

合肥工业大学
硕士学位论文
SRAM设计和Compiler技术
姓名：丁海涛
申请学位级别：硕士
专业：微电子学与固体电子学
指导教师：毛友德
20030618

# SRAM 设计和 Compiler 技术

## 摘 要

随着微电子技术的飞速发展,毫无疑问,21 世纪将是信息的世纪。而半导体存储器却是微电子技术的基础。在半导体存储器这一大家庭中,静态存储器由于其广泛的应用成为其中不可或缺的重要一员。近年来,由于便携式设备的流行和高性能处理器的需要,作为其中重要组件的 SRAMs,速度和功耗成为衡量产品性能的重要参数,因而低功耗和高速正成为设计的主流方向。

本文首先对半导体存储器的工作原理、结构、应用以及国内外发展的概况进行了综述。然后对 SRAM 的存储单元电路以及外围电路中的灵敏放大器和地址译码器进行了设计和模拟,在此基础上,以 128Kb 和 1Mb 全 CMOS SRAM 设计为例,从方法学角度对同步 SRAM 设计中的带时钟分等级字线译码,多级灵敏放大和位线及总线平衡等技术进行了研究,并给出了相应的 Compiler 算法。通过模拟,运用 Compiler 技术设计的 SRAM 在 128Kbit~1Mbit 密度范围内兼容,实现了高速数据读取和较低的功耗。设计出的 SRAM 已应用于专用集成电路中。对设计其它不同种类的存储器 Compiler,具有一定的参考价值。

**关键词:** 静态随机存取存储器 灵敏放大器 地址译码器 编译器 块

# **Design of SRAM and Compiler Technology**

## **Abstract**

With the rapid progress of microelectronics technology, it is undoubtedly that 21<sup>st</sup> century will be the century of information. Moreover, semiconductor memories have been the cornerstone of whole information age. In the large family of semiconductor memories, Static random access memories (SRAMs) are an indispensable part because of their broad applications. Considerable attention has been paid to the design of low-power, high-performance SRAMs since they are a critical component in both hand-held devices and high-performance processors. A key in improving the performance of the system is to use an optimum sized SRAM.

In this thesis, after a brief account of the classification, application and development of semiconductor at home and abroad, the structure and work principle of SRAM are discussed. Then, memory cell array and some parts of peripheral circuits used in SRAM, for example, sense amplifier and address decoder, are designed and verified by simulation. Furthermore, some novel methods, such as clocked hierarchical word decoding structure, multi-stage sense amplifier, common data line and data bus equilibrium technology has been applied in the design of 128Kbit and 1Mbit SRAM. What's more, we have studied compiler technology applied in the designing course of a 1Mbit full CMOS SRAM from the pointview of methodology. Finally, we present the corresponding algorithm. The compiled SRAMs realize high-speed data-read and consume low power with the density ranging from 128Kbit to 1Mbit. This thesis deserves further study when one wants to design other kinds of memory compiler.

**Key words:** Static RAM, Sense Amplifier, Address Decoder, Compiler, Bank

## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得合肥工业大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

学位论文作者签名：丁海涛 签字日期：2003年6月26日

## 学位论文版权使用授权书

本学位论文作者完全了解合肥工业大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权合肥工业大学可以将学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。

(保密的学位论文在解密后适用本授权书)

学位论文作者签名：丁海涛

签字日期：2003年6月26日

导师签名：毛友德

签字日期：2003年6月26日

学位论文作者毕业后去向：

工作单位：

通讯地址：

电话：

邮编：

## 致 谢

本论文是在毛友德教授和合肥工业大学兼职教授、世宏科技（苏州）有限公司总裁叶兆屏博士的悉心指导下完成的。两位导师学识渊博、治学严谨，对科学有着浓厚的兴趣。他们平易近人的工作作风和循循善诱的师长风范，给我留下了深刻的印象，将使我终生受益。感谢两位导师的支持、培养和无私的教诲！

该课题研究工作是在世宏科技（苏州）有限公司完成的。在课题研究期间，世宏公司为我提供了良好的软、硬件环境。在日常生活中受到了世宏公司全体员工的关照，借此机会向他们表示真诚的谢意。特别要感谢存储器组陈亮工程师在我的课题研究期间所给予的无私帮助！同时还要感谢浙江大学研究生王援伟和上海大学研究生高丽，他（她）们在设计过程中和我进行了很多有益的讨论。

感谢我的同学彭增发、郭雪峰和马先林三位研究生对我的帮助。

感谢教育过我的老师，感谢他们在我成长的道路上给予的熏陶、培养和关爱！

感谢我的父母和姐姐对我的支持和鼓励！

作者：丁 海 涛

2003 年 6 月 18 日

## 符 号 清 单

$A_v$	电压增益	$\phi_F$	平衡费米势
$A_{vc}$	共模电压增益	$V_T$	阈值电压
$A_{vd}$	差模电压增益	$\gamma$	体效应因子
CMRR	共模抑制比	$k$	器件增益因子
CL	电容负载	$k'$	工艺跨导参数
L	有效沟道长度	$\lambda$	沟道长度调制系数
PSRR	电源电压抑制比	$g_m$	跨导
W	有效 沟道宽度	$g_{ds}$	小信号沟道电导

## 前 言

我们正处于数字化社会之中。正如美国麻省理工学院著名学者 Nicholas Negroponte 在其著作《数字化生存》中所断言的那样：“我们必须掌握数字化社会的生存法则<sup>[1]</sup>。”数字数据处理正影响着社会的发展。大型机、小型机、个人计算机和笔记本电脑在我们日常生活中的使用频率日益增加。当然更重要的是不仅如此，在电子产品的每一个领域采用数字解决方案日益成为潮流。仪器制造成为公认的第一个使用数字数据控制模拟进程的非计算机领域。近年来，通讯和消费电子产品进行了一场数字化变革。电话数据通过有线和无线网络实现数字化传输和处理。CD（数字唱盘）和 DVD（数字视盘）的出现也引发了一场新的音视频革命<sup>[2]</sup>。

作为集成电路产业的基础，这些年来半导体工艺技术一直在得到迅速提高。1965 年，当时还是仙童公司创始人（后来成为英特尔公司创始人之一）的戈登·摩尔曾在 1965 年 4 月号的 Electronics Magazine 杂志上作出预言：“集成电路的集成度每三年增长四倍，特征尺寸每三年缩小 $\sqrt{2}$  倍。”这一定律为过去四十年间的事实所证明，自该定律发表以来，集成电路产业基本上是按该定律语言的速度持续发展<sup>[3]</sup>。但是，理论上早已证明集成度有物理的极限，不可能无限度地增加，这使得业界人士对半导体的发展前景似乎有些信心不足。但令人感到高兴的是，在英特尔公司 2003 年度的“未来技术发展趋势研讨会”上，与会有关专家一致认为摩尔定律至少在未来十年内继续有效。

在大量的电子产品中，都有存储数据和程序指令的要求。存储器是计算机的重要组成部分，有了它计算机才有记忆功能，才能把计算机要执行的程序、处理的数据以及计算的结果在计算机中进行存储，才能使计算机脱离人的干预自动地进行工作。

存储器的种类很多，但现在常用的是半导体存储器，因为它的容量、速度、体积、价格、功耗等各种性能指标均能满足要求，MOS 型存储器更是得到了广泛的应用，因为它集成度高，功耗低。半导体存储器一般包括 Mask ROM、EPROM、E<sup>2</sup>ROM、SRAM、DRAM 和 Flash 等。上述半导体存储器由于各自不同的特点而有不同的应用。静态随机存取存储器（SRAM）是半导体存储器中的重要一类，近年来，由于其低功耗和快速的数据存取的特点使得其发展势头强劲，现已广泛应用于多媒体技术、视频信号压缩/解压、电视传输、成像识别和音频信号、数字卫星系统、移动电话、图象处理、电子通信、语音处理和合成、声波导航武器等高新技术方面<sup>[4]</sup>。由于便携式设备的流行和高性能

处理器的需要，作为其中重要组件的 SRAMs，设计主要围绕高速和低功耗的要求进行<sup>[5]</sup>。

本课题是世宏科技（苏州）有限公司 ASIC 设计项目的一个子项目，目的在于探讨高速低功耗 SRAM 设计的新方法。本文将首先阐述 SRAM 的结构组成、工作原理和设计方法，对 SRAM 的存储单元以及灵敏放大器、地址译码器等外围电路进行设计和模拟，在此基础上，以 1Mb SRAM 为例，探讨 SRAM 的 Compiler 设计技术，给出相应的算法以及仿真结果。

我们在设计中使用以下工具。使用 Innoveda 公司的 eProduct Designer 中的 Viewdraw<sup>[6]</sup> 软件画出 SRAM 电路图并生成网表。使用 Silicon Canvas 公司的 LEO<sup>[7]</sup> 画出相应的版图。使用 Synopsys 公司的 PowerMill<sup>[8]</sup> 和 Timemill<sup>[8]</sup> 对时序和功耗进行版图前功能验证。使用 Avant!公司的 Star-HSPICE<sup>[9]</sup> 进行时序和功耗的版图后精确验证。使用 Avant! 公司的 Cosmos-Scope<sup>[10]</sup> 分析仿真波形。



# 第一章 半导体存储器的发展

本章主要从半导体存储器的起源、发展现状和分类等方面来概略地介绍半导体存储器的发展。

## 1.1 半导体存储器概述

半导体存储器最早出现于 20 世纪 60 年代，到目前为止，已有 40 多年的发展历程。当前的许多数字化设计都与数值和程序指令的存储有关。在当今的高性能微处理器中，一半以上的晶体管是用在缓冲存储器当中的，而且在可以预见的将来，这一比率还有望增加。这种状况在系统级层次上表现得尤为突出，高性能工作站和计算机包含的半导体存储器容量可达 1024M 字节以上，并且这一数字还在持续增长。因此半导体存储器通常被看作是数字逻辑系统设计中最重要微电子器件，这些系统包括计算级系统和从卫星到消费类电子产品的以微处理器为基础的系统。因此，半导体存储器由于工艺水平的进步而得到的存储容量增大和速度加快的发展，就会对其他数字逻辑系统的性能有很大的影响<sup>[2]</sup>。

随着半导体存储器应用范围的不断扩大，半导体存储器的发展也越来越迅速，它的发展可以从以下几个方面看出：

1. 工艺水平的提高：半导体存储器的制造工艺从一开始的数十微米已发展到现在的 0.12 微米左右，甚至已开始尝试 0.1 微米以下的工艺；
2. 存储容量的增大：工艺水平的提高，使得器件的特征尺寸缩小，从而使存储容量的增加成为可能，已从一开始得不到 100 位增加到现在的 1Gb 左右；
3. 功耗的降低：便携式产品（如手机和掌上电脑等）的大量出现和应用，使得半导体存储器的工作电压越来越低，已从原来的 5 伏左右降低到现在的 1.8 伏左右，同时，电源电压的降低又使得半导体存储器的功耗越来越低；
4. 速度的加快：数据的存取时间，对于可读、写存储器来说，主要是指它的读取数据时间（ $t_{aa}$  或  $t_{ca}$ ），是描述半导体存储器速度的一个最重要的参数。数据的存取时间已从原来的几百纳秒发展到现在的 3 纳秒左右（对于 SRAM 而言）。

以上的一些具体数字和文字介绍了半导体存储器的发展概况，下面用一节的内容来介绍半导体存储器的分类及其简单工作原理。

1.2 半导体存储器的分类

半导体存储器的类型可以根据存储器的功能、读取数据的方式和数据存储的原理来进行划分。如表 1-1 所示，半导体存储器一般化分为两类：一类为挥发性随机存取存储器（RAMs），另一类为非挥发性存储器。挥发性存储器的存储信息掉电后会消失，所以称这一类存储器为挥发性存储器。挥发性随机存取存储器的优点是它是最灵活的一种存储器，它可以同时提供读和写的功能及有可比性的读写时间。根据存储数据的不同原理，它又可分为静态随机存取存储器（SRAM）和动态随机存取存储器（DRAM）。其中，SRAM 是利用一个带正反馈的触发器来存储数据信息的，而 DRAM 是利用电容上的电荷来存储数据信息的，由于电容存在一个电荷泄放的问题，所以对于 DRAM 需定时刷新以弥补被泄放的电荷，从而使存储的信息不被破坏，所以称之为动态 RAM。

表 1-1 半导体存储器的分类

挥发性随机存取存储器	非挥发性存储器	
RAM	ROM	Flash
SRAM DRAM	Mask ROM PROM EPROM E <sup>2</sup> PROM	

非挥发性存储器的存储信息掉电后任然存在，所以称这一类存储器为非挥发性存储器，它主要被用于计算机、航空、远程通讯和消费类电子产品中，用来存储程序和微代码。非挥发性存储器中所存储的数据信息可以是永久不变的（理论上），或者是可编程的，这取决于存储器件的结构。它主要包括两大类存储器：一类是只读存储器（ROM），另一类为 Flash。在 ROM 中，又可分为掩模式 ROM（Mask ROM）、可一次编程 ROM（PROM）、可擦除可编程

ROM (EPROM) (这一类是指用极紫外光进行擦出的那一类 EPROM) 和电可擦除可编程 ROM (E<sup>2</sup>PROM)。Flash 的全称是 Flash E<sup>2</sup>PROM, 虽说它也属于电可擦除可编程 ROM 一类, 但由于它近几年发展非常迅速且引人注目, 所以一般都把它单独归为一类。

下面简略介绍上述半导体存储器的发展现状。

## 一. 挥发性存储器

在过去的半导体存储器发展的四十多年间, DRAM 由于其高密度和每位成本较低的优点而成为产量最高的挥发性存储器, 它被广泛地用作计算机中的主存。SRAM 的存储容量约为 DRAM 的四分之一, 因此每位的成本就约为 DRAM 的四倍<sup>[1]</sup>。但是 SRAM 具有低功耗和高速的优点, 现在 SRAM 的存储容量和性能由于工艺水平的提高, 器件几何尺寸的缩小以及电路结构的改进而得到了很大的增长, 存储密度每三年增加四倍, 存储容量现已可达到 18M 位, 速度可达 3 纳秒, 有效地改善了 SRAM 存储容量小的缺点。如今, 高速、高存储容量的 SRAM 主要用作为超级计算机中的主存、小型机和工作站中的缓存、超大规模集成电路 (VLSI) 测试设备中的测试码模式存储器, 同时它还被广泛应用于远程通讯如人造卫星, 和消费类电子产品如移动电话、存储卡、笔记本电脑、数码照相机和打印机中。

DRAM 的速度比较低的缺点也由于许多新技术的运用而得到了改善, 如同步 DRAM (SDRAM) 的出现, 其刷新速度可达到 8K/64ms, 它在保持了 DRAM 的单管单元的高密度的优势的同时还拥有快速的特点。

早期的 SRAM 是采用双极、NMOS 和 CMOS 三种互相独立的工艺制造的。到了 20 世纪 80 年代中期, 大部分的 SRAM 是采用 CMOS 工艺的。如今, SRAM 的种类繁多, 除了传统的 MOS SRAM 外, 还有高速的双极和 GaAs SRAM。全双极 SRAM, 虽说只占了不足 1% 的市场份额, 但却常被应用于高速、低存储容量的场合。目前的一些消费类 SRAM 采用“混合 MOS”工艺, 即 CMOS 和 NMOS 的组合, 用来满足高存储容量的需求, 还有采用全 CMOS 工艺用来满足高存储容量和低功耗的要求。高速和高存储容量的 SRAM 是采用 CMOS 和混合 MOS 工艺, 同时还可采用双极和 CMOS 工艺的组合, 即 BiCMOS 工艺制造。BiCMOS SRAM 综合了 MOS SRAM 存储单元面积小的优点和双极型外围电路速度快, 速度好的优点。除了双极和 MOS 这两种被称为“体硅”技术的存储器外, 还发展了运用绝缘体上的硅 (SOI) 的隔离工艺如蓝宝石上的硅 (SOS) 所制造的 SRAM, 这种 SRAM 的抗辐射能力大大增强。BiCMOS DRAM 与全 CMOS DRAM 相比, 具有特定的优点, 特别是在存取时间方面有很大的改善。

二. 非挥发性存储器

1. 掩模式只读存储器 (Mask ROM)：Mask ROM 的一个典型运用的例子是用在如洗衣机、计算器和游戏机等有固定用途的产品的处理器中，用来存储那些固定程序。存储在其中的信息是永恒不变的。它的特点是结构简单，易于制造，但灵活性差。

2. 可一次编程 ROM (PROM)：它可以让客户根据自己的需要进行编程，一般是在它的存储单元中加入一些熔丝（可以是镍铬合金、多晶硅或其他导体），然后就根据需要通过给某些熔丝加上大的电流使其熔断而是与器连接的晶体管失效，从而达到编程的目的。它的特点是可编程但只能编程一次。这样的话，在编程中出现的一个小小的错误，就会让整个器件报废。为了解决此问题，人们就又研制出了一些可重复编程的存储器，虽说它们的编程速度比较慢。

3. 极紫外线光可擦除可编程 ROM (UVROM)：它的核心部件就是一个浮栅晶体管 (Floating gate transistor)，其结构如图 1-1 所示。浮栅晶体管的最重要的特性是它的阈值电压的可编程性。当在图 1-1 所示结构的栅-源和漏-源之间分别加上 15-20V 的大电压，则产生的大电场将会引起雪崩注入，衬底中的电子就会获得足够的能量而变为“热电子”并穿过第一层氧化绝

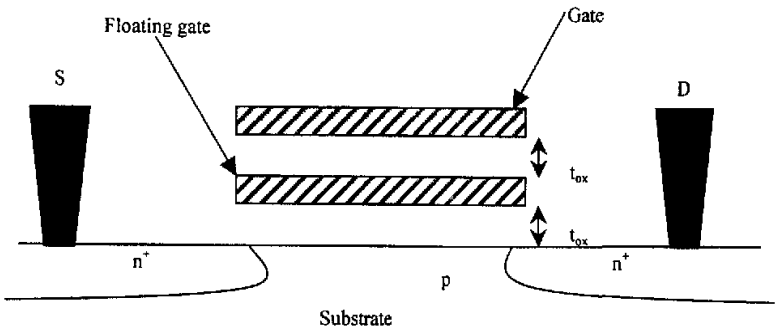


图 1-1 浮栅晶体管的截面示意图

缘层，被浮栅俘获，这就称之为浮栅晶体管的编程过程<sup>[12]</sup>。由于浮栅的四周被绝缘性能优良的二氧化硅 (SiO<sub>2</sub>) 所包围，所以即使在不加电源电压的情况下，浮栅上被俘获的电子仍能保持很多年（一般为十年左右），因此就产生了非挥发性存储器。由于浮栅上所俘获的电子使得在其上产生一个负电压，从一个器件的角度来看，这将意味着阈值电压的增大（典型值为 7V 左右），所以在栅-源之间加上普通的 5V 电压已不足以打开此器件。UVROM 就是通过极

紫外线光来进行擦除的，极紫外线光通过在氧化层内直接产生电子-空穴对而致使其轻微地导电，将存储在浮栅上的电子泄放掉，达到擦除的目的。UVPROM 的擦除速度很慢，根据极紫外线光强度的不同需几秒钟到几分钟的不等的擦除时间，而且擦除时还需将其从系统中拿出，放到特定的装置中用极紫外线光进行擦除，且只能进行全部擦除，很不方便。此种存储器还存在擦除次数有限的缺点，一般可擦除 1000 次左右。但是 UVPROM 也有其特有的优点，它的存储单元简单，存储容量可与 DRAM 媲美，这使得其在制造大容量存储器时成本较低。综上所述，UVPROM 适宜用于制造大容量且不需要经常编程的存储器。

4. 电可擦除可编程 ROM ( $E^2$ PROM)： $E^2$ PROM 的核心部件也是一个浮栅晶体管，其结构示于图 1-2 中。此结构与图 1-1 所示的结构很相像，只是

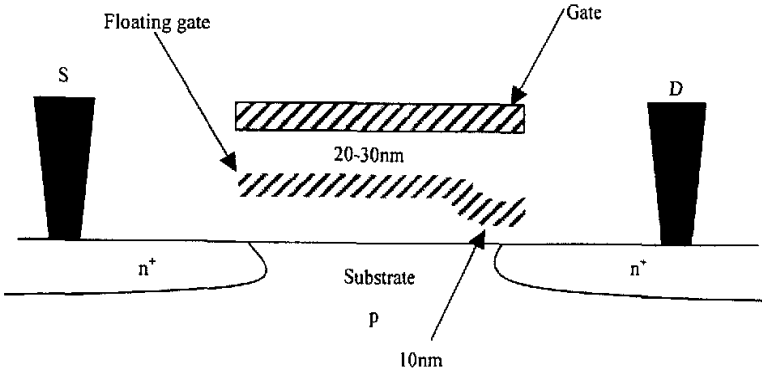


图 1-2 浮栅隧道氧化层晶体管

用来使浮栅与沟道和漏极隔离的部分绝缘层的厚度减小到 10 纳米或更低。当在薄的氧化层上加上 10V 左右的电压时，电子通过隧穿效应到达浮栅并被其俘获，完成编程操作<sup>[13]</sup>。此隧穿效应的 I-V 曲线如图 1-3 所示，由图可见，此

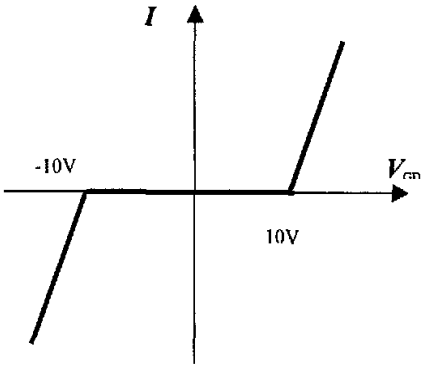


图 1-3 I-V 特性曲线

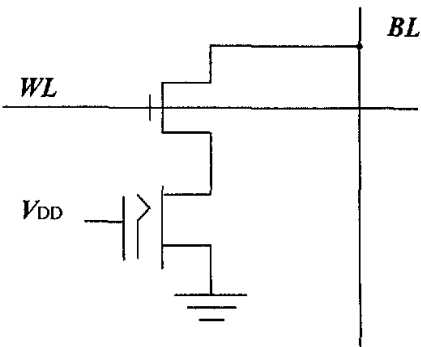


图 1-4  $E^2$ PROM 存储单元

种编程机构的主要优点是其可逆性，即只要将编程时的电压反向就可实现擦除过程。 $E^2PROM$  与  $UVPROM$  相比，既有优点也有不足之处。优点是擦除速度快，可支持  $10^5$  擦除/编程周期和可有选择的进行擦除/编程。不足之处在于当注入电子到达浮栅上时，会增大器件的阈值电压，而当进行擦除操作时，又会降低  $V_T$ ，这样就存在如何控制阈值电压的问题。从浮栅上移去过多的电子，将会导致耗尽型器件的产生，从而使得标准的字线电压无法关断器件，引起错误操作，所以  $E^2PROM$  的存储单元中都包括两个晶体管，一个为浮栅晶体管，用作为存储管，另一个为普通的晶体管，由字线控制，用作为开关管，如图 1-4 所示，因此  $E^2PROM$  比  $UVPROM$  存储单元所占的面积大，且  $E^2PROM$  中特殊结构的浮栅晶体管的造价很高并难以制造，这些都使得  $E^2PROM$  的每位的成本比  $UVPROM$  高且存储容量比  $UVPROM$  小。

5. Flash  $E^2PROM$  (Flash)：Flash 出现于 1984 年，此后就得到了迅猛发展，现已成为应用最广泛的存储器之一。大部分 Flash 的擦除采用的是与  $E^2PROM$  相同的方法即隧穿效应，而编程采用的是与  $UVPROM$  相同的机制即热电子注入效应。

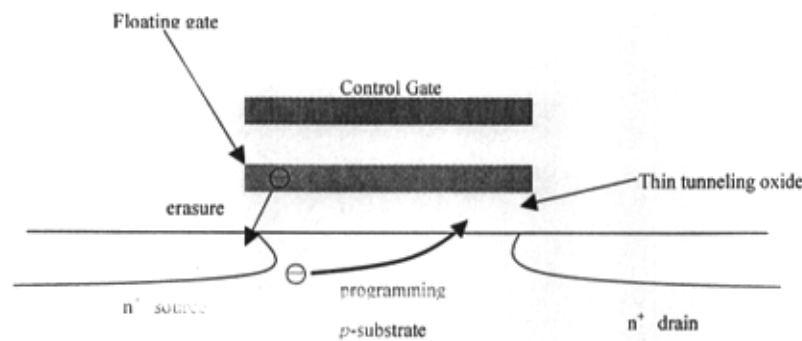


图 1-5 Flash 存储单元结构

图 1-5 所示的 Flash 存储单元结构由 Intel 公司提出<sup>[14]</sup>，它与图 1-1 所示得浮栅晶体管的结构相似，不同的是图 1-5 所示的结构中采用了非常薄的一层隧道氧化层，利用栅氧化层的不同部位分别进行编程和擦除操作。编程时，在栅和漏极上加上 12V 左右的电压，而源极接地，产生热电子注入，擦除时，栅极接地，源极加上 12V 左右的电压，利用薄氧化层的隧穿效应进行擦除操作。与一般的  $E^2PROM$  相比，Flash  $E^2PROM$  不支持有选择的擦除/编程，这虽然降低了它的灵活性，但却可以省略一般的  $E^2PROM$  中多余的开关管，使得它的存储单元减少了 2~3 倍，存储容量大大增加，现已做到 128M 位的存储容量，甚至已开始尝试 1G 位的 Flash 的开发。一次性擦除全部存储单元的信息，需要在擦

除过程中对器件特性的严格检测，以确保未被编程的器件为增强型器件。存储芯片上的监测部件会定期地擦除过程中检查器件的阈值电压，并动态调整擦除时间。上述方法只是在一次性擦除大块存储内容，甚至是整个 Flash 的内容时才是可行的。Flash 主要用于高密度非挥发性存储类的产品中，如存储器模块和存储卡等。

### 1.3 半导体存储器的国内外发展概况

微电子技术的核心是集成电路，所以集成电路是当今世界上发展速度最快、更新换代也最快的电子产品。而存储器始终是一种代表集成电路技术发展水平的典型产品。存储器在集成电路中技术最先进、产量最多、市场最大，因而存储器的研制和生产水平历来就是衡量一个国家科学技术和工业发达程度的标志。存储器驱动着各种工艺和生产技术的发展，它不仅使芯片生产厂商的技术“驱动器”，而且也是半导体材料和设备公司重要的“驱动器”。几乎所有发达国家都把存储器作为发展集成电路技术的突破口。正因为存储器在集成电路产品中的重要地位，研究存储器技术的发展便有着特殊的代表意义。

#### 一、国外发达国家发展动态

1971 年世界上第一块 DRAM 首先诞生于美国，不久 4Kb DRAM 也开始在美国大量生产，其市场占有率一度高达 91%。

二十世纪七十年代中期日本开始加入存储器竞争的行列。他们大举投资，大量引进美国设备和技术，凭借其大规模低成本生产优势和低价倾销战略，逐渐取得了存储器领域的领先地位。

到二十世纪八十年代，韩国步日本之后尘，在存储器领域获得了许多美国和日本的技术。标准化的技术和大量投资，是韩国的低廉劳动力优势得以充分体现。1985 年韩国三星电子公司成为当年 DRAM 的世界头号供应商，年产量达 6000 万块。

存储器的产品寿命周期很短，平均每一代产品的寿命期不足 4 年，工艺技术每跨上一个新台阶，总要以新一代的产品作为它的代表，同时把上一代的产品送上衰落的起点。目前市场上 256Kb、1Mb、4Mb、16Mb DRAM 等 4 代产品共存，4Mb DRAM 市场销量已超过 1Mb DRAM，成为市场的主流产品，16Mb DRAM 也于 1992 年初由 IBM 公司率先投入市场，64Mb DRAM 自 1990

年由日立公司最先开发成功后，1991年2月，东芝、富士通、三菱、松下等四家公司也公布了他们的64Mb DRAM开发成果，继后，美国和韩国也完成了64Mb DRAM的开发工作。在1993年2月的ISSCC会议上，东芝、NEC和日立等三家日本公司同时公布了他们最新研制成功的256M DRAM的成果。这意味着半导体存储器又推向新一代，工艺技术也登上了一个新的台阶，即从0.35 $\mu\text{m}$ 提高到0.25 $\mu\text{m}$ ，集成度从1.4亿个元件（64Mb DRAM）增加到5.6亿，提高了三倍，到九十年代末期，开发成功1Mb DRAM。

此外，高速SRAM和快闪存储器（Flash）也有较大发展。由其用于工作站的高速SRAM的发展更为迅速。1992年NEC公司推出的12ns 16Mb CMOS SRAM和4ns 4Mb SRAM，富士通公司开发成功的15ns 16Mb CMOS SRAM等都是该领域的最新成果。Flash以其性能好、功耗低、体积小、重量轻等优点博得了众多半导体厂商的青睐，在十大热门产品中居首位。Intel公司在此领域仍处于领先地位，1992年相继推出了1Mb、8Mb和20Mb的产品。此外，东芝公司联合IBM、National和韩国三星电子等有力伙伴，推出可擦写100万次的商用16Mb芯片，并在1993年和1994年实现32Mb和64Mb产品的商品化，1995-1996年完成128Mb产品的开发。

当前，各种存储器的产量和市场占有率的排序依次为DRAM、SRAM、ROM、EPROM、E<sup>2</sup>PROM和Flash。再过几年，上述排序会稍有变化，Flash的排位会靠前。今后存储器的发展方向依然是高密度、微细化、高速、低功耗、高性能价格比、多功能品种和封装小型化。

## 二、国内存储器的发展现状

我国的半导体技术仍然很落后，与国外发达国家相比还存在很大的差距，但经过长期的努力，我国存储器技术领域还是取得了一些成绩。1986年中国华晶电子集团公司研制成功了第一块2.5 $\mu\text{m}$ 工艺每块芯片含15万个元件的64Kb DRAM，标志着我国进入了VLSI时代。1990年清华大学微电子学研究所研制出1-1.5 $\mu\text{m}$ 工艺1Mb汉字ROM的样品。我国到2000年研制出4Mb DRAM，大量生产出1Mb DRAM。存储器的研制和生产水平历来是衡量一个国家科学技术和工业发达程度的标志。国家应制定长远规划和政策扶植其发展，如加速科研成果商品化、保护国内市场等，逐步缩小与发达国家之间的差距。

令人感到欣慰的是我国已经开始认识到振兴集成电路产业的重要性和迫切性，2000年财政部、国家税务总局和海关总署联合下发了《关于鼓励软件产业和集成电路产业发展有关税收政策问题的通知》，给相关企业提供了减免税收等优惠政策<sup>[15]</sup>。



上面两节介绍了半导体存储器的分类情况和发展现状。由于本论文的重点是要论述 SRAM 设计和 Compiler 技术，下面一章首先介绍 SRAM 的结构和工作原理，为论文的重点论述做准备。

## 第二章 静态随机存取存储器概述

SRAM 的发展现状已在上一章做了介绍，本章主要讨论 SRAM 的结构和工作原理。

### 2.1 SRAM 的结构

图 2-1 所示的是一个存储容量为  $2^m \times N$  位的 SRAM 的结构框图。

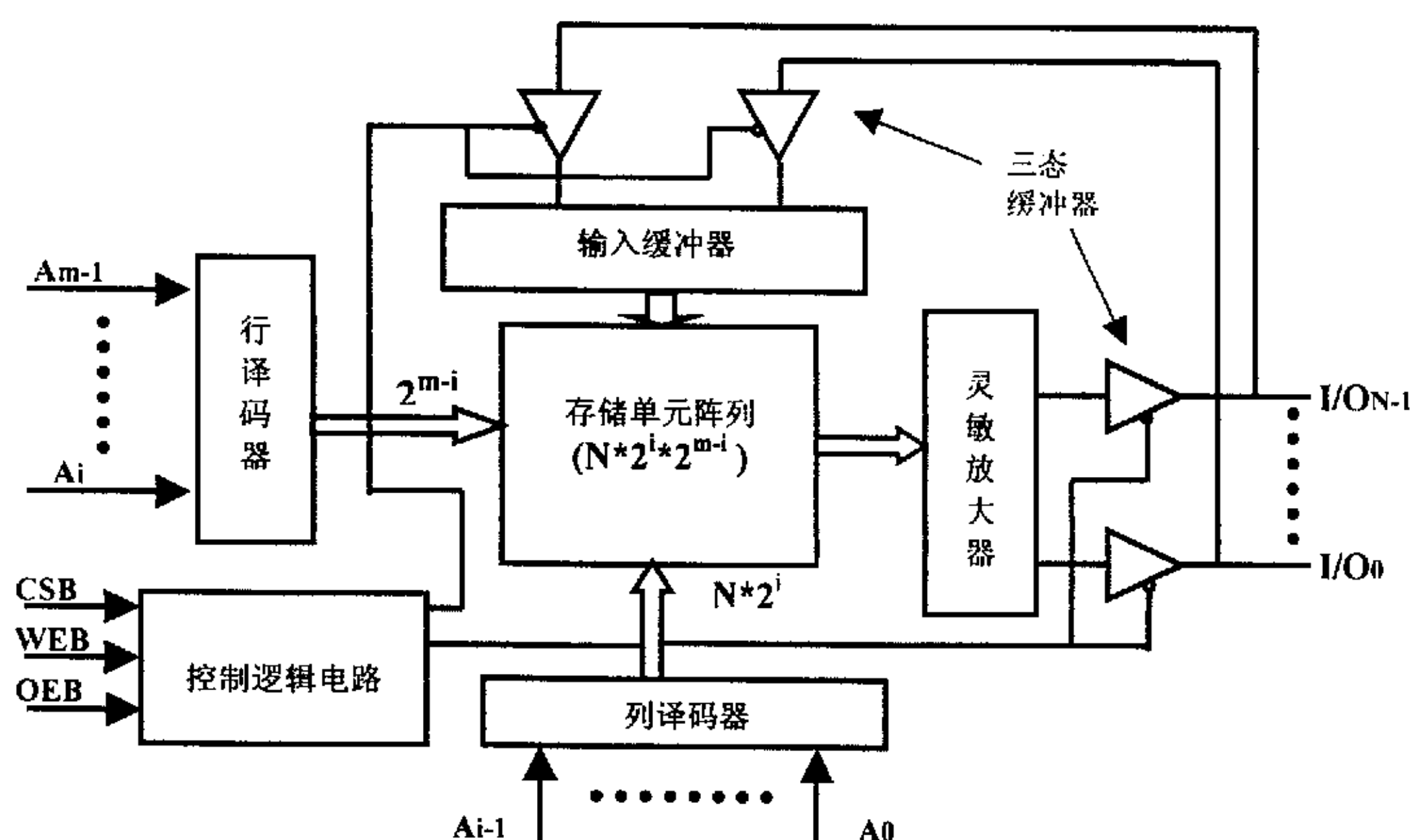


图 2-1 SRAM 的结构框图

由上图可看出 SRAM 一般由五大部分组成，即存储单元阵列、地址译码器（包括行译码器和列译码器）、灵敏放大器、控制电路和缓冲/驱动电路。在图 2-1 中， $A_0 - A_{m-1}$  为地址输入端， $CSB$ 、 $WEB$  和  $OEB$  为控制端，控制读写操作，均为低电平有效， $I/O_0 - I/ON-1$  为数据输入/输出端。存储阵列中的每个存储单元都与其它单元在行和列上共享电学连接，其中水平方向的连线称为

“字线”，而垂直方向上的数据流入和流出存储单元的连线称为“位线”。通过输入的地址可选择特定的字线和位线，字线和位线的交叉点就是被选中的存储单元，每一个存储单元都是按这种方法被唯一选中，然后再对其进行读写操作。有的存储器是设计成多位数据如 4 位或 8 位等同时输入和输出的，这样的话，就会同时有 4 个或 8 个存储单元按上述方法被选中进行读写操作。

在 SRAM 中，排成矩阵形式的存储单元阵列的周围是译码器和与外部信号的接口电路。存储单元阵列通常采用正方形或矩阵的形式，以减少整个芯片面积并有利于数据的存取。以一个存储容量为 4K 位的 SRAM 为例，共需 12 条地址线来保证每一个存储单元都能被选中 ( $2^{12}=4096$ )。如果存储单元阵列被排成只包含一列的长条形，则需要一个 12/4K 位的译码器，但如果排列成包含 64 行和 64 列的正方形，这时则只需一个 6/64 位的行译码器和一个 6/64 位的列译码器，行、列译码器可分别排列在存储单元阵列的两边，64 行和 64 列共有 4096 个交叉点，每一个点就对应一个存储位。因此，将存储单元排列成正方形比排列成一列的长条形要大大地减少芯片的面积。存储单元排列成长条形除了形状奇异和面积大以外，还有一个缺点，那就是排在列的上部的存储单元与数据输入/输出端的连线就会变得很长，特别是对于容量比较大的存储器来说，情况就更为严重，而连线的延迟至少是与它的长度成线性关系，连线越长，线上延迟就越大，所以就会导致读写速度的降低和不同存储单元连线延迟的不一致性，这些都是在设计中需要尽量避免的。

