

CMOS 存储器中地址译码器的开路故障及测试

刘建都, 张安堂

(空军工程大学 导弹学院, 陕西 三原 713800)

摘要: 对 CMOS 存储器中地址译码器的开路故障进行了分析和分类, 得出了其中有一类开路故障不能用常用的测试算法可靠的测试出, 给出了测试该类开路故障的测试方法以及针对该类开路故障的容错性设计方案。

关键词: CMOS 存储器; 地址译码器; 开路故障; 推进测试算法

中图分类号: TP277 **文献标识码:** A **文章编号:** 1009-3516(2000)02-0078-04

在许多方面, 存储器芯片的测试同一般芯片的测试, 存储器芯片的集成度高并且面积大, 导致其测试变得更加困难。常用于存储器的测试方法为推进式测试; 在推进测试中, 是通过对存储器的每一个单元分别进行读写操作来检测故障, 推进测试算法的复杂度与测试的存储空间成线性关系, 所以也称为线性算法, 常用的推进算法有 $3N$ 、 $6N$ 和 $9N$ (N 为存储单元数)。

存储器中地址译码器是组合逻辑电路, 它被用来根据输入地址来选中唯一的存储单元, 图 1 示出了存储器中的典型字线译码器电路。假如有故障的地址译码器不引起时序行为, 有故障的地址译码器的故障行为可分为下列两种情况:

无选中——译码器未选中任何已编址的存储单元;

多选中——译码器选中多个已编址的存储单元。

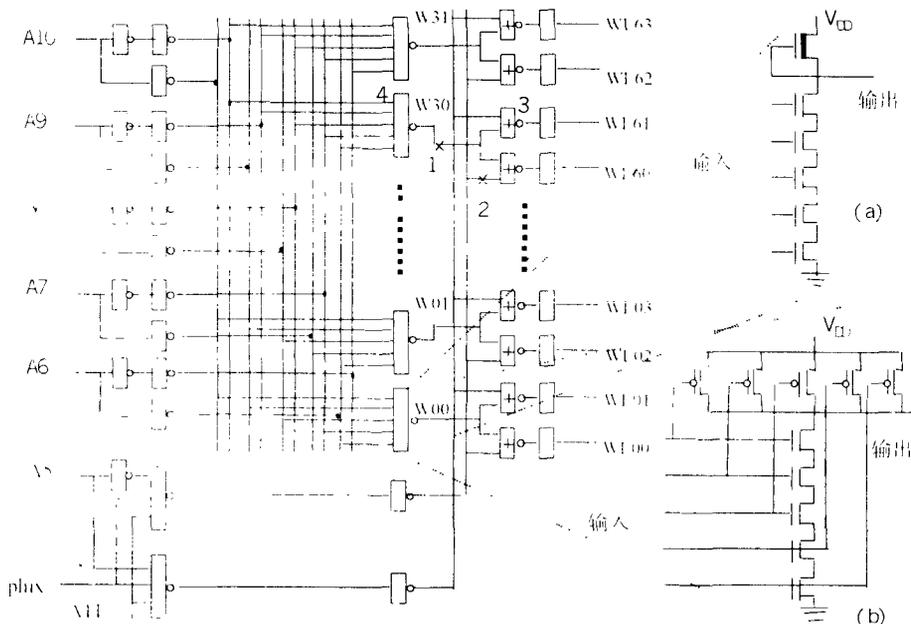


图 1 存储器中的典型字线译码器电路

收稿日期: 2000-01-12

作者简介: 刘建都(1962-), 男, 副教授.

在无选中的情况下,故障可转换为输入地址所对应单元的固定 0 或固定 1 故障,在多选中的情况下,可转化为多个不同单元有相同地址的问题,上述两种情况下均可用传统的测试算法可靠地检测出。如我们处理的是 NMOS 存贮器(译码器),上述两种情况完全适合,也就是说 NMOS 存储器中地址译码器的开路故障可用传统的测试算法可靠地测试出。但当我们处理 CMOS 存储器中地址译码器的开路故障时,上述假设不一定成立,只有一部分开路故障可转化为固定性故障,其余故障将引起时序行为,因此不能用传统的测试算法可靠地测试出。

