

ECE 2050 Digital Logic and Systems

# **Chapter 1 : Introductory Concepts**

Instructor: Yue ZHENG, Ph.D.



# Teaching Team

- **Instructor:**

- **Name:** Prof. Tinghuan Chen (陈廷欢), Ph.D.
- **Email:** [chentinghuan@cuhk.edu.cn](mailto:chentinghuan@cuhk.edu.cn)
- **Office:** TC413B
- **Office Hour:** Thur. 15:00-16:00 (TC413B)

L1 Session	Location
Mon. 9:00-10:30	Teaching Complex C203
Wed. 9:00-10:30	

- **Instructor:**

- **Name:** Prof. Yue Zheng (郑月), Ph.D.
- **Email:** [zhengyue@cuhk.edu.cn](mailto:zhengyue@cuhk.edu.cn)
- **Office:** CD414
- **Office Hour:** Just drop by or send email for appointment

L2 Session	Location
Mon. 10:30-12:00	Teaching Complex C203
Wed. 10:30-12:00	

# Teaching Assistants

<ul style="list-style-type: none"><li>• <b>Name:</b> Junguang Yao <b>Lead TA</b></li><li>• <b>Office:</b> TD205</li><li>• <b>Email:</b> junguangyao@link.cuhk.edu.cn</li><li>• <b>Office Hour:</b> Tue 14:00-15:00</li></ul>	<ul style="list-style-type: none"><li>• <b>Name:</b> Xin Wang</li><li>• <b>Office:</b> TD205</li><li>• <b>Email:</b> xinwang3@link.cuhk.edu.cn</li><li>• <b>Office Hour:</b> Wed 10: 00-11: 00</li></ul>
<ul style="list-style-type: none"><li>• <b>Name:</b> Kexin Chen</li><li>• <b>Office:</b> ZX307</li><li>• <b>Email:</b> kexinchen2@link.cuhk.edu.cn</li><li>• <b>Office Hour:</b> Tue 15:00-16:00</li></ul>	<ul style="list-style-type: none"><li>• <b>Name:</b> Yiyang Li</li><li>• <b>Office:</b> ZX107</li><li>• <b>Email:</b> 222010039@link.cuhk.edu.cn</li><li>• <b>Office Hour:</b> Tue 11: 00-12: 00</li></ul>
<ul style="list-style-type: none"><li>• <b>Name:</b> Wangqian Chen</li><li>• <b>Office:</b> ZR305</li><li>• <b>Email:</b> 221019052@link.cuhk.edu.cn</li><li>• <b>Office Hour:</b> TBD</li></ul>	<ul style="list-style-type: none"><li>• <b>Name:</b> Hanxu Zhang</li><li>• <b>Office:</b> ZX201</li><li>• <b>Email:</b> 120030021@link.cuhk.edu.cn</li><li>• <b>Office Hour:</b> Thu 20:00-21:00</li></ul>
<ul style="list-style-type: none"><li>• <b>Name:</b> Huaiyu Li</li><li>• <b>Office:</b> ZX301B</li><li>• <b>Email:</b> 223010059@link.cuhk.edu.cn</li><li>• <b>Office Hour:</b> Thu 14:00-15:00</li></ul>	<ul style="list-style-type: none"><li>• <b>Name:</b> Zhipeng Xu</li><li>• <b>Office:</b> TBD</li><li>• <b>Email:</b> 222010049@link.cuhk.edu.cn</li><li>• <b>Office Hour:</b> TBD</li></ul>



# Tutorials

Monday	18:00-19:00	Xin Wang	224010106	TD114
Monday	19:00-20:00	Junguang Yao	224010136	
Tuesday	18:00-19:00	Kexin Chen	221019028	
Tuesday	19:00-20:00	Yiyang Li	222010039	
Wednesday	18:00-19:00	Wangqian Chen	221019052	TC108
Wednesday	19:00-20:00	Hanxu Zhang	120030021	
Thursday	18:00-19:00	Huaiyu Li	223010059	
Thursday	19:00-20:00	Zhipeng Xu	222010049	

# Activity Highlights

## Company Visit in 2020



## Company Visit in 2023



## Invited seminars

**从处理器发展史的点滴，看电子工程与计算机的学习和实践**  
——一个人经历和推动行业发展的分享

时间：2022 4/24 周日 16:30-18:00

嘉宾：江海 北京大学博士  
海思半导体器件和设计可靠性技术专家

个人简介：...

**Exploration of CPU Microarchitecture with Instruction-level Parallelism**

Abstract: With the end of Dennard scaling and the slowdown of Moore's Law, modern computer architectures demonstrate a substantial extent of heterogeneity, ranging from general-purpose cores (CPUs) to domain-specific architectures (DSAs), aiming at a higher level of energy efficiency. However, the design techniques of heterogeneous engines are mostly inspired from the knowhows during exploration of the CPU architectures. Originating from the single instruction pipeline, the principle of data-level parallelism (DLP) evolves the simple RISC core into DSA engines such as GPU, GPGPU, and NPU. In contrast, instruction-level parallelism (ILP) advances the standard RISC core into high-performance CPUs such as Superscalar, VLIW, and Simultaneous multithreading (SMT). Between both principles, DLP offloads the task of extracting parallelism to software, while ILP mostly keeps such workloads within the scope of microarchitecture, leading to a series of vintage designs. In this talk, the evolution of the CPU following the principle of ILP is illustrated. Concepts such as pipelining, dynamic and static instruction scheduling, multi-issue, out-of-order execution, branch prediction, and multithreading are introduced. The seminar aims to enlighten undergraduate students with the fun of computer architecture.

2024.4.22  
2:00PM-5:00PM Location: Tx202

**SPEAKER**  
Prof. Zheng WANG  
Shenzhen Institute of Advanced Technology, CAS

About the Speaker: Dr. Zheng Wang received his B.Sc., M.Sc., and Ph.D. degrees from Shanghai Jiao Tong University (2007), Technical University of Munich (2009), and RWTH Aachen University (2015) respectively. He worked in Infineon AG (Munich) on virtual prototyping of ARM-based multi-processor System-on-Chip (2009). His doctoral dissertation focused on design automation of CPU, ASIC, and ASIC architectures using the Processor description language USA (2010-2015). He participated in the CHIST-ERA GEMSCAM project (2012-2015) as a digital designer for RISC and VLIW processors. Afterwards, he worked as a research fellow at NTU Singapore on designing ASICs for machine learning and hardware security (2015-2016). He was appointed as an assistant (2017) and associate professor (2020) at Shenzhen Institute of Advanced Technology (SIAT) of the Chinese Academy of Science (CAS). He is currently the executive director of the Center of Heterogeneous Intelligent Computer Architecture and Systems (HICAS) and the founder of UnityWare Co., Ltd., an SIAT-cofounded spinoff for tools, chips, and devices supporting edged AI. He has published 70+ research articles and received 2 NSFC (国家自然科学基金), 2 Guangdong S&T (广东省科技研发计划), 广东省电源联合重点基金 and 4 Huawei R&D funds as the principal investigator. His research interests lie in computer architecture and VLSI design techniques for heterogeneous tensor processing units.

SSE INDUSTRIAL TALKS 与行业专家面对面

**浅谈半导体器件的可靠性**

SSE Career

SSE Career

SSE INDUSTRIAL TALKS 与行业专家面对面

**FPGA 产业现状与EDA 软件关键技术发展趋势**

嘉宾：夏炜  
紫光同创软件研发中心研发经理

精彩看点

- 深入了解FPGA相关前沿技术最新进展
- 与FPGA布局布线算法专家面对面交流
- 紫光同创芯片与软件研发实习岗位推送

讲座摘要  
可编程系统平台芯片行业充满机遇与挑战，门槛之高在芯片行业里无出其右，此次讲座邀请到FPGA布局布线算法专家为您介绍FPGA产业格局、发展机遇和未来，分享可编程系统平台芯片产品编程体系架构、IP开发、硬核集成、EDA软件开发及算法等关键技术和未来趋势。同时，紫光同创热忱邀请每一位热爱技术的同学，关注并深入了解这个充满魅力的行业，公司研发实习岗位正虚位以待，期待与同学们共同书写FPGA技术的辉煌篇章。

嘉宾介绍  
夏炜，研究生毕业于武汉大学微电子学与固体电子学专业，现任紫光同创软件研发中心研发经理，具有十年以上软件开发、EDA核心算法开发经验，对EDA时序优化的布局布线核心算法具有深刻的理解，并在团队能力建设及重大项目攻关做出突出贡献。

2024 5/20 (周一) 15:00-16:30  
地点：Tx202

主持人：陈廷欢教授，香港中文大学(深圳)

香港中文大学(深圳) 理学院  
The Chinese University of Hong Kong, Shenzhen School of Science and Engineering



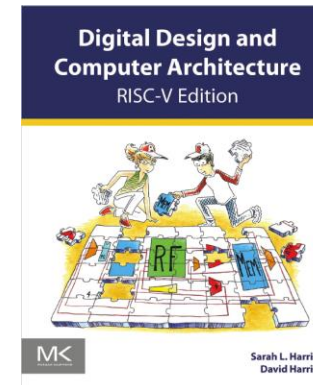
# Course Outline

- This course gives science and engineering students exposure to the basic concepts and techniques in digital logic and system design. Topics include digital system concepts, numbering systems and codes, Boolean algebra, logic gates and logic circuit elements, logic functions and simplification, logic circuits design, latches and flip-flops, counters, registers, memory and storage systems, FPGA, and introduction to CAD tools.