存储单元阵列是 SRAM 的核心部分，其余部分统称为 SRAM 的外围电路。下面分别阐述这两大部分的内容。

### 2.1.1 存储单元阵列

SRAM 是依靠存储单元中的具有正反馈特性的交叉耦合反向器对所构成的锁存器来存储“0”或“1”信息的，掉电后，信息即丢失，所以电源电压一直要加在 SRAM 上，以使其存储单元能被所存在正确的状态，因此 SRAM 属于挥发性存储器。

图 2-1 所示的 SRAM 中共有  $2^m \times N$  个存储单元，排列成  $2^{m-1} \times N2^i$  的矩阵形式。大容量 SRAM 和其它一些半导体存储器面积的大小主要是由其存储单元阵列所占的面积决定的，所以对于基本存储单元的要求是要面积尽可能的小，同时还要求存储单元功耗低、读写响应时间快且工艺简单。目前最常用的 CMOS SRAM 存储单元结构如图 2-2 所示。其中，两个 PMOS 管 M5 和 M6 是负载管，而栅极与字线 (WL) 相连的两个 NMOS 管 M3 和 M4 起开关的作用，与 M5 和 M6 构成反向器对的两个 NMOS 管 M1 和 M2 是存储管。BL 和 BLB 为两条位线，数据的原信息和反信息分别存放在 Q 和 QB 点。负载管 M5

和 M6 主要是用来补偿存储管和开关管的漏端电荷的泄放。此种结构的静态功耗为零，易于制造，所以被广泛应用于现在的 SRAM 中。最开始采用的是 SRAM 存储管的不同，它采用的是耗尽型 NMOS 管，它的静态功耗比 CMOS 结构的大但速度稍快。

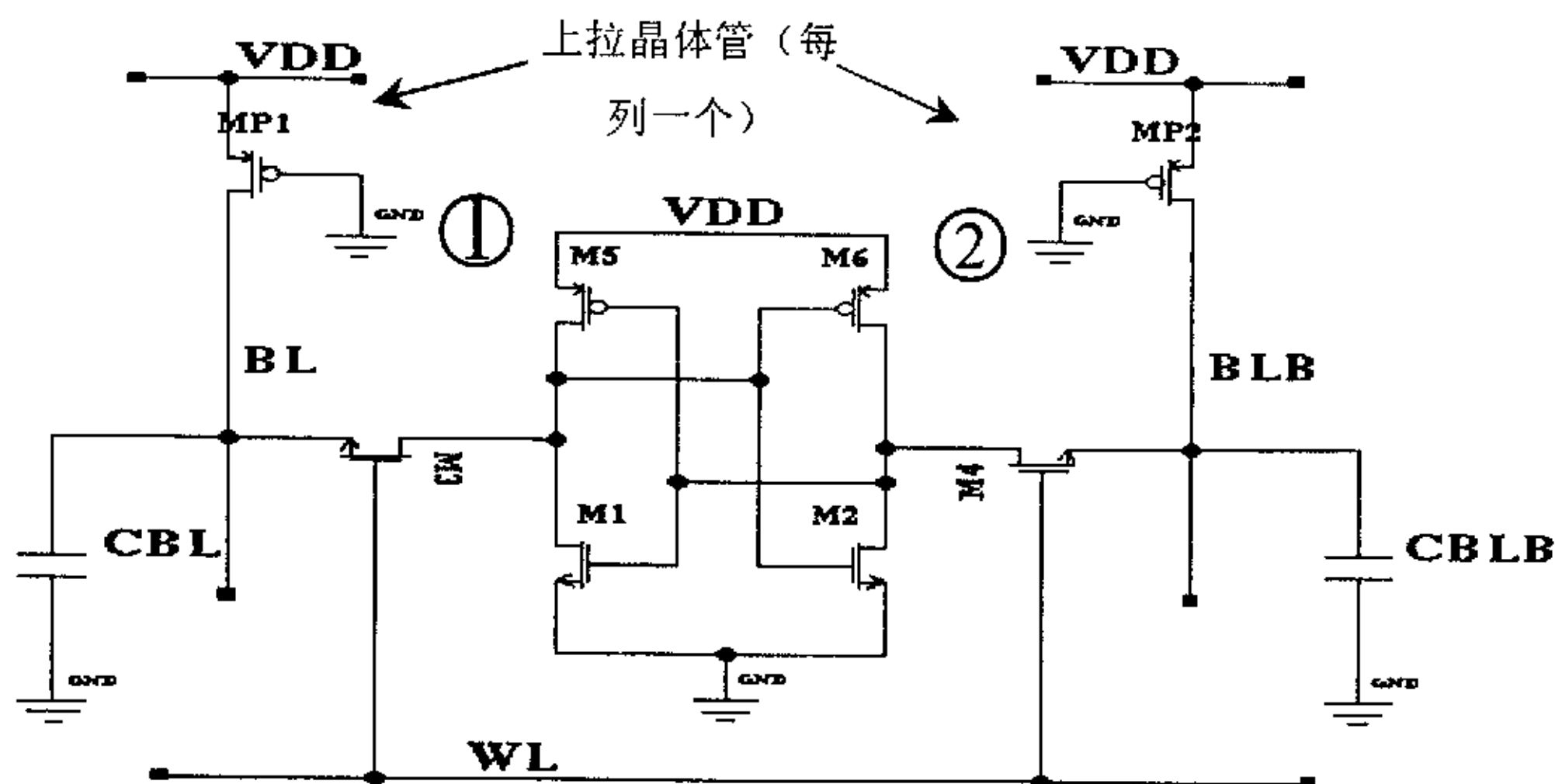


图 2-2 SRAM 常用六管存储单元

六管 MOS SRAM 存储单元虽说结构简单且稳定性好，但是都会占用很大面积。对于存储容量大的 SRAM 来说，这样的单元结构就会不适用，于是出现了一种被称为“四管 SRAM 单元”或“电阻负载 SRAM 单元”的单元结构。它用电阻代替了图 2-2 中的 PMOS 管，这样电路连线就会变得简单。在 1M 位 SRAM 中，运用四管单元作为基本存储单元比用六管单元结构，可使存储单元面积减少 30% 左右，但这种结构的缺点就是工艺复杂并且电阻值的准确设定比较困难，因为一方面希望电阻值能尽可能的高，以便于减少静态功耗，另一方面电阻值太高又会使从低到高的传输延迟严重恶化和造成单元面积的增加。

在高性能的超级计算机和大型机中都需要用到速度极快的 SRAM，使得其中的缓存和主存的周期时间可以达到所要求的 5 纳秒以下。到目前为止，还是只有双极型的才能满足上述要求。如今双极型存储单元的发展方向是降低结构复杂度或进一步提高速度。双极型存储单元的特点是高速但结构复杂且面积较大。

除了上述的一些 SRAM 存储单元结构外，现在还出现了有着特定用途的 SRAM 基本存储单元结构。图 2-3 和图 2-4 给出了其中的两种结构。

图 2-3 所示的是一种八晶体管，具有双数据通道和双端口的存储单元结构。它包括 BL1 和 BLB1、BL2 和 BLB2 两组位线和 WL1 与 WL2 两条字线。可以

同时通过两个端口和两条通道到达芯片。此种结构特别适用于微处理器芯片的内置缓存中。

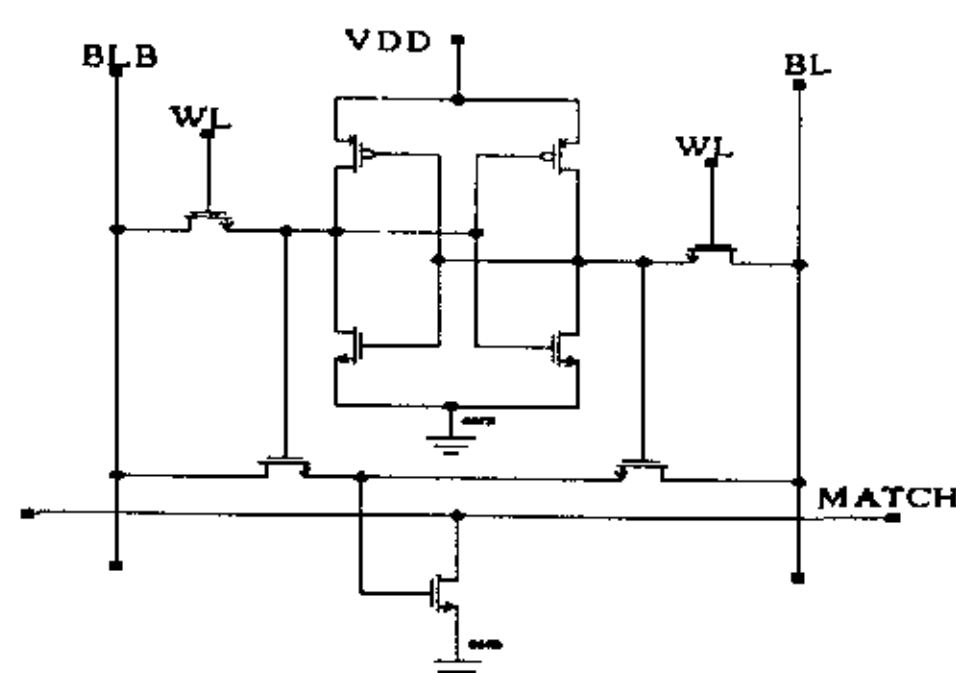


图 2-3 双端口 SRAM 存储单元

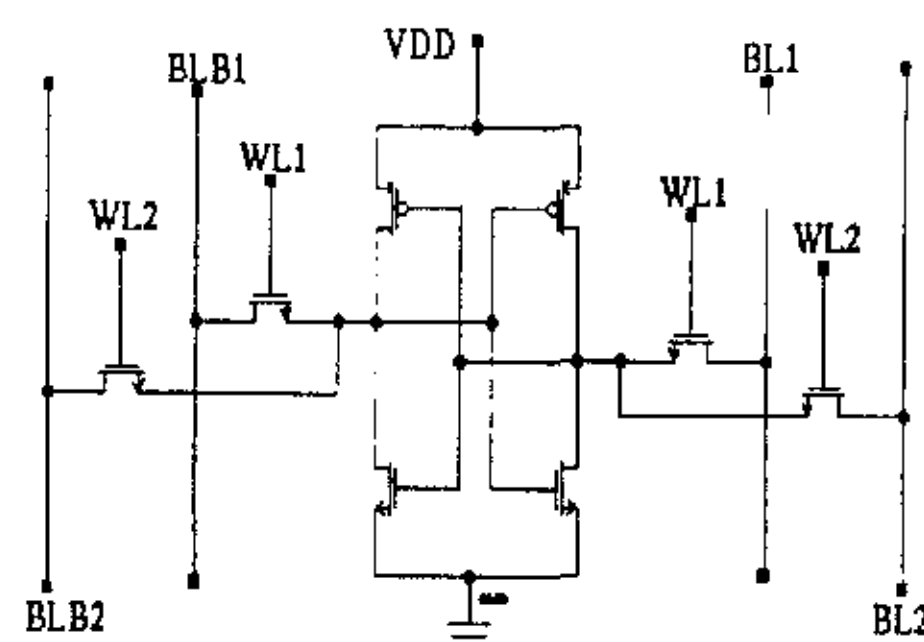


图 2-4 内容和地址可同时获得的  
SRAM 存储单元

图 2-4 所示的是一种九晶体管，可在获得单元内容的同时获得单元的地址的存储单元结构。它主要用于在得到存储单元里存储的内容的同时还要求知道存储单元所处的位置的产品中。

## 2.1.2 外围电路

为了减小 SRAM 或其它类型半导体存储器的面积，我们希望其中的基本单元的面积越小越好，在减少单元面积的同时势必会影响其它对数字电路来说很有价值的一些参数，如噪声容限、逻辑摆幅、输入/输出的隔离、扇出能力或速度等。正是因为牺牲了上述性能才得以让如今的存储器的基本存储单元可以被控制在一个到六个晶体管之间。虽说这些性能的降低在存储器的核心部分这一有限领域是可以接受的，但是当存储器与外部电路相连的时候，这种情况是绝不允许发生的。所以就需要在存储器中加入一些外围电路来补偿为了减少基本存储单元面积而造成的性能的降低，从而使得存储器在与外围电路相连时能正常有效地工作。在 SRAM 中，外围电路一般包括灵敏放大器、地址译码器、三态输入/输出缓冲器和控制逻辑等。下面就分别简略介绍这些基本的外围电路的主要作用及特性。

### 2.1.2.1 灵敏放大器

灵敏放大器是 SRAM 中必不可少的一个组成部分，它用在 SRAM 中所起的作用主要表现在以下几个方面：

1. 放大----这是灵敏放大器 SRAM 中的最主要的作用。因为从存储单元

到达位线（BL）及位线#（BL#）上的信号之间的电压幅值差很小，一般约为0.5V左右，这样的幅值是不适合标准的逻辑“高”和“低”电平值，如果直接把位线上的信号加到外部电路上，那么外部就会由于无法辨认信号的逻辑值而不能正常工作，所以就需要在位线和三态输出驱动级之间设置一灵敏放大器，将位线上的信号查差放大成标准的逻辑电平“1”和“0”输出。

2. 改善性能----灵敏放大器能通过加快位线的状态转换而补偿存储单元被限制的扇出驱动能力，并有助于提高速度。

3. 减小功耗----灵敏放大器的存在可以减小位线上的电压幅值差，从而能够消除很多对位线的充、放电而造成的功耗。

关于灵敏放大器 SRAM 中的具体用法和对它的一些结构的介绍和有关讨论，将在第四章中给出。

### 2.1.2.2 地址译码器

任何一个支持随机存取的半导体存储器中都很肯定会有地址译码器的存在，因此 SRAM 也不例外，它的外围电路中的一个很重要的部件就是地址译码器。从 SRAM 的一般结构图可以看到，地址译码器包括行译码器和列译码器。采用行、列译码其主要是为了与排成矩阵形式的存储单元阵列在形状和尺寸上能够匹配，因为它们直接与存储单元阵列相连，若不匹配的话，就会导致连线长度的增加以及由此所带来的延迟和功耗的相应增加。

地址译码器主要是对存储单元进行编码，使得在任何时刻都只有特定的单元被选中（是由输入的地址决定的）进行读写操作，因为地址译码器可以保证对应任何一组输入地址都只有唯一的输出是有效的，这样的话，每次由行译码器选中每一行，由列译码器选中每一列，然后在行、列的交叉处的存储单元就是被选中进行读写操作的那一个。应用地址译码器还可以减少封装的管脚数，因为译码器的输入与输出之间满足  $N=2^n$  的关系，其中  $N$  是其输入端数， $2^n$  为其输出端数。仍以 4K 位 SRAM 为例，若不采用地址译码器，则共需  $2^{12}=4096$  个地址输入端，来保证每一个存储单元都能对应一个特定的地址。若采用行、列译码器（存储单元阵列为  $64 \times 64$ ）结构，则只需两个 6/64 位的二进制译码器，共 12 个地址输入端。输入端数比值为  $4096:12=341$ ，这就大大降低了封装难度。

关于地址译码器的结构将会在第四章中详细讨论。

### 2.1.2.3 控制电路

SRAM 的读写操作都是由一系列的过程按顺序来完成的，所以需用控制电



路来保证其能正确且有效的工作。图 2-1 所示的 SRAM 结构框图中，CSB、WEB 和 OEB 分别称为片选控制端、写控制端和读控制端。在实际应用中，并不是直接由它们来控制读写，而是通过一定的控制电路，使三者经过逻辑组合后，再产生两个信号，来分别控制读写操作。

图 2-5 所示的控制电路中，CSB、WEB 和 OEB 为输入端，WE' 和 OE' 为输出端，分别称为内部写允许和内部输出允许，可用来直接控制 SRAM 的写入

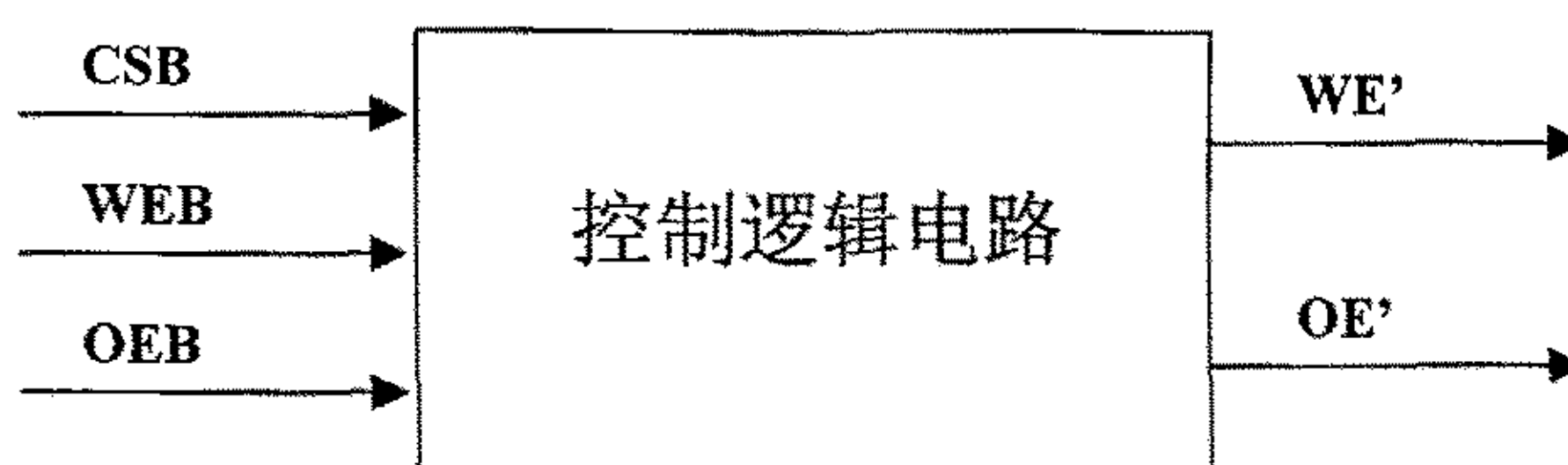


图 2-5 控制电路示意图

和读出电路，均为低电平有效。控制电路的真值表如表 2-1 所示。

表 2-1 SRAM 控制电路真值表

CSB	OEB	WEB	WE'	OE'	SRAM 状态
1	X	X	1	1	维持
0	X	0	0	1	写入
0	0	1	1	0	读出
0	1	1	1	1	不读不写

由表可得，当 CSB 为“1”，也即整个 SRAM 未被选中时，无论 OEB 和 WEB 的状态如何，SRAM 都不会工作，维持原状态。只有当 CSB 为“0”时，SRAM 才可能正常工作，此时当 WEB 为“0”时，无论 OEB 的状态如何，WE' 都为“0”，即写入电路打开，可对选中的存储单元进行操作，当 OEB 为“0”且 WEB 为“1”时，OE' 为“0”，即读出电路打开，可进行读操作，而当 OEB 和 WEB 都为“1”时，则 WE' 和 OE' 均输出“1”状态，写入和读出电路均关闭，读写操作皆不进行。

对于存储容量比较大的异步 SRAM，为了减小功耗和提高速度还会采用一种“地址转换监控电路”（ATD），它可以通过监测外部信号的变化而自动产生内部控制信号如 SE，控制灵敏放大器开关的信号，从而有效地降低功耗和提高速度。

#### 2.1.2.4 输出驱动电路

随着存储器存储容量的不断加大，字线和位线的长度也随之增加，这将会导致数据存取时间的加大，速度变缓，性能变差，为了改善这一状况，同时为了增强输出端的驱动能力，就需要在数据的输出端设置一个输出驱动器，通常都采用三态推挽电路<sup>[16]</sup>。其示意图示于图 2-6 中。其中，DIN 和 DINB 为从灵敏放大器输出的两个反向数据，DIN 与存储单元中存储的数据一致，而 DINB 为它的反信号，OE' 为输出使能端，低电平有效，D 为最终的数据输出。当 OE' 为“1”时，不管 DIN 和 DINB 为何种状态，输出端 D 都为高阻态，当 OE' 为“0”时，输出端 D 就会跟随 DIN 的变化，但驱动能力比 DIN 大。

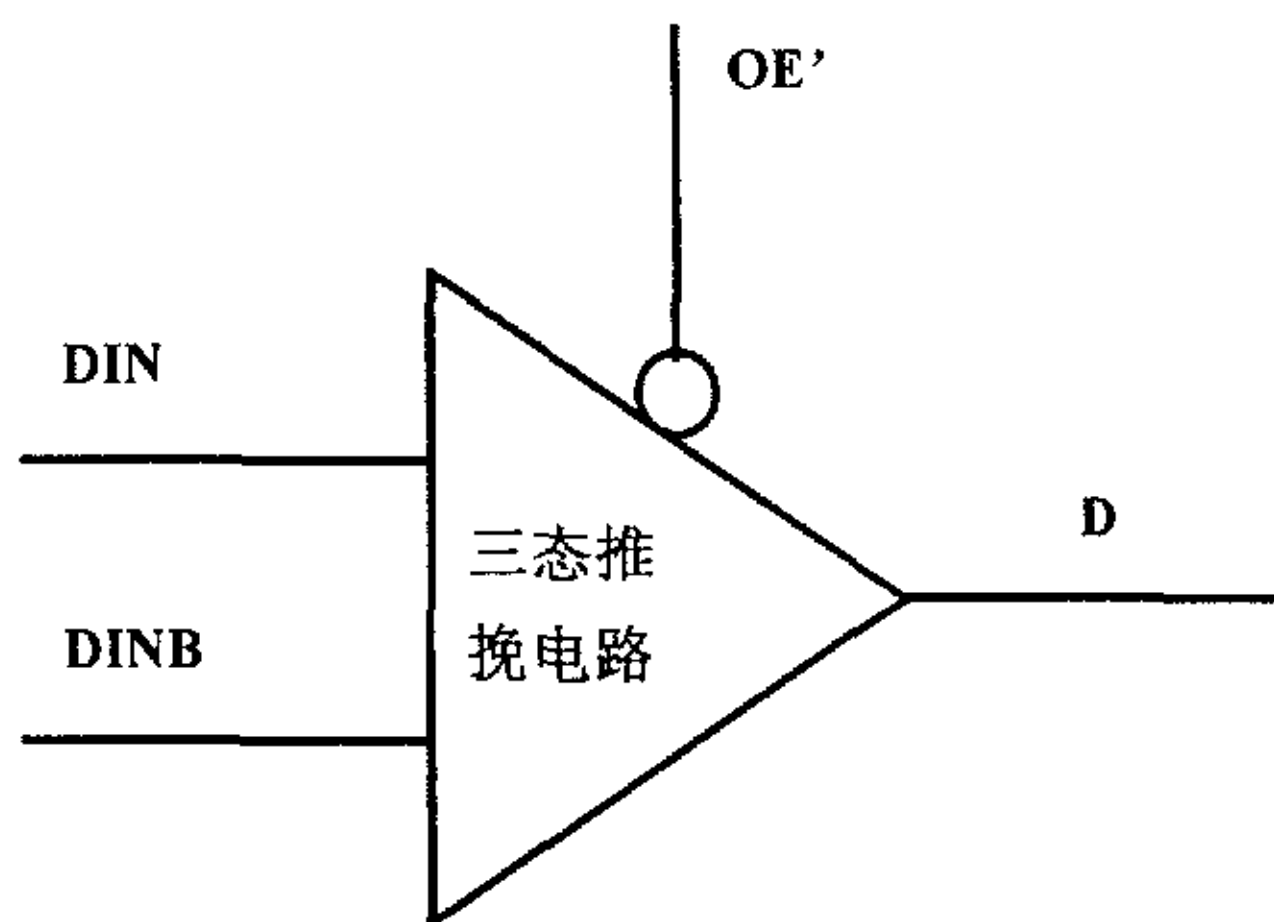


图 2-6 三态输出驱动器示意图

以上介绍的是一般用途的 SRAM 的结构，现在为了满足某些特殊需要还出现了很多其它类型的 SRAM，如零总线等待 SRAM（No Bus Latency SRAM，简称为 NoBL SRAM），此种结构的 SRAM 适用于读写转换比较频繁的系统中，因为它在连续的读/写转换状态间不会插入任何的等待状态，这样就会大大地提高系统的工作速度。SRAM 还可根据其工作过程是否受时钟的控制分为同步和异步两种，若其中所有的工作过程都是由时钟信号控制的，则为同步 SRAM，否则就为异步 SRAM。两者相比较，同步 SRAM 的速度快但结构较复杂，异步 SRAM 的结构简单但速度较慢。本文所设计的 SRAM 均为同步 SRAM。



## 2.2 SRAM 的工作原理

### 一. SRAM 的写过程

假设准备往存储单元采用图 2-2 所示的六管 CMOS 结构的 SRAM 中写入“1”，则先使片选信号 CSB 有效（CSB=“0”），将某一组地址值输入到行、列译码器中，选中特定的单元，然后使写使能信号 WE 有效

（WE=“0”），将要写入的数据“1”通过写入电路变成了“1”和“0”后分别加到选中单元的两条位线 BL 和 BLB 上，此时选中存储单元中的 WL=“1”，晶体管 M5 和 M6 打开，把 BL 和 BLB 上的信号分别传送到 Q 和 QB 点，从而使 Q=“1”，QB=“0”，这样数据“1”就被锁存在由晶体管 M1、M2、M3、M4 所构成的锁存器中。写入数据“0”的过程与此类似。

SRAM 的写周期时序如图 2-7 所示。其中，Ai 为地址输入信号，CSB 和 WEB 分别为外部片选和写使能信号，Data In 为写入的数据信号。

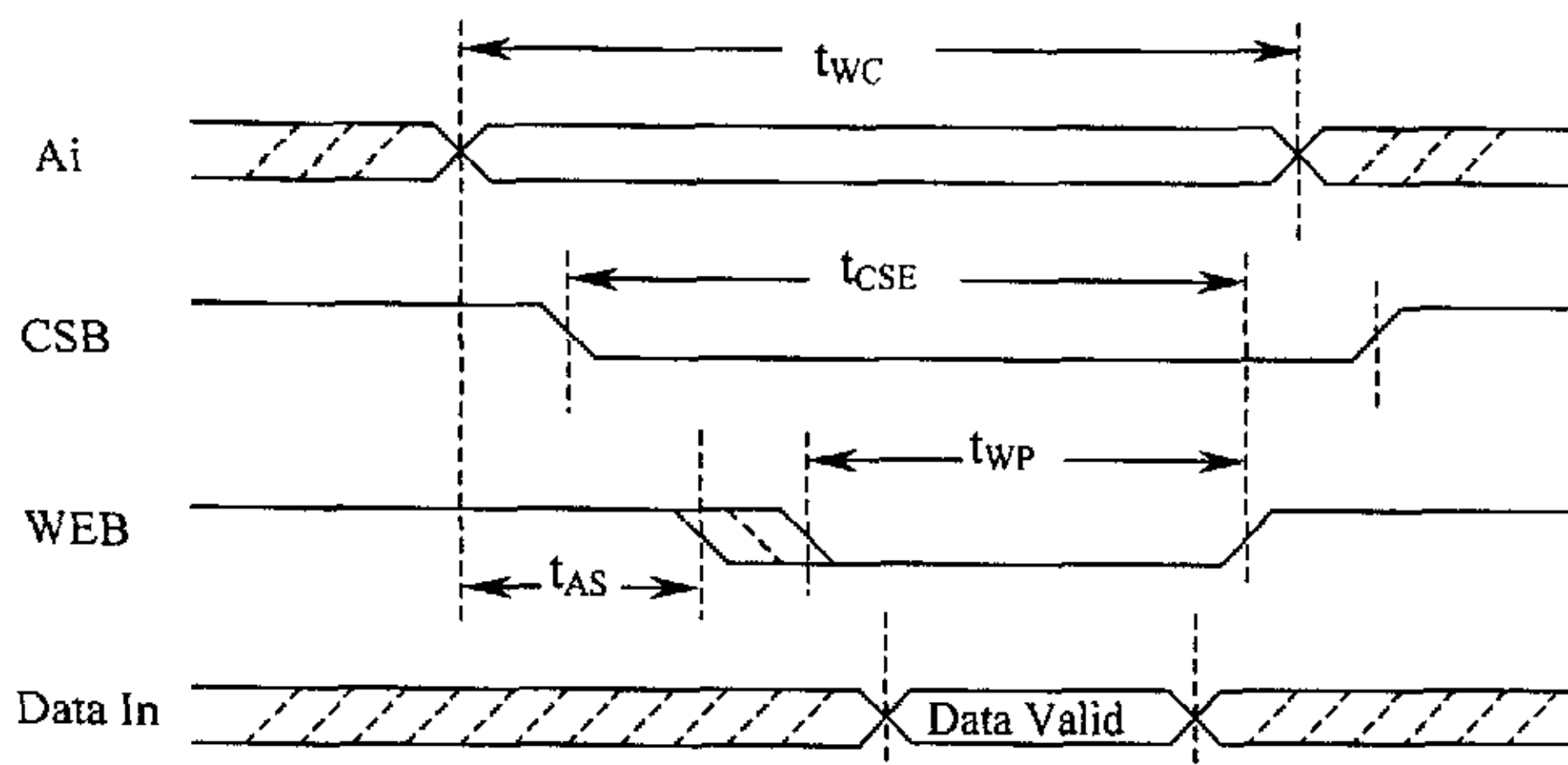


图 2-7 SRAM 写周期时序图

在写周期中，关键是写脉冲宽度  $t_{WP}$ ，即 WEB 为“0”的持续时间，它表示将一个数据写入单元所必需的时间，主要决定于写入电路对数据线、位线和单元的驱动能力，当然也和单元位置、写前位线及数据线的状况等因素有关。一般按最坏情况，取最长的写入时间作为  $t_{WP}$  的最小值给出。另外一个非常重要的参数就是写周期时间  $t_{WC}$ ，即两次写操作之间所需的最小时间间隔，是电路速度性能的标志。其它参数分别定义如下：

1.  $t_{CSE}$  是指从片选信号 CSB 建立到写使能信号 WEB 结束的时间间隔。

2.  $t_{AS}$  是指地址信号  $A_i$  建立到写使能信号  $WEB$  建立的时间间隔。

## 二. SRAM 的读过程

以读“1”为例来阐述 SRAM 的读过程，读“0”过程同理可得。读“1”时，首先要保证片选信号  $CSB$  在整个读过程中一直有效

( $CSB = "0"$ )，然后通过列译码器选中某位列线对  $BL$  和  $BLB$  进行预充电，一般预充到电源电压  $VDD$ ，主要是为了防止发生误操作，预充电后再通过行译码器选中某行，则某一存储单元被选定（存储单元结构如图 2-2 所示），此时，由于其中存放的是“1”，则  $WL = "1"$ ， $Q = "1"$ ，晶体管  $M1$  和  $M5$  导通，有电流经  $M1$  和  $M5$  到地，从而使  $BLB$  电位降低， $BL$  和  $BLB$  间产生电压差，当电压差增大到一定的程度（一般为  $0.5V$  左右）后打开灵敏放大器（ $OEB = "0"$ ），对  $BL$  和  $BLB$  间的电压差进行放大，最后再将灵敏放大器的输出接到三态输出驱动器上，驱动器的输出就是被读出的数据。到此为止，就完成了整个读过程。

SRAM 的读周期如图 2-8 所示。其中， $A_i$  为地址输入信号， $CSB$  和  $OEB$  分别为片选控制和读使能信号， $Data Out$  为读出的数据信号。

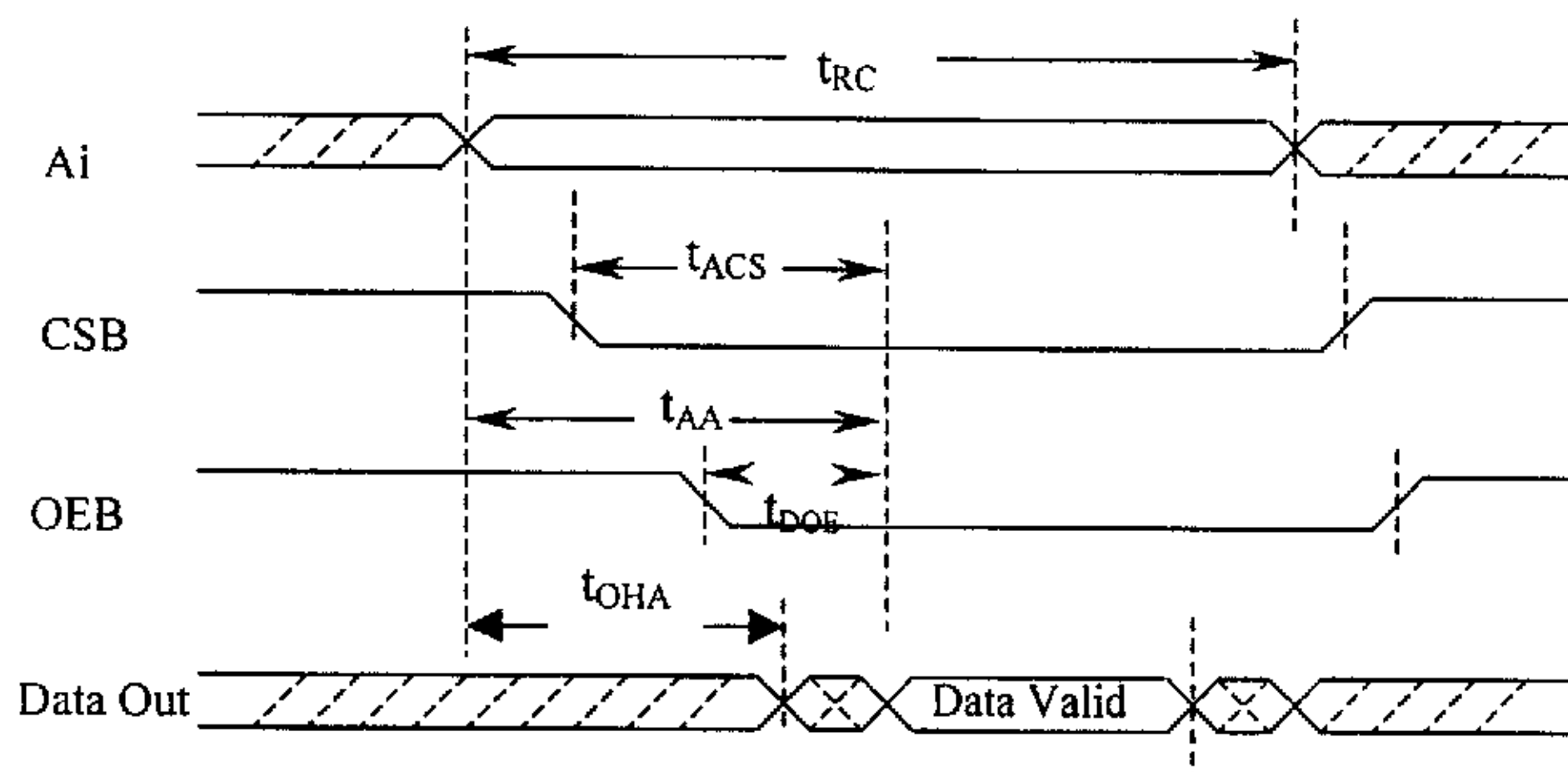


图 2-8 SRAM 读周期时序图

在图 2-8 中给出了一些描述 SRAM 读周期的时间参数，其定义分别如下：

1.  $t_{AA}$ ：取数时间，指的是从改变地址到相应单元的内容出现在 I/O 端所需的延迟时间。显然它是由译码速度、单元读出速度、读出放大与驱动电路的速度所决定，当然也与 I/O 端的负载情况有关。由于各个单元在芯片中所处

的位置不同，从不同单元取数实际所需的时间会有所区别。一般给出它的最大值。

2.  $t_{OHA}$ : 数据保持时间。在地址改变为另一个新地址后，I/O 端的输出信号将滞后变化。地址改变之后数据保持时间为  $t_{OHA}$ 。一般给出它的最小值。

3.  $t_{RC}$ : 读周期时间。定义为两次连续的读操作间所必须的最小时间间隔。

4.  $t_{ACS}$ : 指的是从 CSB 开始下降到有效数据开始出现在 I/O 端所需的时间。

5.  $t_{DOE}$ : 指的是从 OEB 开始到有效数据开始出现在 I/O 端所需的时间。

上述各时间参数中，最重要的是  $t_{AA}$ 、 $t_{OHA}$  和  $t_{RC}$ ，它们是决定 SRAM 工作速度的基本依据和电路速度性能的标志。

