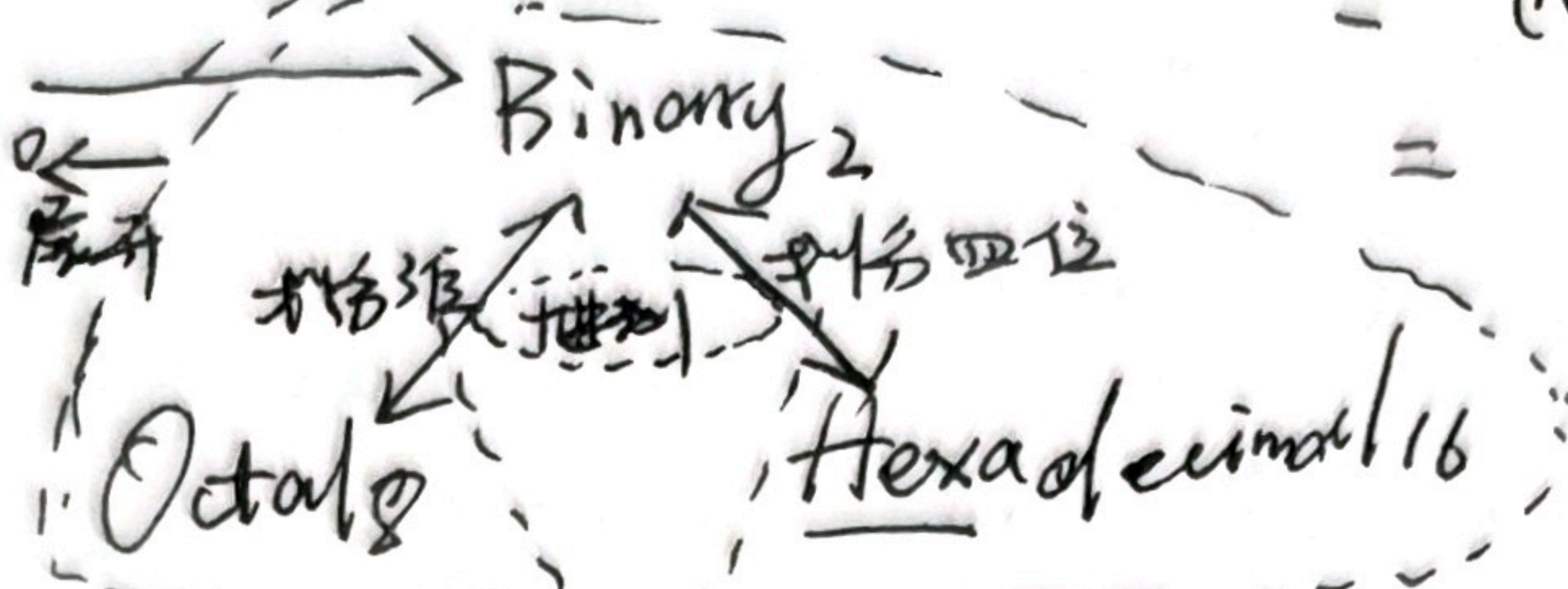


数字电路设计

< Block 1 >

1.1 进制与编码

< 进制 >
Decimal



Most Significant Bit
MSB/MSD LSD/LSB

$$ABC.DEF_{10} \hat{=} A'B'C'.D'E'F'_2$$

$$= (A \cdot 10 + B)10 + C + \dots$$

$$= ((A' \cdot 2) + B')2 + C'$$

$$+ ((F' \cdot 2^1 + E')2^1 + D')2^1$$

注意可以进位

和最高端(小段)的 0 padding

两部分

除以 2, 取余数, 从 LSD -> MSD

小数

乘以 2, 取整数

从 MSD 向左给至小数点前一个

< 数值的表示 >

a. 176 码 Sign-Magnitude Representation.

MSB 表符号, 0 正 1 负, 值域 $[-2^{n-1}, 2^{n-1}-1]$, 两种 "0" 表示方式.

四则运算: 同符号相加可能溢出, 不同符号相加判断绝对值大小, 大数减小数, 赋予大数的符号.

b. 补码 Two's Complement.

一种 "0" 表示方式, MSB 表数值位, 有 1 即为负, 值域 $[-2^{n-1}, 2^{n-1}-1]$

补码定义: $2^n - B = (2^n - 1 - B) + 1 = B$ 按位取反 + 1.

不溢出情况下, 正数的补码是其本身二进制, 负数的补码是其按位取反 + 1.

补码延拓: $n \rightarrow m$ bit, 在 n 位补码前加 $(m-n)$ 位 MSB 相同的 bit.

四则运算及 Overflow 规则: 相同 MSB 相加, 若 MSB 改变, Overflow!

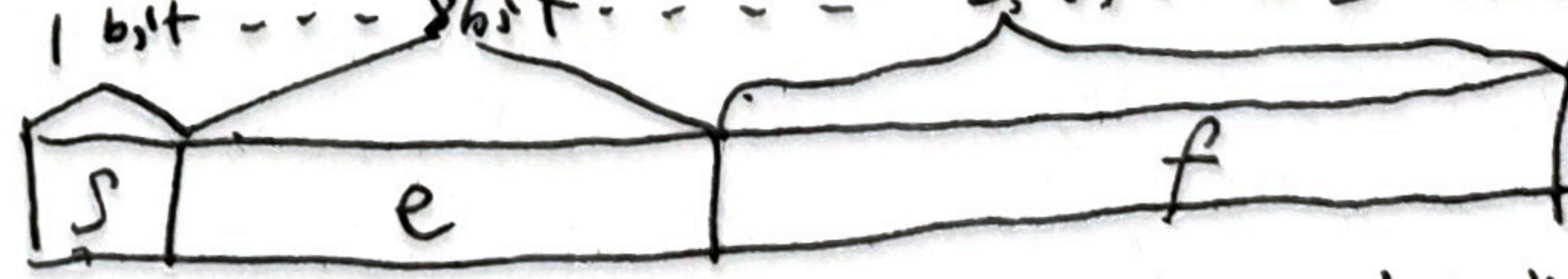
(Hint: 必须同符号, 不同符号如果进位补码, 结果依然对)

减法规则: 将减数变反 -> 取反补码, 然后相加.

< Floating Point (f.p.) >

科学记法: $Value = \pm 1. Mantissa \times Base^{(Exponent)}$

IEEE-754 Format f.p.:

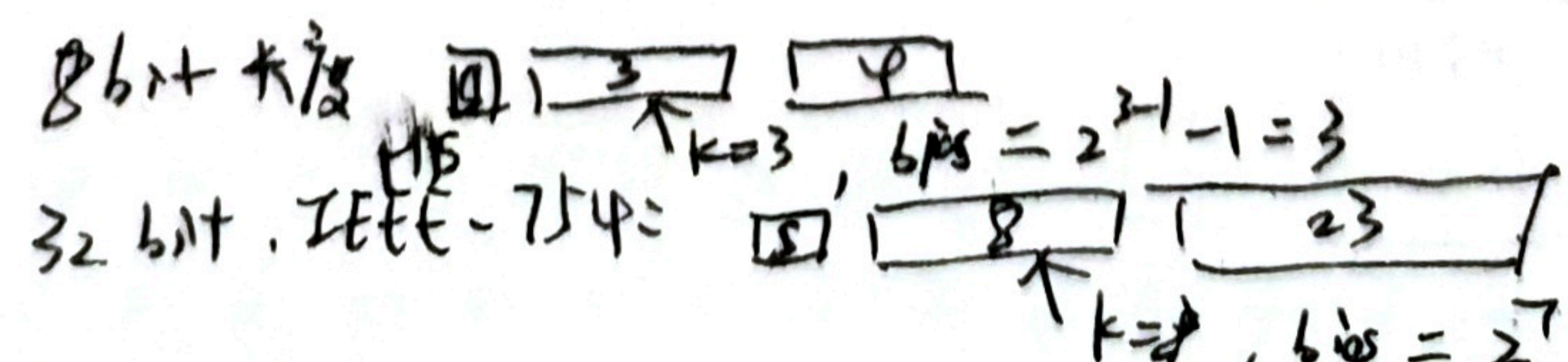


分为 Normal 和 Denormal 类型, 前者是标准形式, 后者是 Normal

$$Normal Value = (-1)^S 1.f \times 2^{e-bias}$$

任意数表示: $value = \pm 1. \overbrace{m_1 \dots m_k}^{mantissa} \times 2^{(e-bias)}$ 左移位数 (为 2 的幂), 这样是自然, 因为 8 bit

$$bias = 2^{k-1} - 1, k \text{ 为 } e \dots e \text{ 的 bit 数.}$$



< Code, 编码 >

a. 4-bit BCD 码, 对应0-9的10, 11, 即 1010, 1011 对应 "+" "-" 符号, 其它十进制数对应

b. Gray Code, 格雷码, 每两个相邻码之间只有一个数位变化

构造方式: 子串顺序二进制构造, 从 LSB 到 MSB, 如果本位与前一位相同, Gray Code 置 0, 否则置 1, 首位不变.

