

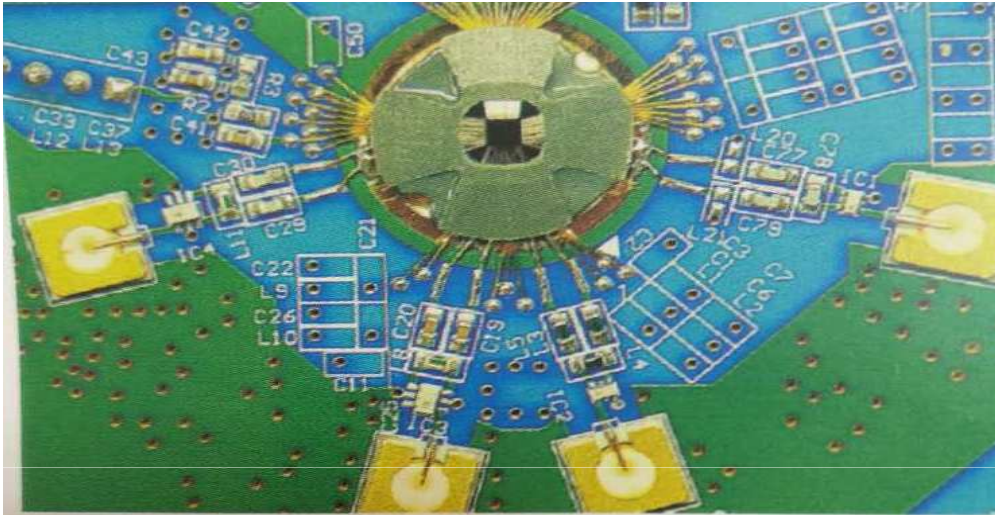
第七章 测试技术 和可测试性设计

王晓华

wxhsnow@163.com

简介

- 任何集成电路不论在设计过程经过了怎样的仿真与检查，在制作完成后都必须通过测试来最后验证设计和制作的正确性



G-probe (high frequency RF)

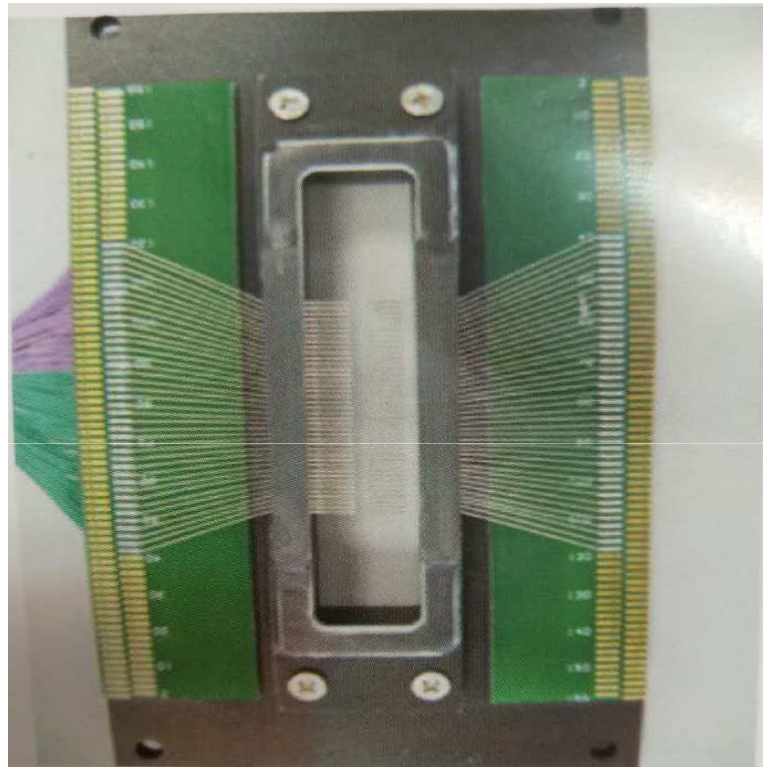
Probe material: BeCu

Max frequency: 3.5 GHz

Min pitch: 50

Min Pad size: 40x40 um

Temperature: 0-125C

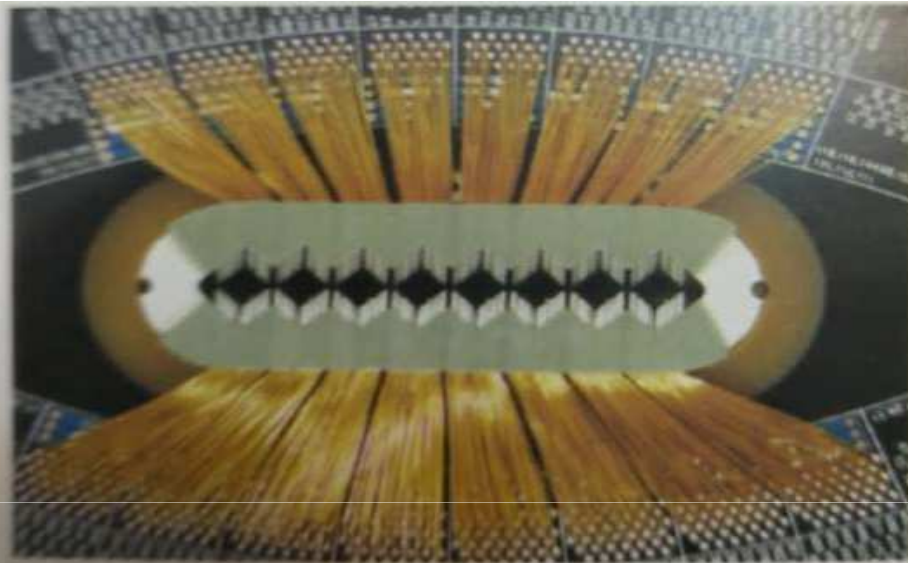


Low end probe card for resistor test

Probe material: W, ReW

Wire Diameter: 15 mils

wxhsnow@163.com



Multi-dut diagonal

Probe material: W, ReW, BeCu

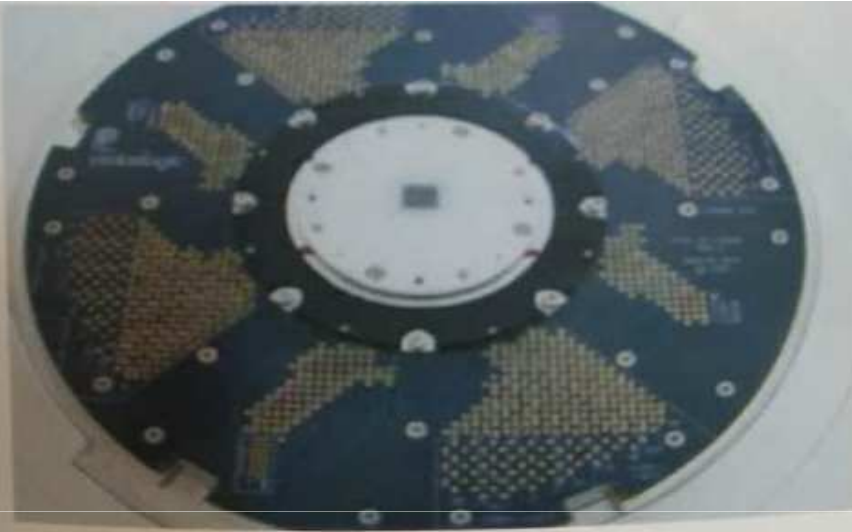
Max Duts: 8

Min pitch: 60um

Min Pad size: 30x40 um

Temperature: -40-160C

wxhsnow@163.com



Vert!™ Vertical Probe Card

Probe material: Paliney, BeCu

Min bump array: 150u

Max pin count: 5,000

Dut Array: 1X2, 1X4

Pad material: Standard, Eutectic, and Lead Free solder bump

Probe style: Flat, Wedgee

wxhsnow@163.com

目录

- ◆ 7.1 VLSI可测试性的重要性
- ◆ 7.2 测试基础
- ◆ 7.3 可测试性设计

7.1 VLSI可测试性的重要性

□测试的目的:

- 直观的检查设计的具体电路是否能像设计者要求的那样**正确的工作**
 - 通过测试确定电路**失效的原因**以及**失效**发生的具体**部位**，以便改进设计和修正错误
- ### □为实现对芯片中的错误和缺陷定位，从测试技术的角度而言
- 要解决测试的**可控制性**和**可观察性**
 - 希望内部节点是“**透明的**” ——可测试性设计

wxhsnow@163.com

可测试性的三个方面

- 可测试性的三个方面包括：测试生成、测试验证、测试设计
- **测试生成**：产生验证电路的一组测试码——**测试矢量**
- **测试验证**：指一个给定测试集合的有效性测度
通常通过故障模拟来估算
- **测试设计**：通过在逻辑和电路设计阶段考虑测试效率问题，加入适当的附加逻辑或电路以提高将来芯片的测试效率
目的：提高前两种工作的效率

测试的基本形式

□ 集成电路的测试分为两种基本形式

➤ **完全测试**：对芯片进行**全部**状态和功能的测试
要考虑集成电路**所有的可能**状态和功能

—— **完备测试**

➤ **功能测试**：只对集成电路设计之初**所要求的**运算功能或逻辑功能是否正确进行测试

—— **局部测试**

测试的基本形式的选择

- 在集成电路研制阶段，为分析电路可能存在的缺陷和隐含的问题，应对样品进行完全测试
- 在集成电路产品的生产阶段，则通常采用功能测试以提高测试效率，降低测试成本

完全测试

□ N个输入端

❖ 纯组合逻辑电路： 静态： 2^N 个状态

动态： 考虑状态转换时的延迟配合问题

❖ 时序逻辑电路： 考虑记忆单元

➤ 与输入有关

➤ 与上一时刻的信号有关

最坏情况： 2^N 个输入的全排列

($N=3$ ——40320; $N=4$ ——20922789888000)

wxhsnow@163.com

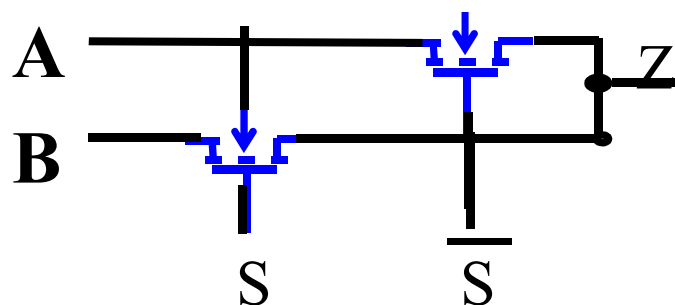
二选一MUX的test bench举例

□ 输入：A、B

□ 控制：S

□ 输出：Z

控制S	输出Z
0	A
1	B



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

若输入端状态不定？ X高阻？ Z

可测试设计的优化

□ 为解决测试问题而设计多种测试方案和测试结构，
在逻辑设计之初就考虑测试的问题，经可测试设计
作为逻辑设计的一部分加以设计和优化

□ 基本原理：

- 转变测试思想，将输入信号的枚举与排列的测试方法
转变为对电路内各个节点的测试，
即直接对电路硬件组成单元进行测试：
- 降低测试的复杂性，即将复杂的逻辑分块，使模块易于测试
- 采用附加逻辑和电路使测试生产容易，改进其可控制性和可
观察性，覆盖全部的硬件节点
- 添加自检测模块，使测试具有智能化和自动化

[返回](#)

7.2 测试基础

□ 7.2.1 内部节点测试方法的测试思想

□ 7.2.2 故障模型

固定故障

桥接故障

□ 7.2.3 可测试性分析

□ 7.2.4 测试矢量的生成

7.2.1 内部节点测试方法的测试思想

- ❑ 直接对电路内部的各节点测试可降低测试的工作量，提高工作效率
- ❑ 电路制作完成后内部各节点将**不可直接测试**
——对系统输入一定的测试矢量，在系统的输出端观察所测试节点的状态
- ❑ **测试矢量**——控制被测试节点的状态，并将该节点的状态效应传送给输出观察点

□对节点测试的思想：

- 假设在待测试节点存在一个故障状态
- 将该节点状态反映和传送到输出观察点
- 测试中发现该输出观察点出现故障效应
==》该节点确实存在假设的故障

节点测试

- 为减少测试的工作量，测试生成通常是针对门级器件的外节点
 - 直接针对晶体管级生成测试具有更高的定位精度，但难度与工作量将大大增加
 - 电路内的节点并不是全部可测的
- 要求采用新的技术和算法生成测试
 - 可采用具有可测试性的电路结构以及其他辅助结构，提高测试的覆盖率和测试效率

电路失效原因

- 造成电路失效的原因很多
 - 微观 的缺陷：半导体材料中存在的缺陷
 - 工艺加工中引入的器件不可靠或者错误：
带电离子玷污、接触区接触不良、金属线不良连接或开路
 - 设计不当引入的不稳定
- 测试技术只能对由失效原因所导致的客观结果——电路中信号故障进行测试：
 - 节点电平 不正确
 - 短路引起引线间不正确的连接
 - 引线开路引起信号传输失效

有效测试过程

- ❑ 将电路失效抽象成一个故障模型
- ❑ 测试矢量针对故障模型产生一组测试信号
- ❑ 测试的过程实际是一个对比结果的过程：
通过在芯片的输入端施加测试输入，检出输出信号并与预先生成的输出进行对比，判断电路的正确性
- ❑ 根据输入输出信号以及测试生产的信息得出失效的位置以及状态，再通过其他的技术手段分析具体的失效原因

测试技术的关键

□ 测试技术中要解决的问题主要有：

- 故障模型的提取
- 测试矢量的生成技术
- 电路的可测试结构设计方法
- 其他辅助测试技术

7.2.2 故障模型

- ❑ 逻辑电路的实际逻辑值与预期逻辑值不相吻合
 - 逻辑电路出现故障
- ❑ 若逻辑设计正确
 - 电路中的信号没有按照设计的要求动作
- 电路中的某一点或某一部分出现不符合设计的状态
- 出现不应该有的连接（短路、开路）

7.2.2 故障模型

□ 节点状态的错误所导致的故障可分为两大类：

➤ 永久型：主要是固定故障

电路中某一节点的逻辑值不符合设计要求或电路连接不
正确

不随时间的变化而变化，一直保持在某种状态固定不变

➤ 间歇型：随机出现的故障

电路节点有时正常有时不正常

* 电路连接错误：信号线对电源或地短路；输入与输出
短路构成信号反馈；开路

wxhsnow@163.com

固定故障

□ 按照节点或信号被固定的状态可分为两大类:

➤ 固定于1故障: 符号s-a-1 (stuck-at-1) --*stick*

➤ 固定于0故障: 符号s-a-0 (stuck-at-0)

* 导致固定型故障的原因很多:

信号短路;

器件错误状态 (晶体管一直导通或者截止)

信号线开路

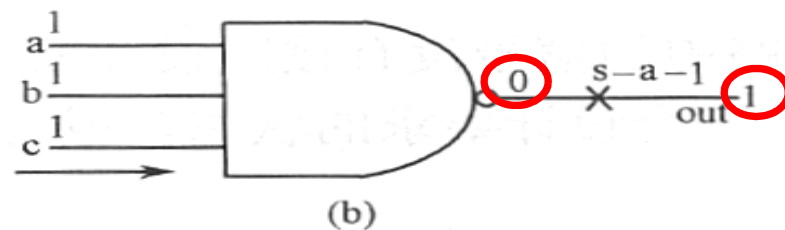
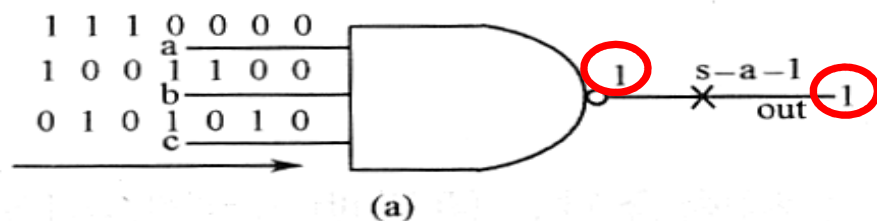
* 从测试和测试生成技术的角度看, 这些故障都可通过假设故障和测试故障加以检测

wxhsnow@163.com

固定故障举例——输出故障

□ 设三输入与非门的三个输入分别为：a，b，c
输出为out

➤ 输出端存在s-a-1故障



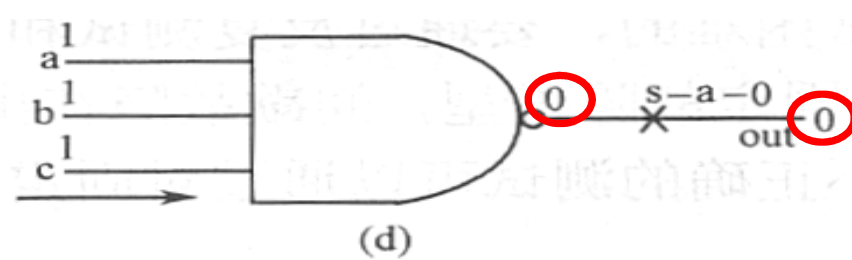
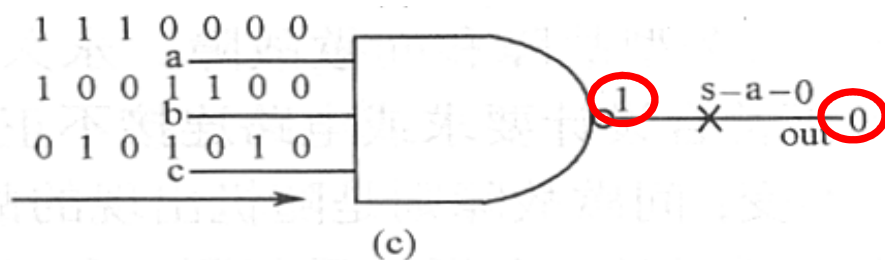
□ * 当输入a，b，c中任意一个为0时不会发现故障；

* 只有当输入为111时，正确的输出应为out=0，而故障输出out=1；

固定故障举例——输出故障

► 输出端存在s-a-0故障

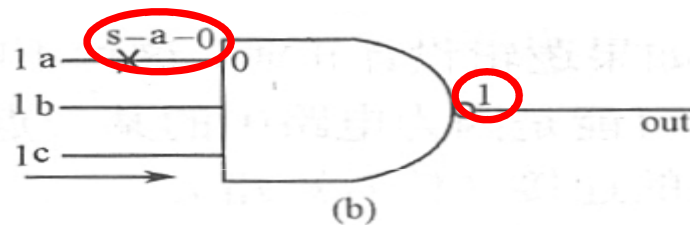
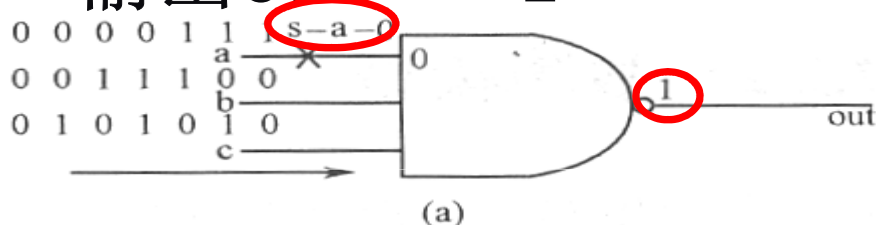
- * 当输入为111时不会发现故障;
- * 当输入a, b, c中至少有一个为0时,
正确的输出out=1, 而故障out=0;



固定故障举例——输入故障

► 输入端a存在s-a-0故障:

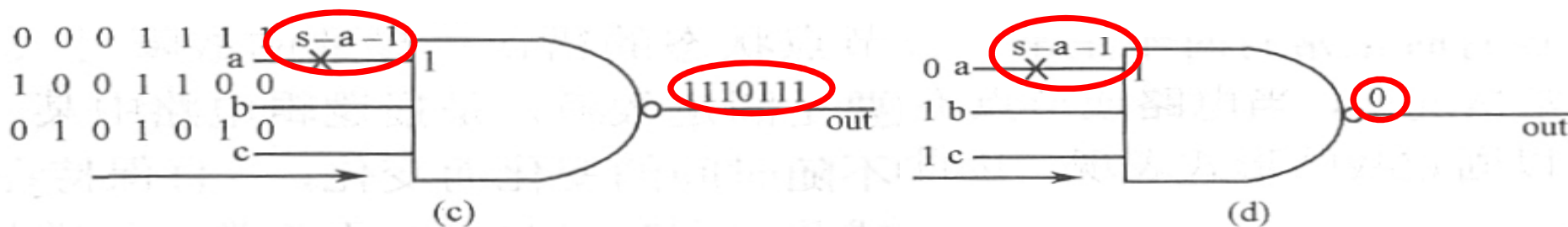
输出out == 1



- * 当输入a, b, c中至少有一个为0时,
正确的输出out=1, 与故障out=1相同,
不能反映故障;
- * 当输入为111时才发现故障;

固定故障举例——输入故障

- 输入端 **a** 存在 **s-a-1** 故障:
- * 当输入 **abc = 011** 时, 正确的输出 **out = 1**, 故障 **out = 0**, 才发现故障;
 - * 其他七种输入组合都不能反映故障;



固定故障总结

□ 对于输出端的固定型故障：

- 当正常输出值与故障值相同时，不能反映故障的存在，或者说故障状态被正常状态所掩盖；
- 只有正常输出值与故障值相反时，故障才有可能被暴露

□ 对于输入端存在的故障

- 当正常输入信号（对应故障线）与故障状态相同时，故障状态不能被反映；
- 只有正常输出值与故障值相反时，可以区分正常信号与故障信号，但要求其他信号端对逻辑门不产生逻辑控制