图 2-7 和 2-8 中所示的是描述 SRAM 瞬态特性的基本时间参数，在实际运用中，不同的产品或不同公司的同一类产品在产品规范中所给出的参数不一定都相同，有可能有多有少。

在阐述了 SRAM 的结构和工作原理的基础上，下一章将开始进入本篇论文的重点，论述用于 2.5V 全 CMOS 128Kb 和 1Mb SRAM 中的六管存储单元，两级 / 四级灵敏放大器和带时钟信号两级地址译码器，其具体内容将分别由第三章和第四章给出。

### 第三章 SRAM 存储单元设计

在密度 SRAM 中，存储单元阵列的面积占据了版图面积的绝大部分，而芯片的面积显然越小越好，因此存储单元的设计显得尤为重要。

#### 3.1 SRAM 存储单元结构形式及选择

图 3-1 中所示的是一种广泛应用于高密度 SRAM 中的由一对交叉耦合反相器组成的四晶体管电阻负载 SRAM 单元。这个基本锁存电路的两个稳态工作点被用来存储一位数据信息；因此，这对交叉耦合反相器构成了 SRAM 单元的关键组件。为了完成读和写操作，使用了两个开关管，这两个管子由字线/行选择信号 WL 控制。值得注意的是图中的单元通过两根位线/列，进行读写操作，而不是只用一根位线/列，这样互补列的安排可以保证操作更可靠。

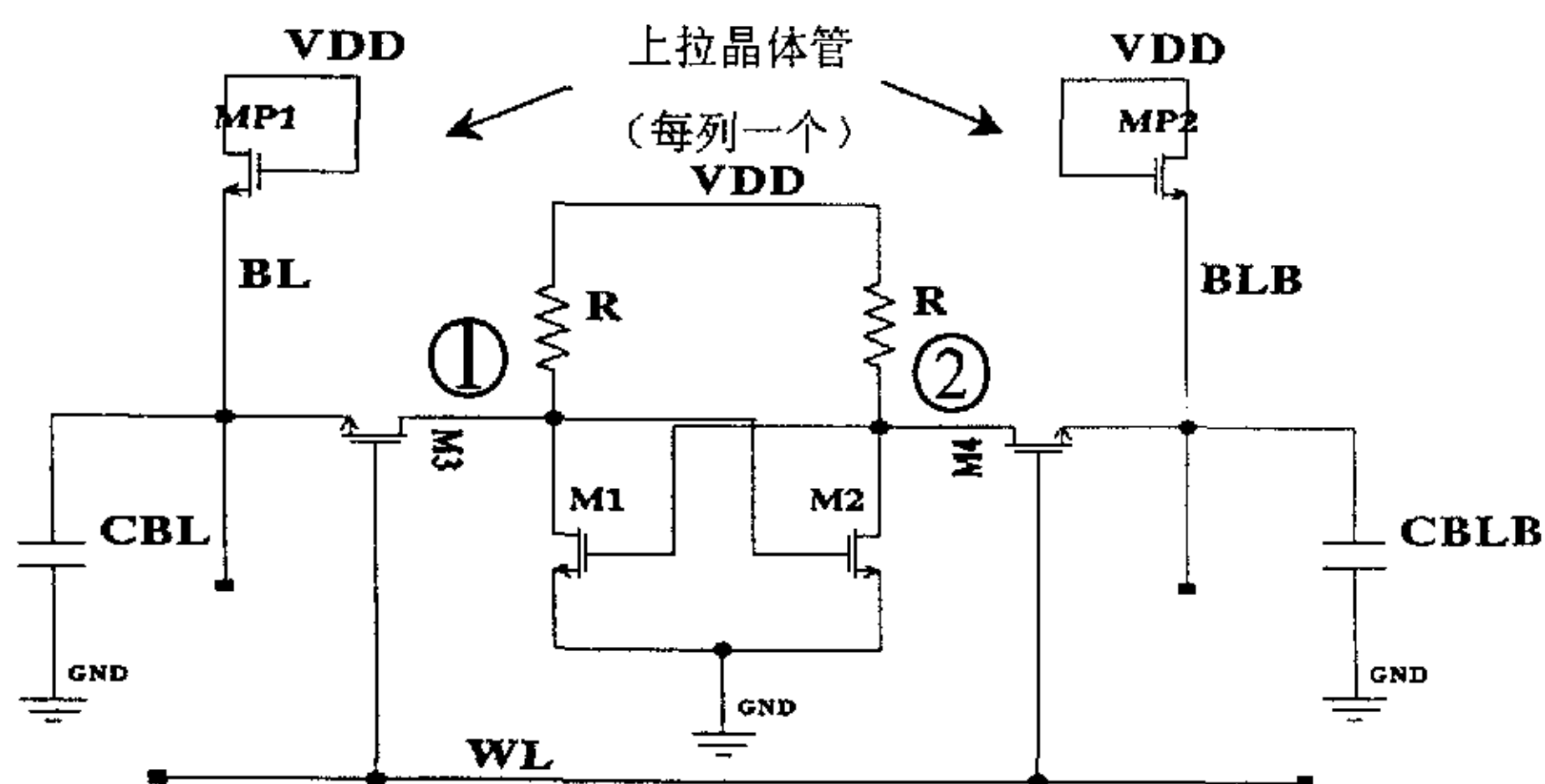


图 3-1 四晶体管电阻负载 SRAM 单元

当字线 WL 未被选中的时候，也就是说行选择信号 WL 电平为逻辑“0”，开关管 M3 和 M4 断开，这个由一对交叉耦合反相器组成的简单锁存电路保持其两个稳定工作点中的一个状态，因此数据被保持。此时，如果所有的字线未被选中，列上的上拉管 MP1 和 MP2 对位线上相对较大的电容充电。因为两个管子都处于饱和区，因此两列上的稳态电压由以下等式决定：

$$V_{BLB} = V_{BL} = V_{DD} - \left( V_{T0} + \gamma \left( \sqrt{|2\phi_F| + V_{BL}} - \sqrt{|2\phi_F|} \right) \right) \quad (3-1)$$

假设  $V_{DD}=5V$ ,  $V_{T0}=1V$ ,  $|2\phi_F|=0.6V$ ,  $\gamma=0.4V^{1/2}$ , 计算得  $V_{BLB}$  近似等于  $3.5V$ 。显然这两列互补位线电压在稳态时相等。

现在假设通过将它的字线电压抬高到逻辑“1”来选中存储单元, 因此开关管 M3 和 M4 导通。一旦存储单元被选中, 这个单元就可以完成四个基本操作。

a) 写“1”操作: 位线 BLB 上的电位被数据写入电路拉为低电平, 驱动管 M1 关断, 节点①保持逻辑高电平, 而节点②被驱动管 M2 拉为低电平;

b) 读“1”操作: 位线 BL 保持预充电电平, 而位线 BLB 被 M2 管和 M4 管拉低, 数据读取电路将这一很小的电压差 ( $V_{BL} > V_{BLB}$ ) 放大为逻辑“1”输出;

c) 写“0”操作: 位线 BL 上的电位被数据写入电路拉为低电平, 驱动管 M2 关断, 节点②保持逻辑高电平, 而节点①被驱动管 M1 拉为低电平;

d) 读“0”操作: 位线 BLB 保持预充电电平, 而位线 BL 被 M1 管和 M3 管拉低, 数据读取电路将这一很小的电压差 ( $V_{BL} < V_{BLB}$ ) 放大为逻辑“0”输出;

图 3-2 定性地示意出字线和两列互补位线的典型电压波形图。注意在读操作过程中两列位线上的电压差可能只有几百毫伏, 它可被数据读取电路放大为逻辑高电平输出。产生这一现象的原因是在数据读取时串联的两个 NMOS 管 (比如读“0”时的 M1 管和 M3 管) 不能将位线上的大电容快速地放电。

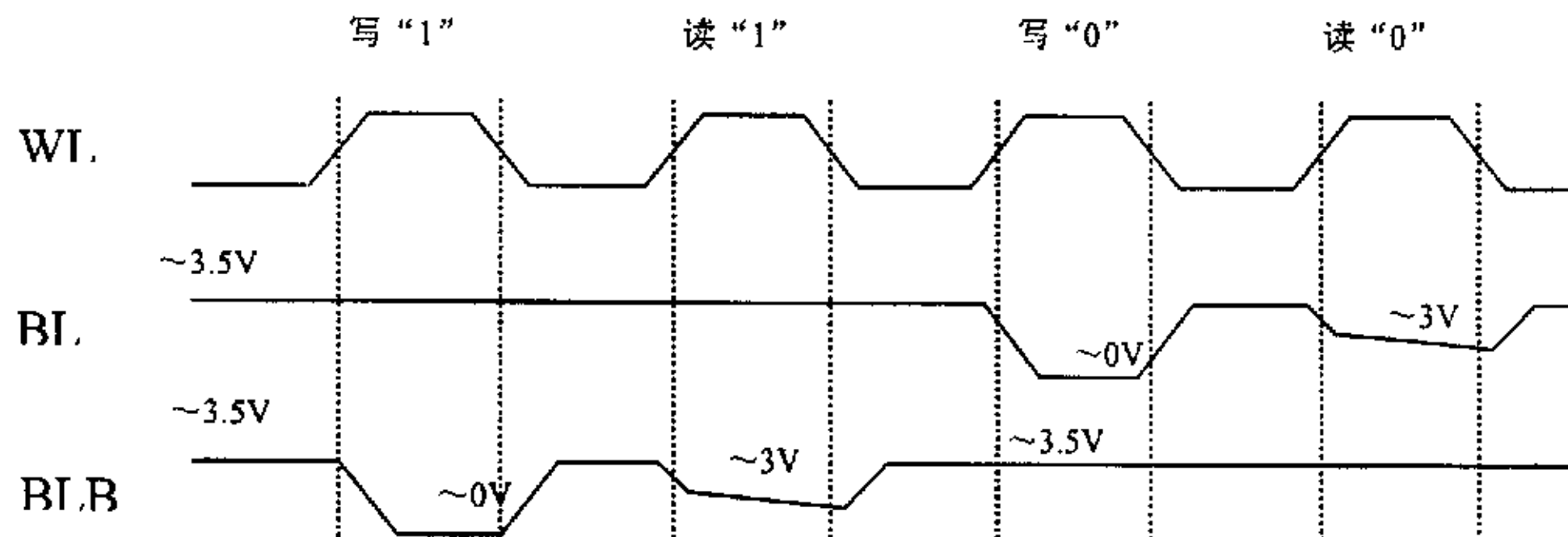


图 3-2 SRAM 读写典型电压波形图

这里我们关心的是在全 CMOS 六管单元(如图 3-3 所示)和电阻负载单元中, 作为存储单元比值  $\beta$  (即驱动管与开关管的沟道宽度之比) 的静态噪声容限 SNM 和电源电压  $V_{DD}$ 。图中示出了当位线和字线接到  $V_{DD}$  时的结果。图 3-4 表明, 对于单元比值为 2, 全 CMOS 六管单元的 SNM 比电阻负载单元的大了三倍左右, 因此能够做得更小, 从而降低了所需面积。图 3-5 表示了 SNM 与  $V_{DD}$  的关系。尽管具有更大的比值  $\beta$ , 电阻负载单元的 SNM 降得比 CMOS 六管单元的更快。SRAM 存储单元稳定性研究方法和结果在文献<sup>[17]</sup>中

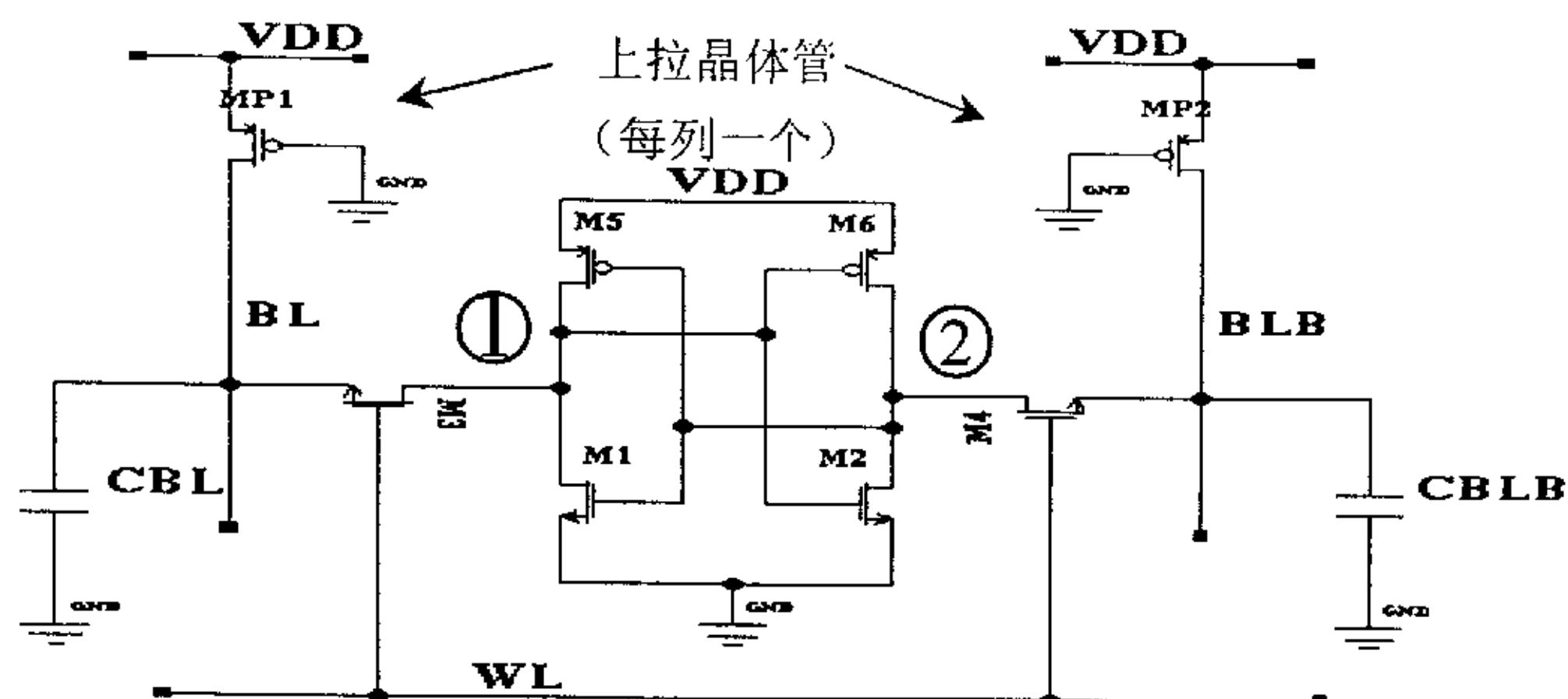


图 3-3 六管全 CMOS SRAM 存储单元

已有论述。由于具有很高的静态噪声容限和较小的面积，加上其低静态电流的特性，使得全 CMOS 六管单元成为低电压应用的最佳选择。所以我们选择全 CMOS 六管单元作为存储器的存储单元。

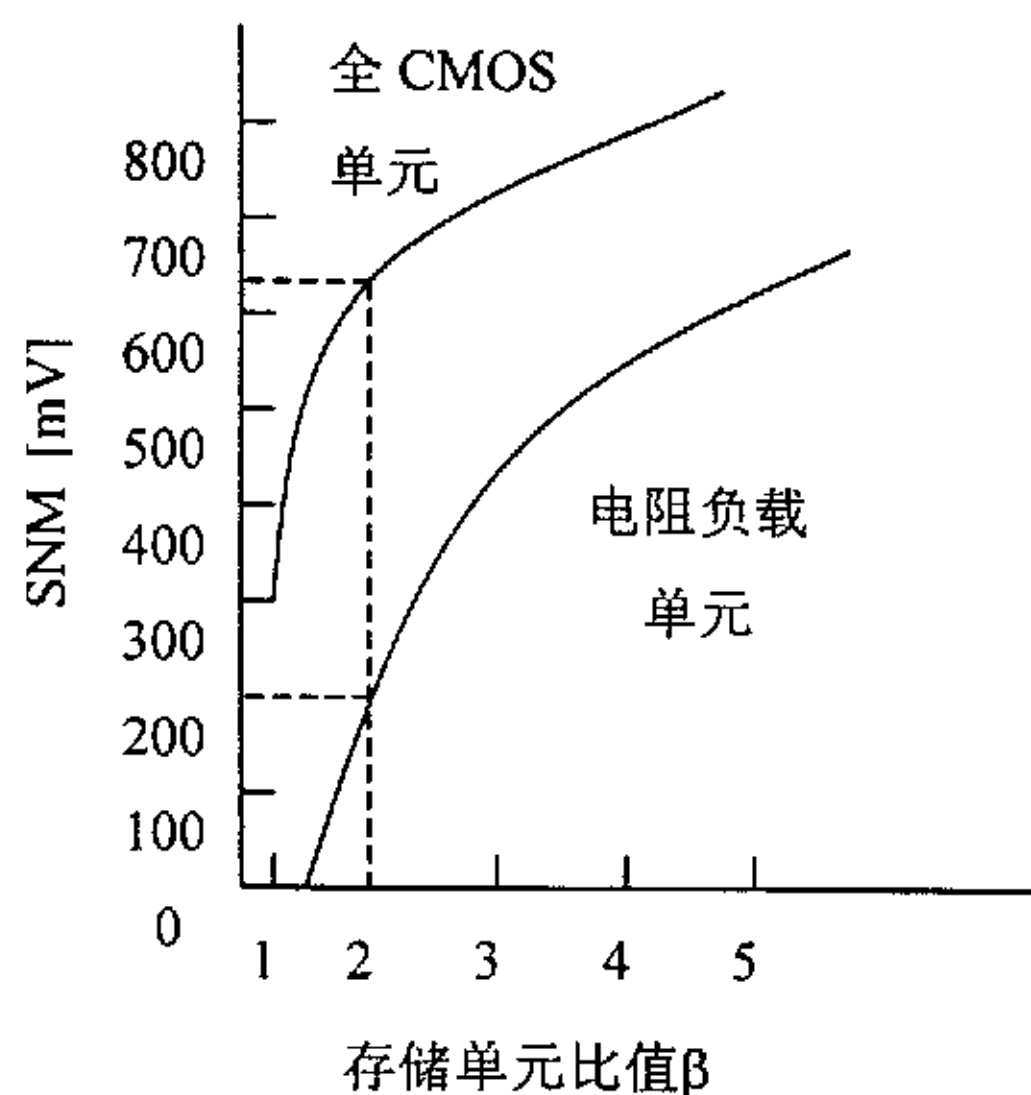


图 3-4 SNM 与  $\beta$  关系示意图

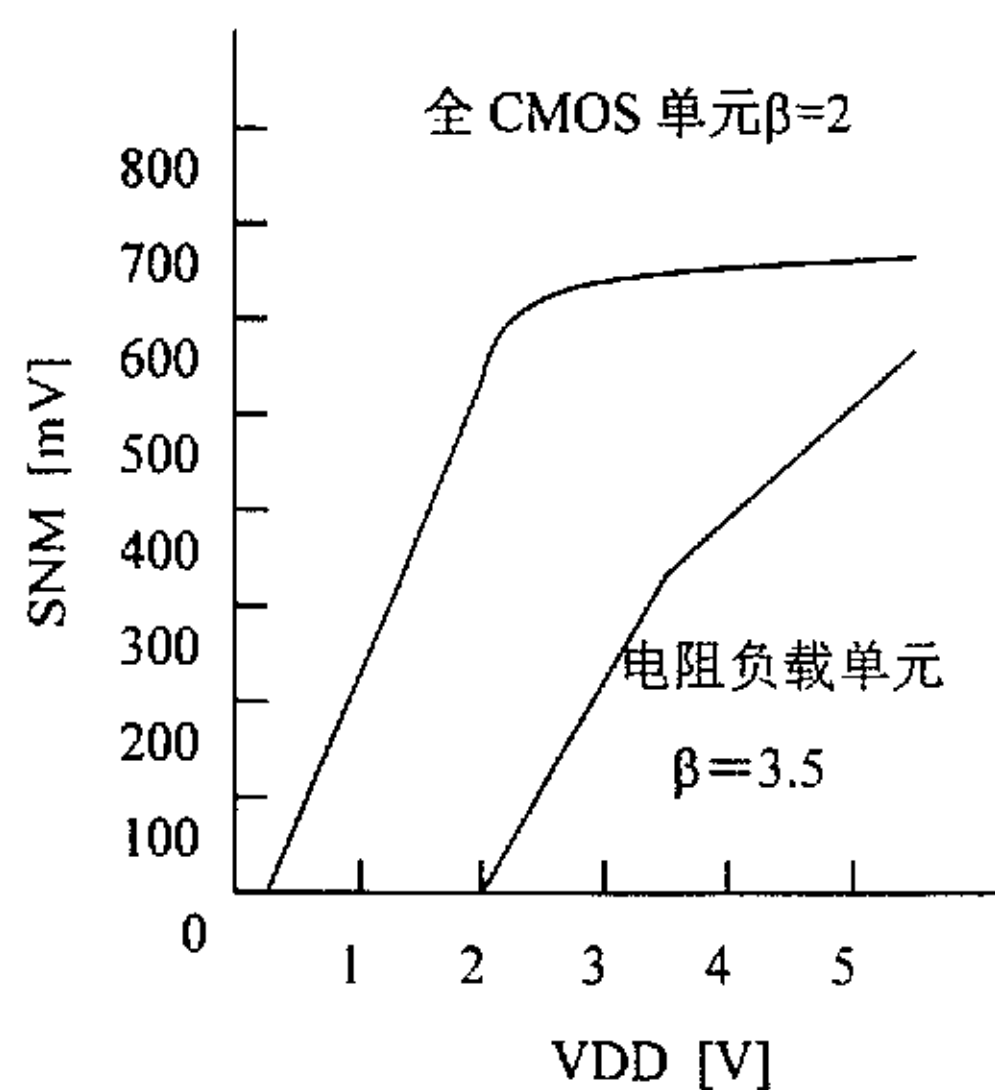


图 3-5 SNM 与  $V_{DD}$  关系示意图

### 3.2 SRAM 存储单元的设计

为了确定图 3-3 所示典型 SRAM 单元中各个管子的 W/L 的比值

(宽长比), 有一些基本原则需要考虑。其中决定  $W/L$  值的两个基本约束条件是: (1) 数据读取操作不应该破坏存储单元存储的信息; (2) 在数据写入过程中存储单元应该允许存储信息的改变。首先考虑数据读取操作, 假设此时单元中存储的是逻辑“0”, 图 3-6 表示的是读操作起始时 SRAM 单元各个节点电压情况。如图所示, 晶体管  $M2$  和  $M5$  关断, 而  $M1$  管和  $M6$  管工作于线性区。因此, 在单元中的开关管  $M3$  和  $M4$  导通以前, 内部节点①和②的电压分别为  $V_1=0$  和  $V_2=V_{DD}$ 。

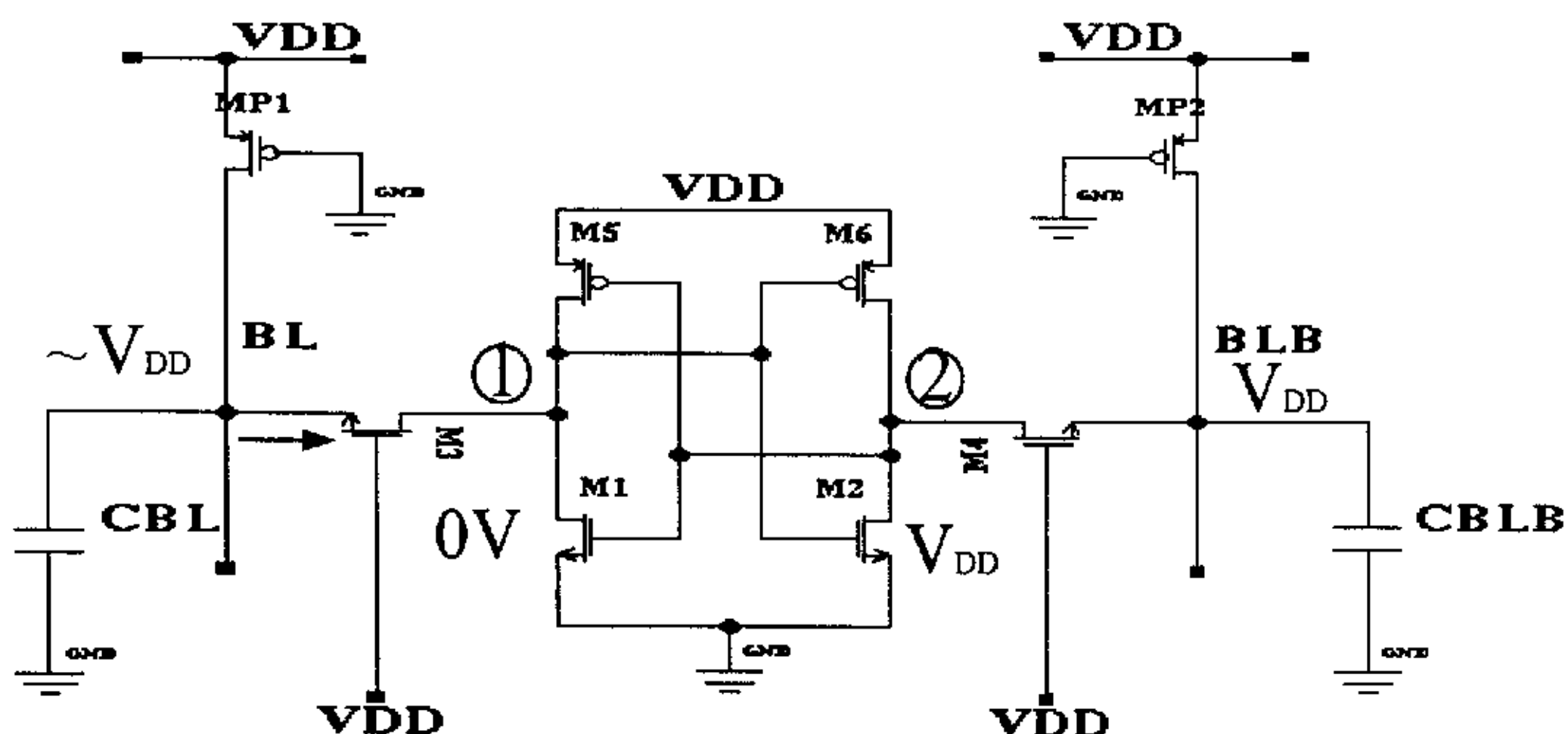


图 3-6 读操作起始时 SRAM 单元各个节点电压情况

在行选择电路选中开关管  $M3$  和  $M4$  以后, 位线  $BLB$  的电位没有明显的变化, 因为没有电流流过  $M4$  管。而在存储单元的另一边,  $M3$  管和  $M1$  管组成一条非零电流泄放通路, 因此位线  $BL$  电位开始缓慢下降。值得注意的是位线上的电容  $C_{BLB}$  通常很大, 因此在读过程中电压下降的数值只有几百毫伏。因此需要数据读取电路将位线上很小的压降放大为逻辑“0”输出。尽管  $M1$  管和  $M3$  管对位线上的电容  $C_{BLB}$  放电缓慢, 节点①的电位将从起始值  $0V$  开始升高。特别是如果开关管  $M3$  的比值与  $M1$  管相比大到一定程度的时候, 节点①的电位在这一过程中可能超过  $M2$  管的阈值电压, 出现不希望有的存储状态变化。因此在数据读取操作中设计的关键问题就是如何保证节点①的电位不超过  $M2$  管的阈值电压, 使得在读过程中  $M2$  管保持关断, 即有下列不等式:

$$V_{1,max} \leq V_{T,2} \quad (3-2)$$

式中  $V_{1,max}$  为节点 1 处的最大电压,  $V_{T,2}$  为  $M2$  管的阈值电压。

我们假设开关管导通以后, 位线  $BL$  电位保持在约等于  $V_{DD}$ 。因此  $M3$  管工作于饱和区, 而  $M1$  工作于线性区, 不难得出下列等式:

$$\frac{k_{n,3}}{2}(V_{DD} - V_1 - V_{T,n})^2 = \frac{k_{n,1}}{2}(2(V_{DD} - V_{T,n})V_1 - V_1^2) \quad (3-3)$$



式中  $k_{n,3}$ ,  $k_{n,1}$  分别是 M3 管和 M1 管的电导系数, 单位是  $\mu\text{A}/\text{V}^2$ 。  
将上式与 (3-2) 式联立, 得

$$\frac{k_{n,3}}{k_{n,1}} = \frac{\left(\frac{W}{L}\right)_3}{\left(\frac{W}{L}\right)_1} < \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} - 2V_{T,n})^2} \quad (3-4)$$

以上宽长比的上限实际上更保守, 因为 M3 管漏端电流的一部分也被用来对节点①的寄生电容充电。总之, 如果在读“0”操作过程中满足条件 (3-4), M2 管将保持在截止区。

根据对称关系, M2 管和 M4 管的宽长比满足同样不等式关系:

$$\frac{k_{n,4}}{k_{n,2}} = \frac{\left(\frac{W}{L}\right)_4}{\left(\frac{W}{L}\right)_2} < \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} - 2V_{T,n})^2} \quad (3-5)$$

现在考虑写“0”操作, 假设起始时 SRAM 单元中存储的是逻辑“1”。图 3-7 表示的是写操作起始时 SRAM 单元各个节点电压情况。晶体管 M1 和 M6 关断, 而 M2 管和 M5 管工作于线性区。因此, 在单元中的开关管 M3 和 M4 导通以前, 内部节点①和②的电压分别为  $V_1 = V_{DD}$  和  $V_2 = 0$ 。

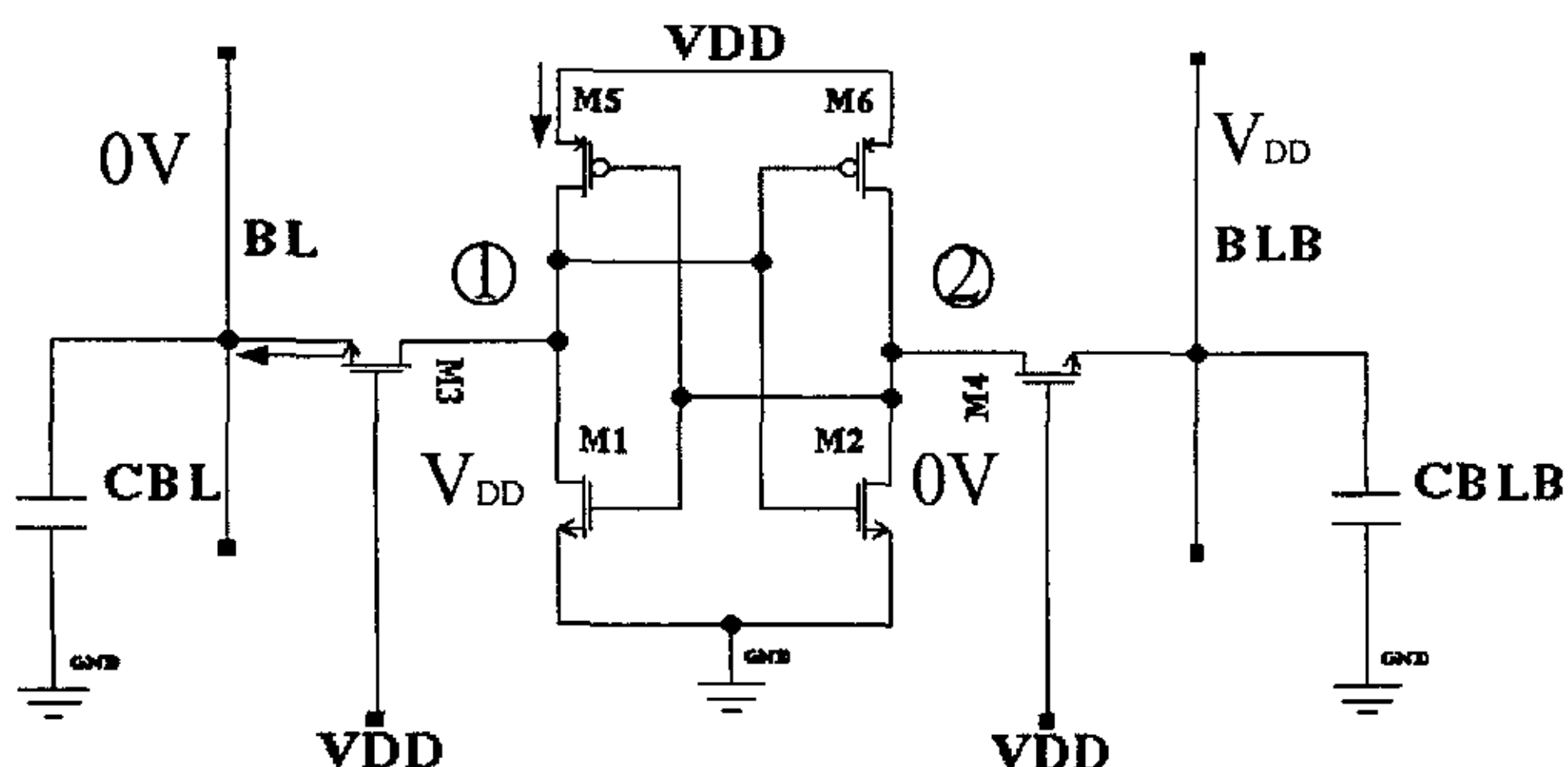


图 3-7 写操作起始时 SRAM 单元各个节点电压情况

位线 BL 上的电位  $V_{BL}$  被数据写入电路置为逻辑“0”, 因此我们可假设  $V_{BL}$  约等于 0V。一旦 M3 管和 M4 管被行选择电路选中, 即晶体管 M3 和 M4 导通, 节点②的电位保持在小于 M1 管的阈值电压, 因为晶体管 M2 和 M4 是根据条件 (3-5) 设计的。所以节点②的电位将不足以使 M1 导通。为了改变存储的信息, 也就是说使内部节点①和②的电压分别为  $V_1 = 0$  和  $V_2 = V_{DD}$ , 节点①的电位必须降到小于 M2 管的阈值电压, 目的是为了 M2 管先关断。当  $V_1 = V_{T,n}$  的时候, M3 管工作于线性区, 而 M5 管工作于饱和区。



不难得出下列等式：

$$\frac{k_{p,5}}{2}(0 - V_{DD} - V_{T,p}) = \frac{k_{n,3}}{2}(2(V_{DD} - V_{T,n})V_{T,n} - V_{T,n}^2) \quad (3-6)$$

有下列不等式：

$$\frac{k_{p,5}}{k_{n,3}} < \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} + V_{T,p})^2} \quad (3-7)$$

即

$$\frac{\left(\frac{W}{L}\right)_5}{\left(\frac{W}{L}\right)_3} < \frac{\mu_n}{\mu_p} \cdot \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} + V_{T,p})^2} \quad (3-8)$$

总之，如果在写“0”操作过程中满足条件（3-8），M2 管将保持在截止区。这就保证了 M1 管导通，改变存储的信息。

根据对称关系，M6 管和 M4 管的宽长比满足同样不等式关系：

$$\frac{\left(\frac{W}{L}\right)_6}{\left(\frac{W}{L}\right)_4} < \frac{\mu_n}{\mu_p} \cdot \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} + V_{T,p})^2} \quad (3-9)$$

因为设计中使用 TSMC（台积电）0.25 $\mu\text{m}$  工艺，为了估算的简便，我们取典型情况下的  $V_{T,n} = 0.411\text{V}$ ,  $V_{T,p} = -0.556\text{V}$ ,  $\mu_n = 0.0275\text{cm}^2/\text{V} \cdot \text{sec}$ ,  $\mu_p = 0.01080\text{cm}^2/\text{V} \cdot \text{sec}$ ,  $V_{DD} = 2.5\text{V}$  带入式（3-4）、（3-5）、（3-8）和（3-9）中，不难得出

$$\frac{k_{n,3}}{k_{n,1}} = \frac{\left(\frac{W}{L}\right)_3}{\left(\frac{W}{L}\right)_1} = \frac{\left(\frac{W}{L}\right)_4}{\left(\frac{W}{L}\right)_2} < \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} - 2V_{T,n})^2} \approx 0.547$$

$$\frac{\left(\frac{W}{L}\right)_5}{\left(\frac{W}{L}\right)_3} = \frac{\left(\frac{W}{L}\right)_6}{\left(\frac{W}{L}\right)_4} < \frac{\mu_n}{\mu_p} \cdot \frac{2(V_{DD} - 1.5V_{T,n})V_{T,n}}{(V_{DD} + V_{T,p})^2} \approx 1.032$$

