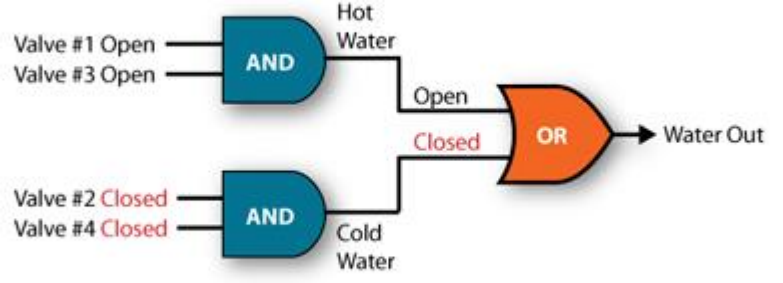
Logic and Computer Design Fundamentals

Introduction



Ming Cai

cm@zju.edu.cn

College of Computer Science and Technology
Zhejiang University

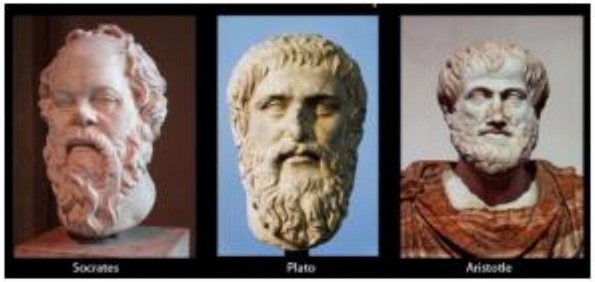
Course Information

- **Text book**
  - Mano and Kime 《Logic and Computer Design Fundamental》 5th Edition
- **QQ group: 868200982 (rename to "ID + name")**

  password: logic2023

TA1:	**Zishu Wei**	**3210102498@zju.edu.cn**
TA2:	**Jiefeng Wu**	**jaymaple0223@gmail.com**
TA3:	**Tianhao Liang**	**2662248501@qq.com**

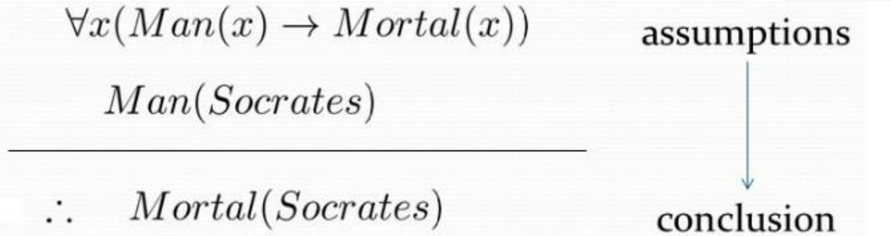
Logic, Boolean Logic and Logic Circuit



ARISTOTLE

the ways of thinking, cognition...

Contribution: Propositional Logic and Syllogism



$$\forall x(Man(x) \rightarrow Mortal(x))$$

$$Man(Socrates)$$

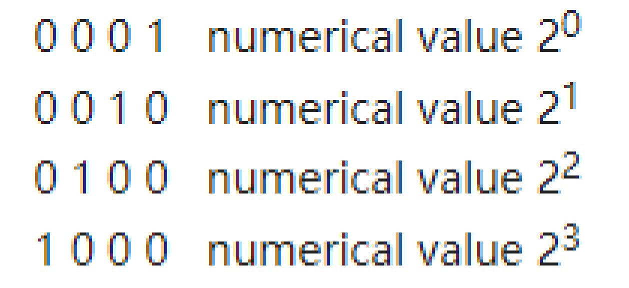
$$\therefore \quad Mortal(Socrates)$$

assumptions

conclusion

Leibniz

Contribution: Leibniz studied binary numbering in 1679 and his work appeared in his article in 1703.

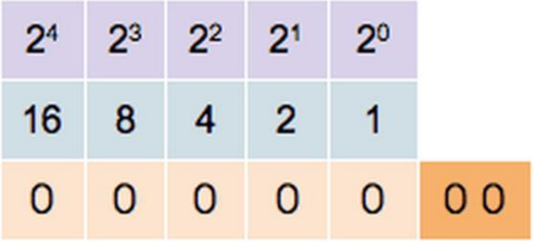


0 0 0 1 numerical value 2^0

0 0 1 0 numerical value 2^1

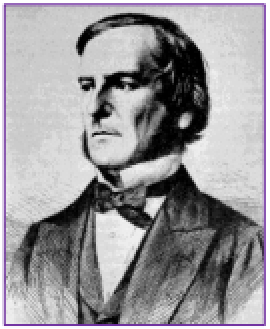
0 1 0 0 numerical value 2^2

1 0 0 0 numerical value 2^3



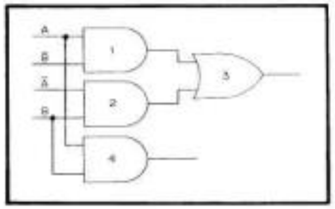
2^4	2^3	2^2	2^1	2^0	
16	8	4	2	1	
0	0	0	0	0	0 0

Logic, Boolean Logic and Logic Circuit



George Boole: An Investigation of the Laws of Thought (1854)

Claude Shannon: 1937 master's thesis,
A Symbolic Analysis of Relay and Switching Circuits

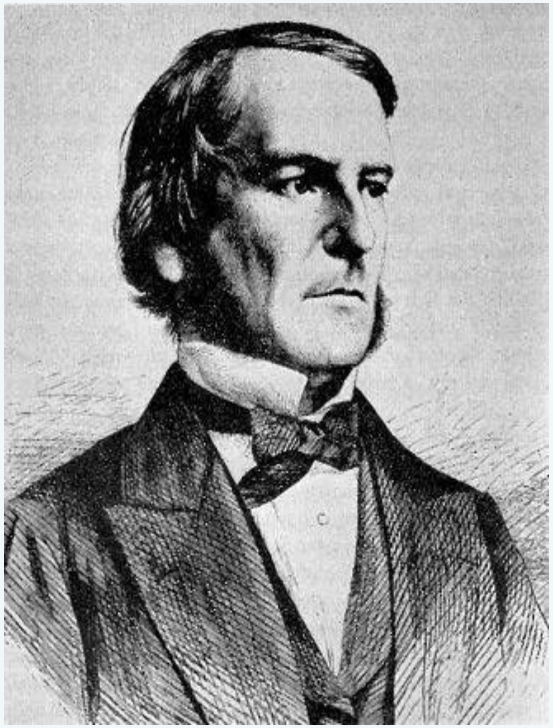


Gödel (1931)
Incompleteness theorem

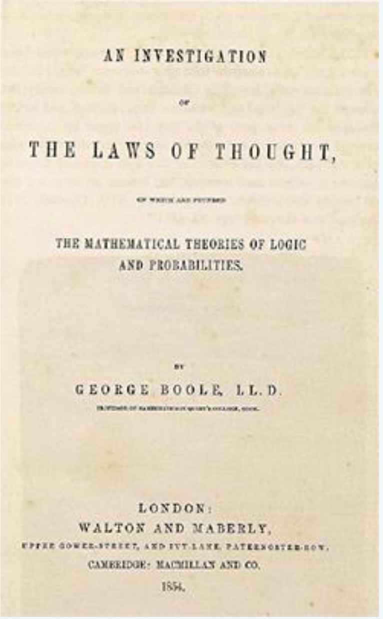
Turing (1936)
Turing machine

von Neumann
von Neumann architecture

From George Boole to….



George Boole
1815-1864



AN INVESTIGATION

OF

THE LAWS OF THOUGHT,

ON WHICH ARE FOUNDED

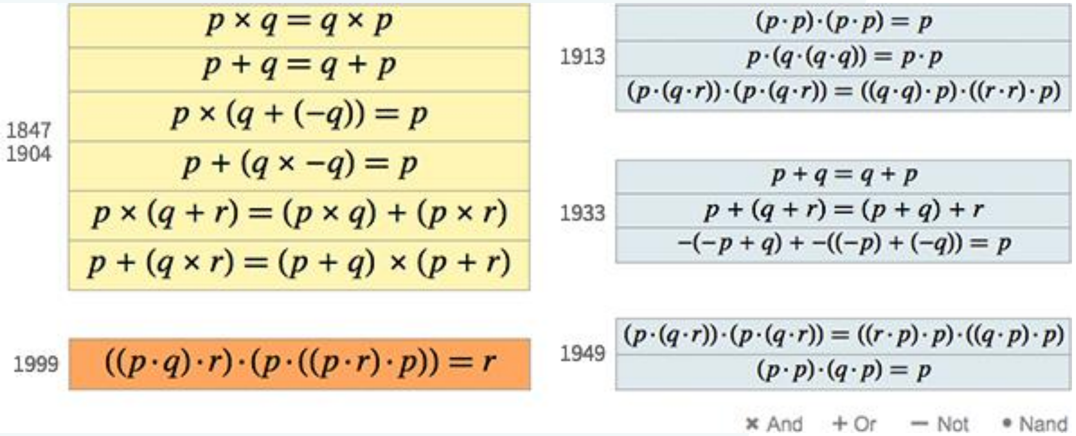
THE MATHEMATICAL THEORIES OF LOGIC
AND PROBABILITIES.

BY

GEORGE BOOLE, LL.D.

PROFESSOR OF MATHEMATICS IN QUEEN'S COLLEGE, CORK.

LONDON:
WALTON AND MABERLY,
UPPER GOWER-STREET, AND IVY-LANE, PATERNOSTER-ROW.
CAMBRIDGE: MACMILLAN AND CO.
1854.

