EDA 软件设计 I Lecture 18

EDATEMENT

FD VIII.

ED PHILL

FD VIII

AND THE PARTY.

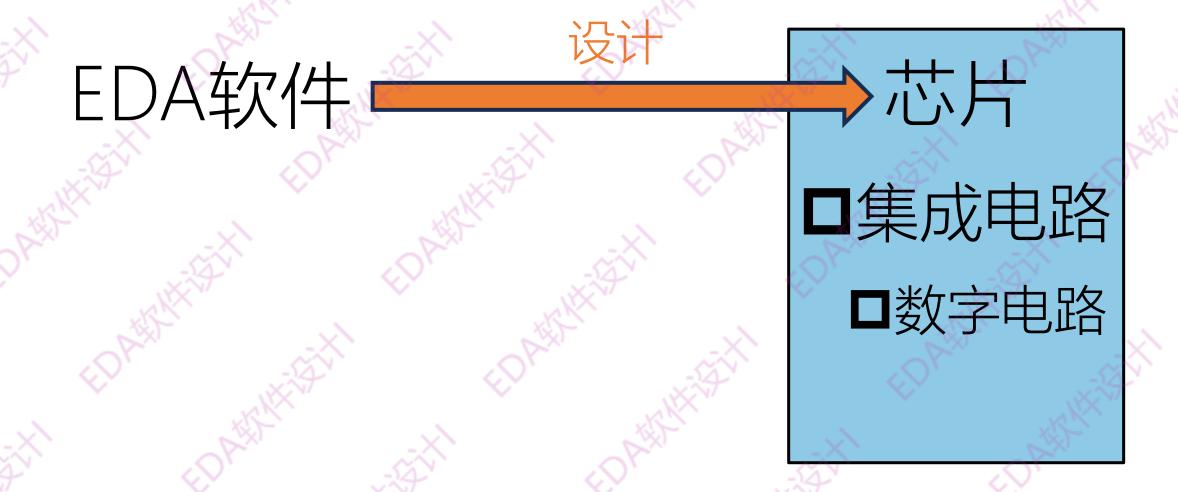
AND THE WALL

EDAWKI JAKA

EDRIFIET

ED RIKKHT TO THE

EDATELLE



Chapter 3: 数字电路基础

- ・逻辑代数 (布尔代数)
 - 组合逻辑电路
 - 时序逻辑电路

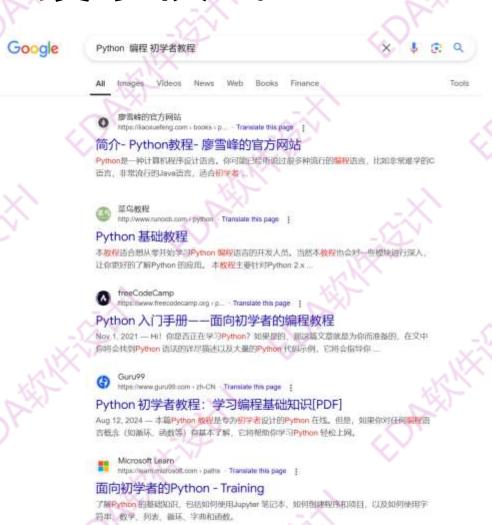
布尔代数——数字电路的"语言"

你觉得你在生活中"接触、使用"过布尔代数吗?

搜索引擎: 常用搜索模式

关键字1 关键字2 关键字3— —例如: Python 编程 初学 者教程

- 返回:包含所有这三个关键词的页面
- 例子: 查找Python教程的初 学者教程



搜索技巧: 1

- ·关键字1 | 关键字2 ——例 如: 布尔代数 | 逻辑代数
 - •返回:包含任一关键词的页面,可以更全面的获取信息
 - 例子: 寻找关于"布尔代数" 或者"逻辑代数"的资料



搜索技巧: - (排除关键词)

- · "关键字1" –"关键字2"" —例如: 布尔代数 – 百度百 科
 - •返回:包含"关键字1",但会排除包含"关键字2"的页面
 - 例子: 查找"布尔代数"内容 的同时避免了"百度百科""相 关的内容



提高"搜商"

- **精准搜索 ("")** : 关键词加上双引号,返回和关键词完全吻合的搜索 结果
- 站内搜索 (site: 网址 关键字): 在输入的网站内进行关键字搜索
- **同义词搜索 (~)** : 在未能准确判断关键词的情况下,可以通过~进行 同义词搜索
- 使用*填充内容:比如在搜索时,只记得其中一部分内容,另一部分忘记了,可以用*代替不确定内容
- 指定搜索范围 (...): 指定数字范围的内容, 可以是年份、版本等

什么是布尔代数?

What is Boolean Algebra

什么是布尔代数?

