



中国海洋大学
OCEAN UNIVERSITY OF CHINA

数字逻辑 08 时序线路分析

一个完整的组合线路案例和触发器

杨永全

计算机科学与技术学院

目录

1. 课程目标
2. 课程内容
3. 课堂讨论
4. 课堂总结
5. 作业

1.课程目标

1. 目标

1. 根据一个组合线路设计案例，全面掌握组合线路设计，融会贯通
2. 了解触发器

2.课程内容

1. 一个完整的加法器

1.1 位二进制全加器

最简单的二进制加法器，很明显，就是加数和被加数都只有 1 位，再考虑低位对自己进位问题，那么输入就一共有三个数字。而输出，除了和之外，还有本位相加后对高位的进位（可能是 0，也可能是 1），这样的一种加法器，被称为 1 位二进制全加器，其计算过程如图所示。当然，如果当前要计算的两个数，已经是最低位了，那么默认的低位对自己的进位就是 0。

$$\begin{array}{r} 1 \\ + 1 0 \\ \hline 1 0 \end{array}$$

3个输入

2个输出

1. 一个完整的加法器

1.1 位二进制全加器

通过分析可以看出，1 位二进制全加器，一共有三个输入，分别是加数、被加数和低位对自己的进位。而输出，共有两个，分别是和，以及自己（本位）对高位的进位。这样就可以列出 1 位二进制全加器的真值表，共有 8 种情况。其中， A 、 B 和 C_{i-1} 是输入， S 和 C_i 是输出。 A 和 B 分别表示加数和被加数， C_{i-1} 表示低位对本位的进位， S 表示和， C_i 表示本位对高位的进位。我们可以列出真值表。

1. 一个完整的加法器

1.1 位二进制全加器

编号	A	B	C_{i-1}	S	C_i
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

1. 一个完整的加法器

1.1 位二进制全加器

根据真值表，我们就可以得到输出 S 和 C_i 与输入 A 、 B 和 C_{i-1} 之间的关系。

$$S = \Sigma(1, 2, 4, 7) \quad (1)$$

$$C_i = \Sigma(3, 5, 6, 7) \quad (2)$$

1. 一个完整的加法器 1.1 位二进制全加器

此时，可以画出 S 和 C_i 的卡诺图。

BC_{i-1}					
		00	01	11	10
A	0		1		2
	1	4		7	

(a) 输出 S 的卡诺图

BC_{i-1}					
		00	01	11	10
A	0			3	
	1		5	7	6

(b) 输出 C_i 的卡诺图

1. 一个完整的加法器

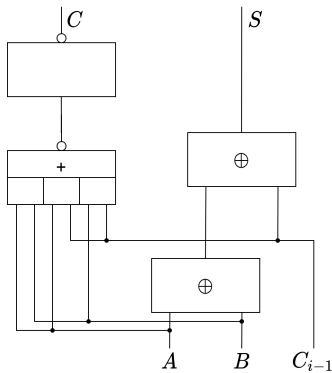
1.1 位二进制全加器

通过卡诺图可以看出，输出 S 已经不能再化简了，但是 C_i 却可以得到更简单的形式。这样，通过化简，得到 S 和 C_i 的最终表达式。

$$\begin{aligned} S &= \Sigma(1, 2, 4, 7) \\ &= \overline{A} \cdot \overline{B} \cdot C_{i-1} + \overline{A} \cdot B \cdot \overline{C_{i-1}} + A \cdot \overline{B} \cdot \overline{C_{i-1}} + A \cdot B \cdot C_{i-1} \\ &= C_{i-1} \cdot (\overline{A \oplus B}) + \overline{C_{i-1}} \cdot (A \oplus B) \\ &= C_{i-1} \oplus (A \oplus B) \\ C_i &= \Sigma(3, 5, 6, 7) \\ &= A \cdot B + B \cdot C_{i-1} + A \cdot C_{i-1} \end{aligned} \tag{3}$$

