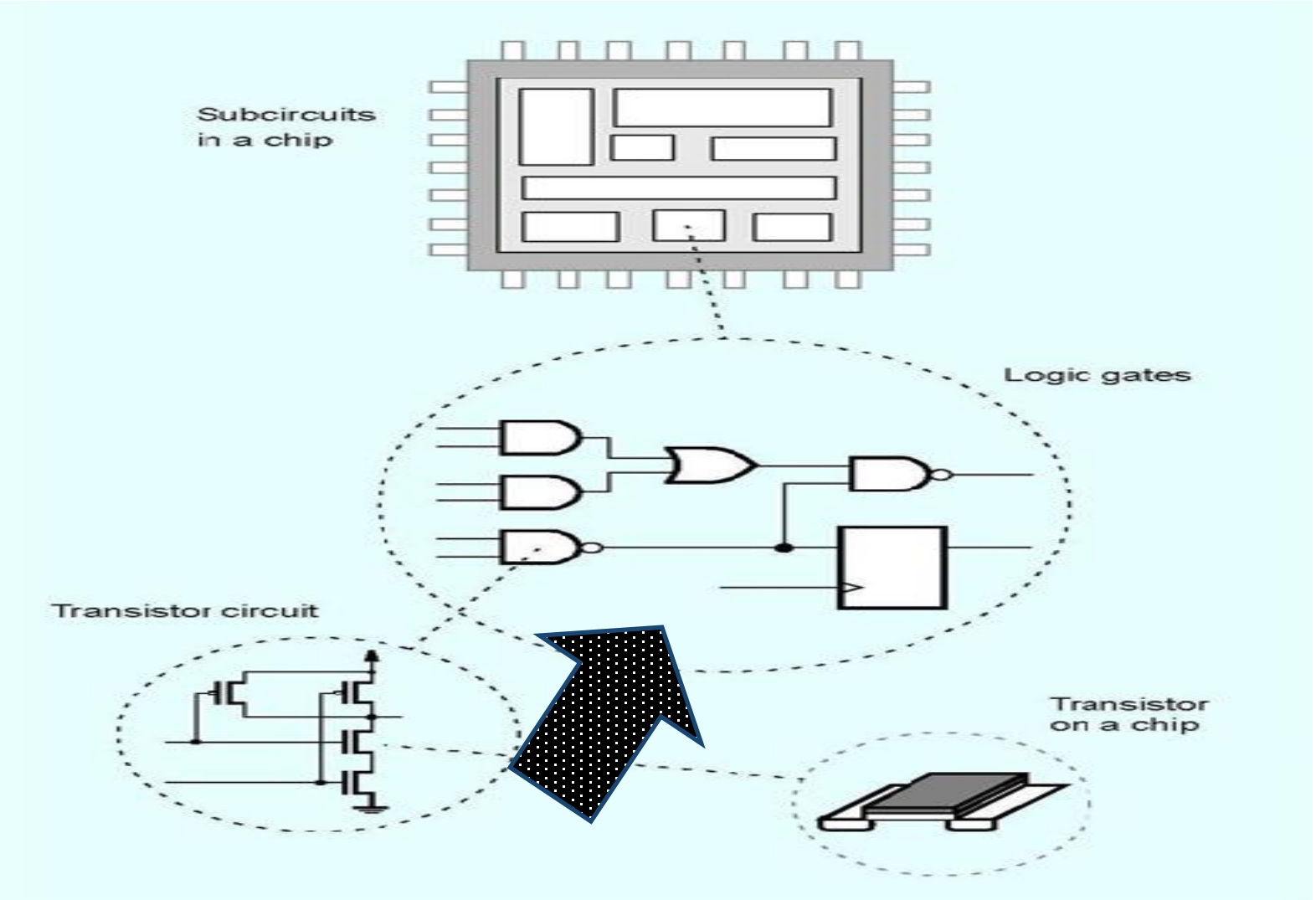
Note that while CAD tools provide efficiency and do many tasks automatically, the designer is still responsible for designing the digital system based on the knowledge of the underlying theory of logic design. It is also the designer who is familiar with the nature of systems, and chooses the different components.



