

数字电路与逻辑设计笔记

陈鸿峥

2019.01 *

目录

1	概述	1
2	基本概念	1
2.1	数字与模拟	1
2.2	数的表示	1
2.3	存储器	3
2.4	信号处理	3
2.5	其他	4
3	集成电路	4
4	组合电路	4
4.1	基本逻辑门	4
4.2	布尔(Boolean)代数	4
4.3	卡诺(Karnaugh)图	8
4.4	功能器件	8
4.5	竞争与冒险	12
5	时序电路	12
5.1	锁存器	12
5.2	触发器	13
5.3	单稳态触发器	14
5.4	555计时器	14
5.5	移位寄存器	16
5.6	计数器	17

*Build 20190117

6 电路设计	17
6.1 电路分类	17
6.2 设计步骤	17
6.3 实例操作	19

1 概述

1. 目的：系统掌握数字系统的基本概念、思维模式与设计方法，帮助深入理解计算机架构
2. 应用：

- 人工智能+GPU/FPGA
- 云计算+FPGA
- 物联网+传感硬件系统
- 软件定义网络系统

2 基本概念

2.1 数字与模拟

1. 数字(digital)量：离散值，01
2. 模拟(analog)量：连续值

2.2 数的表示

2.2.1 进制(system)

二进制(binary)、八进制(octonary)、十进制(decimal)、十六进制(hexadecimal)

- 十进制转二进制：整数部分除以2取余，小数部分乘2取整
- 二进制转十六进制：四位四位统计

2.2.2 符号数

- 符号数值(sign-magnitude)形式：首位0为正数，1为负数，n位范围为 $-(2^{n-1} - 1) \sim +2^{n-1} - 1$
- 反码(1's complement)：除符号位不变，其他位取反，n位范围为 $-(2^{n-1} - 1) \sim +2^{n-1} - 1$
- 补码(2's complement)：反码+1，按照原来十进制转二进制方法即可得对应符号十进制数，n位范围为 $-2^{n-1} \sim +2^{n-1} - 1$ （由于没有正负0，故表示的数多了一位），补码的补码为原码

2.2.3 浮点数

单精度(float)32位，双精度(double)64位

指数加127相当于做了一个平移，科学记数法如下表示，

$$\text{Number} = (-1)^S (1 + F) 2^{E-127}$$

例 1.

$$1\ 0110\ 1001\ 0001 = 1.0110\ 1001\ 0001 \times 2^{12}$$

指数：12 + 127 = 139 → 1000 1011

尾数：011 0100 1000 1000 0000 0000 左对齐，因为有小数点

符号 S	指数 $E(exponent)$	尾数 $F(mantissa)$
0	1000 1011	011 0100 1000 1000 0000 0000
1位	8位	23位

2.2.4 运算法则

用补码进行计算，操作跟原码相同，且不会出现两个0

2.2.5 其他表示

1. BCD码/8421码：即四位的二进制表示
2. 格雷(Gray)码：相邻只变一位

二进制码转格雷码：

$$\begin{array}{lcl}
 \text{Binary :} & 1 & \xrightarrow{+} 0 & \xrightarrow{+} 1 & \xrightarrow{+} 1 & \xrightarrow{+} 0 \\
 & \downarrow & & \downarrow & & \downarrow \\
 \text{Gray :} & 1 & & 1 & & 0 & & 1
 \end{array}$$

格雷码转二进制码：

$$\begin{array}{lcl}
 \text{Gray :} & 1 & & 1 & & 1 & & 0 & & 1 \\
 & \downarrow & \nearrow + & \downarrow & \nearrow + & \downarrow & \nearrow + & \downarrow & \nearrow + & \downarrow \\
 \text{Binary :} & 1 & & 0 & & 1 & & 1 & & 0
 \end{array}$$

2.3 存储器

2.3.1 随机存取存储器(RAM)

- 静态(SRAM)：用锁存器作为存储单元，只要有电源就可以一直存，读快
- 动态(DRAM)：用电容器作为存储单元，需要不断刷新(refreshing)，存储容量大

2.3.2 只读存储器(ROM)

永久或半永久存储数据

2.3.3 存储扩展

- 字长(word-length)扩展

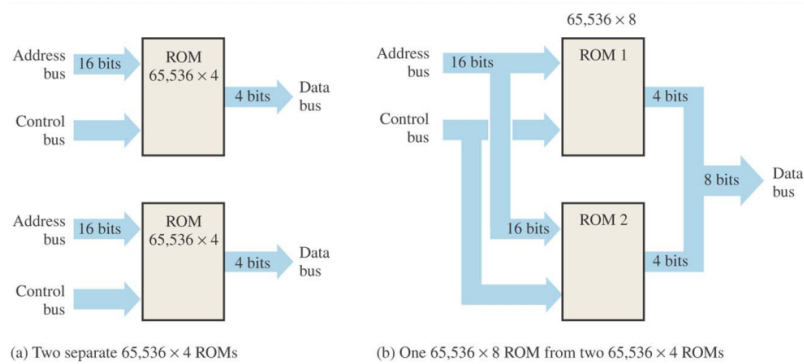


图 1: 字长扩展

- 字容量(word-capacity)扩展

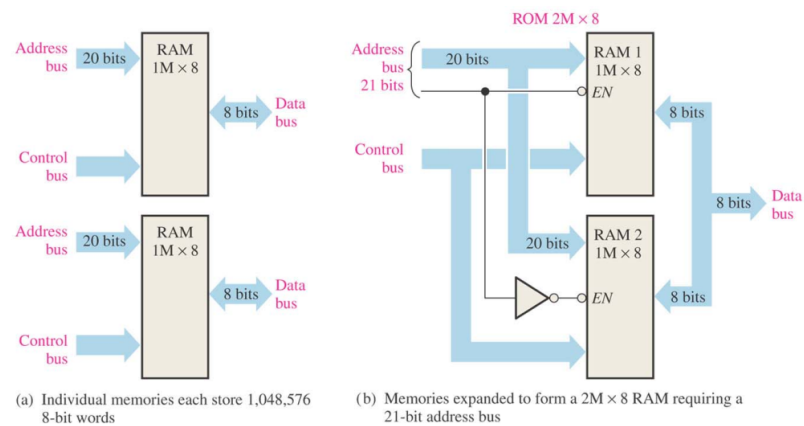


图 2: 字容量扩展

2.4 信号处理

1. 采样(sampling)