以上两个不等式决定了存储单元各个管子之间的约束关系。器件的宽长比之间的关系确定以后，只要根据噪声容限和面积最小化要求以及标准工艺条件等，即可确定一个管子的（W/L）后，就可以得到其他管子的 W/L 值。具体数值是根据经验大致确定一个 W/L 值，通过模拟进行优化，最后确定每个管子的实际尺寸。

## 第四章 SRAM 外围电路设计与分析

一般说来，存储器中存储单元的结构和布局主要取决于当时所能达到的工艺水平，从某种意义上将是超出电路设计者的控制能力范围的，因此其中的外围电路对存储器的可靠性、性能和功耗等方面有着巨大的影响，也就成为电路设计者的设计重点。SRAM 也不例外，在 SRAM 中，存储单元大多采用标准的六管 CMOS 结构，为了改善器速度、噪声容限和逻辑摆幅等性能，就需要通过对外围电路的良好设计来完成。本章论述了 SRAM 的外围电路中的一个很重要的组成部分——灵敏放大器。

图 4-1 中所示的是典型 SRAM 的框图。SRAM 的取数时间由四级电路的延迟时间的和组成，即地址缓冲器，译码器，存储单元，灵敏放大器和输出缓冲器<sup>[18]</sup>。在这些电路中，当存储器密度增加的时候，译码器和灵敏放大器电路的延迟会相应增加很大。这是因为译码电路必须驱动很大的存储阵列，同时灵敏放大电路必须能对很长的位线上产生的的一对小信号进行放大。因此在高密度 SRAM 中取数时间主要由译码电路和灵敏放大器电路决定。

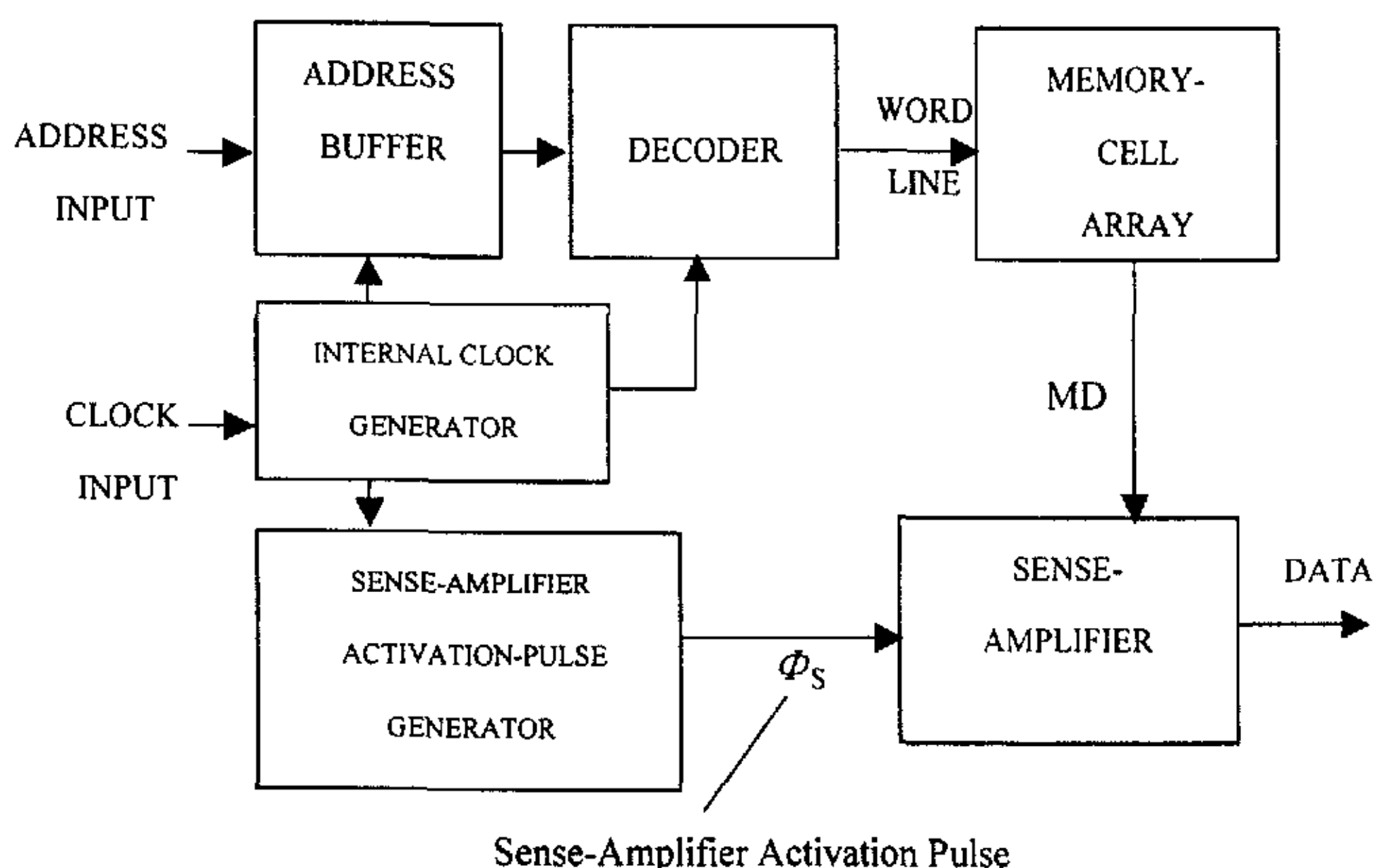


图 4-1 典型 SRAM 的框图

## 4.1 灵敏放大器设计考虑

通常, SRAM 存储单元的两条位线上的电压变化幅度是低于电源电压的, 这样可有效地减小由于对位线充、放电而造成的传输延迟和功耗。在存储单元阵列内部通过对串扰和其它干扰因素的控制, 可保证即使对于如此小的信号摆幅, 仍能得到足够大的噪声容限。但当与外部电路相连接的时候, 就需要将两条位线上的信号分别放大成逻辑“1”和逻辑“0”, 这样才能保证外部电路能正确识别从 SRAM 的存储单元所取出的数据, 从而得到正确的输出结果。在 SRAM 中, 这项工作是由灵敏放大器来完成的。灵敏放大器的性能好坏主要会影响 SRAM 的速度和功耗, 因此对其结构要进行不断地优化以改善整个 SRAM 的性能。

### 4.1.1 灵敏放大器结构分析

灵敏放大器一般为双端输入、双端输出, 如图 4-2 所示<sup>[19]</sup>。其中  $V_{i1}$ 、 $V_{i2}$  为输入信号,  $V_{o1}$ 、 $V_{o2}$  为输出信号, 它们是对地极性相同的电平。输入信号来自两条位线, 其高低电平相差很小, 而且都是绝对值较高的电压, 如图 4-3 中的  $V_{i1}$ 、 $V_{i2}$  所示, 它的输出信号则应是高低电平合格的逻辑信号, 如图 4-3 中的  $V_{o1}$ 、 $V_{o2}$  所示。对灵敏放大器的要求是灵敏度高, 即只要输入很小的电平差, 就能使输出满足要求; 另一个要求是延迟时间小, 即当输入信号改变极性时, 希望输出信号能迅速响应, 这样就可以大大地缩短取数时间。

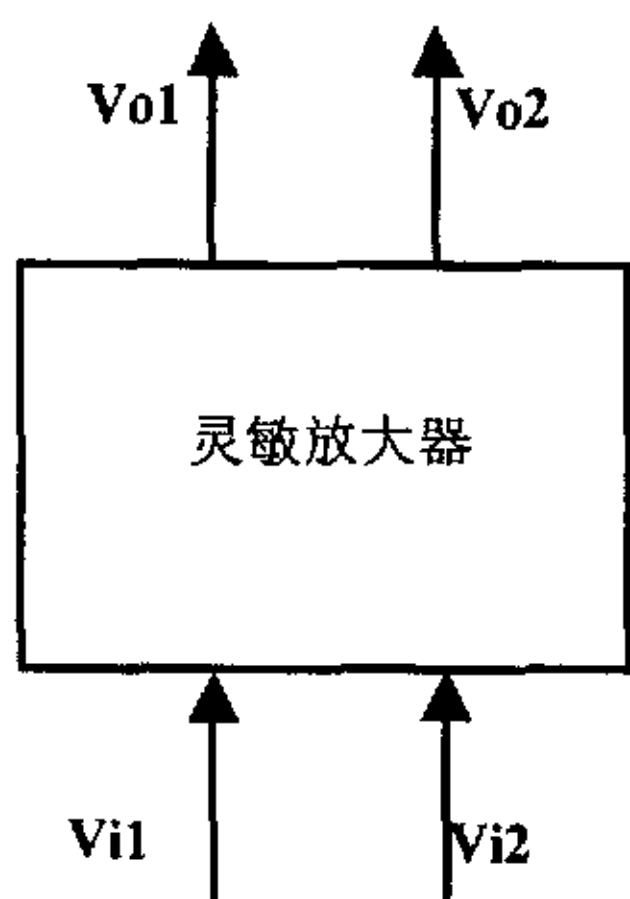


图 4-2 灵敏放大器功能框图

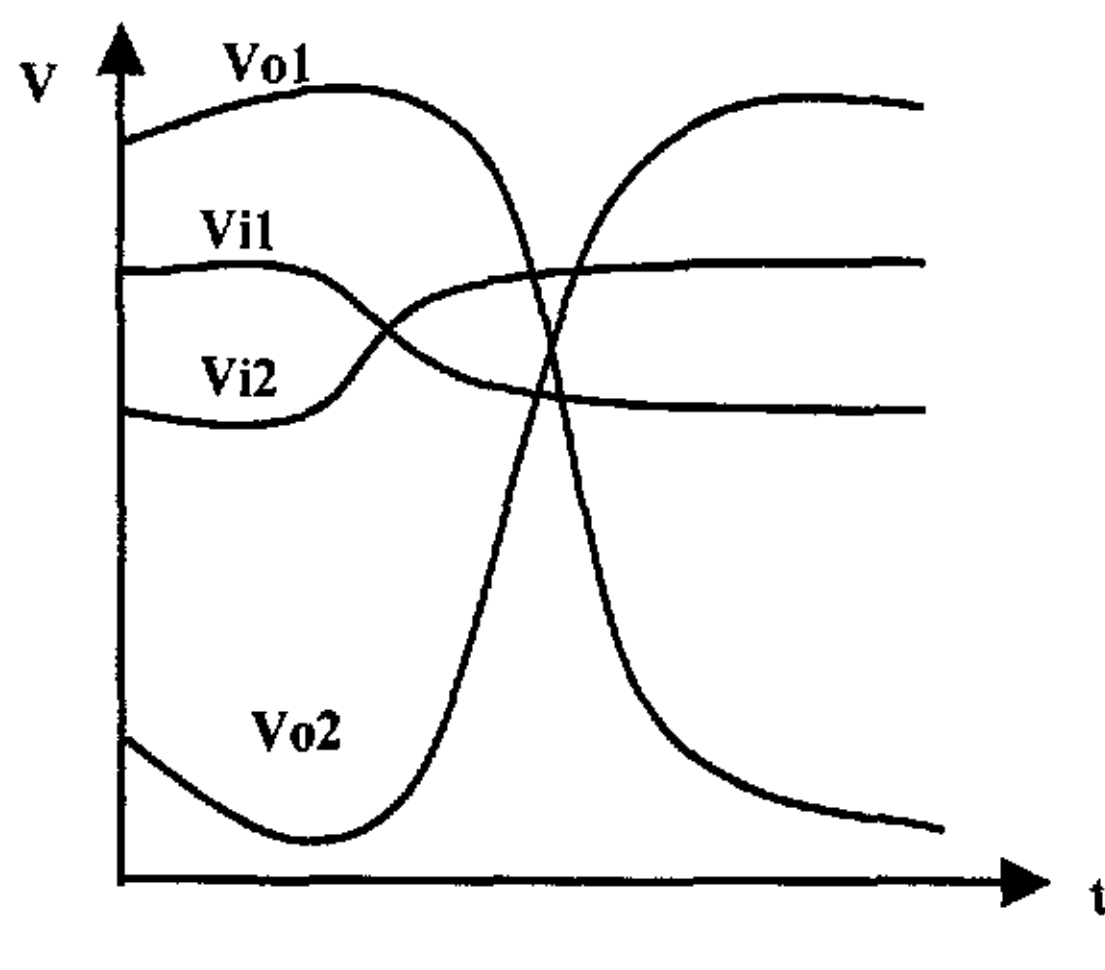


图 4-3 灵敏放大器输入输出电压变化示意图

现在常用于 SRAM 的灵敏放大器大多是差分结构的，因为差分结构能够提供很好的共模抑制比（CMRR）和电源抑制比（PSRR），而且由 SRAM 的存储单元所提供的一对位线 BL 和 BLB 两个信号正好可以作为差分放大器的差分输入端。CMRR 的定义为  $20\lg(A_{vd}/A_{vc})\text{dB}$ ，其中  $A_{vd}$  和  $A_{vc}$  分别为放大器在开环运用下的差分放大倍数和共模增益，它是用来衡量放大器抑制噪声的能力。PSRR 的定义为：假如电源电压中包括一个由于噪声和其它一些干扰信号所引起的增量  $v$ ，则在放大器的输出端就会有相应的电压  $A_p \cdot v$  出现， $\text{PSRR} = 20\lg(A_{vd}/A_p)$ 。由以上定义可看出 CMRR 和 PSRR 越大，放大器的稳定性越好，抗干扰能力越强<sup>[20]</sup>。

下面将论述运算放大器型灵敏放大器和交叉耦合灵敏放大器的结构及特点。这两种具有差分放大的结构的灵敏放大器由于各自独特的特点，现在被广泛应用于各种半导体存储器中。

## 一. 运算放大器型灵敏放大器

### （一） 运算放大器型灵敏放大器的结构

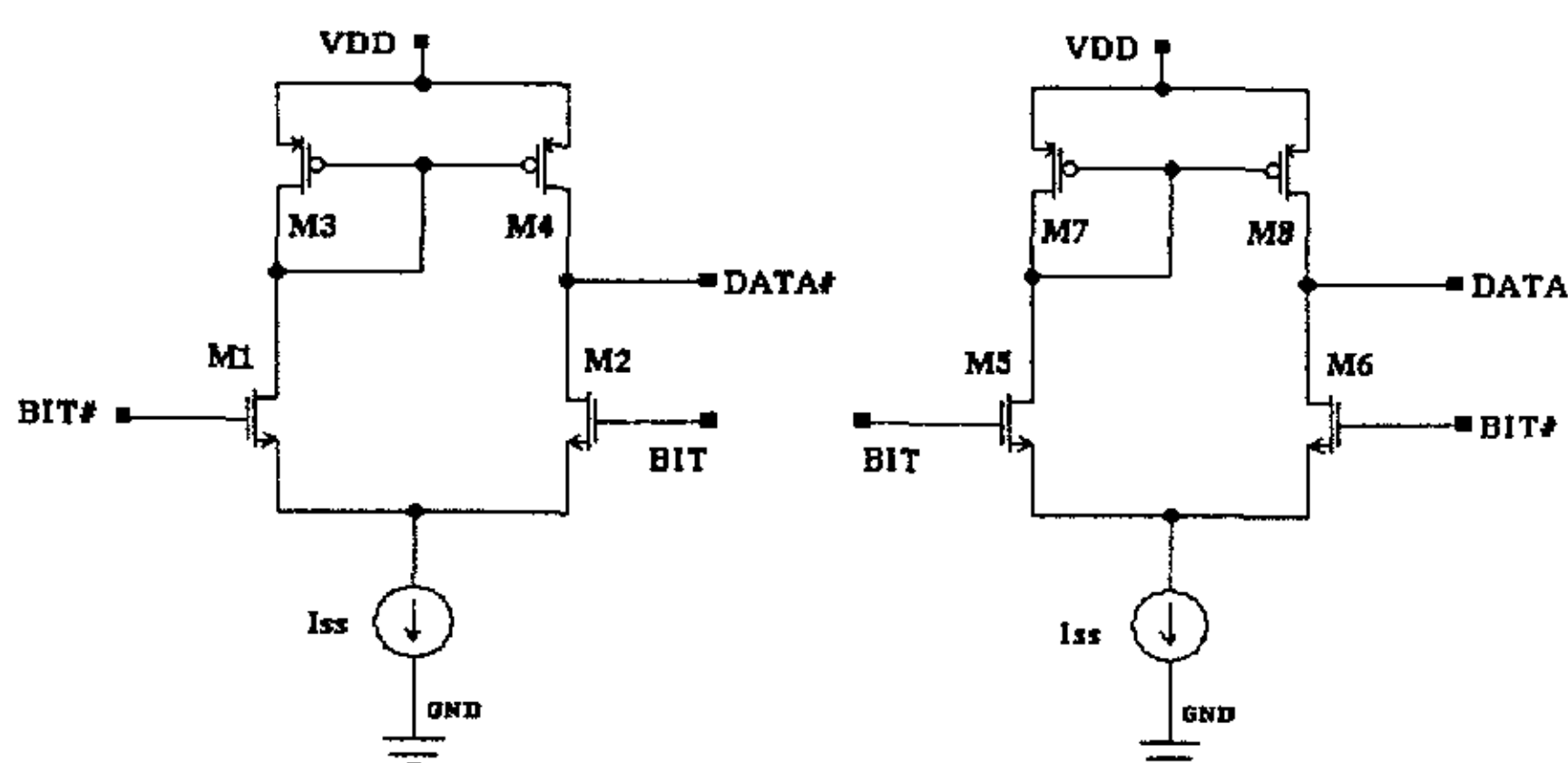


图 4-4 运算放大器型灵敏放大器结构示意图

图 4-4 中给出了一个典型的运放型灵敏放大器的结构示意图，其中 BIT 和 BIT# 为放大器的两个差分输入端，对应于存储单元的两条位线 BL 和 BLB 上的信号，DATA 和 DATA# 为放大器的两个输出端，分别对应于输出数据总线上的一对互补数据信号。以下关于 BIT 和 BIT#、DATA 和 DATA# 的定义，如不加特殊说明，皆与此处相同。在图 4-4 中，左半部分和右半部分的结构完全相同，只是输入端的接法相反，这样做是因为 SRAM 中的灵敏放大器一般是双端输入、双端输出的，而运放是双端输入、单端输出的，所以为了得到两个相反的输出端，就用了双倍的如图 4-4 的左半部分所示的结构。图中，晶体

管 M1、M2、M3、M4 和  $I_{SS}$  所组成的是典型的运放结构，M1、M2 为差分输入管，M3、M4 构成的电流镜为有源负载， $I_{SS}$  为电流源，为放大支路提供稳定的工作电流，在放大电路的前一级是偏置电路，为整个电路提供稳定偏置电压，从而建立起稳定的稳态工作点，因为此种结构属于模拟电路的范围，所以建立稳定的静态工作点是保证整个电路正常工作的必要前提。在放大电路的后一级应是驱动电路，用来增强电路的带负载能力。

## (二) 工作原理分析

这是一个源极耦合的结构，为了在不增加面积或功耗的基础上增大其放大能力，图中由电流镜充当的有源负载代替了传统的电阻负载[21]。电流镜的原理是：如果两个相同的 MOS 晶体管的栅源电压和漏源电压相等，沟道电流就应该相等。所以流过晶体管 M1 的电流就被 M3、M4 组成的电流镜反映到 M2，并与流过 M2 管的电流进行比较，流过 M1 和 M2 的电流差按式

(4-1) 被转换成很高的电压增益，从而实现放大功能，其放大倍数  $A$  由下式表示：

$$A_v = 2g_{m2} / [I_{SS}(\lambda_2 + \lambda_4)] = 2[(K_2'W_2)/(I_{SS}L_2)]^{1/2} / (\lambda_2 + \lambda_4) \quad (4-1)$$

式 (4-1) 中， $A_v$  为放大倍数， $g_{m2}$  为输入晶体管 M2 的跨导，定义为  $(\partial i_{D3} / \partial v_{GS2})|_{\text{静态工作点}}$ ， $\lambda_2$  和  $\lambda_4$  分别为晶体管 M2、M4 的沟道长度调制系数，单位为  $V^{-1}$ ， $I_{SS}$  为流过电流源的静态电流，它被平均分配到晶体管 M1、M2 所在的支路。 $K_2'$  为晶体管 M2 的电导系数，单位为  $\mu A / V^2$ ， $W_2 / L_2$  为 M2 管的宽长比。

式 (4-1) 的推导过程如下：以图 4-4 所示电路的左半部分为例，为了推导它的放大倍数，需先给出其小信号等效图，如图 4-5(a)所示。因为电流源在理想状况下，即当放大器两边的器件完全匹配时，其交流变化量为零，此时晶体管 M1、M2、M3、M4 的源极 S1、S2、S3、S4 就可都认为是地电位。图 4-5(b) 给出的就是图 4-4 所示的电路在理想情况下的简化的小信号等效图，因为现主要是求其放大倍数的绝对值，所以在图中未考虑电容因素。对图中两支路分别应用欧姆定律可得式 (4-2) 和 (4-3)：

$$v(N1) = -g_{m1}v(\text{BIT\#}) / (g_{ds1} + g_{ds3} + g_{m3}) \quad (4-2)$$

$$v(\text{DATA\#}) = -[g_{m2}v(\text{BIT}) + g_{m4}v(N1)] / (g_{ds2} + g_{ds4}) \quad (4-3)$$

其中  $v(N1)$  是节点 1(即节点 D1、G3、D3、G4)的电位，

将式 (4-2) 中的  $v(N1)$  表达式代入式 (4-3) 中可得

$$v(\text{DATA\#}) = -\{g_{m2}v(\text{BIT}) - g_{m4}[g_{m1}v(\text{BIT\#}) / (g_{ds1} + g_{ds3} + g_{m3})]\} / (g_{ds2} + g_{ds4}) \quad (4-4)$$

因  $g_{m3} \gg g_{ds1}$ ， $g_{m3} \gg g_{ds2}$ ，且在理想状况下， $g_{m1} = g_{m2}$ ，

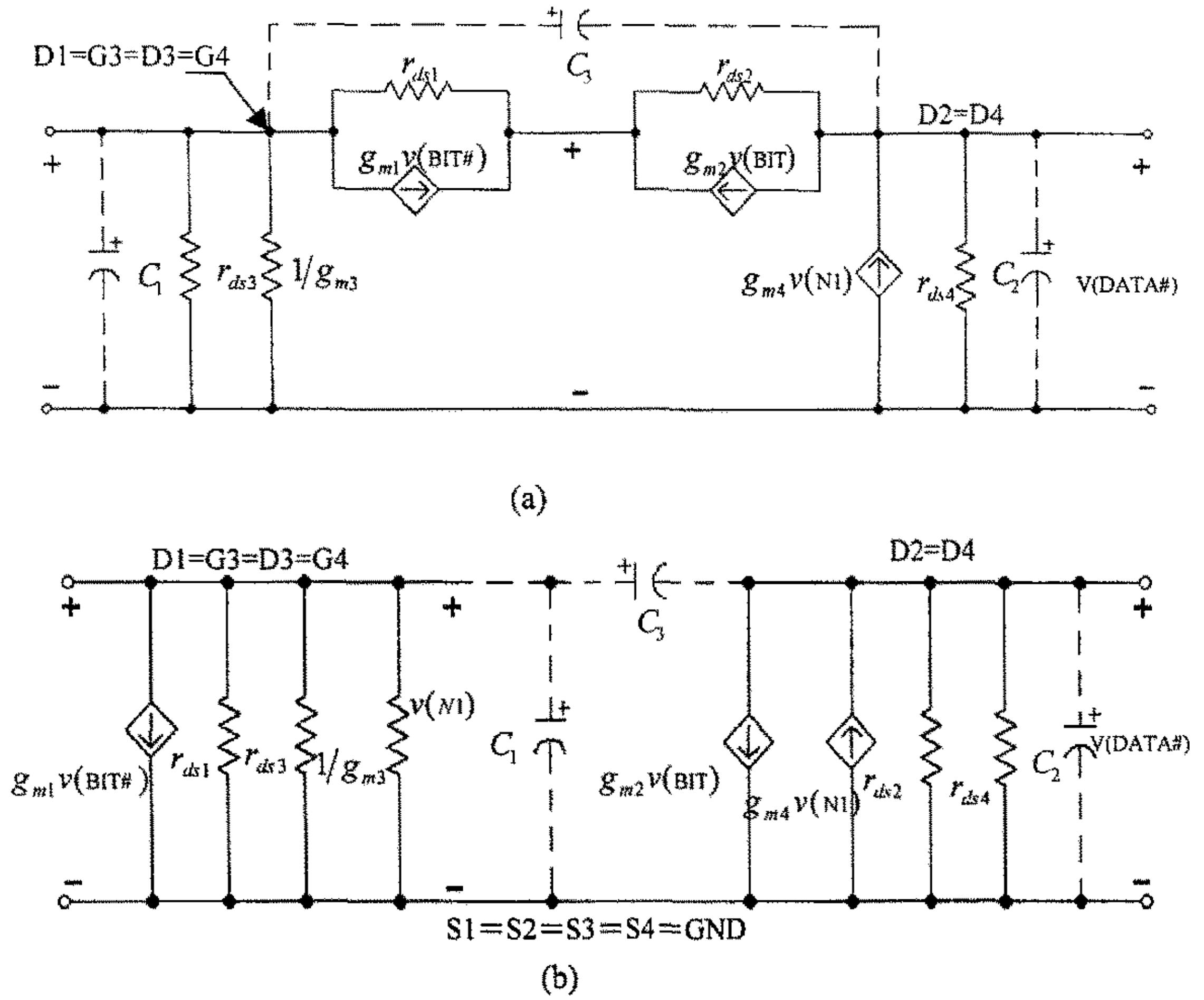


图 4-5 CMOS 差分放大器的小信号模型

(a) 精确的模型

(b) 简化的等效模型

$g_{m3} = g_{m4}$ ，所以式 (4-4) 可简化为

$$\begin{aligned} v(\text{DATA\#}) &= -[g_{m2}v(\text{BIT}) - g_{m1}v(\text{BIT\#})]/(g_{ds2} + g_{ds4}) \\ &= g_{m2}[v(\text{BIT\#}) - v(\text{BIT})]/(g_{ds2} + g_{ds4}) \end{aligned} \quad (4-5)$$

由式 (4-5) 可得

$$A_v = v(\text{DATA\#})/[v(\text{BIT\#}) - v(\text{BIT})] = g_{m2}/(g_{ds2} + g_{ds4}) \quad (4-6)$$

当工作于饱和区时，晶体管的漏端电流  $i_D$  与  $v_{GS}$  和  $v_{DS}$  的关系式为

$$i_D = (K'W)/(2L)(v_{GS} - V_T)^2(1 + \lambda v_{DS}) \quad (4-7)$$

则  $g_{ds2} = (\partial i_{D2} / \partial v_{DS2})|_{\text{静态工作点}}$

$$= \lambda_2 (K'_2 W_2)/(2L_2)(v_{GS2} - V_{T,2})^2|_{\text{静态工作点}} = \lambda_2 (I_{SS}/2) \quad (4-8)$$

同理可得， $g_{ds4} = \lambda_4 (I_{SS}/2)$  (4-9)

再由  $g_{m2}$  的定义可得，

$$g_{m2} = (\partial i_{D2} / \partial v_{GS2})|_{\text{静态工作点}}$$



$$= K_2' / (W_2 / L_2) (v_{GS2} - V_{T,2}) |_{\text{静态工作点}} = [K_2' (W_2 / L_2) / I_{SS}]^{1/2} \quad (4-10)$$

由式 (4-10) 可看出, 输入晶体管跨导的增大可以通过增大其宽长比的方法实现。

将式 (4-8)、(4-9) 和 (4-10) 带入式 (4-6) 中可得

$$A_v = 2g_{m2} / [I_{SS}(\lambda_2 + \lambda_4)] = 2[(K_2' W_2) / (I_{SS} L_2)]^{1/2} / (\lambda_2 + \lambda_4) \quad (4-11)$$

### (三) 特点

1. 图 4-4 所示的 CMOS 差分放大器采用的是 NMOS 管作为输入管, PMOS 管作为负载管, 但也常会采用 PMOS 管作为输入管而 NMOS 管作为负载管的结构, 此时的偏置接法会相应的改变。同时两种结构的小信号增益也会由于器件迁移率的不同而造成一定的差别。一般情况下, 以 NMOS 管作为输入管的结构比以 PMOS 管作为输入管的结构增益要大, 图 4-4 的结构的开环增益很高, 灵敏度高, 即对于非常小的电压差都能放大, 得到符合逻辑的“高”“低”电平;

2. 图 4-4 所示结构的共模增益在理想情况下为零, 有很高的 CMRR 和 PSRR。这是因为电流镜负载将这些共模信号全反射掉了。当差分放大器失配时, 可能出现共模响应。这些失配指电流镜的电流增益不为 1 以及 M1 和 M2 的几何尺寸不匹配;

3. 运放型灵敏放大器由于差分放大器的双端输入、单端输出的结构特点, 就需要占用较大的面积, 通过采用两个完全相同的差分放大器结构来满足 SRAM 中灵敏放大器的双端输入、双端输出的要求;

4. 采用图 4-4 所示结构的灵敏放大器的速度比较慢, 存取时间较长, 具体数值将在下面的仿真结果中给出。

## 二. 交叉耦合型灵敏放大器<sup>[22]</sup>

交叉耦合型灵敏放大器的结构示于图 4-6, 此种结构速度较快的原因就是存在一个由 MP1、MP2 所构成的正反馈, 而正反馈又有利于提高速度。假设节点 DATA 电位上升, 则 MP1 管的  $V_{SG}$  减小, 由式 4-7 可得  $i_D$  也相应减小, 从而使其电阻变大,  $V_{SD1}$  增大, DATA# 点电位下降, 反馈再使 MP2 管电阻变小,  $V_{SD2}$  减小, DATA 电位进一步上升, 于是形成正反馈, 直至稳定。SAE 信号用来控制开关管 MN3 的导通, 从而控制整个灵敏放大器在适当的时候才打开, 这样就可有效地减小功耗。而且, 此种结构本身就是双端输入、双端输出的, 所以占用面积较小。但同时由于其中正反馈结构的影响, 使得它容易由于器件尺寸的不匹配而被锁存在一个错误的状态, 不能正常工作, 所以器件尺寸不易设计。

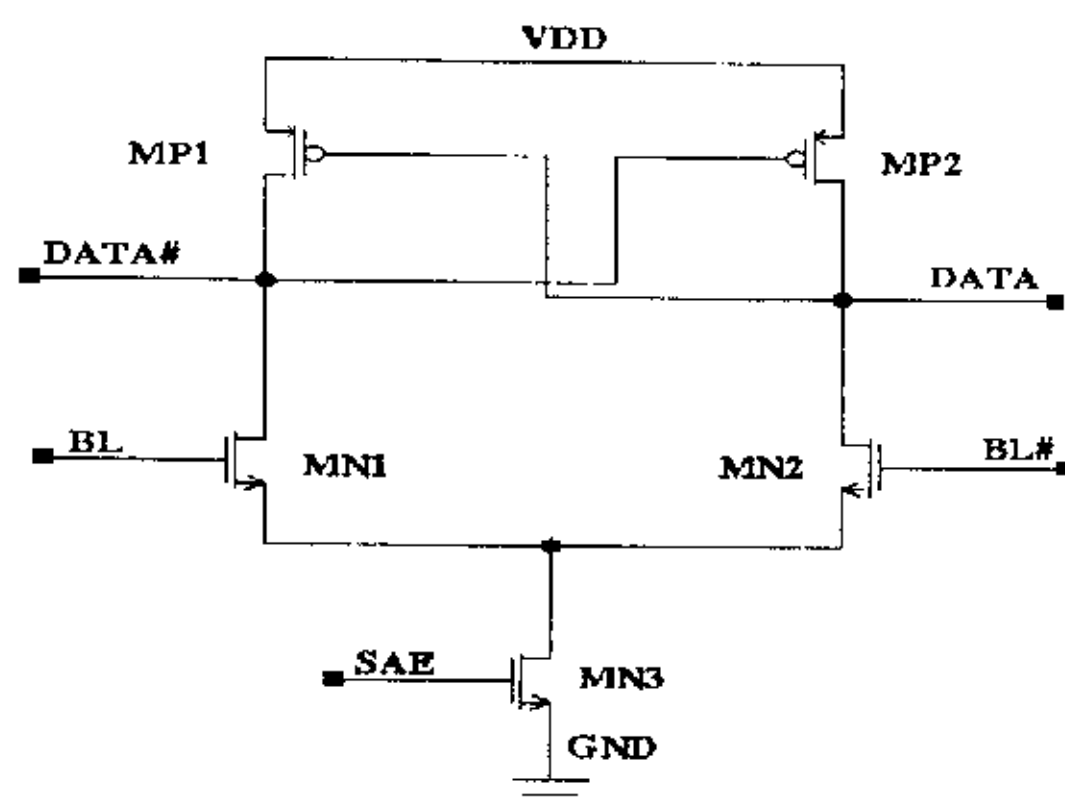


图 4-6 交叉耦合型灵敏放大器示意图

从图 4-6 中可以看出，当 SAE 为逻辑“高”时，MN3 管导通，整个电路开始工作，输出为有效状态。此时若 BL 和 BL# 分别为“高”和“低”，则 MN1 管导通，节点 DATA# 电位下降，MP2 管导通，DATA 点电位升高，再通过 MP1、MP2 管所构成的正反馈，使得 DATA 点的电位越来越高，DATA# 点电位越来越低，最终分别达到逻辑“1”和“0”，从而实现灵敏放大。

#### 4.1.2 两级/四级灵敏放大器的设计

在本课题中，需要设计 128Kb 和 1Mb 的 SRAM，所以先进行了两种用于不同密度下的灵敏放大器的设计，设计中采用了两级和四级灵敏放大器的结构。

##### 一、用于 128Kb SRAM 中的两级灵敏放大器

在 128Kb SRAM 中，由于一对位线上电压的特点，其中一个电平为 VDD，另一个比 VDD 仅仅小 100mv 左右，为了使得灵敏放大器能够对一对位线上很小的电压差进行放大，我们首先使用交叉耦合灵敏放大器对电平位移后的信号进行放大，再使用两个输入端接法相反而结构完全相同的差分放大器结构对放大后的信号进一步放大，得到符合要求的高电平。

下面简单说明两级灵敏放大器的工作原理。在如图 4-7 所示的带有源极跟随器的灵敏放大器中，第一级放大器由与 pMOS 管耦合的 nMOS 管源极跟随器和一对交叉耦合的 nMOS 管组成。当 SAE = “0” 时，pMOS 管 MP1 和 MP2 对 MN1 管和 MN2 管进行预充电，此时灵敏放大器不工作。XBT 和 XBB 分别是位线真值和位线真值互补信号，假设 XBT 电位为 VDD，XBB 电位比



