

## 摘 要

随着移动互联网的迅速发展，SOC 作为便携式智能终端的信息处理能力和待机时间的要求也日趋提高。存储器作为 SOC 的重要组成，一般而言面积占整个 SOC 芯片面积的一半以上。尤其在高性能处理器中，SRAM 占芯片的面积已经超过 80%。存储器的性能对 SOC 芯片的影响很大。

如今的 SRAM 设计中，越来越多的低功耗技术和高速技术被提出，比如 DVFS 技术、电荷泵技术等。在诸多优化技术带来诸多便利的同时，SRAM 的设计周期相应也延长了，尤其在仿真环节中，相比于原本就耗时的 SRAM 设计仿真周期而言，这些优化技术无疑也增加了仿真周期和设计周期。

本文主要内容和研究对象是：第一，介绍机器学习，详细阐述了最小二乘法回归算法；第二，介绍了标准单元库，分析了十个标准单元的时序，通过电路参数建模实现时序预测和通过时序建模实现时序预测；第三，介绍 SRAM 的电路结构、基本原理和特征化，主要对表征中的时序和功耗进行分析，并通过时序建模实现时序预测、按温度划分功耗实现功耗预测，分析并指出本方法在 SRAM 设计周期中对于设计周期和仿真周期的提升效果；第四，对本文的工作进行了总结，并对未来的工作进行了展望。

关键词：静态随机存取存储器 最小二乘法 编译器 表征 预测

作者：余一奇

指导教师：张立军

## Abstract

With the rapid development of mobile Internet, the information processing capacity and standby time requirement of SOC as a portable intelligent terminal is also increasing. As an important part of SOC, memory accounts for more than half of the total SOC chip area. Especially in high-performance processors, SRAM accounts for over 80% of the chip area. The performance of memory has a great influence on the SOC chip.

Nowadays, more and more low-power technologies and high-speed technologies have been put forward in SRAM design, such as DVFS technology, charge pump technology and so on. While many optimization technologies have brought many conveniences, the design cycle of SRAM has been extended correspondingly, especially in the simulation link. Compared to the time consuming SRAM design simulation cycle, these optimization techniques undoubtedly increase the simulation cycle and design cycle.

The main content and research object of this paper are: Firstly, this paper introduced machine learning, and described the least square regression algorithm in detail. Part two introduced the standard cell library, and analyzed the timing of ten standard cells, and realized sequential prediction and through time series modeling to realize time series prediction through the modeling of circuit parameters. Then this paper introduced the circuit structure, basic principle and characteristic of SRAM. In this part, the time series and power consumption in the characterization are analyzed. The time series prediction was realized and the power consumption was divided according to the temperature division. Finally, the improvement of the design cycle and the simulation cycle in the SRAM design cycle is also pointed out. The last part summarized the work of this paper and forecasted the future work.

Keywords: Static Random Access Memory, Least Squares, Compiler, Characterization, Prediction.

Written by She Yiqi

Supervised by Zhang Lijun

# 目录

第一章 绪论 .....	1
1.1 半导体集成电路介绍 .....	1
1.1.1 存储器及 SRAM 介绍 .....	2
1.1.2 SRAM 设计优化技术 .....	3
1.2 SRAM 仿真工作存在的优化可能 .....	4
1.3 论文的主要内容 .....	5
第二章 最小二乘法 .....	6
2.1 机器学习 .....	6
2.1.1 机器学习的定义 .....	6
2.1.2 机器学习算法分类 .....	6
2.2 最小二乘法回归 .....	7
2.2.1 线性回归模型 .....	7
2.2.2 最小二乘法 .....	7
2.2.3 算法实现 .....	9
2.3 本章小结 .....	9
第三章 28nm 标准单元库时序预测 .....	10
3.1 标准单元库 .....	10
3.1.1 标准单元方法 .....	10
3.1.2 标准单元库 .....	11
3.1.3 标准单元库流程 .....	12
3.2 时序 .....	14
3.2.1 Liberty 库 .....	14
3.2.2 数据收集 .....	16
3.3 时序预测 .....	19
3.3.1 电路参数方法 .....	20
3.3.2 时序方法 .....	22
3.4 本章小结 .....	23
第四章 SRAM 编译器 .....	24
4.1 SRAM 基本原理 .....	24
4.1.1 SRAM 架构 .....	24
4.1.2 SRAM 存储单元 .....	25
4.1.2 SRAM Compiler .....	26
4.2 存储器表征 .....	30
4.2.1 特征 .....	30
4.2.2 实例 .....	32
4.2.3 输入压摆和输出负载 .....	33
4.2.4 工艺角 .....	34
4.3 本章小结 .....	36
第五章 SRAM 表征数据预测 .....	37
5.1 时序 .....	37
5.1.1 时序数据 .....	37
5.1.2 时序预测 .....	43
5.2 功率 .....	52
5.2.1 功率数据 .....	52
5.2.2 功率预测 .....	53
5.3 本章小结 .....	59
第六章 总结与展望 .....	60