1 CMOS 存贮器中译码器开路故障的分类及可测性分析

一般来说,地址译码器中的开路故障可发生在逻辑门的内部和外部。图 1 中所示的故障 1 和故障 2 是门间故障,这类故障由门间相互连线断开引起,因此至少一个存贮单元不能读写,呈现为固定性的故障,换句话说,门间开路故障不引起时序行为,推进式测试可以可靠地测试出该类故障。但是,门内的开路故障(故障 3 和故障 4)可能仅由一个 MOS 管的开路引起;如果一个门的开路故障使门的输出到 V_{DD} 和 V_{SS} 的通路均断开,可等效为该门的输出端到下一级门间的开路故障,也可以可靠地检测出;如果一个门内的开路故障仅使其输出到 V_{DD} 或 V_{SS} 中的串联通路断开,其上拉或下拉网络中的一个一直处于开路,因此可等效为固定性故障,也可以可靠地检测出;如果一个门内的开路故障仅使其输出到 V_{DD} 或 V_{SS} 中并联通路的一部分(如一个管子)断开,那么这个门的输出不为固定的状态,它不仅与现在的输入有关,而且与门的以前状态有关,也就是说该故障引起时序行为,不能用传统的测试算法可靠的检测出。

2 难于测试故障的测试试验

与非门(或非门)中的单个 P 管(N 管)的开路故障是难于测试的。为理解地址译码器中这类难于检测的故障,图 2 示出了存贮器的原理框图。它由存贮矩阵、字线译码器和列线译码器组成。图中分别根据 $A_{11}-A_5$ 及 A_4-A_0 选择字线和列线,图中标明了三个存贮单元 A、B、C,其字线地址分别为 $(A_{10}-A_6)$ 00100、00110、10110。可画出单元 C 所对应地址译码器中的与非门电路如图 3 所示,假定与 A_{10} 相联的 P 管开路,我们作下面的试验:

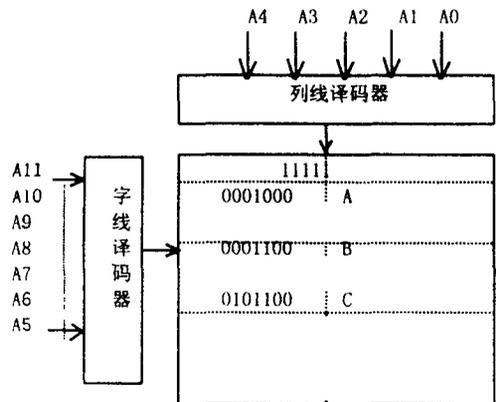


图 2 存储器的组成原理框图

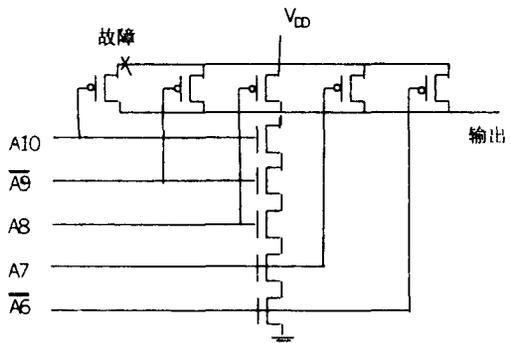


图 3 与 A_{10} 相连的 P 管开路的完成译码的故障与非门 W_{22}

- (1)①C 单元写“1”($A_{11}-A_0=010110011111$)
图 3 所示的地址译码器门输出为“0”,C 单元写“1”
- ②A 单元写“0”($A_{11}-A_0=000010011111$)
图 3 所示的地址译码器门输出为“1”,A 单元所对应地址译码器门输出为“0”,A 单元写“0”
- ③读 C 单元, ($A_{11}-A_0=010110011111$)
C 单元读出结果为“1”。
未检测出故障。

- (2)①C 单元写“1”($A_{11}-A_0=010110011111$)
图 3 所示的地址译码器门输出为“0”,C 单元写“1”。

- ②B 单元写“0”($A_{11}-A_0=000110011111$)
由于与 A_{10} 相连的 P 管开路,不能使故障门输出为“1”,图 3 所示的地址译码器门输出仍保持原值“0”,C 单元写“0”。
B 单元所对应的地址译码器门输出也为“0”,B 单元写“0”