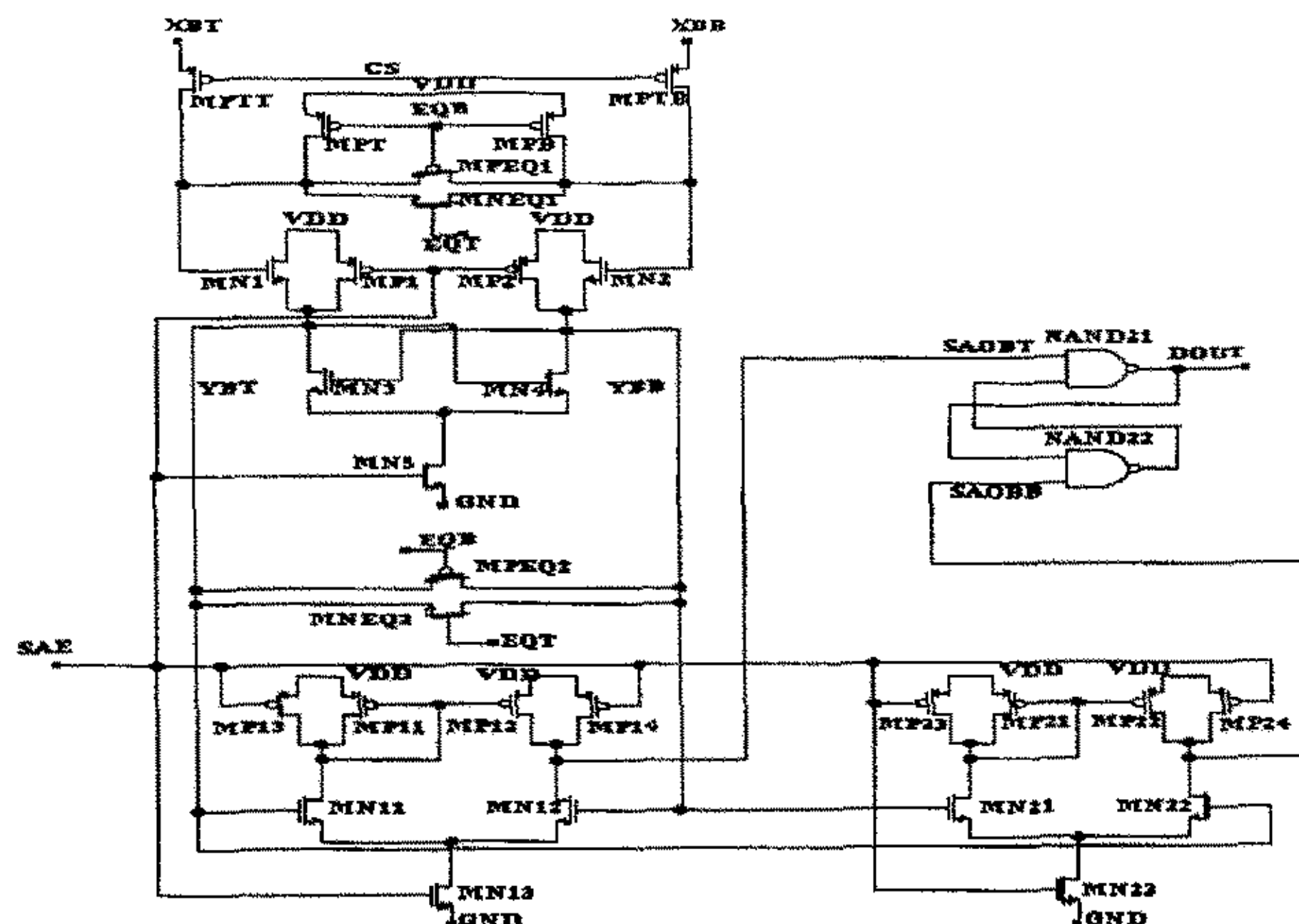


图 4-7 128Kb SRAM 中两级灵敏放大器示意图

VDD 小 50~100mv, 当 CS (列选择信号) 为逻辑“0”时, MPTT 和 MPTP 管导通, 把 XBT 和 XBB 点电位分别传到 MN1 和 MN2 的栅极, 当灵敏放大时钟信 SAE 有效时 (高电平), nMOS 源极跟随器对 XBT 和 XBB 进行电平位移, 然后交叉耦合的一对 nMOS 管立即将对得到的电平进行进一步放大。由于其中存在的正反馈作用, 最终使得 YBT 的电平比 YBB 的电位要高, 即第一级子放大器不仅可以对电平信号进行位移, 而且也可以对此信号进行放大, 如图所示。在第二级灵敏放大器中, 使用两个输入端接法相反而结构完全相同的差分放大器结构把 YBT 和 YBB 分别放大为逻辑“1”和“0”, 再通过两个两输入端的与非门交叉耦合后输出。因此, 在实际设计中所用的灵敏放大器仅由两级放大器组成。不仅如此, 为了降低交叉连接的与非门锁存器的延迟, 与非门的阈值电压比通常的要高。前面提到, 第一级交叉耦合灵敏放大器由于其中正反馈结构的影响, 使得它容易由于器件尺寸的不匹配而被锁存在一个错误的状态, 从而产生毛刺, 不能正常工作, 所以器件尺寸不易设计。为了消除毛刺, 需要添加额外的器件, 通过选择合适的时序, 确保没有毛刺, 保证了数据读出的可靠性<sup>[23]</sup>。

经过计算确定了器件的宽长比后 (W/L 的具体数值从略), 在 TSMC 0.25 $\mu m$  工艺条件下进行了 HSPICE 模拟 (电路的网表文件和 sp 文件从略)。因为所有的灵敏放大器都需要带负载, 为了更好的模拟, 我们在输出端接上 0.1pF 的电容负载和 100K 的电阻负载。对所设计的两级灵敏放大器在 T=

25°C, VDD=2.5V, 工艺条件为 TT(典型情况)情况下进行了模拟, 得出其有关性能参数波形图如图 4-8 所示。仅计算, 得出如表 4-1 所示的两级灵敏放大器性能参数。当然, 在实际中, 由于灵敏放大器在整个 SRAM 中的工作环境无法很好的模拟(如所带负载的精确估算), 因此这里的模拟只是指导性的, 最后还需要把两级灵敏放大器放在 SRAM 整个电路中通过 HSPICE 模拟进一步优化。

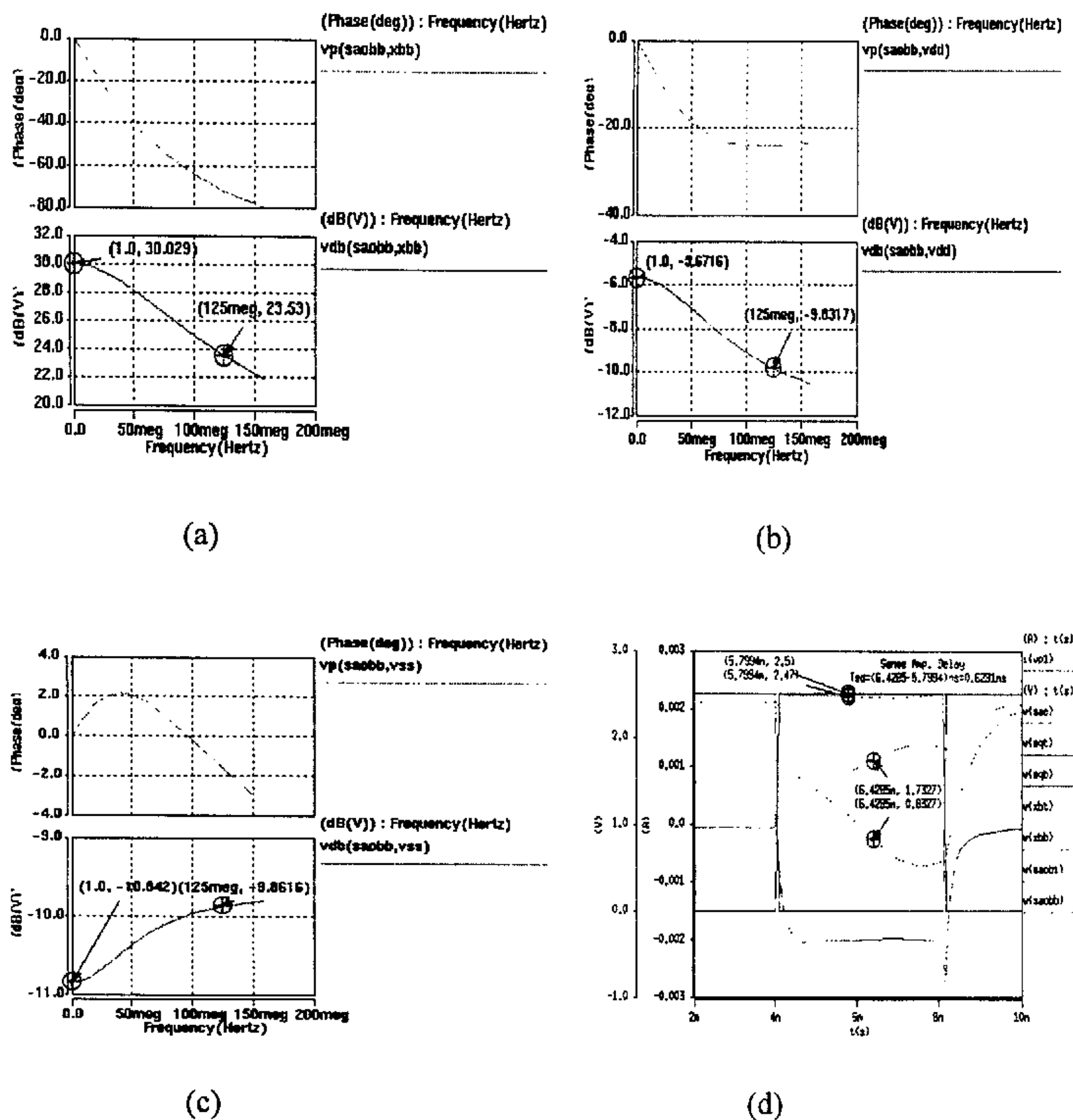


图 4-8 两级灵敏放大器模拟波形图

(a)小信号增益 (b)PSRR+ (c)PSRR- (d)灵敏放大器延迟

另外, 通过选择合适的时序使灵敏放大器在充电时不工作, 从而降低功耗。图 4-9 示出了 128Kb SRAM 中两级灵敏放大器工作波形图。由于图 4-7 中的 MPT 和 MPB 管为预充电管, MPEQ1 和 MNEQ1 管为平衡管, 在灵敏放大器工作以前, 即 SAE=“0”, EQB=“0”, EQT=“1”, 通过 MPT

$(C_L=0.1\text{pF}, R_L=100\text{k})$ 

和 MPB 管进行预充电, 使一对位线上的电平都为 VDD, 此时灵敏放大器不工作, 这就实现了降低功耗。当 EQB = “1”, EQT = “0” 时, MPT 和 MPB 管关断, 若 SAE = “1”, 灵敏放大器开始工作, 当位线对 XBT 和 XBB 点的电位分别传到 MN1 和 MN2 的栅极, 如果达到数据放大所需的较大的真值 / 互补差分电压, 则灵敏放大器就会输出正确的数值。第一级电压分辨率可达 30mV, 第二级电压分辨率可达 150 mV。

图 4-9 128Kb SRAM 中两级灵敏放大器工作波形图

长的数据总线后，会产生很大程度的信号衰减，此时不能得到正确的读出数据。为了解决这一难题，日立公司的 K. Sasaki 等人采用了三级灵敏放大器技术，提高了数据读取的速度。受此启发，在设计中我们使用了四级灵敏放大技术，目的是把衰减后的信号进一步放大为正确的数据输出。图 4-10 示出了 1Mb SRAM 中的四级灵敏放大器的结构。

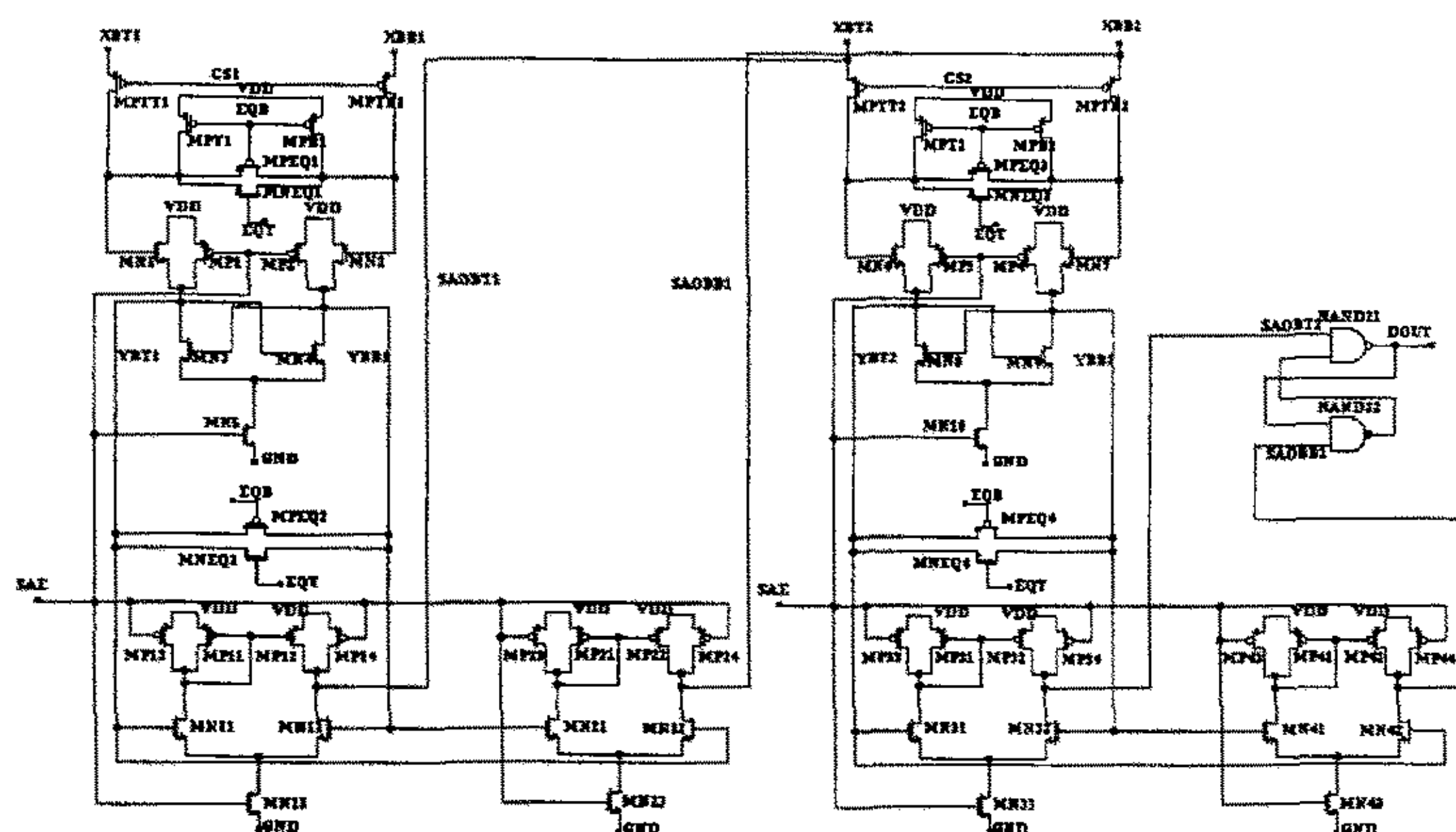


图 4-10 1Mb SRAM 中的四级灵敏放大器

图 4-11 示出了模拟得到的 1Mb SRAM 中四级灵敏放大器工作波形图。其

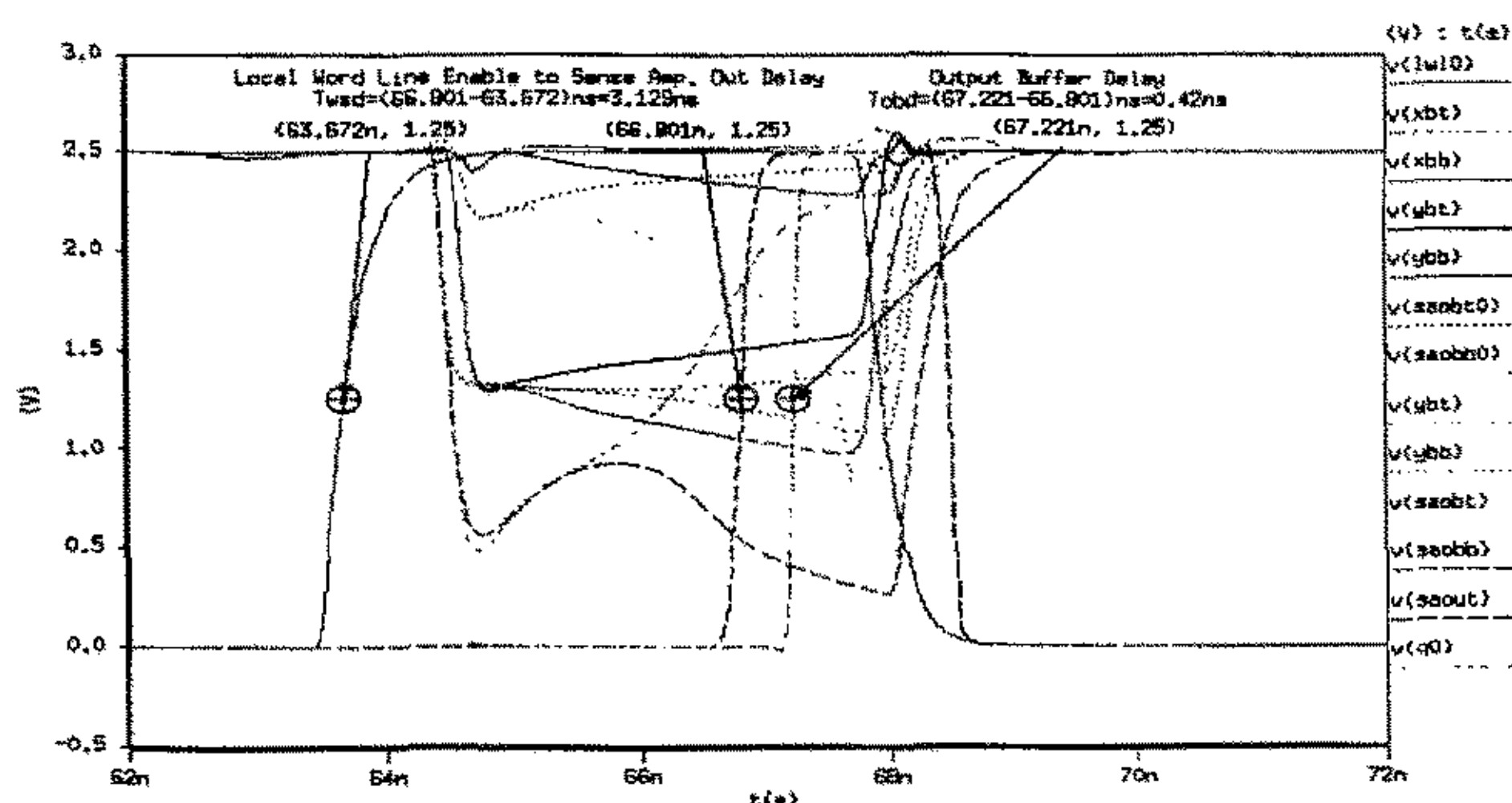


图 4-11 1Mb SRAM 四级灵敏放大器工作波形图

工作原理与两级灵敏放大器完全相同，模拟过程与方法也相同，这里不再赘述。为了分析的方便，这里省去了中间级数据总线的波形，在第五章中我们再给出其波形图。

## 4.2 带时钟两级地址译码器的设计

SRAM 的速度和功耗，主要由外围电路决定。以读出过程为例，从给出地址到数据输出所需时间（称为取数时间）的大部分消耗在地址译码器和灵敏放大器上。整个芯片的功耗的绝大部分也是消耗在外围电路上。因此，要求外围电路的速度快、功耗低。在 SRAM 芯片中，地址译码电路所占面积仅次于存储单元阵列，用来选中特定的存储单元进行读、写操作，实现随机存取的功能，同时还可以减少封装的管脚数。地址译码器一般包括行译码器和列译码器，这是因为考虑到译码器直接与排列成矩阵形式的存储单元阵列相连，如果两者之间尺寸不匹配的话，就会造成互连线的增加，从而导致不必要的延迟和功耗的增加。

译码器的传输延迟非常重要，因为它是读写时间的一部分。为了实现大而快的译码器，大多数存储器使用了复杂逻辑门。同时，将复杂逻辑门分为两层或更多层逻辑可以得到速度更快、更经济的实现方法。

在行译码器中，引入了在第一级对部分地址译码的预译码器，而第二级逻辑门产生字线信号。在我们设计的两种不同密度的 SRAM 中，第一级使用了常用的 5 个二-四预译码器，如图 4-12 所示。第二级使用了带有时钟信号的动态译码结构。我们通过产生很窄的内部时钟脉冲来使字线有效的时间尽可能的短，从而大大降低功耗<sup>[24]</sup>。两级译码结构组成了 10-1024 译码器，实现对 1024 根字线的译码功能。

在图 4-12 中，ADR[0:9]为输入的地址信号，CCLK 是用来实现同步的内部时钟，它是由内部时钟产生电路产生的。WL[0:1023]为译码输出得到的字线信号。

两级地址译码器的工作原理如下：当 CCLK 为逻辑低电平“0”时，预充电管 MP1 导通，将反相器 INV4 输入端的电平充高到高电平“1”，经过反相器的转换，使得预充电管 MP2 的栅极 WC 电位为“0”，因此节点 WW 被预充到“1”。此时输出的字线信号 WL 为“0”，即没有选中任何一条字线。反之当 CCLK 变为高电平时，MP1 管截止，而 MN1 管导通，对应于一组特定的输入地址 A9A8A7A6A5A4A3A2A1A0，必有一组 WC、PA、PB、PC 和 PD 为高电平，从而使得对应的字线电平为“1”，此时对应电路处于求值状态。译

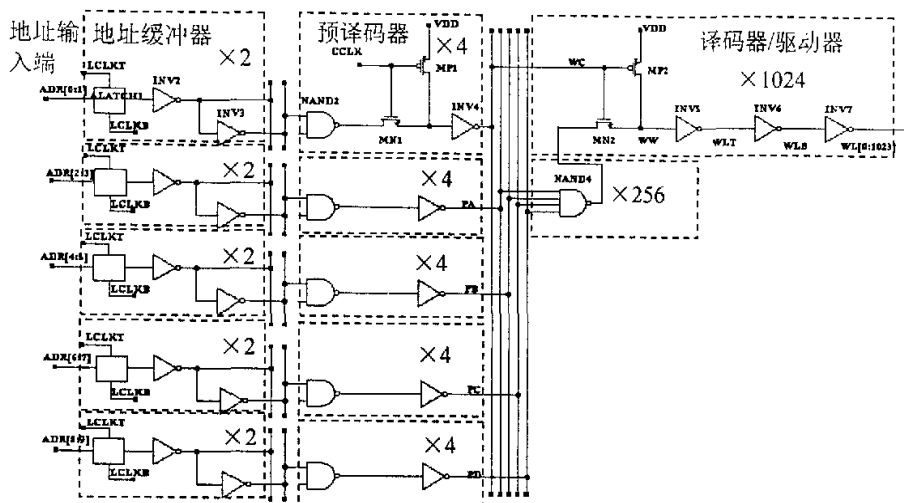


图 4-12 带有时钟信号的地址缓冲器和两级 10-1024 地址译码器

码器逻辑真值表如图 4-2 所示。

表 4-2 10-1024 译码器真值表

地址输入	ADR0	0	1	0	• • • • •	0	1
	ADR1	0	0	1	• • • • •	1	1
	ADR2	0	0	0	• • • • •	1	1
	ADR3	0	0	0	• • • • •	1	1
	ADR4	0	0	0	• • • • •	1	1
	ADR5	0	0	0	• • • • •	1	1
	ADR6	0	0	0	• • • • •	1	1
	ADR7	0	0	0	• • • • •	1	1
	A8	0	0	0	• • • • •	1	1
	A9	0	0	0	• • • • •	1	1
译码输出	WL0	1	0	0	• • • • •	0	0
	WL1	0	1	0	• • • • •	0	0
	WL2	0	0	1	• • • • •	0	0
	•	•	•	•	• • • • •	•	•
	•	•	•	•	• • • • •	•	•
	•	•	•	•	• • • • •	•	•
	WL1022	0	0	0	• • • • •	1	0
	WL1023	0	0	0	• • • • •	0	1



经过计算确定了器件的宽长比后（W/L 的具体数值从略），在 TSMC 0.25 $\mu\text{m}$  工艺条件下进行了 HSPICE 模拟（电路的网表文件和 sp 文件从略）。在未加字线负载的情况下，对所设计的两级地址译码器在  $T=25^{\circ}\text{C}$ ， $V_{\text{DD}}=2.5\text{V}$ ，工艺条件为 TT(典型情况)情况下进行了模拟，得出其仿真波形图如图 4-13(a)所示。测得从内部时钟有效到字线有效的延迟时间为 0.6ns。图 4-13(b)示出了放大的译码延迟（从 cclk 有效到 w10 有效）波形图。

互连线电容的增加，特别是全局信号线上负载电容的增加，迫切需要一种能够以较快速度进行冲放电的驱动电路。一般可以用多级反向器来组成驱动电路。具体到 SRAM 来说，因为字线上的负载很大，所以需要相应的驱动电路来提高它的冲放电速度。在图 4-12 中，INV5、INV6 和 INV7 即组成了这样的三级反向器。关于它们的宽长比之间关系的确定与优化，Jaeger 等人在相关文献中早有论述<sup>[25][26][27]</sup>。我们需要做的只是根据译码速度的要求利用 HSPICE 进行模拟，进行相关的优化。

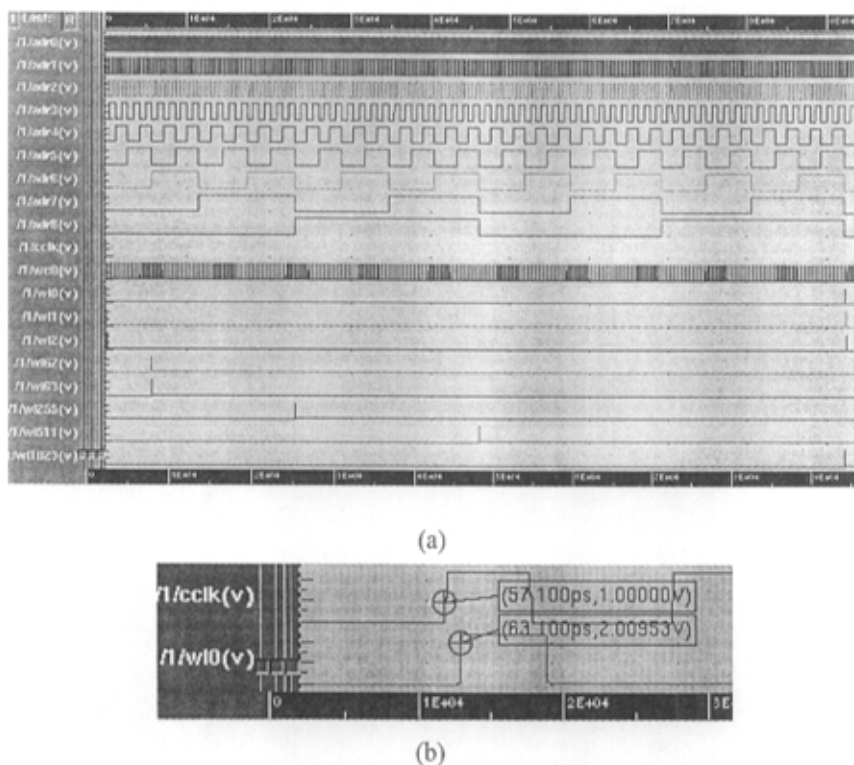


图 4-13 带时钟信号的两级 10-1024 地址译码器仿真波形  
(a) 全图 (b)放大的译码延迟（从 cclk 有效到 w10 有效）波形图

## 第五章 SRAM 的 Compiler 设计

前两章对 SRAM 的基本电路，包括其存储单元和外围电路的设计进行了讨论，本章将在 SRAM 基本电路设计的基础上，探讨 SRAM 的一种新的设计方法，即 Compiler 设计技术。

### 5.1 SRAM Compiler 概述

SRAM Compiler 是一种能生成不同存储器配置的计算机智能化程序。决定特定存储器配置的变量有字宽，字数和存储器分块数。SRAM Compiler 程序功能主要包括：产生存储器网表（netlist），模拟最坏情况下读写路径以产生时序信息，为在原理图中取代 SRAM 而建立相应的符号，建立带有时序信息的模拟模型，产生对应 SRAM 的物理版图，建立电源栅结构。通常 SRAM Compiler 用来为诸如门阵列或标准单元电路的专用集成电路（ASICs）生成存储器<sup>[28]</sup>。

常用 SRAMs 是全定制设计，这一设计以存储器密度、速度、功耗、产率和封装尺寸为主要考虑因素。为了在高度竞争的市场中引入一种成功的 SRAM 产品，上述限制因素应在短期内得到解决。全定制 SRAM 设计中的趋势是建立有限种的配置。SRAM Compiler 允许大多数普通尺寸可变字宽的架构用于以减小设计周期为目的的存储器中。同时，大的 SRAMs 在设计中能够被改变为更小的配置（例如，从 256K 位到 64K 位）。如果较高密度存储器产率很低或在更低密度配置中可以获得更快的 SRAM 的话，这就使得部分损失可以挽回。

大多数的 SRAM Compiler 设计有两个典型的特征。首先，编译器为应用建立一个单块存储器。其次，设计出使版图变化最小的译码级电路，从而降低物理版图编译的复杂程度。

性能是在计算机生成的 SRAM（即编译 SRAM）中主要的限制。建立 SRAM Compiler 是一项包括设计和计算机资源在内的任务。ASIC 芯片上的存储器尺寸和字宽因为用户应用的需要可能有很大的变化范围。设计工程师创造的存储器设计方法，简化了代码的编写以达到综合不同存储器配置和降低建立 SRAM 物理版图的复杂性的目的。



编译 SRAM 的目的是设计出使版图变化最小的译码级电路，从而降低物理版图编译的复杂程度。显然，对于大尺寸存储器，这些标准编译器属性降低了 SRAM 的性能。更大的存储器阵列尺寸增加了译码器电路输出的负载，降低了 SRAM 的读写速度。在有大量的排数或列数的情况下，遵循简化版图变化的原则建立译码器电路经常导致性能的降低。

如表 5-1 所示，编译器生成的 SRAM 和全定制 SRAM 有很大的不同。使用传统的全定制设计周期通常约需要 3~6 月，所需人员多，适应性差，未来扩展成本高。而采用 Compiler 设计仅需四个小时，所需人员少，适应性强，未来扩展成本低。用户希望编译出来的 SRAM 拥有类似于市场上标准 SRAM 的性能水平。因此，拥有一种有在不同配置情况下容易生成而且接近全定制设计的速度和密度结构的设计方法是很有现实意义的。

表 5-1 Compiler 方法与全定制设计方法优劣比较

设计方法	设计周期	可重配置性	扩展成本	性能
传统全定制	约 3~6 月	不可以	很高	很高
Compiler	四个小时	高度灵活	很低	适中

Compiler 的设计有两个主要目标：高性能和灵活性<sup>[29]</sup>。性能由速度和密度衡量，而灵活性由可扩展性和可重配置性来衡量。为了达成上述目标，设计中必须采用特别的技术，如分等级字线、多级灵敏放大等技术。在设计诸如存储器阵列、地址译码器和输入 / 输出电路等每个主要模块时使得每一部分增长的同时而不会招致很大的速度或密度损失。电路设计必须尽力考虑平面规划图并将每个部分分区来满足性能的要求。

## 5.2 SRAM Compiler 的设计原理

在大容量的 SRAM 设计中，由于存储器密度很大，为得到较快的速度和较低的功耗，在设计中常采用分块结构，即如图 5-1 所示的一种被称为“分等级字线译码”（hierarchical-word decoding）结构<sup>[30]</sup>。在只有两级的情况下，这种结构习惯上也被称为 Divided Word Line，简称 DWL 结构<sup>[31]</sup>。与传统的将译码地址通过多晶硅字线传给所有的块（Bank）不同的是，这种结构使用金属线作为“全局字线”（Global word line），从而大大降低了字线上的延迟。本地

字线（Local word line）被限制在单个的块中，只有在特定的块被选中的情况下才会被激活。

采用 DWL 结构有很多优点。首先，这一方法的优点是消除了额外的列电

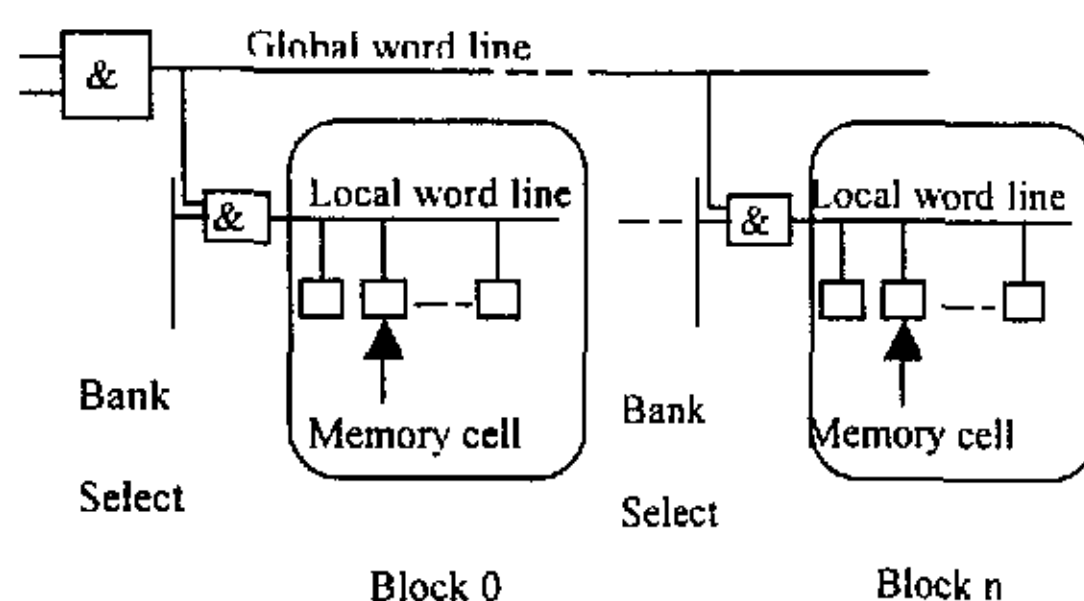


图 5-1 分等级字线译码结构

流的浪费。列电流只在一个选定的块里流动。特别在 CMOS SRAM 中，列电流占据了大部分的直流消耗。因此，在有许多列的高密度 SRAM 中采用这种结构可使总功耗降低很多。其次，这种结构可以减小字线选择延迟。从图 5-1 中可以看出，总的字线延迟包括全局字线延迟和本地字线延迟两部分。因为全局字线不包括存储单元中两个开关管的栅电容，因此全局字线上的负载比不分块时减小很多。同时，由于每一个独立的块中的本地字线长度比不分段时减小很多，所以本地字线延迟大大降低。第三，由于采用这一结构所带来的额外增加的芯片面积很小。

这样，DWL 结构通过把存储单元阵列分块和把字线分段的方法，很好地解决了在大规模集成电路静态存储器中所遇到的速度慢和功耗大的难题。

在所讨论的 128Kb~1Mb SRAM Compiler 设计实例中，SRAM 被组织成一维或两维的 1 到 8 块，每一块最多可以有 1024 行（字线）和 128 列（位线），即最多可以有 128kb 的密度。分块结构技术可以降低字线上的负载，同时单块的读写时间比不分块的要小。我们以 1Mb SRAM 设计为例，说明其设计原理。为了实现 1Mbit SRAM，将 8 个 128Kbit 存储器子块堆叠在一起，每个子块中都有相应的本地行译码器，位线负载，列译码器，选择器和区灵敏放大器，并可实现 SRAM 的全部功能。SRAM 读过程包括：地址译码；将选中存储单元中的数据置于位线上；放大位线上的低电压差；最后将读出的数据置于输出总线上。因此，数据读取时间分为以下四部分：译码延迟，字线延迟，位线/灵敏放大器延迟，数据总线/数据输出延迟<sup>[32]</sup>。亚微米集成电路中，逻辑门延迟主要由连线的 RC 延迟和扇出系数（Fan-out）决定，可用  $\pi$  模型等效<sup>[33]</sup>。

图 5-2 是用来进行模拟的 1Mb SRAM 关键数据读写信号路径图。为了更好地理解和分析读写关键路径的概念，先对设计的 1Mb SRAM 作个简单介绍，然后再分析关键路径的构成。

我们设计的 1Mb 单端口同步 SRAM 有关情况如下：

存储器名：SRAM128KX8

存储器配置：131072 字 x 8 位

芯片描述：这是一颗高性能、单端口同步静态存储器芯片。当存储器使能信号为低时，不管外部时钟、地址或其它控制信号处于何种状态，其静态电流消耗都为零。芯片能够在 2.25~2.75V 电源电压范围内正常工作，并获得很高的性能。在牺牲部分性能的前提下，最低工作电压仅为 1.8V。表 5-2 中给出了有关管脚功能描述。

表 5-2 SRAM128KX8 管脚功能描述

端口名称	SRAM128KX8 管脚功能描述
ADR[0:16]	地址输入端，用来选中特定的单元进行读 / 写
D[0:7]	数据输入总线，用来在写周期时将数据写入特定地址指定的单元
ME	存储器使能端，为逻辑高有效。当它为逻辑高电平时，存储器允许进行读 / 写；反之，存储器不允许数据读写，输出端口处于高阻态
OE	输出使能端，当它为高时，输出选中单元中存储的数据；反之处于高阻态
WE	写使能端，当它为逻辑高电平时，存储器处于写状态，反之存储器处于读状态
CLK	时钟输入端，存储器的外部时钟
Q[0:7]	数据输出总线，受当它为高时，输出选中单元中存储的数据；当输出使能端或存储器使能端无效时，数据输出总线处于高阻态

现在我们结合图 5-2 简要分析芯片的工作原理。如图所示，关键信号路径图主要由以下部分组成：行译码电路、全局驱动电路、本地驱动单路、中心电路（包括时钟产生电路、控制电路、行预译码电路、列译码电路、块选择译码电路）、存储单元、环境设置单元、哑元单元、参考电平电路和输入 / 输出电路。ADR[0:17]为地址输入端，CLK 为外部时钟，ME 为存储器使能端，WE 为写使能端，OE 为输出使能端。我们把地址输入端分成三组，其中最低的三位