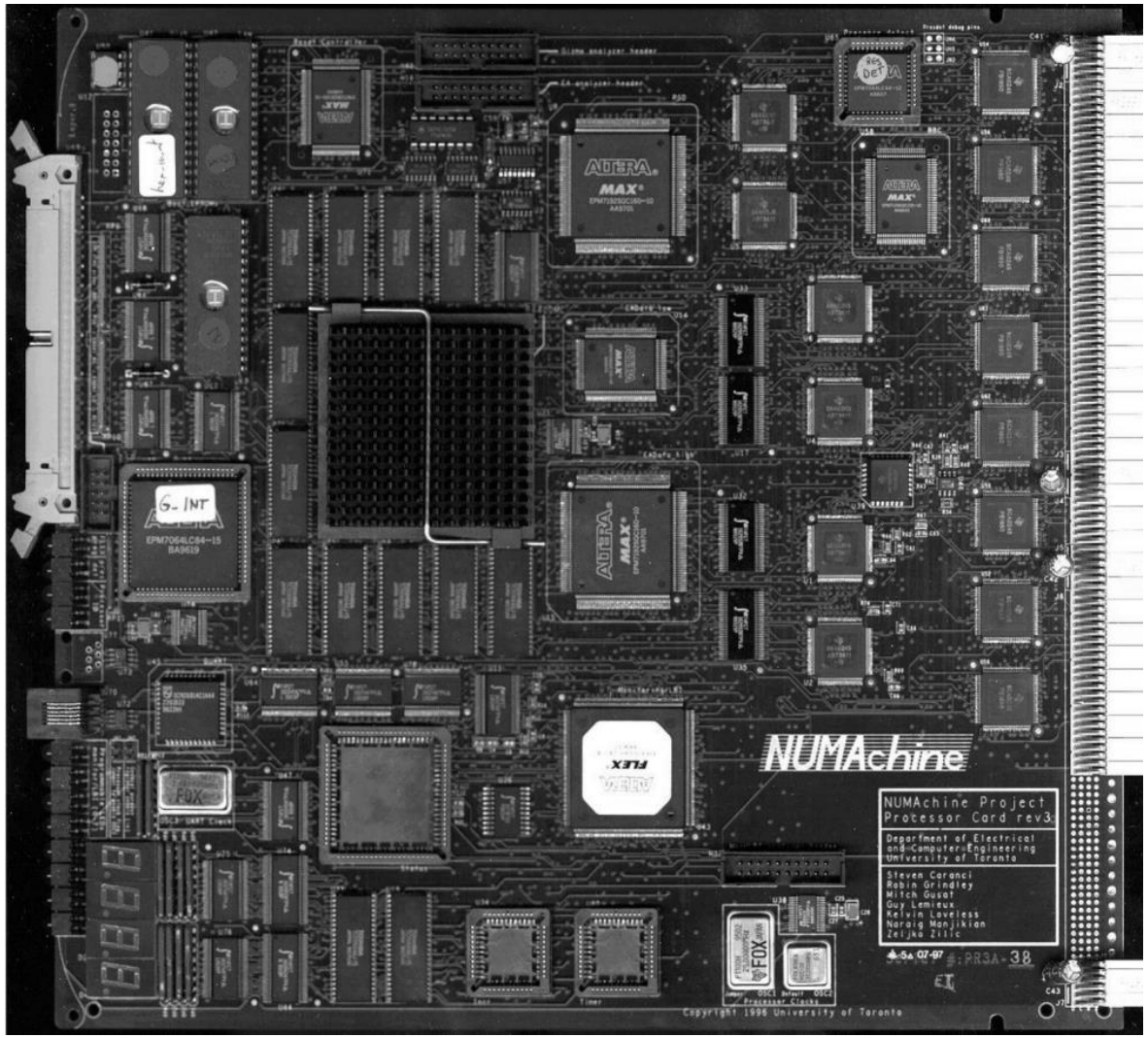
Scope of this course