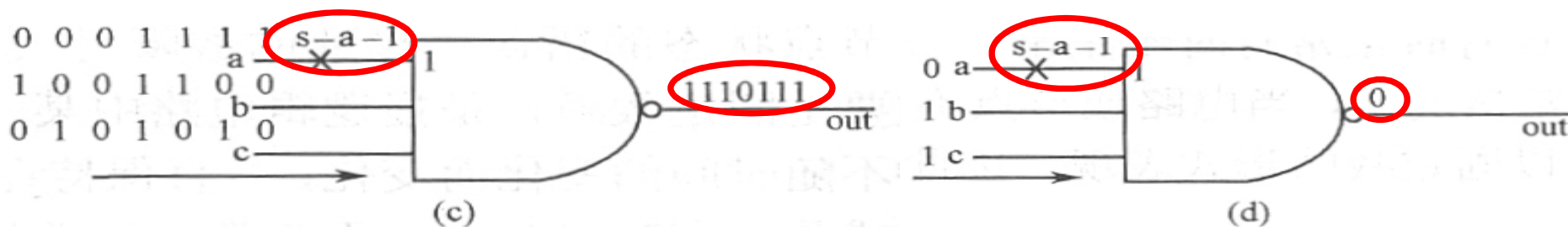
*与非门：其他为1；或非门：其他为0

固定故障举例——输入故障

► 输入端 **a** 存在 **s-a-1** 故障:

* 当输入 **abc = 011** 时, 正确的输出 **out = 1**,
故障 **out = 0**, 才发现故障;

* 其他七种输入组合都不能反映故障;

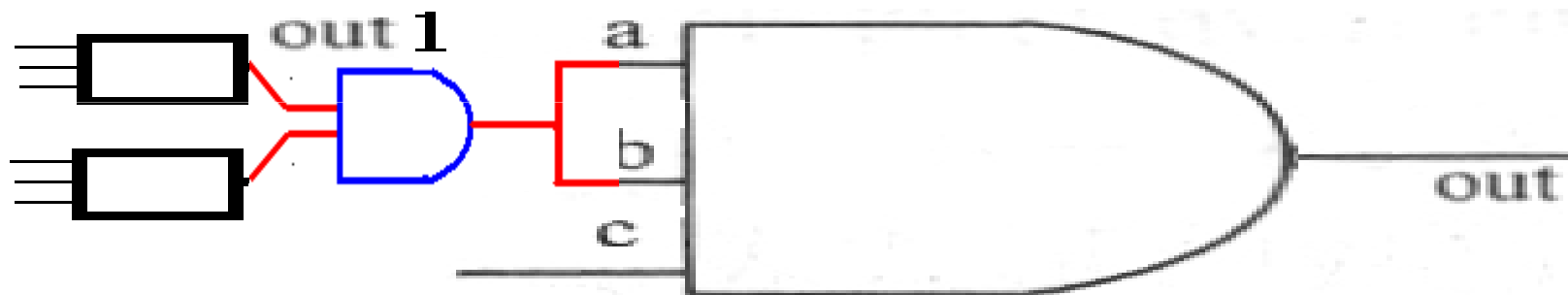


桥接故障

- ❑ 桥接故障：由于发生了不应有的信号连接而导致的逻辑错误。
- 对于电源和地线的连接错误将导致固定型故障
- 桥接故障不包括对电源和地短接的连接性错误
- ❑ 桥接故障比较复杂，可能由于两条金属连条而产生，也可能似乎由于二氧化硅的针孔造成，或者电路器件失效造成的
- ❑ 简介：
输出桥接；输入、输出桥接；

桥接故障——输出桥接

- 假设输出桥接发生了“线与”逻辑
- 即前面输出out1和out2连接后为“与函数”关系
- 若这个桥接对应了一个三输入与门输入桥接
- 假设三输入a、b、c中的a、b分别对应out1和out2

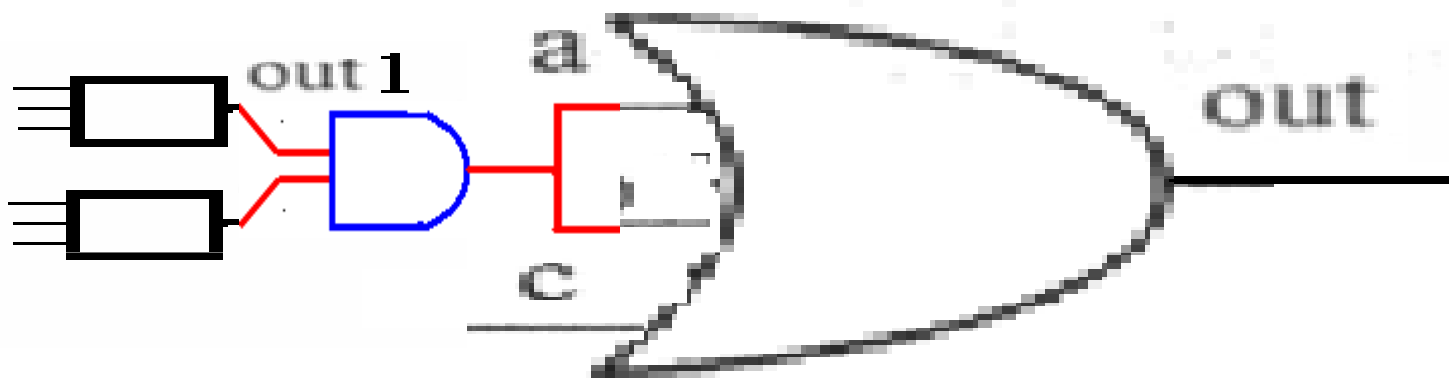


- 则：
$$\begin{aligned} a.b.c &= (out1.out2). (out1.out2).c \\ &= out1.out2.c = a.b.c \end{aligned}$$

*“线与”桥接对与门正常逻辑没有影响

桥接故障——输出桥接

□ 假设桥接发生在一个或门上



$$\begin{aligned} \text{a} + \text{b} + \text{c} &= (\text{out1.out2}) + (\text{out1.out2}) + \text{c} \\ &= \text{out1.out2} + \text{c} = \text{a.b} + \text{c} \end{aligned}$$

* “线与” 桥接改变将或门逻辑

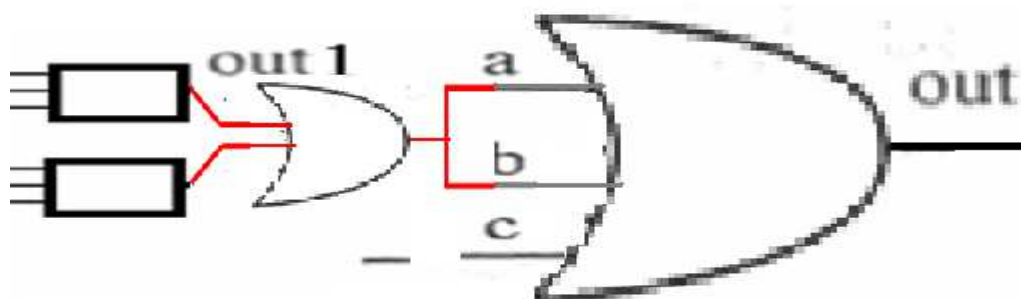
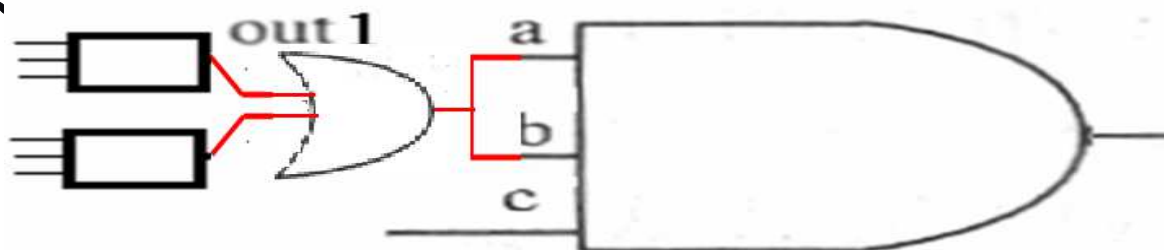
桥接故障——输出桥接