Assessment	% weight
Assignments	25%
Project	15%
Mid-term Test	25%
Final Exam	35%

# Textbook & References

- Digital Design and Computer Architecture RISC-V Edition ([main](#))
  - by Sarah L. Harris and David Harris



[Companion Resources - All \(zip\)](#)

[Lecture slides \(zip\)](#)

[Labs \(zip\)](#)

[Figures \(zip\)](#)

[Chapter 9: Embedded I/O Systems \(pdf\)](#)

[Appendix A: Digital System Implementation \(pdf\)](#)

[Appendix B: RISC-V ISA Summary \(pdf\)](#)

[Appendix C: C Programming \(pdf\)](#)

[HDL \(zip\)](#)

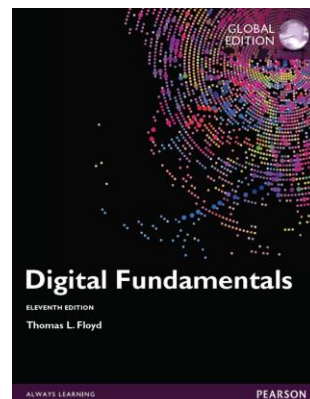
[Errata \(pdf\)](#)

[Solutions: Odd-numbered Exercises \(pdf\)](#)

[Videos](#)

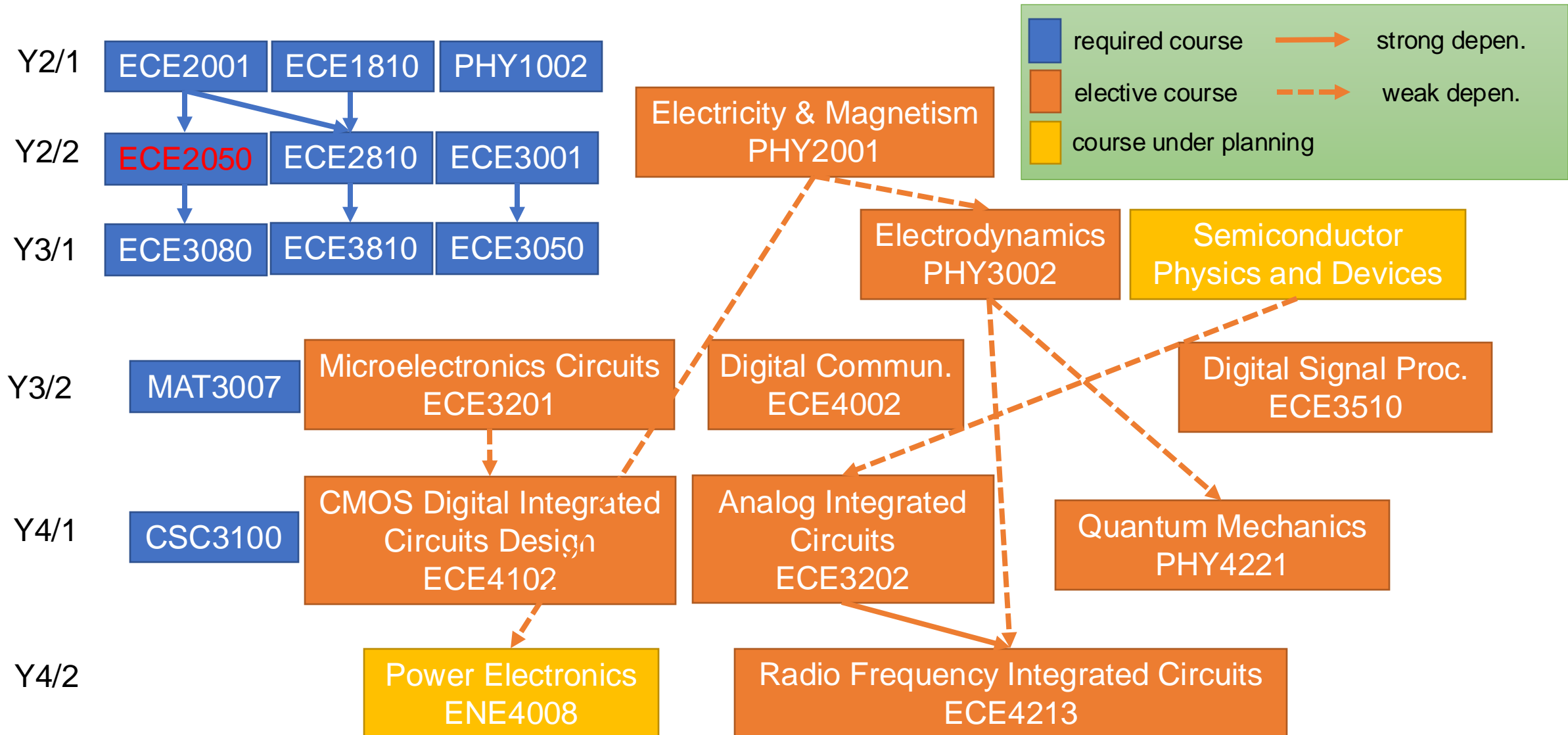
[Verilog Resources](#)

- Digital Fundamentals (11<sup>th</sup> Edition)
  - By Thomas Floyd



<https://pages.hmc.edu/harris/ddca/ddcarv.html>

# Course Map: Circuits





# Teaching Plan

Week		Content/ topic/ activity
1	1/6 - 1/12	<b>Ch1. Introductory concepts</b> Digital and analog quantities, Binary digits, Basic logic operations, System concept, Fixed-function integrated circuits
2	1/13 - 1/19	<b>Ch2. Number systems</b> Number systems and conversions, Binary arithmetic and operations, Signed numbers and operations, Binary coded decimal, Digital codes, Error detection codes
	1/20 - 2/9	Lunar New Year Holiday
3	2/10 - 2/16	<b>Ch3. Logic gates</b> The inverter, Logic gates (AND, OR, NAND, NOR, XOR, XNOR), logic levels
4	2/17 - 2/23	<b>Ch4. Boolean algebra and logic simplification</b> Boolean operations, Boolean algebra, De Morgan's theorem, Boolean analysis of logic circuits, Simplification using Boolean algebra, Standard forms, Truth tables, Karnaugh map and minimization
5	2/24 - 3/2	<b>Ch4. Boolean algebra and logic simplification &amp; Ch5. Combinational logic design</b> Basic combinational logic circuits and implementation, the universal property, Combinational logic with NAND and NOR gates
6	3/3 - 3/9	<b>Ch5. Combinational logic design &amp; Ch6. Combinational building blocks</b> Basic adders, Parallel adders, Comparators, Decoders, Encoders, Code converter

# Teaching Plan (Cont'd)

Week		Content/ topic/ activity
7	3/10 - 3/16	<b>Ch6. Combinational Building Blocks</b> Adders, Comparators, Encoder & Decoder, Multiplexer
8	3/17 - 3/23	Midterm & <b>Ch7. Sequential Logic Design</b> bistable circuits, SR Latch, D Latch, D Flip Flop, JK Flip Flop,
9	3/24 - 3/30	<b>Ch8. Shift registers</b> Serial/Parallel in Serial/Parallel out, bidirectional shift registers, applications
10	3/31 - 4/6	<b>Ch9. Finite State Machine (Qingming Festival)</b> Moore/Mealy FSM, timing, set up time, hold time
11	4/7 - 4/13	<b>Ch10. Counters</b> Asynchronous counters, Synchronous counters, Design, Applications
12	4/14 - 4/20	<b>Ch10. Counters</b> Asynchronous counters, Synchronous counters, Design, Applications
13	4/21 - 4/27	<b>Ch11. Memory &amp; Logic Arrays</b> Memory basics, RAM, ROM, Flash memory, PLAs & FPGAs

# Blackboard

香港中文大學(深圳)  
The Chinese University of Hong Kong, Shenzhen

Lecture Notes

EIE2050: Digital Logic and Systems\_L02L03

Lecture Notes

Tutorial Material

Homework

Discussions

Groups

Tools

Help

Lecture Notes

Build Content Assessments Tools Partner Content

Course Outline

Lecture notes

Submit your homework here

All homework assignments are submitted online via blackboard

Why do we need to learn  
digital logic & systems?



# Some Reasons

- Major Required
- An insight into the specialty
- Circuit design
- Computer architecture
- Built upon Boolean algebra, theoretical CS
- High-performance software design requires you to understand the hardware layer
- Microprocessors, datasheet, schematic diagram

Quora

As a CS student, why do I need to learn digital circuit design? Isn't it more relevant to EE?