1. 一个完整的加法器 1.1 位二进制全加器

得到了 S 和 C_i 的表达式后，就可以画出线路图。



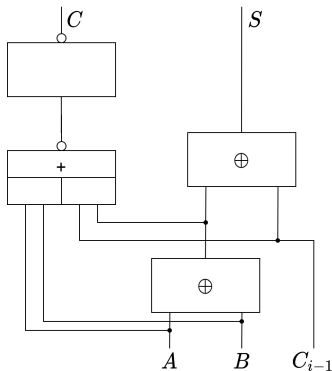
1. 一个完整的加法器 1.1 位二进制全加器

实际上，这个线路还可以进行简单的优化，以得到更好的效果。优化的方法是：表达式 C_i 还可以进行简单的函数变换。

$$\begin{aligned}C_i &= \Sigma(3, 5, 6, 7) \\&= A \cdot B + B \cdot C_{i-1} + A \cdot C_{i-1} \\&= A \cdot B + (A + B) \cdot C_{i-1} \\&= A \cdot (B + \bar{B} \cdot C_{i-1}) + B \cdot (A + \bar{A} \cdot C_{i-1}) \\&= A \cdot B + A \cdot \bar{B} \cdot C_{i-1} + \bar{A} \cdot B \cdot C_{i-1} \\&= A \cdot B + (A \oplus B) \cdot C_{i-1}\end{aligned}\tag{4}$$

1. 一个完整的加法器 1.1 位二进制全加器

通过变换，输出 S 和 C_i 共用了一个组件 $A \oplus B$ ，通常情况下，可以节省一些门电路。
优化后的线路图如图所示。



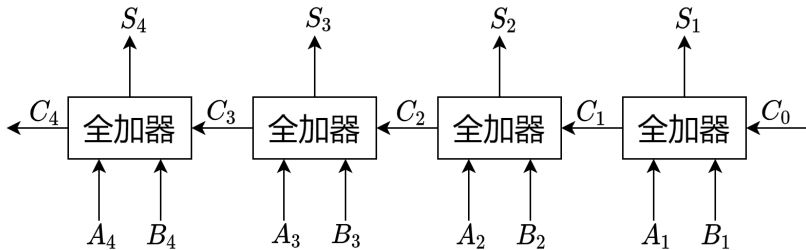
1. 一个完整的加法器 2. 多位二进制加法器

很明显，如果线路只能实现 1 位的二进制加法，基本上它什么也做不了。为了让线路能够实现多位二进制加法，需要为它设计更加复杂的逻辑。

例如，如果需要计算 $1011+1000$ ，根据二进制加法规则，很自然的做法就是先用一个全加器，算出最后一位的两个数 ($1+0$) 相加的结果，得到本位的和，以及对上一位的进位。把和输出后，对上一级的进位作为上一级的进位，再输入到上一级的全加器中。

1. 一个完整的加法器 2. 多位二进制加法器

其逻辑图如图所示。若 C_4 是最后一位进位，则将和 $S_1 - S_4$ 一起，作为输出，组成最后的结果。



1. 一个完整的加法器 2. 多位二进制加法器

使用这种方法有一些问题难以解决。

1. **加法器的位数必须固定。**由于该加法器，由多个 1 位二进制加法器组合而成，每一个加法器计算 1 位，这样就意味着，一个加法器能够算几位的加法，是需要提前设计好的，并且每次输入的加数和被加数，其长度都不能超过加法器的位数，否则，加数和被加数的高位将无法被输入到线路中。
2. **高位必须晚于低位到来。**因为高位的计算，依赖于低位的计算结果，来决定本位计算时，是否有进位，从而决定 C_{i-1} 的值，如果高位加数和被加数早于低位到来，在低位有进位的情况下，将产生计算错误。

上述问题，使用时序线路进行设计后，将可以得到解决。

1. 一个完整的加法器

3.设计一个简单的十进制计算器

有了多位的二进制加法器，就可以实现十进制的加法器。十进制的加法，也是按照从低位到高位，每一位单独进行，而且，进位也只能是 0 或者 1。最自然的想法是，用二进制来表示十进制，去计算每一位的十进制加法。

