

《数字逻辑》 Digital Logic

三 门电路

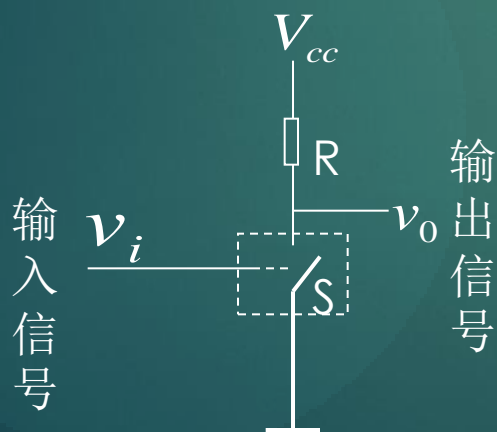
北京工业大学软件学院
王晓懿

数字逻辑的物理实现

- ▶ 门电路
- ▶ 门电路的电气特性
- ▶ 集成电路

正逻辑与负逻辑

- ▶ 在逻辑电路中，常把电平的高、低和逻辑0、1联系起来，若 $H=1, L=0$ ，称正逻辑；若 $H=0, L=1$ ，称负逻辑。
- ▶ 在本课程中，一律采用正逻辑。

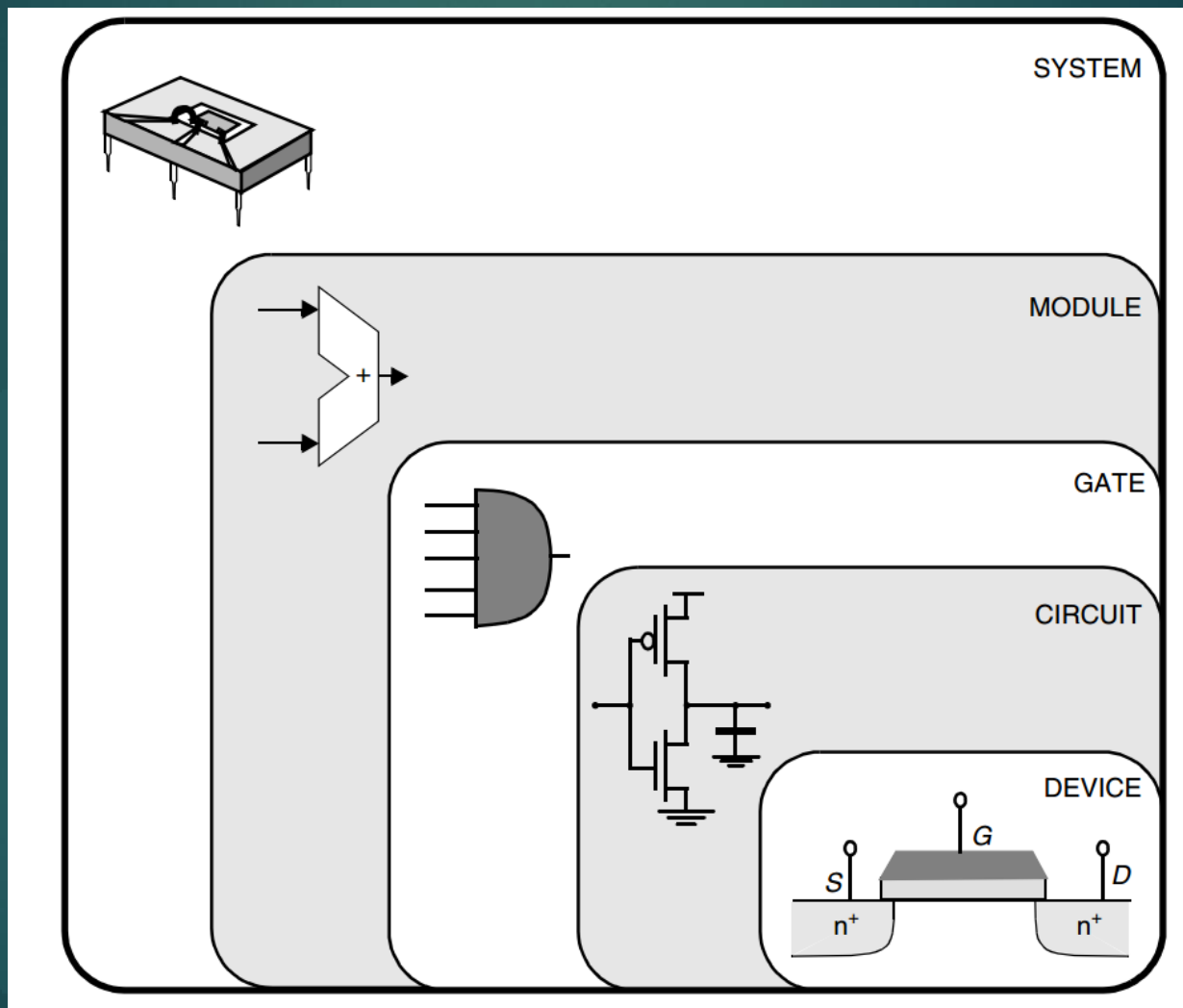


正逻辑

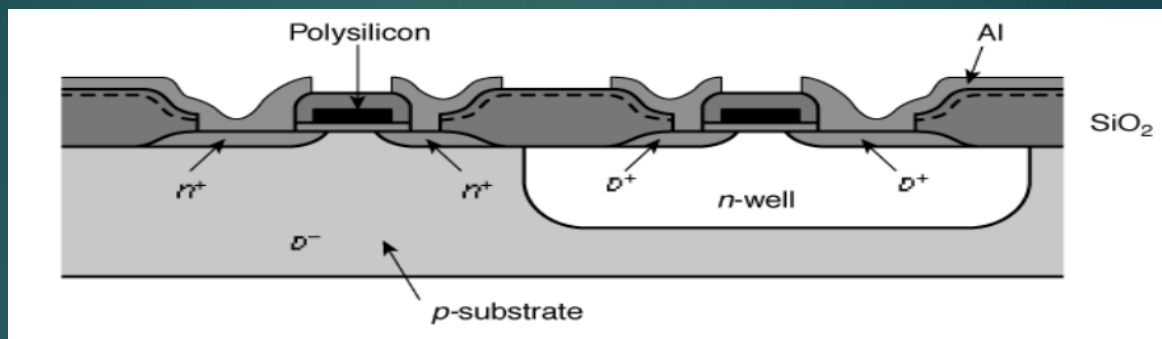


负逻辑

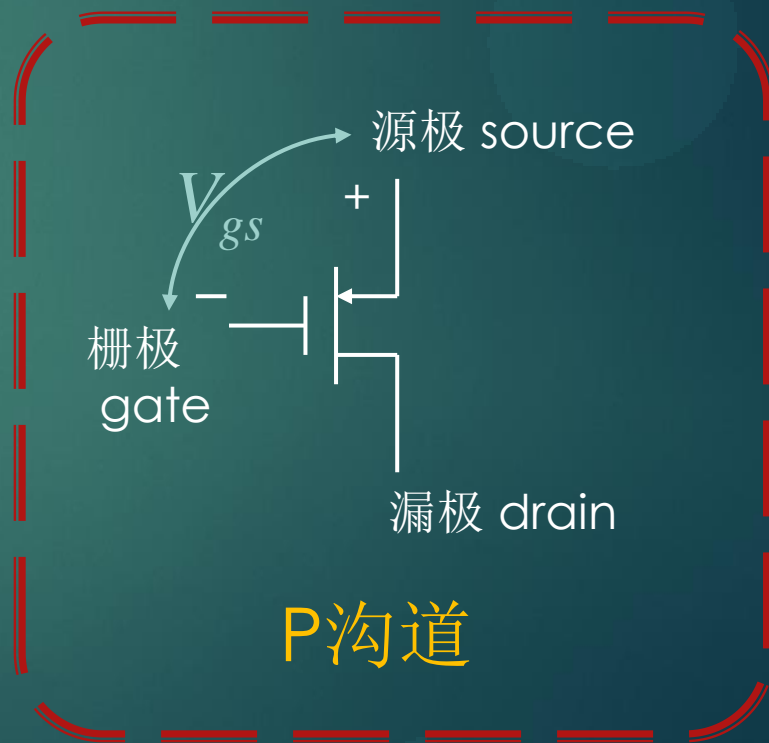
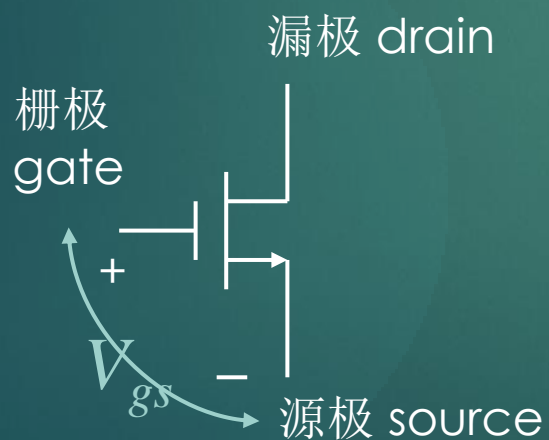
数字电路的设计与实现