All related (36) ▾

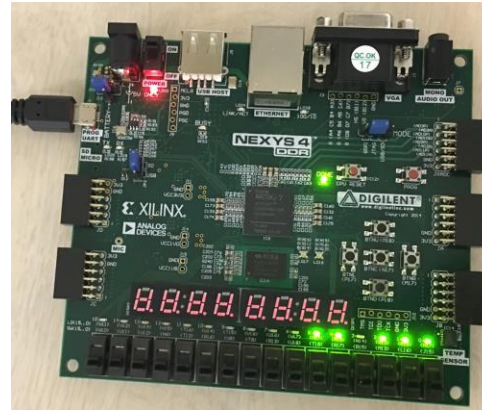
Sort Recommended ▾

# Microprocessors

- CPU
- GPU
- DSP
- MCU
- SoC
- ASIC
- FPGA
- ...



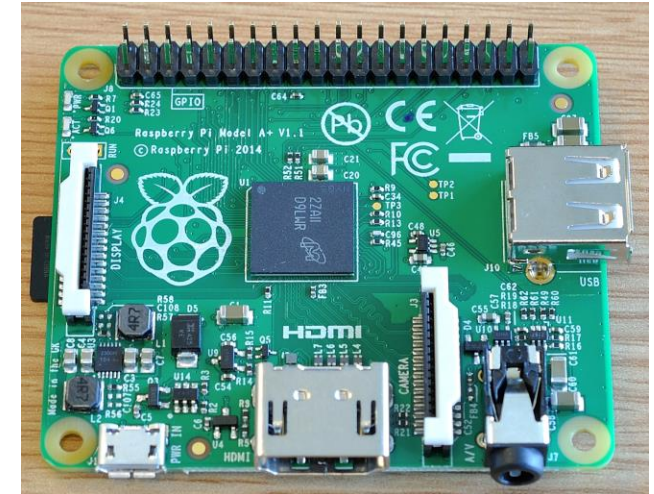
CPU



FPGA



GPU



MCU  
(microcontroller)



# How to design a Microprocessor?



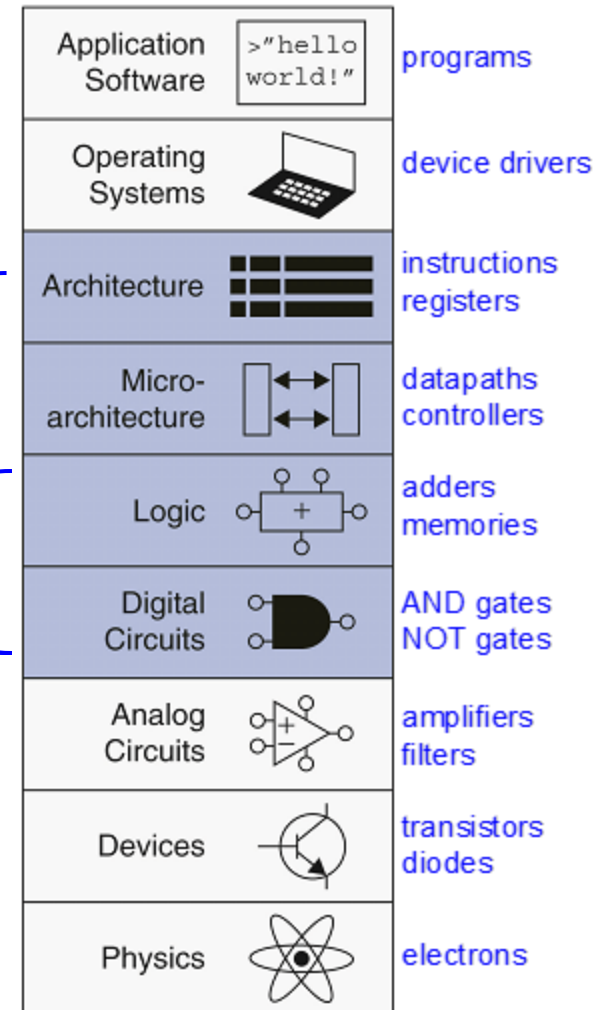
# Managing Complexity

- Modern digital systems:
  - Millions or billions of transistors
- How do we design things that are too big to fit in one person's head at once?
  - Abstraction
  - Discipline
  - The 3-Y's
    - Hierarchy<sub>y</sub>
    - Modularity<sub>y</sub>
    - Regularity<sub>y</sub>

# Abstraction

- Hiding details when they aren't important

Focus of this course

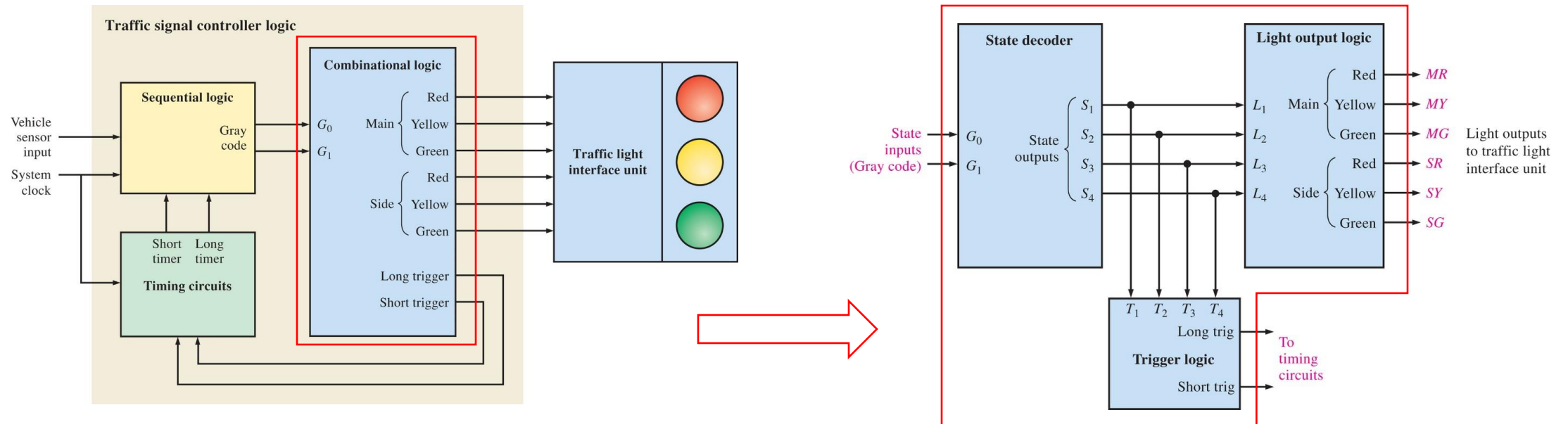


# Discipline

- Intentionally restrict design choices
- Example: Digital discipline
  - Discrete voltages instead of continuous
  - Simpler to design than analog circuits – can build more sophisticated systems
  - Digital systems replacing analog predecessors:  
i.e., digital cameras, digital television, cell phones, CDs

# The 3-y's

- Hierarchy
  - A system divided into modules and submodules
- Modularity
  - Having well-defined functions and interfaces
- Regularity
  - Encouraging uniformity, so modules can be easily reused



# Example: Model T Ford

- Famous early (1908) example of interchangeable parts.
  - Most previous cars were hand-crafted by skilled tradesmen.
  - Mass production on moving assembly lines greatly reduced cost.



[en.wikipedia.org/wiki/Ford\\_Model\\_T#/media/File:1925\\_Ford\\_Model\\_T\\_touring.jpg](https://en.wikipedia.org/wiki/Ford_Model_T#/media/File:1925_Ford_Model_T_touring.jpg)

- Henry Ford:

*I will build a motor car for the great multitude. It will be large enough for the family, but small enough for the individual to run and care for. It will be constructed of the best materials, by the best men to be hired, after the simplest designs that modern engineering can devise. But it will be so low in price that no man making a good salary will be unable to own one – and enjoy with his family the blessing of hours of pleasure in God's great open spaces.*



# Example: Model T Ford

- Hierarchy
  - Car has chassis, wheels, seats, engine.
  - Engine has cylinders, spark plugs, exhaust, carburetor.
  - Carburetor has air intake, inlet needle, feed pipe, coupling nut.

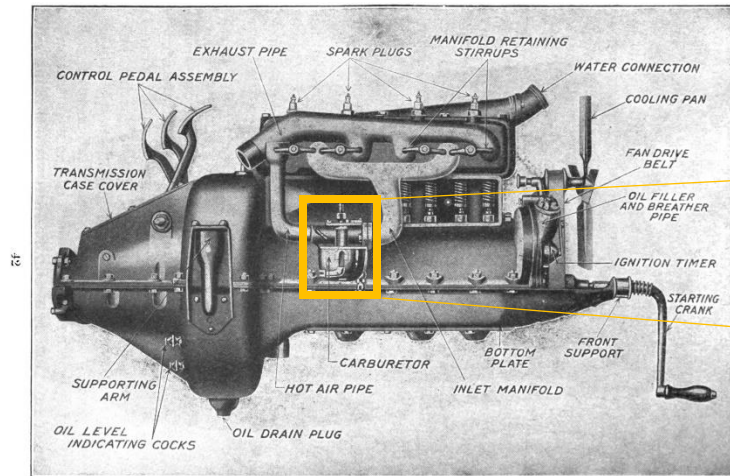


Fig. 8.—Valve Side of the Ford Model T Unit Power Plant Showing Manifolds, Carburetor and Interior of One of the Valve Spring Chambers.