□ 假设输出桥接发生了“线或”逻辑呢？

➤ 桥接对应
一个三输入

与门输入桥接

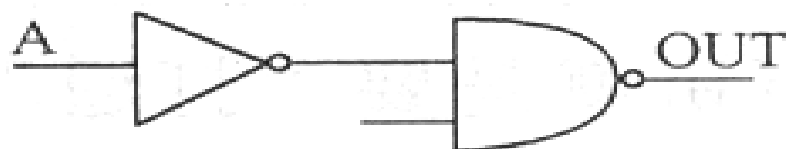
➤ 桥接发生
在一个或门上



* “线或”桥接改变与门逻辑，对或门无影响

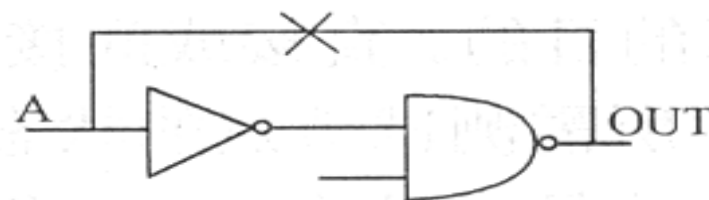
输入、输出桥接

□ 组合逻辑电路的相关输入、输出桥接



(a)

$$out = \overline{\overline{A} \cdot B} = A + \overline{B}$$



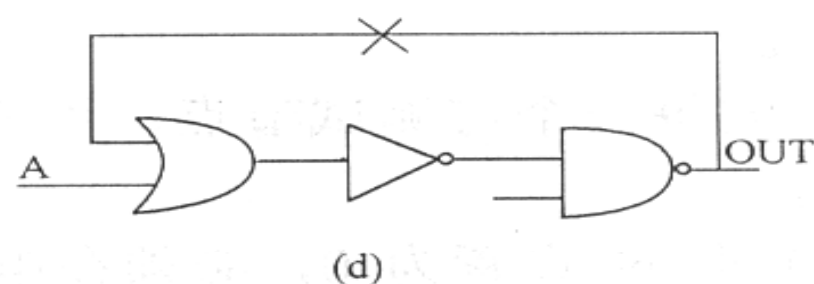
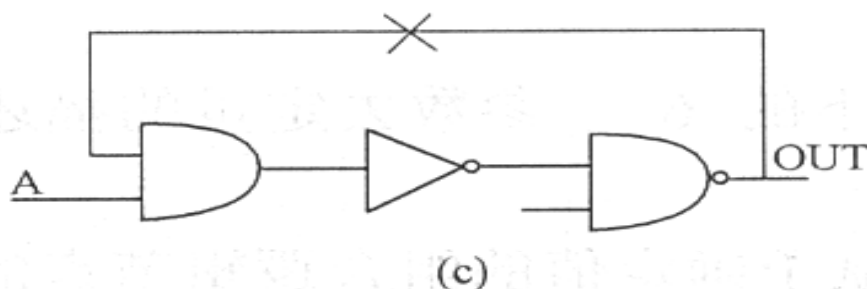
(b)

输入输出短接：

$$\begin{aligned} out &= \overline{\overline{out} \cdot B} \\ &= out + \overline{B} \end{aligned}$$

输入、输出桥接

□ 组合逻辑电路的相关输入、输出桥接



输入输出发生“线与”桥接

输入输出发生“线或”桥接

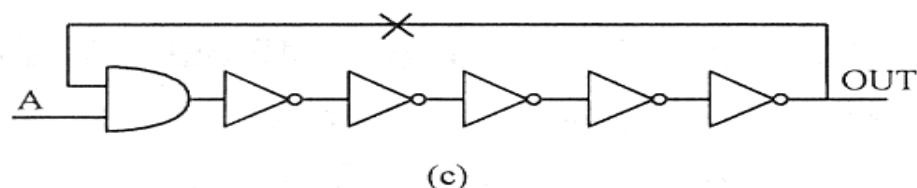
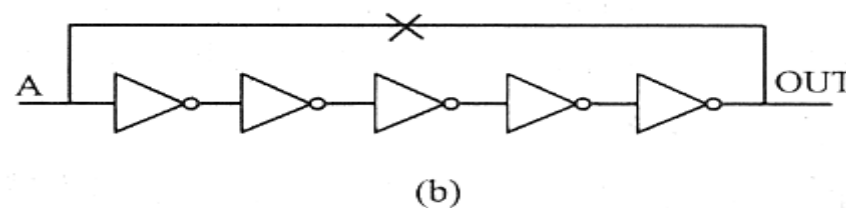
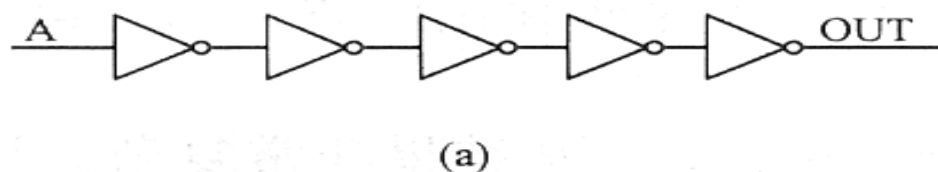
$$\begin{aligned} out &= \overline{A \cdot out \cdot B} \\ &= A \cdot out + \overline{B} \end{aligned}$$

$$\begin{aligned} out &= \overline{A + out \cdot B} \\ &= A + out + \overline{B} \end{aligned}$$

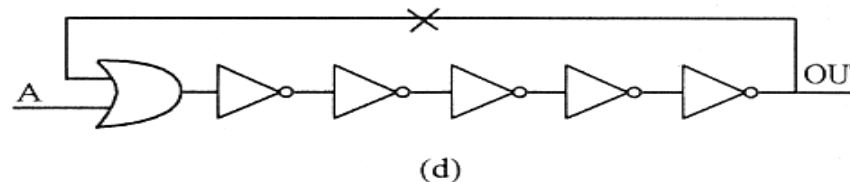
组合逻辑改变为具有记忆功能的逻辑单元

倒相延迟逻辑

□ 桥接使倒相延迟逻辑变为可控振荡器
A 变为控制端



$A = 1$ 时,
OUT 输出振荡信号



$A = 0$ 时,
OUT 输出振荡信号

7.2.3 可测试性分析

- 表征电路可测试性的关键：
电路内部节点的可控制性和可观察性
- 可控制性：
对电路内部每个节点的置0和置1的能力
- 可观察性：能否直接或间接地观察电路内部
任何节点状态的能力