- ▶ 逻辑系列：TTL系列 和 CMOS系列
- ▶ CMOS逻辑电平

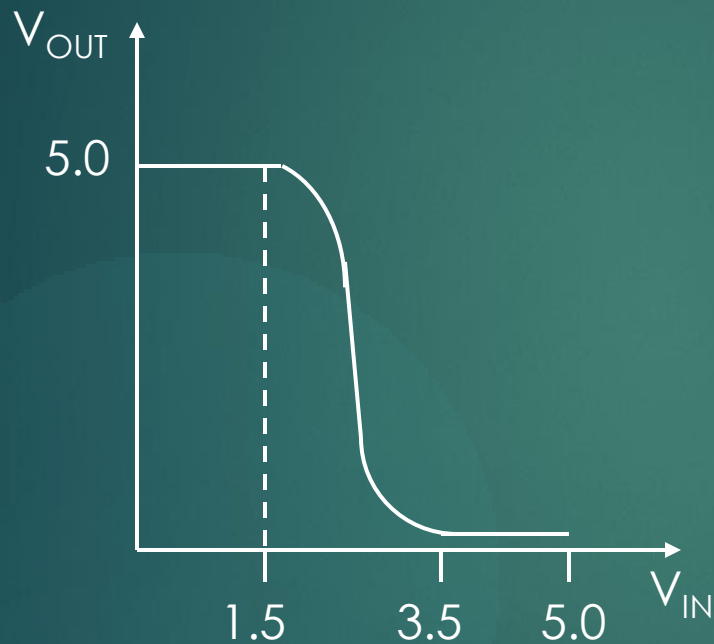


N沟道

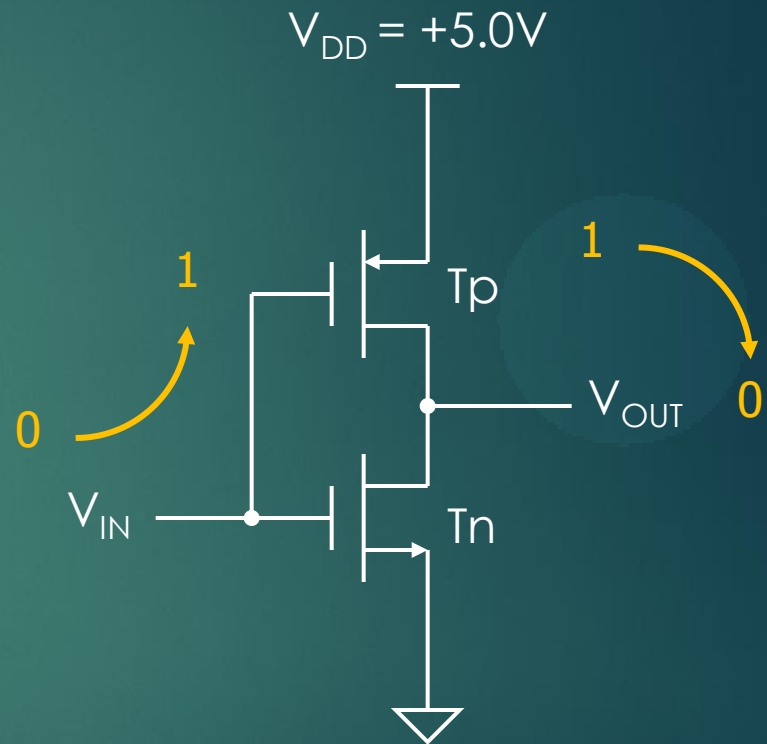


P沟道

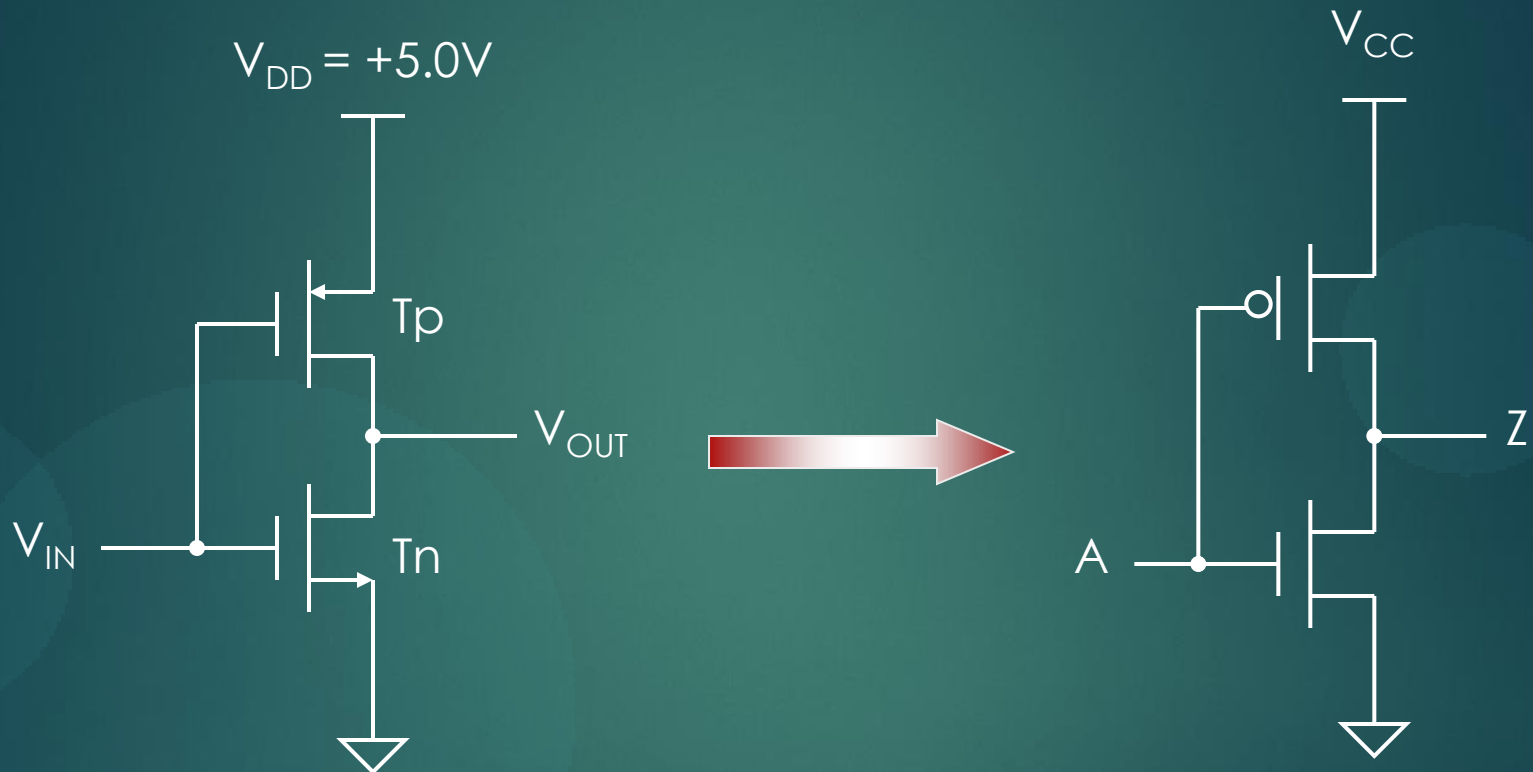
CMOS的输入输出关系



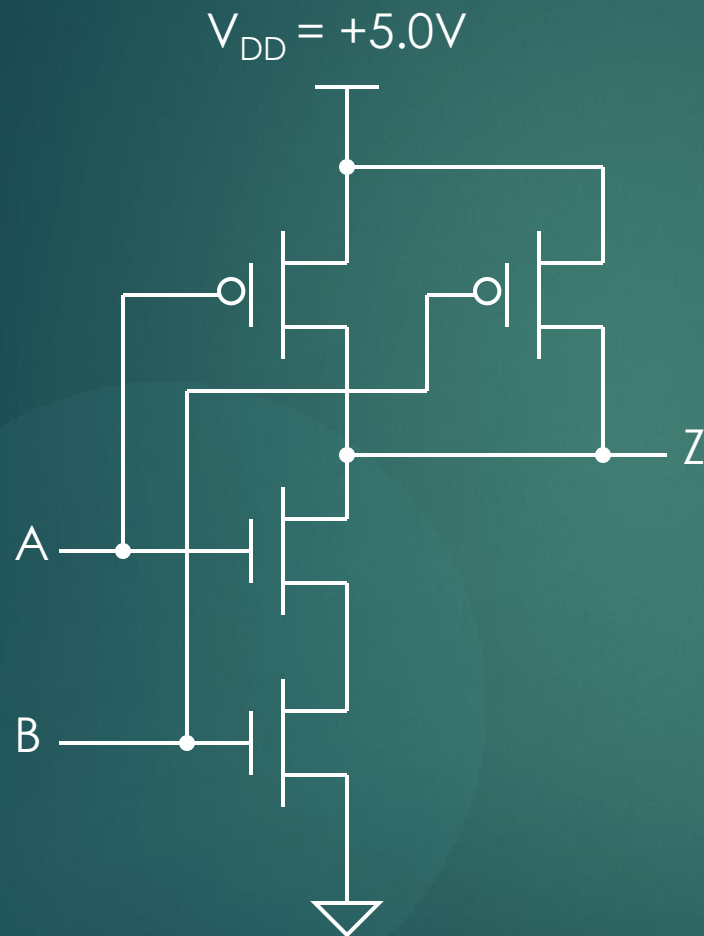
电压传输特性



CMOS反相器

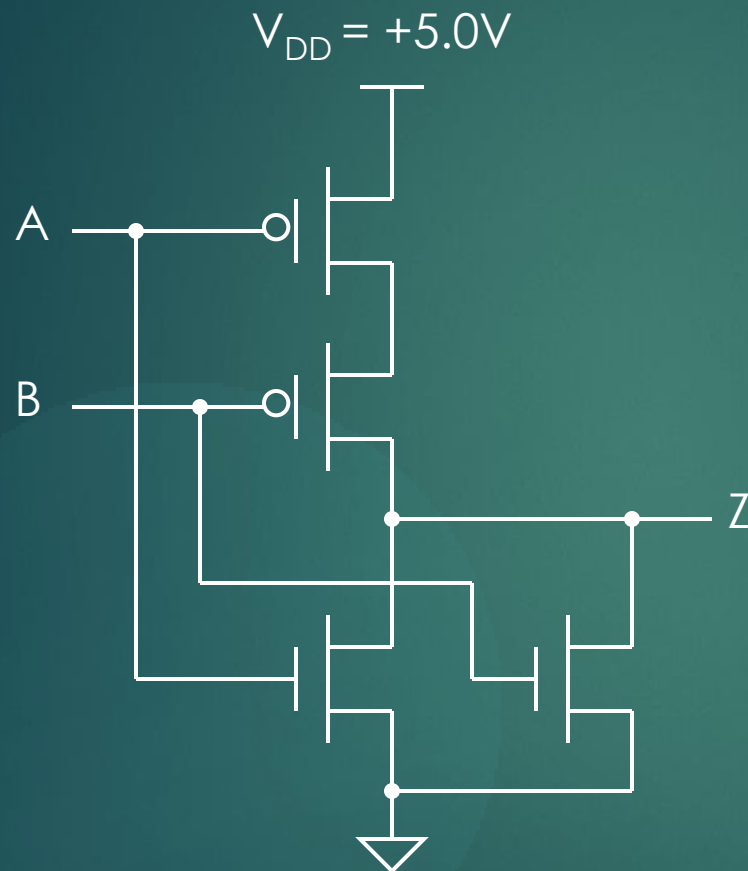


CMOS与非门



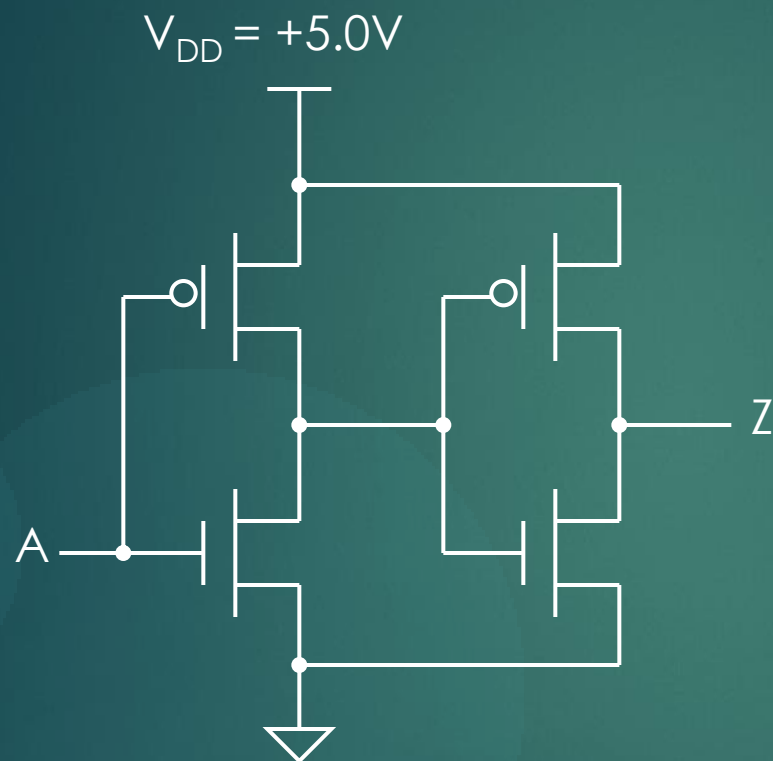