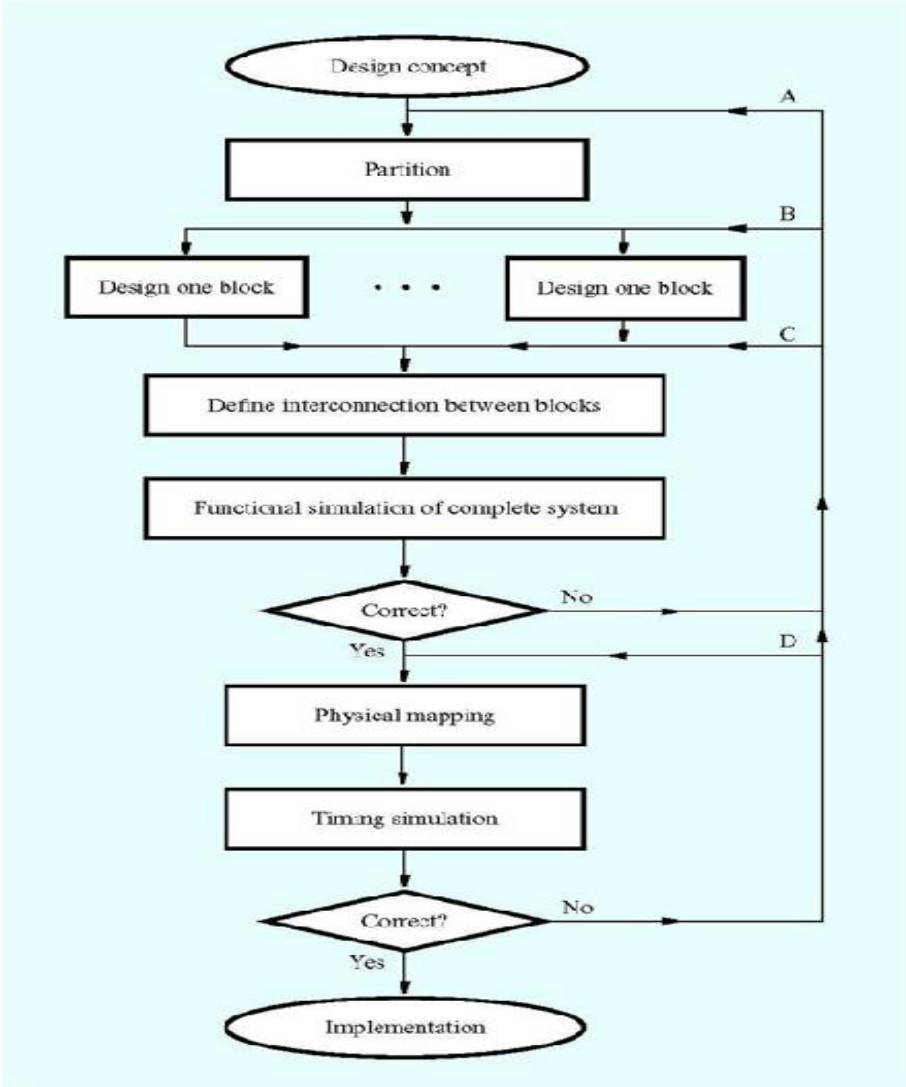
Scope of this course (Cont'd)