c. ASCII Code

7-bit Code, 但是经常存储在 8-bit 中, 第 8 位为校验码.

d. Unicode

16-bit 码, 使用 4 位 Hex 表示

e. Parity Checking

单个位校验码, 奇数个 1, 偶数个 1 在 8 位首位, 例如在 7-bit ASCII 前加 1-bit 校验码

1.2 Switching Algebra & Combinational Logic Design.

< 开关代数 >

① AND, OR, NOT, NAND, NOR, Exclusive-OR, Exclusive-NOR, Buffer $\rightarrow$ 上拉, 下拉. $F=A$

$$A \oplus B = AB + \bar{A}\bar{B}$$



$$\begin{aligned} & \left. \begin{aligned} & x + yz = (x+y)(x+z) \\ & x\bar{y} + xz + yz = x\bar{y} + \bar{x}z \\ & x + \bar{x}y = x + y \\ & (x+y)(\bar{x}+z)(y+z) = (x+y)(\bar{x}+z) \end{aligned} \right\} \end{aligned}$$

De Morgan's Theorem.

$$[f(x_1, \dots, x_n, +, \cdot)]' = f(\bar{x}_1, \dots, \bar{x}_n, \cdot, +)$$

变量取反, 交换 +, \cdot

③ 使用 De Morgan's Theorem, 可以转化 NAND/OR 电路到 NAND/NOR 电路! (两次取反)

XOR 中的运算, 任何范围取反, 偶数个反为 XOR, 奇数个反为 XNOR.

④ 化简, Boolean equation $\rightarrow$ Canonical SOP & POS.

$\rightarrow$ 原函数项分解
 $\rightarrow$ 反函数项分解

真值表
最小项代数和 (主析取范式)
最大项代数和 (主合取范式)
 $\rightarrow$ 表示一个组合逻辑函数.

<Block 2>

2.1 组合逻辑 → 卡诺图.

组合逻辑

画圈

从列号而按最大 2^n 开始, 寻到 ^{最大} Essential Prime Implicants (EPIs)
然后寻找其他最大 Prime Implicants (PI)
成为 2^n 的 1 重新选框, 迭代消元.

从大到小, 从分子到非必要

子逻辑 → mSOP, 如果子寻找 mPOS, 可以先到子逻辑到 1, 再 SOP → POS 转化.

否则, 寻找 0, 对变量元素值表示规则发生变化.

$$f(w, x, y, z) = \sum m(\dots)$$

$$d(w, x, y, z) = \sum m(\dots) \rightarrow \text{无关项, don't care case.}$$

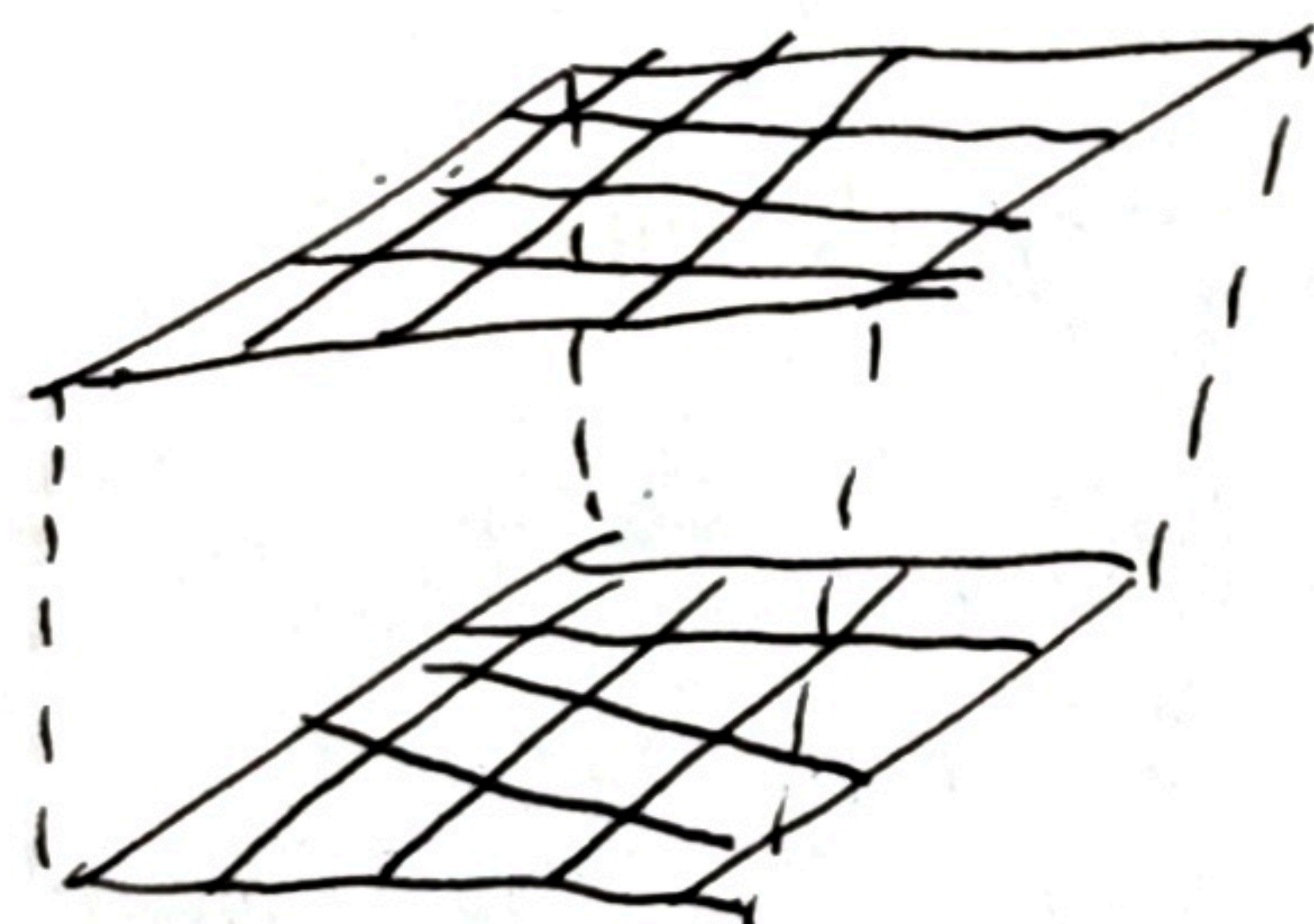
<7段数码管>

使用 BCD 码到状态输入.