6.1 论文总结.....	60
6.2 工作展望.....	60
参考文献.....	62
攻读学位期间本人出版或公开发表的论著、论文 .....	66
致 谢 .....	67

# 第一章 绪论

## 1.1 半导体集成电路介绍

随着物联网和移动互联网的迅捷发展，集成电路芯片被广泛应用于各种与日常生活密切相关的领域，比如个人电脑，智能手机，可穿戴设备等。其中手机的出货量已经超过 71 亿部。正是六十多年来集成电路的高速发展为基础，才有如今辉煌的电子信息产业链[1]。

2017 年全球集成电路芯片市场的销售额已经达到 4000 亿美元，与之相关的电子信息产业的产值达到了 15 万亿美元，已经超过 2017 年全球 GDP 的五分之一。

国际半导体产业协会（SEMI）发布的各家产业研究机构对 2018 年半导体产业营收的预测成长率为 7~8%。在全世界集成电路行业高度竞争和发展中，我国的集成电路行业也有着很好的表现。其中，中国在智能手机为代表的通信模块方面和移动终端的 AP 方面已占据全球市场份额的 17% 左右。然而，在集成电路芯片应用的其他领域中，国产芯片所占有市场份额几乎没有。从国内的集成电路芯片发展的规模来看，我国集成电路芯片产业正在进入一个快速的发展期，近年来保持着近 20% 的增长率并持续增长，而且在 2017 年集成电路产业的产值已达到 5000 多亿[2]。即使在这种情况下，国内的集成电路芯片才只有 30% 左右能满足自身的需求，其余均依赖于进口，集成电路芯片的进口额已经超过了石油，成为了我国的需求量第一的进口商品，这种现状亟待依赖于我国集成电路行业的迅速发展来解决[3]。从集成电路行业与其密切影响的电子信息行业的发展来看，不论是中国还是全球，发展集成电路行业都具有非常重要的意义。

### 1.1.1 存储器及 SRAM 介绍

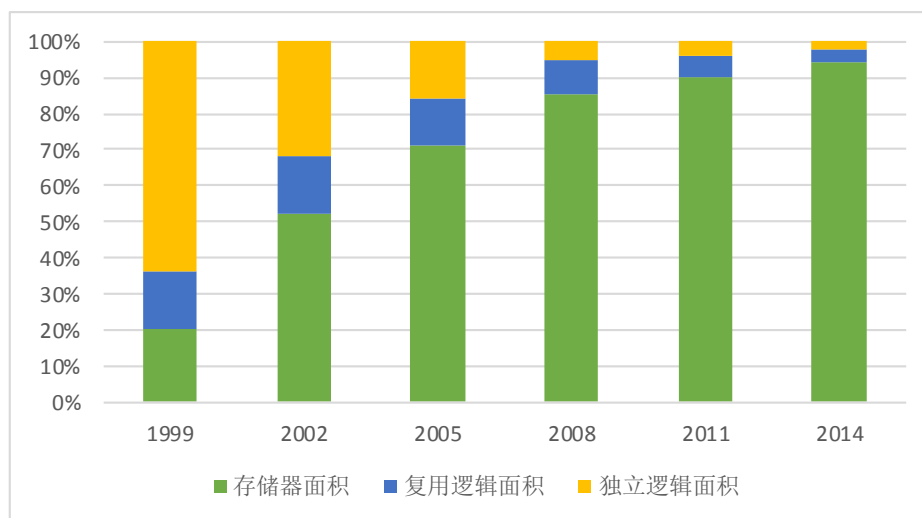


图 1-1 近年来 SOC 中存储器的使用情况

存储器作为现在超大规模集成电路（Very Large Scale Integration Circuit, VLSI）中非常重要的组成部分，随着近几年来半导体工艺技术的迅速发展，存储器在 SOC 中占据着越来越重要的比重，2014 年的存储器占有 SOC 的面积已经达到了 94%。与此同时，SRAM 作为存储器家族的重要成员，也具有十分重要的意义。