地址  $ADR[0:2]$ ，用来产生块选择（Bank Select）信号，一次只选中唯一的一个存储器子块，这样，我们把整个 1Mb SRAM 大阵列等分成八个子阵列。 $ADR[3:6]$ 用来产生列选择信号，目的是从 16 对位线中选中其中的一对。 $ADR[7:16]$ 用来产生行选择信号，从而实现对 1024 根字线的译码。为了很好验证 SRAM 的读写时序和功耗，我们作出如下假设：选取距离中心电路最远的一个 Bank（编号 Bank0，使用 BS0 信号选择）里的一根字线 GWL0 上相邻的

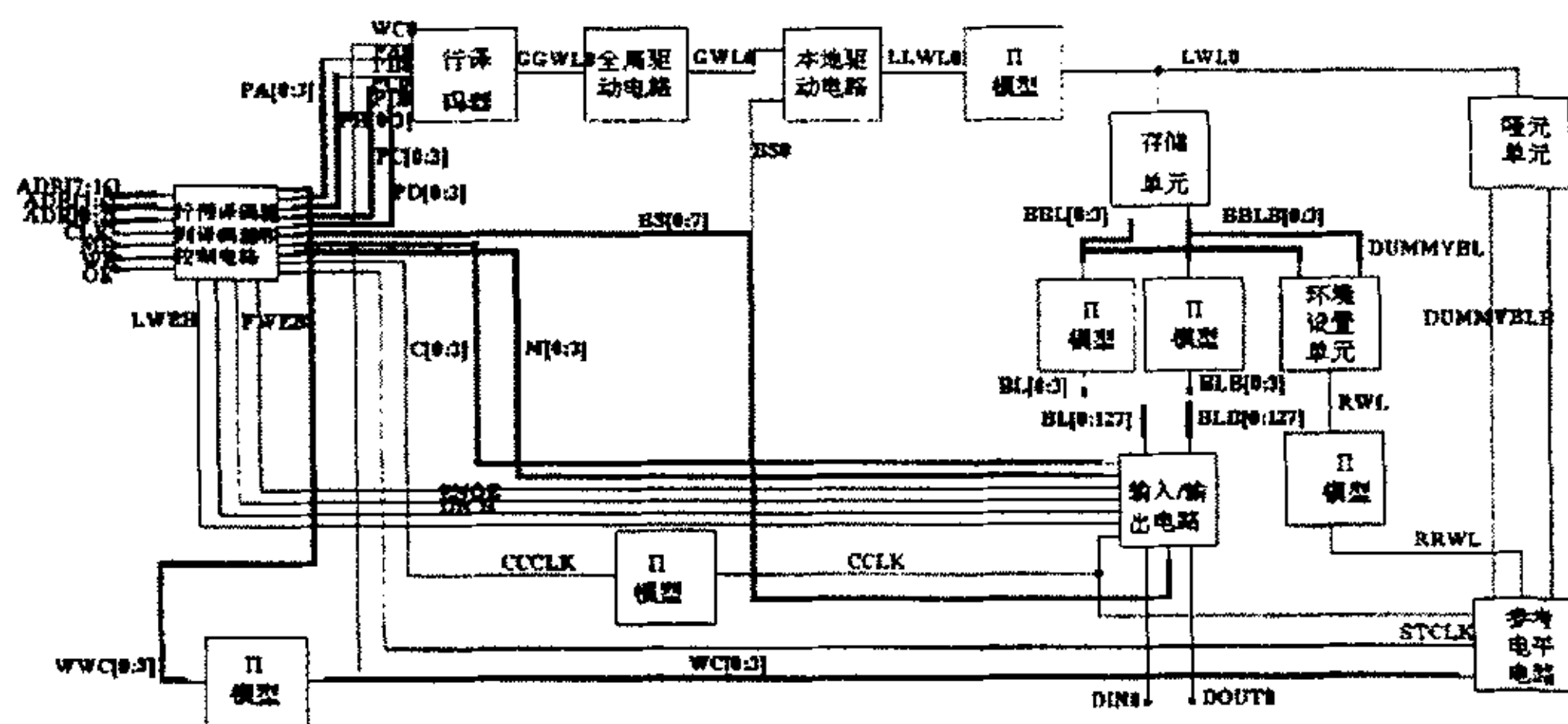


图 5-2 1Mb SRAM 关键数据读写信号路径图

四个存储单元（对应的位线编号为  $BL[0:3]$  和  $BLB[0:3]$ ），对其进行以下仿真：

1. 在 1~4 个时钟周期内，假设  $ADR[0:2] = "000"$ ，块选择信号  $BS0 = "1"$ ，而  $BS[1:7]$  都等于零，此时只有最远端的一个 Bank 被激活。将数据“1010”依次写入这四个存储单元中；
2. 在 5~8 个时钟周期内，假设  $ADR[0:2] = "000"$ ，块选择信号  $BS0 = "1"$ ，而  $BS[1:7]$  都等于零，此时只有最远端的一个 Bank 被激活。将数据“1010”依次从这四个存储单元中读出；
3. 在 9~12 个时钟周期内，假设  $ADR[0:2] = "000"$ ，块选择信号  $BS0 = "1"$ ，而  $BS[1:7]$  都等于零，此时只有最远端的一个 Bank 被激活。将数据“0101”分别写入这四个存储单元中；
4. 在 13~16 个时钟周期内，假设  $ADR[0:2] = "000"$ ，块选择信号  $BS0 = "1"$ ，而  $BS[1:7]$  都等于零，此时只有最远端的一个 Bank 被激活。将数据“0101”依次从这四个存储单元中读出。

在使用 HSPICE 软件进行实际仿真时，却发现 1Mb SRAM 太大了，完成一次完整的时序和功耗仿真需要很漫长的时间。因此，设计中通常采用关键信号路径来进行仿真。实际上，在减小存储密度和一次读出位数的情况下，使用



关键信号通道仿真，因为外围电路中的译码器和输入 / 输出电路的结构都没有改变，得到的 SRAM 的取数时间几乎不会受到任何影响，但是整个 SRAM 的功耗会大许多。只要我们把仿真得到的一个输入 / 输出电路的功耗，加上实际中其余未加入关键信号路径的位数对应的输入 / 输出电路的功耗，就可以比较准确地估算出整个芯片的功耗。

版图中使用了一层多晶硅和五层金属。其中，本地字线 LWL 使用了多晶硅和第二层金属混合字线结构，全局字线 GWL 使用第二层金属，第一层金属用作存储单元内部的连线，WC 信号线，位线 BL 和块选择信号 BS 使用第三层金属线，内部时钟 CCLK 使用第四层金属线，电源环状结构连线使用第五层金属。经计算，1Mb SRAM 芯片面积为  $4.9\text{mm} \times 4.8\text{mm}$ 。

下面分别估算关键信号路径图中  $\pi$  模型中 RC 的近似值。

如前所述，亚微米集成电路中，逻辑门延迟主要由连线的 RC 延迟和扇出系数 (Fan-out) 决定。金属互连线或多晶硅互连线的寄生电阻较大时会对线上的信号产生显著的影响，连线的电阻取决于材料（比如多晶硅，铝或金），线的尺寸和接触的数目和位置。

如图 5-3 所示，假设连线沿电流方向长度为  $l$ ，宽度为  $w$ ，厚度为  $t$ 。沿电流方向的连线电阻由下式决定：

$$R_{wire} = \rho \cdot \frac{l}{w} = R_{\square} \left( \frac{l}{w} \right) \quad (5-1)$$

其中， $\rho$  表示互连线材料的特征电阻率， $R_{\square}$  表示连线的方块电阻，单位是  $\Omega/\square$ 。

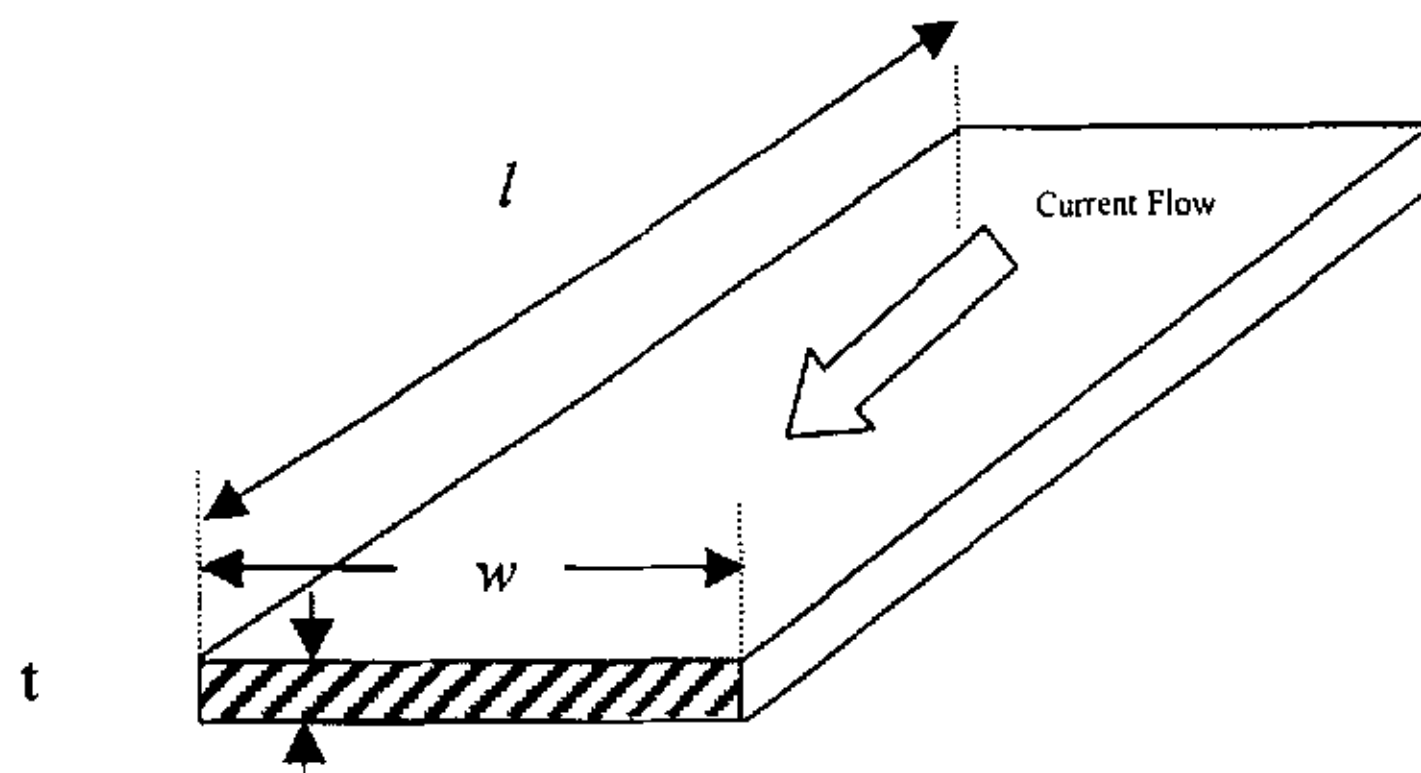


图 5-3 连线材料电阻的示意图

下面计算本地字线 LWL 上的寄生电阻。根据版图，本地字线由多晶硅和第二层金属两部分混合组成，它的特点是每隔 8 个存储单元使用金属连线在多晶硅字线上打一个“Strap”，即旁路一次，具体的结构分析会在后面的版图设计中加以说明。为了较准确地估算 WL 上的寄生电阻，我们使用表 5-3 所示的工艺寄生方块电阻参数，并使用式 (5-1) 计算：

其中， $R_{LWL\_metal2} = 75.35\Omega$ ， $R_{LWL\_poly} = 861.04\Omega$ ，

$R_{LWL\_total} = R_{LWL\_metal2} + R_{LWL\_poly} = 936.39\Omega$ 。同理，经过计算我们得到其它相关的寄生电阻值列于表 5-4 中。

表 5-3 TSMC  $0.25\mu m$  工艺寄生方块电阻参数表

Film	Sheet Resistance( $\Omega/\square$ )
N+ Poly w/I Silicide	10
Metal1	0.076
Metal2	0.076
Metal3	0.076
Metal4	0.076
Metal5	0.076

表 5-4 1Mb SRAM 关键路径中信号连线上的寄生电阻

信号连线	估算得到的寄生电阻( $\Omega$ )
GWL	391.24
LWL	936.39
RWL	1474.59
BL/BLB	517.05
WC	735.29
CCLK	17.34
BS	461.73

下面我们再计算互连线寄生电容。一般在版图中，互连线寄生电容有两种结构：即结构 A 和结构 B。结构 A 的示意图，如图 5-4 所示。

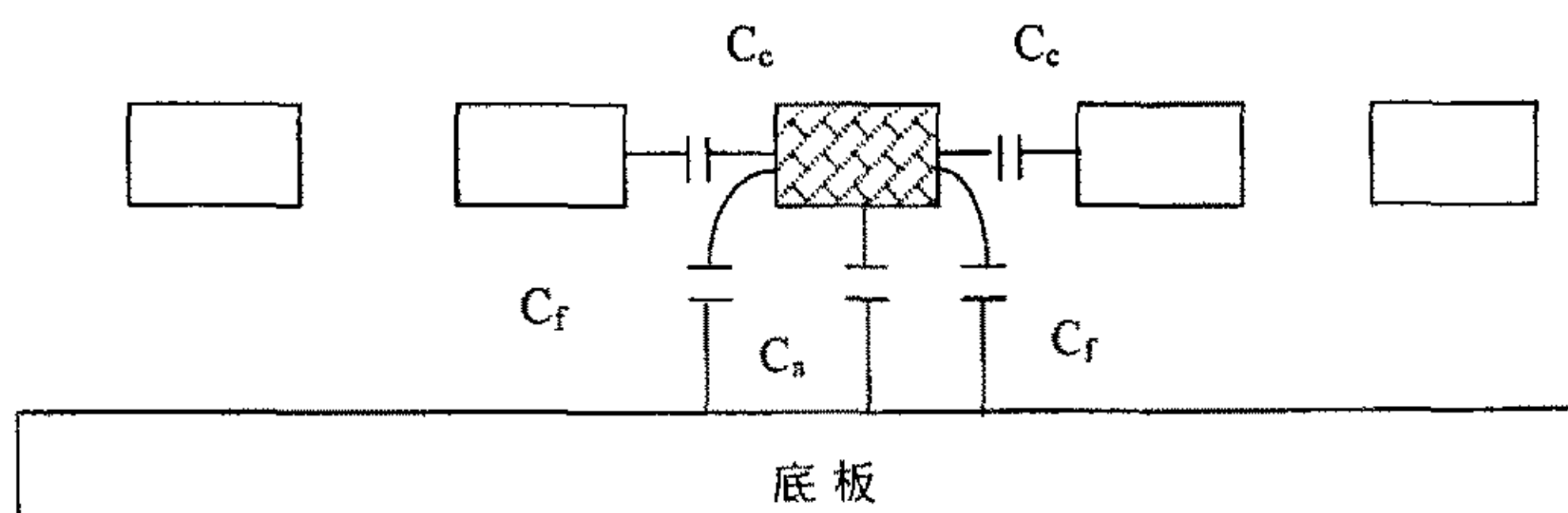


图 5-4 互连线寄生电容结构 A 示意图

在图 5-4 中，各种寄生电容含义及计算公式如下：

$C_{total}$ ：顶部中央连线实际总寄生电容；

$C_c$ ：顶部中央连线与相邻信号线之间的线-线间耦合电容；

$$C_c = (\epsilon_{ox}/d_c) * L_c * t_c \quad (5-2)$$

其中， $\epsilon_{ox}$  为介质介电常数，单位是 F/m， $d_c$  为相邻两条线之间的距离， $L_c$  为相邻两条线共同走过的距离， $t_c$  为信号线厚度。

$C_{bottom}$ ：  $C_a + 2 * C_f$ ；

$C_a$ ：顶层对底层的面积电容；

$$C_a = (\epsilon_{ox}/t_{ox}) * W * L \quad (5-3)$$

其中， $\epsilon_{ox}$  为介质介电常数，单位是 F/m， $t_{ox}$  为介质厚度， $W$ 、 $L$  为连线的长和宽。

$C_f$ ：顶层中央连线的每一边对无限大底板的边缘电容；

$$C_f = L_f * C_{fu} \quad (5-4)$$

其中， $L_f$  为连线其中一边的长度， $C_{fu}$  为单位长度的边缘电容。

$C_{sum}$ ：近似估算得到的顶部中央连线总寄生电容；

$$C_{sum} = C_a + 2 * C_f + 2 * C_c \quad (5-5)$$

因此，把式 (5-2)、(5-3) 和 (5-4) 代入上式中，得到顶部中央连线总寄生电容的近似公式：

$$C_{sum} = (\epsilon_{ox}/t_{ox}) * W * L + 2L_f * C_{fu} + 2(\epsilon_{ox}/d_c) * L_c * t_c \quad (5-6)$$

工艺测试表明，使用公式 (5-6) 估算寄生电容的误差一般在 10% 以内。

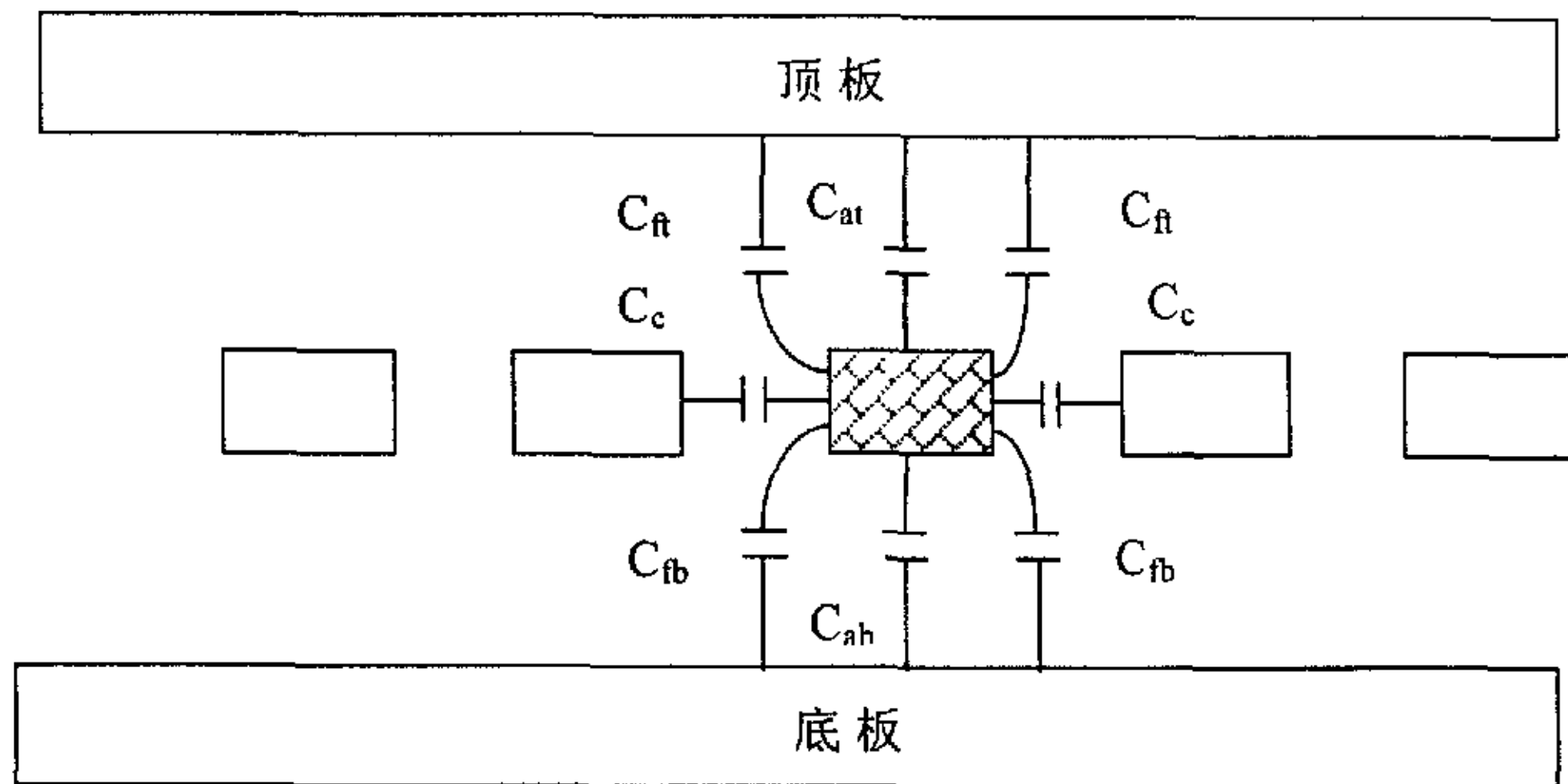


图 5-5 互连线寄生电容结构 B 示意图

在图 5-5 中，各种寄生电容含义如下：

$C_{total}$ ：中间层中央连线总寄生电容；

$C_c$ ：中间层中央连线与相邻信号线之间的线-线间耦合电容；



$$C_c = (\epsilon_{ox}/d_c) * L_c * t_c \quad (5-7)$$

其中， $\epsilon_{ox}$  为介质介电常数，单位是 F/m， $d_c$  为相邻两条线之间的距离， $L_c$  为相邻两条线共同走过的距离， $t_c$  为信号线厚度。

$$C_{bottom} = C_{ab} + 2 * C_{fb};$$

$C_{ab}$ : 中间层对底层的面积电容;

$$C_{ab} = (\epsilon_{ox}/t_{ox}) * W * L \quad (5-8)$$

其中， $\epsilon_{ox}$  为介质介电常数，单位是 F/m， $t_{ox}$  为介质厚度， $W$ 、 $L$  为连线的长和宽。

$C_{fb}$ : 中间层中央连线的每一边对无限大底板的边缘电容;

$$C_f = L_{fb} * C_{fu} \quad (5-9)$$

其中， $L_{fb}$  为连线其中一边的长度， $C_{fu}$  为单位长度的边缘电容。

$C_{sum}$ : 近似估算得到的顶部中央连线总寄生电容;

$$C_{top} = C_{at} + 2 * C_{ft};$$

$C_{at}$ : 中间层对顶层的面积电容;

$$C_{at} = (\epsilon_{ox}/t_{ox}) * W * L \quad (5-10)$$

其中， $\epsilon_{ox}$  为介质介电常数，单位是 F/m， $t_{ox}$  为介质厚度， $W$ 、 $L$  为连线的长和宽。

$C_{ft}$ : 中间层中央连线的每一边对无限大底板的边缘电容;

$$C_{ft} = L_{ft} * C_{fu} \quad (5-11)$$

其中， $L_{ft}$  为连线其中一边的长度， $C_{fu}$  为单位长度的边缘电容。

$C_{sum}$ : 近似估算得到的顶部中央连线总寄生电容;

$$C_{sum} = C_{ab} + 2 * C_{fb} + C_{at} + 2 * C_{ft} + 2 * C_c \quad (5-12)$$

因此，把式 (5-7) 到式 (5-11) 代入上式中，得到中间层中央连线总寄生电容的近似公式:

$$\begin{aligned} C_{sum} \\ = 2(\epsilon_{ox}/t_{ox}) * W * L + 2L_{fb} * C_{fu} + 2L_{ft} * C_{fu} + 2(\epsilon_{ox}/d_c) * L_c * t_c \end{aligned} \quad (5-13)$$

工艺测试表明，使用公式 (5-13) 估算寄生电容的误差一般在 10% 以内。

为了较准确地估算关键信号路径中各种信号连线上的寄生电容，我们使用表 5-5 所示的工艺寄生电容参数，并使用公式 (5-6) 或公式 (5-13) 计算。

经过计算我们得到相关的寄生电容值列于表 5-6 中。

使用从版图中提取出来的工艺参数估算出  $\pi$  模型中 RC 的值，写入电路文件中，在 TSMC0.25  $\mu m$  条件下，使用 Hspice 模拟，可得到所需时序和功耗参数如表 5-7 所示。表中也列出了 128Kb SRAM 的模拟结果。128Kb SRAM 芯片面积为 0.61mm  $\times$  4.4mm。同理对于 128Kb SRAM，译码器的尺寸和参考电路等都保持不变，我们只需要对其余电路作出很少改变，就可以实现兼容。图 5-6 是用来进行模拟的 128Kb SRAM 关键数据读写信号路径图。

表 5- 5 TSMC 0.25 $\mu\text{m}$  工艺互连线寄生电容参数表

Parameter		Capacitance	Capacitance	DielectricThickness
Top	Bottom	Area (aF / $\mu\text{m}^2$ )	Fringe (aF/ $\mu\text{m}$ )	Dielectric Thickness (nm)
Poly	Field	87.5	54.03	396.3
Metal1	Field	30.2	39.63	1165.5
Metal1	Active	41.3	46.84	858.7
Metal1	Poly	57.0	54.48	621.4
Metal2	Field	13.3	24.67	2697.6
Metal2	Active	14.9	26.88	2419.6
Metal2	Poly	16.6	28.71	2175.5
Metal2	Metal1	35.8	45.36	1015.4
Metal3	Field	8.87	18.28	4077.0
Metal3	Active	9.40	19.32	3842.1
Metal3	Poly	10.0	20.09	3603.5
Metal3	Metal1	14.7	26.64	2473.7
Metal3	Metal2	40.7	48.62	891.7
Metal4	Field	6.48	14.10	5569.8
Metal4	Active	6.81	14.78	5312.2
Metal4	Poly	7.00	15.09	5172.5
Metal4	Metal1	8.91	18.42	4075.5
Metal4	Metal2	14.6	26.73	2481.3
Metal4	Metal3	35.3	45.16	1029.2
Metal5	Field	5.16	12.04	7000.9
Metal5	Active	5.41	12.36	6697.4
Metal5	Poly	5.44	12.39	6654.8
Metal5	Metal1	6.56	14.48	5538.9
Metal5	Metal2	9.10	19.31	3989.3
Metal5	Metal3	14.3	27.43	2545.9
Metal5	Metal4	38.3	52.07	949.2

表 5-6 1Mb SRAM 关键路径中信号连线上的寄生电容

信号连线	估算得到的寄生电容(fF)
GWL	291.24
LWL	93.88
RWL	63.75
BL/BLB	730.59
WC	635.29
CCLK	57.34
BS	299.73

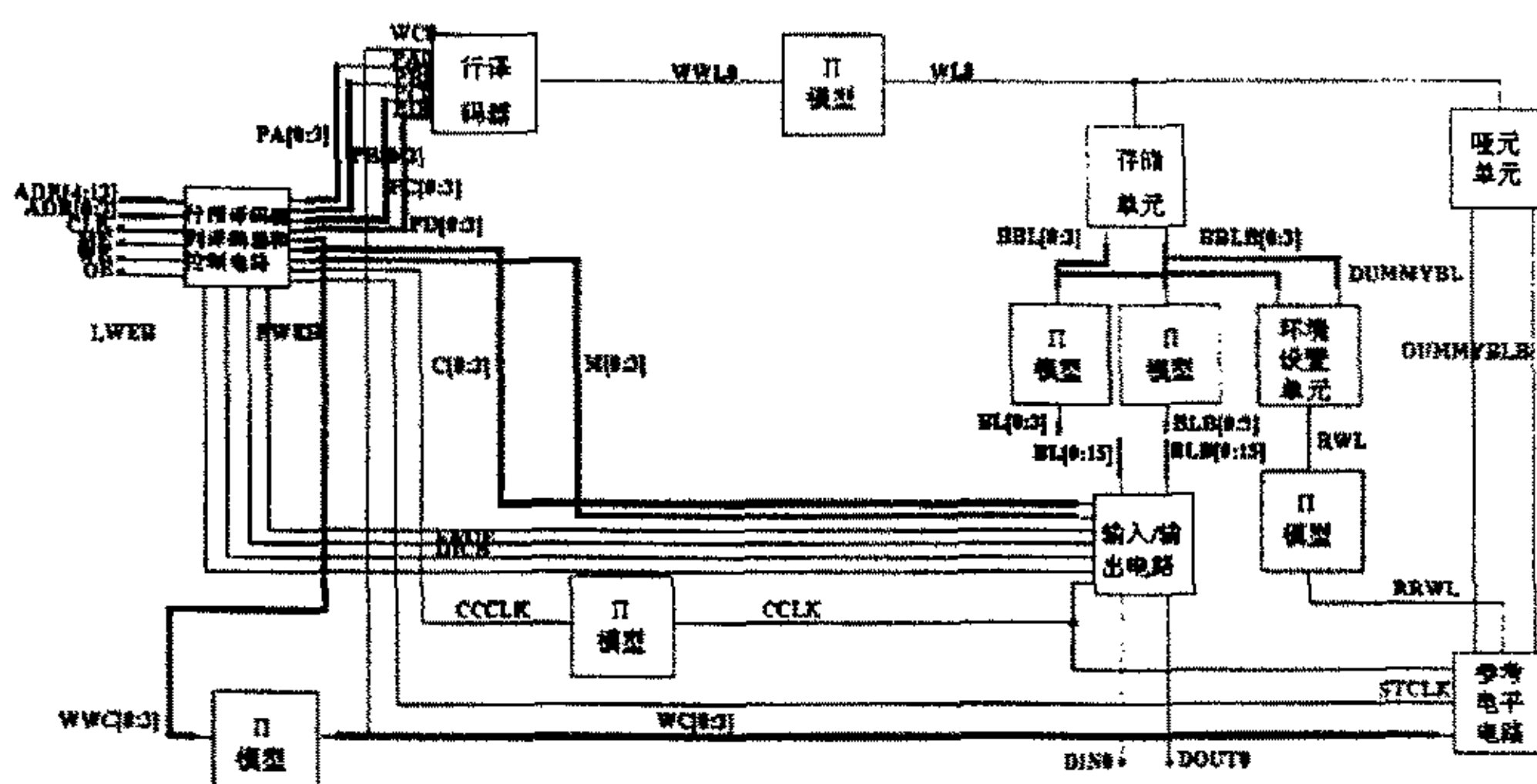


图 5-6 128Kb SRAM 关键数据读写信号路径图

从表 5-7 中可以看出，我们设计的 128Kb SRAM 在  $V_{DD}=2.5V$ ， $T=25^{\circ}C$ ，工艺条件为典型的情况下，取数时间为 3.092ns，在 125MHz 工作频率下功耗为 39.555mW。而采用 Compiler 技术设计的 1Mb SRAM 在  $V_{DD}=2.5V$ ， $T=25^{\circ}C$ ，工艺条件为典型的情况下，虽然密度增大为原来的八倍，但取数时间仅为 5.971ns，工作频率仍可高达 74MHz，功耗仅为 63.435mW。

表 5-7 SRAM 时序和功耗参数

速度 密度	块 数	数据读取时间 $T_{ca}$ (ns)			数据读写平均功耗 P(mW)		
		Best	Typical	Worst	Best	Typical	Worst
		$V_{DD}=2.75V$ $T=-40^{\circ}C$	$V_{DD}=2.5V$ $T=25^{\circ}C$	$V_{DD}=2.25V$ $T=125^{\circ}C$	$V_{DD}=2.75V$ $T=-40^{\circ}C$	$V_{DD}=2.5V$ $T=25^{\circ}C$	$V_{DD}=2.25V$ $T=125^{\circ}C$
128Kb	1	2.062 @ $T_{clk}=5.6$	3.092 @ $T_{clk}=8$	5.575 @ $T_{clk}=12$	74.995 @125MHz	39.555 @125MHz	19.640 @125MHz
1M	8	3.865 @ $T_{clk}=9$	5.971 @ $T_{clk}=13.6$	9.416 @ $T_{clk}=21$	127.483 @74MHz	63.435 @74MHz	30.623 @74MHz

注:  $T_{ca}$ ——数据读取时间, 即时钟上升沿到数据有效输出的上升沿;

Best, Typical, Worst 分别指工艺的最好、典型和最坏情况;

$T_{clk}$  为时钟周期, @表示当...时;