可测试性分析

- ❑ 节点所存在的故障只有在某个或某些信号的控制下才能反映出来
- ❑ 直接或间接观察节点就是要在某个观察位置“**透过**”堆积的硬件“**看到**”所需测试节点的状态——“**打通**”一条或者几条从待测试节点到观察位置的路径
- ❑ 靠近输入端的内部节点，其可控制性较好，可观察性较差；
- ❑ 靠近观察位置的内部节点，其可观察性较好，可控制性较差；

可测试节点的6个参数

- $CC^0(N)$: 为了将节点N设置为0, 必须在电路中赋予确定值的组合逻辑节点的最少个数
组合逻辑节点: 电路的原始输入或标准组合单元输出
- $CC^1(N)$: 为了将节点N设置为1, 必须在电路中赋予确定值的组合逻辑节点的最少个数
- $SC^0(N)$: 为了将节点N设置为0, 必须在电路中赋予确定值的时序逻辑节点的最少个数
时序逻辑节点: 标准时序单元输出
- $SC^1(N)$: 为了将节点N设置为1, 必须在电路中赋予确定值的时序逻辑节点的最少个数
- 这四个参数描述了节点N的可控制性
参数值越大, 可控制性越差

可控制性

可测试节点的6个参数

- **CO(N)**: 为了将节点N的值传输到某个原始输出所需经过的组合逻辑单元的个数，以及为了把节点N的逻辑值传播到原始输出，必在电路中赋予确定值的组合逻辑节点的最少个数
- **SO(N)**: 为了将节点N的值传输到某个原始输出所需经过的标准时序单元的个数，以及为了把节点N的逻辑值传播到原始输出，必在电路中控制的标准时序单元的最少个数
- 这两个参数描述了节点N的可观察性
参数值越大，可观察性越差
- * 并不是所有的节点都具有可观察性和可控制性
——节点不可测

可测试性分析

- 可测试性分析必须借助某些计算机软件
- 借助测试矢量生成软件对电路产生测试矢量，确定不可测试的节点的数量与位置，修改逻辑，使测试覆盖率达到100%
 - 给出各可测试节点的测试矢量
- 借助可测试性分析软件确定节点的可测试覆盖率，然后修改逻辑，使测试覆盖率达到100%
 - 给出各可测试节点的可控制性和可观察性的强弱
- * 修改逻辑或者插入测试观察点

易测试逻辑的特点

□ 一个好的易测试逻辑应该具备的特点：

- 容易产生测试矢量
- 尽量小的测试矢量集
- 容易实现故障定位
- 附加电路尽可能少
- 附加电路引出线尽可能少

7.2.4 测试矢量生成

□ **测试矢量：** 一组测试码

包含了测试输入和应有的测试输出

➤ 测试输入： 加到电路原始输入端的激励信号

➤ 测试输出： 用于比对实测结果的输出信息

□ 根据待测节点的置位要求，以及将加上的故障传播到输出所应给出的信号要求，产生的测试信号就是所谓的测试矢量

生成测试矢量的环节

□生成测试矢量包括 3 个环节：

➤对该节点设置正常逻辑值；

在原始输出端观察得到的逻辑值效应

➤敏化路径；

为将故障效应传播到某个原始输出，沿着故障传播路径的所有逻辑门必须被选通

与门 / 与非门： 1 ； 或门 / 或非门： 0 ；

➤对应到原始输入端的信号；

求出原始输入端的信号——>生成测试矢量

wxhsnow@163.com

例题：生成测试矢量

□ 例7-1:

假设存在 A: s - a - 1 故障，求测试矢量

➤ 设置 A 节点的正常逻辑值为 0:

即: $a+b=0$

➤ 敏化路径: 与非门 1/或门 0;

$\Rightarrow c=1, D=0$;

➤ 对应到原始输入端的信号;

$a+b=0 \Rightarrow a=0, b=0$

$D=0 \Rightarrow d \cdot \bar{c}=0 \quad c=1$, 故 d 可为任意值 (0 或 1)

□ 结论: 存在 A: s - a - 1 故障的测试输入为

$abcd=0010$ 或 0011

输出 f 的正常信号值为 1, 故障值为 0

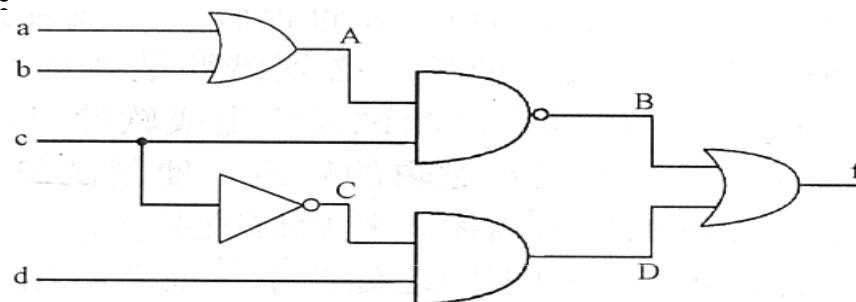


图 7-5 测试生成例子

例题：生成测试矢量

□ 例7-2： 假设存在B: s - a - 1 故障，求测试矢量

➤ 设置B节点的正常逻辑值为0；

$$B = 0 \Rightarrow A \cdot c = 0$$

$$\therefore A = 1, c = 1$$

$$\therefore a + b = 1,$$

即 a b 可以是 0 1, 1 0 或 1 1

➤ 敏化路径：或门：0；

==> D=0；

$$D = 0 \Rightarrow C \cdot d = \bar{c} \cdot d = 0 \quad \because c = 1 \therefore d \text{ 可是任意值}$$