・严谨定义

- 布尔代数 (Boolean Algebra) 是代数的一个数学分支,用来研究在特定集合上定义的运算及其规则
- 它由一个包含二值(0和1)的集合以及两种主要运算组成
- 这些运算满足特定的公理和定律,使布尔代数成为一种特殊的代数系统
- 简化理解: "用于处理二值变量 (0和1) 的代数运算"

"布尔代数"="逻辑代数"

- Note: Boolean Algebra背景下,<mark>"布尔"="逻辑"</mark>
- ·如何理解逻辑代数作为数字电路的基础or语言:
 - ① 所谓"逻辑",指"条件"与"结果"的关系
 - ② 数字电路的输入信号反映"条件",而用数字电路的输出反映 "结果"
 - ③ 从而使数字电路的输入和输出之间代表了一定的逻辑关系

逻辑代数起源及如今地位

- •起源:布尔代数由数学家**乔治·布尔 (George Boole)**在19世纪提出, 作为**逻辑学**的一种形式化方法
 - 由乔治·布尔(George Boole)在他的第一本书 *The Mathematical Analysis of Logic(1847)* 中引入
 - 并在他的另一本书 *An Investigation of the Laws of Thought(1854)* 中得到了完善
 - "布尔代数" 这个术语最初由 Sheffer 与 1913 提出
- 地位:布尔代数已经称为数字电子的基础,在所有现代编程语言中都提供了逻辑代数操作,逻辑代数也被用于集合论和统计学

布尔代数基本概念

FD VIEW

EDATE

EDARKER

EDRIFT

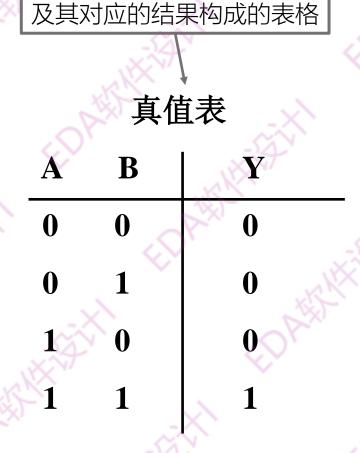
布尔代数基本概念

- 布尔代数中的变量按一定逻辑关系进行运算,是分析和设计数字
 电路的数学工具
- **布尔变量/逻辑变量**:布尔(逻辑)代数中的变量即是布尔(逻辑)变量,**只有0或1两种取值(又称为逻辑状态)**
 - 0和1并不表示数值的大小
 - 而是表示两种对立的逻辑状态
 - · 如: 电平的高与低、事件的是与非、开关的通与断

- 布尔代数的基本运算:
 - ① **与运算 (AND)**:决定事件发生的各条件中,所有条件都具备,时间才会发生或成立,符号一般为·或^
 - 当且仅当所有输入为1时,结果为1。例如, A ^ B 仅在A=1且B=1时为1
 - ② <mark>或运算(OR):决定事件发生的各条件中,有一个或者一个以上的条件具备,事件就会发生或成立,符号一般为+或></mark>
 - 只要有一个输入为1, 结果为1。例如, A v B 在A=1或B=1时为1
 - ③ <mark>非运算(NOT)</mark>:决定事件发生的条件只有一个,条件不具备时事件发生,条件具备时事件不发生,符号一般为<mark>¬或-</mark>
 - 非运算将输入反转,即A的非(¬A)在A=1时为0,在A=0时为1

- **与运算**的逻辑表达式: Y=A⋅B=A∧B
 =AB
- 与运算的运算规则:

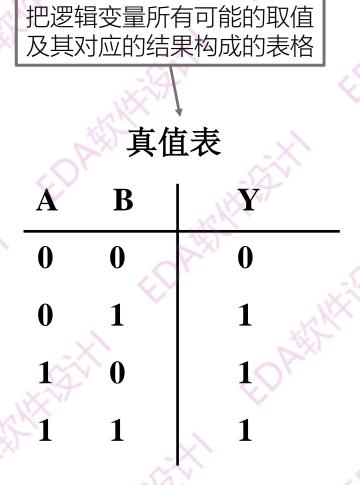
$$0 \cdot 0 = 0$$
 $0 \cdot 1 = 0$
 $1 \cdot 0 = 0$ $1 \cdot 1 = 1$



把逻辑变量所有可能的取值

- 或运算的逻辑表达式: $Y = A + B = A \vee B$
- 与运算的运算规则:

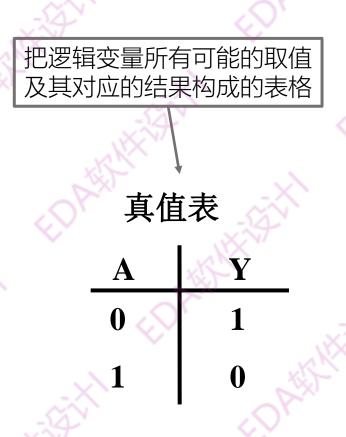
$$0 + 0 = 0$$
 $0 + 1 = 1$
 $1 + 0 = 1$ $1 + 1 = 1$



- **非运算**的逻辑表达式: **Y**= **A**
- 与运算的运算规则:

$$\overline{0} = 1$$

$$\overline{1} = 0$$



布尔代数基本概念

- 这三个运算(A、V、¬)被称为**基础运算**,"基础"意味着**其他的** 逻辑运算可以通过它们的组合得到——称为复合逻辑运算
 - ☑ 从数学上更严谨地说,可以认为只有¬运算和(∧、∨)其中之一才是基础运算,因为:
 - ☑可以通过¬运算和∧运算,推导出∨运算(练习)
 - 暨也可以通过¬运算和v运算,推导出∧运算(练习)
- ●复合逻辑运算: "与非"、"或非"、"异或"的运算规则与真值表呢——练习
 —练习

Question

一如果真值0和1被看作整数的话,逻辑运算

也可以由普通的运算表示,比如与运算:

 $x \wedge y = \min(x, y)$, 那或运算和非运算呢?

布尔代数基本定律和规则

打开布尔代数世界的钥匙

布尔代数基本定律和规则

• 交換律:

- 与运算: A·B = B·A
- 或运算: A + B = B + A

• 结合律:

- 与运算: (A · B) · C = A · (B · C)
- 或运算: (A + B) + C = A + (B + C)

• 分配律:

- $A \cdot (B + C) = A \cdot B + A \cdot C$
- $A + (B \cdot C) = (A + B) \cdot (A + C)$

布尔代数基本定律和规则

• 双重否定律:

•
$$\neg(\neg A) = A$$

- 德摩根定律(证明,练习):
 - $\neg(A \cdot B) = \neg A + \neg B$
 - $\neg(A + B) = \neg A \cdot \neg B$

貸恒等律(同一律)、零律、单位律

运算律:

- 布尔代数的核心规则,掌握它们可以简化逻辑表达式
- · 难点(精髓):熟练掌握不同布尔表达 式推导出的结果

例如: (x ∨ y) ∧ ¬(x ∧ y)

布尔/逻辑函数

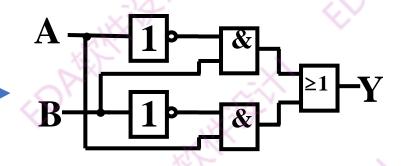
Logic Function

描述逻辑变量之间关系的数学表达或规则

 $(x \lor y) \land \neg(x \land y)$

逻辑函数

- 输入和输出均为逻辑变量
- ・逻辑函数的4种表达方式:
 - ①逻辑电路图
 - ②逻辑表达式 (代数式)
 - ③ 真值表: n个输入变量有2ⁿ种组合
 - 4 卡诺图



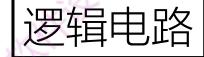
$$Y = \overline{A}B + A\overline{B}$$

①逻辑电路

- •逻辑电路:
 - ① 是数字电路的组成部分(数字电路不仅包含逻辑电路,还包括寄存器、计数器、触发器等存储和控制元件)
 - ② 用于实现布尔代数运算的电路
 - ③ 通常由**逻辑门(与门、或门、非门)**构成,<mark>它们是实现特</mark> 定逻辑运算的基本单元,连接在一起形成复杂的逻辑功能
- 逻辑代数与逻辑电路的关系:逻辑代数是逻辑电路设计的数学基础,通过布尔运算规则指导逻辑电路的实现和简化





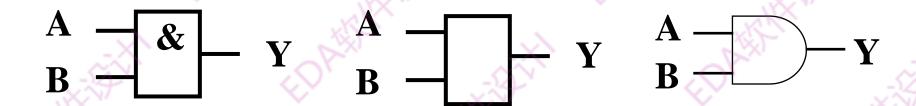




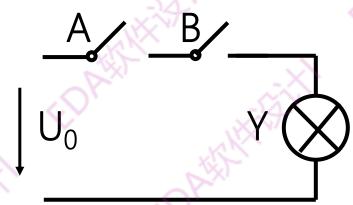
逻辑代数

逻辑电路中的逻辑门

• 实现"与运算"的逻辑电路成为**与门**, 其**逻辑符号**通常表达为:



• 与门基本结构:



规定: 开关(A、B)合为逻辑"1"

