四、路径敏化法

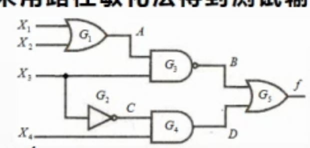
- 测试码的产生有多种方式，按其生成的手段可归为两大类：代数法及算法。
- 代数法是根据描述电路功能的布尔等式求解出测试矢量，其主要缺点是占用存储空间大，当电路夏杂时，不存在或很难求得布尔等式。
- 算法则是使用各种电路机理跟踪、敏化路径，使得故障效应传播到电路的原始输出，然后给原始输入分配满足故障传播和生成条件的值。
- 一种简单的测试码产生方式是信号传播路径敏化法。
- 路径敏化法和信号传播的基本原理是相当简单的，为使输入 $\overrightarrow{X_i}$ 能够检测信号点As-a-j(j=0，1的故障)，$\overrightarrow{X_i}$ 必须保持正常（无故障）电路的信号点A为 $\bar{j}$ 值（暴露故障），这只给出了对检测该故障的必要条件，但并不是充分的。同时，故障信号的效应必须由故障部位或故障原始点沿某条路径传播到输出 。

路径敏化法
(鼠标滑过播放视频)

四、路径敏化法

- 测试码的产生有多种方式，一种简单的测试码产生方式是路径敏化法和信号传播。
- 路径敏化法和信号传播的基本原理是相当简单的，为使输入 $\overline{X_i}$ 能够检测信号点As-a-j(j=0，1的故障)，$\vec{X_i}$ 必须保持正常（无故障）电路的信号点A为 $\bar{j}$ 值（暴露故障），这只给出了对检测该故障的必要条件，但并不是充分的。同时，故障信号的效应必须由故障部位或故障原始点沿某条路径传播到输出 。

采用路径敏化法得到测试输入码



① As-a-1

1.反映故障 $X_1+X_2=0$　$\therefore X_1=X_2=0$

2.传播故障 $X_3=1$，$X_4\overline{X_3}=0$

3.确定输入 $\therefore X_4\overline{X_3}=0$，$X_3=1$，$\therefore$ $X_4=0,1$

测试输入0010或0011。

正确输出为1，故障输出为0。

② Cs-a-1。

反映故障 $X_3=1$

传播故障 $X_4=1$，$\overline{(X_1+X_2)X_3}=0$

确定输入 $X_3=1$　$\therefore \overline{X_1+X_2}=0$　$X_1+X_2=1$

即 $X_1X_2=11,10,01$

测试输入0111，1011，1111。

正确输出为0，故障输出为1。

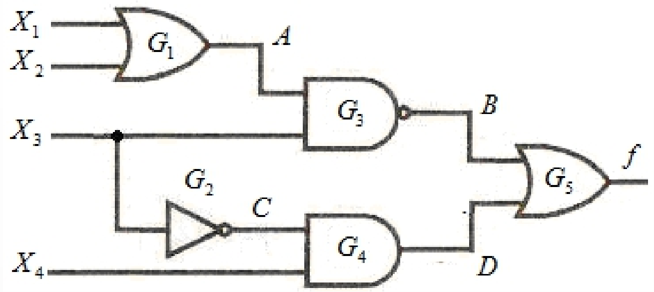
③ Cs-a-0。

反映故障：$X_3=0$，但是G_3输出1，$f\equiv1$, 故障不可测，但是该故障是冗余的，因为 $X_3=0$ 时，不论C点是否有故障，或其他输入的什么，都有$f\equiv1$。冗余过程为：$X_3=1$，C点正常信号与现在的s-a-