③读C单元, ($A_{11}-A_0=010110011111$)

C单元读出结果为“0”。

检测出了故障。

下面对上面的试验现象进行分析:由于首先选中C单元,图3所示门输出为“0”,当 A_9-A_6 不变时,仅变化 A_{10} 时(如:变化为B单元的地址),由于与 A_{10} 相连的P管开路,不能使图3所示的门输出为“1”,仍保持为“0”,该门仍处于激活状态,也就是说此时同时选中了B单元和C单元,因此在给B单元写“0”的同时,也把“0”写入到C单元,覆盖了C单元原来的值,因此检测出了故障。当 A_9-A_6 变化时(变化为A单元的地址),上拉网络中的并行路径至少有一个导通,虽然与 A_{10} 相连的P管开路,仍能使图3所示的门输出为“1”(不激活状态),在给A单元写内容时仅写入了A单元,读出C单元的值仍为原来的“1”,所以未检测出故障。由上面的分析可以看出,该类开路故障能否检测出与测试序列有关。

3 推进式测试的测试分析

表1说明了推进式测试的6N算法的测试过程:第一步,初始化存贮器所有存贮单元为“0”;第二步,按地址递增的顺序对每一个单元进行先读初始化“0”并写“1”值的操作;第三步,按地址递减的顺序对每一个单元进行先读“1”,后写“0”,再读“0”的操作。

由6N算法的测试过程可以看出,由于C单元的地址为: $A_{10}-A_6=10110$,在6N测试的第二步,下一个地址为:10111,有故障门不能激活,因此不能检测出故障;类似地,在6N测试的第三步,下一个地址为:10101,有故障门不能激活,因此不能检测出故障。推进式(线性)算法只有在下一地址激活该故障时才能检测出该类故障,它必须保持激活直到对该单元的一个读操作来检测该类故障,并且要由初时值及覆盖的数值来决定是否可测试出该类故障。但是,完成地址译码的与非门中的大多数故障并不满足该条件,因此也就不能可靠地检测出该类故障。当然,推进式测试可以选用任何的测地址序列,只要他访问了所有的存贮单元即可,为了简单起见,通常选用递增或递减的地址序列;一般来说,不存在能够检测所有该类故障的地址序列。

表1 推进式测试的6N算法的测试过程

地址	步 骤		
	第一步	第二步	第三步
0	写“0”	读“0”、写“1”	读“1”、写“0”、读“0”
1	写“0”	读“0”、写“1”	读“1”、写“0”、读“0”
2	写“0”	读“0”、写“1”	读“1”、写“0”、读“0”
$N-1$	写“0”	读“0”、写“1”	读“1”、写“0”、读“0”

4 提议的测试算法

完成地址译码的与非门中的P网络的单一开路故障是难以测试的,我们可以附加一个小的循环来检测这类难以测试的故障。对于地址译码器中的每一个与非门,它都有相应的字线地址,比如说发生故障4的与非门 W_0 的相应地址为 $A_{10}-A_6=00000$,假如说它选中相应的存贮单元D,下一次我们仅变化地址的一位(如 A_6),即 $A_{10}-A_6=00001$,假如说它选中相应的存贮单元E,如果门 W_0 中与 A_6 相连的P管开路,在选中单元E的同时也选中了单元D,因此对单元E的写入操作同时也完成了对单元D的写入,下一时刻读D单元的内容就可检测出门 W_0 中与 A_6 相连P管的故障,依此类推,可以完成地址译码器中的完成地址译码的门 W_0 中其它P管开路故障的检测:

- | | | | |
|-------------------------|------|-------------------------|------|
| (1) $A_{10}-A_6=00000$ | 写“1” | (2) $A_{10}-A_6=00001$ | 写“0” |
| (3) $A_{10}-A_6=00000$ | 读“1” | (4) $A_{10}-A_6=00010$ | 写“0” |
| (5) $A_{10}-A_6=00000$ | 读“1” | (6) $A_{10}-A_6=00100$ | 写“0” |
| (7) $A_{10}-A_6=00000$ | 读“1” | (8) $A_{10}-A_6=01000$ | 写“0” |
| (8) $A_{10}-A_6=00000$ | 读“1” | (10) $A_{10}-A_6=10000$ | 写“0” |
| (11) $A_{10}-A_6=00000$ | 读“1” | | |