□ 结论：存在B: s - a - 1 故障的测试输入为

abcd=0111, 1011, 1111, 0110, 1010, 1110中任意 输出
f的正常信号值为0，故障值为1

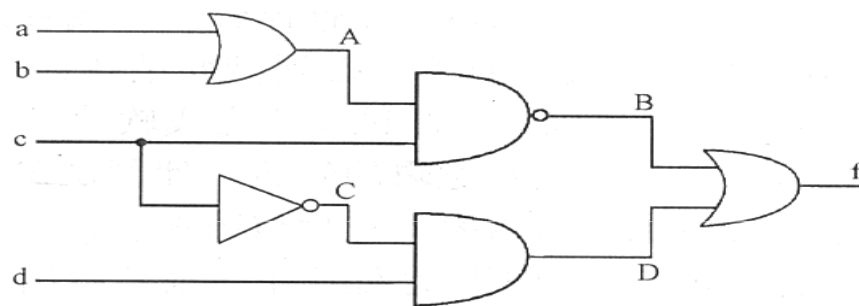


图 7-5 测试生成例子

例题：生成测试矢量

□ 例7-3： 假设存在C: s - a - 1 故障，求测试矢量

➤ 设置C节点的正常逻辑值为0；

即：c=1

➤ 敏化路径：

与门：1； \Rightarrow d=1；

或门：0； \Rightarrow B=0；

➤ 确定原始输入端的信号；

$$B = 0 \Rightarrow \overline{A \cdot c} = 0$$

$$\therefore A = 1, c = 1$$

$\therefore a + b = 1$ ，即 a b 可以是 0 1, 1 0 或 1 1

□ 结论：存在C: s - a - 1 故障的测试输入为

abcd=0111, 1011, 1111中任意

输出f的正常信号值为0，故障值为1

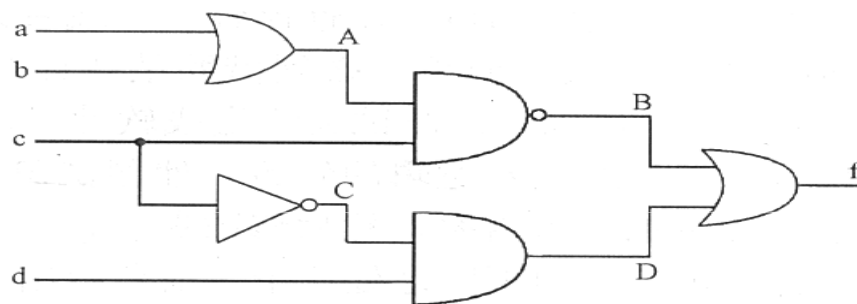


图 7-5 测试生成例子

C点存在
s - a - 0
故障？

C: s - a - 0故障

□ 假设存在C: s - a - 0故障，求测试矢量

➤ 设置C节点的正常逻辑值为1；

即: $c=0$

➤ 敏化路径:

与门: 1; $\implies d=1$;

或门: 0; $\implies B=0$;

➤ 确定原始输入端的信号;

➤ $c=0$ 与 $B=0$ 矛盾

□ C: s - a - 0故障不可测

➤ 实际C: s - a - 0故障不影响逻辑电路的正常工作

➤ 若 $c=0$ ，则应该 $C=1$ ，敏化后的电路输出 $f=1$

➤ 但由于 $c=0 \implies B=1 \implies f=1$

□ 故障冗余

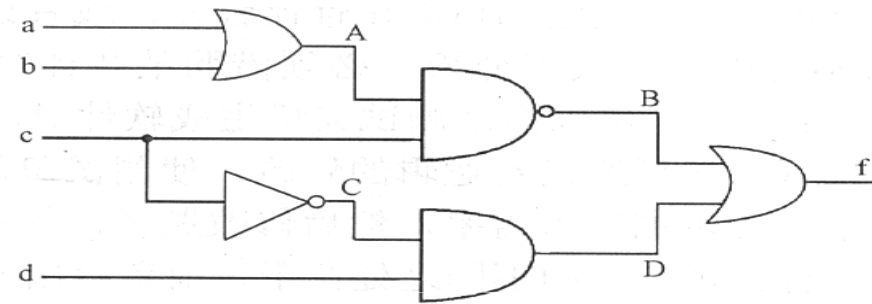


图 7-5 测试生成例子

作业

第
50
页

□ 假设存在**D**: s - a - 1 故障, 求测试矢量

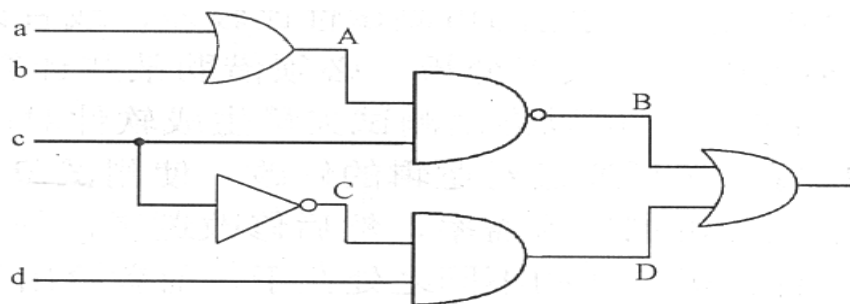


图 7-5 测试生成例子

7.3可测试性设计

- 7.3.1 分块测试
- 7.3.2 可测试性的改善
- 7.3.3 内建自测试技术

7.3.1 分块测试

- 系统规模越来越大，系统结构日趋复杂及多元化，测试难度增大
- 采用并行设计方法，由多个设计者分别完成不同部分的设计
- 分块测试 \Rightarrow 降低测试难度，同步完成测试
- 适合于易分块的结构形式：
大型时序电路、模块/总线结构

7.3.2 可测试性的改善设计

- ❑ 必须先进行逻辑系统的设计
- ❑ 对设计完成的系统做可测试性分析，确定测试困难或不可测试的节点
- ❑ 在分析的基础上决定改善可测试性的方案
- ❑ 简单方法举例：
 - 增加逻辑电路的测试点，断开长逻辑链使测试生成过程简化
 - 提高时序逻辑单元初始状态的预置能力，简化测试
 - 对不可测试点增加观察点，使其成为可测试的节点
 - 插入禁止逻辑单元，断开反馈链，将时序逻辑单元变为组合逻辑单元进行测试
 - 增加附加测试电路，改善复杂逻辑的可测试性

例：可测试性的改善设计

控制性参数 定义

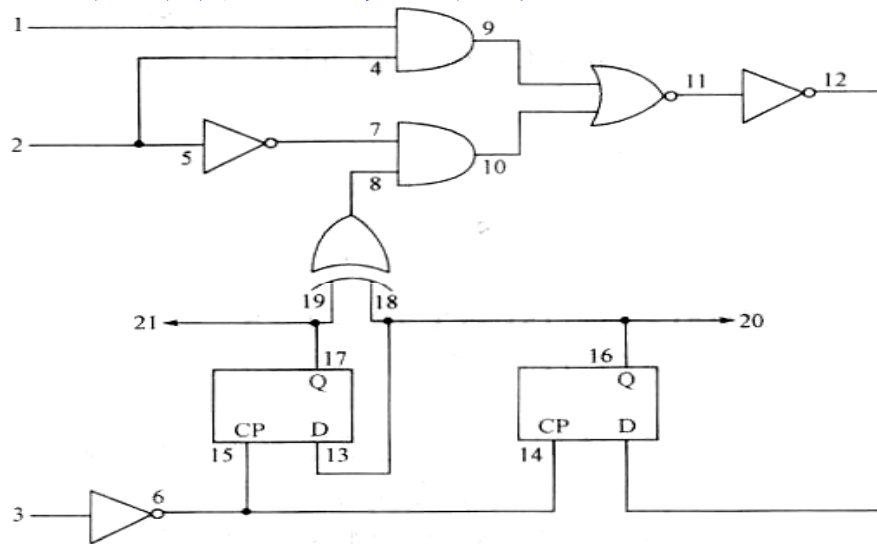


图 7-7 逻辑电路实例

各端点可控制性参数值

端点号 i	$CC^0(i)$	$CC^1(i)$	端点号 i	$CC^0(i)$	$CC^1(i)$
1	1	1	11	5	7
2	1	1	12	8	6
3	1	1	13	16	14
4	2	2	14	3	3
5	2	2	15	3	3
6	2	2	16	15	13
7	3	3	17	23	21
8	37	39	18	16	14
9	2	4	19	24	22
10	4	43			