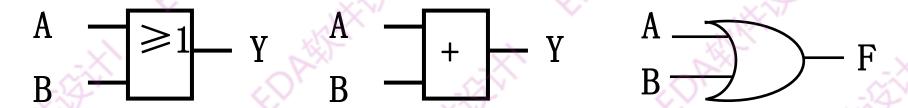
开关(A、B) 断为逻辑"0"

灯(Y) 亮为逻辑"1"

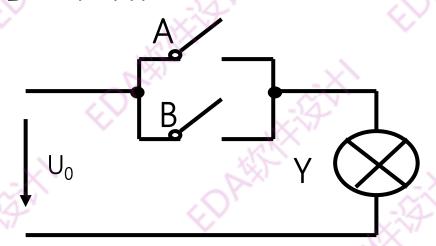
灯(Y)灭为逻辑"0"

逻辑电路中的逻辑门

•实现"或运算()"的逻辑电路成为或门,其逻辑符号通常表达为:



•或门基本结构



规定: 开关(A、B)合为逻辑"1"

开关 (A、B) 断为逻辑 "0"

灯(Y) 亮为逻辑"1"

灯(Y)灭为逻辑"0"

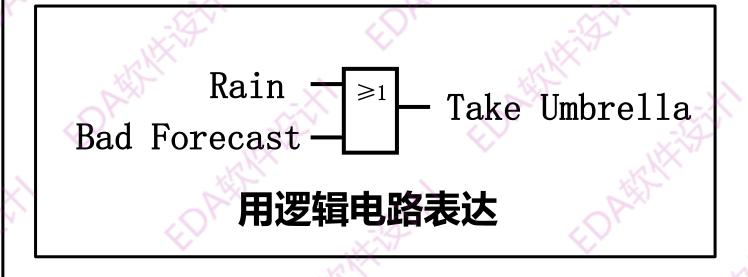
逻辑门电路图形符号

		XX		Towns and the second	V-34-1
		1/iX	表 C1 基本逻辑门电路图	形符号	XX
序号	名称	GB/I	4728.12-1996	国外流行图形符号	曾用图形符号
	1370	限定符号	国标图形符号	pa(2) this 2 14/2/ 3	El Allano (1 3
1	与门	& XX		±D-	
2	或门	≥1		\Rightarrow	(F)
3	- 非们-	_€ 			
XX.	× "	逻辑非入和出	1	>-	<u></u>
) 4	与非门		*- <		
5	或非门	8	≥1	\Rightarrow	1+-
-		27	VN/	77.7	

I will take an umbrella with me if it is raining **or** the weather forecast is bad

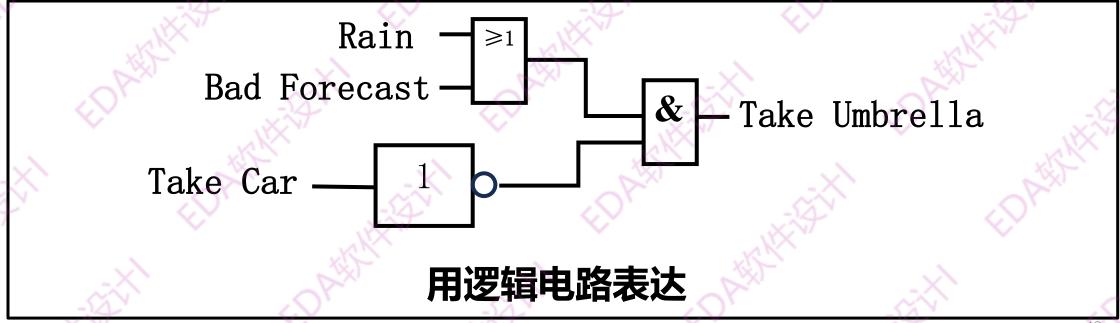
用逻辑表达式表达: Umbrella = Rain + Bad Forecast

Ó	Raining	Bad Forecast	Umbrella
	0	0	0
	0	1	1
	1	0	1
Ž	1		1 - 3
	用真	值表表	达



If I do **not** take the car then I will take an umbrella with me if it is raining **or** the weather forecast is bad

用逻辑表达式表达: Umbrella = ¬(take car) · (Rain + Bad Forecast)



若逻辑表达式复杂 →逻辑电 路复杂

逻辑代数化简的意义

化简称最简单的形式来对应逻辑电路

布尔函数
$$F = \overline{X} Y Z + X Y + X Y \overline{Z}$$

$$F = X Y (Z + \overline{Z}) + \overline{X} Y (分配律和交换律)$$

$$F = X Y (\overline{Z} + \overline{Z}) + \overline{X} Y (补余律/对偶律)$$

$$= 1$$

$$F = X Y + \overline{X} Y$$

$$F = (X + \overline{X}) Y (分配律)$$

$$F = (X + \overline{X}) Y (补余律/对偶律)$$

$$= 1$$

$$F = Y$$