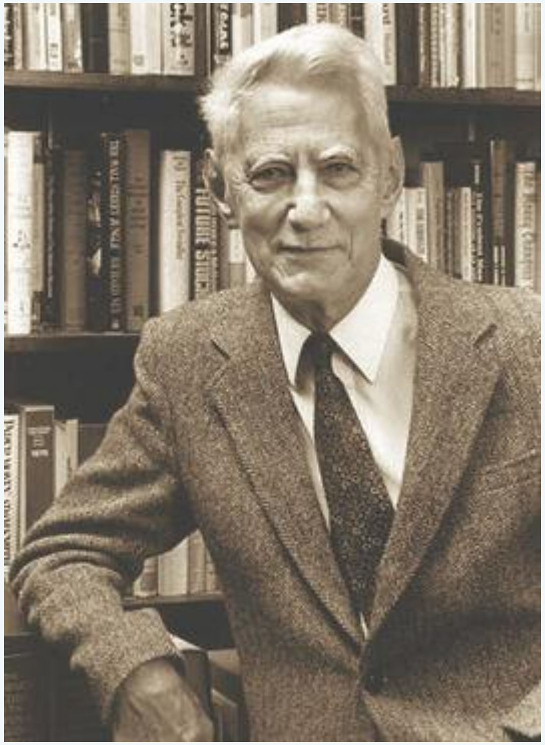


1847 1904	$p \times q = q \times p$
	$p + q = q + p$
	$p \times (q + (-q)) = p$
	$p + (q \times -q) = p$
	$p \times (q + r) = (p \times q) + (p \times r)$
	$p + (q \times r) = (p + q) \times (p + r)$
1999	$((p \cdot q) \cdot r) \cdot (p \cdot ((p \cdot r) \cdot p)) = r$

1913	$(p \cdot p) \cdot (p \cdot p) = p$
	$p \cdot (q \cdot (q \cdot q)) = p \cdot p$
	$(p \cdot (q \cdot r)) \cdot (p \cdot (q \cdot r)) = ((q \cdot q) \cdot p) \cdot ((r \cdot r) \cdot p)$
1933	$p + q = q + p$
	$p + (q + r) = (p + q) + r$
	$-(-p + q) + -((-p) + (-q)) = p$
1949	$(p \cdot (q \cdot r)) \cdot (p \cdot (q \cdot r)) = ((r \cdot p) \cdot p) \cdot ((q \cdot p) \cdot p)$
	$(p \cdot p) \cdot (q \cdot p) = p$

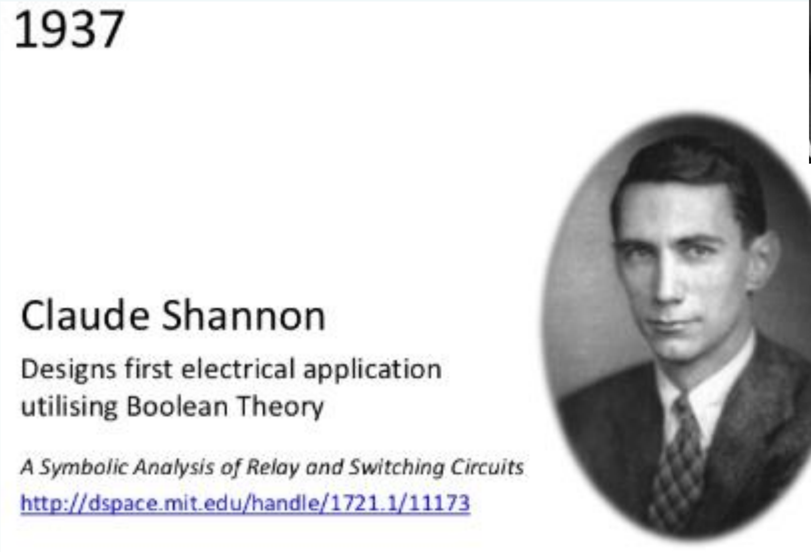
× And + Or − Not • Nand

From George Boole to….



Claude Elwood Shannon
1916–2001

the father of information theory



1937

Claude Shannon

Designs first electrical application utilising Boolean Theory

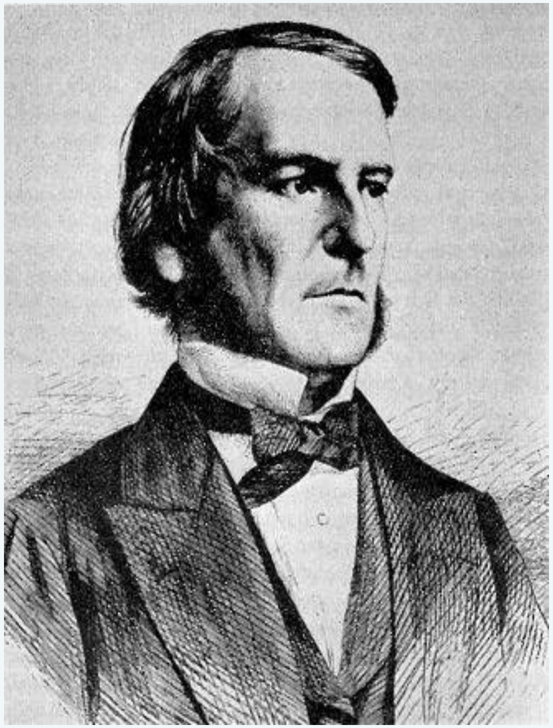
A Symbolic Analysis of Relay and Switching Circuits

http://dspace.mit.edu/handle/1721.1/11173

“ possibly the most important, and also the most famous, master’s thesis of the century”

——Howard Gardner

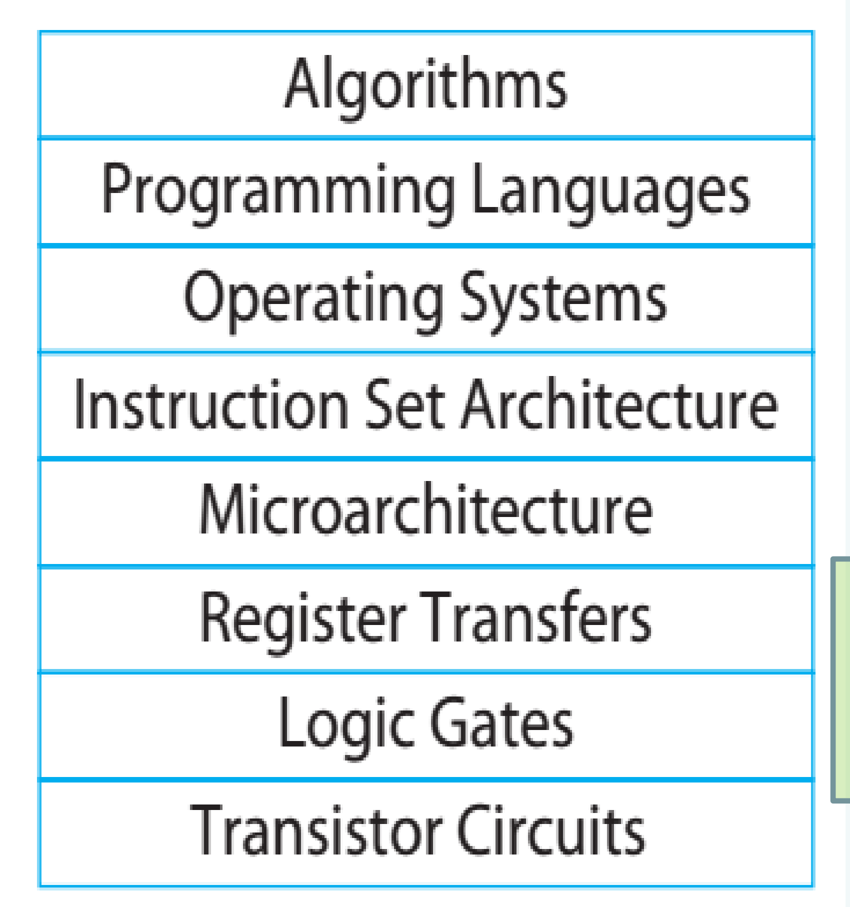
From George Boole to….



George Boole
1815-1864

Geoffrey Hinton
1947-
2019 Turing Prize

Abstraction Layers in Computer Systems



Algorithms

Programming Languages

Operating Systems

Instruction Set Architecture

Microarchitecture

Register Transfers

Logic Gates

Transistor Circuits

- □ **greedy, heuristic, LP, DP**
- □ **C/C++, Java, Python**
- □ **Linux, Windows, Android**
- □ **X86, ARM, MIPS, PPC**
- □ **Pipeline, OOE, Multiprocessing**
- □ **Register, Datapath, Control Unit**
- □ **AND, OR, NOT, NAND, NOR**
- □ **BJT, JFET, IGFET**

Topics Covered

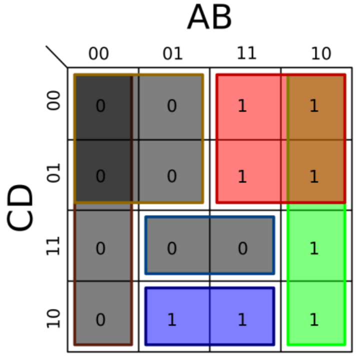
- Topics
  - Number representation, digital codes
  - Boolean algebra and logic minimization techniques
  - Combinational circuit design and analysis
  - Sequential circuit design and analysis, timing analysis of sequential circuits
  - Logic design with structural and behavioral modeling (Verilog = Verification+Logic)
  - Programmable logic devices (PLD) and memories

You are supposed to acquire basic skills and knowledge to analysis and design logic circuit

