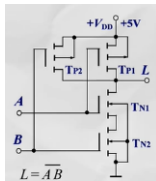
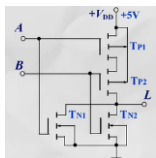


## 知识点：

1. 同样是输入端  $10k\Omega$  接地，TTL 算 1，CMOS 算 0。CMOS 不可悬空，TTL 悬空算 1。
2. 三种基本逻辑运算：与或非
3. 早期 TTL 技术成熟，但功耗比较大，只用于中小规模电路
4. 54/74HC00，54 比 74 温度使用范围更宽，筛选检测更严格，HC 表示系列，00 表示不同逻辑功能芯片的编号
5. MOS 反相器输出上升下降沿缓慢的原因是结电容的充放电
6. TTL 静态功耗是主要的，CMOS 动态功耗是主要的。
7. CMOS
  - a) 静态功耗几乎为零
  - b) 开关速度快 (RC 的 R 小)，带负载能力强 (输出电阻小)
  - c) 扇入数也大 (输入电阻大)
  - d) 电路互补对称  $\rightarrow t_{pHL} = t_{pLH}$
  - e) 总功耗  $P_D = (C_{PD} + C_L)V_{DD}^2 f$ ,  $C_{PD}$  是芯片参数,  $C_L$  是负载电容
  - f) 扇出数受到拉电流负载限制，拉电流会拉低高电平；受到灌电流负载限制，灌电流会提高低电平。扇出数取两者中数值小的作为扇出数。
  - g) 延时功耗积  $DP = t_{pd}P_D$  小的芯片性能高
8. CMOS 逻辑门：



- a) 这是二输入与非。N 输入与非则是串 N 个 N 型，并 N 个 P 型，由于 N 型电阻，即使低电平，L 处电压也会水涨船高。



- b) 这是二输入或非。N 输入或非并 N 个 N，串 N 个 P。逻辑 1 受影响

- c) 传输门，控制信号 1 导通，0 截止

- d) 输入保护和缓冲电路


- i. 为何要保护？—— $R_i$  大，容易感应静电，产生大电流
- ii. 如何保护？
  1. 两二极管限制电压
  2. RC 积分网络使过冲电压延迟衰减
  3. 加个反相器缓冲

- e) 输出保护缓冲电路

- i. 两二极管限制电压
- ii. 加反相器缓冲


- f) 输入输出位置加了反相器缓冲，会使该电路逻辑功能发生变化

- g) 实现线与

- i. 漏极开路门 (OD) , 要外接上拉电阻  $R_p$ ，有  $R_{pmin}, R_{pmax}$ ，若想要工作速度快选  $R_p$  小 ( $\tau$  小)，若想功耗低，选  $R_p$  大。带负载能力下降 (输出电阻

大)

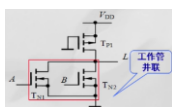
h) 噪声容限:  $V_{NH} = V_{OH(min)} - V_{IH(min)}$ ;  $V_{NL} = V_{IL(max)} - V_{OL(max)}$

9. 三态输出门  (TSL), 主要用于数据总线传输

10. 类 NMOS 门电路

a) NMOS 比 PMOS 好, N 载流子是电子, 迁移率比空穴高。生产工艺优, 使用正电源, 兼容性好。

b) CMOS 芯片面积较大 (管子多)



c) 或非门, 增加输入端个数不会影响电平电压。因此类 NMOS 电路多以或非门为基础搭建电路。

d) 之所以叫类 NMOS, 是因为把 N 负载管换成了 P 负载管。(但工作管还是 N)

11. BiCMOS 门电路采用 BJT 作为输出级, 驱动力强, 功耗低, 速度快, 集成度高

12. RAM—Random-Access Memory 随机是指可直接存取任意位置存储单元数据且访问时间与位置无关

13. ROM—Read-Only Memory。分固定 ROM (只能由制造商写入) 和可编程 ROM。

14. 可编程 ROM

a) PROM 仅编程一次 (熔丝)

b) EPROM 紫外光擦除。(SIMOS, 浮栅, 雪崩注入, 紫外光擦除)

c) EEPROM 电擦除, 即在线编程 (FLOTOX 管, 选通管, 电注入擦除, 写入前要擦除, 集成度低)

d) FLASH 比 EEPROM 更快写入速度, 更高密度 (快闪叠栅 MOS 管, 无选通管, 写入前要擦除, 擦除速度快)

15. RAM

a) SRAM

i. 存储单元是双稳态电路

ii. 电路复杂, 集成度受到限制

iii. 同步 SRAM—时钟脉冲控制下完成读写

b) DRAM

i. 存储单元是电容

ii. 需要定时刷新

16. FIFO 存储器 队列

17. 双口存储器, 当同时访问地址单元, 内部仲裁电路根据微小时差决定哪个端口先访问

18. 半导体存储器的容量扩展

a) 位扩展: RAM 地址线并联, 输出线分开

b) 字扩展: 译码器片选, 地址线分开, 输出线并联

19. PLD 可编程逻辑器件

a) SPLD (simple PLD)

i. PLA 可编程逻辑阵列——与或阵列都可编程

ii. PAL 可编程阵列逻辑——与可编程, 或固定

iii. GAL 通用型阵列逻辑——与可编程, 或固定 (优化了输出结构)

iv. PROM, EPROM——与固定, 或可编程

b) CPLD (complex PLD)