A	B	Z
0	0	1
1	0	1
0	1	1
1	1	0

CMOS或非门

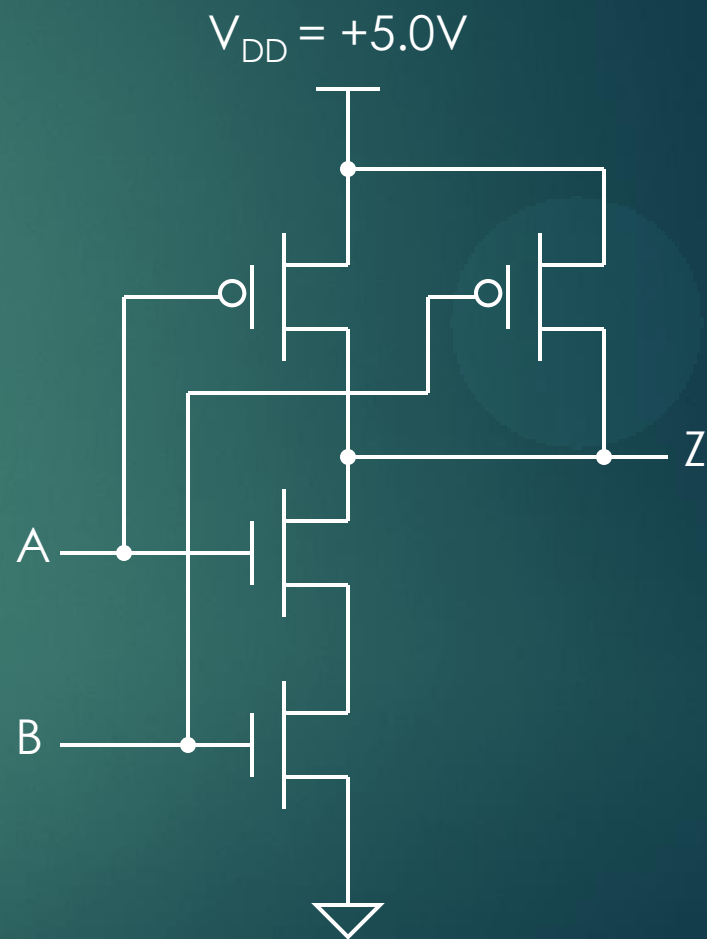
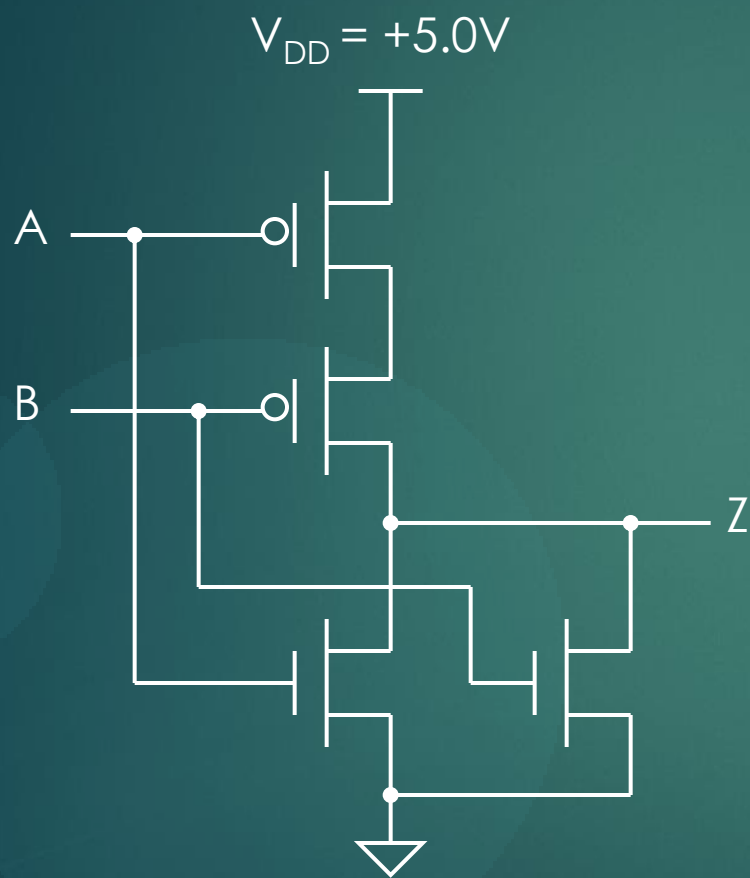


A	B	Z
0	0	1
1	0	0
0	1	0
1	1	0

非反相门



- 串联晶体管导通电阻的可加性限制了CMOS门的**扇入数**



CMOS电路的电气特性

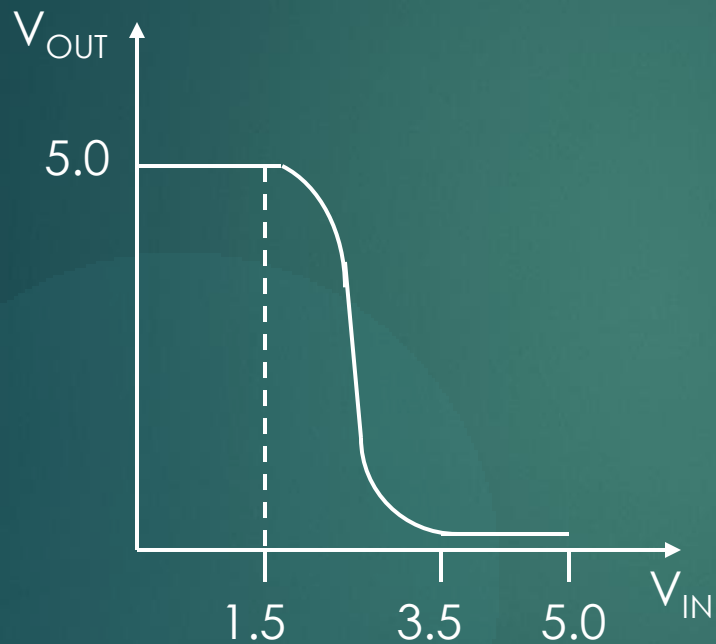
- ▶ 逻辑电压电平
- ▶ 直流噪声容限
- ▶ 扇出
- ▶ 速度、功耗
- ▶ 噪声、静电放电
- ▶ 漏极开路输出、三态输出