Big Picture of Logic Design

Boolean algebra and minimization

$$
\begin{aligned}
F&=\overline{AB+\bar{C}}+A\bar{C}+B\\
&=(\bar{A}+\bar{B})\,C+A\bar{C}+B\\
&=\bar{A}C+\bar{B}C+A\bar{C}+B\\
&=B+C+\bar{A}C+A\bar{C}\\
&=B+C+A+\bar{A}C\\
&=A+B+C
\end{aligned}
$$

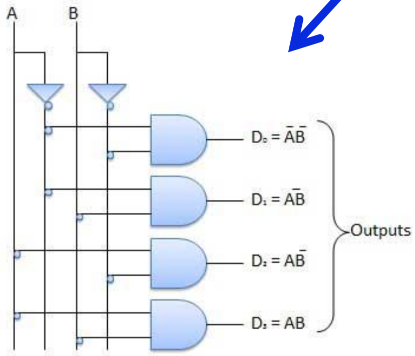


CD \ AB	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	0	0	0	1
10	0	1	1	1

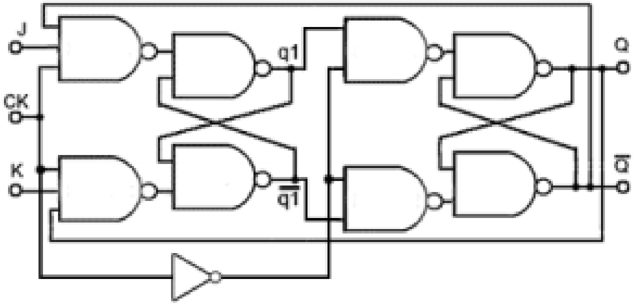
f(A,B,C,D) = E(6,8,9,10,11,12,13,14)

F=AC'+AB'+BCD'

F=(A+B)(A+C)(B'+C'+D')

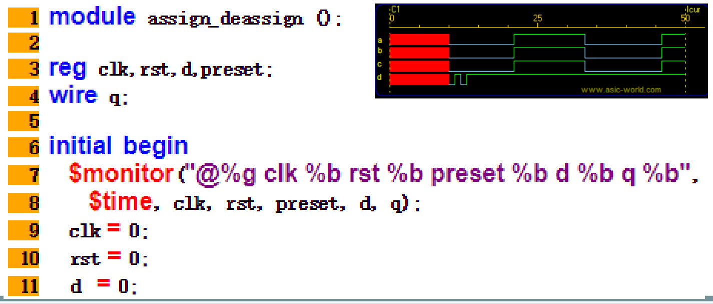


Combinational circuit



Sequential circuit

Logic design using Verilog



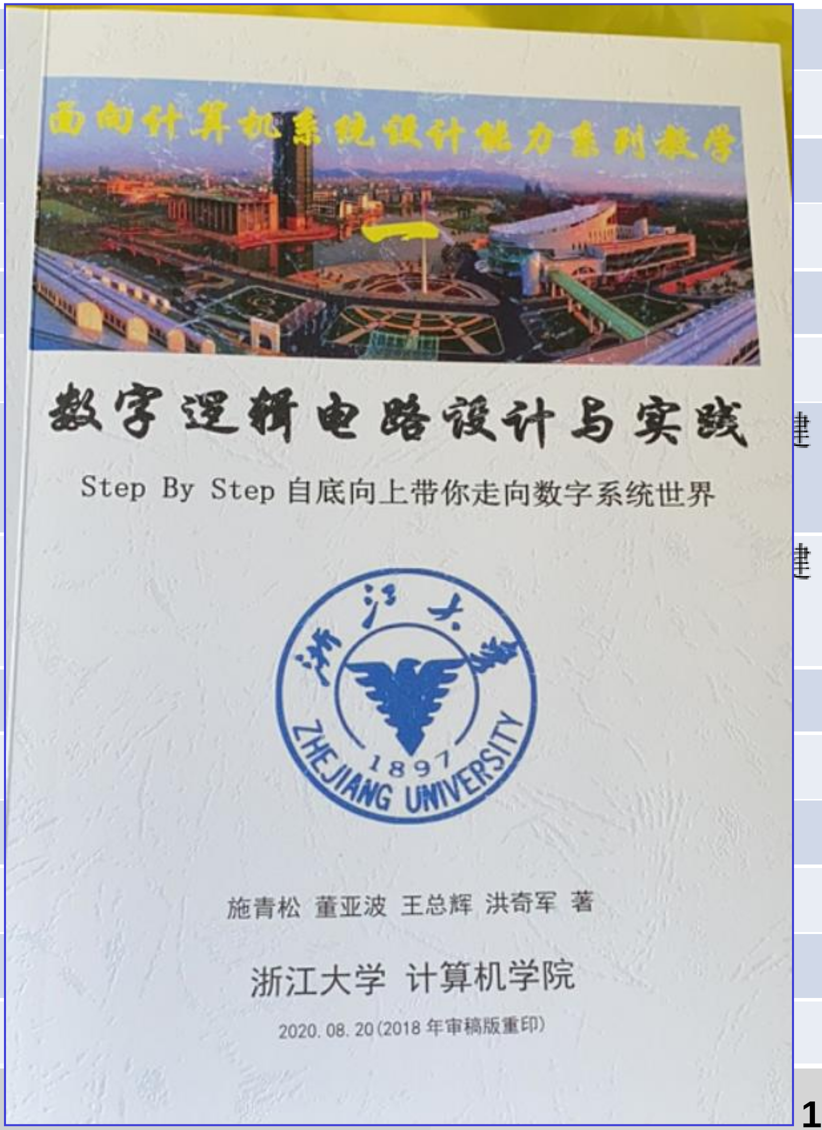
```
module assign_deassign ();

reg clk,rst,d,preset;
wire q;

initial begin
  $monitor("@%g clk %b rst %b preset %b d %b q %b",
    $time, clk, rst, preset, d, q);
  clk = 0;
  rst = 0;
  d  = 0;
```

Experiment

编号	实验名称	实验方式
1	常用数字仪器的使用	
2	二极管与三极管开关电路	
3	集成逻辑门电路的功能及参数测试	
4	EDA实验平台与实验环境运用	
5	变量译码器设计与应用	
6	7段数码管显示译码器设计与应用	
7	多路选择器设计及应用	建
8	全加器的设计实现	建
9	加减法器和ALU基本原理与设计	
10	锁存器与触发器基本原理	
11	同步时序电路设计	
12	寄存器堆及寄存器传输设计	
13	计数器/定时器设计与应用	
14	移位寄存器设计与应用	



面向计算机系统设计能力系列教学

数字逻辑电路设计与实践

Step By Step 自底向上带你走向数字系统世界

施青松 董亚波 王总辉 洪奇军 著

浙江大学 计算机学院

2020.08.20(2018年审稿版重印)

Course Assessment

- **Experiments: 30%**
- **Project: 15%**
  - Source code, source project and technical report should be submitted
  - Deadline: before the final examination

Practice is a vital process of the class

- **Quiz: 25%**
  - Random, without notification, 4 – 5 times
  - Topics from textbook and home assignments
- **Final: 30%**
  - Score of Final Examination: ≥50

Course Requirements

- **Home assignments:**
  - Need not submit
  - Answer sheet will be published

- **Getting familiar with technical terms**
  - The final examination will be written in English

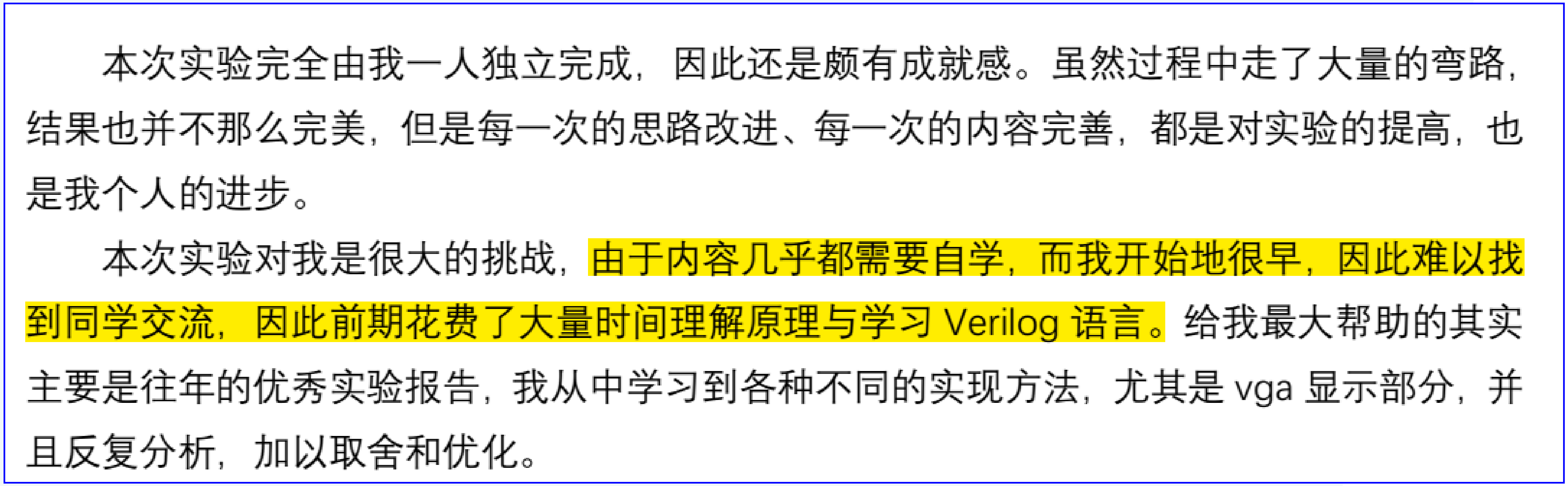
Course Resources

- **Software & Online practice**
  - 学生APP端参与随堂测试：https://v.youku.com/v_show/id_XMzc4Mzc3NTcyMA==.html
  - HDLBits-Verilog Practice: https://hdlbits.01xz.net/wiki/Problem_sets#Getting_Started
  - Logisim: https://sourceforge.net/projects/circuit/?source=typ_redirect
  - Logisim tutorial: https://www.bilibili.com/video/av842668792/
- **References**
  - 夏宇闻. Verilog数字系统设计教程. 北京：北京航空航天大学出版社
  - Verilog 教程：https://www.runoob.com/w3cnote/verilog-tutorial.html
  - FPGA Verilog开发实战指南：https://doc.embedfire.com/fpga/altera/ep4ce10_pro/zh/latest/index.html
  - course notes from a senior：https://note.isshikih.top/cour_note/D2QD_DigitalDesign/
  - FPGA真的很难学吗？：https://bbs.elecfans.com/jishu_1666264_1_1.html
  - FPGA projects and tutorials: https://www.fpga4fun.com
  - FPGA projects and tutorials: https://opencores.org