[https://en.wikipedia.org/wiki/Ford\\_Model\\_T\\_engine#/media/File:Pagé\\_1917\\_Model\\_T\\_Ford\\_Car\\_Figure\\_08.png](https://en.wikipedia.org/wiki/Ford_Model_T_engine#/media/File:Pagé_1917_Model_T_Ford_Car_Figure_08.png)

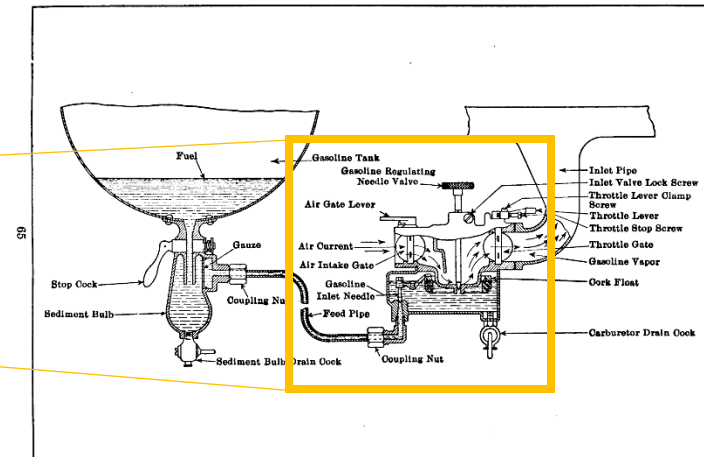


Fig. 14.—The Ford Model T Fuel Supply and Gas Making System.

[https://en.wikipedia.org/wiki/Ford\\_Model\\_T\\_engine#/media/File:Pagé\\_1917\\_Model\\_T\\_Ford\\_Car\\_Figure\\_14.png](https://en.wikipedia.org/wiki/Ford_Model_T_engine#/media/File:Pagé_1917_Model_T_Ford_Car_Figure_14.png)

# Example: Model T Ford

- Modularity
  - Function of coupling nut:
    - Hold fuel line to intake elbow
    - Prevent leaks
    - Easily removable
  - Interface of coupling nut:
    - Standardized diameter, thread pitch, torque
- Regularity
  - Interchangeable parts
    - Standard nut could be purchased from many suppliers
  - “Any customer can have a car painted any color that he wants so long as it is black.” - Henry Ford

# The Digital Abstraction

- Most physical variables are **continuous**

- Voltage on a wire
- Frequency of an oscillation
- Position of a mass

**Analog** quantity :  
*continuous* values



Vinyl records  
12 inches (~30cm), 3.5 hours

- Digital abstraction considers **discrete subset** of values

**Digital** quantity :  
*a discrete set* of values

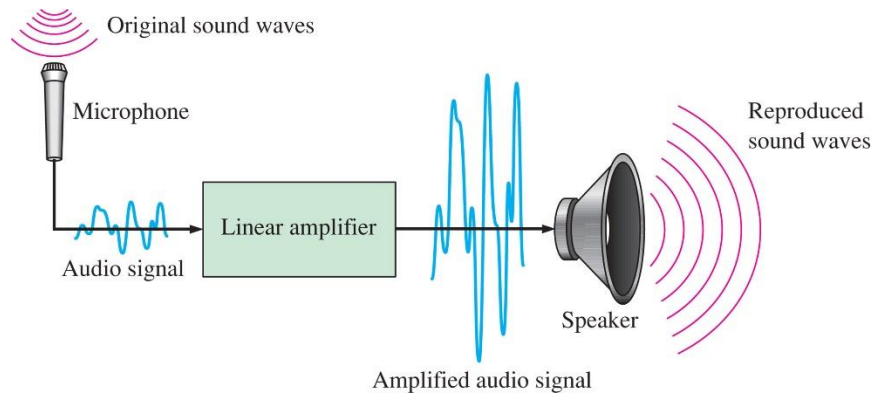


Compact Disk (CD)  
12cm, 74 minutes

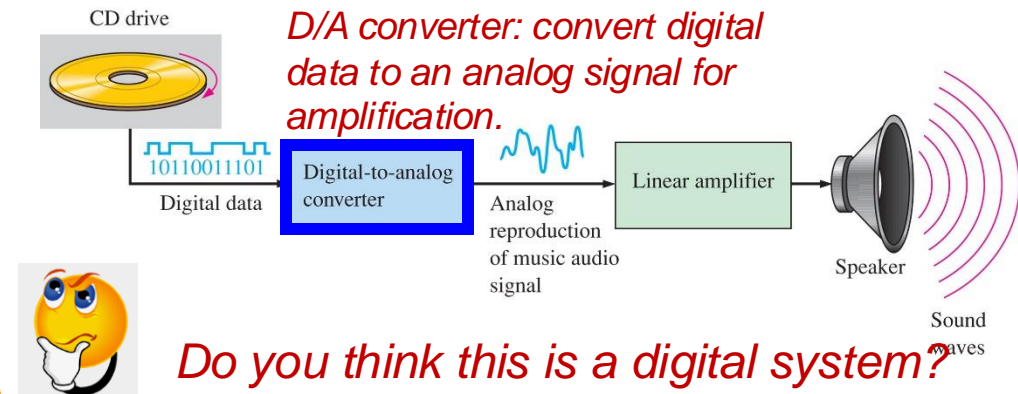
# The Digital Advantages

- Digital data can be stored, processed and transmitted more efficiently and reliably
- Digital data is more noise resilient

A typical analog system



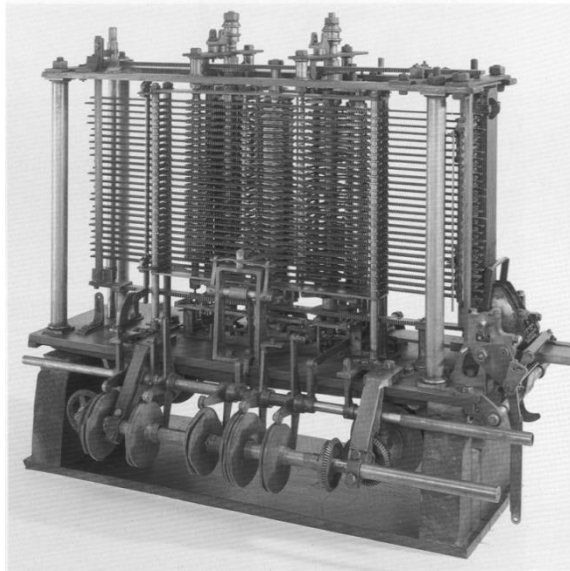
An audio system with a CD player



- Can you think of any disadvantages of digital systems?

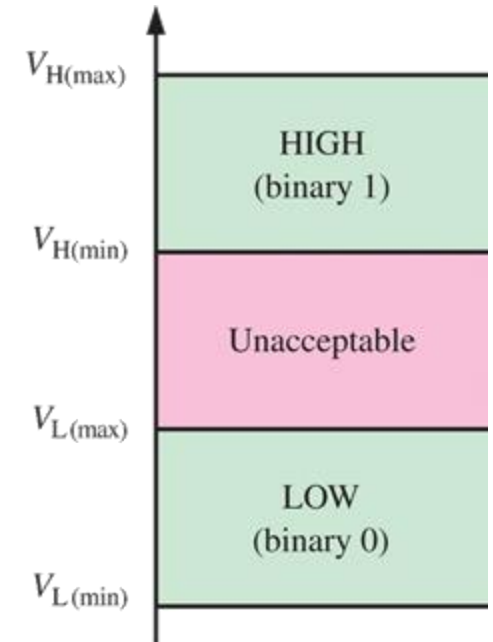
# The Analytical Engine

- Designed by Charles Babbage from 1834 – 1871
- Considered to be the first digital computer
- Built from mechanical gears, where each gear represented a discrete value (0-9)
- Babbage died before finishing it



# Digital Discipline: Binary Values

- **Two discrete values:**
  - 1's and 0's
  - 1, TRUE, HIGH
  - 0, FALSE, LOW
- **1 and 0:** voltage levels, rotating gears, fluid levels, etc.
- Digital circuits use voltage levels
  - 0: low voltage (GND)
  - 1: high voltage (VDD)
- **Bit:** Binary digit





# Bit

- Binary digit: a single number is either 0 or 1
- Positive Logic:
  - 1 is represented by the **higher voltage level** → **HIGH**,
  - 0 is represented by the **lower voltage level** → **LOW**
- Negative logic : 1 → Low, 0 → High