A Printed Circuit Board



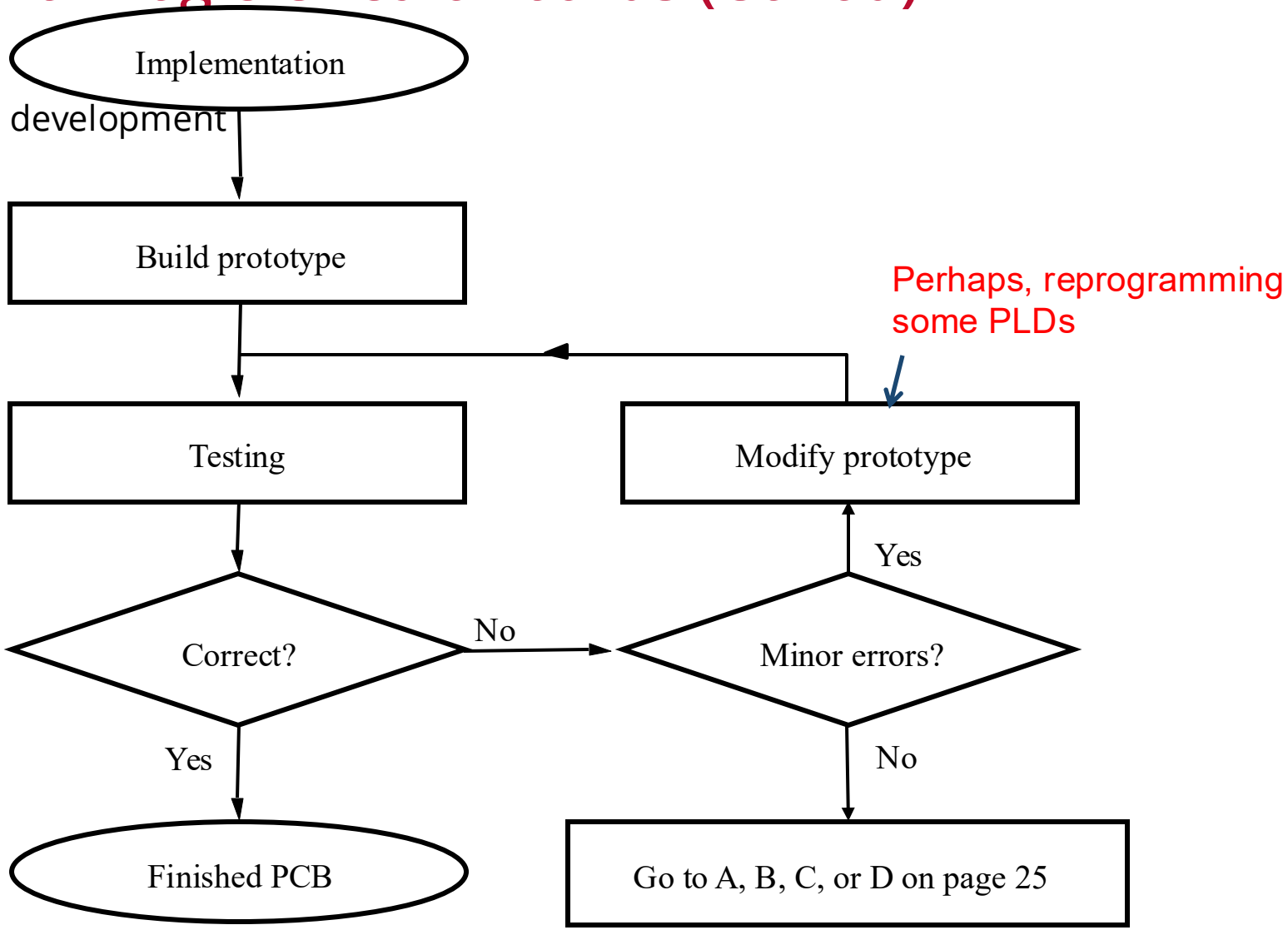
Design Flow for Logic Circuit Boards



A: Partitioning problem
B: Poor block design
C: Incorrect connections
D: Wire impact on signal delay

Design Flow for Logic Circuit Boards (Cont'd)