将十进制数转换成二进制数，或者说，用二进制数表示十进制数，去进行计算，这种转换或者表示方法，就是编码。最常见的编码方式之一，是自然编码。自然编码的意思是，按照二进制所表示的大小，对十进制进行编码。为了表示 0-9 这 10 个数字，需要用到 4 位的二进制数，但是 4 位的二进制数，不但可以表示 0-9，还可以表示 10-15。所以，自然编码，并不是常规意义上的二进制到十进制的编码，它只是把二进制的编码当成是真实的数字而已。

1. 一个完整的加法器 3.设计一个简单的十进制计算器

编号	B_3	B_2	B_1	B_0	十进制数
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	1	1	0	0	12
13	1	1	0	1	13
14	1	1	1	0	14
15	1	1	1	1	15

1. 一个完整的加法器 3.设计一个简单的十进制计算器

事实上，使用二进制来进行十进制的编码，用的最多的，还是 BCD 码。BCD 码 (Binary-Coded Decimal)，指的是用 4 位二进制数来表示 1 位十进制数中的 0-9 这 10 个数码，是一种二进制的数字编码形式，用二进制编码的十进制代码。BCD 码这种编码形式利用了一个四位的二进制数来储存一个十进制的数码，使二进制和十进制之间的转换得以快捷的进行。

BCD 码因为使用了 4 位二进制数来表示 10 个十进制数，意味着有 $6(2^4 - 10)$ 个编码没有被使用。

常见的 BCD 码主要有：8421 码、余三码和格雷码等。

1. 一个完整的加法器 3.设计一个简单的十进制计算器

8421 码。有权码。因为代码中从左至右看每一位“1”分别代表数字“8”“4”“2”“1”，故得名 8421 码。

余三码。无权码。余三码（余 3 码）是由 8421BCD 码加上 0011 形成的一种无权码，由于它的每个字符编码比相应的 8421 码多 3，故称为余三码。

5421 和 2421 码。有权码。它们从高位到低位的权值分别为 5、4、2、1 和 2、4、2、1。这两种有权 BCD 码中，有的十进制数码存在两种加权方法，例如，5421 BCD 码中的数码 5，既可以用 1000 表示，也可以用 0101 表示；2421 BCD 码中的数码 6，既可以用 1100 表示，也可以用 0110 表示。这说明 5421 BCD 码和 2421 BCD 码的编码方案都不是惟一的。

Gray 码。无权码。Gray 码（格雷码）也称循环码，在一组数的编码中，若任意两个相邻数的代码只有一位二进制数不同。Gray 码的编码方案有多种。这种代码除了具有单位距离码的特点外，还有一个特点就是具有反射特性，即按表中所示的对称轴为界，除最高位互补反射外，其余低位数沿对称轴镜像对称。利用这一反射特性可以方便地构成位数不同的 Gray 码。

1. 一个完整的加法器 3.设计一个简单的十进制计算器

十进制数	8421	5421	2421	余三码	格雷码
0	0000	0000	0000	0011	0000
1	0001	0001	0001	0100	0001
2	0010	0010	0010	0101	0011
3	0011	0011	0011	0110	0010
4	0100	0100	0100	0111	0110
5	0101	1000	1011	1000	0111
6	0110	1001	1100	1001	0101
7	0111	1010	1101	1010	0100
8	1000	1011	1110	1011	1100
9	1001	1100	1111	1100	1101