# Number Systems

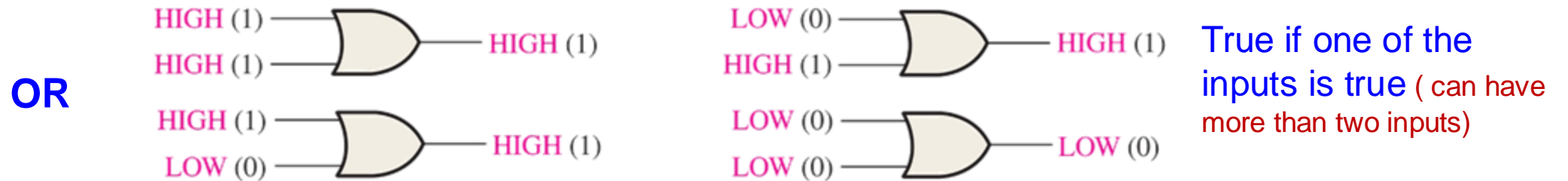
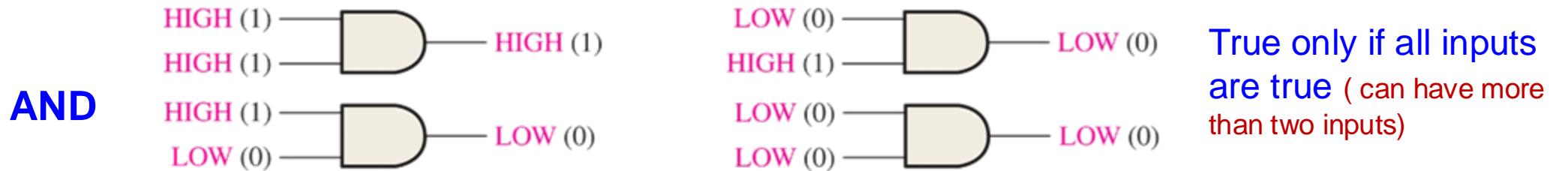
Decimal	Binary	Octal	Hex
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	7
8	1000	10	8
9	1001	11	9

Decimal	Binary	Octal	Hex
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F



# Logic Gates

- Logic **gate** : A **circuit** that performs a specified logic **function**



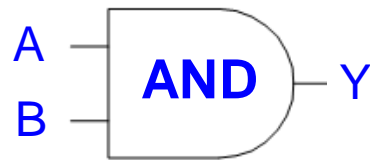
# Combinational Circuits & Sequential Circuits

- **Combinational Logic**

- Memoryless
- Outputs determined by current values of inputs

- **Sequential Logic**

- Has memory
- Outputs determined by previous and current values of inputs



Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

Boolean Equation

$$Y = AB$$

# Boolean Algebra and Logic Simplification

## • Boolean Algebra

### Laws

- Commutative laws
- Associative laws
- Distributive law

1.  $A + 0 = A$

2.  $A + 1 = 1$

3.  $A \cdot 0 = 0$

4.  $A \cdot 1 = A$

5.  $A + A = A$

6.  $A + \bar{A} = 1$

7.  $A \cdot A = A$

8.  $A \cdot \bar{A} = 0$

9.  $\bar{\bar{A}} = A$

10.  $A + AB = A$

11.  $A + \bar{A}B = A + B$

12.  $(A + B)(A + C) = A + BC$

### Rules

### Demorgan's Theorems

$$\overline{A \cdot B} = \bar{A} + \bar{B}$$

$$\overline{A + B} = \bar{A} \cdot \bar{B}$$

$$\begin{aligned} Y &= \bar{A}B\bar{C} + \bar{A}BC + ABC \\ &= \bar{A}B\bar{C} + \bar{A}BC + \bar{A}BC + ABC \\ &= \bar{A}B(\bar{C} + C) + (\bar{A} + A)BC \\ &= \bar{A}B + BC \end{aligned}$$

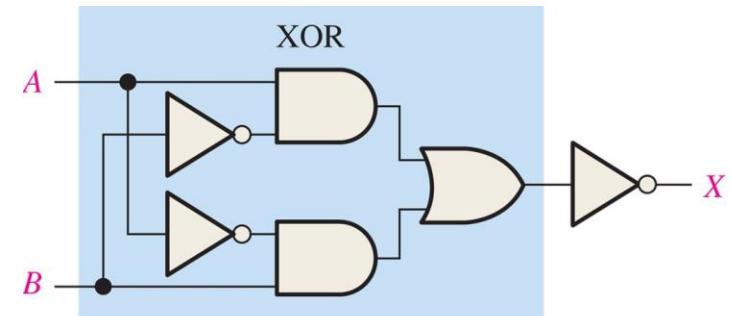
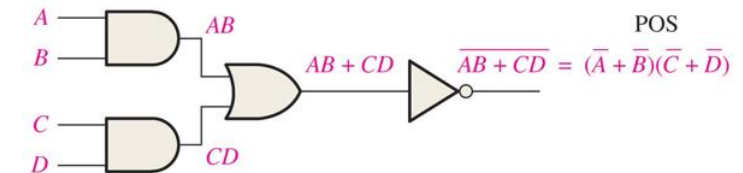
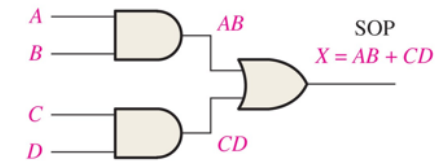
Karnaugh  
Map

Y \ C \ AB				
	00	01	11	10
0	0	1	0	0
1	0	1	1	0

$$Y = \bar{A}B + BC$$

# Combinational Logic

- Basic combinational logic circuits
  - AND-OR Logic
  - AND-OR-Invert Logic
  - Exclusive-OR Logic
  - Exclusive-NOR Logic
- Implementing combinational logic
  - From Boolean equation to logic circuit
  - From truth table to logic circuit

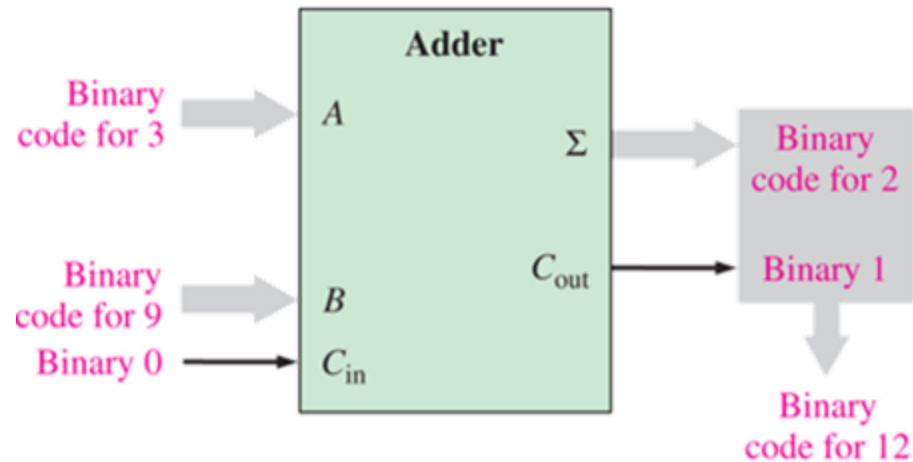




# Combinational Building Blocks

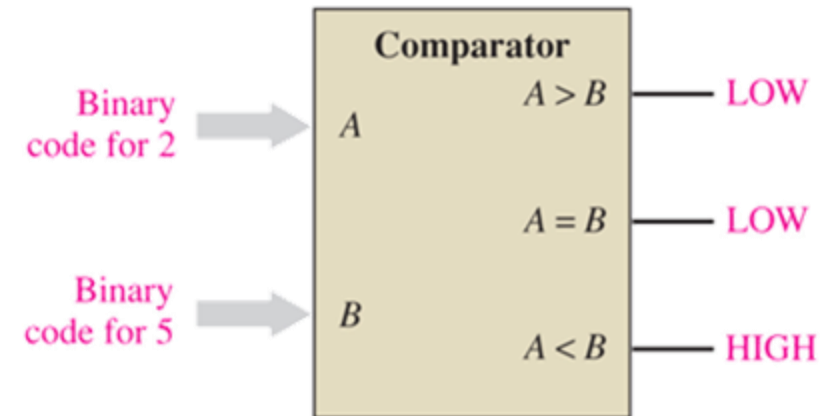
- Adders

E.g. A plus B ( $3+9 = 12$ )



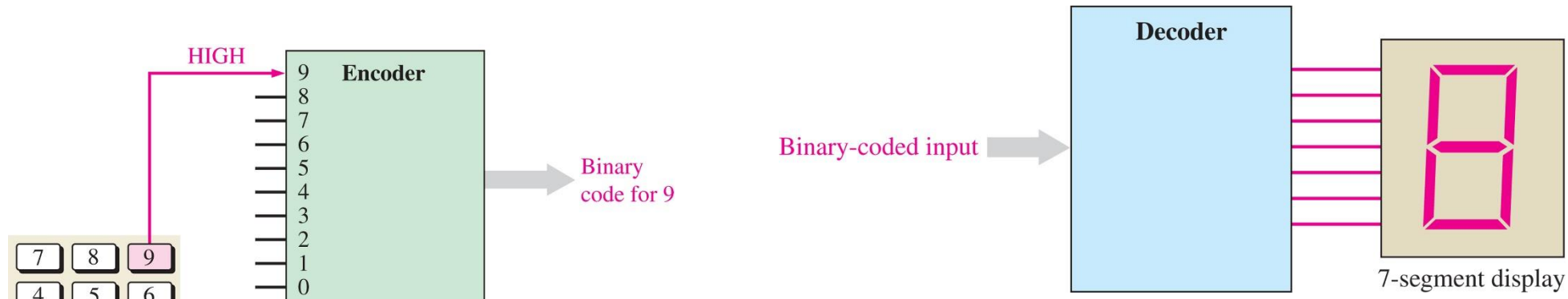
- Comparators

E.g. A is less than B ( $2 < 5$ ) as indicated by the High output ( $A < B$ )