同理也可对其它完成地址译码的与非门完成相应的操作,只是不同的与非门的起始地址不同而已,可用一个循环来实现对所有完成地址译码的与非门该类故障的测试。

5 容错性设计方案

当地址译码器中的与非门中的 P 型 MOS 管开路时,有可能选中多个存储单元,这是因为有故障的与非门在地址变为其它地址时仍然输出为“0”,为了克服这个问题,可把存储器的时序信号作为完成地址译码的与非门的一个输入,这样在时序信号不使能时,地址变化之前,使所有完成地址译码的与非门输出均为“1”,当地址变化后,时序信号使能时,仅有一个与非门输出为“0”,仅选中一个存储单元,不会选中前一个地址所对应的存储单元(有无故障均是如此)。采用容错设计后的电路如图 4 所示,与图 1 电路不同的是时序信号后加一个反相器(为了提供正确的时序信号),原来的 5 输入与非门变为 6 输入与非门,与 A₅ 和 A₁₁ 相连的两个 3 输入与非门改为 2 输入与非门,改进后的电路未改变电路的逻辑功能,只是当译码器中出现文中所述的开路故障时,改进后的电路能够正常工作,提高了电路的可靠性。

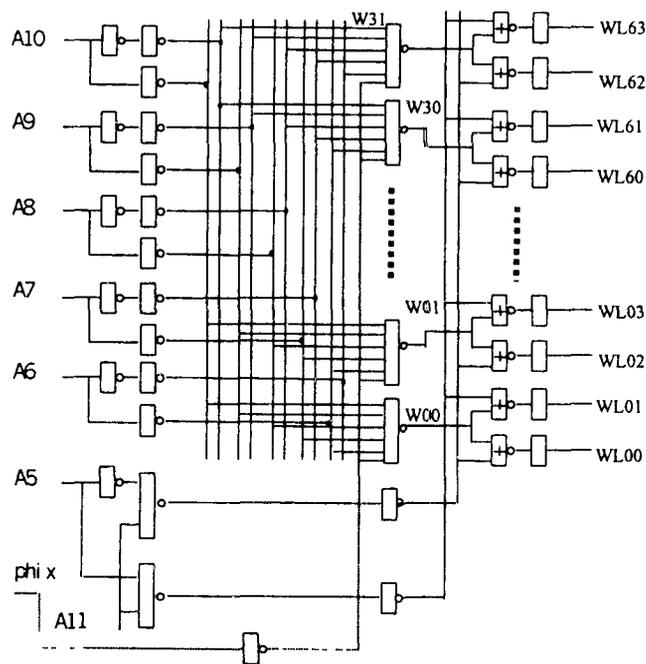


图 4 容错性设计方案

6 结论

由于 CMOS 电路结构上的特殊性,对文中所述的开路故障不能用传统的测试方法可靠的检测出的问题,用本文提出的测试该类故障的测试方法及当出现该类故障时能保证电路正常工作的容错性设计方案,在实际应用中得到了行之有效的证明。

参 考 文 献

- [1] MANOJ SACHDEV. open defects in CMOS RAM address decoders[J]. IEEE DESIGN & TEST COMPUTERS, 1997, 32(1):26~33.
- [2] 刘建都. 开路故障可测的 CMOS 电路实用设计方案[J]. 微电子测试, 1992, 1(4):24~28.
- [3] 刘建都. CMOS 存储单元的开路故障可测性设计[J]. 微电子测试, 1994, (2):22~24.

Open Defects and Testing in CMOS RAM Address Decoders

LIU Jian-du, ZHANG An-tang

(The Missile Institute, AFEU., Sanyuan 713800, China)

Abstract: At first, this paper analyzes the open defect of CMOS RAM address decoder, it comes out that one type open defect cannot be detected by march test algorithm, and then we give the test method of this type undetectable fault and the design scheme with built-in tolerance against hard-to-detect open defects.

Key words: CMOS RAM; address decoder; open defect; march test