路

在组合电
的一
- ①反
障部
故障
1）。
- ②传
部位
定这
以便
- ③线
得产

采用路径敏化法得到测试输入码



① As-a-1

1.反映故障 $X_1+X_2=0$ $\therefore X_1=X_2=0$

2.传播故障 $X_3=1$, $X_4\overline{X_3}=0$

3.确定输入 $\therefore X_4\overline{X_3}=0$, $X_3=1$, $\therefore$ $X_4=0,1$

测试输入0010或0011。

正确输出为1，故障输出为0。

② Cs-a-1。

反映故障 $X_3=1$

传播故障 $X_4=1$, $\overline{(X_1+X_2)X_3}=0$

确定输入 $X_3=1$ $\therefore \overline{X_1+X_2}=0$ $X_1+X_2=1$

即 $X_1X_2=11,10,01$

测试输入0111，1011，1111。

正确输出为0，故障输出为1。

③ Cs-a-0。

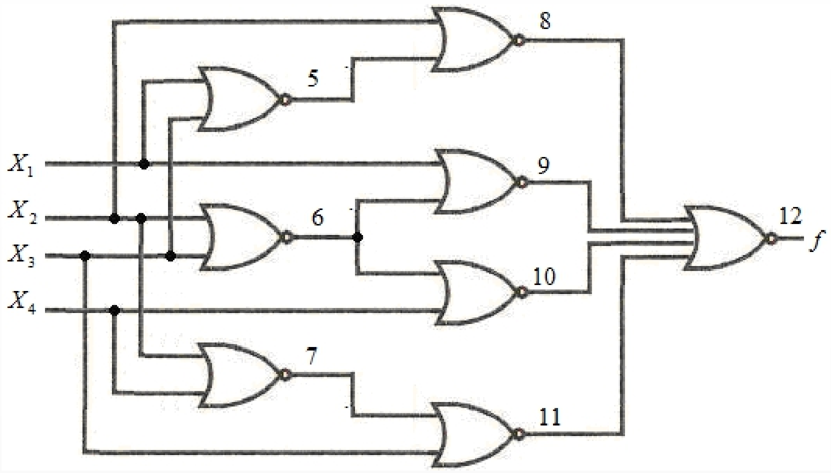
反映故障：$X_3=0$，但是G_3输出1，$f=1$，故障不可测，但是该故障是冗余的，因为 $X_3=0$ 时，不论C点是否有故障，或其他输入的什么，都有$f=1$。冗余过程为：$X_3=1$，C点正常信号与现在的s-a-0为相同信号$X_3=0$，f=1，C点的“0”故障对信号的正确性无影响。

路径敏化法步骤

在组合电路中，为了产生检测故障的一个测试码。

- ①反映故障：确定输入，在故障部位产生需要的值（s-a-1故障时为0，s-a-0故障时为1）。
- ②传播故障：选择一条从故障部位到达输出的路径，并且确定这条路经的另一些信号值，以便将故障传播到输出。
- ③线合理性：确定输入值，使得产生②中规定的信号值。

施奈德电路



- ① 让故障沿6→9→12传播，故需要敏化该路径。
- 反映故障：由于6s-a-0，故 $\overline{X_2+X_3}=1$ ，即 $X_2=X_3=0$ 。
- 传播故障：由于沿6→9→12，故且要求X_8，X_{10}，X_{11}=0。为使X_{10}=0，使 X_4为1，而X_4=1使X_7=0，$X_{11}=\overline{X_3+X_7}=1$，所以 X_{10} 为0，$X_{11}=1$，路径不能被敏化，故障不能被传播。不能沿6→9→12传播。
- ② 同理,故障不能沿6→10→12传播
- ③ 同时沿6→9→12和6→10→12传播。
- 反映故障：$X_2=X_3=0$ 。
- 传播故障：$X_1=X_4=0$，使6点故障同时传播到9、10，那么和是否为0呢？
- 由于X_1 ~ X_4 均为0，故 $X_8=0$,$X_{11}=0$ ，这样6点的故障被反映到f点，所以6s-a-0的测试码为（0，0，0，0）。