- i. 编程机理基于 EEPROM, 快闪存储器
  - ii. 逻辑单元是可编程与或阵列
  - iii. 集成度低
- c) FPGA
  - i. 编程机理基于 SRAM, 外部用 PROM 保存编程数据 (每次开启时要初始化)
  - ii. 逻辑单元是查找表
  - iii. 集成度高, 能完成比较复杂的算法
- 20. 单稳态触发器
  - a) 分可重复触发和不可重复触发
  - b) 应用: 定时, 延时, 延时采样——噪声消除
  - c)  $t_w = 0.7RC$
- 21. 施密特触发器
  - a) 同频率波形变换, 输出信号占空比与触发器中电阻比值相关
  - b)  $V_{T+}, V_{T-}$  的差值称为回差电压
  - c) 用于波形变换, 波形整形, 消除干扰, 幅度鉴别
- 22. 多谐振荡器

## a) 反馈延时环节和开关电路组成

- i. 门电路充当开关电路:  $T=1.4RC$
- 23.

$$=RC \ln \frac{V_{DD}-V_{T-}}{V_{DD}-V_{T+}} \cdot \frac{V_{T+}}{V_{T-}}$$

- i. 比较器充当开关电路:
  - b) 双相脉冲产生电路中用到石英晶体和 JK 触发器
- 24. D/A 转换器

$$V_O = -\frac{V_{REF}}{2^n} \cdot \frac{R_f}{R} \left[ \sum_{i=0}^{n-1} (D_i \cdot 2^i) \right]$$

a)

以8位D/A为例:

$$\text{单极性: } \pm 0 \sim \pm V_{REF} \frac{255}{256}$$

b)

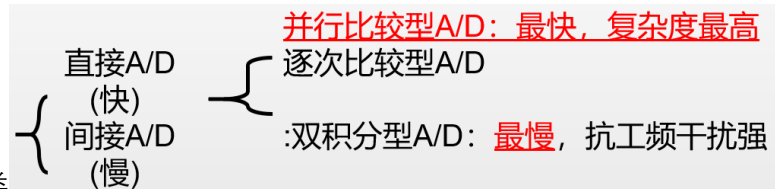
$$\text{双极性}(-1/2V_{REF}): -\frac{128}{256}V_{REF} \sim \frac{127}{256}V_{REF}$$

? ? ?

$$\text{分辨率: } \frac{1}{2^n - 1}$$

- c)
  - d) 转换精度: 比例误差 (斜率) 失调误差 (零漂) 非线性误差
- 25. A/D 转换器
  - a) 过程

- i. 取样:  $f_s \geq 2f_{imax}$ , 一般工程上 3~5 倍
- ii. 保持
- iii. 量化,  $\epsilon$  属于原理误差无法消除  
(4 位码为例)
  - 1. 舍尾取整时, 门限设置为  $1/16, 2/16, 3/16, \dots, 15/16$ . 最大误差一个  $\Delta$
  - 2. 四舍五入时, 门限设置为  $1/31, 3/31, 5/31, \dots, 30/31$ . 最大误差  $1/2\Delta$
- iv. 编码



- b) 分类
  - c) 分辨率: 最大电压/ $2^n$
  - d) 并行比较型 A/D 要  $2^n - 1$  个比较器和触发器 (触发器的输出不是输出, 最后有个优先编码器)
  - e)  $n$  位逐次比较器每次准换至少需要  $n$  个时钟周期
26. 小数的补码反码, 先忽略小数点, 进行操作之后早把小数点补上。
27. 补码  $\rightarrow$  原码    数值位减一取反
28. 原码  $\rightarrow$  补码    数值位取反加一
29. 正数和负数的补码相加相当于两正数相减
30. 余三码是 8421BCD+3
31. 余三循环码是格雷码首尾各去掉 3 个数
32. 二进制码转格雷码自身相加, 比较好看, 格雷码转二进制码加尾巴

◎  $n$  位带符号二进制数范围:

原码:  $-(2^{n-1}-1) \sim +(2^{n-1}-1)$

反码:  $-(2^{n-1}-1) \sim +(2^{n-1}-1)$

补码:  $-2^{n-1} \sim +(2^{n-1}-1)$

- 33.
- 34. 2421, 5421 前五个数最高位为 0, 后五个数最高位为 1

```

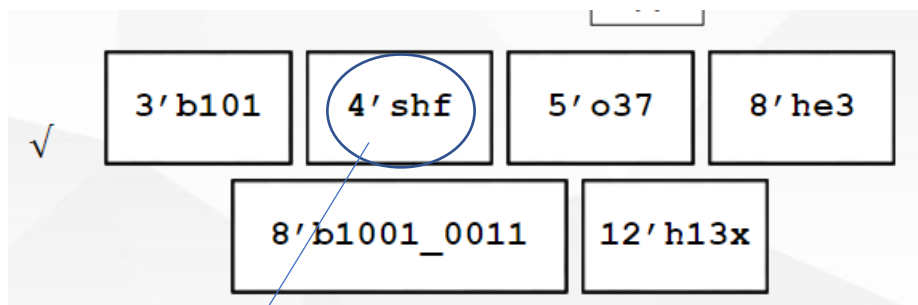
module Updowncount
#( parameter n = 4 )
( input Load, Up_down, En, CP,
  input [n-1:0] D,
  output reg [n-1:0] Q
);
  integer direction;

  always @(posedge CP)
  begin
    if (Up_down)
      direction <= 1;
    else
      direction <= -1;

    if (Load)
      Q <= D; //同步置数
    else if (En)
      Q <= Q + direction;
    else
      Q <= Q; //输出保持不变
  end
endmodule

```

一定得是 integer 吗, reg 行不行



四位有符号数能表示 15? ?