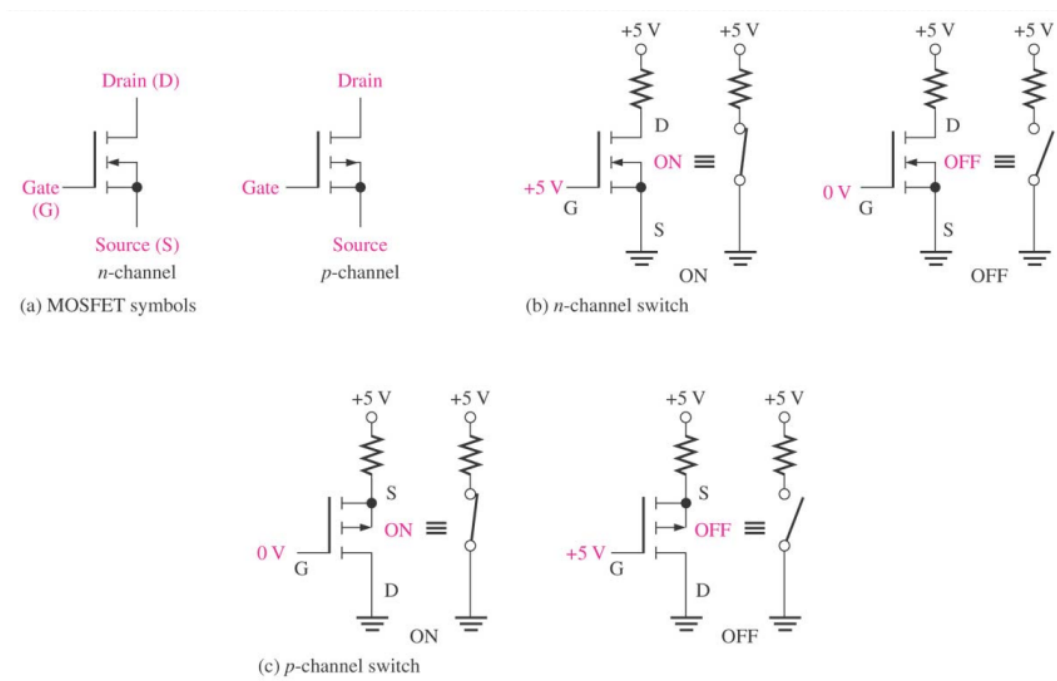
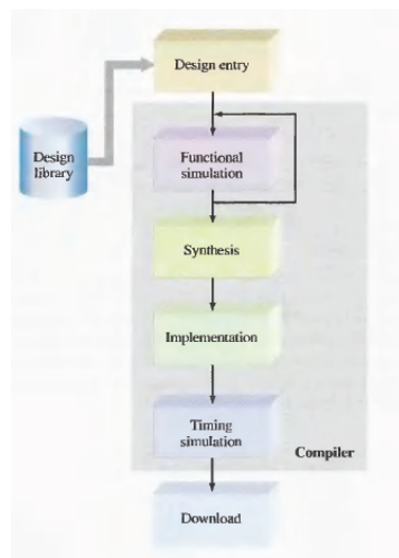
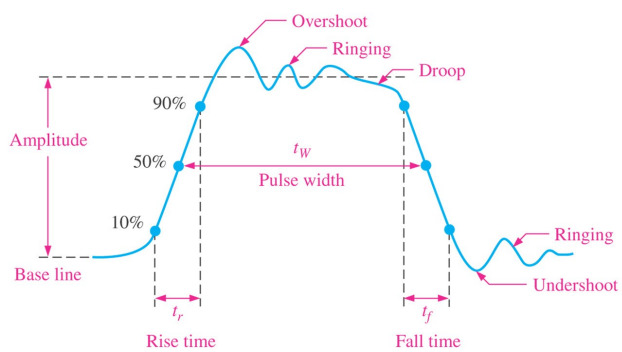
采样频率至少是原来最高频率的两倍

2. 滤波(filtering)

奈奎斯特(Nyquist)频率等于采样频率的一半

2.5 其他

占空比(duty cycle): $\frac{t_W}{T} \times 100\%$



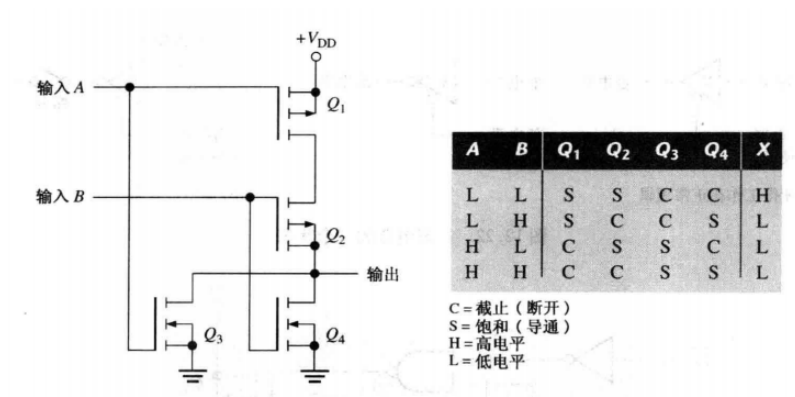


图 4: CMOS或非门

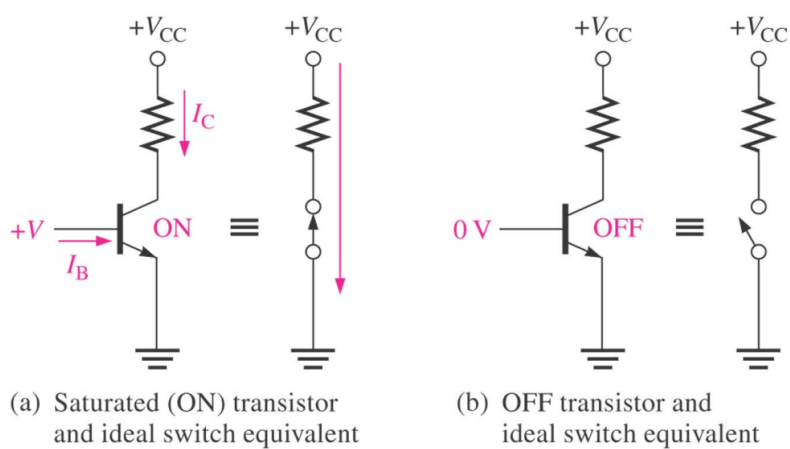


图 5: TTL电路

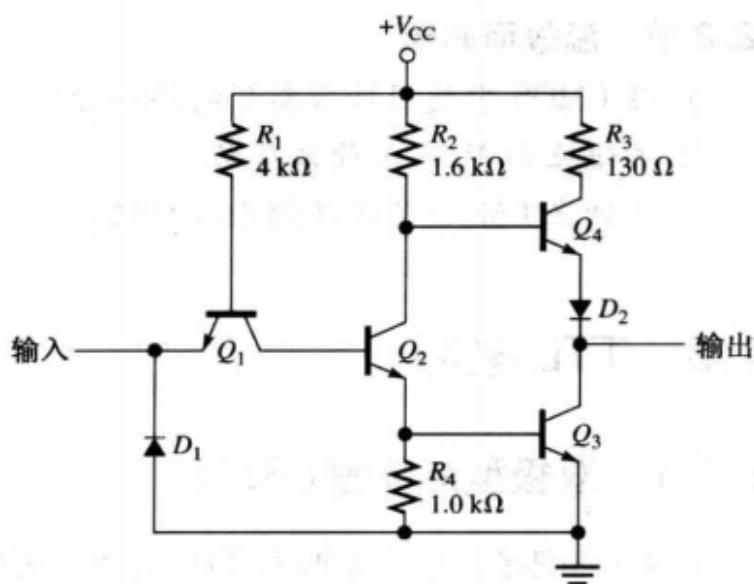


图 6: TTL反相器

3 集成电路

由图3, Q_1 被 V_{CC} 上拉, 始终导通. 若输入为高电平, Q_2 导通, Q_3 导通, 输出被下拉为低电平. 同时, Q_2 在集电极处足够低的电压可以使 Q_4 截至.