1. 一个完整的加法器 4.七段译码器

译码器的定义是：将一种代码翻译成另外一种代码的线路的统称。这种线路实际上并没有实际的逻辑功能，只是简单的将一种编码翻译成另外一种编码。

这里介绍一种译码器，叫做：七段译码器。

计算器上用来显示数字的 LED 屏幕，通过仔细观察可以发现，每个数字都是使用七个线段组成的，如图所示。如果给这七个线段编上号码，分别用 ABCDEFG 来表示，那么只需要给 ABCDEFG 都赋值为 0 或者 1，就可以实现数字的显示了。例如，如果需要显示一个 4，则输入的编码应该是 0100111，这样，B、E、F 和 G 亮起来，其他的线段熄灭，就是数字 4 的形状。这种编码被称为七段编码。

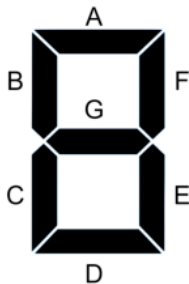
可以很容易想到的是，如果购买一块只能显示一个数字的 LED 面板，那么它的控制器一定有 7 个引脚，便于输入七段编码。如果想要让它按照编码的格式显示正确的数字，需要给它正确的七段编码。从数字编码到七段编码，需要先设计一个七段译码器。

1. 一个完整的加法器

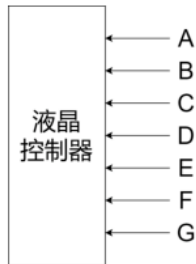
4. 七段译码器



(a) 液晶显示器



(b) 七段编码



(c) 七段编码控制器

1. 一个完整的加法器 4.七段译码器

如果采用 8421 码对十进制数进行编码，那么七段译码器的主要任务，是把 8421 码翻译成七段编码。根据要求，可以很容易写出七段译码器的真值表，如表所示。

十进制	8421	B_8	B_4	B_2	B_1	A	B	C	D	E	F	G
0	0000	0	0	0	0	1	1	1	1	1	1	0
1	0001	0	0	0	1	0	0	0	0	1	1	0
2	0010	0	0	1	0	1	0	1	1	0	1	1
3	0011	0	0	1	1	1	0	0	1	1	1	1
4	0100	0	1	0	0	0	1	0	0	1	1	1
5	0101	0	1	0	1	1	1	0	1	1	0	1
6	0110	0	1	1	0	1	1	1	1	1	0	1
7	0111	0	1	1	1	1	0	0	0	1	1	0
8	1000	1	0	0	0	1	1	1	1	1	1	1
9	1001	1	0	0	1	1	1	0	1	1	1	1

1. 一个完整的加法器 4.七段译码器

得到真值表后，就可以列出 ABCDEFG 的表达式了。

$$\begin{aligned}A &= \Sigma(0, 2, 3, 5, 6, 8, 9) \\B &= \Sigma(0, 4, 5, 6, 8, 9) \\C &= \Sigma(0, 2, 6, 8) \\D &= \Sigma(0, 2, 3, 5, 6, 8, 9) \\E &= \Sigma(0, 1, 3, 4, 5, 6, 7, 8, 9) \\F &= \Sigma(0, 1, 2, 3, 4, 7, 8, 9) \\G &= \Sigma(2, 3, 4, 5, 6, 8, 9)\end{aligned}\tag{5}$$

1. 一个完整的加法器 5.输入板

基于上述的工作，已经可以实现，给定一个数字的编码，可以显示在液晶显示屏上。但是一个新的问题同时出现了：应该如何把需要相加的数字，输入到线路中？常见的输入，基本上就是键盘，现在市面上购买的计算器，也都是通过键盘输入的，这里把问题简化一下，使用最简单的输入板输入数字，这个输入板，只有 0-9 这 10 个数字。每次输入 1 个数字，输入 2 个数字后，把结果，显示在液晶屏上。现在的问题是：当我们按下输入板上的按键时，到底发生了什么？毫无疑问的是，输入板将输出一串信号，如果没有意外，它输出的信号也是二进制形式的。如果它输出的信号，其编码格式也是 8421 码，那么可以直接接入七段译码器后接入 LED 显示器。如果输入板无法输出 8421 码，那么还需要将输入板的编码，翻译成 8421 码，再接入七段译码器。假设输入板输出的编码格式，形如 0000000000、0000000001、0000000010 这样的格式，那么还需要将这样的编码格式，转换为 8421 码。