图 5-7 和图 5-8 分别是 1Mb SRAM 和 128Kb SRAM 的取数时间仿真波形图。

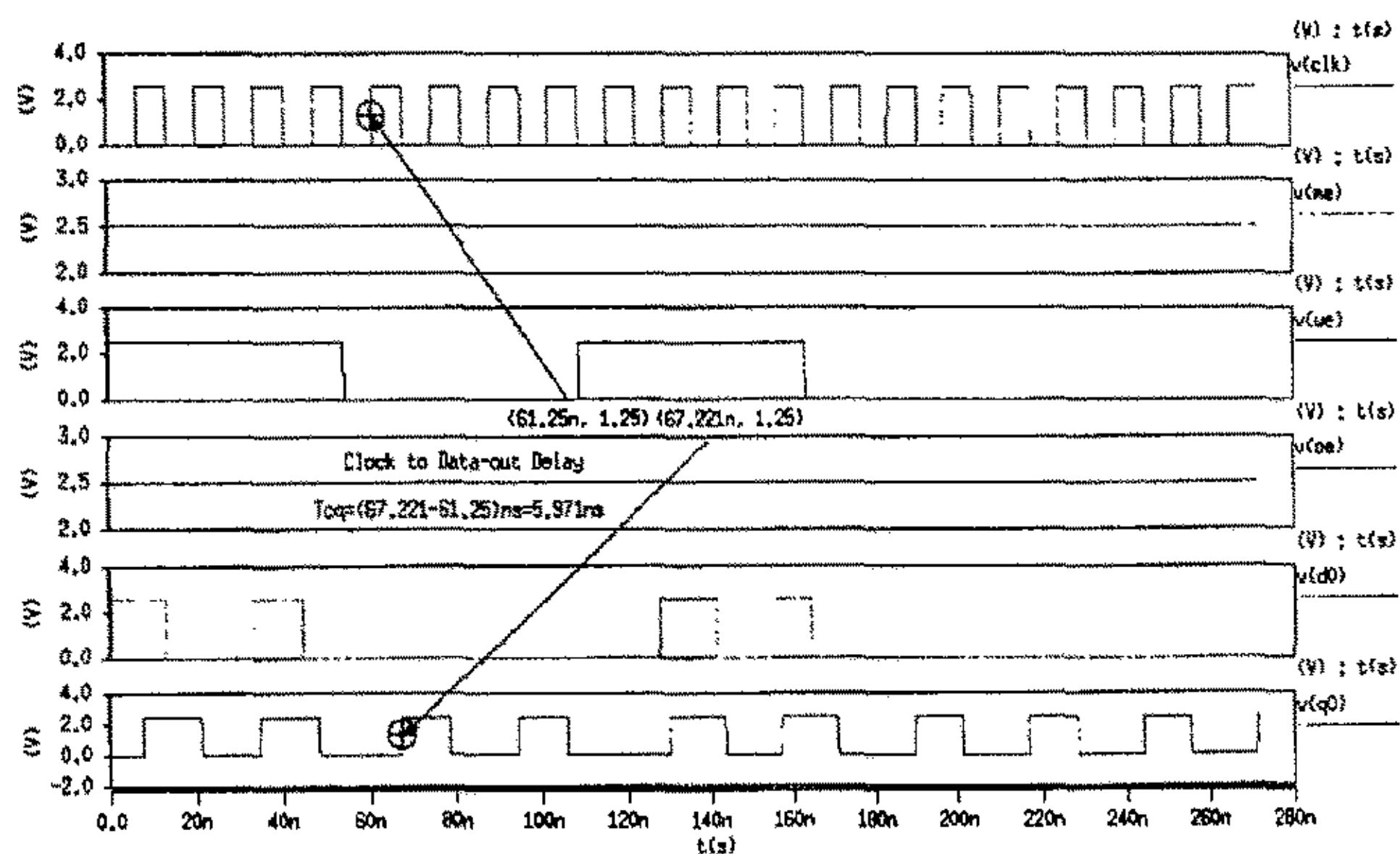


图 5-7 1Mb SRAM 仿真时序图

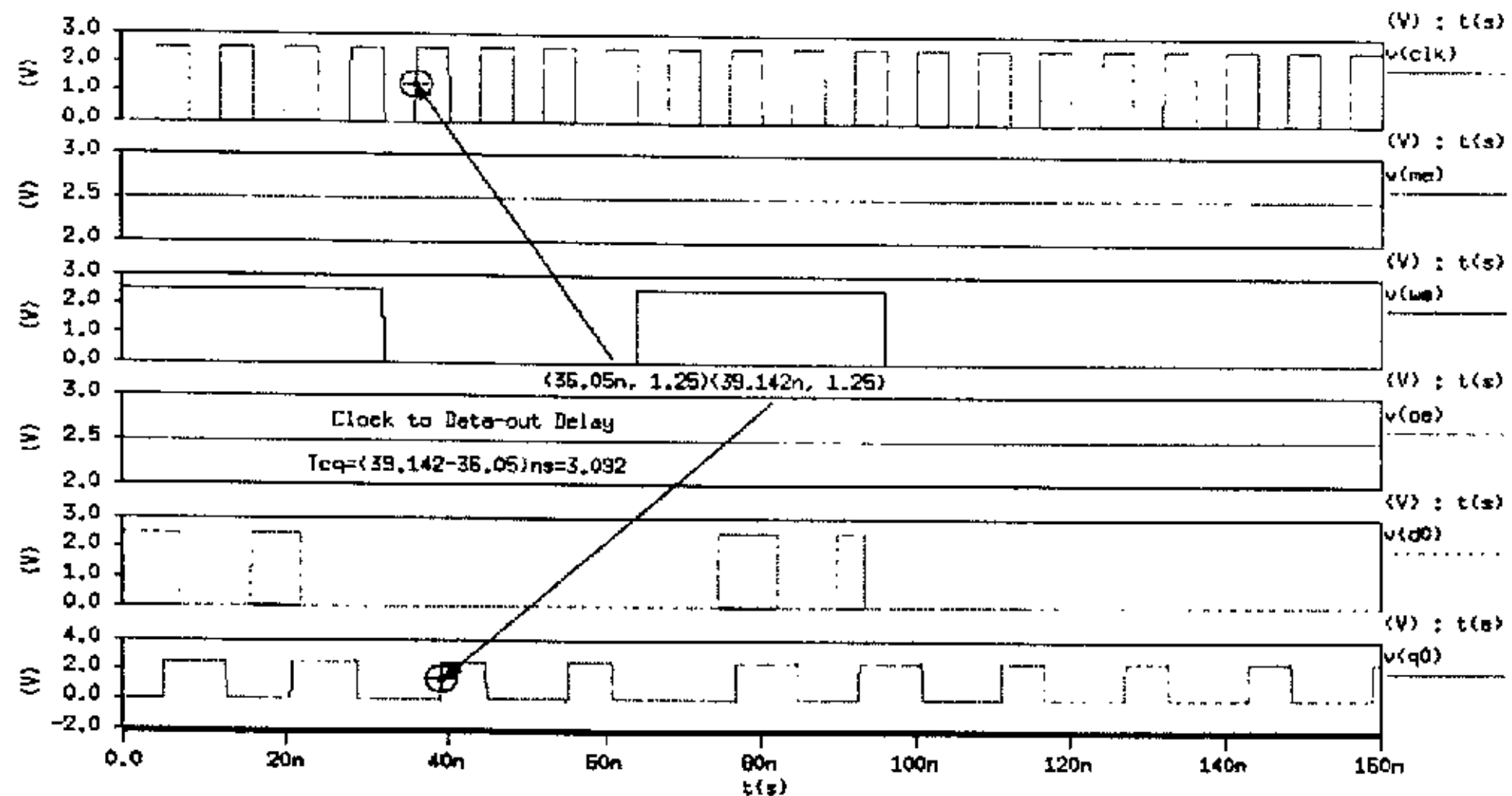
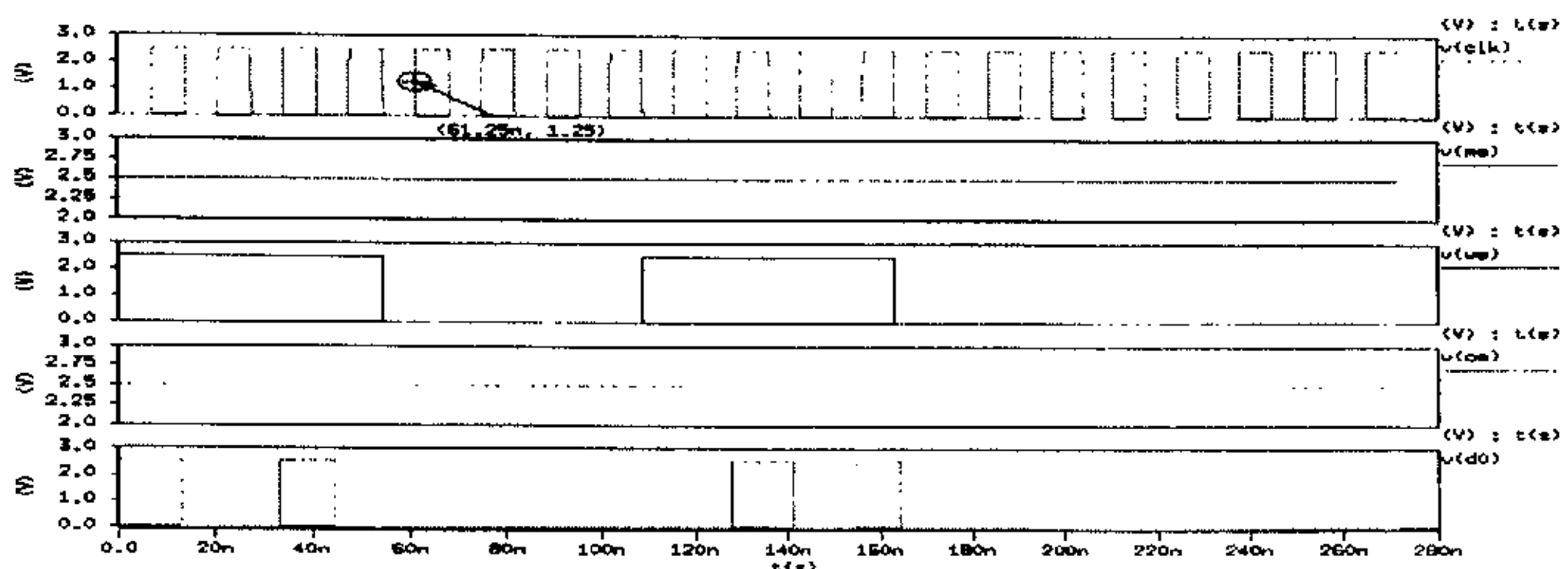


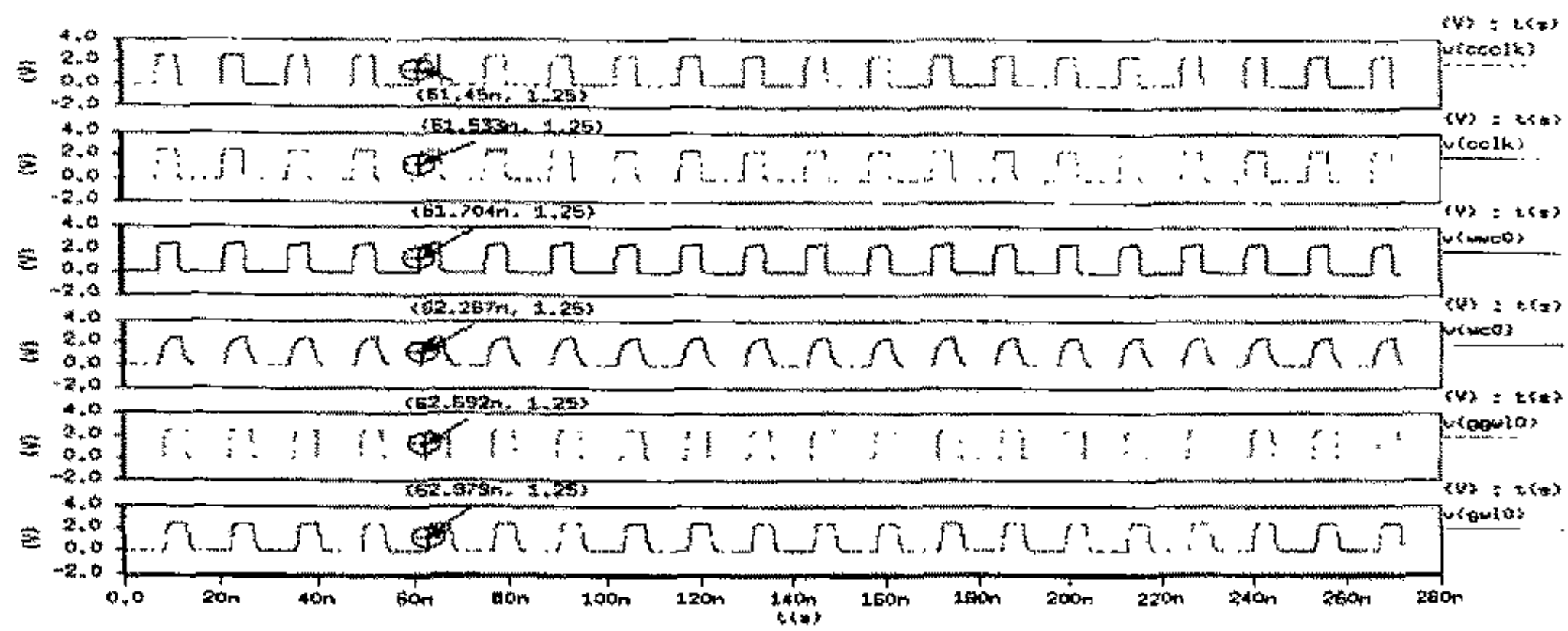
图 5-8 128 Kb SRAM 仿真时序图

图 5-9 说明了包含在整个 SRAM 阵列和子模块的关键时序控制中的信号流程。为了更好地分析时序，我们把图 5-7 所示的波形图进一步分解为 5-9(a)~(f) 六个波形图。如图所示，周期开始于产生时序信号的 SRAM 时钟，然后 SRAM 时钟将时序信号传给译码时钟，译码时钟再把时钟信号传给全局字线 GWL。相应地，产生的全局字线信号被传送到本地字线重驱动缓冲器，产生被 SRAM 单元接收的本地字线信号 LWL，使其在存储器阵列字线上产生真值 / 互补的差分电压。伴随于全局字线信号的产生，译码时钟产生一个依赖于尺寸的延迟，并将它传给全局控制时钟驱动器。收到时序延迟信号后，全局控制时钟驱动器将全局控制信号传给本地不同的控制逻辑和速度控制电路，在其中产生一个依赖于尺寸的延迟灵敏放大时钟信号，这一信号传给灵敏放大器，与此同时位线上出现为正确数据放大所需的较大的真值 / 互补差分电压。通过选择合适的时序使灵敏放大器在充电时不工作，从而降低功耗。

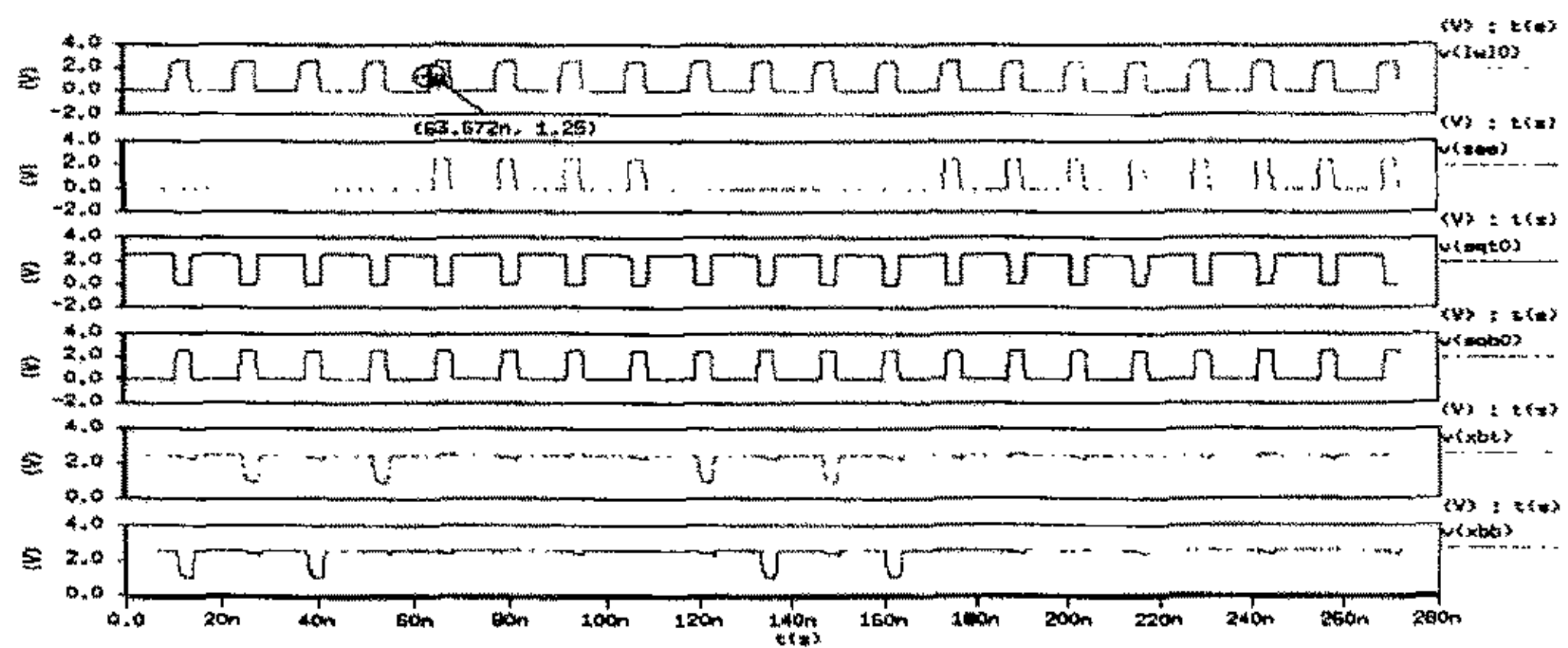


(a) 1Mb SRAM 输入信号波形图

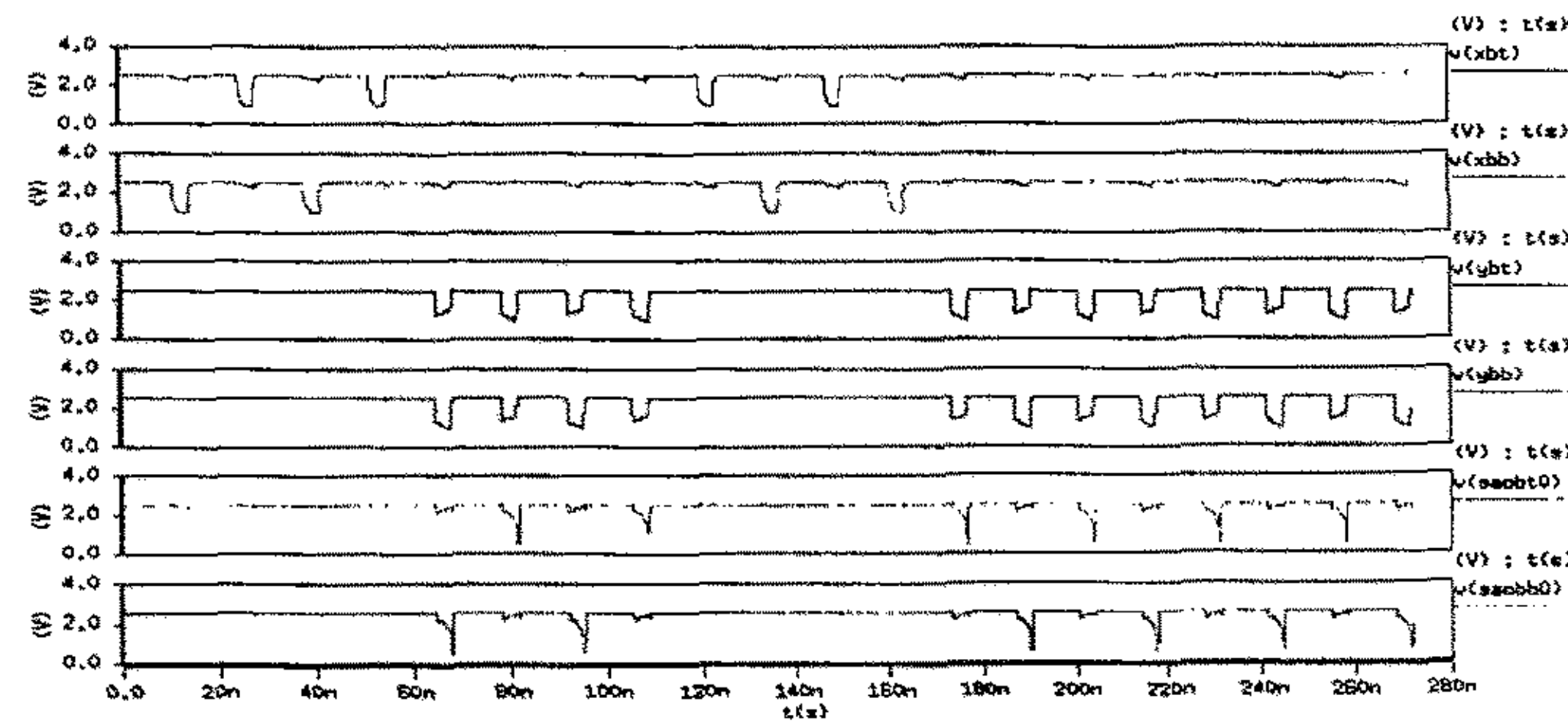




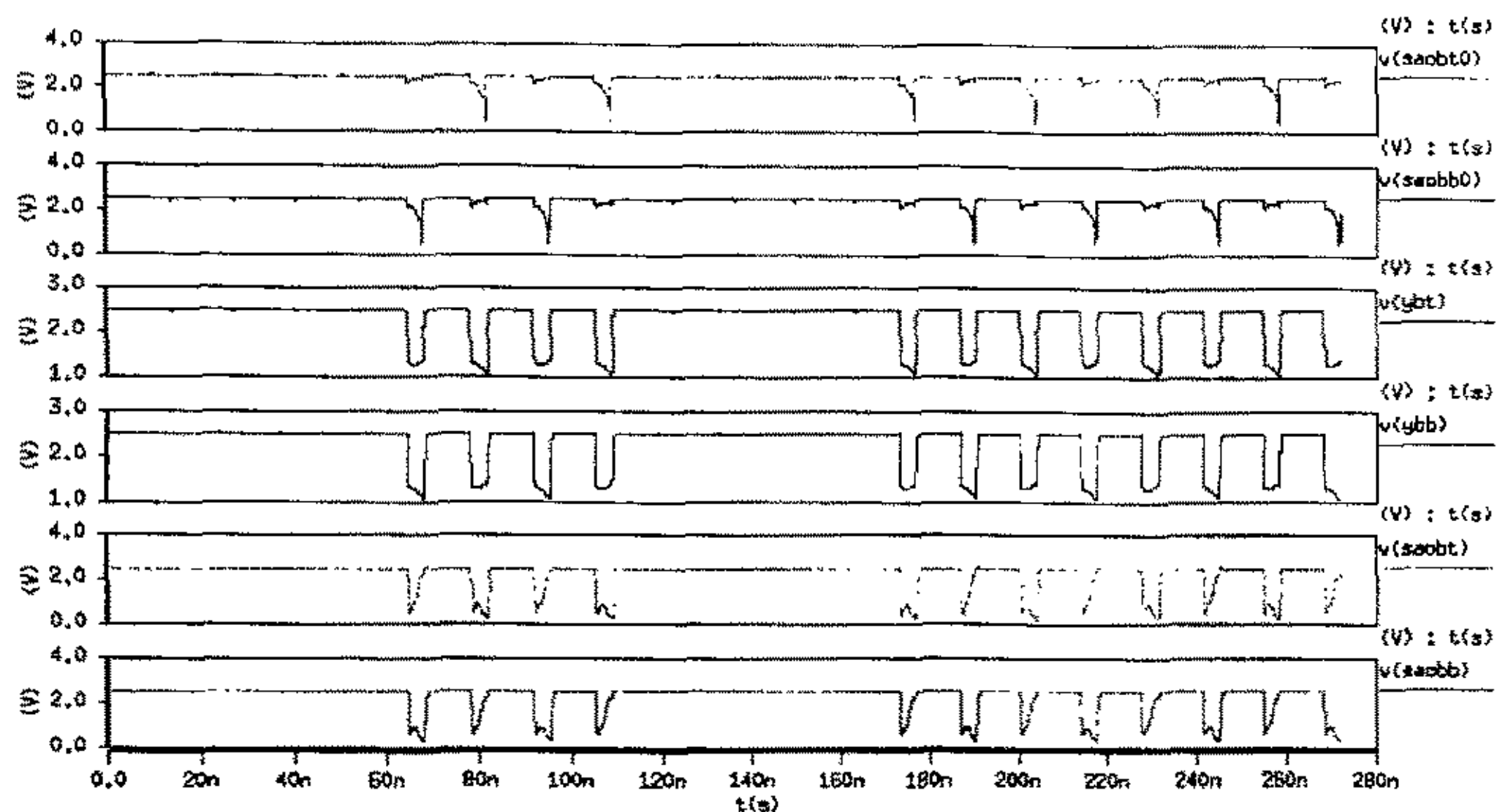
(b) 1Mb SRAM 内部时钟 cclk 有效到全局字线 gw10 有效时序图



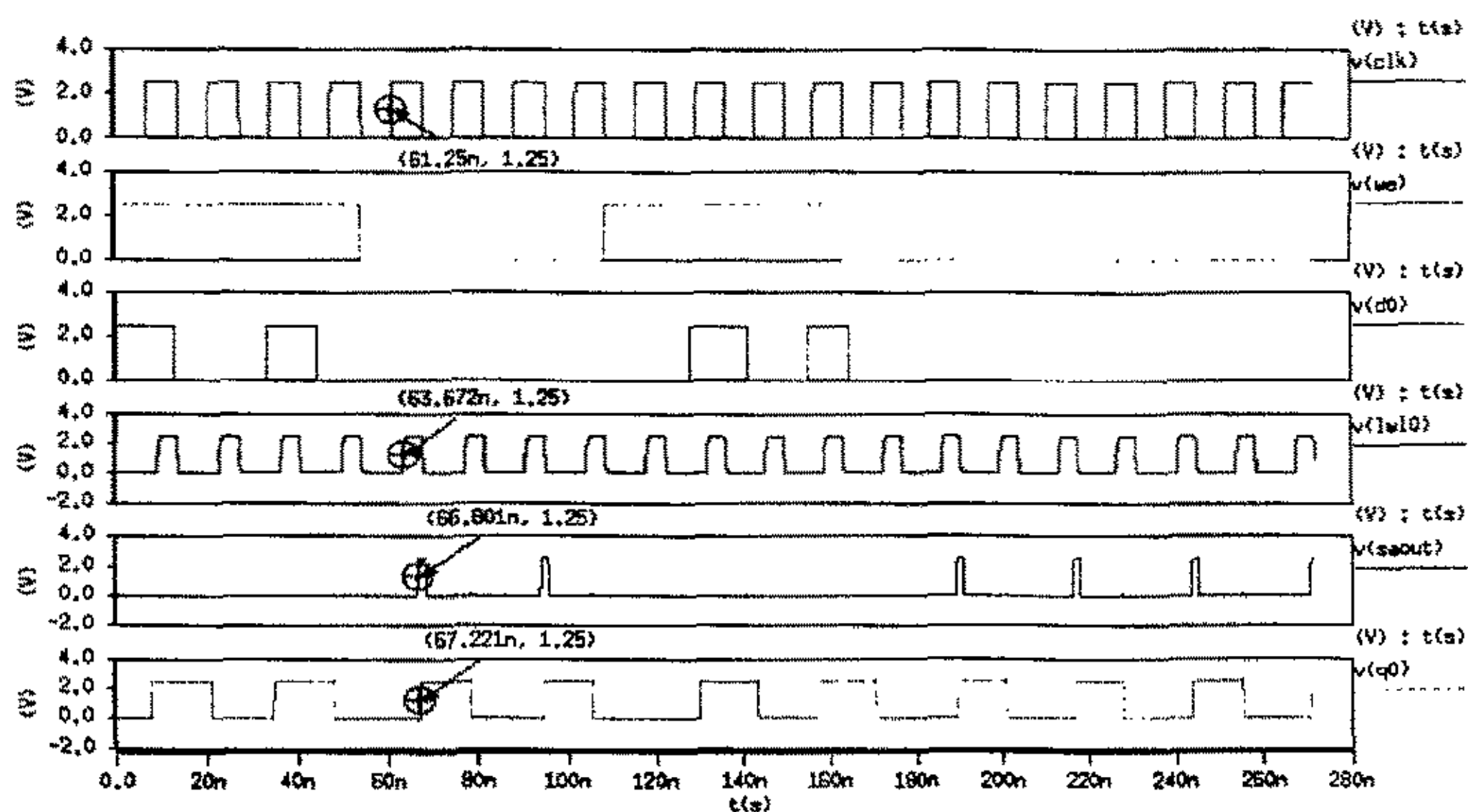
(c) 1Mb SRAM 读 / 写操作过程中本地字线 lw10、灵敏放大使能信号 sae、本地数据总线平衡信号 eqt0、eqb0 和本地数据总线信号 xbt、xbb 时序图



(d) 1Mb SRAM 读 / 写操作过程中本地数据总线信号 xbt、xbb、第一级灵敏放大器输出信号 ybt、ybb、第二级灵敏放大器输出信号 saobt0、saobb0



(e) 1Mb SRAM 读 / 写操作过程中第二级灵敏放大器输出信号 saobt0、saobb0、第三级灵敏放大器输出信号 ybt、ybb、第四级灵敏放大器输出信号 saobt、saobb



(f) 1Mb SRAM 取数时间时序图

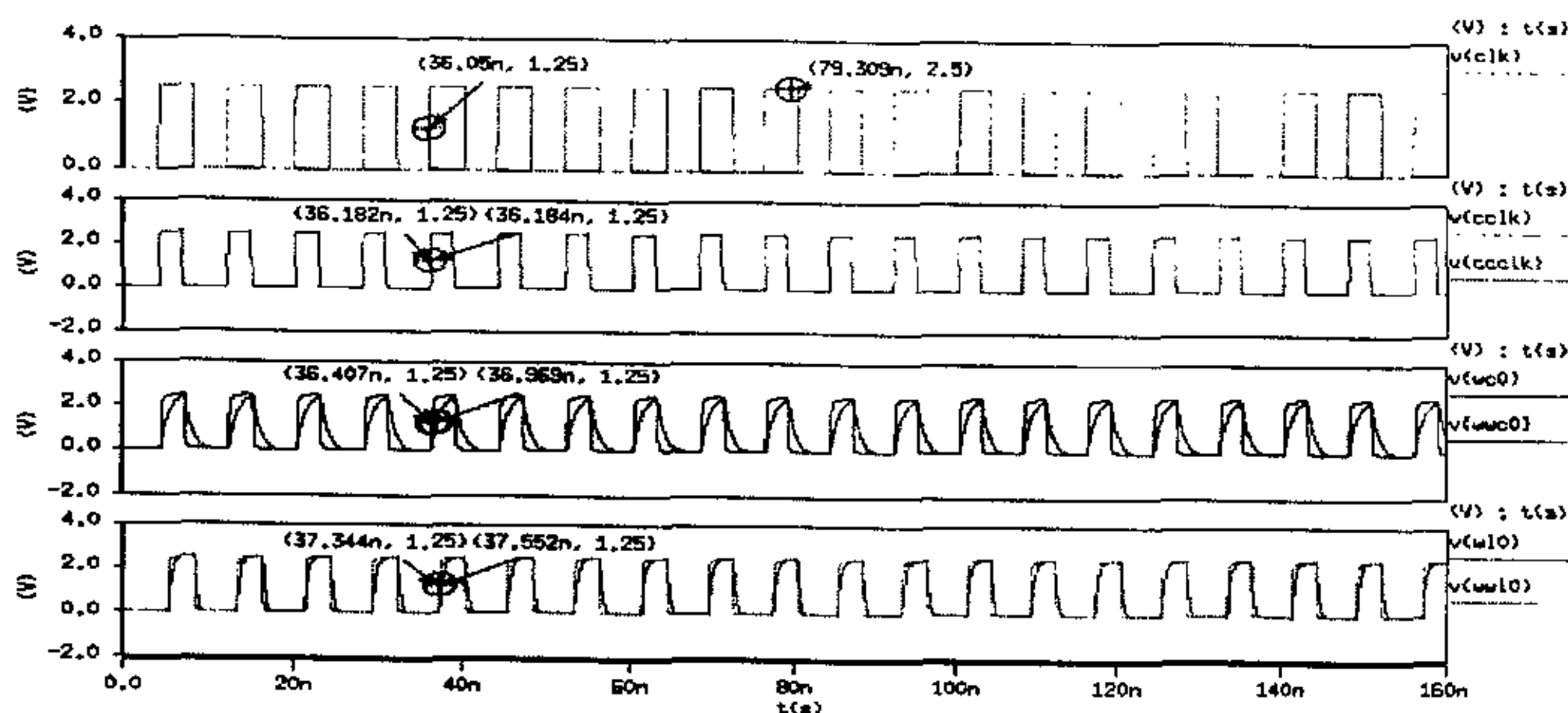
图 5-9 1Mb SRAM 仿真时序图

图 5-10 示出了 1Mb SRAM 地址译码器延迟仿真波形。图 5-11 示出了 1Mb SRAM 四级灵敏放大器延迟工作波形。

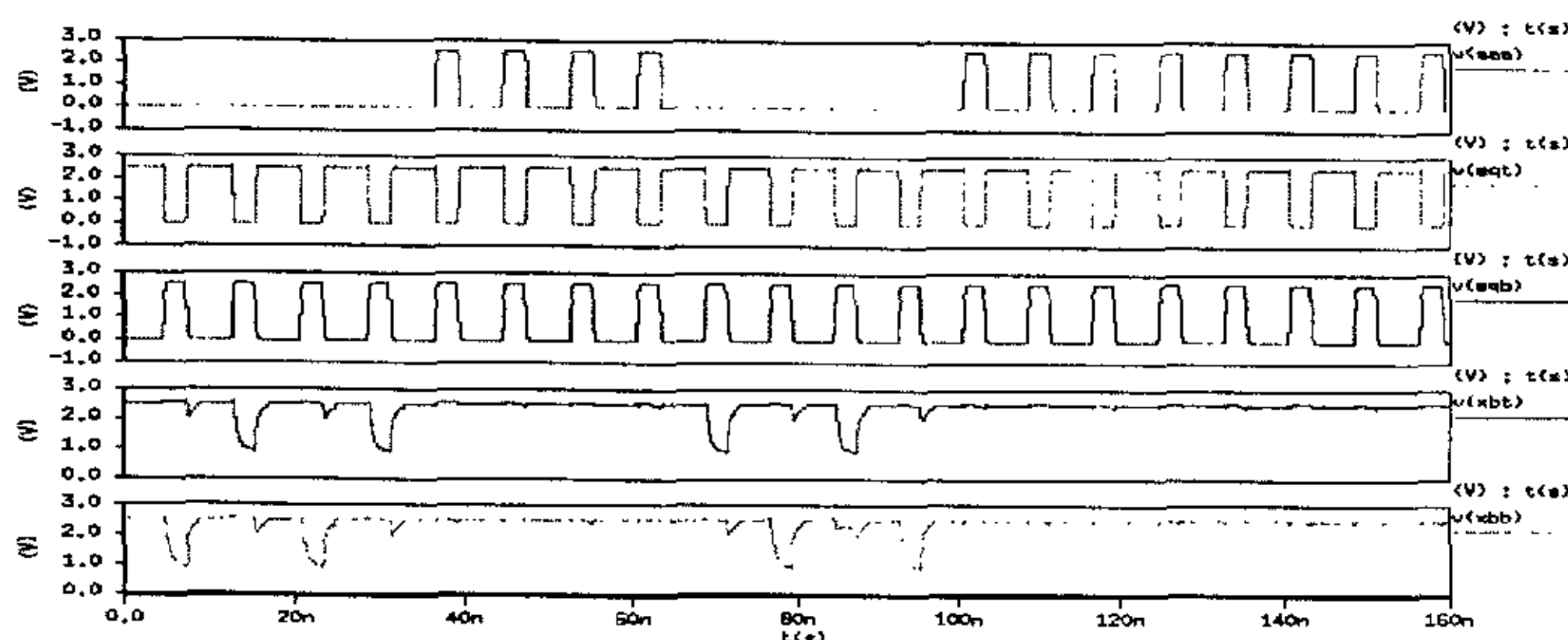




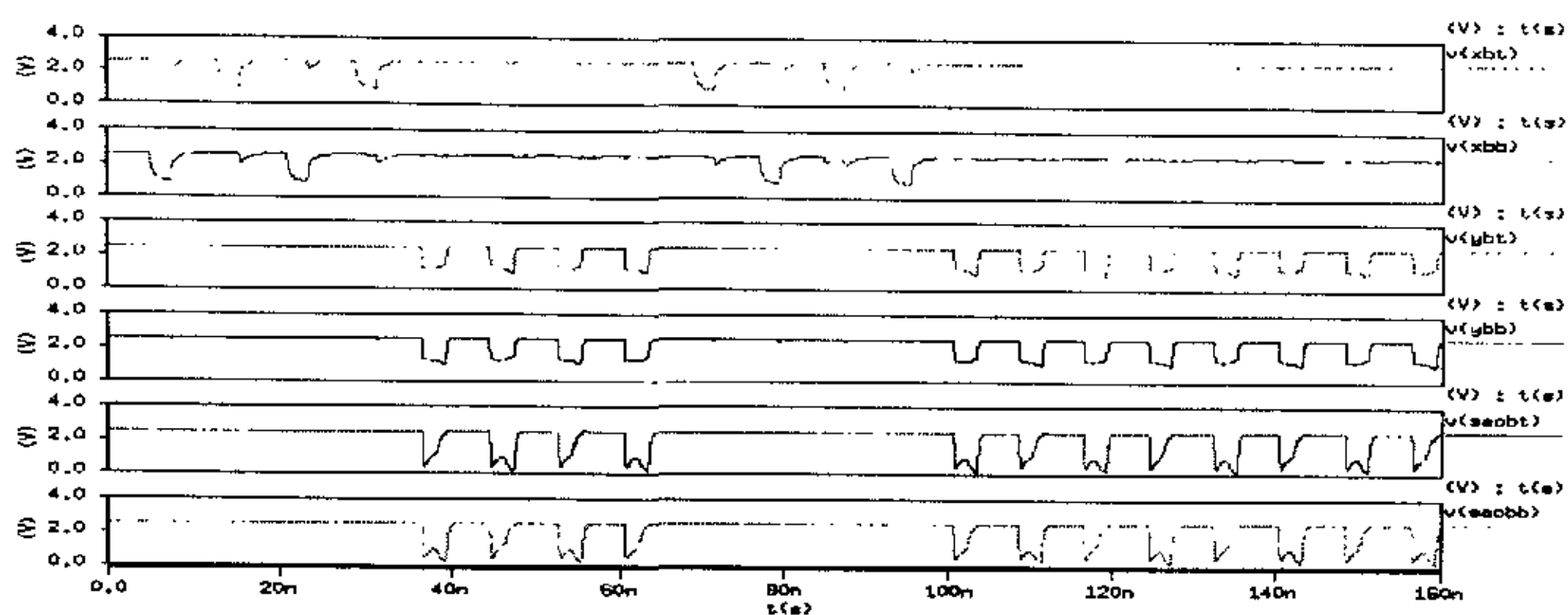
图 5-13 说明了包含在整个 128Kb SRAM 阵列和子模块的关键时序控制中的信号流程。为了更好地分析时序，我们把图 5-8 所示的波形图进一步分解为 5-13(a)~(e) 五个波形图。如图所示，周期开始于产生时序信号的 SRAM 时钟 CLK，然后 SRAM 时钟 CLK 将时序信号传给译码时钟 CCLK，译码时钟再把时钟信号传给字线 WL。相应地，产生的字线信号，使其在存储器阵列字线上产生真值 / 互补的差分电压。伴随于字线信号的产生，译码时钟产生一个依赖于尺寸的延迟，并将它传给控制时钟驱动器。时序延迟信号产生一个依赖于尺寸的延迟灵敏放大时钟信号，这一信号传给灵敏放大器，与此同时位线上出现为正确数据放大所需的较大的真值 / 互补差分电压。图中 XBT 和 XBB 是位线经过由传输门组成的列选择器后的一对信号，YBT, YBB 是 XBT 和 XBB 经过第一级放大后得到的信号，SAOBT 和 SAOBB 是 YBT 和 YBB 经过第二级放大后得到信号。