五、D算法

- D-算法就是一种多路径敏化法。
- D-算法是一种基于集合理论的算法，1966年由 Roth提出，属于多路径敏化算法。
- D算法，是拓扑结构测试中最经典的方法，也是最早实现自动化的测试生成算法之一。
- 它是完备的测试算法，它可以检测非冗余电路中所有可以检测的故障。虽然它是在20世纪60年代提出的，而且被改正过多次，但是，许多新的测试方法都是在它的基础上发展起来的。而且一直沿用至今。
- D算法在具体应用时，计算工作量很大，尤其是对大型的组合电路计算时间很长，原因是在作敏化通路的选择时其随意性太大，特别是在考虑多通路敏化时各种组合的情况太多，然而真正"有效"的选择往往较少，做了大量的返回操作。
- 改进的算法，如PODEM和FAN算法，有效地减小了返回次数，提高了效率。
- D-cube

多路径敏化D算法
(鼠标滑过播放视频)

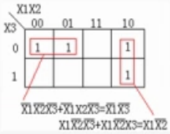
五、D算法

- D-算法就是一种多路径敏化法。
- D-算法是一种基于集合理论的算法，1966年由 Roth提出，属于多路径敏化算法。
- D算法，是拓扑结构测试中最经典的方法，也是最早实现自动化的测试生成算法之一。
- 它是完备的测试算法，它可以检测非冗余电路中所有可以检测的故障。虽然它是在20世纪60年代提出的，而且被改正过多次，但是，许多新的测试方法都是在它的基础上发展起来的。而且一直沿用至今。
- D算法在具体应用时，计算工作量很大，尤其是对大型的组合电路计算时间很长，原因是在作敏化通路的选择时其随意性太大，特别是在考虑多通路敏化时各种组合的情况太多，然而真正"有效"的选择往往较少，做了大量的返回操作。
- 改进的算法，如PODEM和FAN算法，有效地减小了返回次数，提高了效率。
- D-cube

（1）布尔函数的立方表示法

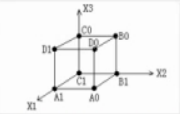
- 表示一个布尔函数通常有几种办法：
- 真值表表示法（卡诺图）
- 几何表示法
- 立方表示法
- **真值表表示法（卡诺图）**

X1	X2	X3	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



- **几何表示法**
- 布尔函数可用集合体上所有使f=1的点的集合来表示，又可用所有使f=0的点的集合来表示。
- f=1；S1=｛A1，B1，C1，D1｝；
- f=0；S0=｛A0，B0，C0，D0｝。

$f=X1\overline{X2}+\overline{X1}\,\overline{X3}$

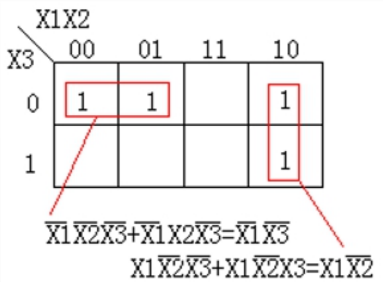


- **立方表示法**
- 显然对多变量函数，上述方法有各自的缺点。对于几何表示法，当自变量较多时，就成为n值空间，失去了直观性；真值表法又十分繁琐；立方表示法是介于二者之间的表示方法。利用函数的有1顶点构成的各个子立方体来表示布尔函数。例如有两个子立方体（A1,D1）和(C1,B1)，它们的坐标可以分别写成(X1,X2,X3)=(1,0,X)和(0,X,0)，每一个坐标叫做一个立方。

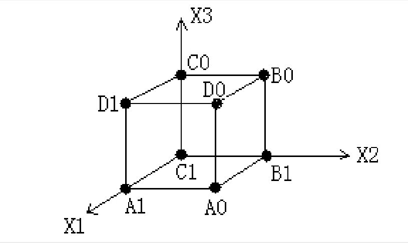
（1）布尔函数的立方表示法

- 表示一个布尔函数通常有几种办法：
- 真值表表示法（卡诺图）
- 几何表示法
- 立方表示法
- **真值表表示法（卡诺图）**

X1	X2	X3	f
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0



- **几何表示法**
- 布尔函数可用集合体上所有使f=1的点的集合来表示，又可用所有使f=0的点的集合来表示。
- f=1；S1=｛A1，B1，C1，D1｝；
- f=0；S0=｛A0，B0，C0，D0｝。



$$f=X1\overline{X2}+\overline{X1}\,\overline{X3}$$

- **立方表示法**
- 显然对多变量函数，上述方法有各自的缺点。对于几何表示法，当自变量较多时，就成为n值空间，失去了直观性；真值表法又十分繁琐；立方表示法是介于二者之间的表示方法。利用函数的有1顶点构成的各个子立方体来表示布尔函数。例如有两个子立方体（A1,D1）和(C1,B1)，它们的坐标可以分别写成(X1,X2,X3)=(1,0,X)和(0,X,0)，每一个坐标叫做一个立方。