- Completion of PCB development



Binary Numbers

The Basics of Binary Numbers

Fixed Point Numbers and Positional Number Representation

- The base or radix of a number system defines the range of possible values that a digit may have: 0 - 9 for decimal; 0, 1 for binary.
- In any fixed point number system, each fixed point number has exactly the same number of digits, and the fixed point is always in the same place. For instance, 11.10, 01.10 in base 2, or 5.11, 0.34 in base 10.

$$Value = \sum_{i=-m}^{n-1} b_i . k^i$$

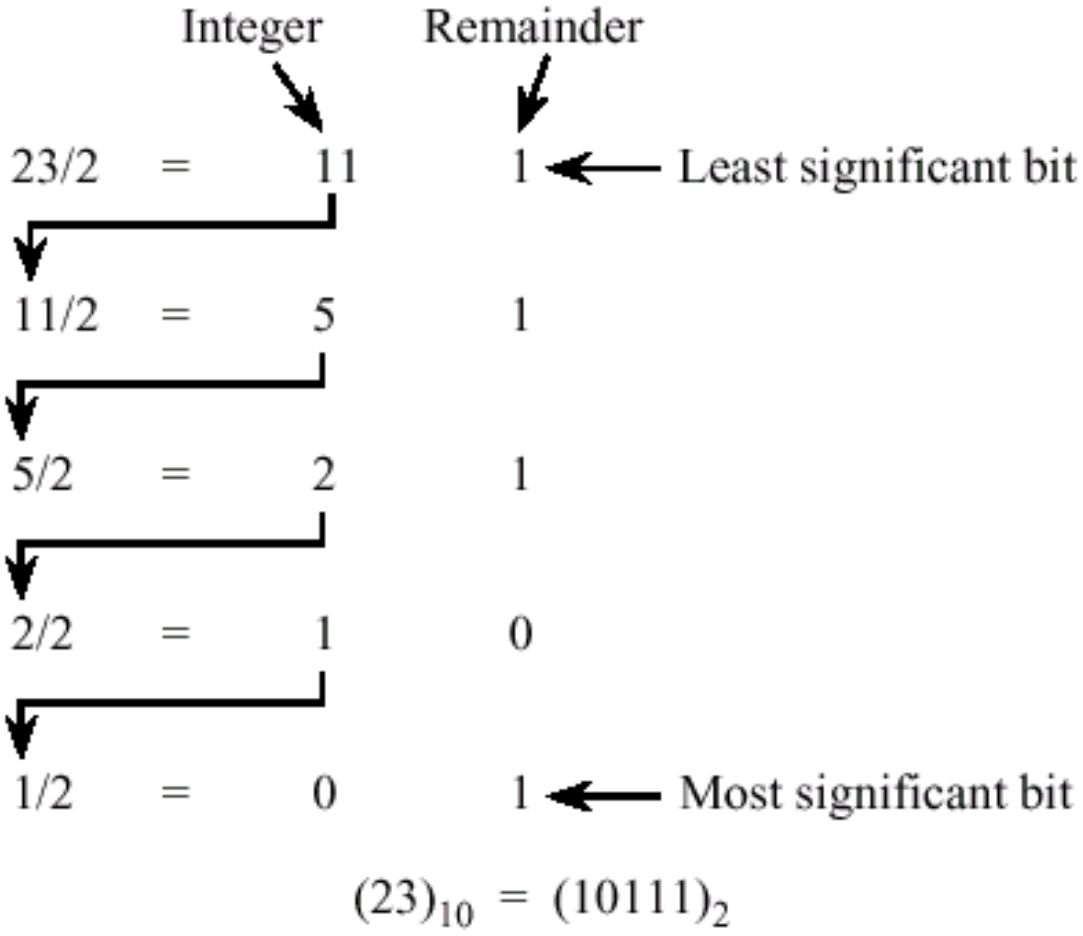
- The general form for determining the decimal value of a number in base *k* is given by:
- Example: Binary to decimal conversion
  - Using *polynomial method*, find the decimal equivalent of the number $(1010.01)_2$