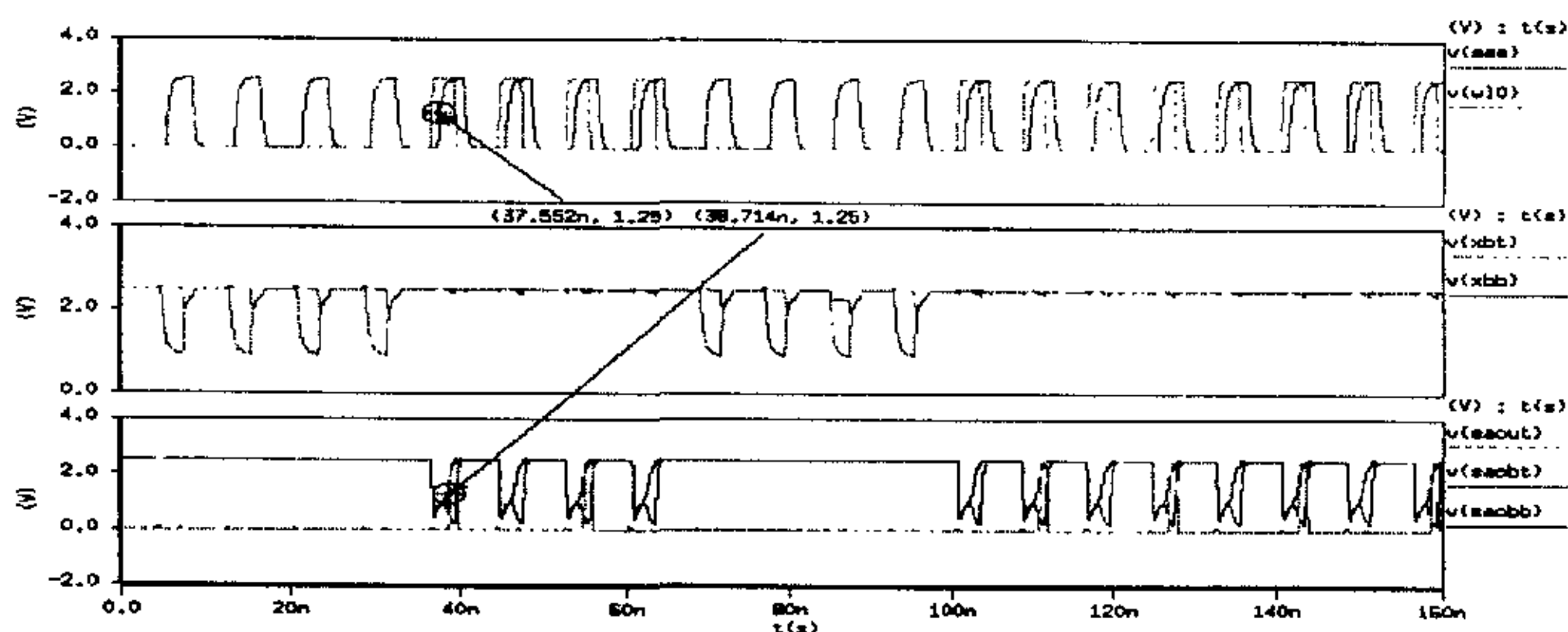
(a) 128Kb SRAM 中内部时钟 cclk 有效到字线 wl0 有效时序图



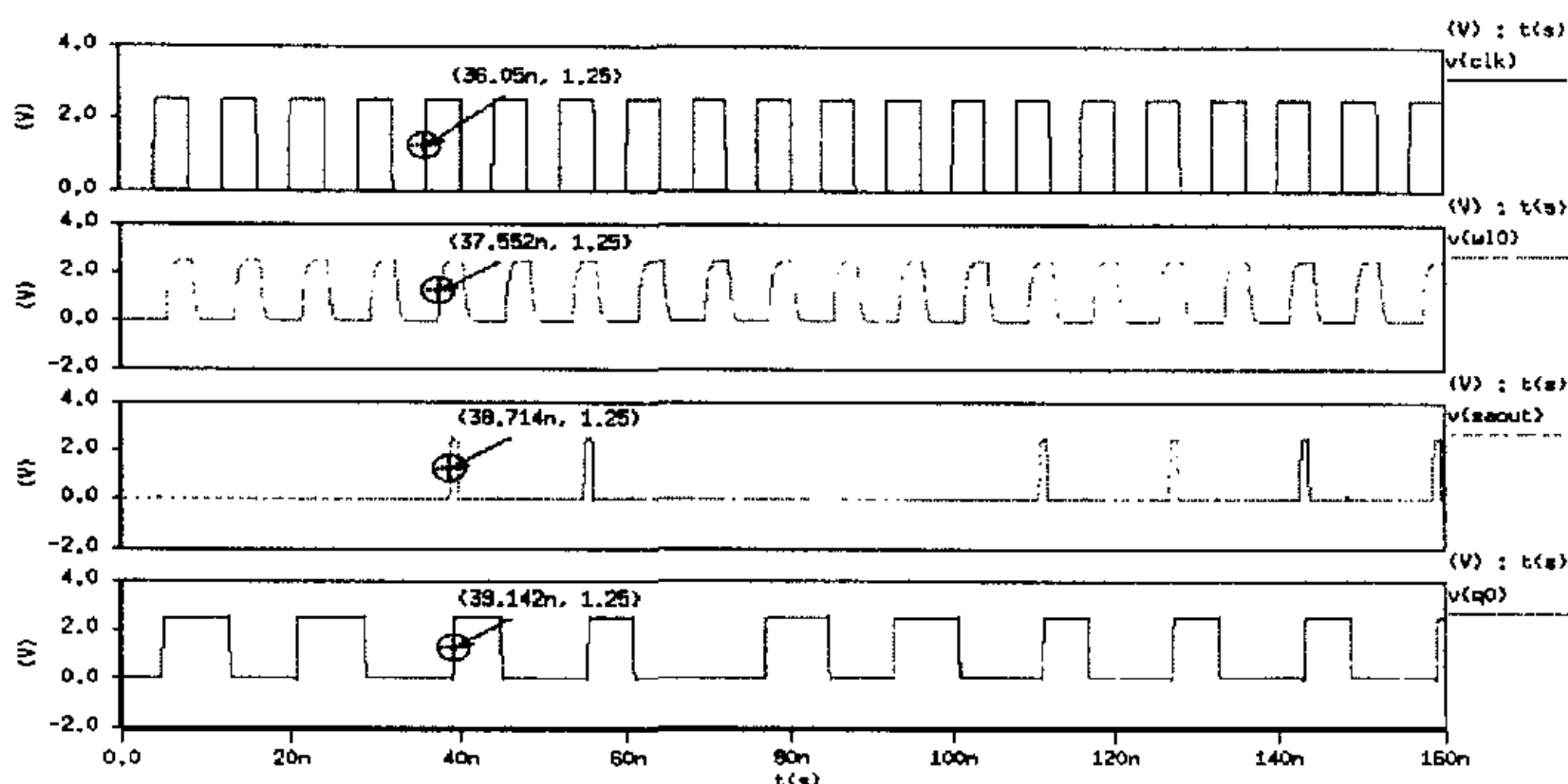
(b) 128Kb SRAM 读 / 写操作过程中灵敏放大器使能信号 sae、本地数据总线平衡信号 eqt、eqb 和本地数据总线信号 xbt、xbb 时序图



(c) 128Kb SRAM 读 / 写操作过程中本地数据总线信号 xbt、xbb、第一级灵敏放大器输出信号 ybt、ybb、第二级灵敏放大器输出信号 saobt、saobb 时序图



(d) 128Kb SRAM 两级灵敏放大器延迟时序图



(e) 128Kb SRAM 取数时间时序图

图 5-13 128Kb SRAM 仿真时序图

图 5-14 示出了 128Kb SRAM 地址译码器延迟仿真波形。图 5-15 示出了 128Kb SRAM 四级灵敏放大器延迟工作波形。

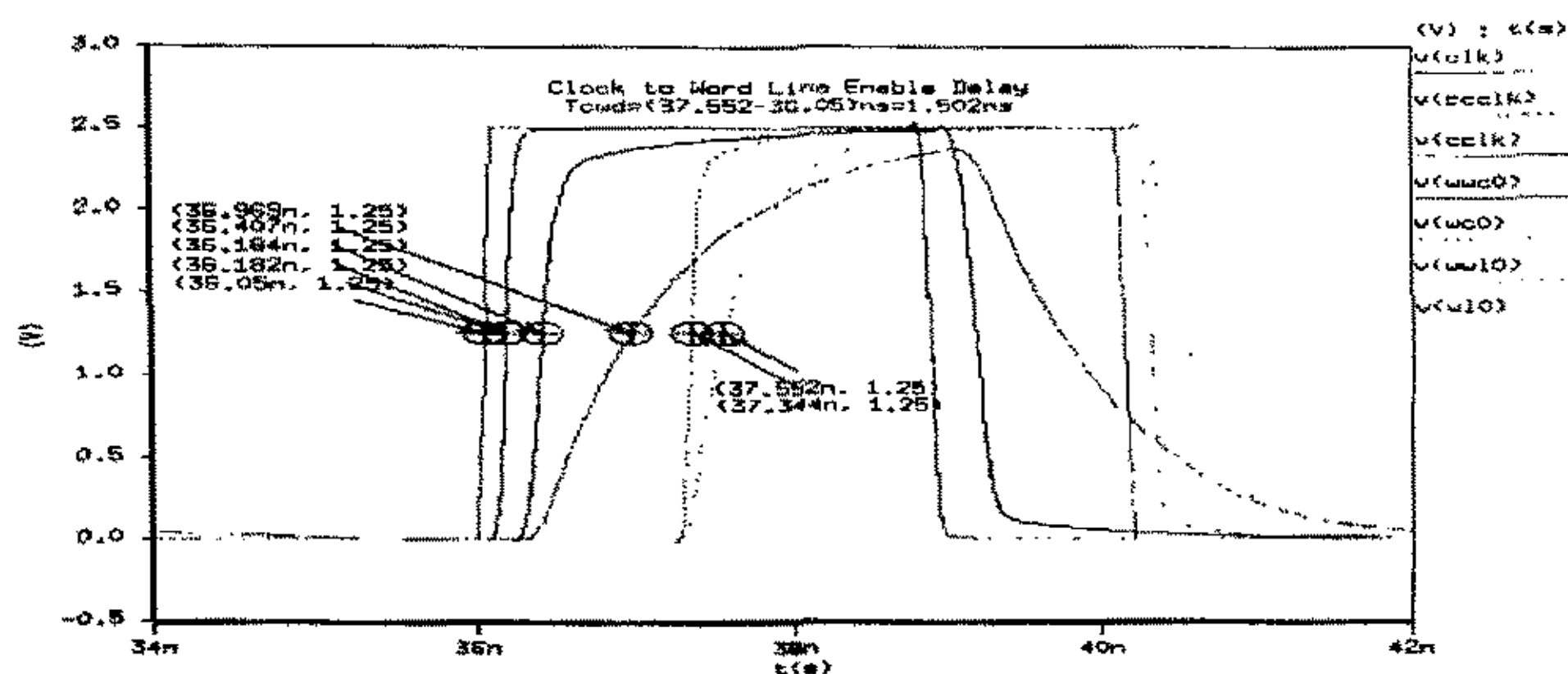


图 5-14 128Kb SRAM 地址译码器延迟仿真波形

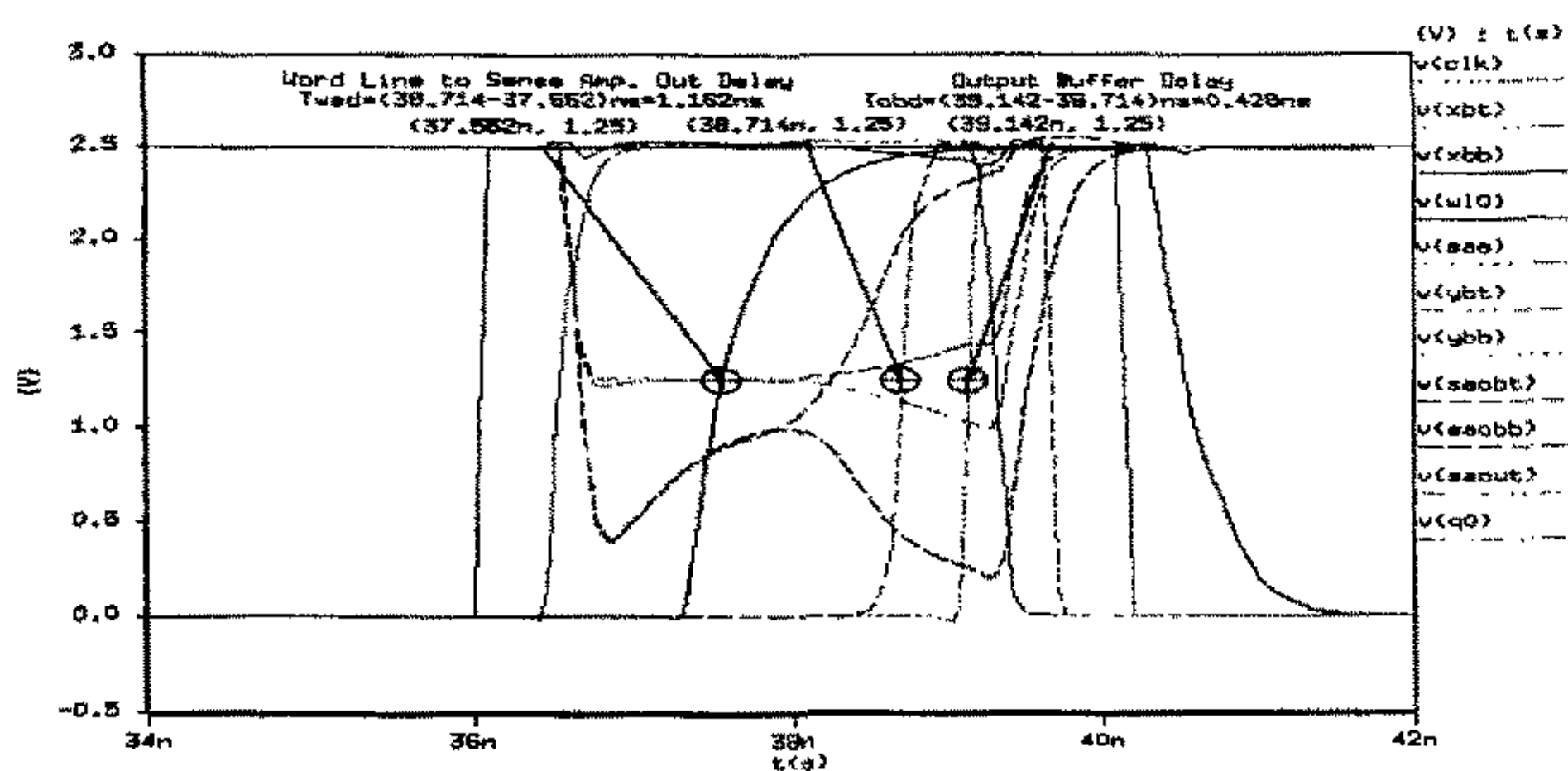


图 5-15 128Kb SRAM 中字线到灵敏放大输出延迟波形图

如图 5-16 所示，电路仿真结果表明设计的 128Kb SRAM 中，译码延迟和灵敏放大器延迟大约分别占总的取数时间的 48.6%和 37.6%。

图 5-16 128Kb SRAM 中取数时间的构成

Xpredecoder, xdecoder		Memory cell sense amp. delay	Output buffer
1502ps/48.6%		1162ps/37.6%	428ps/13.8%
919ps/29.7%	583ps/18.9%		

5.3 SRAM Compiler 版图设计

表 5-8 列出了所设计的两种不同密度 SRAM 的工艺特征和性能参数。

表 5-8 两种不同密度 SRAM 的工艺特征和性能参数

Process technology		0.25 $\mu m$ CMOS
		One-polysilicon
		Five-metal
MOSFET	(NMOS)	0.24 $\mu m$
Gate length	(PMOS)	0.28 $\mu m$
Gate Oxide thickness		5.4 nm
Chip Size		0.61mm $\times$ 4.4mm/4.9mm $\times$ 4.8mm
Memory Cell Size		3.0 $\mu m$ $\times$ 4.2 $\mu m$
Configuration		16Kx8bits/128Kx8bits
Supply Voltage		2.5V
Clock cycle time		7ns/13.6ns
Clock access time		3.092ns/5.971ns
Power dissipation		39.555mW/63.435mW

在 0.25  $\mu m$  工艺下，SRAM 六管存储单元尺寸仅为 3.0  $\mu m$   $\times$  4.2  $\mu m$ ，物理版图见图 5-17。

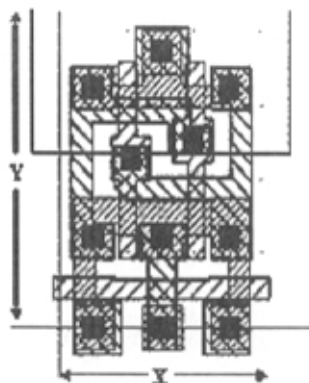


图 5-17 六管存储单元版图

前面已经提到，版图设计中平面规划图（Floor-plan）非常重要。下面的图 5-18 给出了我们所设计的 128Kb SRAM 的平面规划图。

行译码器	行缓冲器	存储单元阵列
地址缓冲器	列译码器	预充电电路
		灵敏放大器电路
	控制逻辑	输出级电路
输入数据分配器		

图 5-18 128Kb SRAM 的平面规划图

如图 5-19 所示，在版图中我们使用了金属字线和多晶硅字线混合的字线结构<sup>[34][35][36]</sup>，每隔八个存储单元打一个“STRAP”，即并联一次，这样做的好处是可以降低字线上的电阻和电容负载，提高读写速度。

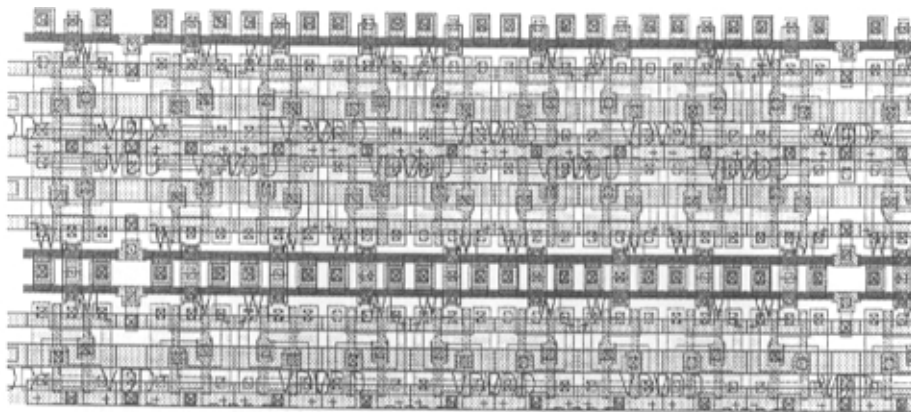


图 5-19 降低字线延迟的方法—金属旁路

为了很容易地用于门阵列环境中，电源环被用来将 SRAM 包裹在其中。图 5-20 示出了带有多个存储器块的外电源环状结构可能的配置。根据存储器的物理高度和宽度，计算出电源环金属宽度。

$V_{ss}$  和  $V_{dd}$  分布在存储器和外围电路之中。它们在平面规划图的上部和底部连接，形成两条主电源线。主电源总线的宽度随着 SRAM 的密度大小而变化。

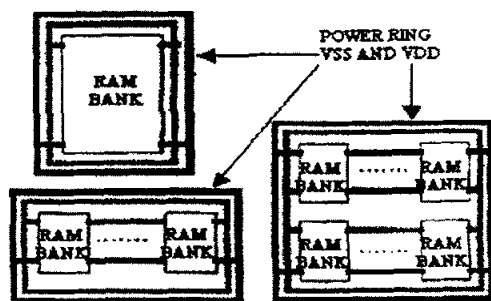


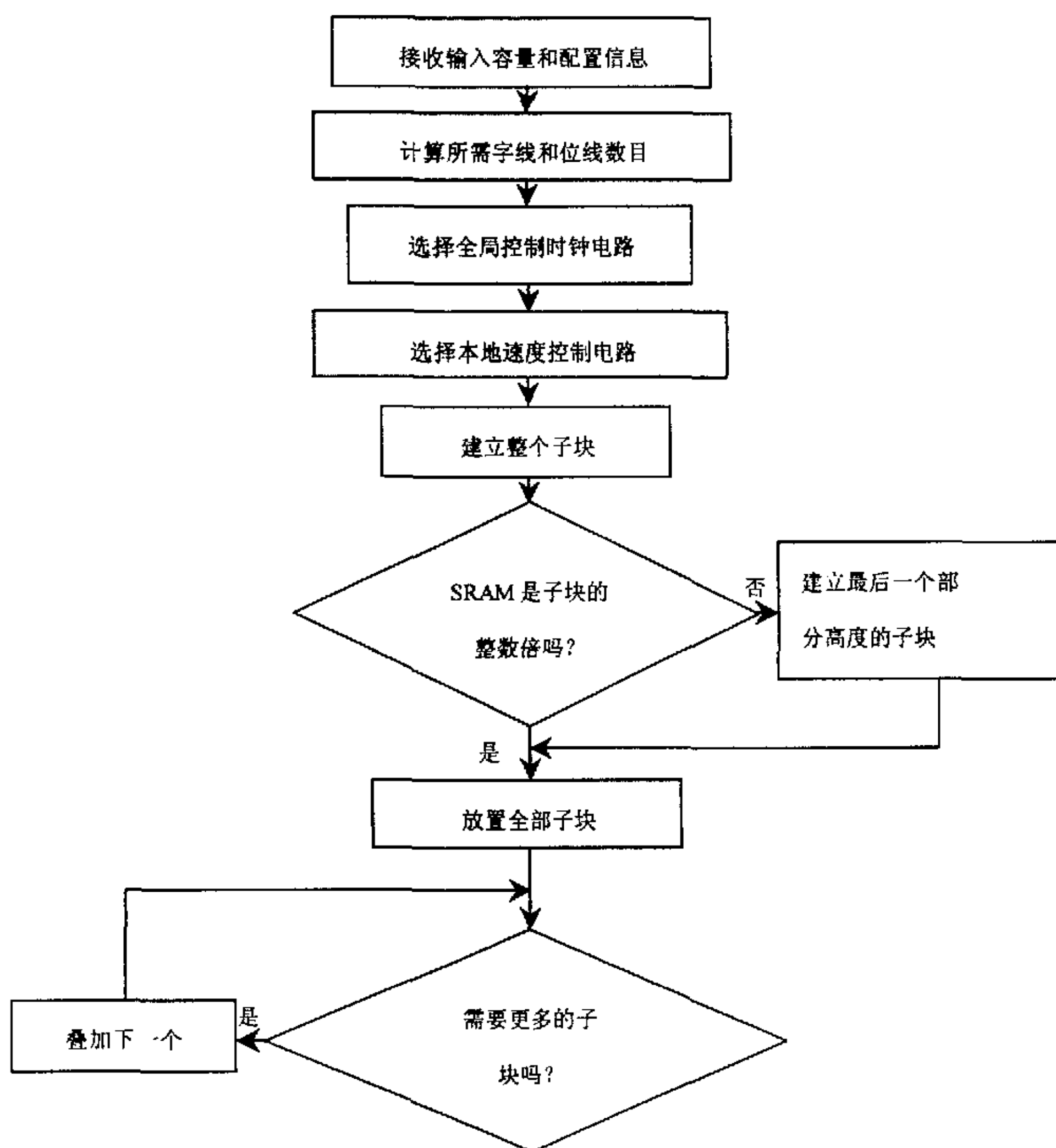
图 5-20 带有一个或多个存储器块的外电源环状结构



## 5.4 SRAM Compiler 实现方法

图 5-21 示出了 Compiler 生成 SRAM 的算法流程图。Compiler 步骤如下：

1. 根据纵横比接近 1 的原则计算出所需的行、列以及分块子模块数；
2. 将所需的子模块堆叠在一起，再把本地控制单元叠加上去；
3. 然后根据总的位线数目将多个子模块堆叠在一起，
4. 把所需的 I/O 连接到存储单元阵列；
5. 产生译码电路版图，叠加块选择电路。
6. 编译器算法会通过将最后的一个子模块放置在恰当的高度来自动的为生成大容量 SRAM 提供特殊的子模块<sup>[37]</sup>。



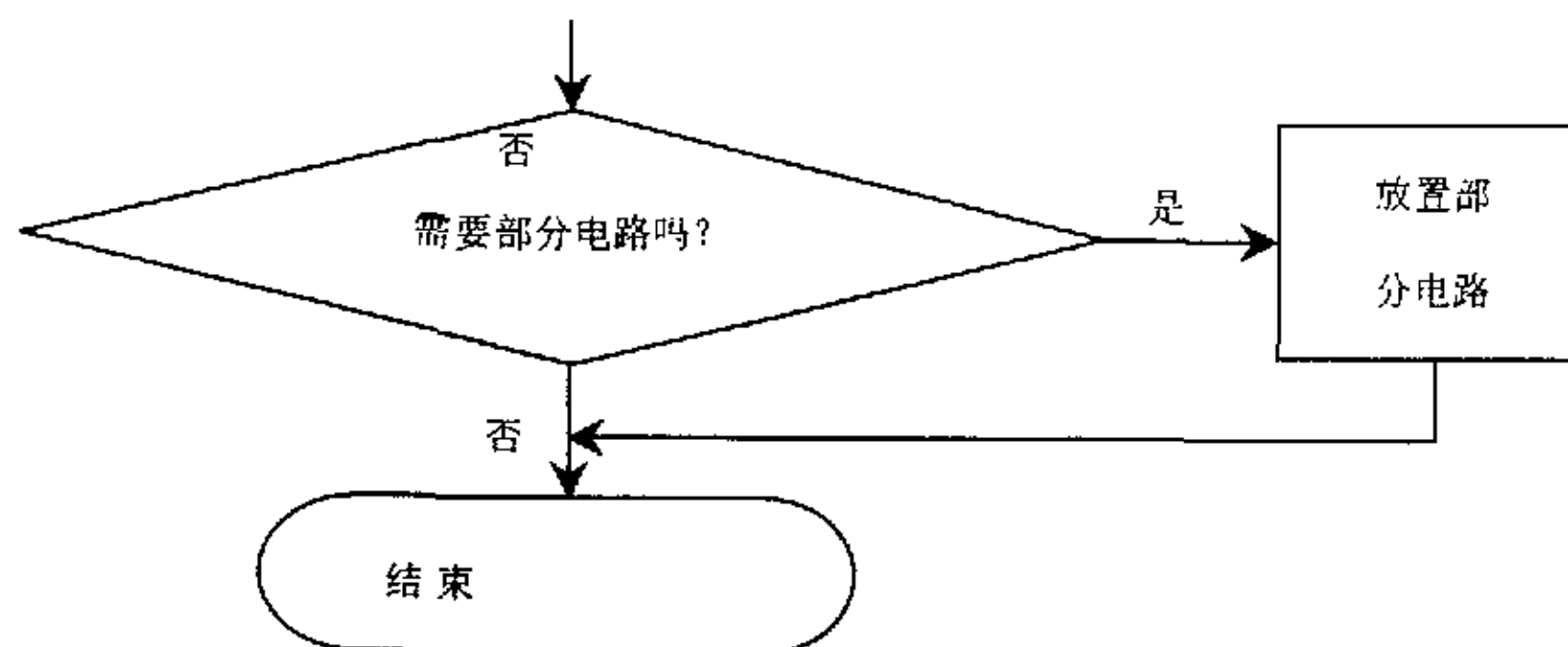


图 5-21 SRAM Compiler 的算法流程图

设计中使用 Cadence 设计环境。分别使用 SKILL 语言和 Virtuoso 作为实现 Compiler 和版图设计工具。其中 1Mb SRAM Compiler 模块的 SKILL 代码实例如下：

```

.....
.....
;; FileName:sram_compiler.il
;; procedure sram_compiler will generate an embedded SRAM layout
;; Usage in CIW:
;; sram_compiler(library cellview Words Wordsize Type)
;; Possible types:
;;      Type = 0 – Simple SRAM array without Array Partitioning
;;      Type = 1 –Array Partitioned SRAM array with the Block Select at
bottom
;; Ex: sram_compiler("sram" "block_128k_8" 16k 8 1)
;;      Create a 128kx8 size SRAM that is partitioned into 8 blocks for low-
power
;;      Layout will be stored in cellview "block_128k_8" under library
"sram"
.....
.....

.....
.....
;;; Load all other necessary files ;;;
.....

```

```

load("sram_array.il")
load("BS_center.il")
load("BS_bottom.il")

.....
;;;;; procedure to layout SRAM ;;;;;
.....
procedure(sram_compiler(clib cname words w type)
  if(type == 0 then
    ;;;; generate a simple SRAM array
    sram_array(clib cname words w)
  else
    ;;;; generate a circuit for 1 block = words/8
    ; create block in a temporary cellview called : "temp_" + cname
    block = int(words/8) ; block size
    blkcvname = buildString (list("temp" pcExprToString (cname)) "_")
    sram_array(clib blkcvname block w)
    ; call array-partitioning function
    array_partition(clib cname words w)
    ;delete temporary block cellview
    ddDeleteObj(ddGetObj (clib blkcvname))
  )

.....
;;;;; Save Cellview before quitting ;;;;
.....
ccv = dbOpenCellviewByType(clib cname "layout" "maskLayout" "a")
dbSave(ccv) ;save cellview
dbClose(ccv) ;close cellview
)

```

## 结 束 语

本文详细论述了 SRAM 的结构组成、工作原理和设计方法，对单块 SRAM 的存储单元和灵敏放大器及地址译码器等外围电路进行了设计和仿真，在此基础上，以用于 ASIC 中的 1Mb SRAM 为例，探讨了 SRAM 的 Compiler 设计技术，给出了相应的 Compiler 算法。设计中为了实现高速和低功耗目标，采用了两极带时钟地址译码电路、两级 / 四级灵敏放大电路和位线 / 本地数据总线 / 全局数据总线平衡等方法。

对所设计的 1Mb 和 128KbSRAM，在 TSMC（台积电）0.25 $\mu\text{m}$  工艺条件下，使用 Hspice 进行模拟，仿真结果表明：设计的 128Kb SRAM 在  $V_{\text{DD}}=2.5\text{V}$ ， $T=25^\circ\text{C}$ ，工艺条件为典型的情况下，取数时间为 3.092ns，在 125MHz 工作频率下功耗为 39.555mW。而采用 Compiler 技术设计的 1Mb SRAM 在  $V_{\text{DD}}=2.5\text{V}$ ， $T=25^\circ\text{C}$ ，工艺条件为典型的情况下，虽然密度增大为原来的八倍，但取数时间仅为 5.971ns，工作频率仍可高达 74MHz，功耗仅为 63.435mW。

综上所述，运用 Compiler 技术设计的 SRAM 在 128Kbit~1Mbit 密度范围内兼容，实现了高速数据读取和低功耗的要求。运用编译方法设计出的 SRAM 已应用于专用集成电路（ASICs）环境中。

## 参 考 文 献

- [1] 尼葛洛庞帝,《数字化生存》,海南出版社,1997.
- [2] 拉贝 (Rabaey, J.M.), 数字集成电路: 设计透视-影印版, 北京: 清华大学出版社, 1998,9.
- [3] 张兴, 黄如, 刘晓彦,《微电子学概论》, 北京大学出版社, 2000,1, pp. 217-218.
- [4] Ashok K. Sharma, Semiconductor Memories, IEEE Press, 1997, pp.10-17.
- [5] M. Jagasivamani, "Development of a Low Power SRAM Compiler", Master Dissertation, Virginia Polytechnic and State University, June 2000.
- [6] Viewdraw Version 7.7.0, a unique portion of eProduct Designer™ Version 1.1, EPD 1.1, Copyright 1984-2000, Innoveda, Inc.
- [7] LEO - A VLSI Layout Editor & Optimizer, Release 2.2 v5 32-bit (SOLARIS) 03/31/2001 Copyright (C) 1999 - 2001 by Silicon Canvas, Inc.
- [8] PowerMill & Timemill Version 5.5, SN: P20000907-SunOS\_5 Copyright (c) 2000 Synopsys Inc.
- [9] Star-HSPICE -- 2002.2 (20020615) linux, Copyright (C) 1985-2002 by Avant! Corporation.
- [10] Cosmos-Scope Version 1.3, Copyright (c) 2001, Avant! Corp. Cosmos-Scope™ provides support for analyzing waveform data.
- [11] Richard C. Jaeger, Microelectronic Circuit Design, The McGraw-Hill Companies, Inc., 1997.
- [12] D. Frohman, FAMOS-A New Semiconductor Charge Storage Device, Solid State Electronics, vol.17, 1974, pp.517-529.
- [13] E. Snow, Fowler-Nordheim Tunneling in SiO<sub>2</sub> Films, Solid State Communications, vol.5, 1967, pp.813-815.
- [14] R. Pashley and S. Lai, Flash memories: The Best of Two Worlds, IEEE Spectrum, December 1989, pp.30-33.
- [15] 财税法规政策摘登,《财务与会计》, 2003,2, pp.76-77.
- [16] John P. Uyemura, CMOS Logic Circuit Design, Kluwer Academic Publishers, Boston, 1999, pp.243-244.

- [17] SOW T. CHU et al., A 25-ns Low-Power Full-CMOS 1-Mbit (128K $\times$ 8) SRAM, IEEE Journal Solid-State Circuits, vol.23, No.5, October 1988, pp.1078-1084.
- [18] Hiroaki Nambu, et al, A 1.8-ns Access, 550-MHz, 4.5-Mb CMOS SRAM, IEEE Journal Solid-State Circuits, vol.33, No.11, November 1998, pp.1650-1658.
- [19] Behzad Razavi, Design of Analog CMOS Integrated Circuits, The McGraw-Hill Company Inc., New York, 2001.
- [20] P.E. 艾伦、D.R.霍尔伯格著，王正华、叶小琳译：《CMOS 模拟电路设计》，科学出版社，1995.
- [21] 李联，《MOS 运算放大器—原理、设计与应用》，复旦大学出版社,1990,5.
- [22] 刘新宇，韩郑生等，CMOS/SOI 64Kb 静态随机存储器，半导体学报，Vol.22, No.1, January 2001.
- [23] James S. Caravella, A Low Voltage SRAM For Embedded Applications, IEEE Journal Solid-State Circuits, vol.32, No.3, March 1997, pp.428-432.
- [24] S. Yamamoto, et al, A 256Kb SRAM with Variable Impedance Data-Line Loads, IEEE Journal Solid-State Circuits, vol. SC-20, No.5, October 1985, pp.924-928.
- [25] R.C. Jaeger, Comments on 'An optimized output stage for MOS integrated circuits', IEEE Journal Solid-State Circuits, vol.SC-10, No.3, March 1992, pp.185-186.
- [26] N.C. Li, et al., CMOS Tapered Buffer, IEEE Journal Solid-State Circuits, vol.25, No.4, August 1990, pp.1005-1008.
- [27] S.R. Vemuru et al., Variable-Taper CMOS Buffer, IEEE Journal Solid-State Circuits, vol.26, No.9, September 1991, pp.1265-1269.
- [28] Allan L. Silburt et al., A 180-MHz 0.8- $\mu$ m BiCMOS Modular Memory Family of DRAM and Multiport SRAM, IEEE Journal Solid-State Circuits, vol.28, No.3, March 1993, pp. 222-232.
- [29] Jarvis C. Tou et al., A Submicrometer CMOS Embedded SRAM Compiler, IEEE Journal Solid-State Circuits, vol.27, No.3, March 1992, pp.417-424.
- [30] T. Hirose, A 20-ns 4-Mb CMOS SRAM with Hierarchical Word Decoding Architecture, IEEE Journal Solid-State Circuits, vol.25, No.5, October 1990, pp.1068-1074.
- [31] Masahiko Yoshimoto, et al, A Divided Word-Line Structure in the Static RAM and Its Application to a 64K Full CMOS RAM, IEEE Journal Solid-State Circuits, vol.18, No.5, October 1983, pp.479-485.
- [32] T. Wada, An Analytical Access Time Model for On-Chip Cache Memories, IEEE Journal Solid-State Circuits, vol.27, No.8, August 1992, pp.1147-1156.

- [33] Kang, Sung Mo, et al, CMOS Digital Integrated Circuits: Analysis Design, The McGraw-Hill Company Inc.New York, 1999.
- [34] Hiroshi Shimada, Yoshinao Tange, et al. An 18-ns 1-Mbit CMOS SRAM, IEEE Journal Solid-State Circuits, vol.23, No.5, October 1988, pp.1073-1077.
- [35] Katsuro Sasaki, Koichiro Ishibashi, et al, A 9-ns 1-Mbit CMOS SRAM, IEEE Journal Solid-State Circuits, vol.24, No.5, October 1989, pp.1219-1225.
- [36] Teruo Seki, Eisaku Itoh, et al, A 6-ns 1-Mb CMOS SRAM with Latched Sense Amplifier, IEEE Journal Solid-State Circuits, vol.28, No.4, March 1997, pp.428-432.
- [37] Jeffery H. Oppold et al, Performance Optimizing Compiler For Building A Compiled SRAM, United States Patent, Patent No. 6,002,633, Dec.14, 1999.