Pass Your Exam with Flying Colors

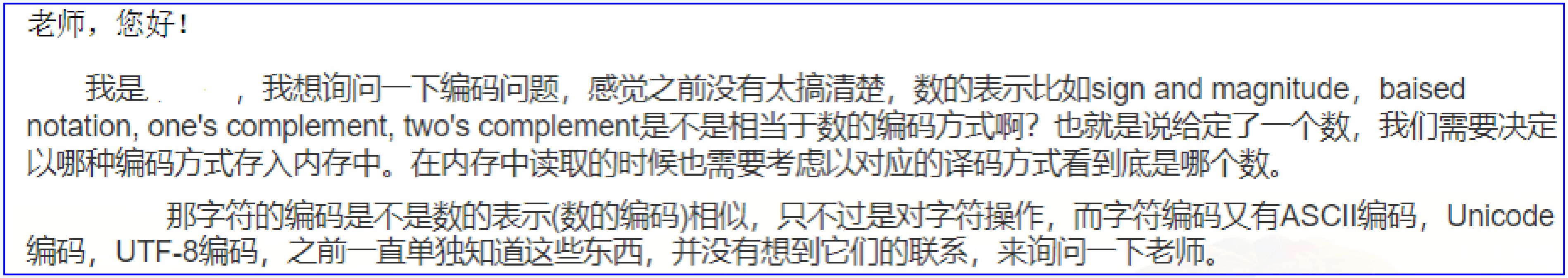
- **Three tips (IPA):**
  - Taking Initiative can help improve your performance.
  - Perseverance is essential for achieving your goal.
  - Do not be afraid to Ask questions or Ask for help.

Taking Initiative



本次实验完全由我一人独立完成，因此还是颇有成就感。虽然过程中走了大量的弯路，结果也并不那么完美，但是每一次的思路改进、每一次的内容完善，都是对实验的提高，也是我个人的进步。

本次实验对我是很大的挑战，由于内容几乎都需要自学，而我开始地很早，因此难以找到同学交流，因此前期花费了大量时间理解原理与学习 Verilog 语言。给我最大帮助的其实主要是往年的优秀实验报告，我从中学习到各种不同的实现方法，尤其是 vga 显示部分，并且反复分析，加以取舍和优化。

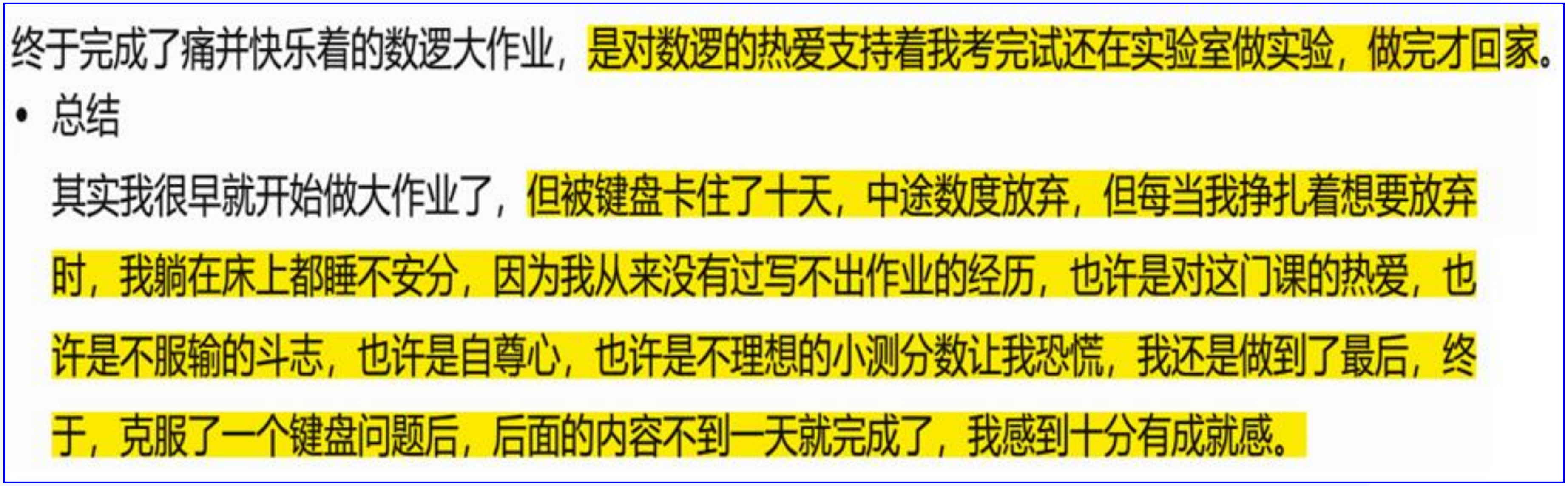


老师，您好！

我是 ，我想询问一下编码问题，感觉之前没有太搞清楚，数的表示比如sign and magnitude，baised notation, one's complement, two's complement是不是相当于数的编码方式啊？也就是说给定了一个数，我们需要决定以哪种编码方式存入内存中。在内存中读取的时候也需要考虑以对应的译码方式看到底是哪个数。

那字符的编码是不是数的表示(数的编码)相似，只不过是对字符操作，而字符编码又有ASCII编码，Unicode编码，UTF-8编码，之前一直单独知道这些东西，并没有想到它们的联系，来询问一下老师。

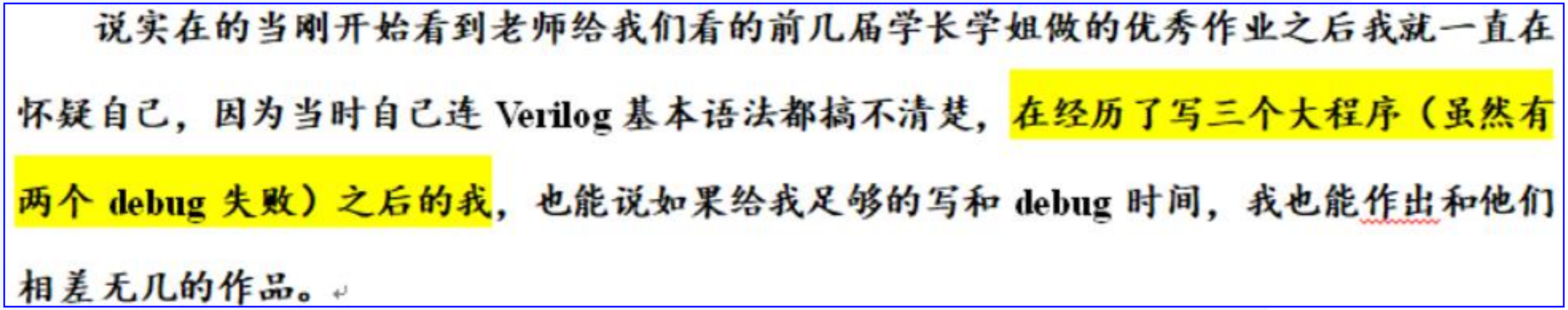
Perseverance



终于完成了痛并快乐着的数逻大作业，是对数逻的热爱支持着我考完试还在实验室做实验，做完才回家。

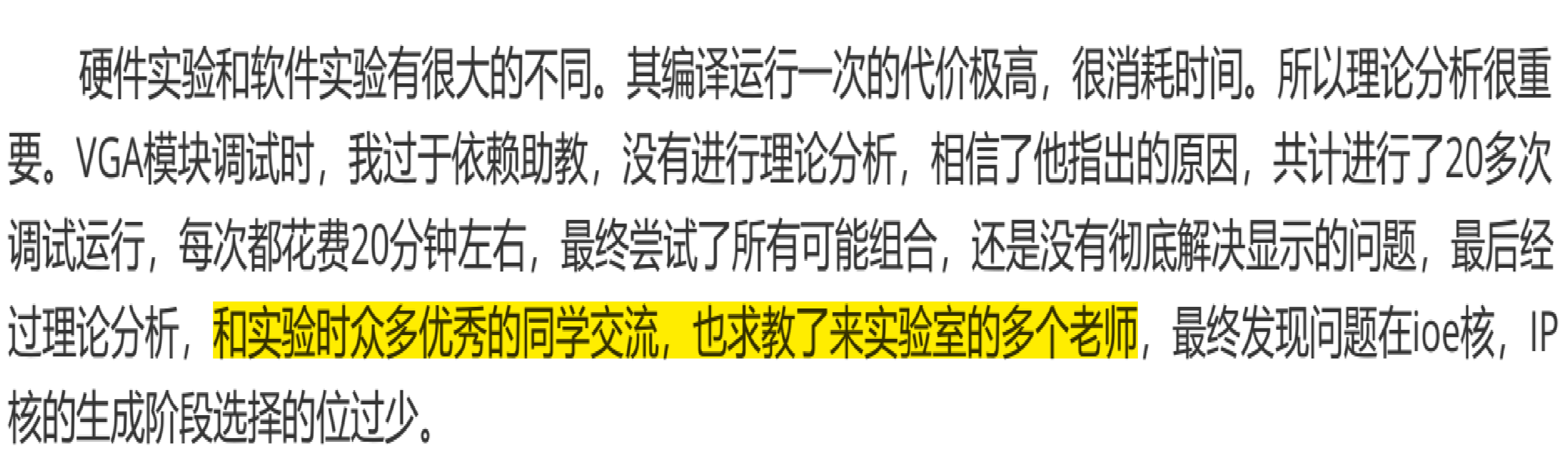
- 总结

其实我很早就开始做大作业了，但被键盘卡住了十天，中途数度放弃，但每当我挣扎着想要放弃时，我躺在床上都睡不安分，因为我从来没有过写不出作业的经历，也许是对这门课的热爱，也许是不服输的斗志，也许是自尊心，也许是不理想的小测分数让我恐慌，我还是做到了最后，终于，克服了一个键盘问题后，后面的内容不到一天就完成了，我感到十分有成就感。