物理上的

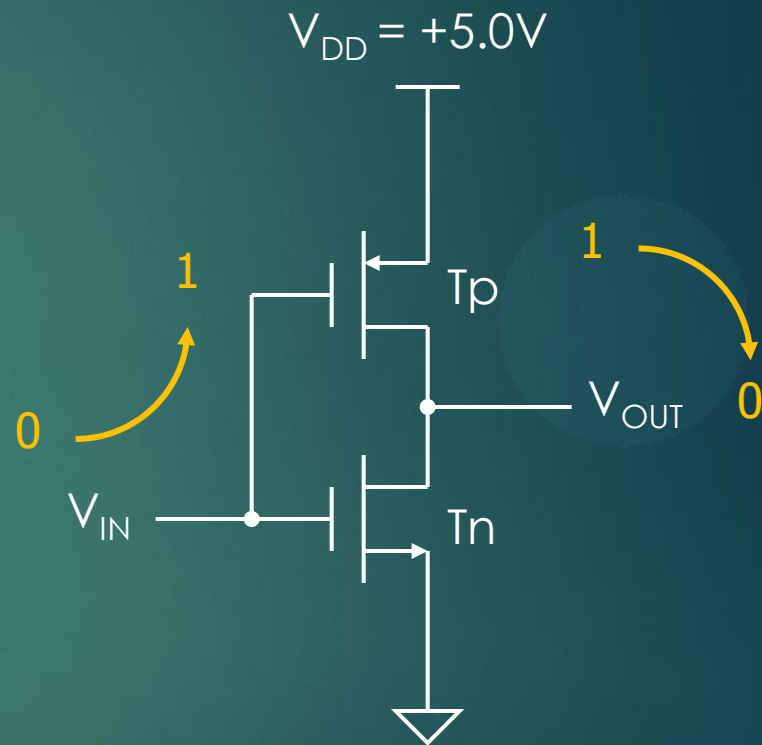
而不是逻辑上的

CMOS稳态电气特性

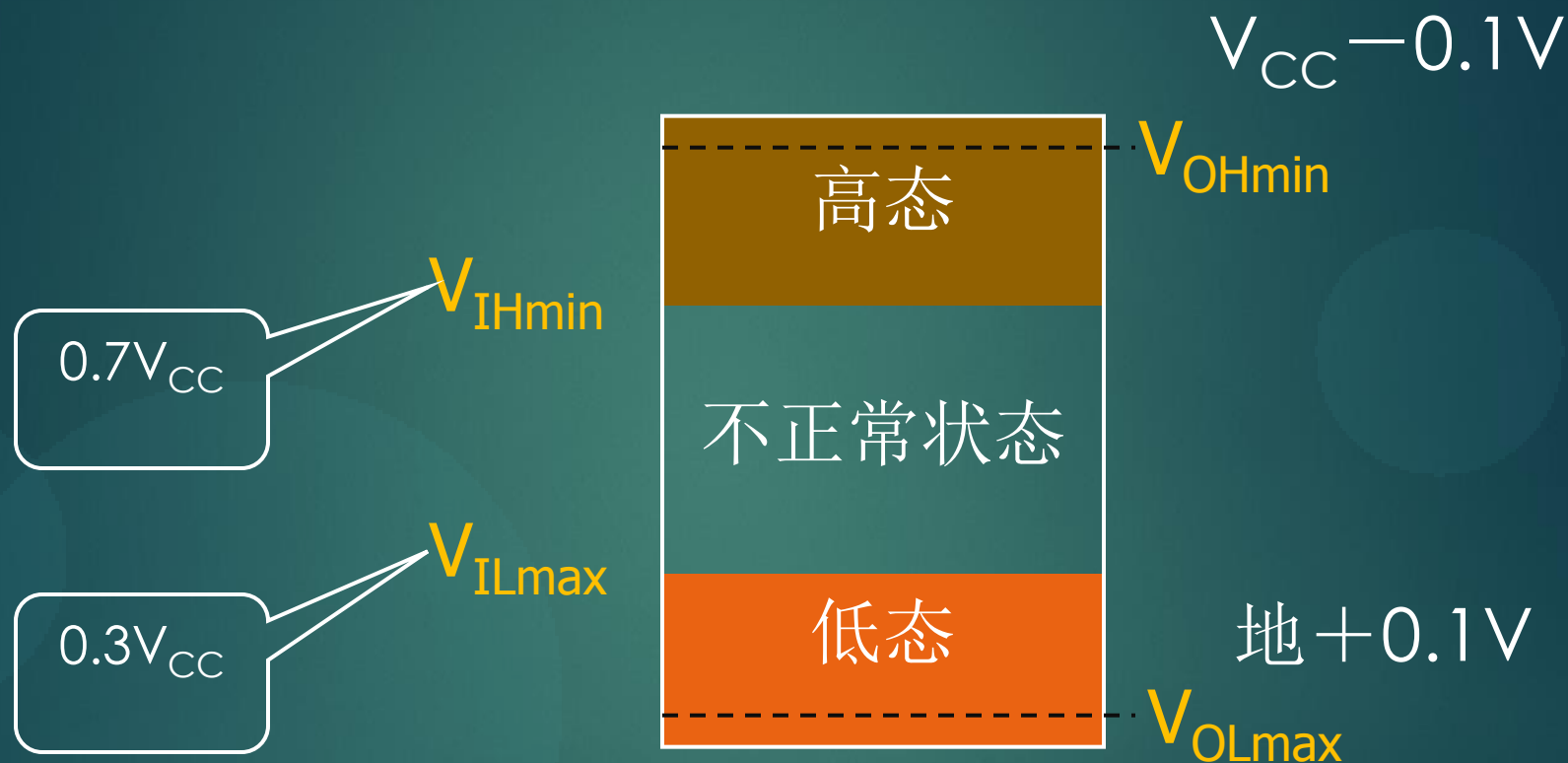
► 逻辑电平和噪声容限



电压传输特性



► 逻辑电平规格



直流噪声容限 (DC noise margin)