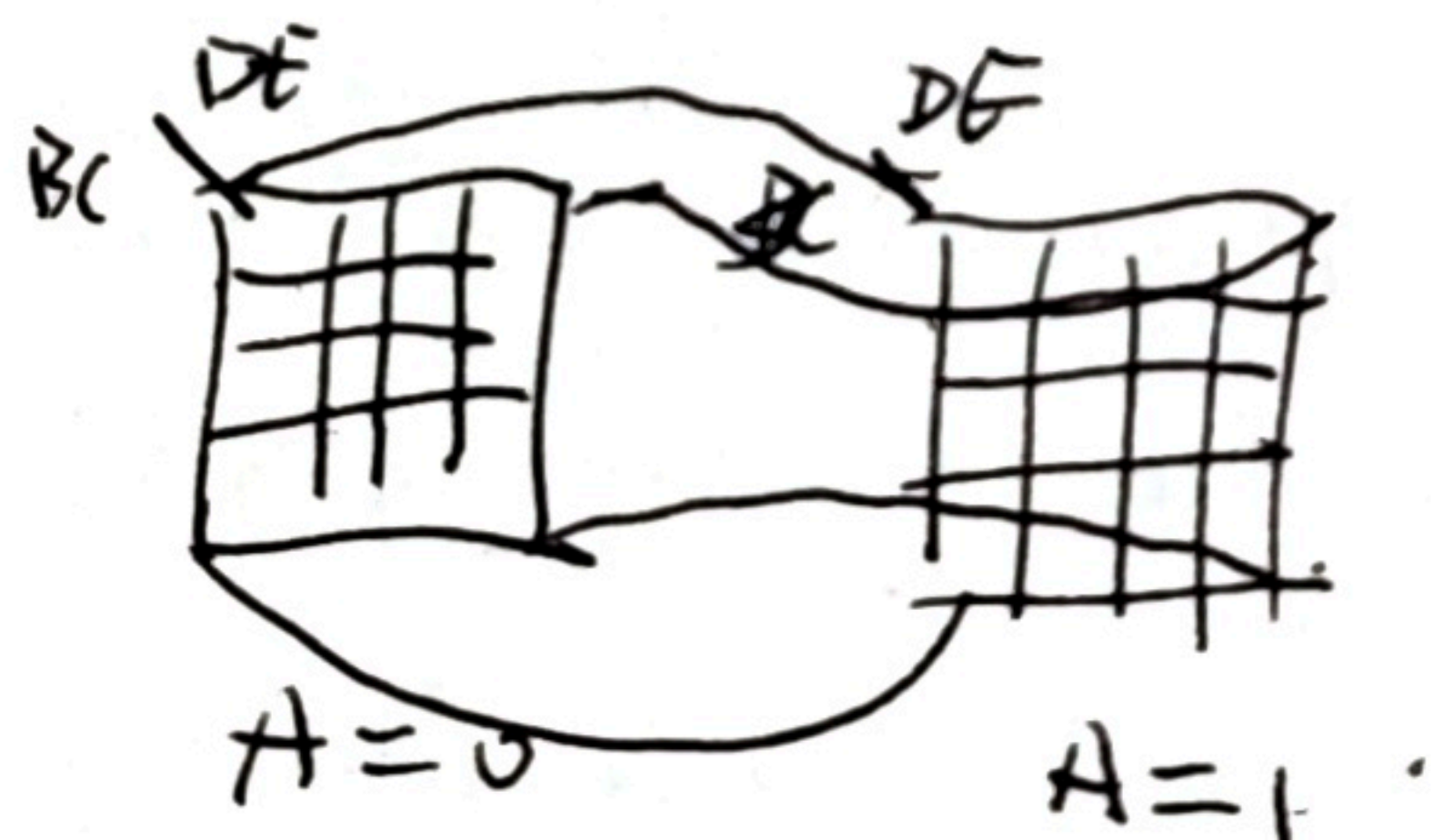
<两个两位产生器>

<卡诺图的 n 种形态>

小于等于 4 个变量的卡诺图均可二维平铺, 当在 5 维时是两个四位的组合



可用两张互联的四变量卡诺图来表示, 但要注意多了一个互联的项



<组合电路的设计>

Design 是 Synthesis 的超集.

自然语言表述 → 真值表, mSOP, mPOS, 卡诺图 → 化简 → 元件切换 → 验证

2.2 Time Hazards and Output Glitches.

有 output glitches 的电路有 hazards. unexpected output change

Technology mapping, 两次取用这行建设快的 NAND 或 NOR. (short pulse)

propagation delay

排一排

“与-或”电路可能有 static-1 hazard, 需增加一项来消除.

“或-与”电路可能有 static-0 hazard, 需增加一项来消除.

或非-或非

“与-或”中出现 static-0, “或-与”中出现 static-1 hazard, 都是设计时很

针对于二级电路.

愚蠢的电路.

<Block 3> 时序逻辑

3.1. Latches and Flip-Flops.

- Synchronous → discrete moments in times. Single 状态 behavior
- Asynchronous → any instant of time Single 状态 behavior.

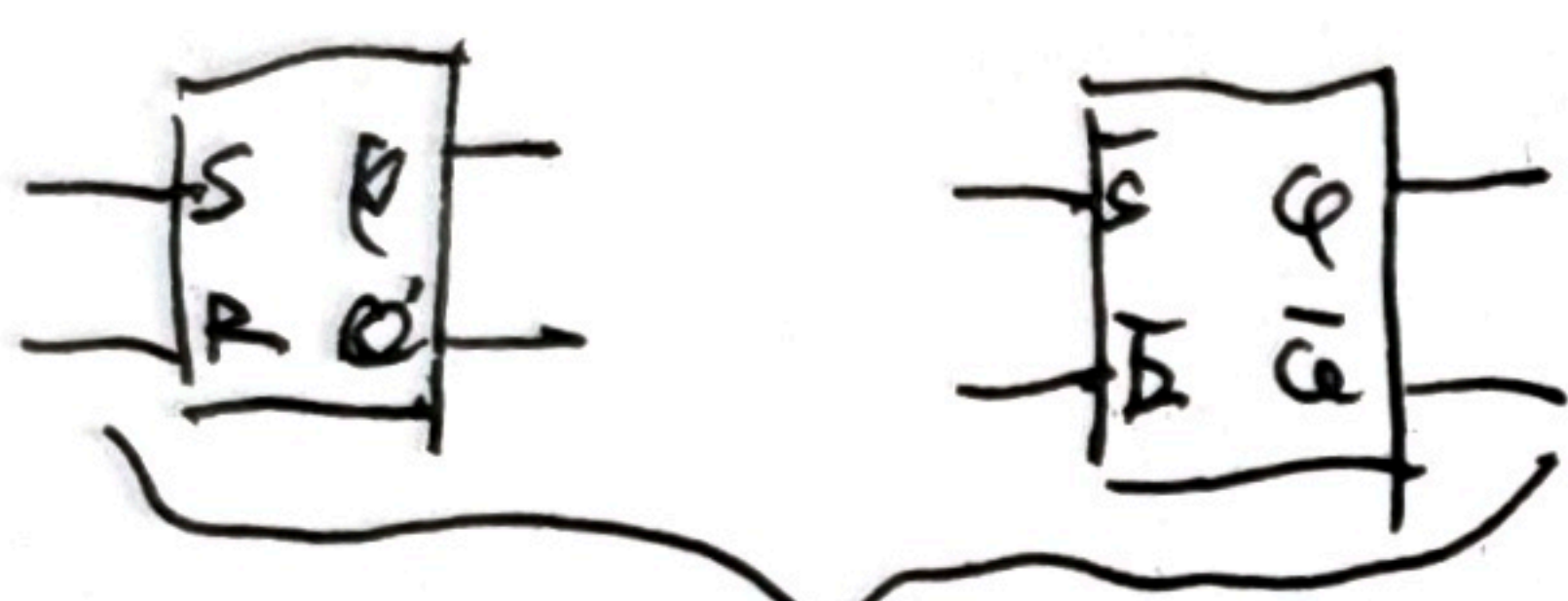
② duty cycle (占空比) } 上升沿有效 (高电平有效) $\frac{t_H}{t_{per}} = \frac{t_H}{t_H + t_L}$
 下降沿有效 (低电平有效) $\frac{t_L}{t_{per}} = \frac{t_L}{t_H + t_L}$

③ 亚稳态 Metastability } 定义: 元件无法在指定时刻达到确定状态。
 原因: asynchronous inputs do not meet the setup and hold times.

三种触发方式: 电平, 脉冲(主从), 边沿, 无论哪种在 setup 和 hold times.
 输入变化, 都不在时钟中的敏感时间 (in clock time).