$(1010.01)_2 = 1 \text{ X } 2^{-2} + 0 \text{ X } 2^{-1} + 0 \text{ X } 2^0 + 1 \text{ X } 2^1 + 0 \text{ X } 2^2 + 1 \text{ X } 2^3$

$= ¼ + 0 + 0 + 2 + 0 + 8 = (10.25)_{10}$

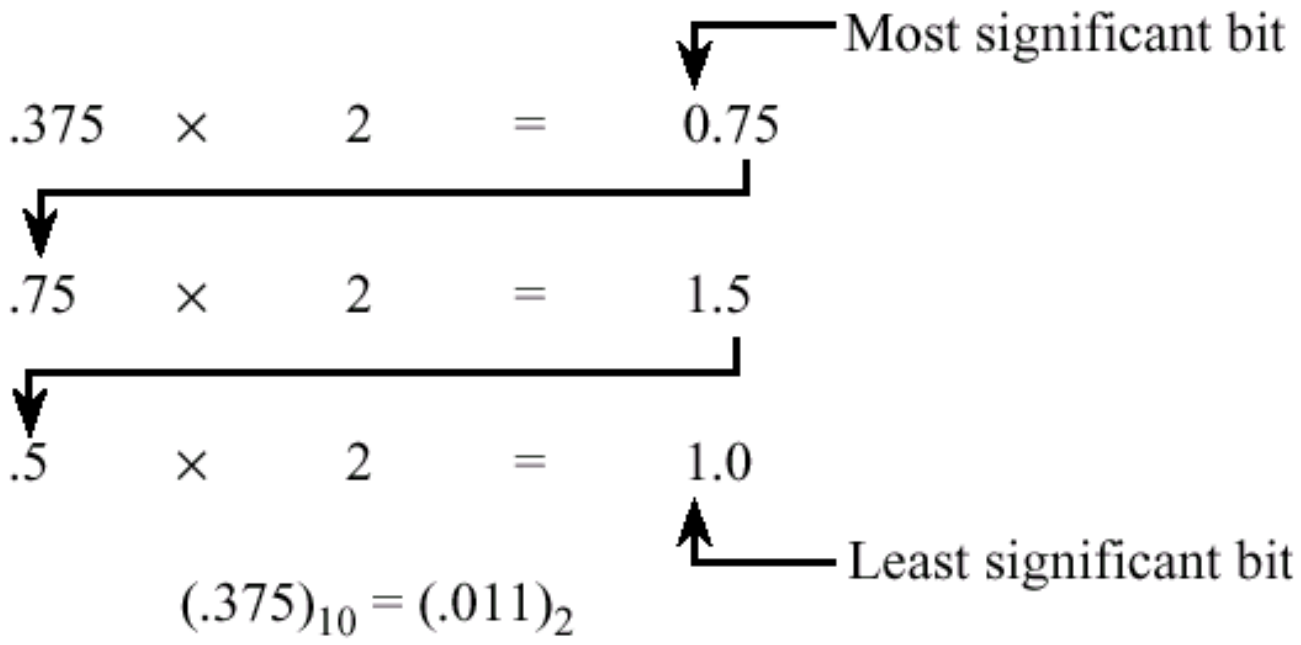
Base Conversion: Remainder Method

- Example: Decimal to binary conversion
  - Convert 23.375_{10} to base 2. Start with the integer part.