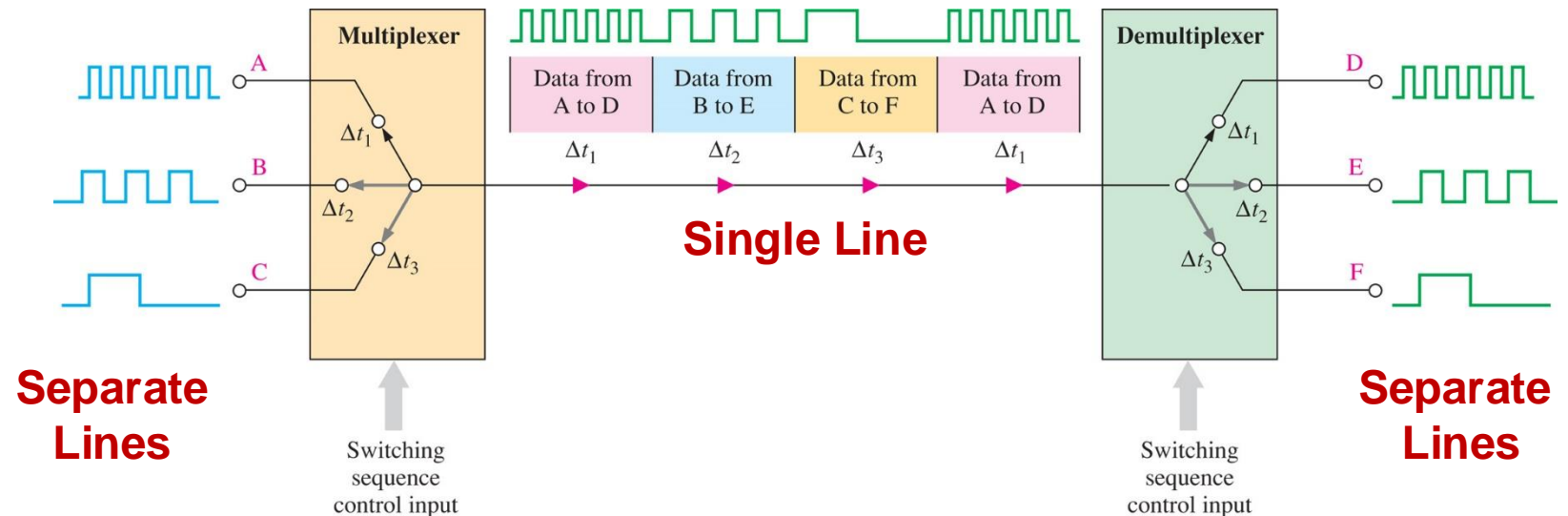


# Combinational Building Blocks

- Encoders & Decoders



- Multiplexer  
& Demultiplexer

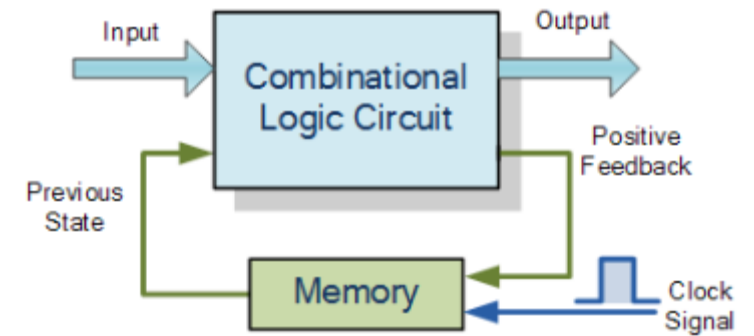


# Sequential Logic Design

- Outputs of sequential logic depend on current *and* prior input values – it has **memory**.
- State: all the information about a circuit necessary to explain its future behavior

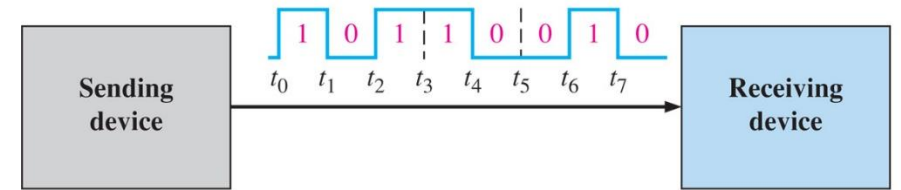
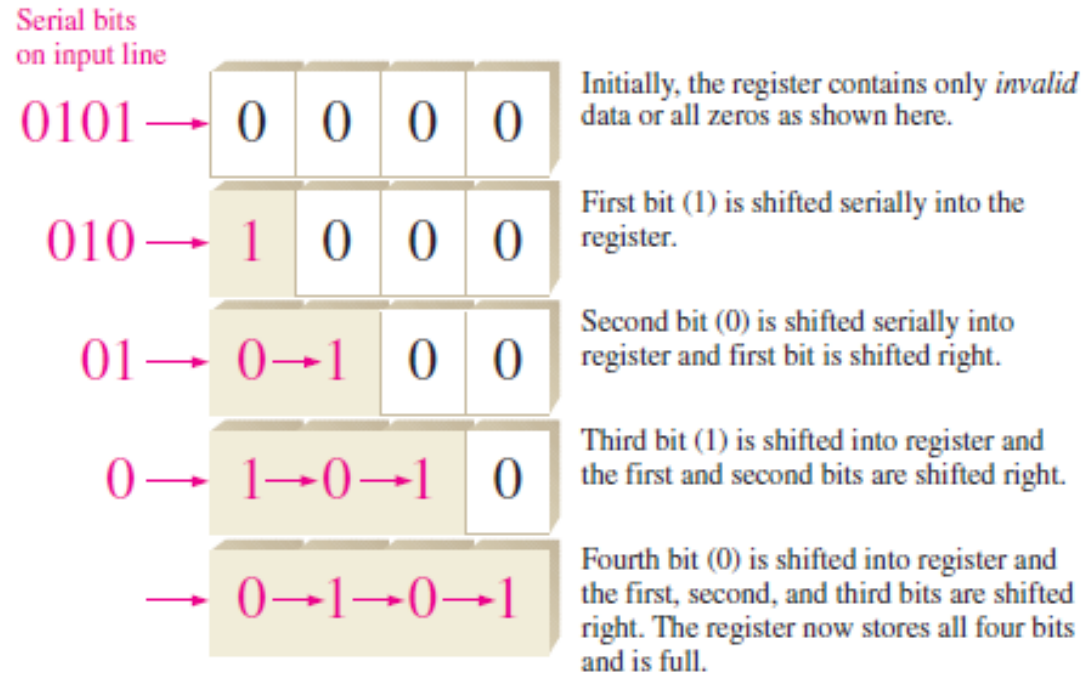


- Latches and Flip-flops:
  - Bistable circuit: 2 stable states
  - store only one bit at a time, either a 0 or a 1
  - SR Latch, D Latch
  - D Flip-flops, JK Flip-flops

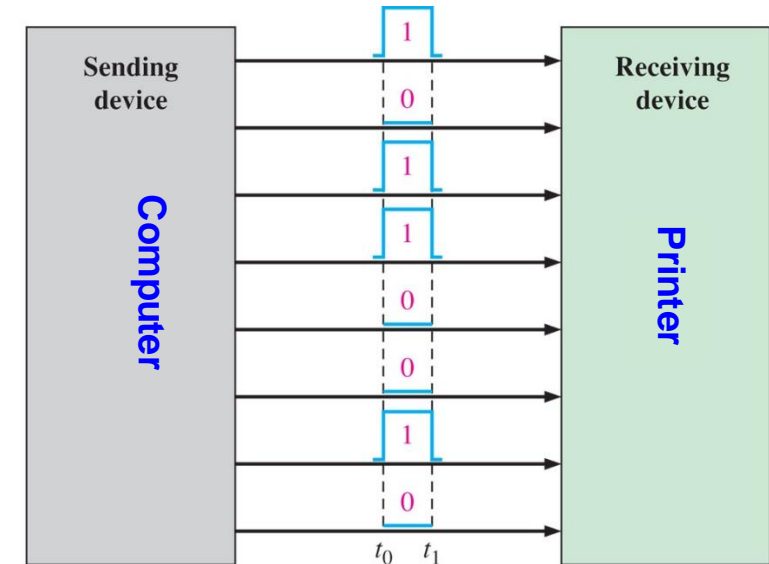


# Shift Registers

- Shift register: Several Flip-Flops
  - Data Storage
  - Data Movement



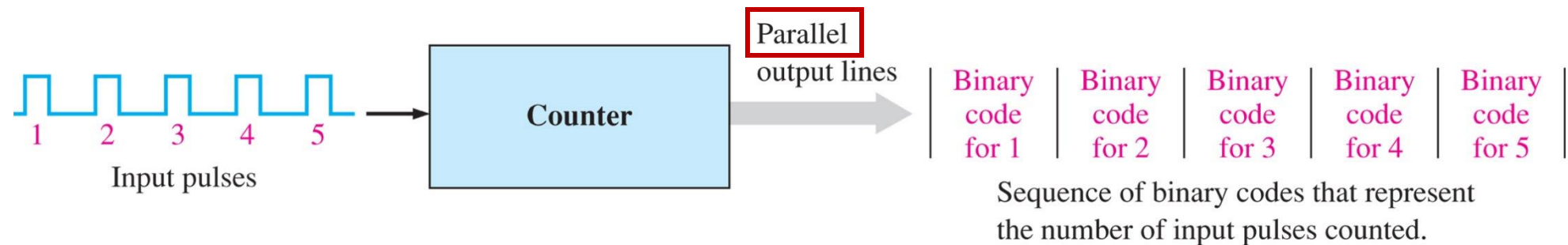
**Serial:** One bit at a time over a single line  
Example: 8 time intervals for 8 bits