8号的可控制性最差 ==》 10号

wxhsnow@163.com

可测试性的改善设计

□ 改善端点8的可控制性

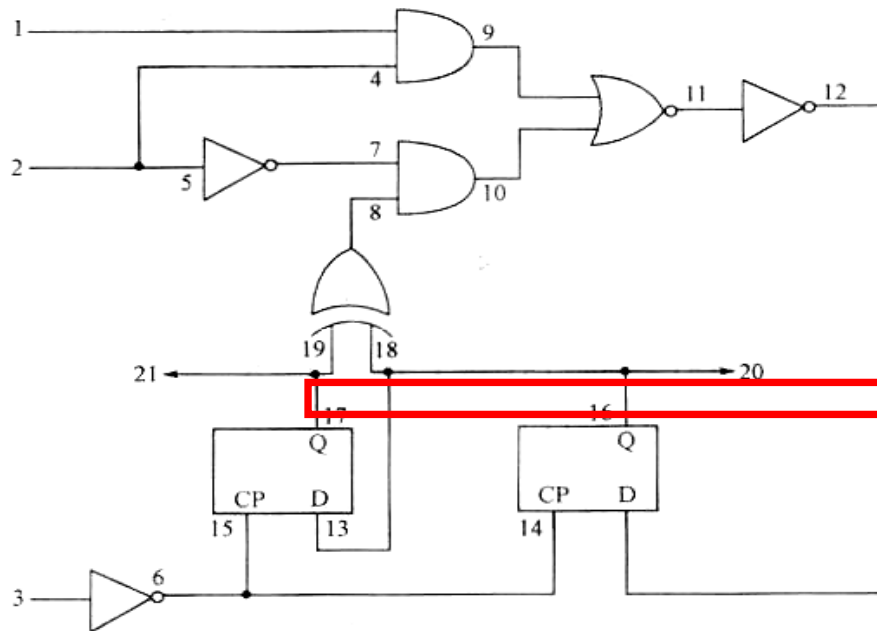
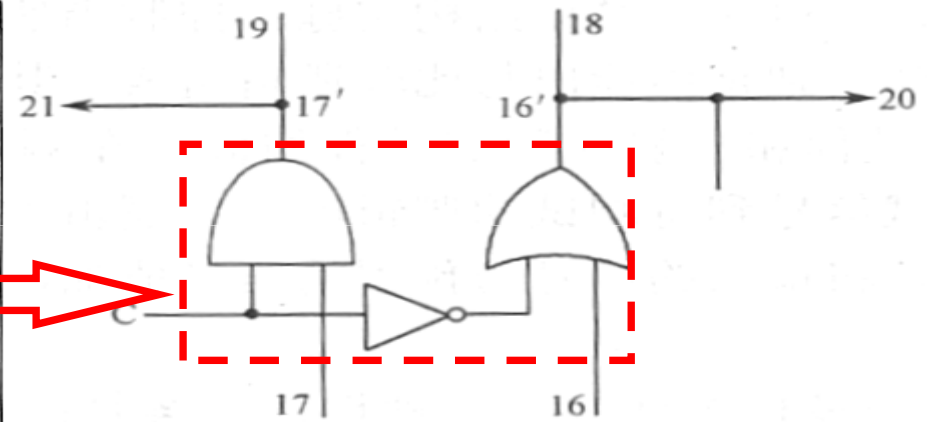


图 7-7 逻辑电路实例



C是测试控制端

当C=1时，逻辑正常工作

当C=0时，处于测试状态

可测试性的改善设计

□改善后

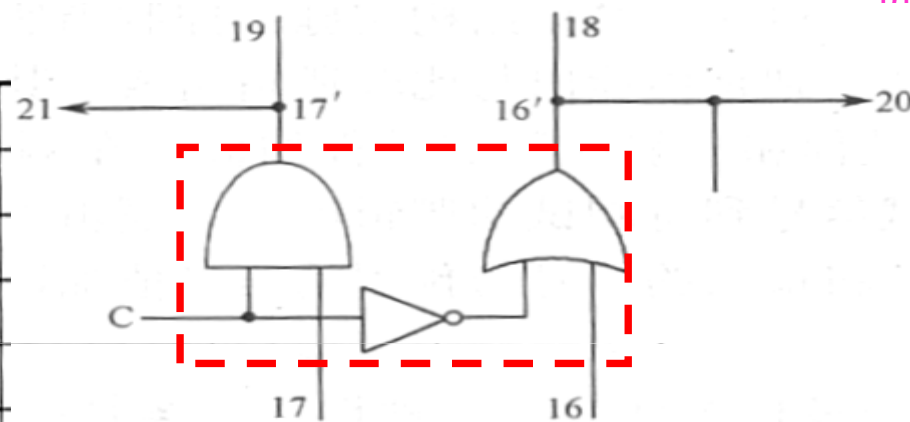
端点号 i	$CC^0(i)$	$CC^1(i)$			
1	1	1			
2	1	1			
3	1	1			
4	2	2			
5	2	2			
6	2	2	15	3	3
7	3	3	16	15	13
8	37	39	17	23	21
9	2	4	18	16	14
10	4	43	19	24	22

10

14

5

4



改善可测试性

❑ 改善可测试性带来的最大问题：增加信号线的数量

- 增加测试的控制性
- 改善测试的可观察性
- ❑ 解决矛盾的方案
- 使用编码器

增加测试时间

增加测试时间

根据测试的需要增加可控制点（ 2^N 个）

设计一个控制端——区别正常工作状态和测试状态

测试状态：N个输入经编码器得到 2^N 个输出，分别控制

- 使用串行移位寄存器

测试控制点的控制值串行移入寄存器，由控制器对控制线施加控制信号，观测点上的信号值再由寄存器收回，然后再串行移出，可观察电路内部节点的值

7.3.3 内建自测试技术

□ 内建自测试技术:

- 在电路内部设计一些附加的自动测试电路
- 与电路系统本身集成在同一块芯片上

□ 电路有两种工作模式:

- 自测试模式
- 正常工作模式

□ 自测试电路的设计必须解决三个问题:

- 隔离——防止测试逻辑影响正常逻辑（禁止结构）
- 控制——使测试有序地工作
- 观察——由检出和比较逻辑组成，监视测试结果

内建自测试技术

- ❑ 内建自测试技术采用的结构：多种多样
 - 用**ROM**结构作为信号发生器
 - 每个字中包含测试输入和预置的正常测试输出值
 - 在外部提供的测试时钟控制下，依次送出测试矢量：
 - 测试输入激励待测试逻辑；
 - 预置输出值送到对比逻辑
 - 对比逻辑（主要有异或门组成）将检出的输出与预置的输出进行比对，并输出结果
- ❑ 采用外部时钟同步，只需很少的几根信号线就可同步地进行观察
 - 附加输入和输出信号少，结构简单

内建自测试技术

□采用内建自测试技术的优点:

➤简化外部测试设备

外部测试设备在内建自测试模式下仅仅完成初始化内建自测试逻辑和提供同步时钟，以及检查对比逻辑的输出来判断待测试逻辑是否正常

➤提高测试效率

内建测试逻辑与被测试逻辑在相同的环境下工作，可在被测试电路的正常工作速度下进行检测 ==> 提高测试速度，检查电路动态特性

xx@163.com

7.3.4 扫描测试技术

- 扫描测试技术主要有两种方式:
 - 利用简单的串行移动寄存器，将电路中的各节点与其相连，利用移动寄存器控制各节点的状态并读入各节点的响应，串行输出
 - 另外增加移位寄存器
 - 在测试状态下，将电路中的所有存储单元（不包括寄存器阵列）连接成移位寄存器，进行串入/串出的状态测试
 - 利用附加的多路转换器和原有存储单元

THANKS