1. 一个完整的加法器 5.输入板

对于该问题，应该有两种做法，第一种：输入板的编码直接转换为 7 段码，输出到显示屏上，同时转换成 8421 码，参与后续的运算；第二种：输入板编码转换成 8421 码，8421 码作为统一的编码格式，参与后续的运算，并转换成 7 段码，输出到显示屏上。

尝试讨论一下，这两种做法的优缺点。(留在课堂讨论环节)

2. 时序线路

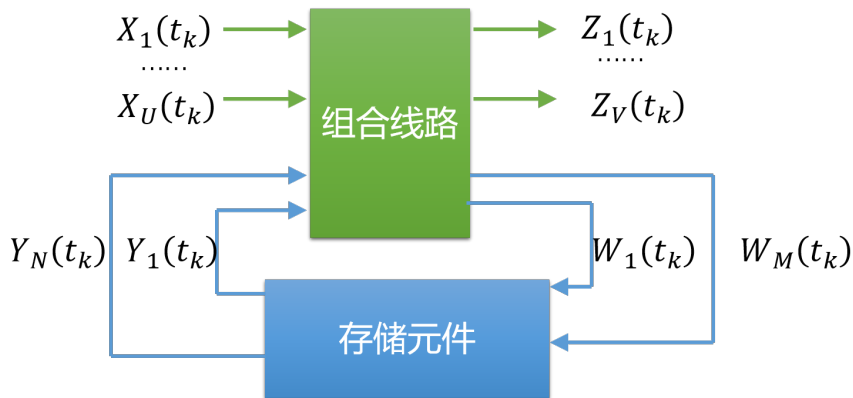
先回忆一下组合线路



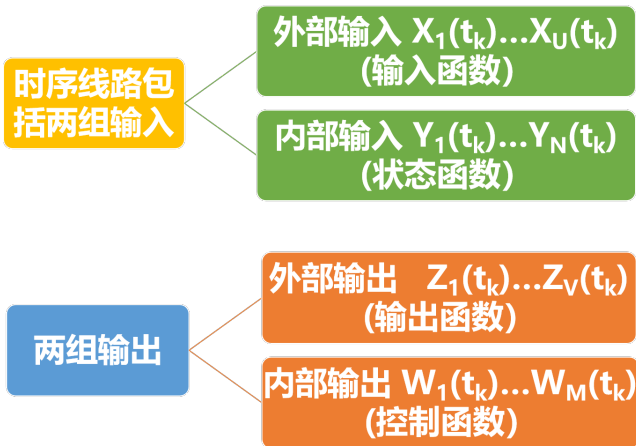
这是一个有 U 个输入和 V 个输出的组合线路，可以看出，某一时刻的输出仅与该时刻的输入有关。