（2）奇异立方

- 将f与$\overline{f}$的立方连同f的值一并列出。表中的每一行（包含自变量与函数值）称为函数的奇异立方。这个表实质上是函数真值表的压缩形式，这个表又称为函数f的奇异覆盖。

X1	X2	X3	f	
1	0	X	1	β_1
0	X	0	1	
1	1	X	0	β_0
0	X	1	0	

$$f = X1\overline{X2} + \overline{X1}\,\overline{X3}$$

D-算法的符号与运算

- D-算法是一种多路径敏化算法，这里引入符号D与 $\overline{D}$ 。
- D表示正常电路值为1而故障电路值为0的信号；
- $\overline{D}$表示正常电路值为0而故障电路值为1的信号。
- D立方代数运算与布尔代数相同，不过此时门的输入为4值（0，1，D ，$\overline{D}$）。
- 注意：

$$D+\overline{D}=1 \qquad \overline{D}+\overline{D}=\overline{D} \qquad D+D=D$$

$$D\cdot\overline{D}=0 \qquad D\cdot D=D \qquad \overline{D}\cdot\overline{D}=\overline{D}$$

- **D算法的运算规则**

与	0	1	D	$\overline{D}$
0	0	0	0	0
1	0	1	D	$\overline{D}$
D	0	D	D	0
$\overline{D}$	0	$\overline{D}$	0	$\overline{D}$

或	0	1	D	$\overline{D}$
0	0	1	D	$\overline{D}$
1	1	1	1	1
D	D	1	D	1
$\overline{D}$	$\overline{D}$	1	1	$\overline{D}$

（3）原始立方

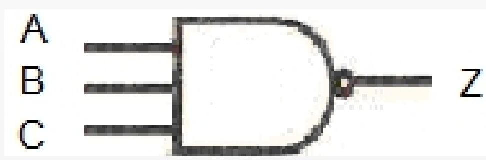
原始立方表示产生规定输出而要求的最小输入条件；

（4）故障原始-D立方

故障原始-立方是反映故障的最小条件。

（5）传播-立方

表示将故障进行传播，即对路径进行敏化，将元件中的一个（或几个）输入错误信号传播到输出所需要的最小输入条件。



原始立方（奇异立方）

A	B	C	Z
0	X	X	1
X	0	X	1
X	X	0	1
1	1	1	0

故障原始D立方

A	B	C	Z	故障
0	X	X	D	Zs-a-0
X	0	X	D	Zs-a-0
X	X	0	D	Zs-a-0
1	1	1	$\overline{D}$	Zs-a-1
0	1	1	D	As-a-1
1	0	1	D	Bs-a-1
1	1	0	D	Cs-a-1

传播D立方

A	B	C	Z
D	1	1	$\overline{D}$
1	D	1	$\overline{D}$
1	1	D	$\overline{D}$
D	D	1	$\overline{D}$
D	1	D	$\overline{D}$
1	D	D	$\overline{D}$
D	D	D	$\overline{D}$

（6）立方 αi，βj的交

- 立方代数的一个重要概念是两个立方的交。
- 定义：两个立方中对应具有相同值的位置上，相交值即为该值；如果在某个位置上，其中一个立方为不定值x时，相交值即为该位置上另一立方的值；若两个立方 αi，βj在某个位置上有确定的但不相等的值，则交不存在。
- 例：$\delta=0x10$，$\varepsilon=x110$，$\gamma=1x1x$
- 则$\delta\cap\varepsilon=0110$，$\varepsilon\cap\gamma=1110$，$\delta\cap\gamma=\varphi$（第一位不一致）