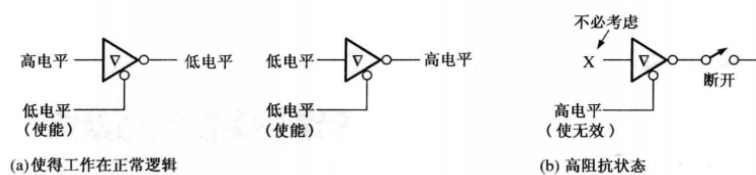


图 7: 三态门

4 组合电路

4.1 基本逻辑门

4.2 布尔(Boolean)代数

满足交换律、结合律、分配律

$$A\bar{A} = 0$$

$$A + \bar{A} = 1$$

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

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

$$\begin{aligned} A + BC &= A(1 + B + C) + BC \\ &= (A + B)(A + C) \end{aligned}$$

$$\begin{aligned} A + \bar{A}B &= A(1 + B) + \bar{A}B \\ &= A + B \end{aligned}$$

4.3 卡诺(Karnaugh)图

4.4 功能器件

4.4.1 加法器

$$\text{Carry generation: } C_g = AB$$

$$\text{Carry propagation: } C_g = AB$$

$$\text{Output carry: } C_g = AB$$

$$C_{in2} = C_{out1} = C_{g1} + C_{p1}C_{in1}$$

$$C_{in3} = C_{out2} = C_{g2} + C_{p2}C_{in2} = C_{g2} + C_{p2}(C_{g1} + C_{p1}C_{in1})$$