2. 时序线路

时序线路



2. 时序线路



2. 时序线路

时序线路的表示方法

输出函数 $Z_i(t_k)=f_i[X_1(t_k)\dots X_u(t_k),y_1(t_k)\dots y_n(t_k)]$

次态函数 $Y_j(t_k)=q_j[X_1(t_k)\dots X_u(t_k),y_1(t_k)\dots y_n(t_k)]$

2. 时序线路

总结一下组合线路和时序线路的区别

组合线路	由门电路组成	输出函数仅与该时刻输入有关	用输出函数描述
时序线路	由门电路及记忆元件构成	输出函数与输入及线路状态有关	用输出及次态函数描述

2. 时序线路

现态与次态关系

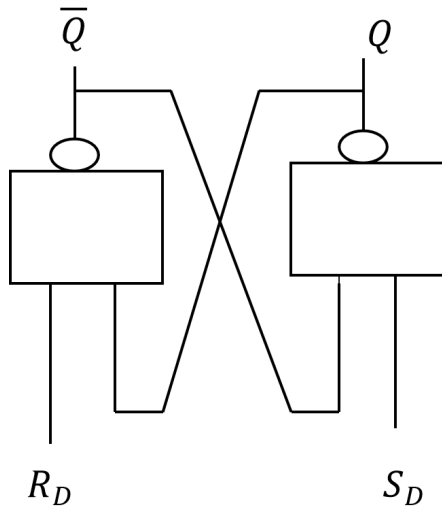
	t=0	t=1	t=2
输入函数	$X_1(0) \dots X_u(0)$	$X_1(1) \dots X_u(1)$	$X_1(2) \dots X_u(2)$
现态函数	$y_1(0) \dots y_n(0)$	$y_1(1) \dots y_n(1)$	$y_1(2) \dots y_n(2)$
控制函数	$W_1(0) \dots W_m(0)$	$W_1(1) \dots W_m(1)$	$W_1(2) \dots W_m(2)$
次态函数	$Y_1(0) \dots Y_n(0)$	$Y_1(1) \dots Y_n(1)$	$Y_1(2) \dots Y_n(2)$
输出函数	$Z_1(0) \dots Z_u(0)$	$Z_1(1) \dots Z_u(1)$	$Z_1(2) \dots Z_u(2)$

3. 触发器

触发器的特点

- 1、触发器必须具备两个稳态，用以记忆两个特征值“0”和“1”。
- 2、触发器的状态要能够预置，即具有置位（置 1）复位（置 0）控制端。
- 3、触发器必须能在外部信号激励下进行状态的转换。

3. 触发器 1.基本触发器



3. 触发器 1. 基本触发器

交叉连接：一个与非门的输出恰好是另一个与非门的输入， Q 与 \bar{Q} 状态相反，也就是互斥。

S_D : 直接置位端， $Q = 1$ 为置位状态

R_D : 直接复位端， $Q = 0$ 为复位状态

1) $S_D = 0, R_D = 1 \quad Q = 1$

2) $S_D = 1, R_D = 0 \quad Q = 0$

3) $S_D = 1, R_D = 1 \quad Q$ 保持原来状态

4) $S_D = 0, R_D = 0 \quad Q = \bar{Q}$ 不允许，所以为约束条件

3. 触发器 1.基本触发器

基本触发器的特征函数表

S_D	R_D	Q	Q^{n+1}
0	1	ϕ	1
1	0	ϕ	0
1	1	0	0
1	1	1	1
0	0	ϕ	ϕ

S_D	R_D	Q^{n+1}
0	1	1
1	0	0
1	1	Q
0	0	ϕ

3. 触发器 1.基本触发器

基本触发器的特征函数

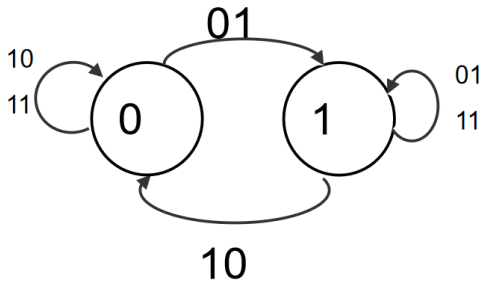
S_D	R_D	Q^{n+1}
0	1	1
1	0	0
1	1	Q
0	0	ϕ

$$Q^{n+1} = R_D \overline{S_D} + \overline{R_D} \overline{S_D} + R_D S_D Q$$
$$= \overline{S_D} + R_D Q \quad (\overline{S_D} \overline{R_D} = 0)$$

3. 触发器 1. 基本触发器

基本触发器的激励表与状态图

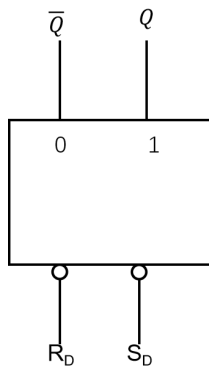
Q	Q^{n+1}	S_D	R_D
0	0	1	Φ
0	1	0	1
1	0	1	0
1	1	Φ	1