（7）算法实现

前面已介绍了若干定义，现在可说明-算法产生测试码的步骤。

① 根据电路组成，在计算机中建立所有门的原始故障传播立方。

② 选择被测故障的一个故障原始-立方（反映故障）。

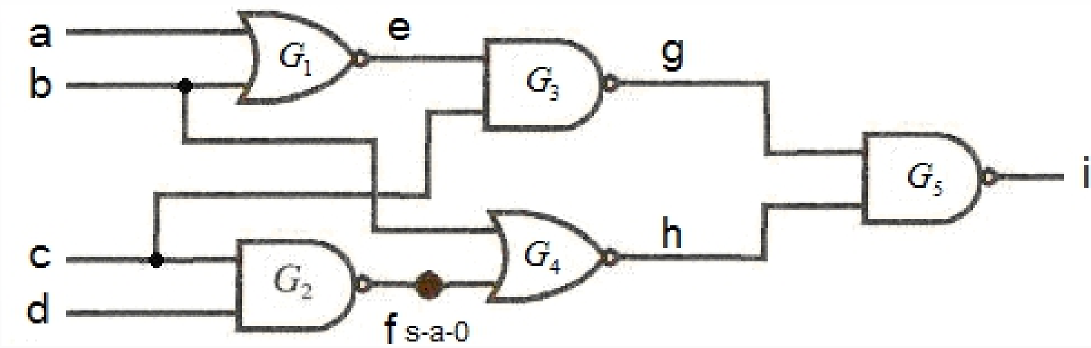
（通常存在选择性：三输入与非的Zs-a-0有三种选择，初始选择是任意的，但在算法实现中可能需要放弃原选择而返回进行其它选择，这称为后退，后退可能需要重复进行，直到所有可能选择都被考虑过为止。）

① 敏化从故障源到输出所选路径（故障传播），通过故障原始-立方或已产生的测试码和传播-立方的交来传播故障效应，建立故障传播路径并任选一条，这一过程称为-驱动，-驱动一直进行到某个原始输出端为或。

② 一致性操作（线合理性）即导出相应的初级输入值。

在-驱动过程中，经-交运算形成的元件信号值称为测试立方tc（未确定的可认为是x）。

D算法的求解示例



- 根据故障位置，首先列出G1、G3为原始立方，G2为故障立方，G4、G5为传播立方：

G1			G2			G3			G4			G5		
a	b	e	c	d	f	c	e	g	b	f	h	g	h	i
X	1	0	X	0	D	0	X	1	0	D	$\overline{D}$	1	D	$\overline{D}$
1	X	0	0	X	D	X	0	1	D	0	$\overline{D}$	D	1	$\overline{D}$
0	0	1	1	1	$\overline{D}$	1	1	0	D	D	D	D	D	$\overline{D}$