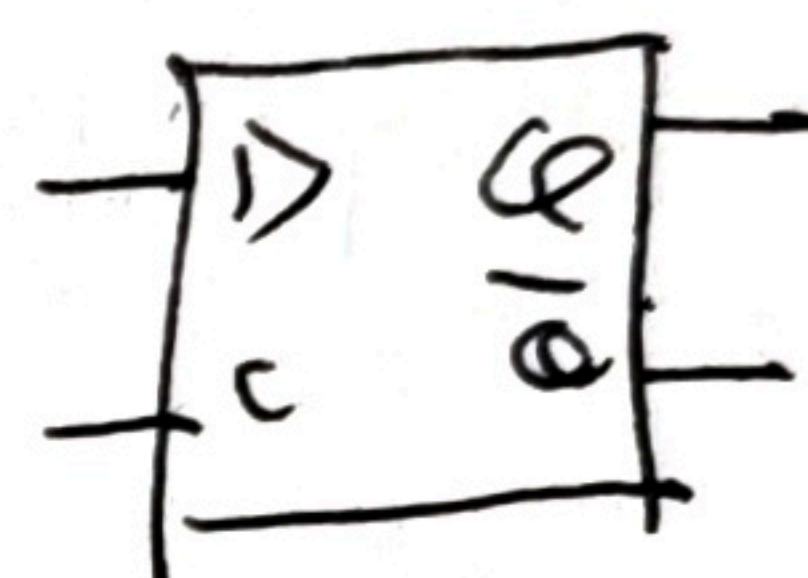
<Latch>

SR → $\bar{S}\bar{R}$ (NAND) → SR with Control. (其内部用低SR) → 使 $S=R$ (禁止D).



当 $S=R=1$ 时 ($\bar{S}=\bar{R}=0$),

二者引出的状态不同, 称为亚稳态。
 存在 indeterminate state/race conditions.
 由 $Q=Q'=0$ 或 $Q=Q'=1$ 时同时得 $S=R=0/1$
 下一状态不可预测。



完全透明的缓冲器

消除了 indeterminate 问题。

<脉冲触发, Master-slave Flip-Flops> 解决

主从 JK SR Latch (门控).

Continually change } 问题.
 Race conditions

为了强制性解决 race conditions 问题, 不适对

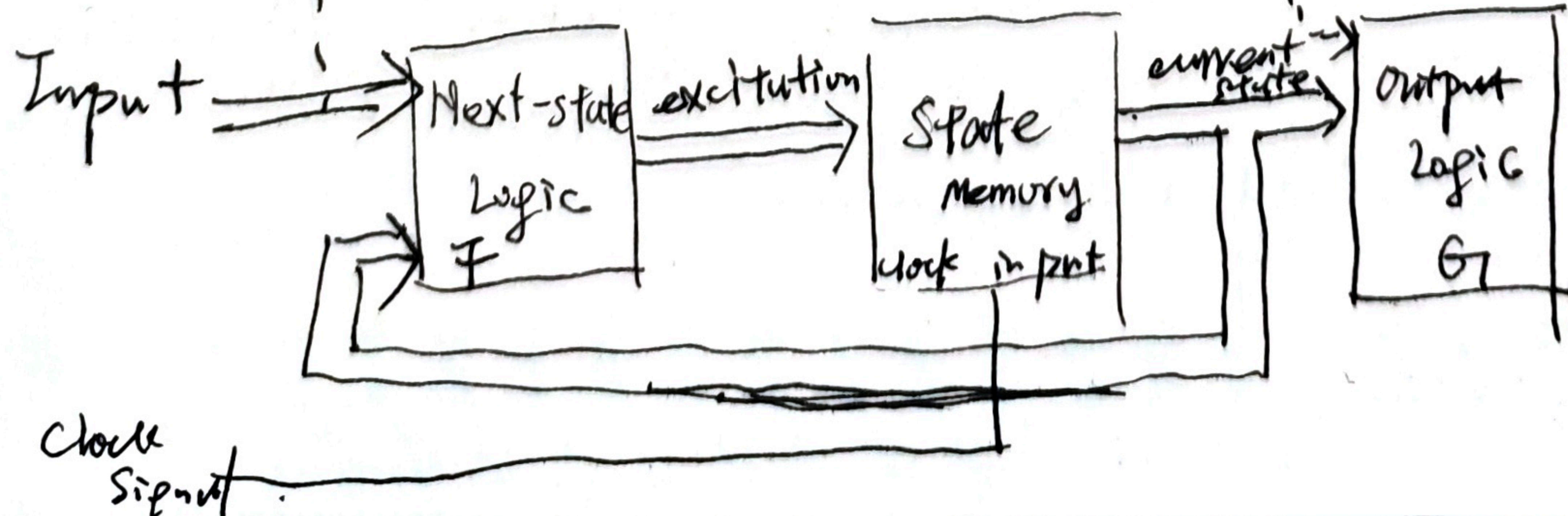
将 Q, Q' 引回输入, 带来了一次亚稳态问题。

每次只有一个输入可能有效, 只能变一次/保持 } 1's Catching (0被1吃)
 "Once toggle" } 0's Catching (1被0吃)

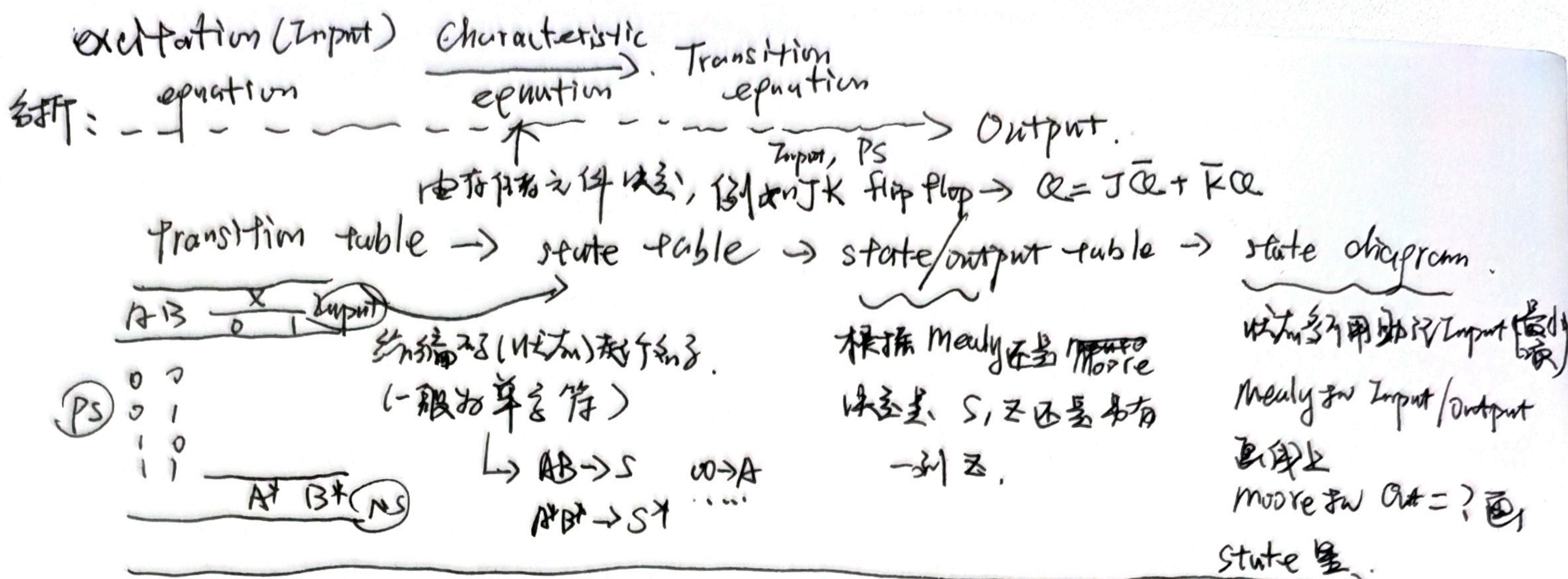
<边沿触发>

D, JK, T (toggle) → 串起来 → register (寄存器).
 使用前一级的 Q 当时钟中的 T 触发 → Counter

3.2 时序电路的分析



More the Output 与 PS 有关.
 Merely the Output 与 PS 和 Input 有关.
 NS 与 Input 和 PS 都有关.



每个表应命名, 编码和状态应明确命名 (切记! Output 和 Input 和 Input (可能) 有歧义!!!)