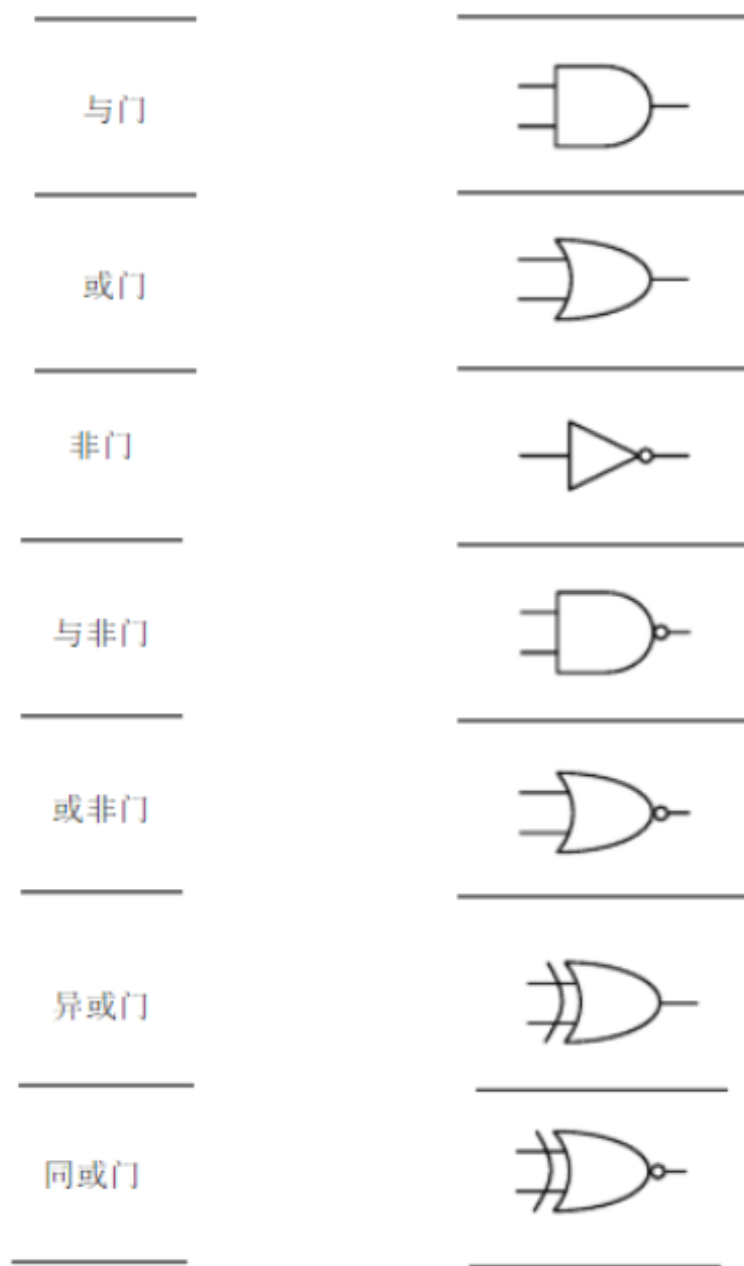


图 8: 基本逻辑门

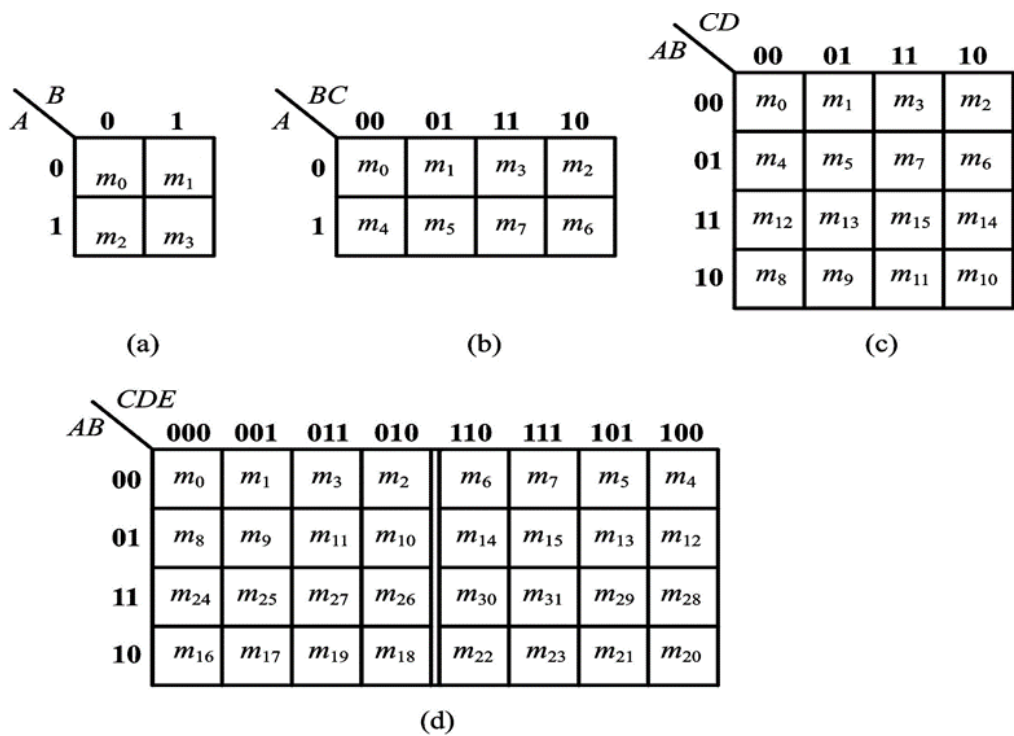


图 9: 不同阶卡诺图

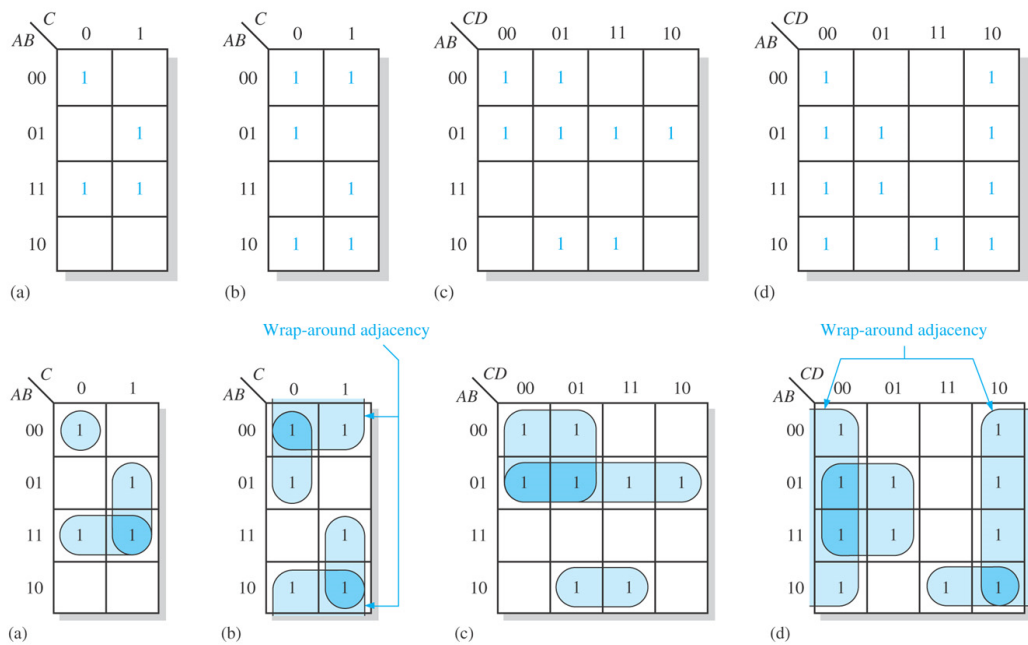


图 10: Sum of Product(SOP)化简

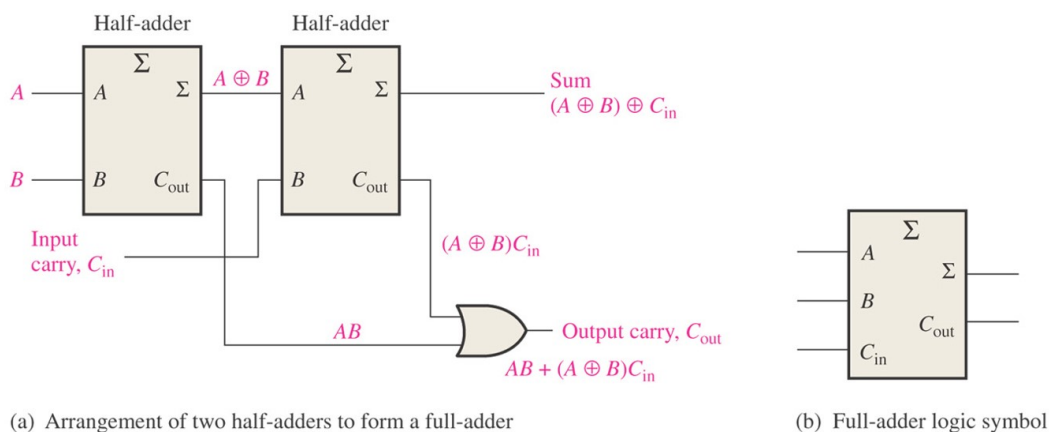


图 11: 半加法器与全加法器

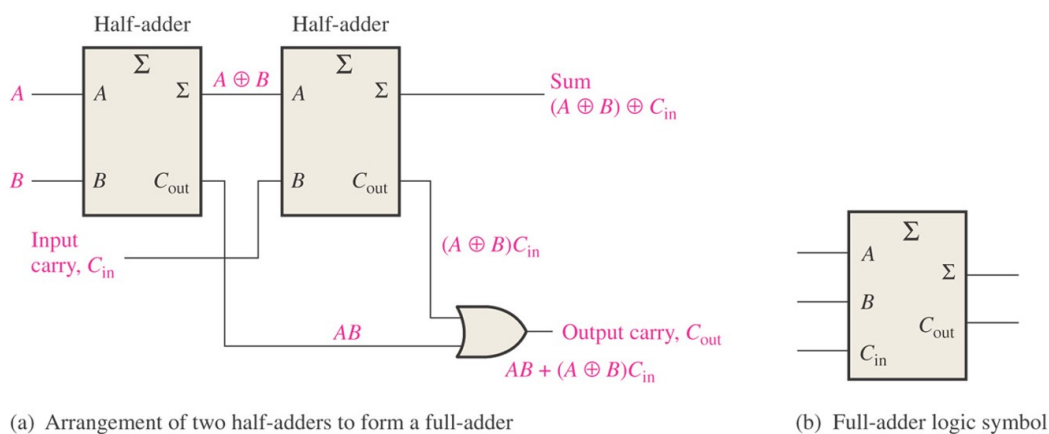


图 12: 异步加法器改造为同步加法器

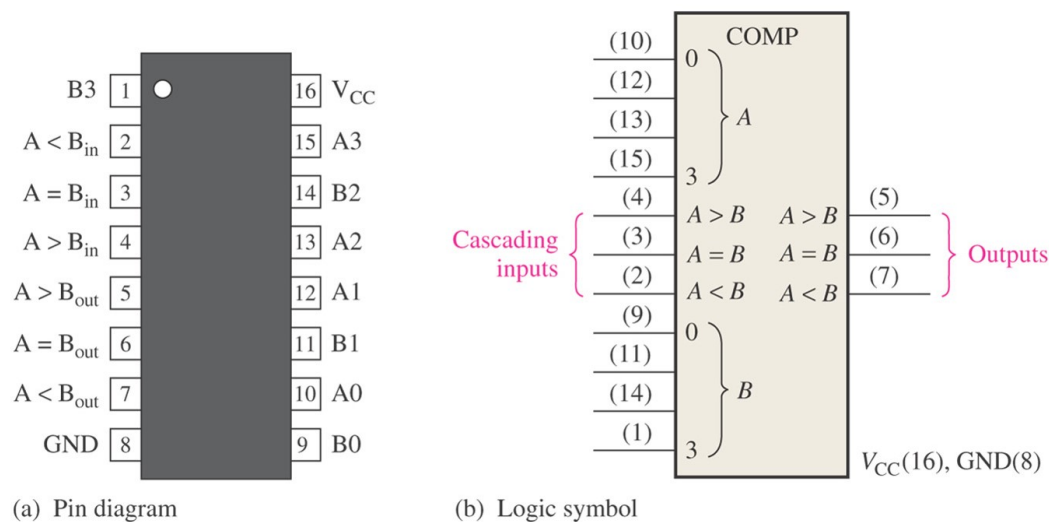


图 13: 比较器

4.4.2 比较器(Comparator)

4.4.3 译码器(Decoder)

BCD码转对应端口输出，注意输出是反的

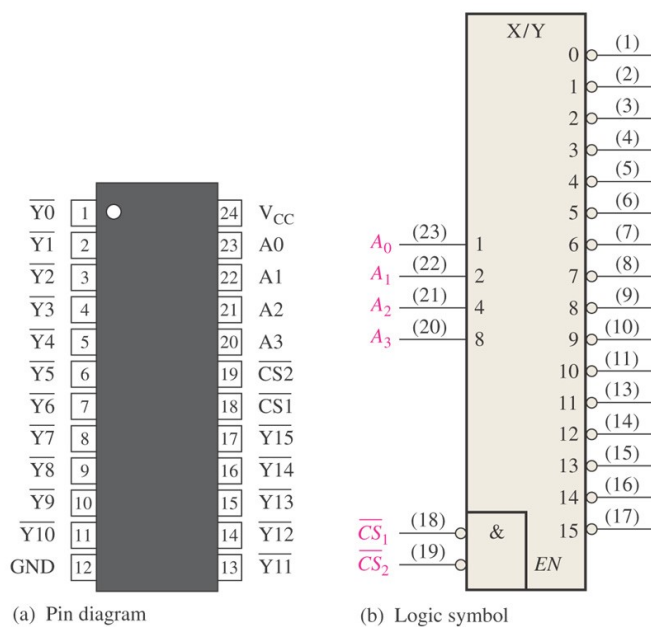