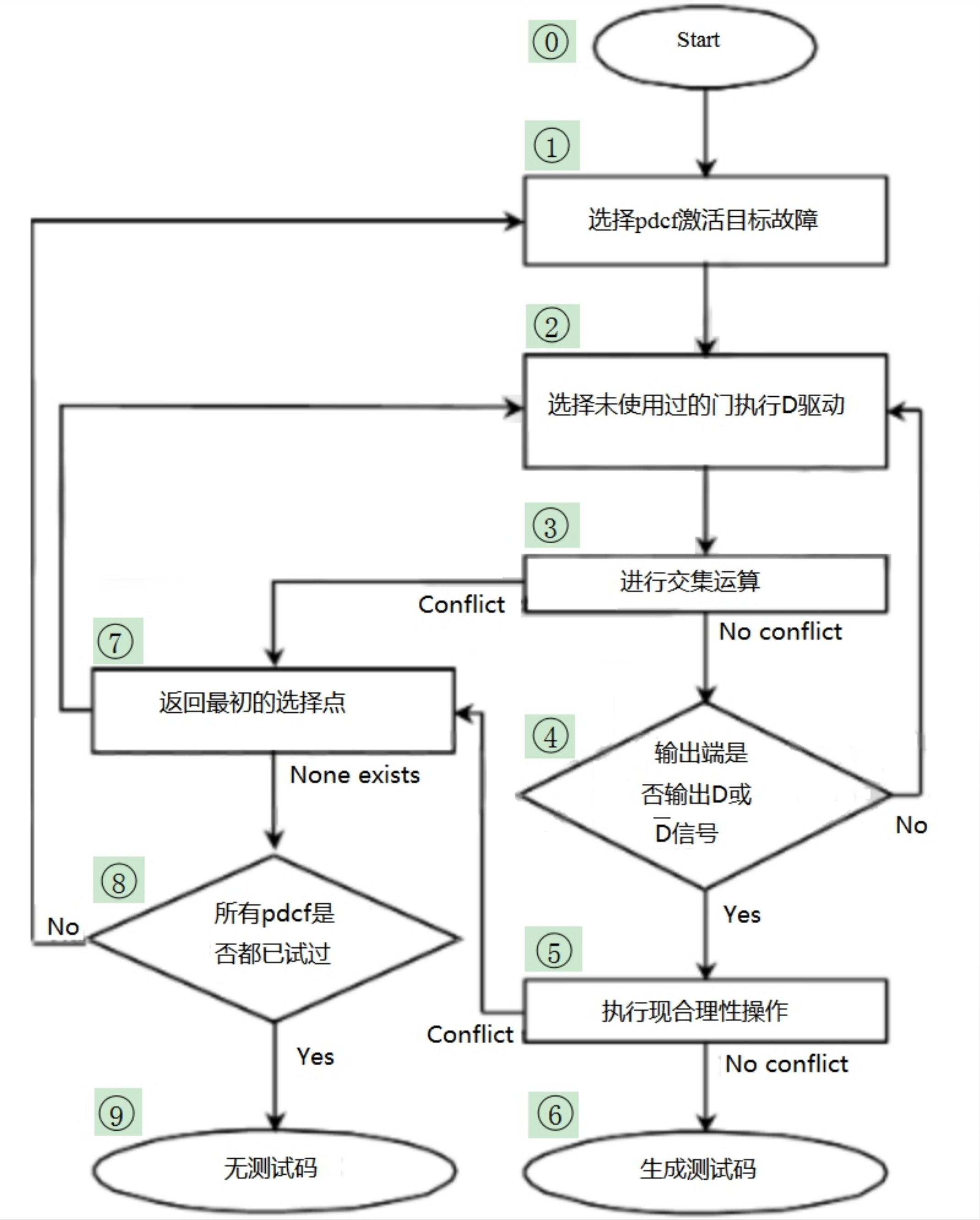
- D算法的操作：

	a	b	c	d	e	f	g	h	i
D驱动操作：									
1) 选择 fs-a-0故障原始D立方t_c^0			1	0		D			
2) 将t_c^0与G4的传播D立方来求交集得t_c^1		0	1	0		D		$\overline{D}$	
3) 将t_c^1与G5的传播D立方来求交集得t_c^2		0	1	0		D	1	$\overline{D}$	D
一致性操作：									
1) 由t_c^2与G3的原始立方来求交集得t_c^3		0	1	0	0	D	1	$\overline{D}$	D
2) 由t_c^3与G1的传播D立方来求交集得t_c^4	1	0	1	0	0	D	1	$\overline{D}$	D

结束fs-a-0故障的测试码为1010D

- 强调：①所有门都运行结束；②每次算法结果为一组输入/输出。
- 以上考虑的是组合电路的例子，D_算法也可以用于时序电路的测试产生，一般都较为复杂

D算法流程图



原始故障D立方Primitive d-cube for failure (pdcf)