< Time Delay & Maximum Clock Speed, 时钟频率限制 >

时钟有效边沿沿到之前, 所有数据都应准备完全.

$$T_{ce} + T_{comb} + T_{setup} \leq T_{clk} + T_{skew}$$

flip-flop 从 Q 到 Q 的 delay | setup delay | 时钟 | 时钟偏移.

组合逻辑 delay

在理想时钟系统

$$T_{comb} \leq T_{clk} \Leftrightarrow T_D \leq T_{clk}$$

取频率有: $\frac{1}{T_{clk}} \leq \frac{1}{T_D} \Rightarrow f_{max} \leq \frac{1}{T_D}$

从从 Q 立即再到激励立即的最长延迟, 可确定最大时钟频率, 如果超出该时钟频率, unstable.

3.3 Autonomous Sequential Circuit Design.

$\rightarrow$ no primary inputs.

- ① 状态列表

PS	NS	flip-flop input	output
----	----	-----------------	--------

根据激励表, 并且需要判断需几个 flip-flop. $n = \lceil \log_2 S \rceil$ 状态数.

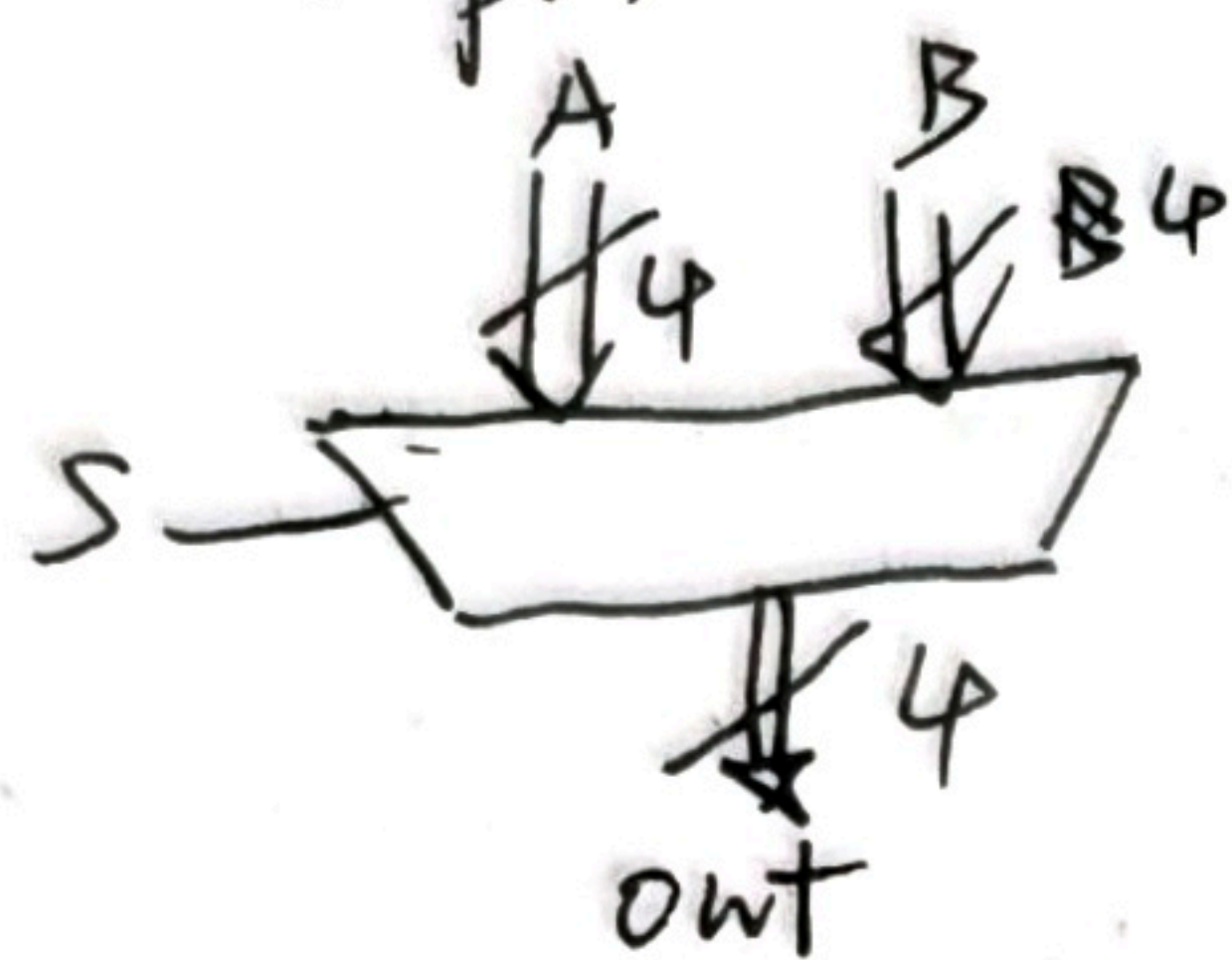
- ② 激励输入 $\rightarrow$ 必须首先使用卡诺图求出激励输入的最简 boolean 表达式.
- ③ 激励输入 $\rightarrow$ 特征方程 状态转移方程 (对于 D flip-flop 为 $Q^+ = D = \dots$).
 $\rightarrow$ 并不复杂, 只求出激励输入 跟步骤 4 合为一体. 已经够设计一个时序电路了.
- ④ 画出状态图 $\rightarrow$ 注意如果有无关项 (那 2^n 个编码未用全, 而我们使用其他 X 无关项), 那么需画出完整 2^n 个状态转移.

<Block 4>

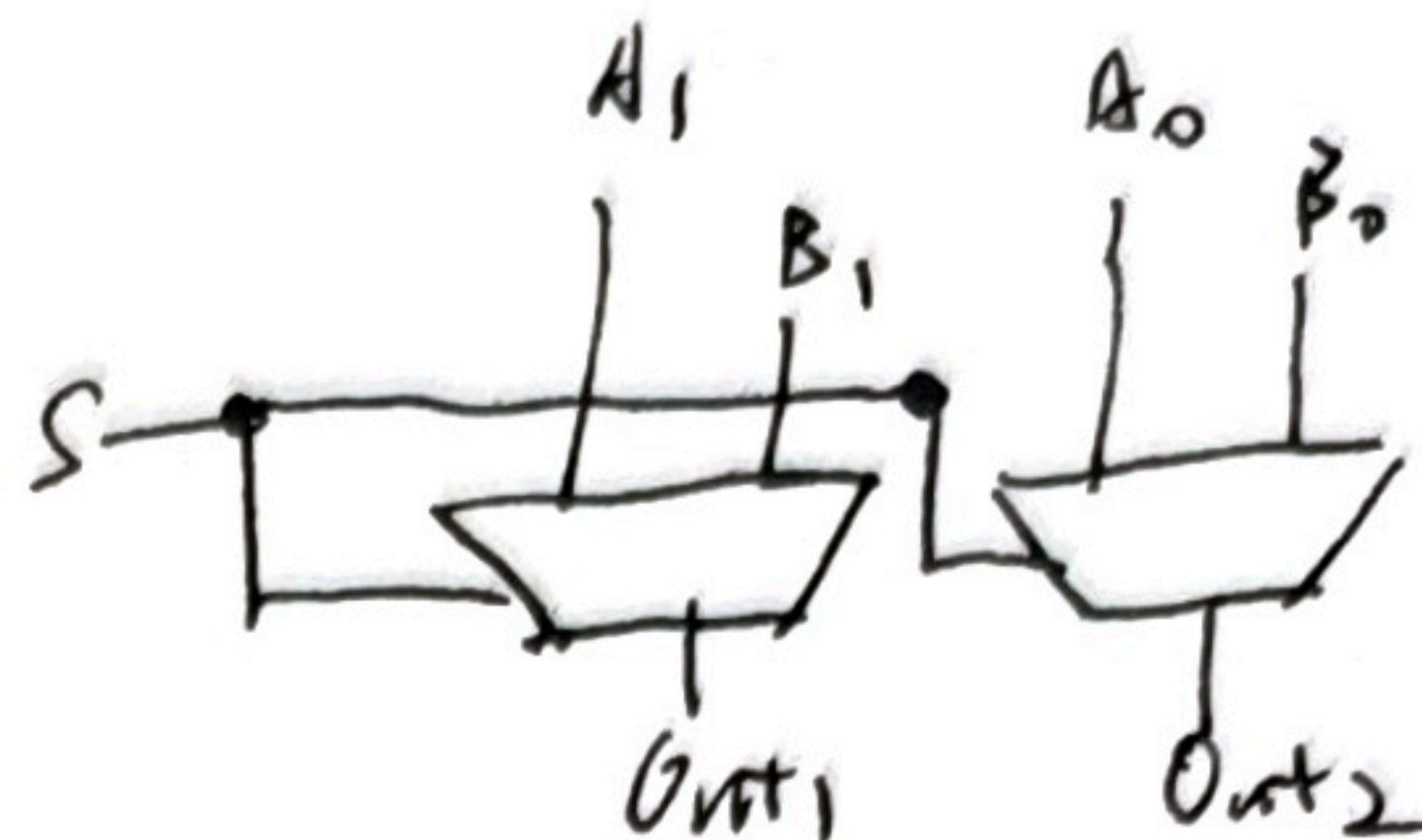
4.1 数字系统部件 Digital System Blocks.