说实在的当刚开始看到老师给我们看的前几届学长学姐做的优秀作业之后我就一直在怀疑自己，因为当时自己连 Verilog 基本语法都搞不清楚，在经历了写三个大程序（虽然有两个 debug 失败）之后的我，也能说如果给我足够的写和 debug 时间，我也能作出和他们相差无几的作品。

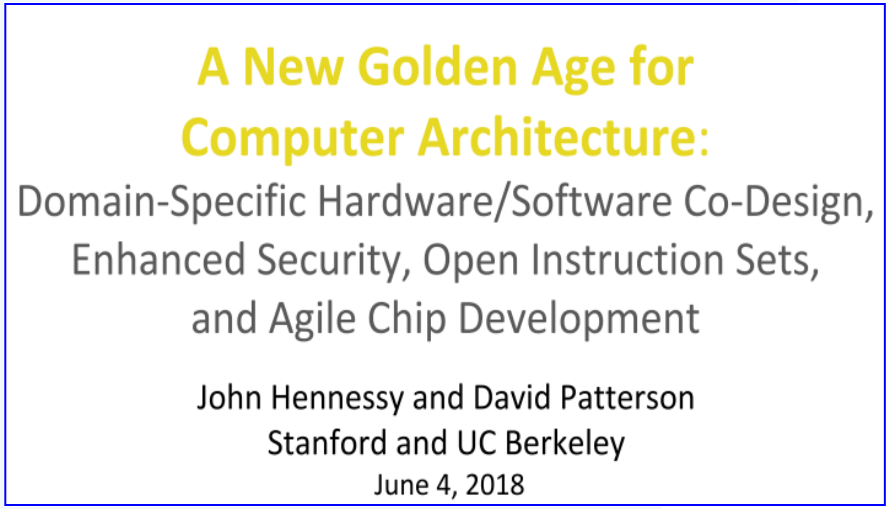
Asking for Help



硬件实验和软件实验有很大的不同。其编译运行一次的代价极高，很消耗时间。所以理论分析很重要。VGA模块调试时，我过于依赖助教，没有进行理论分析，相信了他指出的原因，共计进行了20多次调试运行，每次都花费20分钟左右，最终尝试了所有可能组合，还是没有彻底解决显示的问题，最后经过理论分析，和实验时众多优秀的同学交流，也求教了来实验室的多个老师，最终发现问题在ioe核，IP核的生成阶段选择的位过少。

Appendix A: Insights from 2017 Turing Award Winners

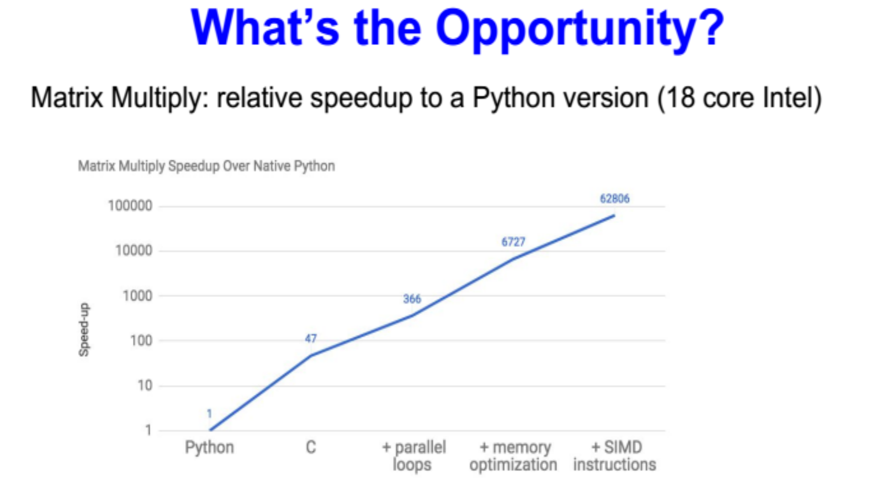
❖ Turing Lecture



A New Golden Age for Computer Architecture:

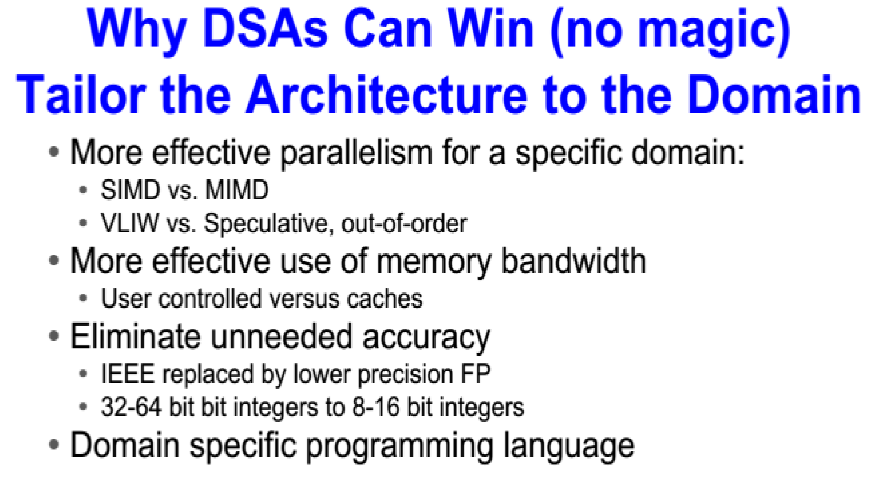
Domain-Specific Hardware/Software Co-Design, Enhanced Security, Open Instruction Sets, and Agile Chip Development

John Hennessy and David Patterson
Stanford and UC Berkeley
June 4, 2018



What's the Opportunity?

Matrix Multiply: relative speedup to a Python version (18 core Intel)

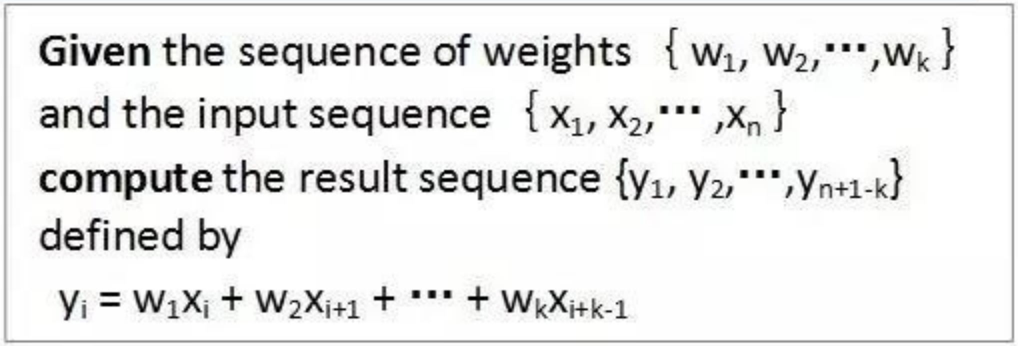


Why DSAs Can Win (no magic) Tailor the Architecture to the Domain

- More effective parallelism for a specific domain:
  - SIMD vs. MIMD
  - VLIW vs. Speculative, out-of-order
- More effective use of memory bandwidth
  - User controlled versus caches
- Eliminate unneeded accuracy
  - IEEE replaced by lower precision FP
  - 32-64 bit bit integers to 8-16 bit integers
- Domain specific programming language

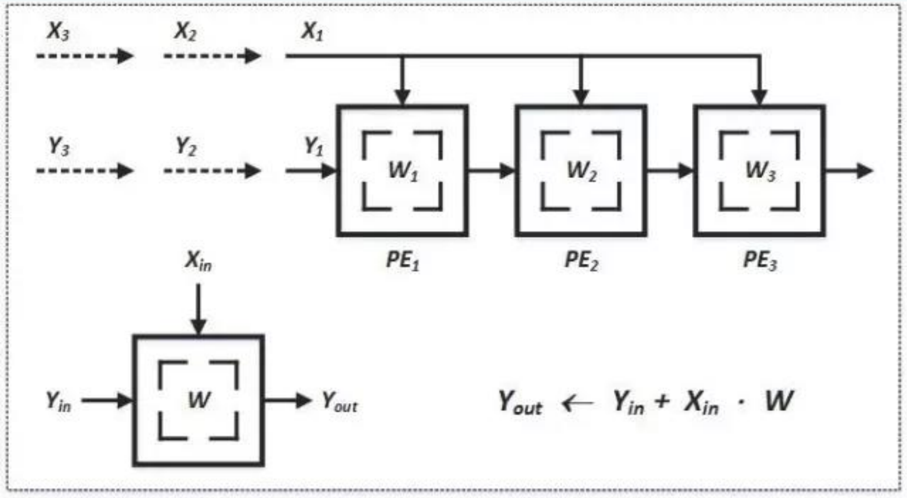
Appendix B: Some Motivating Examples (1/2)