3. 触发器 1.基本触发器

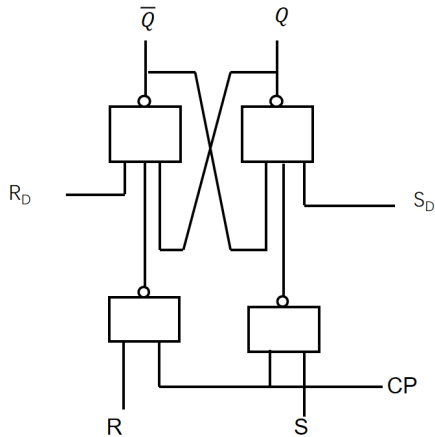
基本触发器的逻辑符号与问题

问题: R_D, S_D 不一定同时到来。



3. 触发器 2.RS 触发器

线路图



3. 触发器 2.RS 触发器

RS 触发器的特征函数表和激励表

R	S	Q^{n+1}
0	0	Q
0	1	1
1	0	0
1	1	ϕ

Q	Q^{n+1}	R	S
0	0	ϕ	0
0	1	0	1
1	0	1	0
1	1	0	ϕ

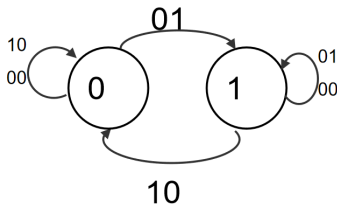
$$Q^{n+1} = \overline{R} \overline{S} Q + \overline{R} S + RS = S + \overline{R} Q \quad (RS = 0)$$

3. 触发器 2.RS 触发器

RS 触发器的特征表达式和状态图

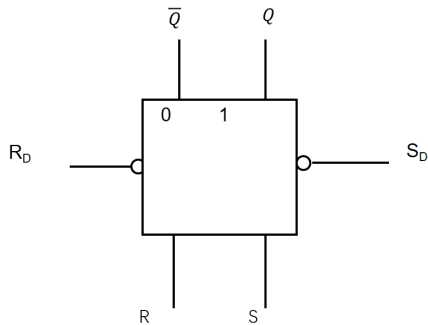
$$Q^{n+1} = \overline{R} \overline{S} Q + \overline{R} S + RS = S + \overline{R} Q \quad (RS = 0)$$

R	S	Q^{n+1}
0	0	Q
0	1	1
1	0	0
1	1	Φ



3. 触发器 2.RS 触发器

RS 触发器的逻辑符号



3.课堂讨论

1. 问题

如果输入板不是 8421 编码形式，参与运算之前，需要将其转换为 8421 码，这时有两种做法，第一种：输入板的编码直接转换为 7 段码，输出到显示屏上，同时转换成 8421 码，参与后续的运算；第二种：输入板编码转换成 8421 码，8421 码作为统一的编码格式，参与后续的运算，并转换成 7 段码，输出到显示屏上。尝试讨论一下，这两种做法的优缺点。

RS 触发器有什么问题？

4.课堂总结

1. 课堂总结

笔记

现在可以总结自己的笔记，提炼大纲，回顾课程。

总结

还可以将课程的总结、心得记录在总结区。

5.作业

1. 题目

试用卡诺图法化简下列不完全定义函数（每题 3 分，共 9 分）

(1) $F(A, B, C, D) = \sum(0, 3, 5, 6, 8, 13)$ 约束方程: $\sum \Phi(1, 4, 10) = 0$

(2) $F(A, B, C, D) = \overline{A}\overline{B}\overline{C} + ABC + \overline{A}\overline{B}C\overline{D}$ 约束方程: $A \oplus B = 0$

(3) $F(A, B, C, D) = ABC\overline{C} + A\overline{B}\overline{C} + \overline{A}\overline{B}C\overline{D} + A\overline{B}C\overline{D}$ 约束条件: A、B、C、D 不可能出现相同值



中国海洋大学
OCEAN UNIVERSITY OF CHINA

问答环节