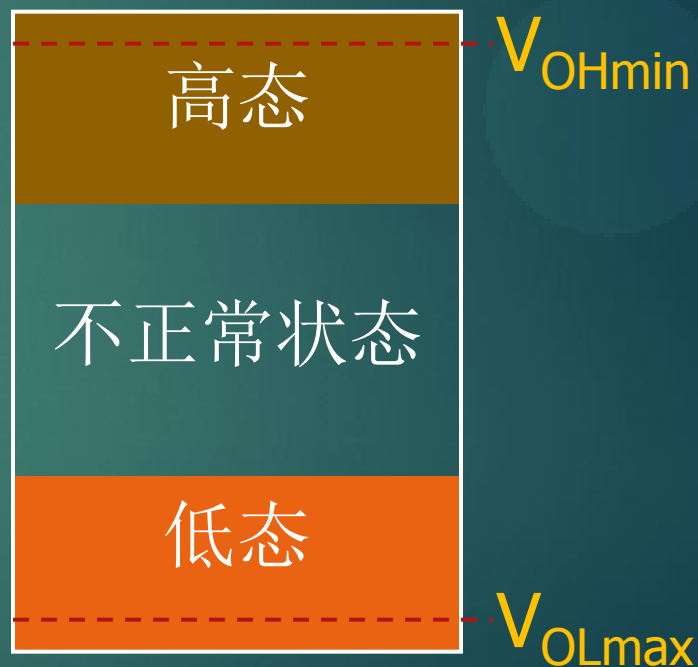
多大的噪声会使最坏输出电压被破坏得不可识别



$$30\%V_{CC} - 0.1V$$

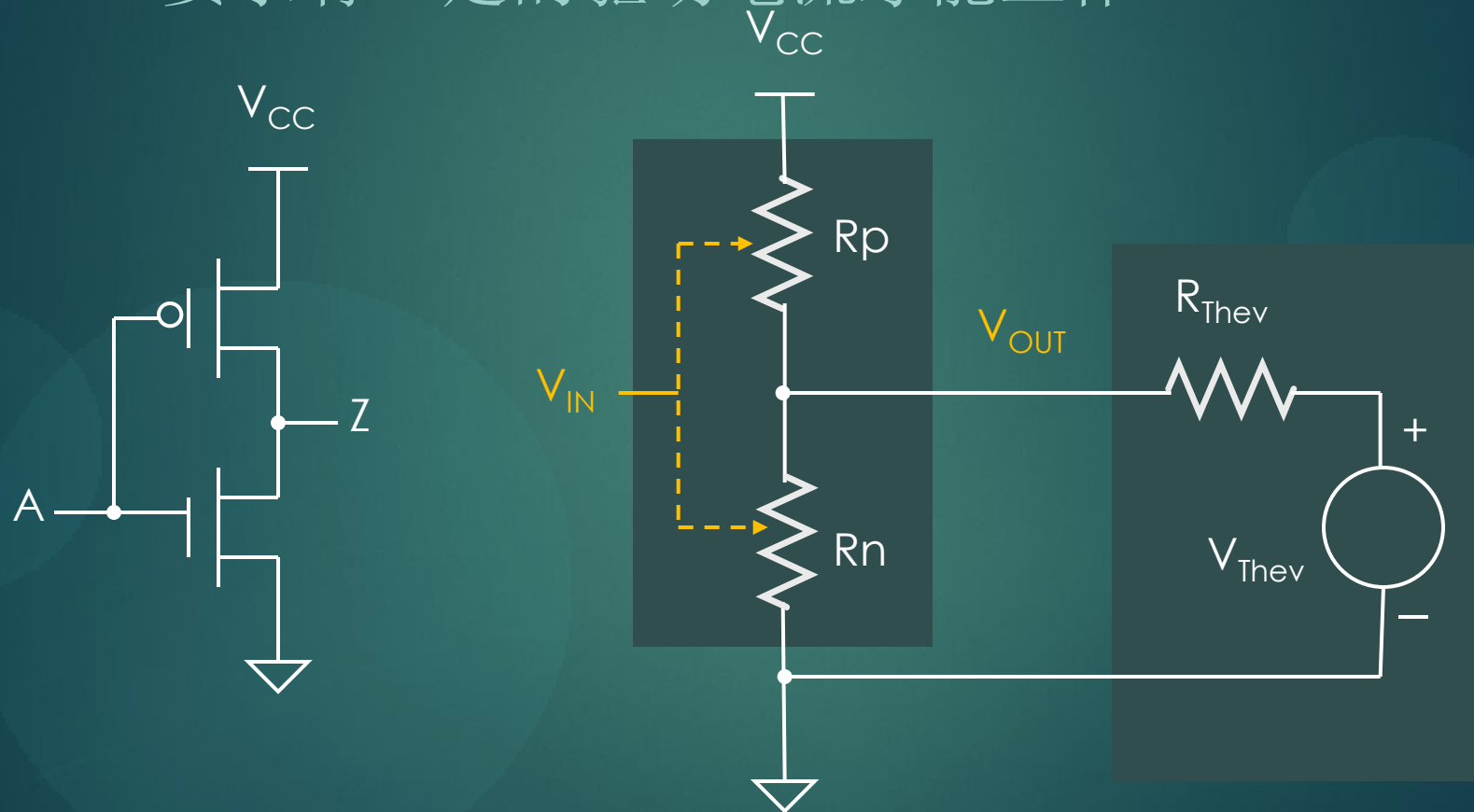
V_{IHmin}

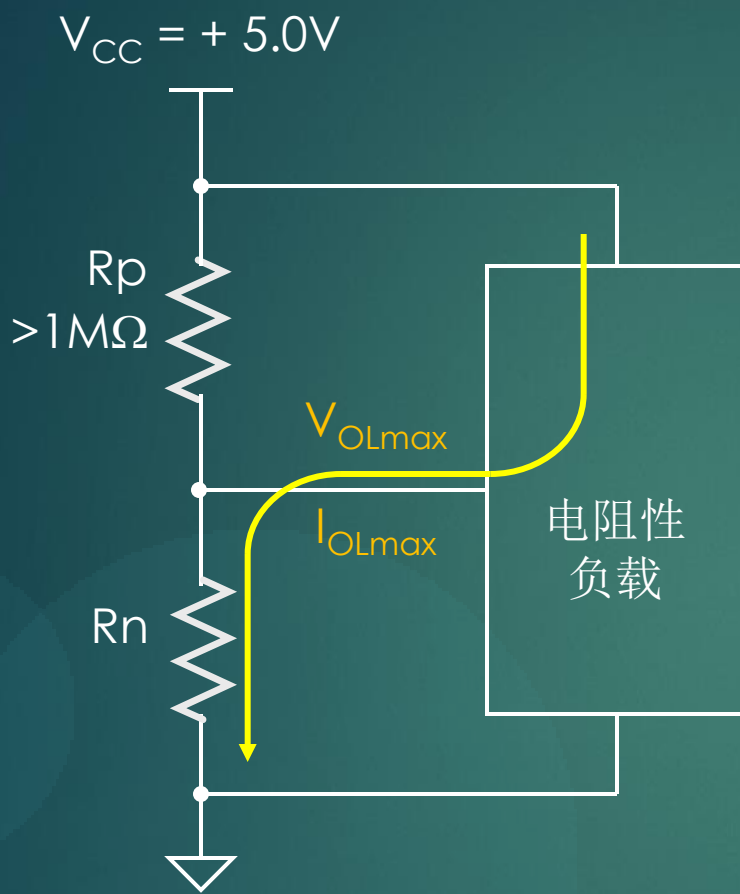
V_{ILmax}



帶電阻性负载的电路特性

要求有一定的驱动电流才能工作





输出为低态时

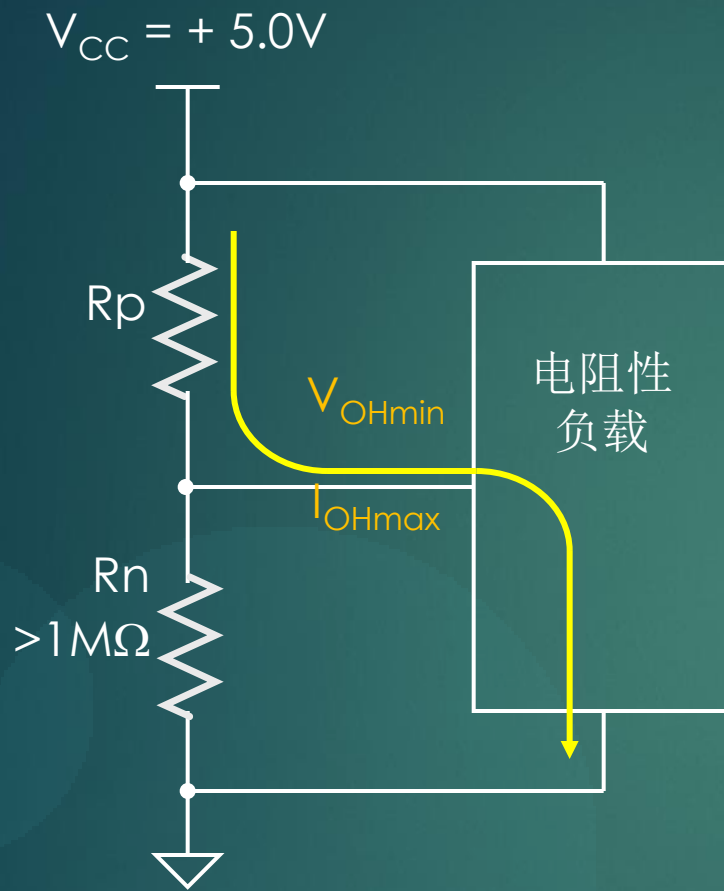
$$V_{OUT} \leq V_{OLmax}$$

输出端吸收电流

sinking current

能吸收的最大电流 I_{OLmax}

(灌电流)



输出为高态时

$$V_{OUT} \geq V_{OHmin}$$

输出端提供电流

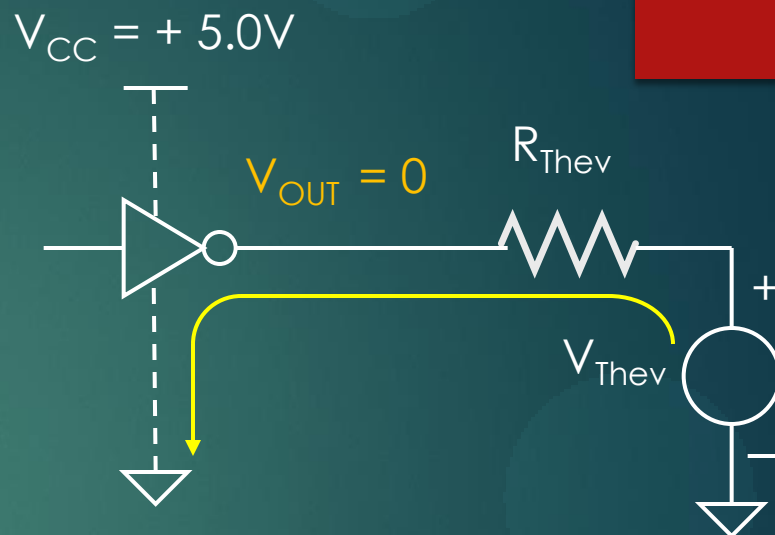
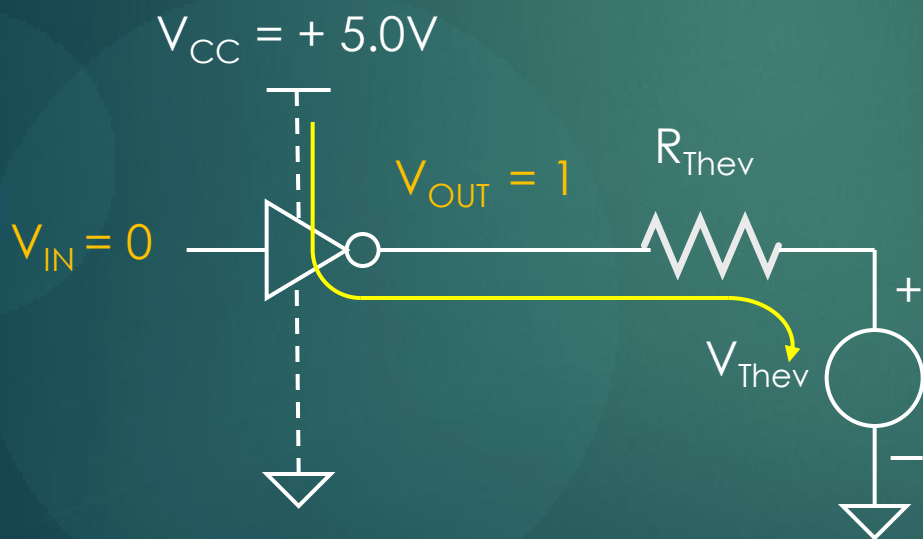
sourcing current

能提供的最大电流 I_{OHmax}

(拉电流)

输出为高态时，
估计提供电流：

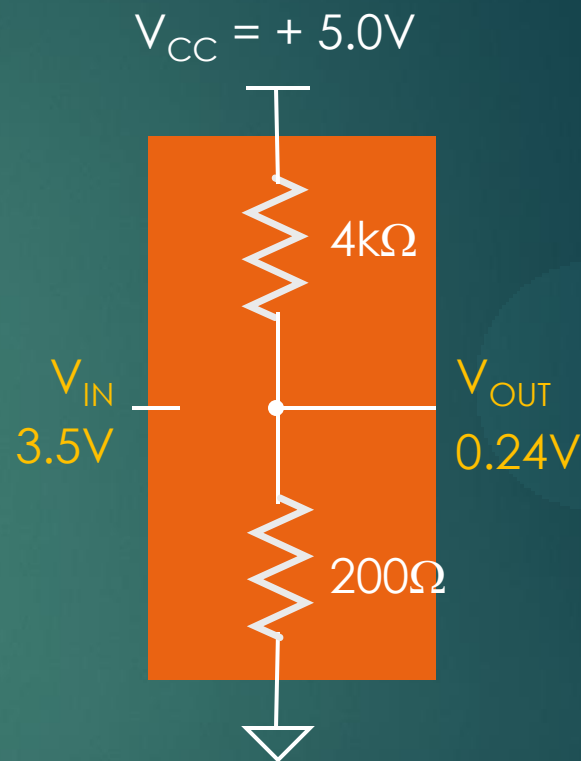
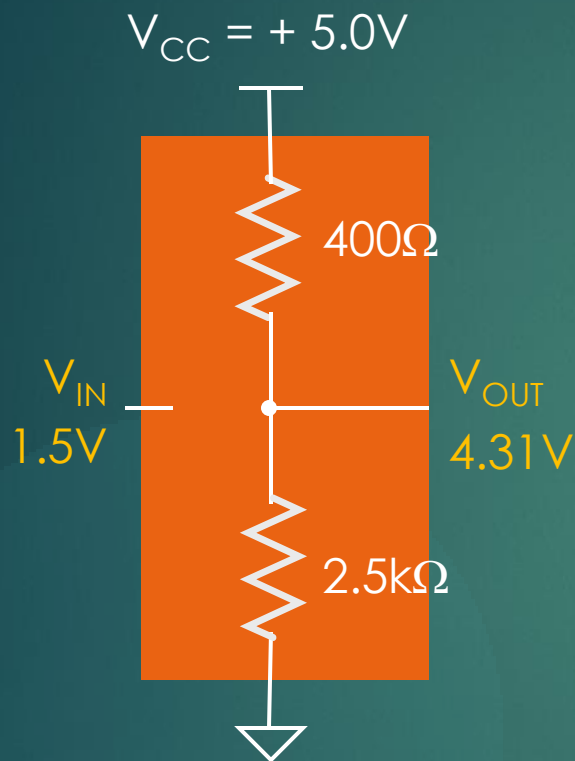
$$I_{OUT} = \frac{V_{CC} - V_{Thev}}{R_{Thev}}$$



输出为低态时，
估计吸收电流：

$$I_{OUT} = \frac{V_{Thev}}{R_{Thev}}$$

非理想输入时的电路特性



输出电压变坏（有电阻性负载时更差）
更糟糕的是：输出端电流 \uparrow ，功耗 \uparrow

扇出 (fan-out)

