Digital Logic & Turing Machine

① Arith : + ≡ (−, ×, ÷) ← 仅支加 （此处三代表等价）

② + ≡ AND . OR . NOT

③ Logical AND / OR / NOT can do everything

Rmk of ②:

```
   a3 a2 a1 a0
    0  0  1  0     2
+  b3 b2 b1 b0
    0  1  1  1     7
----------------
    1  0  0  1     9
   C3 C2 C1 C0
```

$C_0 = a_0 + b_0 \equiv a_0$ XOR b_0 （eXclusive OR）

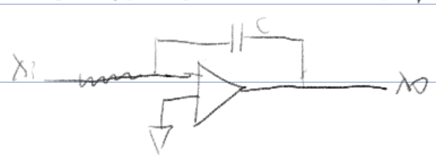
$C_1 = a_1 + b_1 \equiv a_1$ XOR b_1

$Carry_1 = 1 \equiv a_1$ AND b_1

数字逻辑 V.S 模拟计算 in 1940s ?

- 模拟计算 可行性 在1940s 已被验证 （美国雷达）

  （物理世界天生是非线性的）



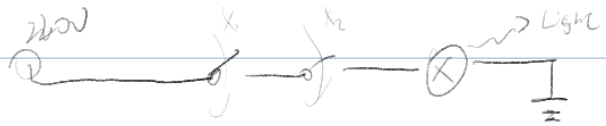
- 冯·诺依曼 结构 纯数字化
- 工程师上场：

  Logical AND、OR、NOT

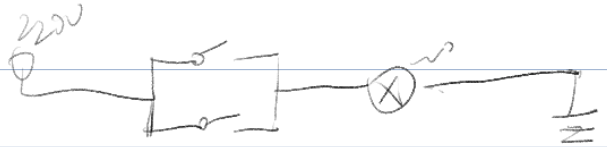
机械：

均透光才能过 AND 门

回顾：



AND

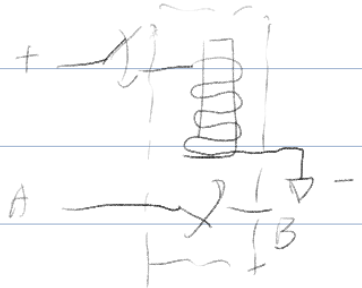


OR

（应可做状态选择）

然而电子系统不能承担机械力

Relay（继电器）

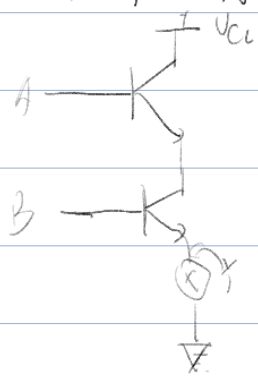


（AB有 mechanical）

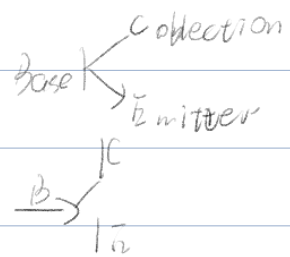
纯电子：电阻、电感、电容、二极管

怎么用其搭建 AND、OR、NOT？

① AND. $Y = A \cdot B$

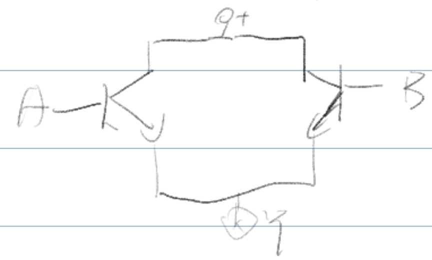


三极管（做成 gate）

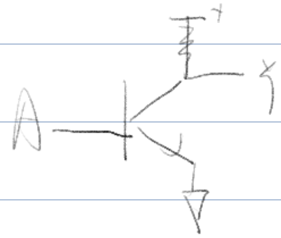


（内包含 90% 不是 gate）

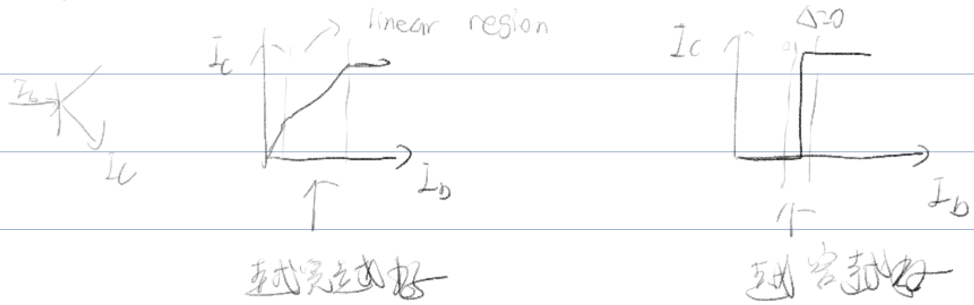
② OR: $Y = A + B$



③ NOT: $Y = \bar{A}$



But. In 6R: In CS:



Mem Tech (黑板看可理 Mem Tech)

stateful logic

Volatile (C)

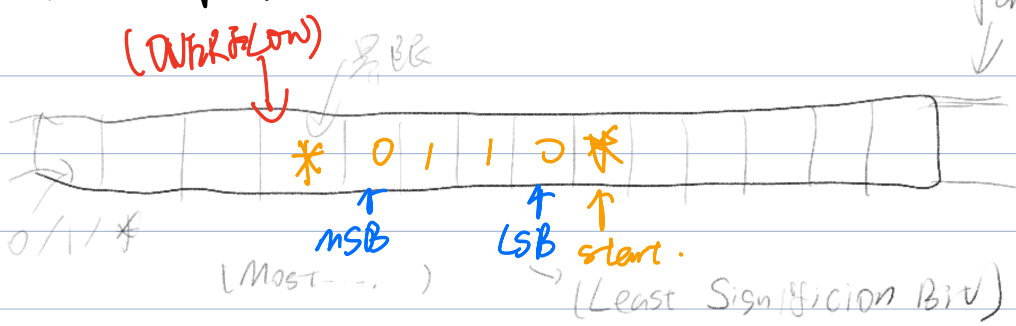
Volatile int x;
→ unic address

小 size
磁鼓记录 (不是写在黑板上写, 用一次就没了)

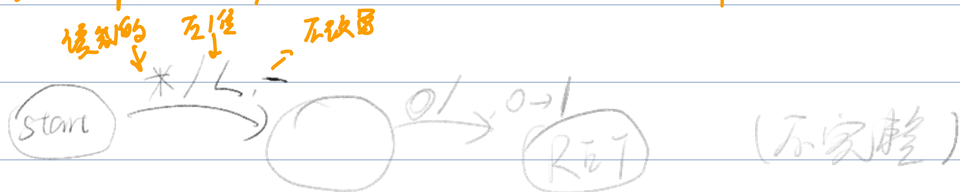
Turing Machine. ($T_m = \{ \Sigma_{0,1,*}, K_{state}, s, b, b \}$)

其一开始发现是解决数学问题
只是后来被广泛应用于计算机

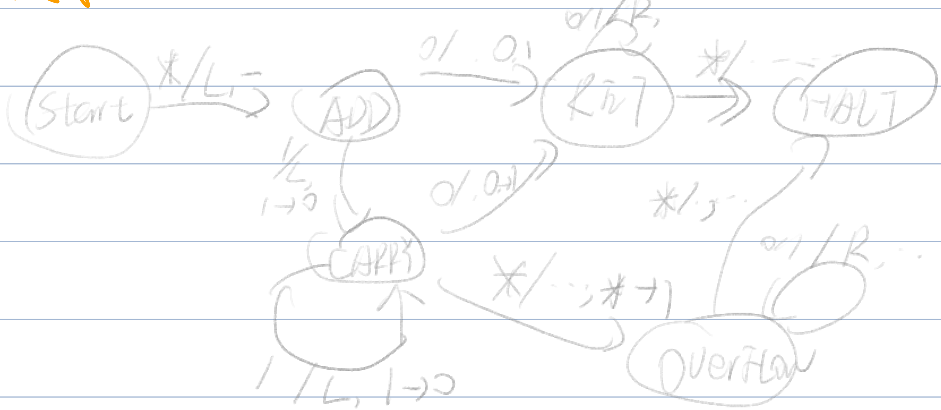
图灵机是最简单的操作



Example. X = 0110   Y = X+1



换成 X = 0111   * 0 1 1 1 *



图灵机图 (CARD)

Card state	input (0, 1, *)	Write (0→1, 1→0, 0,1→*)	shift (L, R, or -)
ADD	0		
	1		
	*		
RET			
CARRY			
HALT			
OVERFLOW			

Conclusion

- Digital Logic: 加法可以实现任何运算
  - \+ 加法可被 AND、OR、NOT 实现
  - => 逻辑门可实现任何运算
  - 最终运用三极管（开关效应）
- Turing Machine: 输入 → 输出
  - $T_m: \{ \Sigma_{0,1,*}, K_{state}, s, h, b \}$
  - 固有 (Σ, K)　起始 (s)　终止 (h)　转移 (b)

Why Digital Logic?

冯·诺依曼：相信技术成熟后成本会下来

（芯片一片 35 ¥ → 0.3 ¥）