❖ Hardware acceleration for Artificial Intelligence(AI)

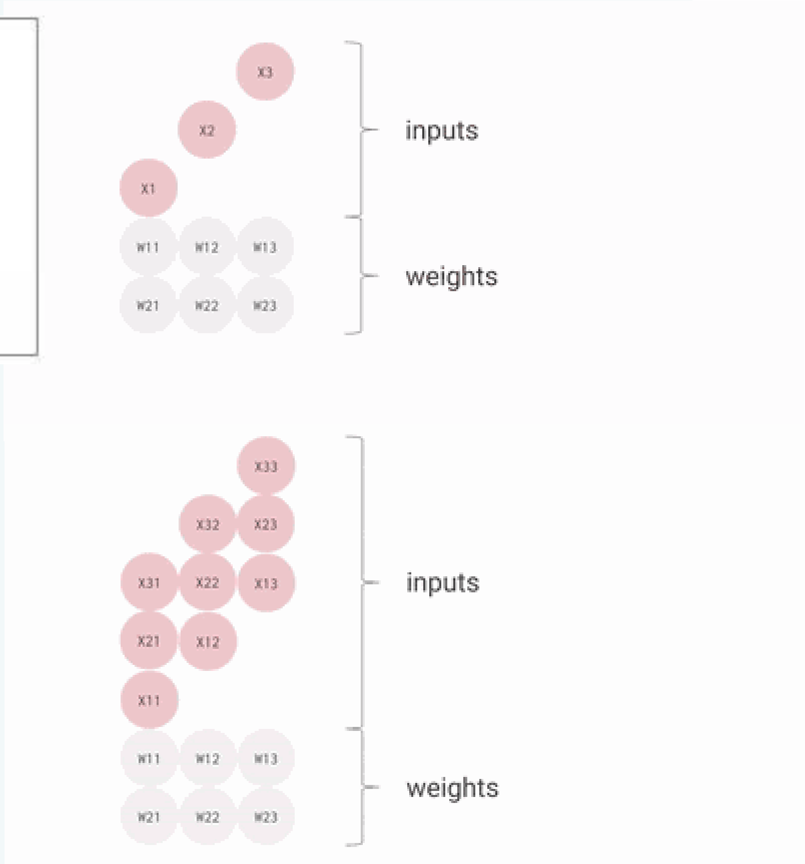


Given the sequence of weights $\{ w_1, w_2, \cdots, w_k \}$ and the input sequence $\{ x_1, x_2, \cdots, x_n \}$ **compute** the result sequence $\{y_1, y_2, \cdots, y_{n+1-k}\}$ defined by

$$y_i = w_1 x_i + w_2 x_{i+1} + \cdots + w_k x_{i+k-1}$$



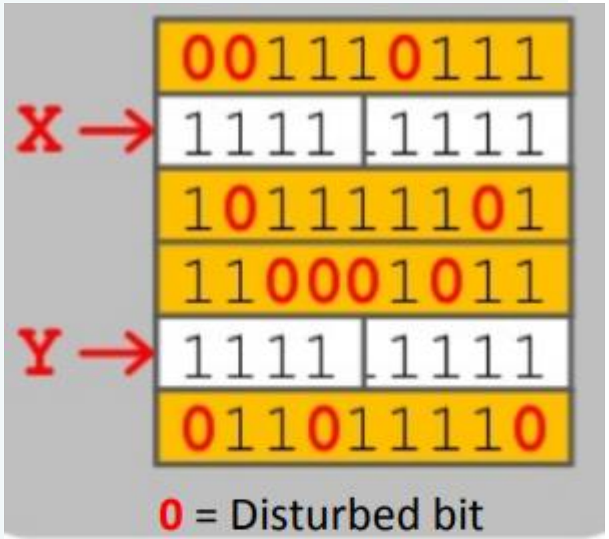
$$Y_{out} \leftarrow Y_{in} + X_{in} \cdot W$$



Appendix B: Some Motivating Examples (2/2)

Row Hammer

- Kim Y, Daly R, Kim J, et al. Flipping bits in memory without accessing them: an experimental study of DRAM disturbance errors. ISCA 2014

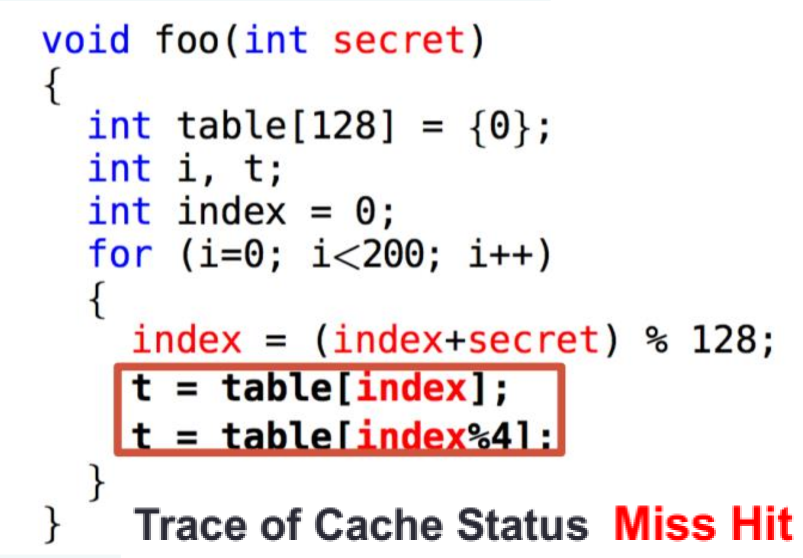


```
    001110111
X → 1111 1111
    101111101
    110001011
Y → 1111 1111
    011011110
```

0 = Disturbed bit

Cache Difference

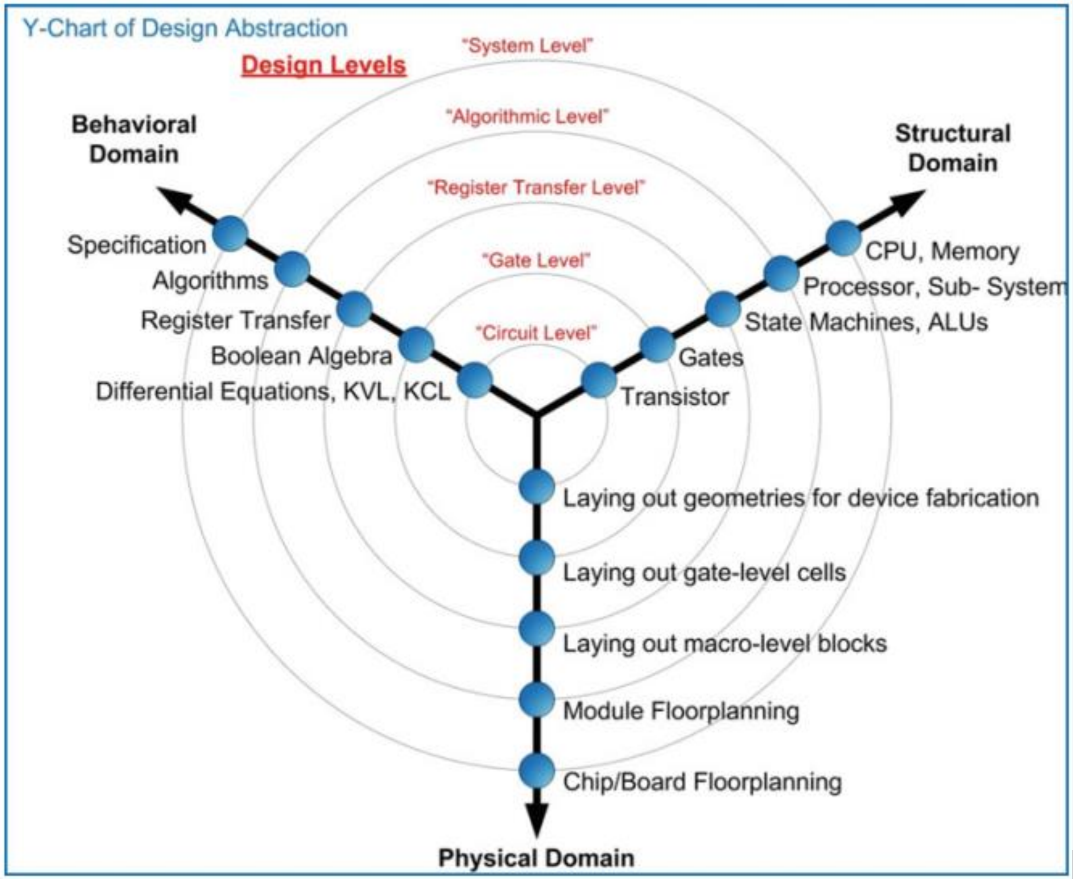
- Wang S, Wang P, Liu X, et al. CacheD: Identifying Cache-Based Timing Channels in Production Software. USENIX security symposium, 2017



```
void foo(int secret)
{
  int table[128] = {0};
  int i, t;
  int index = 0;
  for (i=0; i<200; i++)
  {
    index = (index+secret) % 128;
    t = table[index];
    t = table[index%4];
  }
}   Trace of Cache Status  Miss Hit
```