- ▶ 在不超出其最坏情况负载规格的条件下，一个逻辑门能驱动的输入端个数。
- ▶ 扇出需考虑输出高电平和低电平两种状态

$$\text{总扇出} = \min(\text{高态扇出}, \text{低态扇出})$$

- ▶ 直流扇出 和 交流扇出

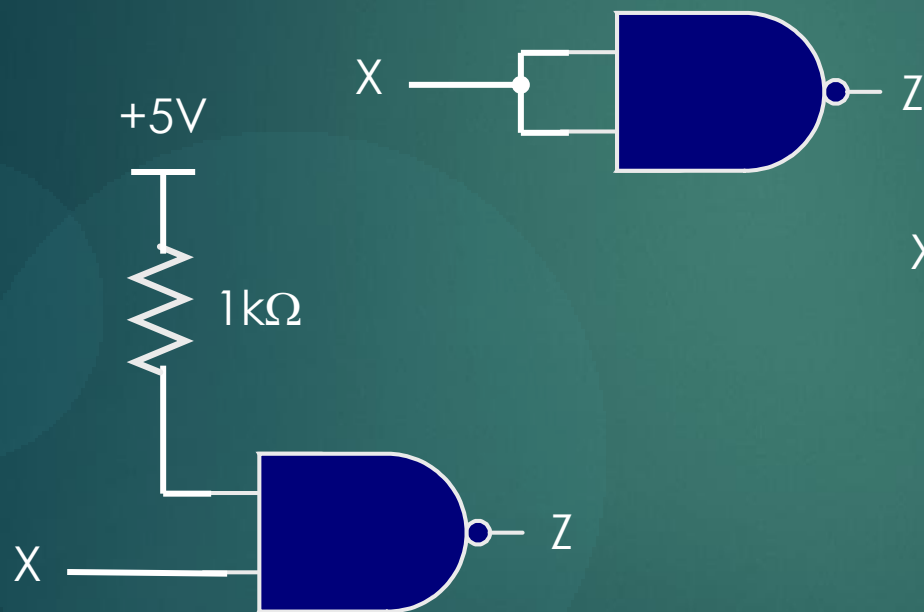
负载效应

当输出负载大于它的扇出能力时

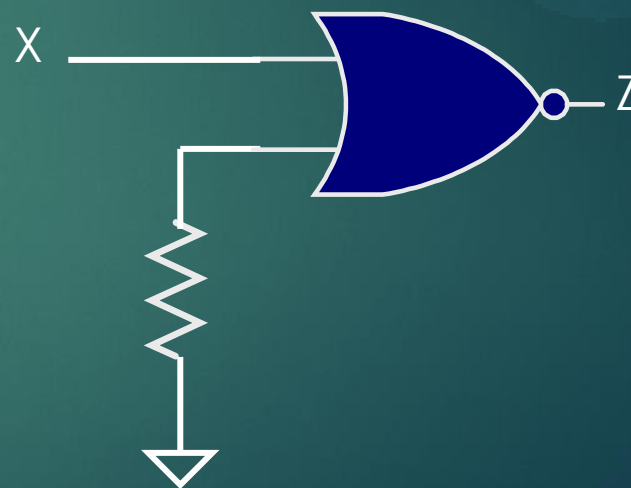
- ▶ 输出电压变差（不符合逻辑电平的规格）
- ▶ 传输延迟和转换时间变长
- ▶ 温度可能升高，可靠性降低，器件失效

不用的CMOS输入端

- ▶ 不用的CMOS输入端不能悬空



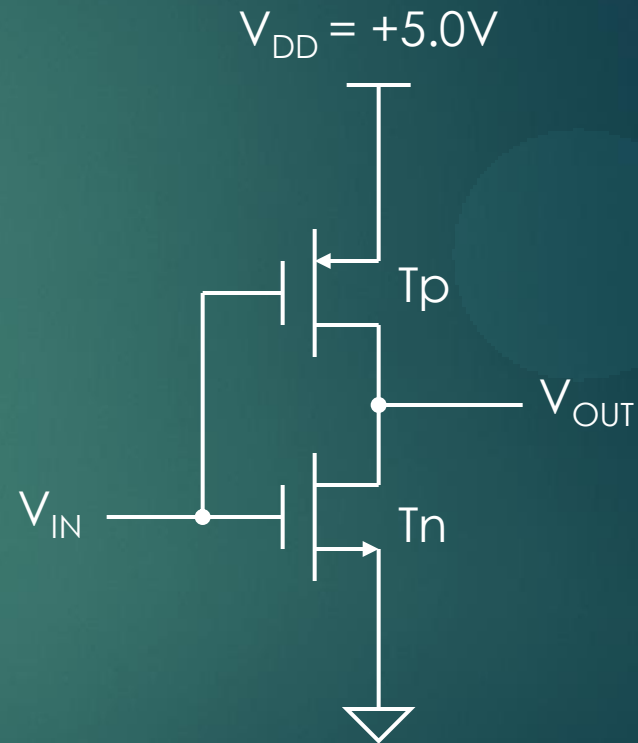
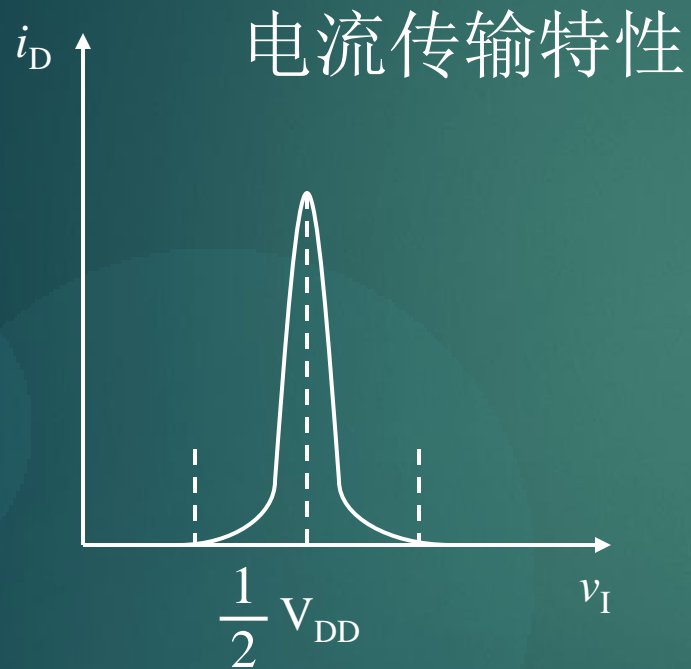
增加了驱动信号的电容负载，使操作变慢



电流尖峰和去耦电容器

current spike & decoupling capacitors

24



CMOS动态电气特性

CMOS器件的**速度**和**功耗**在很大程度上取决于器件及其负载的动态特性。

- ▶ 速度取决于两个特性：

- ▶ **转换时间**（transition time）

- ▶ **传播延迟**（propagation delay）

逻辑电路的输出从一种状态变为另一种状态所需的时间

从输入信号变化到产生输出信号变化所需的时间

- ▶ **互连线延迟**

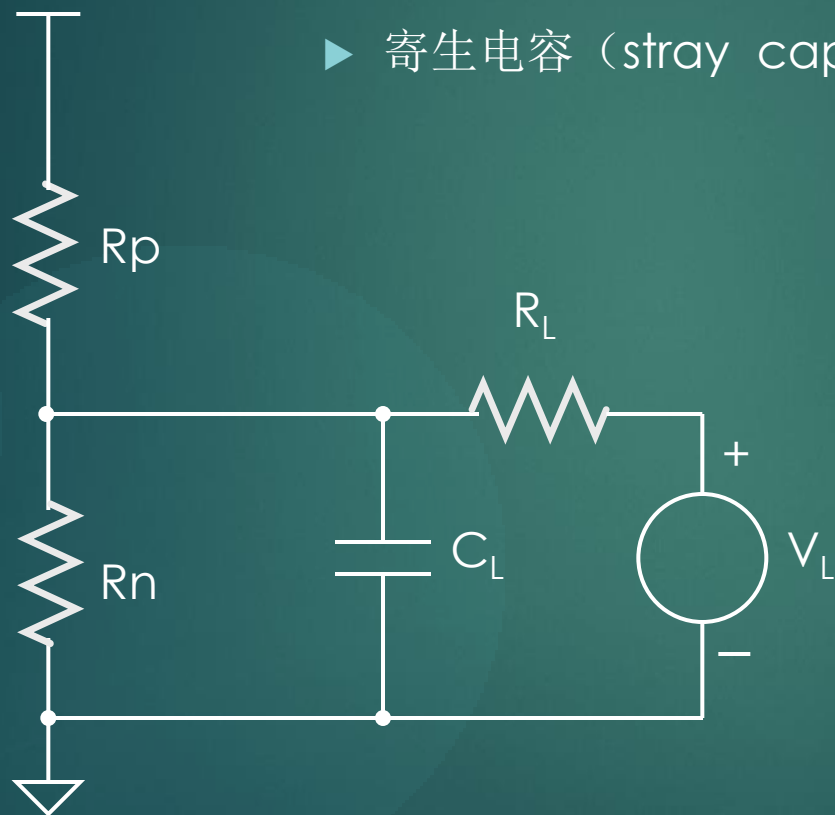
转换时间

上升时间 t_r 和 下降时间 t_f

$V_{CC} = +5.0V$

► 晶体管的“导通”电阻

► 寄生电容 (stray capacitance)