图 14: 译码器

BCD转7段数码管

1. 共阴(cathode): 高电平亮
2. 共阳(anode): 低电平亮

4.4.4 编码器(Encoder)

输入转BCD码

4.4.5 选择器(Multiplexer)

通过BCD码选择对应路输出

4.4.6 多路分配器(Demultiplexer)

将对应输入分配到对应输出路

4.5 竞争与冒险

1. 竞争(race): 输入到输出途径不同, 延时时间不同, 到达输出的时间不同
2. 冒险(hazard): 竞争结果导致逻辑电路产生错误输出

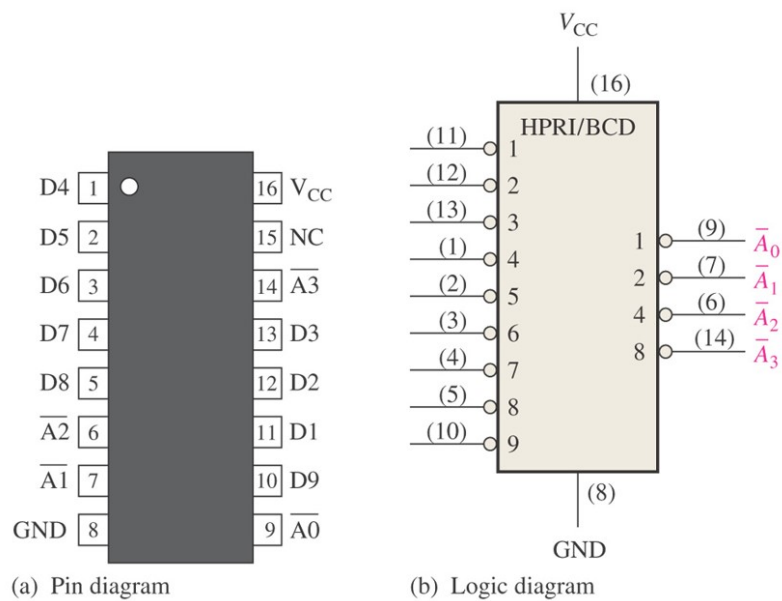


图 15: 编码器

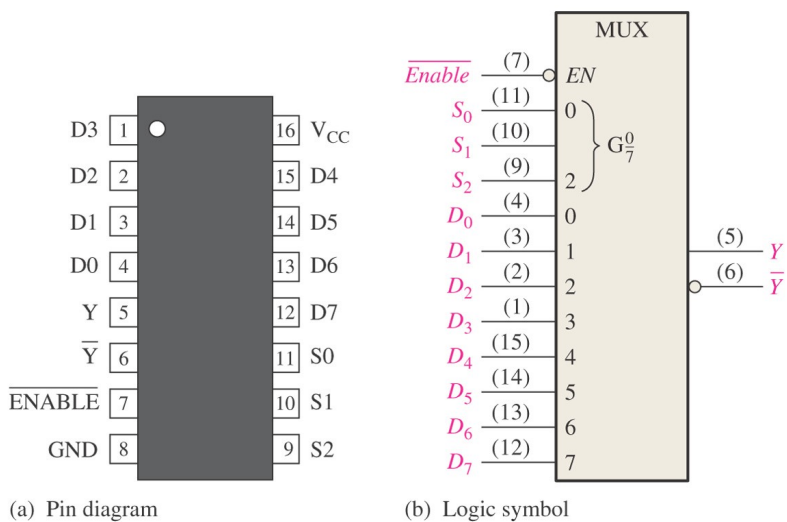


图 16: 选择器

如 $F = AB + \overline{A}C$ ，因为取非，导致两条道路时间不同，使得输出出现毛刺现象
可加入冗余项以避免冒险，如改成 $F = AB + \overline{A}C + BC$