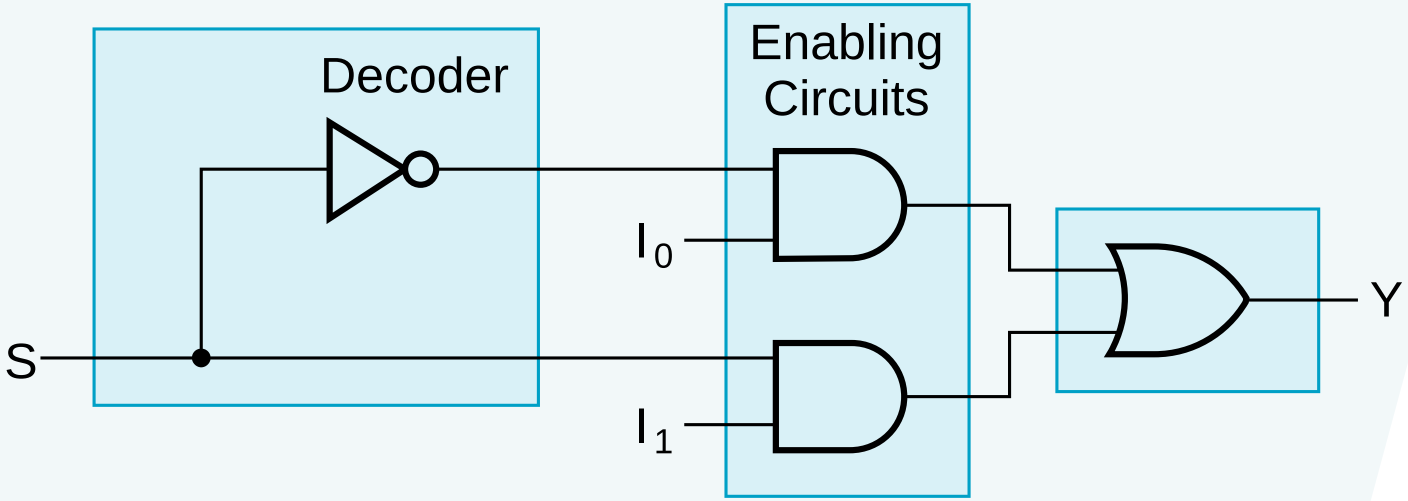
Top picture of the logic design

- **Behavioral**: specifies what a specific system does
- **Structural**: specifies how entities are connected together
- **Physical**: specifies how to build a structure



Behavioral and Structural Modeling

- 2-to-1-Line Multiplexer ($Y = Mux(S, I_0, I_1)$)



- **Behavioral modelling**

```
  if (!S)
     Y = I0;
   else
      Y = I1;
```

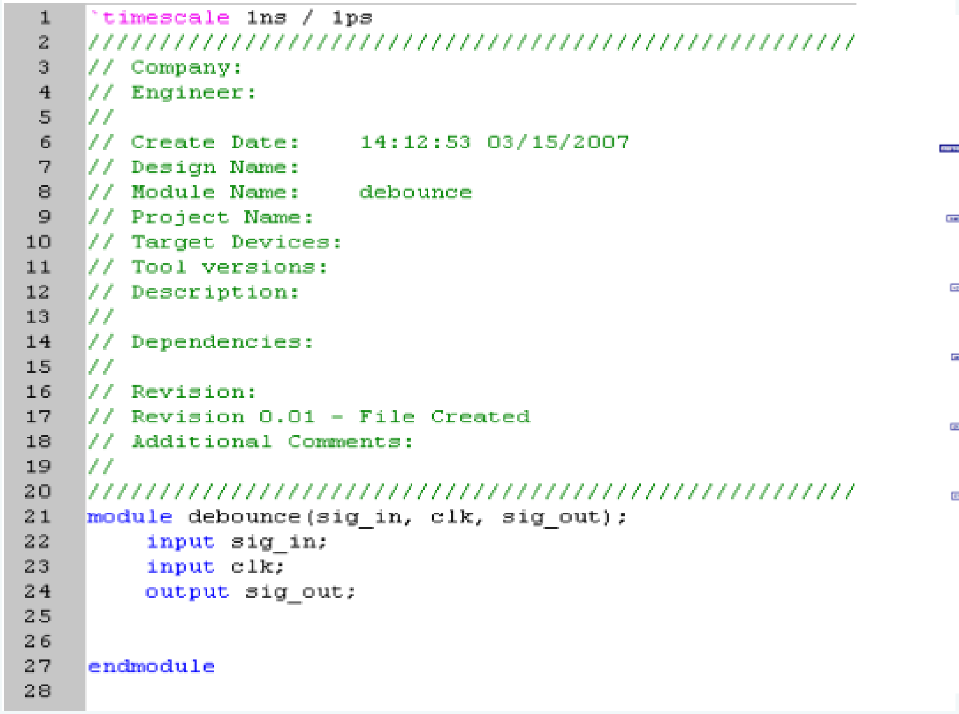
Structural modelling

```
not   u1(NS, S);
and  u2(sel0, I0, NS);
and  u3(sel1, I1, S);
or     u4(Y, sel0, sel1);
```

Two design methods

❖ HDL-Based Design

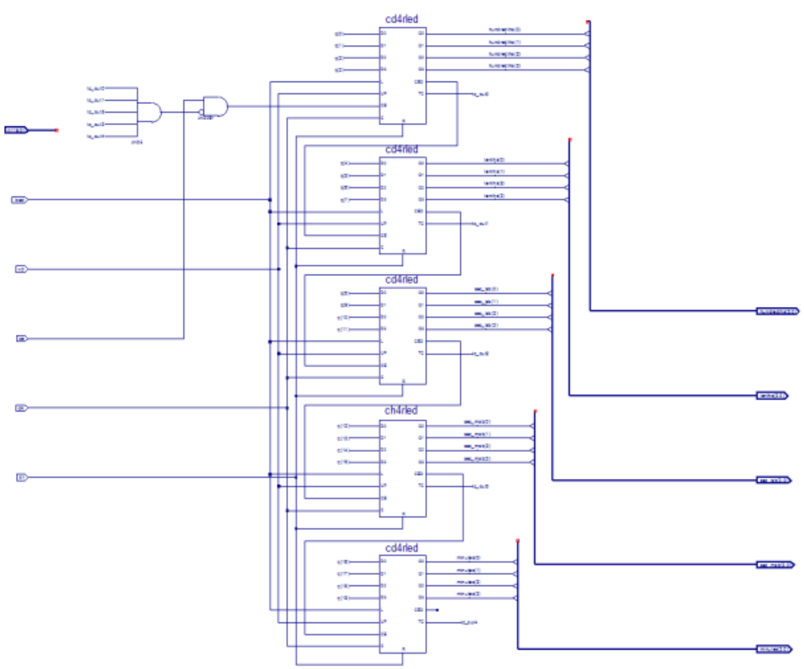
❖ Schematic-Based Design



```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date:    14:12:53 03/15/2007
// Design Name:
// Module Name:    debounce
// Project Name:
// Target Devices:
// Tool versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////
module debounce(sig_in, clk, sig_out);
     input sig_in;
     input clk;
     output sig_out;


endmodule