**Parallel :** multiple bits at the same time over separate lines  
Example: 1 time interval for 8 bits

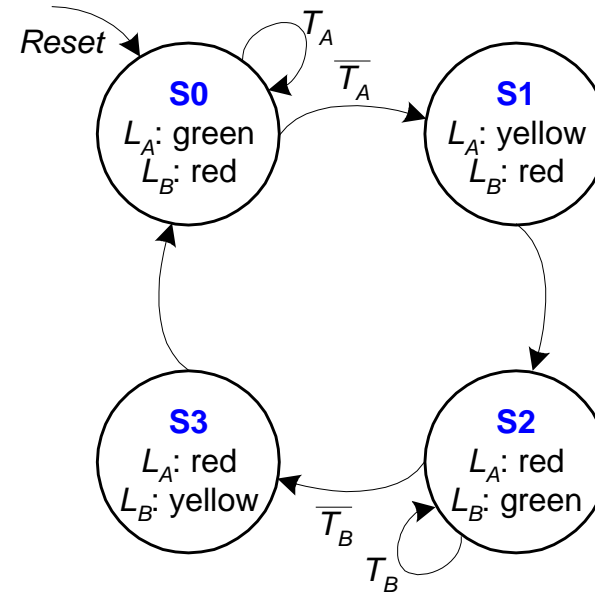
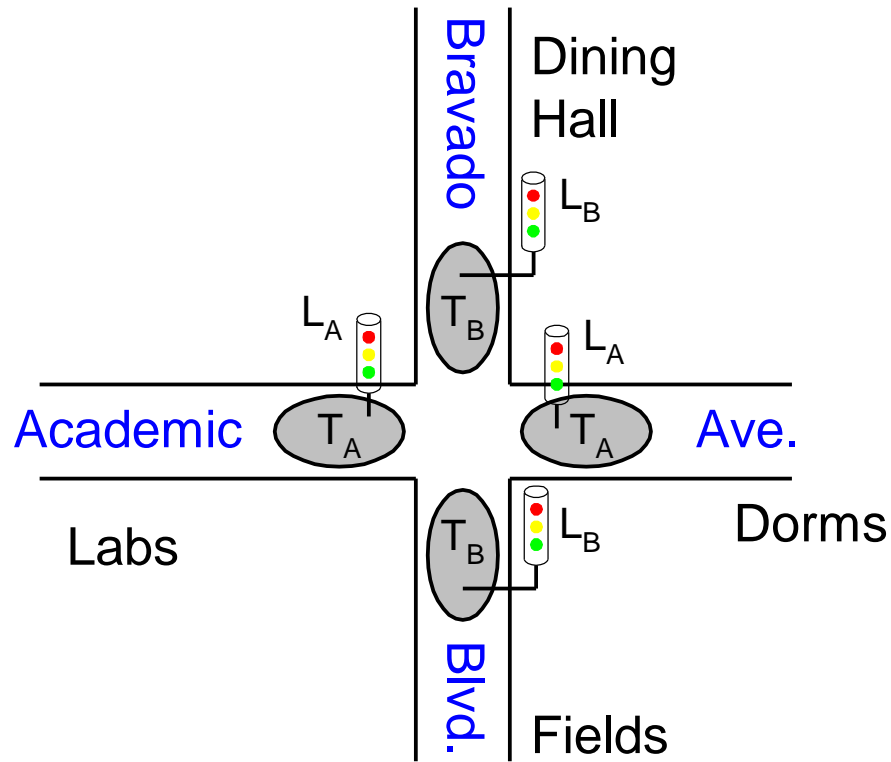
# Counters

- To count, the counter must “**remember**” the present number so that it can go to the next proper number **in sequence**.
- **Storage** capability is an important characteristic of all counters, and **flip-flops** are generally used to implement them.



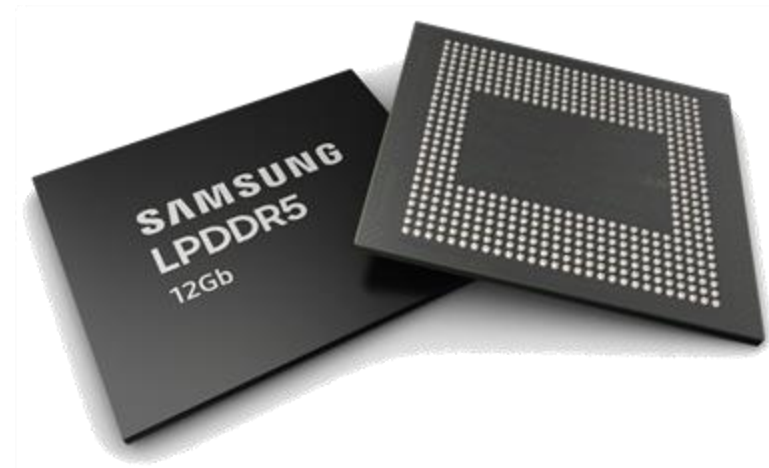
# Finite State Machine

- State Register + Combinational Logic
- States and transitions
- Example: Traffic light controller system