本章小结

1． 概述

• 1．用途：①验证新设计；②分析故障

• 2．冒险概念（静态冒险、动态概念）、竞争概念（临界竞争、非临界竞争）

• 二、逻辑线路描述语言

1. 寄存器传输语言RTL

2. 门级线路描述语言

• 三、逻辑模拟

1. 器件延迟描述：△T；△R，△F；△M-△m；△I

2. 三值模拟（0，1，u）

3. 基本模拟程序结构

• ①编译法；②表格驱动面向事件单位延迟模拟

• 四、测试码生成

• 1.故障模型：⑴固定性故障；⑵桥接故障

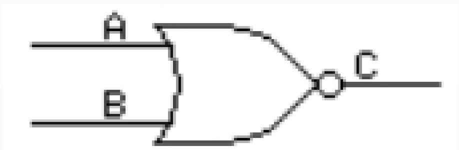
• 2.路径敏化法和D算法

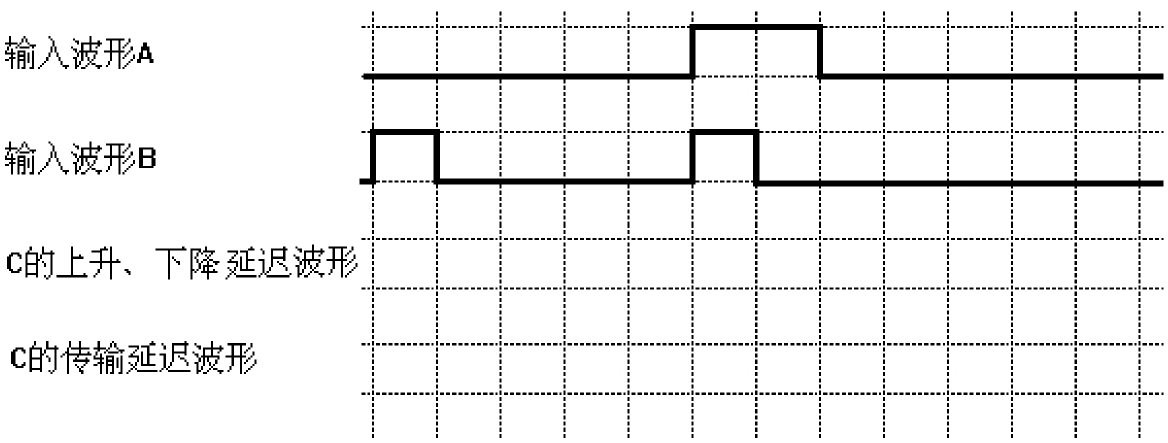
• ⑴路径敏化：单路径敏化和多路径敏化

• ⑵D_算法：几个基本概念，D_算法的实现。

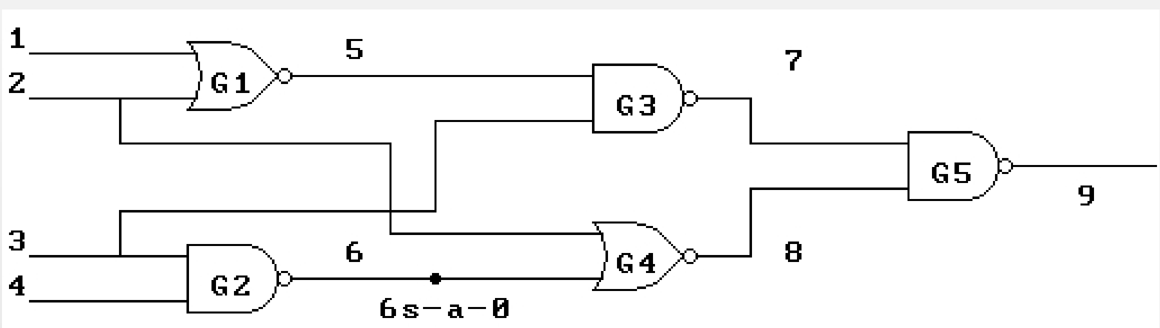
思考题

1．对图示电路进行延迟分析，并按要求画出波形图。分析延迟：$\Delta I = 1;\ \ \Delta R = 1;\ \ \Delta F = 3.$





2．对图示电路按要求进行故障分析。



(1)用路径敏化法求解6s-a-0的测试输入，并讨论输出；

(2)写出D—算法中，各个门在分析6s-a-0故障时的立方情况。

3．写出逻辑函数 $F = \overline{x_1 + x_2 + \overline{x_1 + x_2}}$ 的三值真值表。