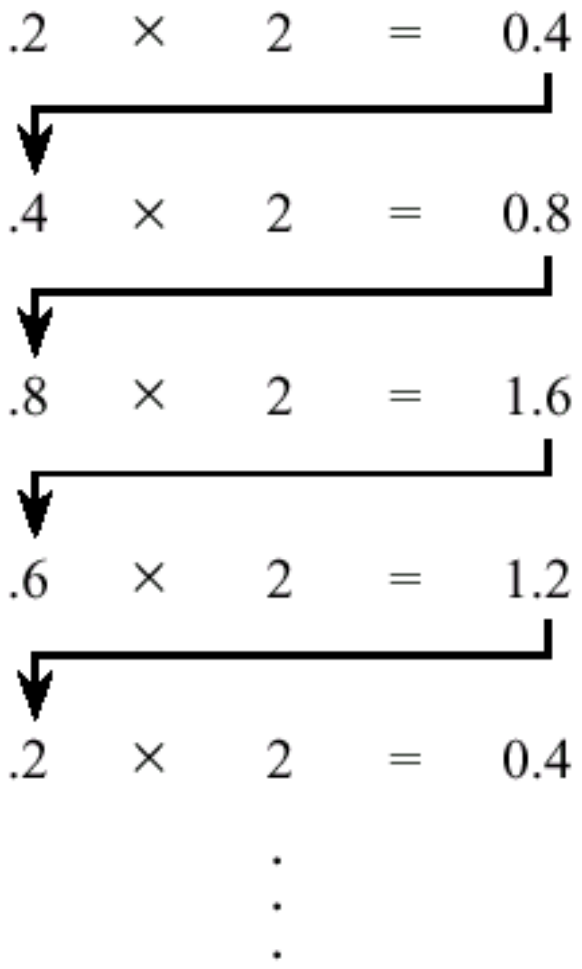
Base Conversion: Multiplication Method

- Now, convert the fraction part.



Non-terminating Fractions

- Not all terminating base 10 fractions have a terminating base 2 form.
- Example: convert $.2_{10}$ to base 2



Base 2, 8, 10, and 16 Number systems

- Most computers use base 2 for internal representation and arithmetic. However, it needs more digits to represent a number. Thus, it is easy to make errors when writing them.

  ➔ Use base 8 (*octal*), or base 16 (*hexadecimal*).

Binary (base 2)	Octal (base 8)	Decimal (base 10)	Hexadecimal (base 16)
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

Base 2, 4, 8, and 16 Conversions

- Example:

  $1011_2 = (10_2)\,(11_2) = (2_4)\,(3_4) = 23_4$

  $101010_2 = (101_2)\,(010_2) = (5_8)\,(2_8) = 52_8$

  $10110_2 = (010_2)\,(110_2) = (2_8)\,(6_8) = 26_8$

  $01101101_2 = (0110_2)\,(1101_2) = (6_{16})\,(D_{16}) = 6D_{16}$

  $10110110_2 = (1011_2)\,(0110_2) = (B_{16})\,(6_{16}) = B6_{16}$

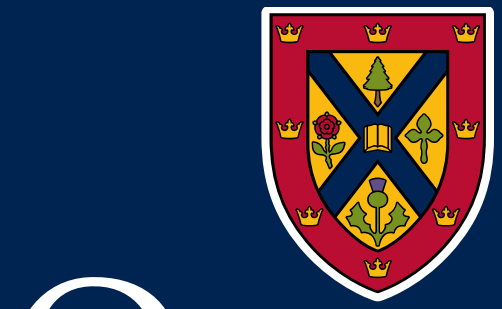
  $11001_2 = (0001_2)(1001_2) = (1_{16})\,(9_{16}) = 19_{16}$

  $231_4 = (2_4)\,(3_4)\,(1_4) = (10_2)\,(11_2)\,(01_2) = 101101_2$

  $537_8 = (5_8)\,(3_8)\,(7_8) = (101_2)\,(011_2)\,(111_2) = 101011111_2$

  $A6E_{16} = (A_{16})\,(6_{16})\,(E_{16}) = (1010_2)\,(0110_2)\,(1110_2) = 101001101110_2$

- For an unnatural conversion, convert first to base 10 as an intermediate step (using polynomial method), then convert from base 10 to the target base (using remainder and multiplication methods).



Queen's
UNIVERSITY