# Memory

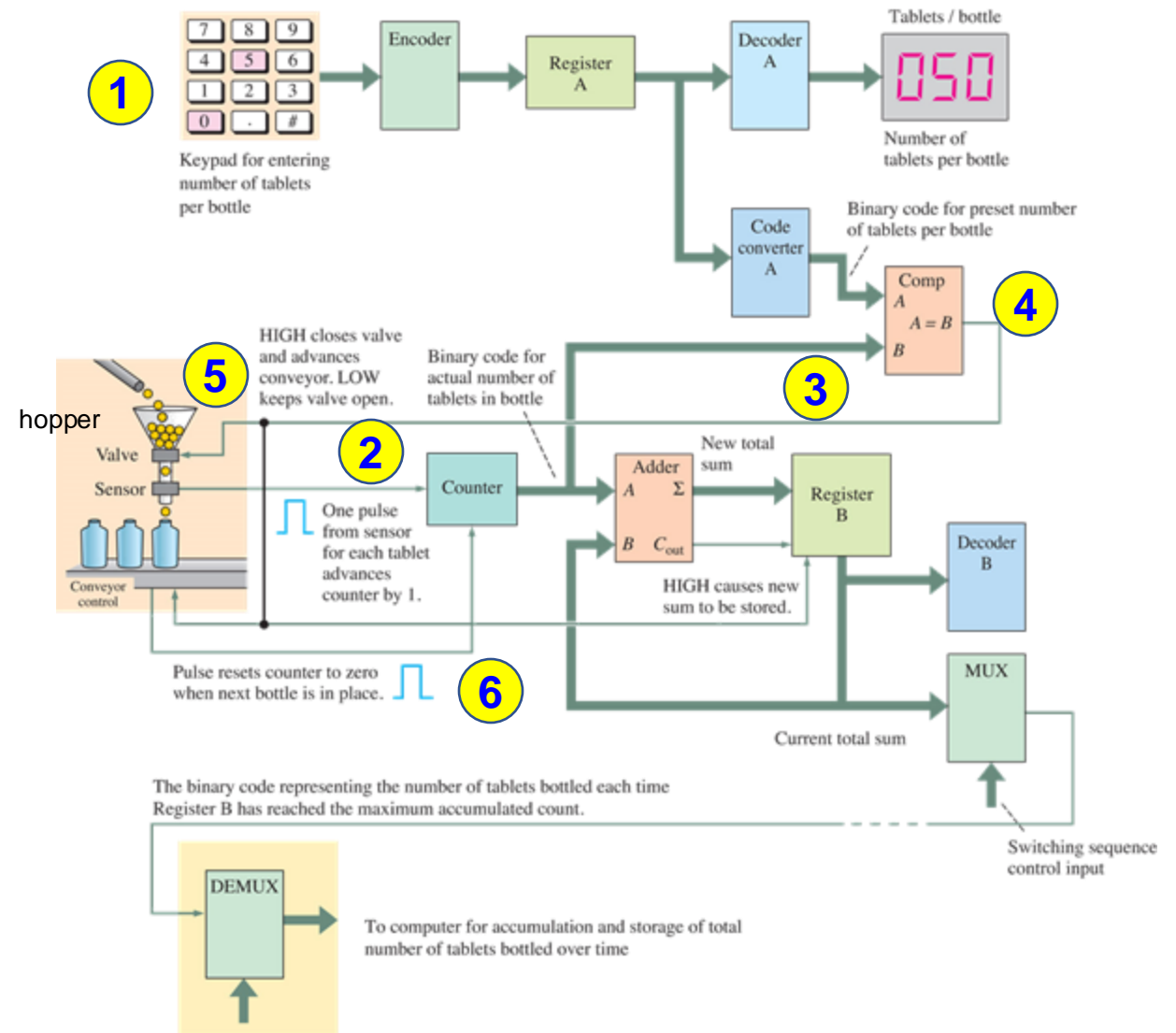
- SRAM, DRAM
- ROM
- Flash





# Example: A Process Control System

1. The maximum # of tablets per bottle is entered from the keypad;
2. An optical sensor detects each passing tablet & produces a pulse going to the **Counter** → Counter advanced by one;
3. The binary count is transferred from the counter to the B input of the **comparator**;
4. If  $A=B$ , **comparator** output goes HIGH, implying the bottle is full;
5. The HIGH comparator output closes the hopper valve & stop the flow of tablets while activating the conveyor to move the next empty bottle into place
6. When the bottle is in place, a pulse is issued to reset the counter to zero → the comparator output goes back LOW & the hopper valve restarts the flow of tablets.

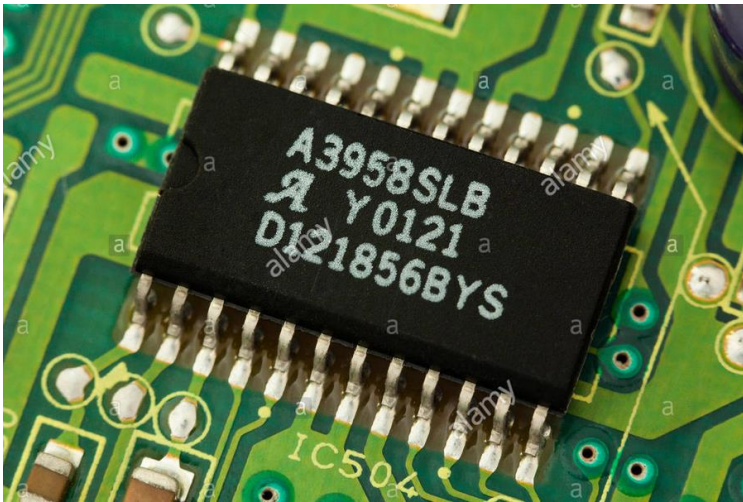




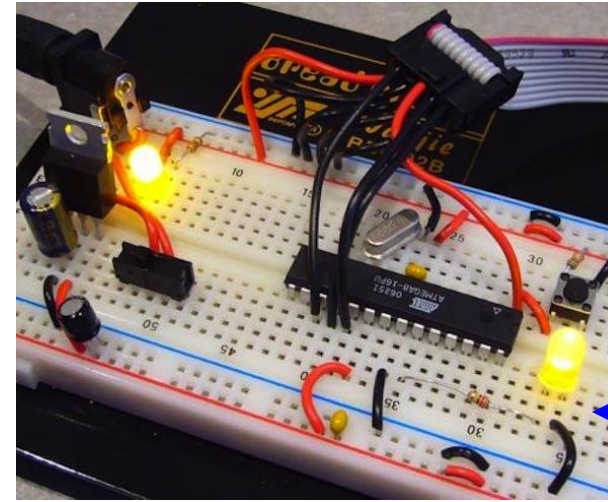
# Integrated Circuits (IC)

- An electronic circuit entirely constructed on a single small chip of silicon
- Programmable and Fixed-function Logic
- IC packages :

Surface-mounted



Through-hole mounted

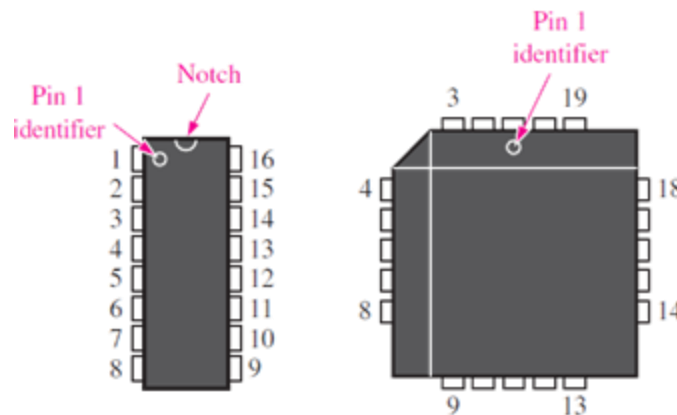


面包板  
or  
免焊万用电路板  
Solderless  
breadboard

# Complexity of Fixed-Function ICs

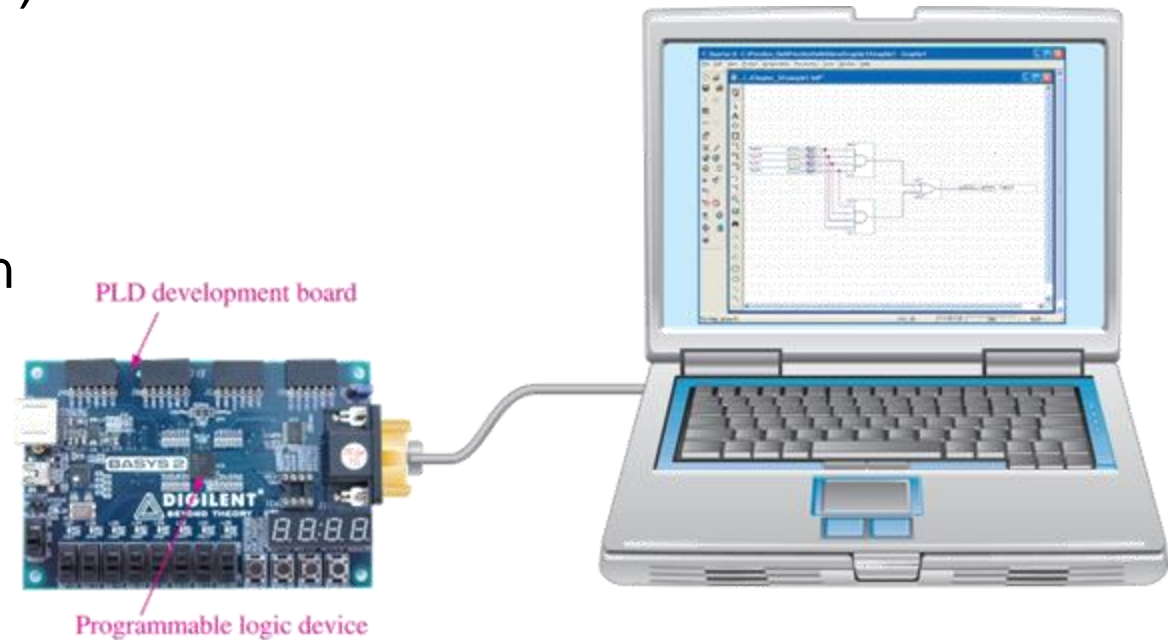
	# of gate circuits on a single chip	Typical Examples
Small-scale integration (SSI)	<10	Basic gates and flip-flop
Medium-scale integration (MSI)	10 to 100	Encoders, Decoders, Counters, Registers, Multiplexers, Arithmetic circuits, Small memories
Very large-scale integration (VLSI)	10,000 to 100,000	Memories
Ultra large-scale integration (ULSI)	>100,000	Very large memories, Larger micro-processors, Larger single-chip computers

## Pin Numbering



# Programmable Logic

- Programmable Logic Devices (PLDs)
  - SPLDs (simple PLDs)
  - CPLDs (complex PLDs)
- Field-Programmable Gate Array (FPGA )
- Programing Process
  - Software development package installed on a computer
    - Graphic entry of a logic circuit
    - Text entry such as VHDL
  - A development board
  - A cable



# Chapter Review

- ❑ Analog versus Digital
- ❑ Bits (Binary digits), Logic Levels and Digital Waveforms
- ❑ Basic logic functions: NOT, AND and OR
- ❑ Combinational & sequential logic functions:  
comparator, adder, encoder/decoder, (de)multiplexer,  
flip-flops, registers, counter
- ❑ Integrated circuit (IC): Programmable versus Fixed-function
  - ◆ Package: Surface-mounted and Through-hole
  - ◆ Programmable: PLD (SPLD and CPLD) and FPGA
  - ◆ Fixed-function : SSI/MSI/VLSI/ULSI

# True/False Quiz



An analog quantity is one having continuous values.



A digital quantity has no discrete values.



There are two digits in the binary system.



The term bit is short for binary digit.



In positive logic, a LOW level represents a binary 1.



An AND function is implemented by a logic circuit known as an inverter.



A flip-flop is a bistable logic circuit that can store only two bits at a time.



Two broad types of digital integrated circuits are fixed-function and programmable.