5 时序电路

5.1 锁存器

用于存储数据

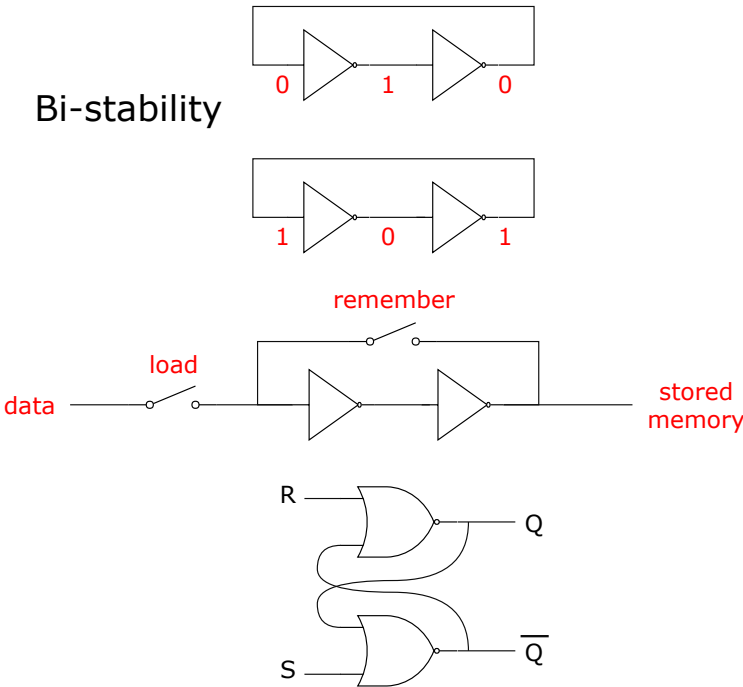


图 17: SR锁存器(latch)

SR锁存器状态表

S	R	状态
0	0	不变
0	1	复位
1	0	置位
1	1	N/A

D锁存器状态：0复位，1置位
门（选通端）：决定是否运作

5.2 触发器

5.2.1 SR/D触发器

触发器状态变化与锁存器相同 边缘触发其实通过竞争实现（如输入加一个与门后与非 $\overline{A\overline{A}}$ ）

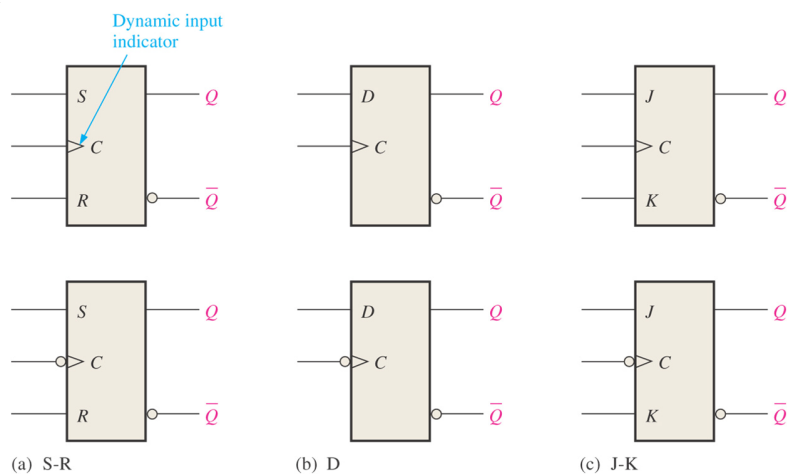


图 18: 触发器

5.2.2 JK触发器

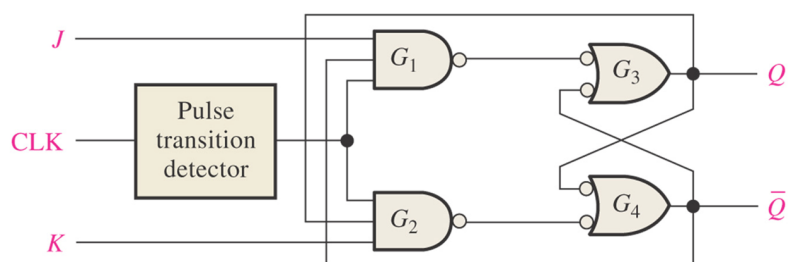


图 19: JK触发器

JK触发器状态表

J	K	状态
0	0	不变
0	1	复位
1	0	置位
1	1	转换

注意看有无bubble，看是上升沿还是下降沿

5.2.3 应用

1. 并行数据传输：接同一时钟
2. 分频：JK均接高，遇上升沿才触发，故可实现
3. 计数器：也相当于分频

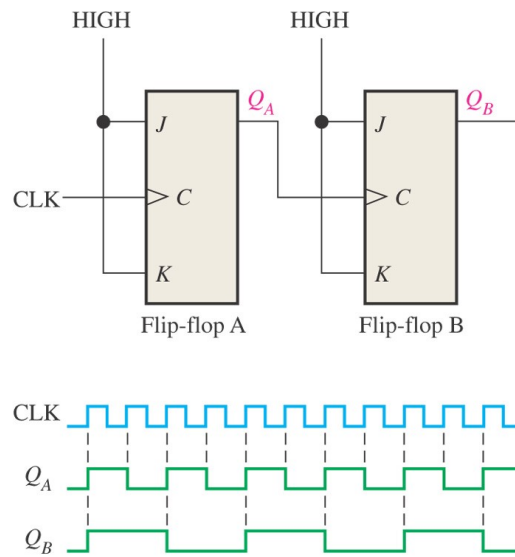


图 20: 分频器

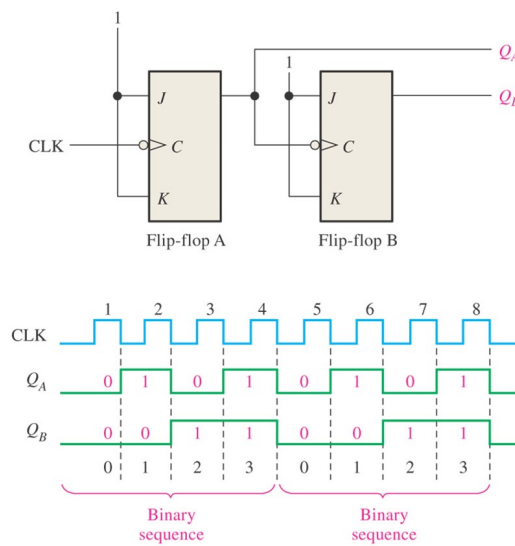


图 21: 计数器

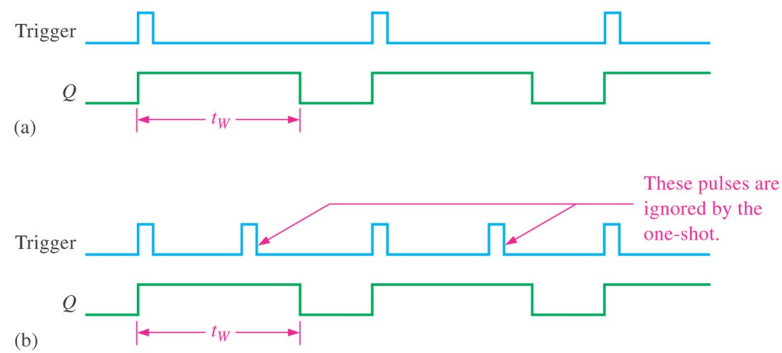


图 22: 单稳态触发器（不可重复触发）

5.3 单稳态触发器

5.4 555计时器

- 单稳态触发器(mono-stable one-shot)
- 非稳态多谐振荡器(astable multi-vibration oscillator)