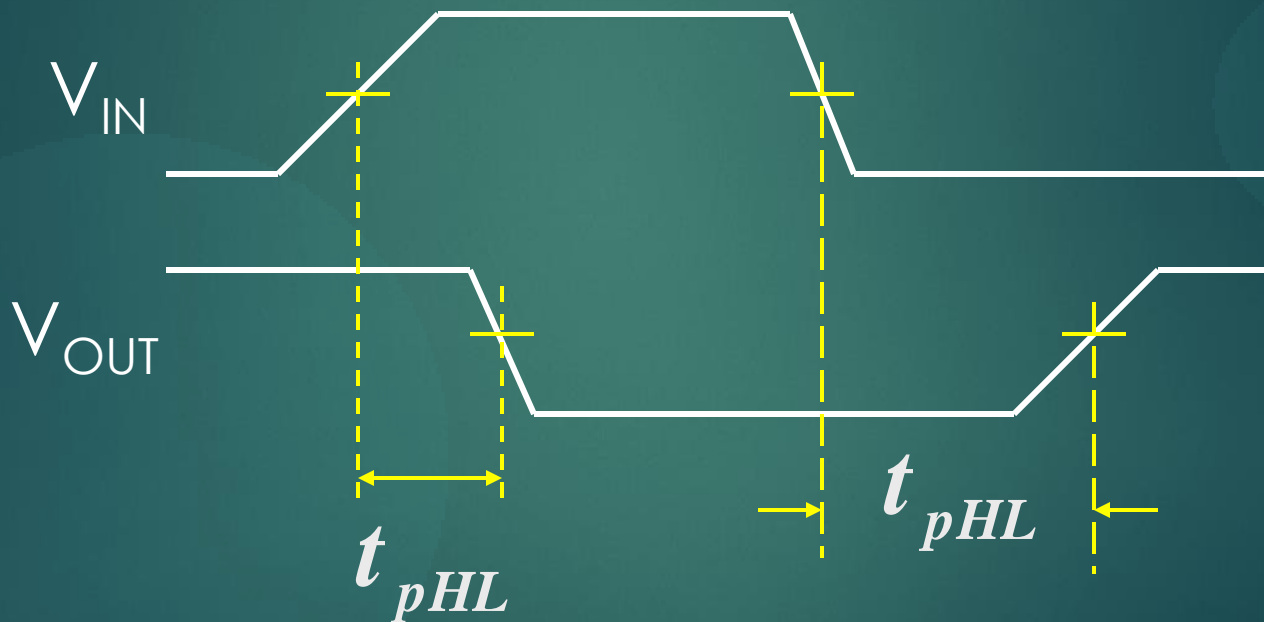


电容两端电压不能突变

在实际电路中
可用时间常数
近似转换时间

传播延迟

信号通路：一个特定输入信号到逻辑元件的特定输出信号所经历的电气通路。



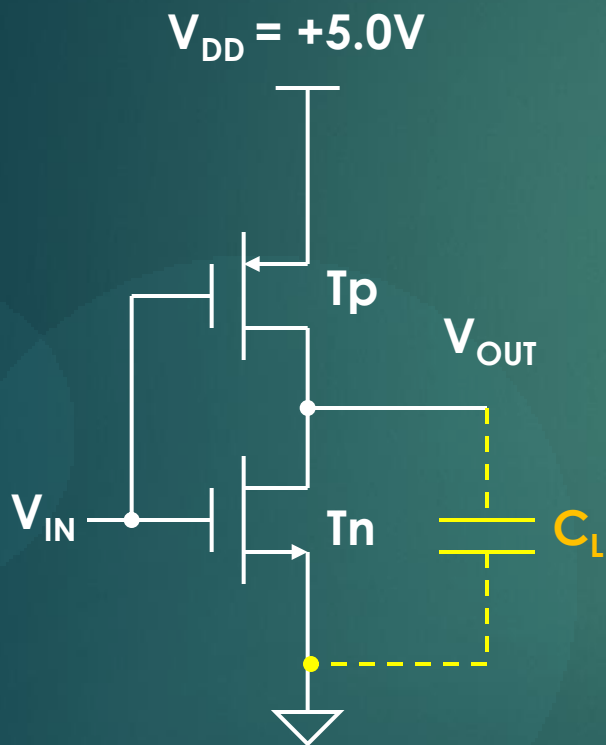
功率损耗

- ▶ 静态功耗 (static power dissipation)
 - ▶ 漏电流 (Leakage Current)
- ▶ 动态功耗 (dynamic power dissipation)
 - ▶ 两个管子瞬间同时导通产生的功耗 P_T
 - ▶ 对负载电容充、放电所产生的功耗 P_L

功率损耗

分为：静态功耗、动态功耗

29



动态功耗的来源：

- ▶ 两个管子瞬间同时导通产生的功耗 P_T
- ▶ 对负载电容充、放电所产生的功耗 P_L

功率损耗

分为：静态功耗、动态功耗

30

动态功耗的来源：

V_{CC} 的大小
输入波形的好坏
输入信号频率

▶ 两个管子瞬间同时导通
产生的功耗 P_T

负载电容
输入信号频率
 $(V_{CC})^2$

▶ 对负载电容充、放电所
产生的功耗 P_L

三态输出

当 $EN=0$ 时,

$C=1$, T_p 截止

$B=1$, $D=0$, T_n 截止

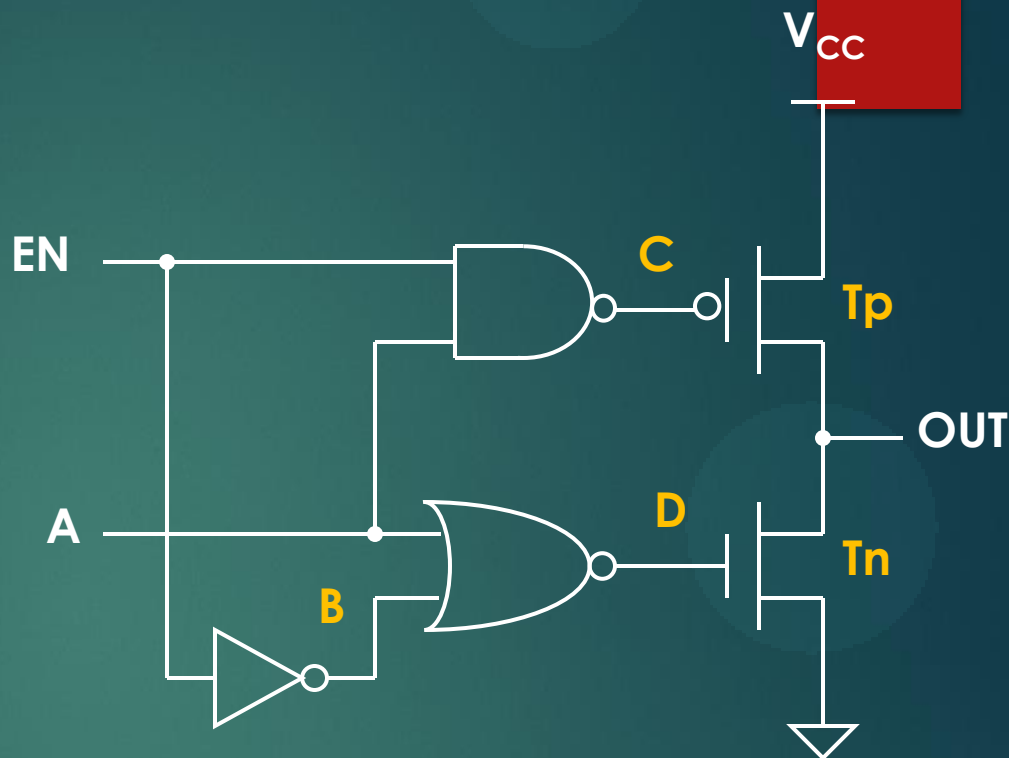
高阻态 (悬空态)