① Buses 总线, more than just one-to-one connection.

② Multiplexers or Selectors



2-input 4-bit MUX



2-to-1 2-bit MUX.

从多输入 (每输入有若干位) 中选择一个作为输出

③ Decoder or Demultiplexer



每次只选中一个 (控制) 对应的输出接口。

④ Magnitude Comparator $\rightarrow$ XNOR (可以通过像 XNOR 和 与门的组合来构造多位比较器).

⑤ Arithmetic Units.

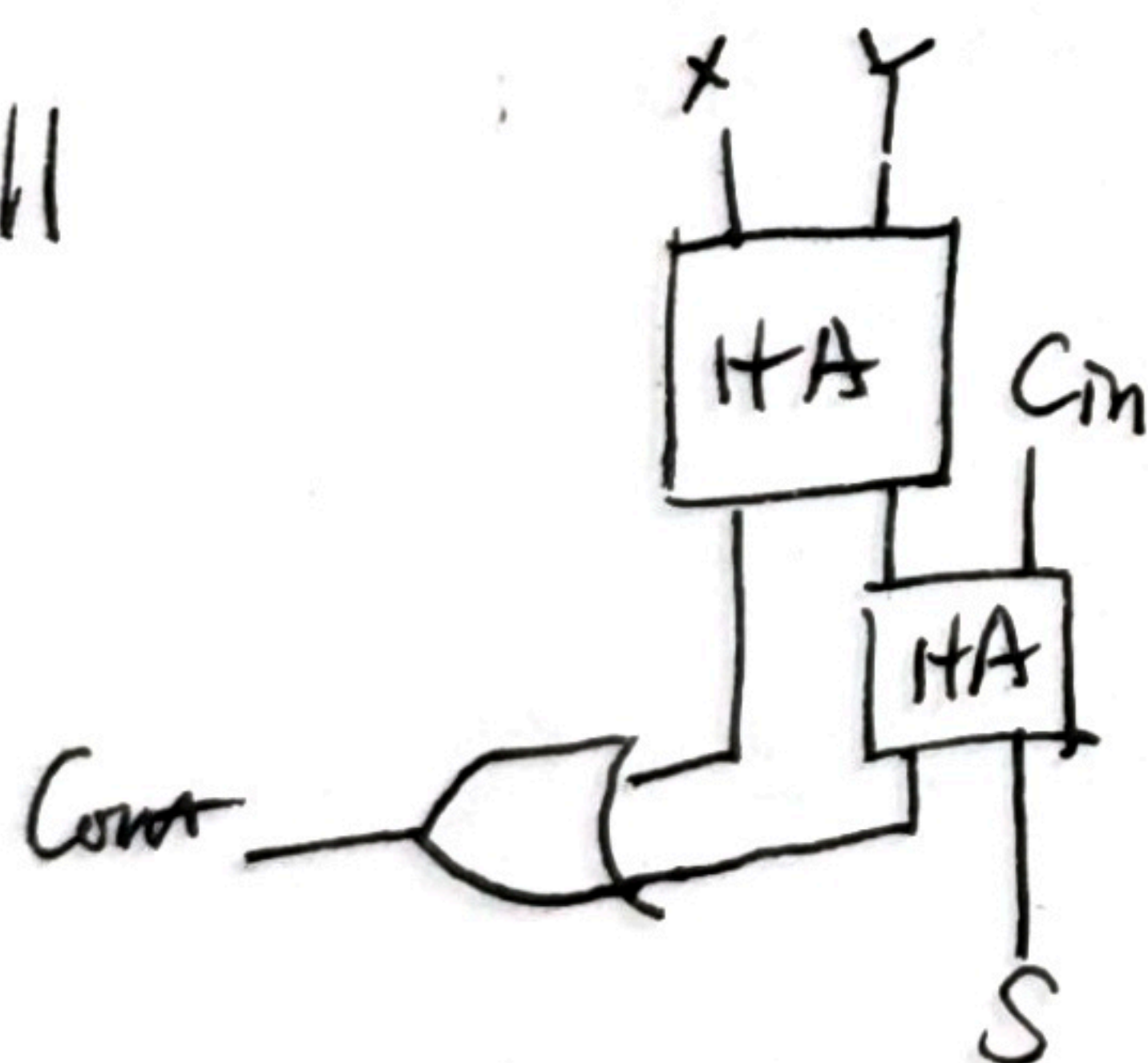
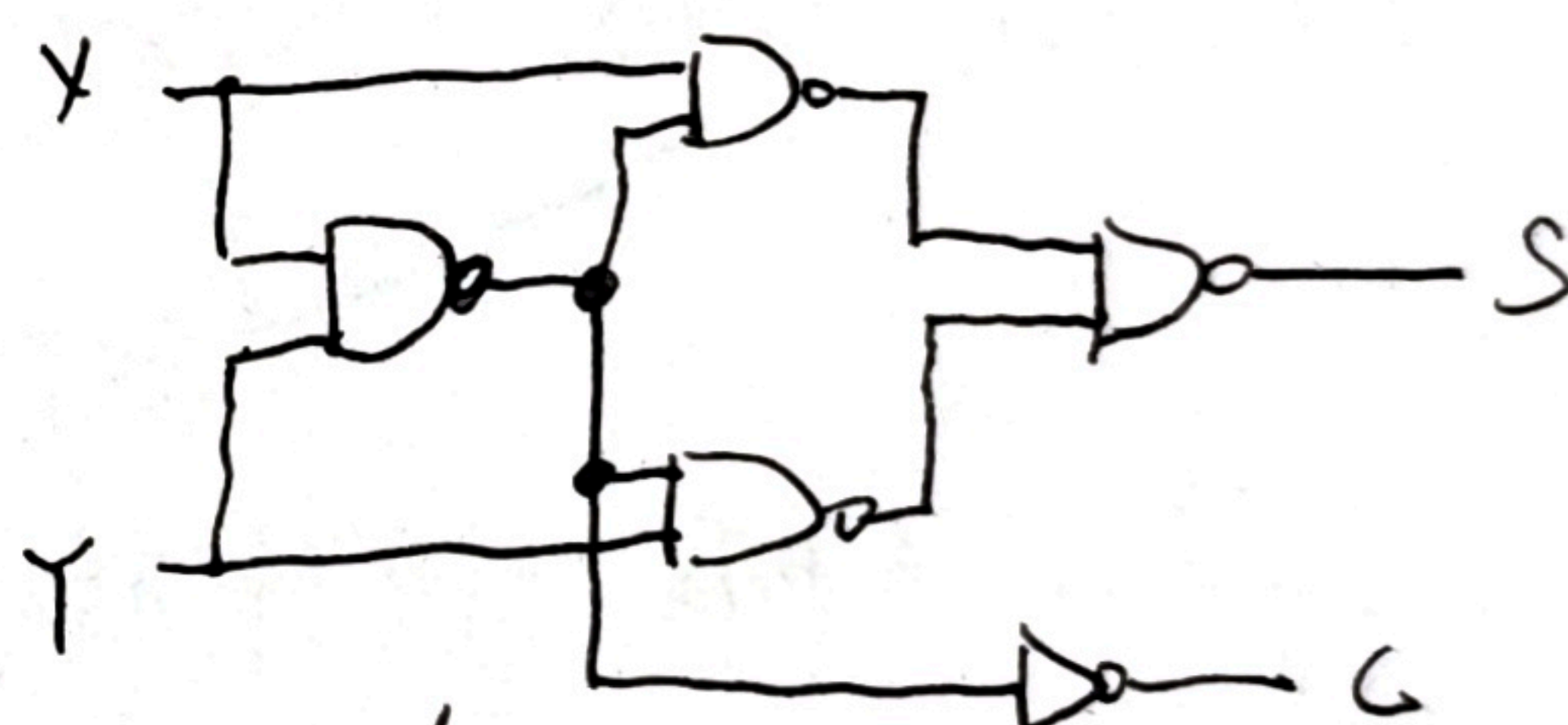
Half
↓
Full