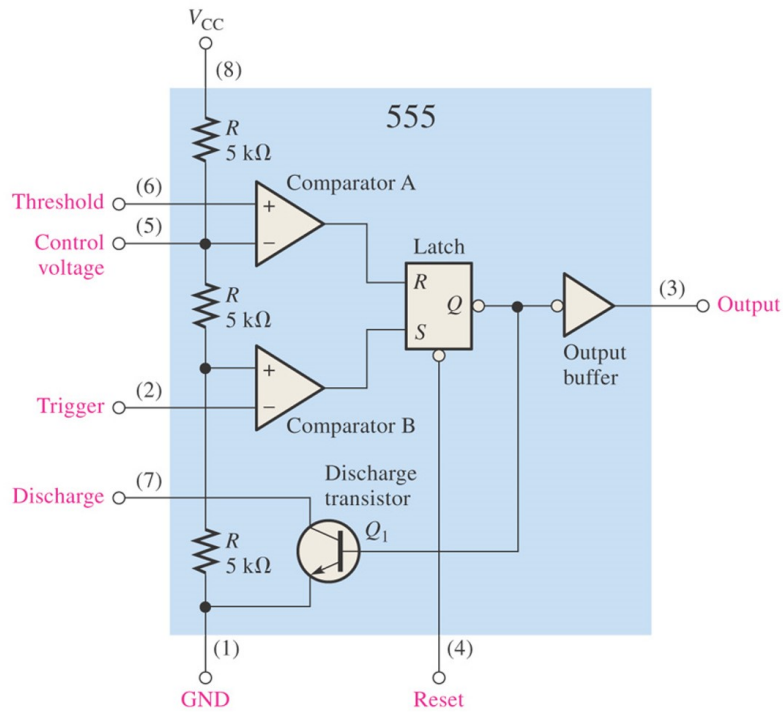


图 23: 555计时器

$$f = \frac{1.44}{(R_1 + 2R_2)C_1}$$

$$DC = \frac{R_1 + R_2}{R_1 + 2R_2} \times 100\%$$

5.5 移位寄存器

5.5.1 约翰逊计数器

计数范围 $M = 2N$

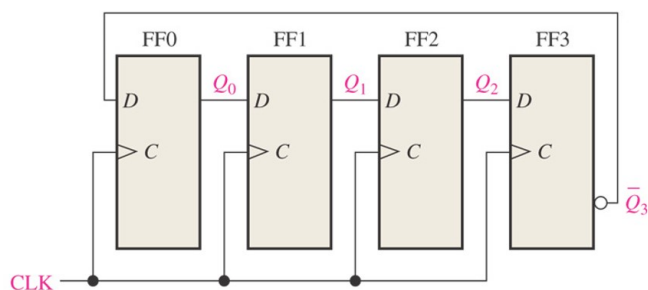


图 24: 约翰逊(Johnson)计数器

计数	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1

5.5.2 环计数器

计数范围 $M = N$

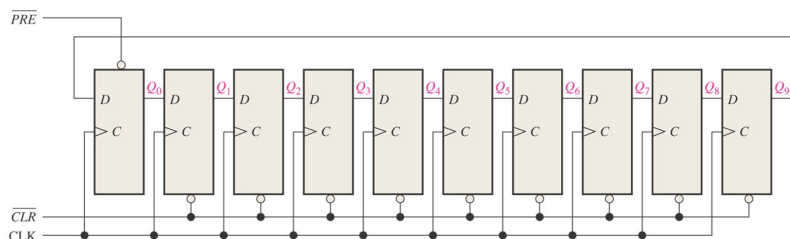


图 25: 环计数器

初始置为1000000000

5.6 计数器

5.6.1 同步异步计数器

RO(1)与RO(2)同时为高时清零, CLK A控制二进制计数器(Q_0), CLK B控制八进制计数器($Q_1 \sim Q_3$), 故将 Q_0 输出与八进制计数器相连可得十六进制计数器