当 $EN=1$ 时,

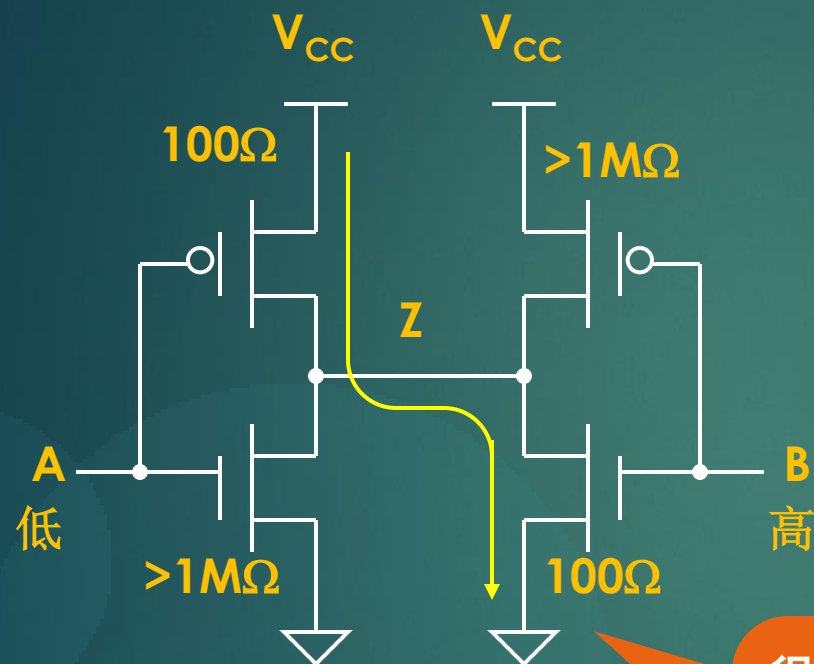
$C=A'$, $B=0$, $D=A'$

由 A 控制输出为

逻辑0 或 逻辑1



漏极开路输出



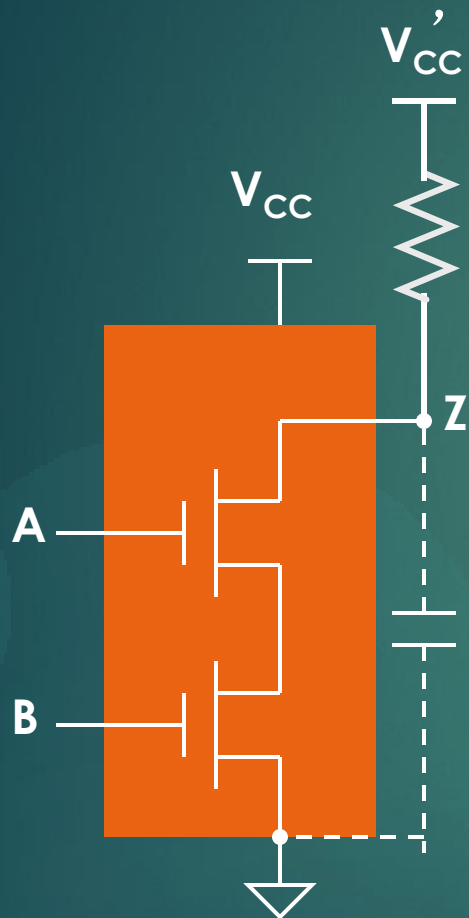
有源上拉
active pull-up

有源上拉的CMOS器件
其输出端不能直接相联

输出电平??
造成逻辑混乱

很大的负载电流
同时流过输出级
可使门电路损坏

漏极开路输出



R 上拉电阻

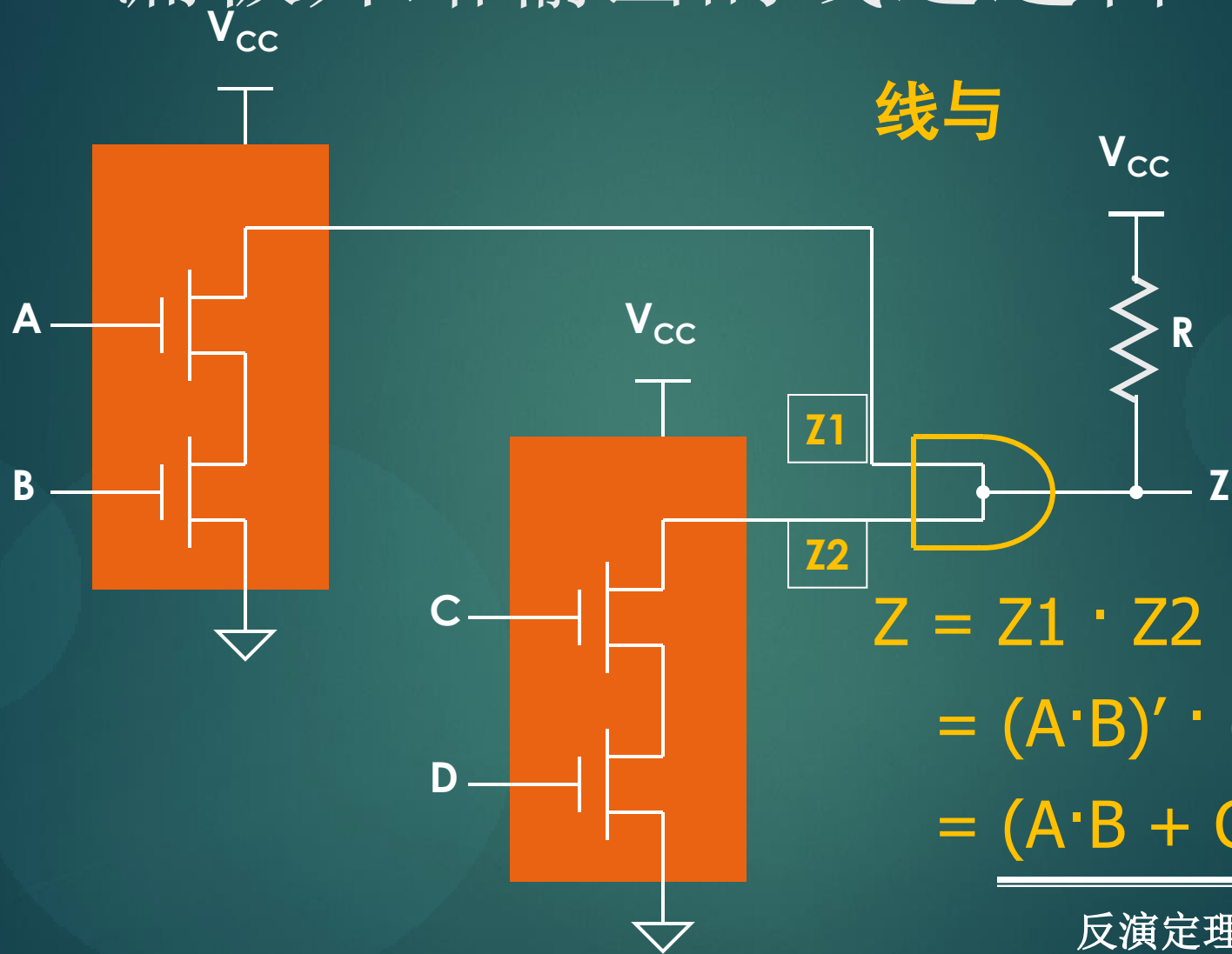
希望尽量小，减少上升时间
太小则吸收电流太大

逻辑符号



应用：驱动LED、线与、
驱动多源总线

漏极开路输出的线连逻辑



集成电路的分类

按功能分：数字电路、线性电路（模拟电路）两大类

数字电路：从门电路到微处理器、存储器等多种

按半导体制造工艺：双极型(TTL,LTTL,STTL,LSTTL,ECL...)

MOS(PMOS,NMOS,CMOS,BiCMOS...)

两大类工艺技术的特点：

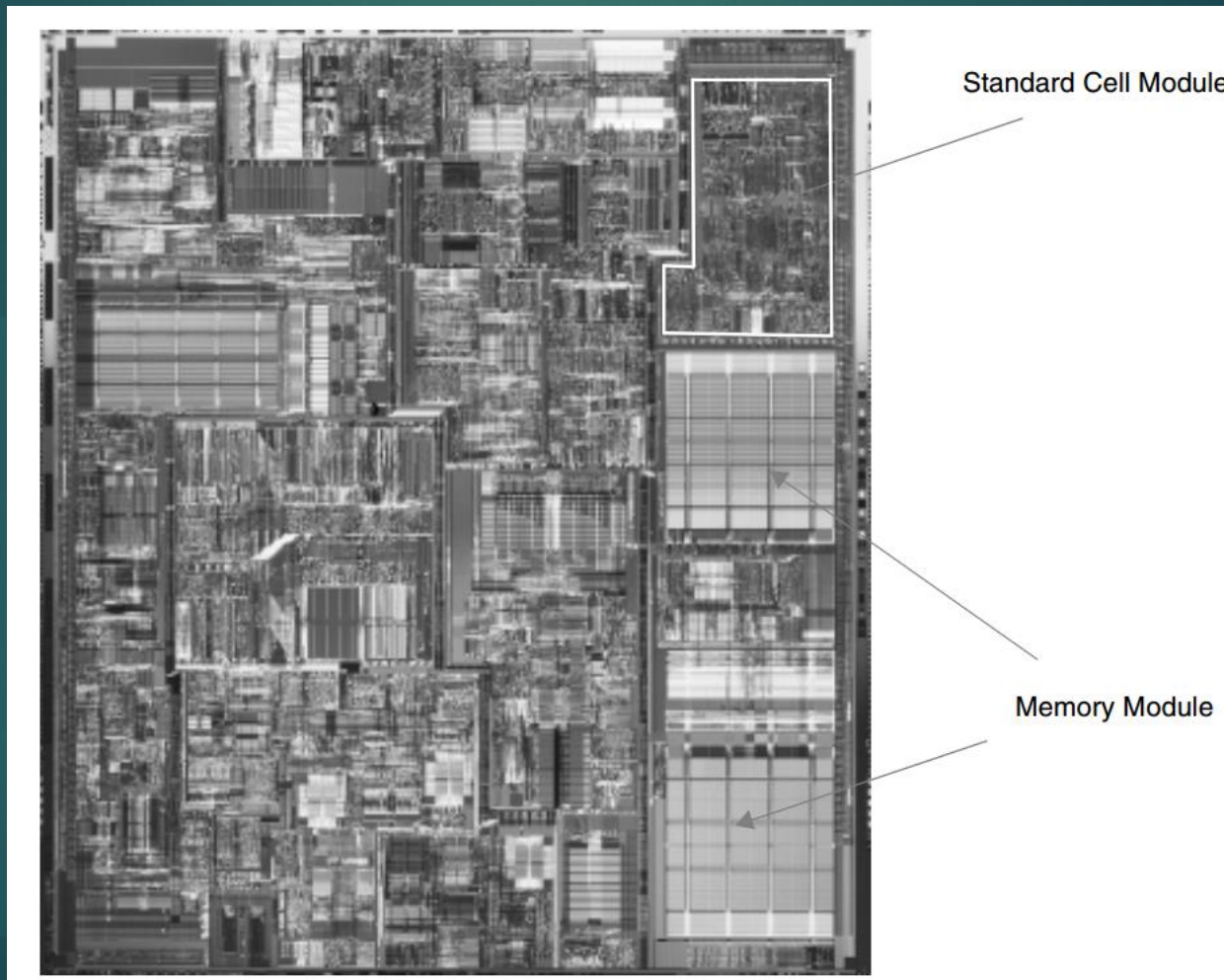
	速度	功耗	集成度
TTL(晶体管晶体管逻辑)	快	大	低
MOS（金属氧化物半导体）	慢	小	高

目前最常用的工艺：CMOS（互补金属氧化物半导体）

按封装（外形）分：双列直插、表面封装、BGA(Ball Grid Array)

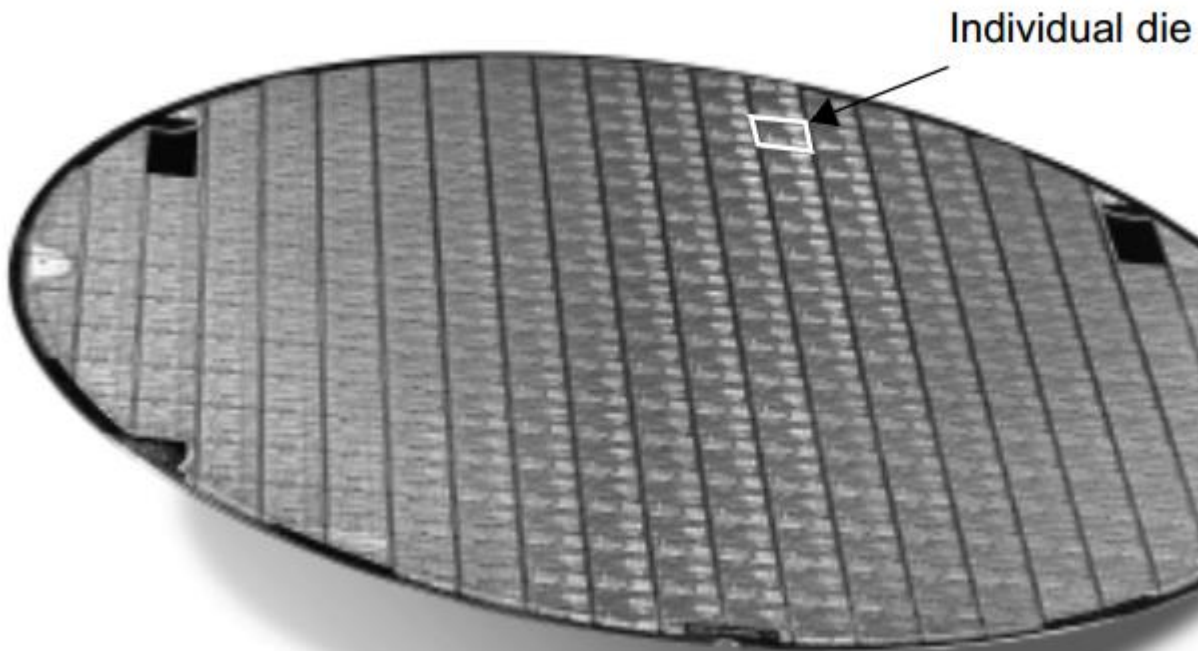
集成电路的物理实现

► Intel Pentium 4



集成电路的制造

► AMD Duron 微处理器



集成电路发展历史

“集成电路” (IC)是相对“分立原件”而言的，是所有以半导体工艺将电路集成到一块芯片的器件总称。

半导体制造工艺的发展带动了集成电路的更新换代。

VLSI时代存储器件制造工艺带动了整个微处理器的更新换代。

摩尔定律：每18个月集成度翻一翻。

集成电路内部的特征尺寸是主要的指标：

0.8 μm , 0.35 μm , 0.25 μm , 0.18 μm , 0.13 μm

集成电路发展历史（续）

（1）Small Scale IC （SSI）

小规模 IC 1965年

规模： 10个门/片电路以下

主要产品： 门电路
触发器（Flip Flop）

集成电路发展历史（续）

（2） Medium Scale IC （MSI）

中规模 IC 1970年

规模：10—100个门/片

主要产品：逻辑功能部件

4位ALU（8位寄存器）

集成电路发展历史（续）

（3）Large Scale IC （LSI）

大规模 IC 1976年

规模：100—1000个门/片

主要产品：规模更大的功能部件

存储器，8位CPU

集成电路发展历史（续）

（4）Very large Scale IC （VLSI）

超大规模 IC 80年代初

规模： 1000个门以上

多个子系统集成

集成电路发展历史（续）

（5）Ultra large Scale IC （ULSI）

甚大规模IC（微处理器等）

每隔18个月，集成度翻一翻

价格1/2

品种多

性能高