$$S = X \oplus Y$$

$$C = XY$$

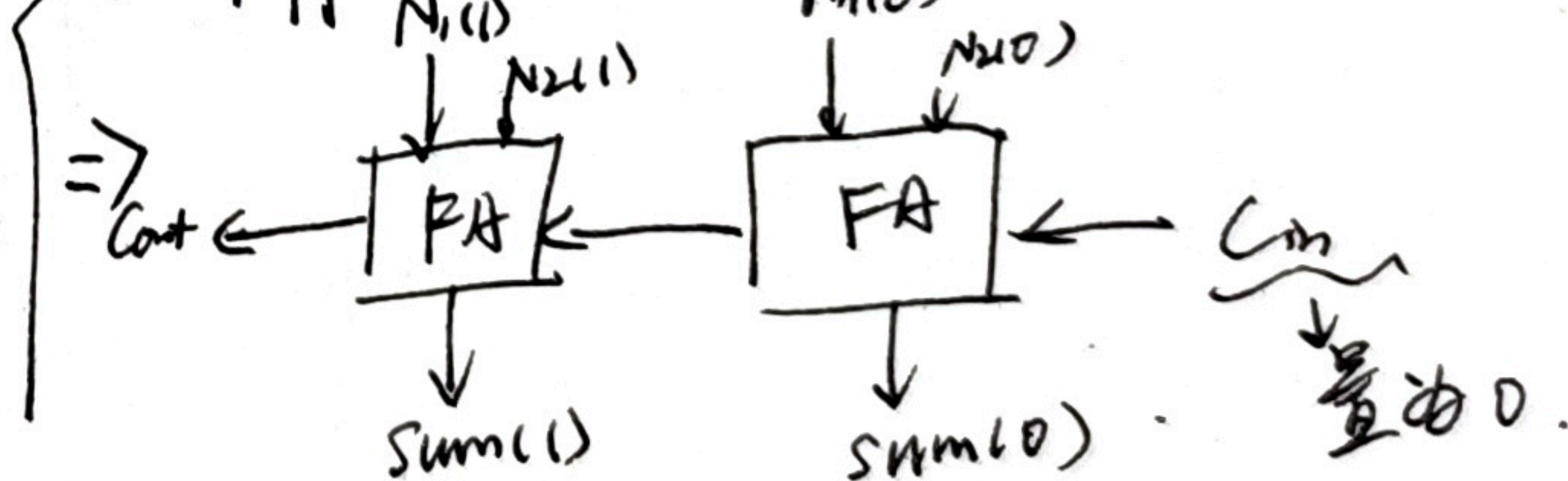
NAND 门



$$S = X \oplus Y \oplus C_{in}$$

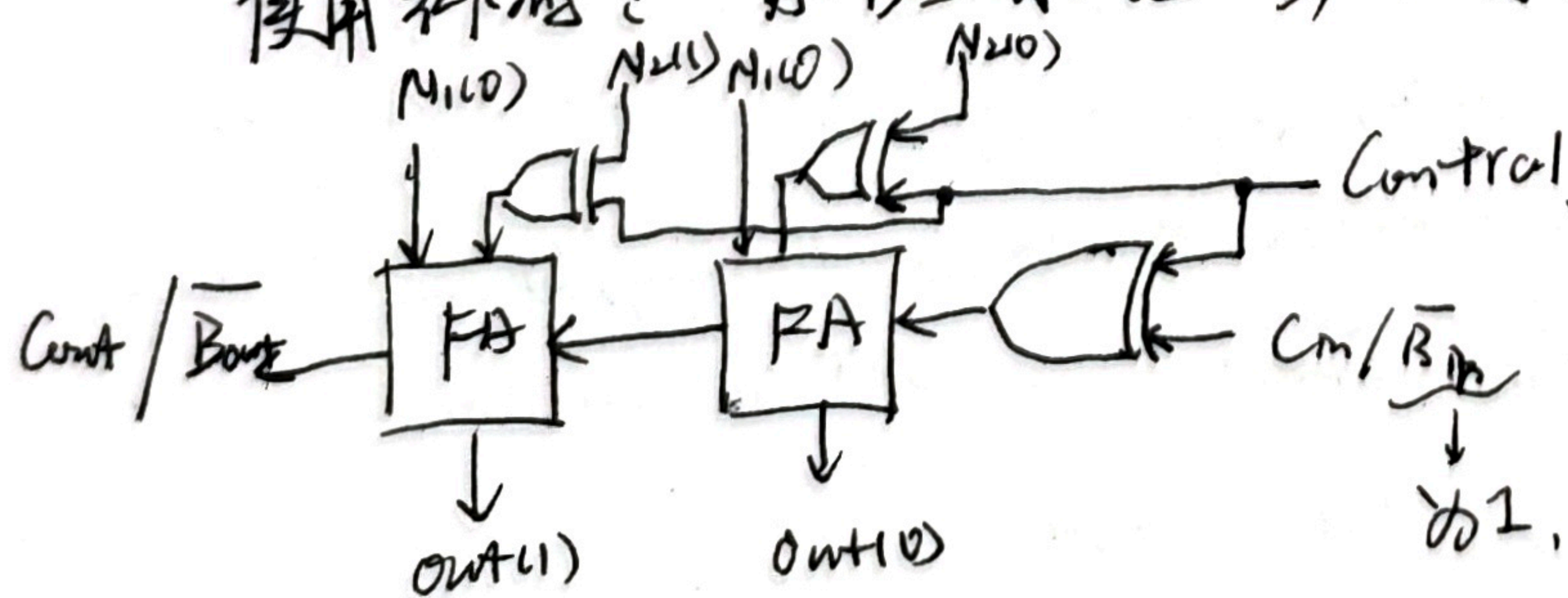
$$C_{out} = XY + (X \oplus Y)C_{in}$$

2-bit Ripple adder

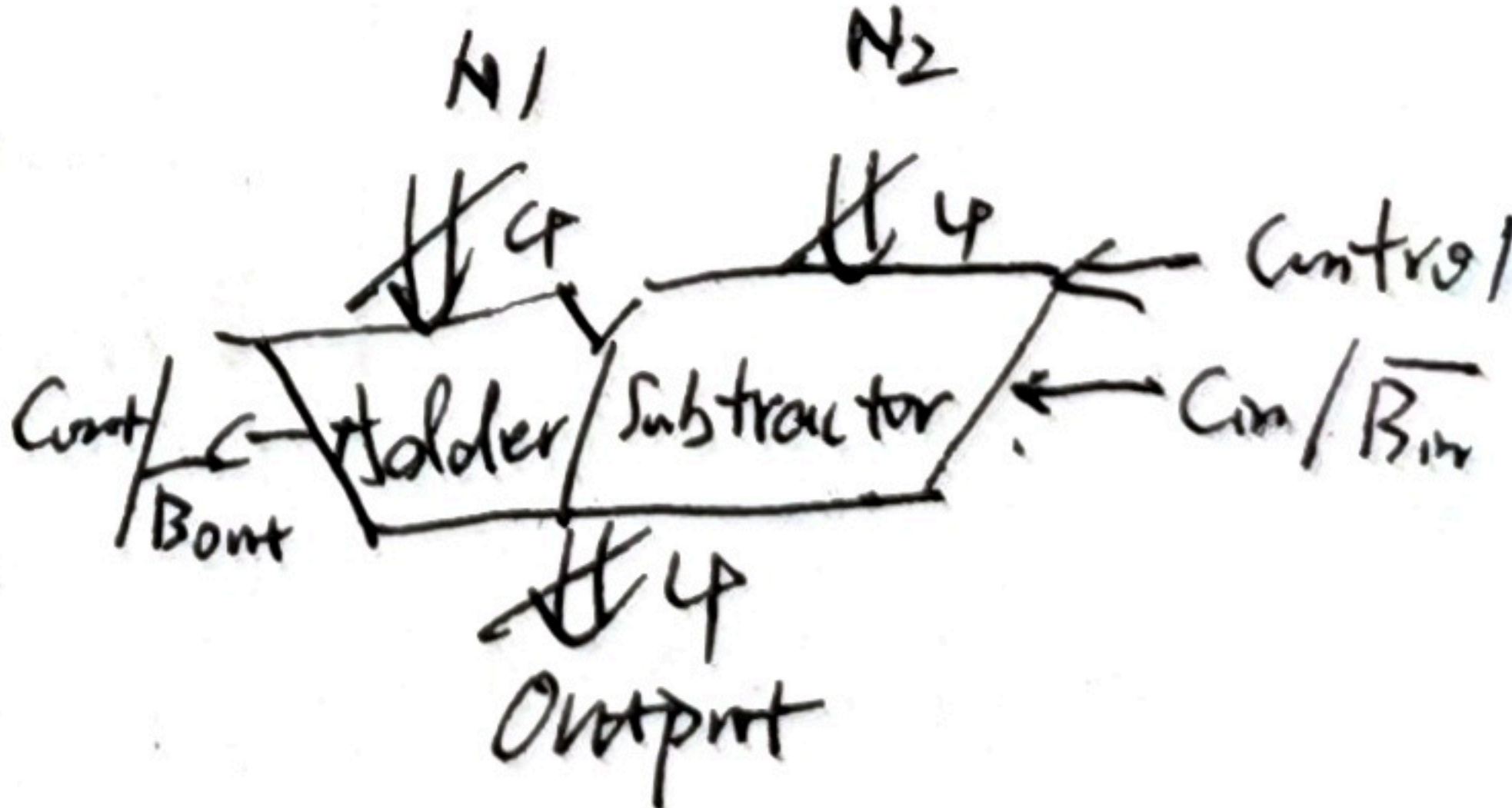


使用补码: $A - B = A + (2^n - B) = A + \overline{B} + 1$

$\rightarrow$ 按位取反



为1时是取反
为0时是取自身

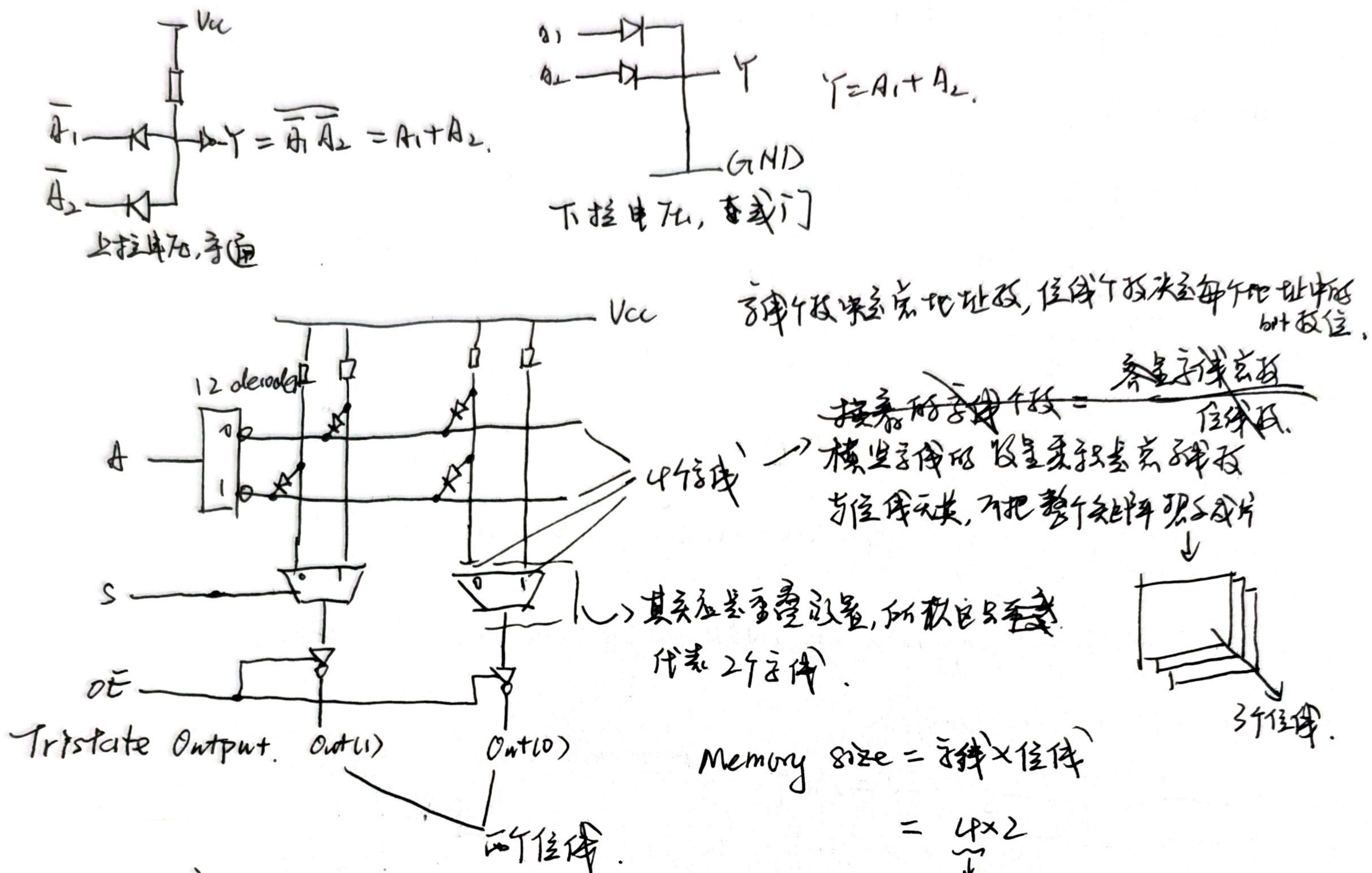


4.2 Memory Devices

① 内存种类

- Serial access
 - in series 串行
 - 需要新位置才能存新数据 $\rightarrow$ time consuming
 - Can store large amounts of data.
- Random access
 - Random $\rightarrow$ Equal time to access any location n-bits address.
 - Volatile $\rightarrow$
 - RAM: Static RAM = transistors to store
 - Dynamic RAM: capacitor
 - Non-volatile $\rightarrow$
 - ROM (Read-Only Memory)
 - periodically refreshed

② ROM (双极型-静态型)



每个字节求它的地址码, 位域个字节至每个地址中的
bit 数

$$\text{接着的数字个数} = \frac{\text{数字位数} - \text{位数}}{1}$$

横竖线条的 改革是积极主动的
与位线无关, 不把整个知识照搬成片

其类应甚多，所拟仅以 ~~三~~ 代表 2 个年代。

$$\text{Memory size} = \text{字长} \times \text{位数}$$
$$= \underbrace{4p \times 2}_{\downarrow}$$

$4=2 \times 2 \rightarrow$ 二维阵列

注意, 两次取反, 有四种形式 $\rightarrow$ 交叉点有元件即存储 1. $4=2 \times 2 \rightarrow$ 二维阵列.
交叉点无元件即存储 0.

↳ 根据这些, 便可以由地址一值表设计 PDM.

② ROM的种类

mask ROM, programmed in manufacture

PROM, Programmable ROM, 只读存储器。

* PROM, } Erasable Programmable Read-Only Memory, It can be programmed
only ~~once~~ and ~~cannot~~ ^{can} be erased afterwards.

不列颠岛 $210m$

EEPROM: Electrically Erasable PROM

研部、更改排除。