\overline{LOAD} 为低时读取数据, ENT、ENP为使能端, 同时高电平有效, RCO为进位端

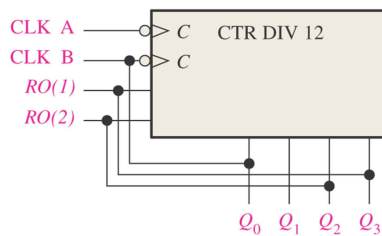


图 26: 16进制计数器

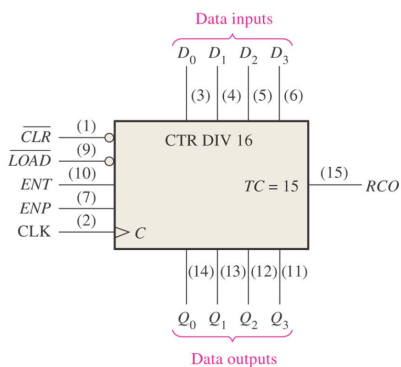


图 27: 4位同步二进制计数器

5.6.2 应用

6 电路设计

6.1 电路分类

1. 摩尔(Moore)电路
2. 米勒(Mealy)电路

6.2 设计步骤

基于状态转移表格的方法

1. 状态图
2. 次态表
3. 触发器转移表

Q^n	Q^{n+1}	J	K
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

4. 触发器JK卡诺图

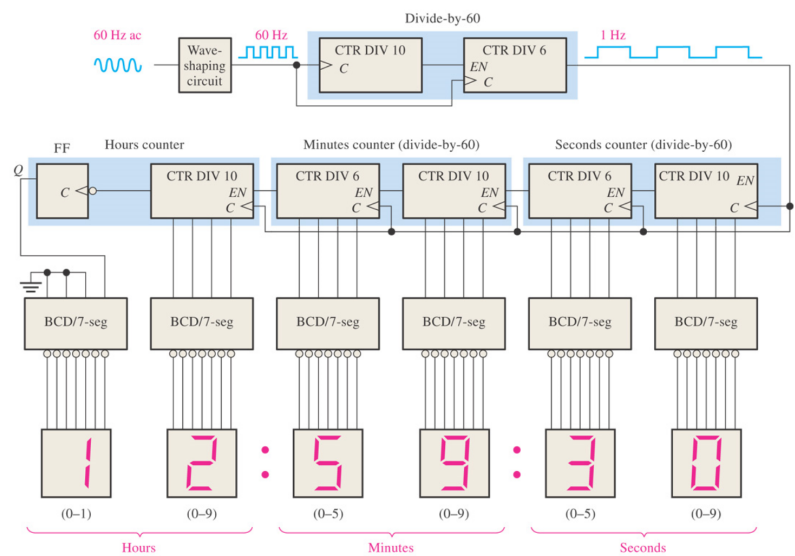


图 28: 时钟

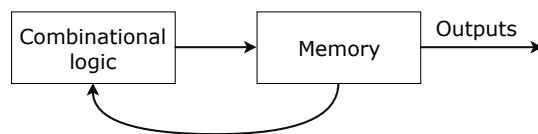


图 29: 触发器

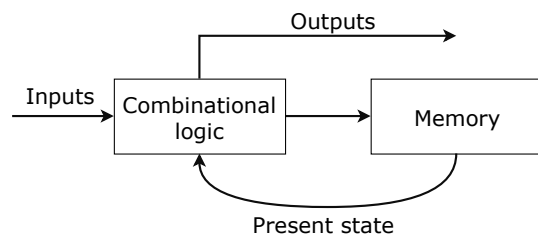


图 30: 触发器

6. 时序电路

1. 状态表

2. 次态卡诺图

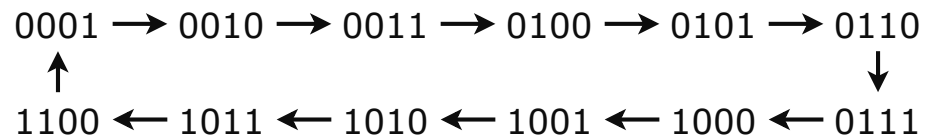
3. 状态方程 $Q^{n+1} = J \overline{Q^n} + \overline{K} Q^n$

4. 驱动方程

5. 时序电路

目的：用JK触发器实现一个12进制同步计数器

1. 状态转换图



2. 确定电路所需触发器数目

由于 $2^4 = 16 > 12$ ，故需要4个JK触发器

3. 次态卡诺图

		Q_1Q_0			
		00	01	11	10
Q_3Q_2	00	X	0010	0100	0011
	01	0101	0110	1000	0111
	11	0001	X	X	X
	10	1001	1010	1100	1011

4. 触发器状态方程，由卡诺图可得

$$\begin{aligned}Q_0^{n+1} &= \overline{Q_0} \\Q_1^{n+1} &= Q_0 \overline{Q_1} + \overline{Q_0} Q_1 \\Q_2^{n+1} &= Q_0 Q_1 \overline{Q_2} + \overline{Q_1} Q_2 \overline{Q_3} + \overline{Q_0} Q_2 \overline{Q_3} \\Q_3^{n+1} &= \overline{Q_2} Q_3 + Q_0 Q_1 Q_2 \overline{Q_3}\end{aligned}$$

5. 触发器驱动方程，由

$$Q^{n+1} = J \overline{Q^n} + \overline{K} Q^n$$

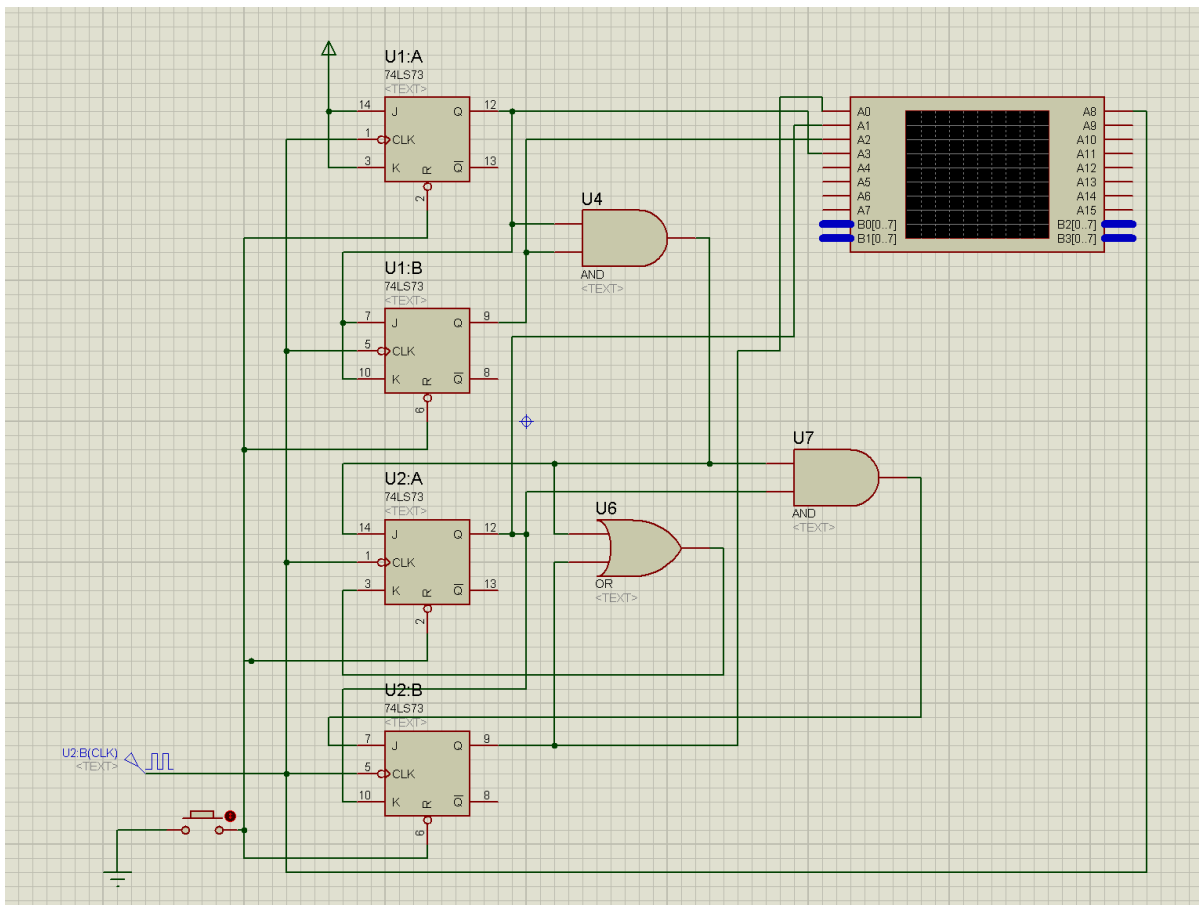
将状态方程整理为上式形式，可得

$$\begin{aligned}J_0 &= 1 & K_0 &= 1 \\J_1 &= Q_0 & K_1 &= Q_0 \\J_2 &= Q_1 Q_0 & K_2 &= \overline{Q_3} \overline{Q_1} + \overline{Q_3} \overline{Q_0} = \overline{Q_3} + Q_1 Q_0 \\J_3 &= Q_2 Q_1 Q_0 & K_3 &= Q_2\end{aligned}$$

6. 检查自启动

当输入为1111和0000时，可自动跳转至0001；输入为1101时，跳转至0010；输入为1110时，跳转至0011

Proteus电路图连接如下



仿真结果如下