```

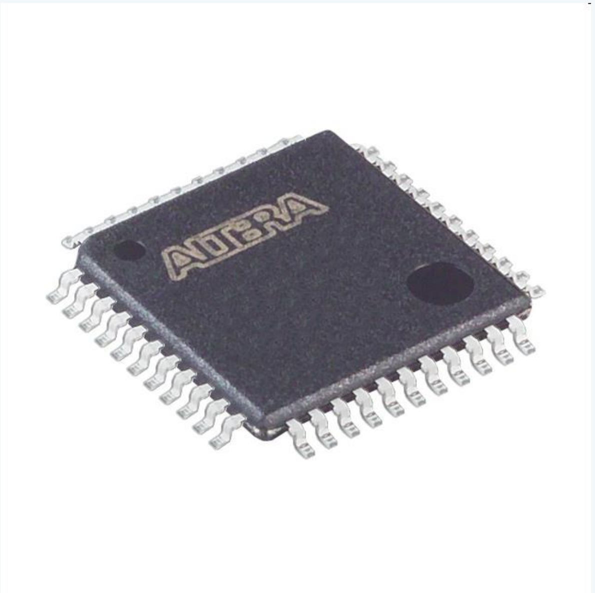


Verilog File in Xilinx ISE
Structural and Behavioral Modeling

Schematic in ISE Editor
Structural Modeling

Course Description

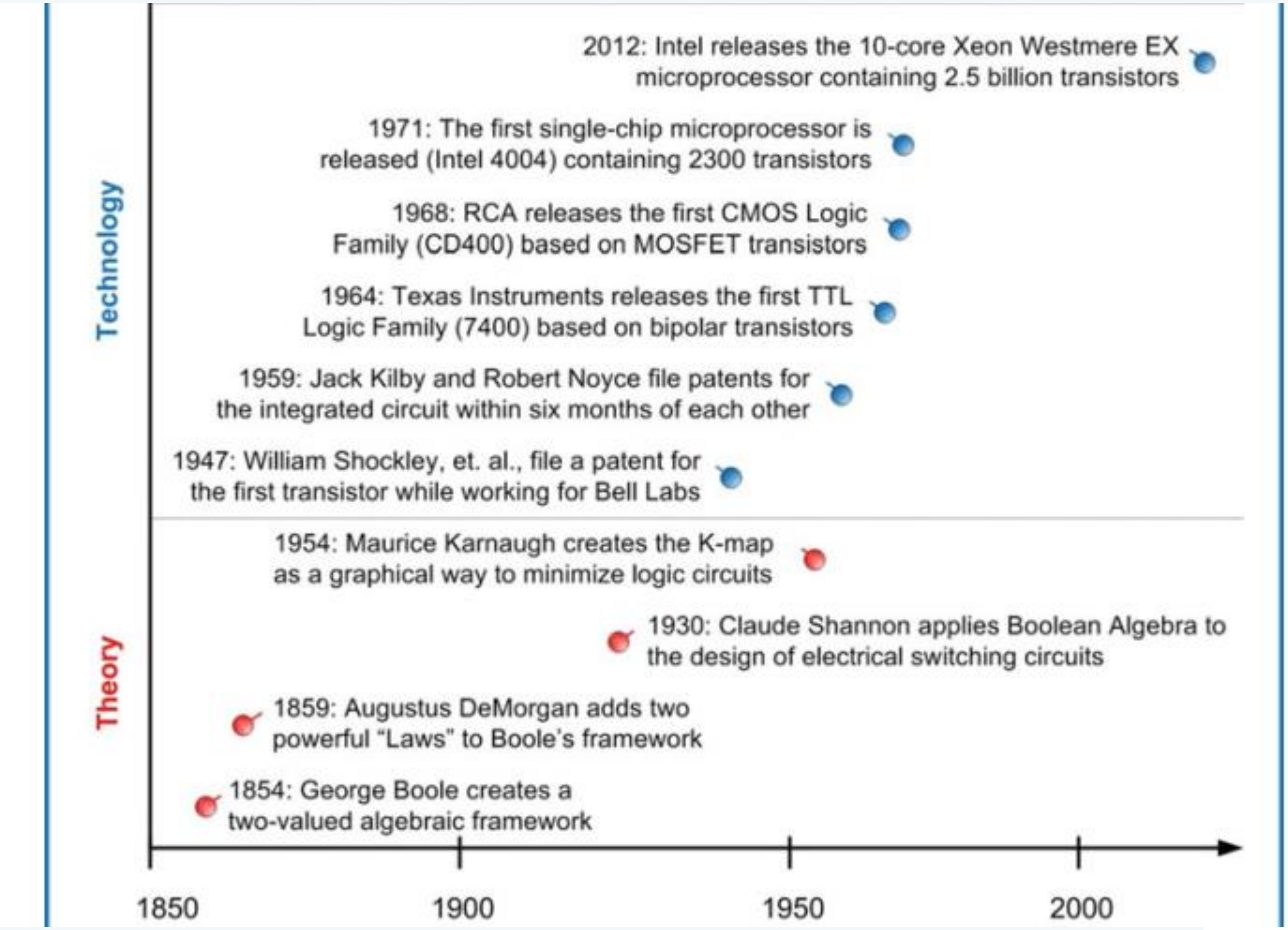
- National 2003 Teaching Program
- Digital System Fundamental



Course Objectives

- To be able to analyze and design digital logic systems by understanding formal foundations and selected design techniques.
  - Introduce basic theory and design methods for digital logic.
  - Give students basic skills to analysis and design electronic digital computer logic circuit
  - Prepare for the further studies on hardware related courses, such as
    - Computer Organization
    - Computer Architecture
    - Embedded Systems
    - Communication

Major Milestones in Digital Logic



2012: Intel releases the 10-core Xeon Westmere EX microprocessor containing 2.5 billion transistors

1971: The first single-chip microprocessor is released (Intel 4004) containing 2300 transistors

1968: RCA releases the first CMOS Logic Family (CD400) based on MOSFET transistors

1964: Texas Instruments releases the first TTL Logic Family (7400) based on bipolar transistors

1959: Jack Kilby and Robert Noyce file patents for the integrated circuit within six months of each other

1947: William Shockley, et. al., file a patent for the first transistor while working for Bell Labs

1954: Maurice Karnaugh creates the K-map as a graphical way to minimize logic circuits

1930: Claude Shannon applies Boolean Algebra to the design of electrical switching circuits

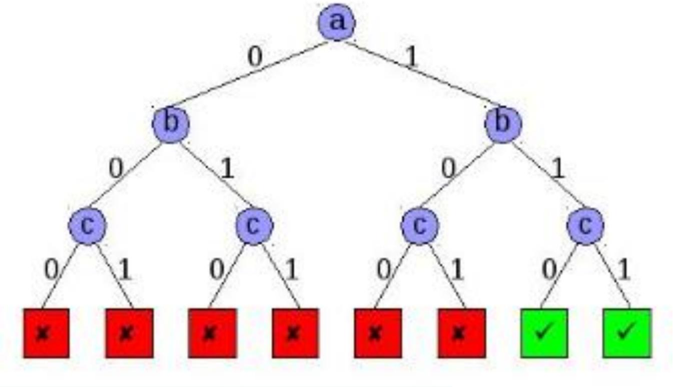
1859: Augustus DeMorgan adds two powerful "Laws" to Boole's framework

1854: George Boole creates a two-valued algebraic framework

From George Boole to….

- The Boolean satisfiability problem (SAT) is the problem of determining if there exists an interpretation that satisfies a given Boolean formula.
- SAT is the first problem that was proven to be NP-complete in 1971.
- Whether SAT has a polynomial-time algorithm is equivalent to the P versus NP problem, which is a famous open problem in the theory of computing.

(a v b) ^ (a v ¬c) ^ (¬a v b) ^ (a v c)

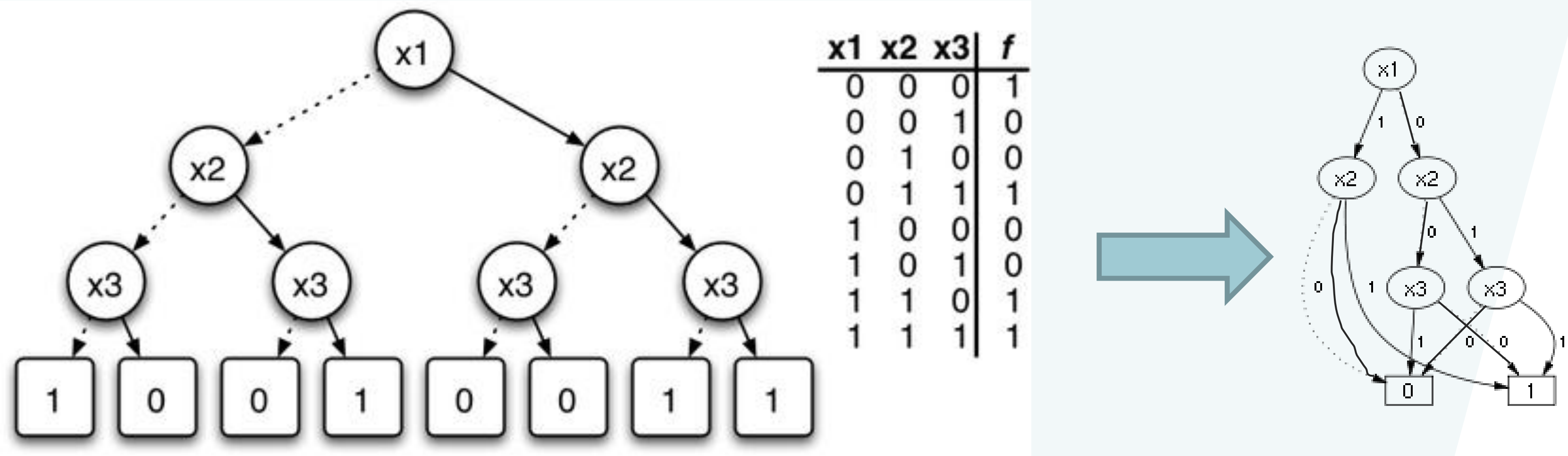


Shaowei Cai, Xindi Zhang, Deep Cooperation of CDCL and Local Search for SAT[C], Conference on Theory and Applications of Satisfiability Testing (SAT 2021) (best paper)

refers to: https://mp.weixin.qq.com/s/c_BFeXbjkN6C8XlrPA1gaQ

From George Boole to….

- A binary decision diagram (BDD) or branching program is a data structure that is used to represent a Boolean function.



x1	x2	x3	*f*
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

- The full potential for efficient algorithms was investigated by Randal Bryant at Carnegie Mellon University.
- “one of the only really fundamental data structures that came out in the last twenty-five years” 、 “Bryant's 1986 paper was one of the most-cited papers in computer science”

—— Donald Knuth

Course Resources

- **Software & Online practice**
  - 学生APP端参与随堂测试：https://v.youku.com/v_show/id_XMzc4Mzc3NTcyMA==.html
  - HDLBits-Verilog Practice: https://hdlbits.01xz.net/wiki/Problem_sets#Getting_Started
  - Logisim: https://sourceforge.net/projects/circuit/?source=typ_redirect
  - Logisim tutorial: https://www.bilibili.com/video/av842668792/
  - online Karnaugh map generator: http://www.32x8.com
- **References**
  - course notes from a senior：https://note.isshikih.top/cour_note/D2QD_DigitalDesign/
  - R. Mano and Ciletti, 《Logic design with an introduction to the Verilog》 Fifth edition
  - Brock J. LaMeres, 《Introduction to Logic Circuits & Logic Design with Verilog》 ISBN: 978-3-319-53882-2, Springer, 2017
  - 夏宇闻. Verilog数字系统设计教程. 北京：北京航空航天大学出版社
  - Verilog 教程：https://www.runoob.com/w3cnote/verilog-tutorial.html
- **Reading Supplements**
  - http://wps.pearsoned.com/ecs_mano_lcdf_5/248/63706/16308896.cw/index.html