存储器家族的成员繁多，通常都有多种分类方式。通常较为主流的分类方式是按照存取方式的不同进行分类，主要分为随机存取存储器（Random Access Memory, RAM）、顺序存取存储器（Serial Access Memory）和按内容寻址存储器（Content Access Memory）。

其中，随机存取存储器还可以根据断电后是否能够继续保存信息而分类，主要分为易失性存储器（Volatile Random Access Memory, VM）和非易失性存储器（Non-Volatile Random Access Memory, NVM）两类。通常易失性存储器又分为静态随机存取存储器（Static Random Access Memory, SRAM）和动态随机存取存储器（Dynamic Random Access Memory, DRAM）。

如今 SOC 处理器的主频已经达到 4GHZ，其片外的大容量主存储器的频率只有 600MHZ 左右[4]。SRAM 凭借面积小，容量合适，速度快和很好的鲁棒性等优势，作为 SOC 芯片的存储设备。SRAM 的频率可以达到逻辑电路的频率，也就是说完全可以满足处理器的主频需求。SRAM 在待机（Standby）的状态下功耗较低，具有更高的带宽，可以满足系统对带宽的需求[2]。SRAM 还拥有更好的工艺缩放性，可以随着工艺节点的缩小得到更好的集成度和性能。SRAM 在拥有这些其他存储器不可比拟的

优势的同时，也意味着 SRAM 具备的设计难度和要求。

### 1.1.2 SRAM 设计优化技术

戈登·摩尔在 1973 年曾经提出了，一个硅片上晶体管的数量每隔 18 个月就会翻一番，后来这一推测成为了著名的摩尔定律[5]。在过去集成电路迅速发展的几十年中，集成电路的迭代更新一直印证着摩尔定律。平均一颗芯片上的晶体管数量一直保持大约每两年翻一番。然而随着 SOC 中晶体管数量的剧增，能耗问题、散热问题成为了 SRAM 设计的首要难题。其中，低功耗技术已经成为 SRAM 设计的首要考虑因素。

目前工业界常用的低功耗设计非常多，主要有电源关断（Power Shut-off）、门控时钟（Clock Gating）、多阈值电压（Multi-V<sub>th</sub>）、多电源电压（Multi-Supply Voltage）、动态电压频率调节（DVFS, Dynamic Voltage and Frequency Scaling）、衬底电压偏置（Substrate Biasing）等。

其中，电源关断技术主要通过使能信号去控制一块区域的电源，去实现降低芯片整体的泄漏功耗。当使能信号处于无效状态的时候，控制的整个区域均不供电，因此不产生泄漏功耗。然而需要注意的是，电源关断方法在实现有效降低泄漏功耗的同时，需要引入更多的开关电路和控制电路，同时在电源系统中增加控制电路可能会让使能信号在开关的瞬间对附近区域的逻辑器件产生电压波动。因此考虑使用电路关断技术的同时，需要仔细考虑电源关断技术对电路产生的影响和变化进行评估。

相对于直接关断部分区域的电源的电源关断技术而言，门控时钟设计主要通过使能信号去控制时钟的传播，去实现降低芯片工作时的动态功耗。同样的，门控时钟技术也需要像电源关断技术一样的需要引入新的设计单元。门控时钟技术为电路带来了更多的信号控制的寄存器数量，与此同时控制层次也相应越高。门控时钟也是较为常用的降低动态功耗的设计手段。

多阈值电压技术则主要通过使用不同阈值电压的器件，来实现芯片泄漏功耗的降低。

多电源电压技术主要根据芯片的不同区域的电压需求，选择不同的电源电压值，从而实现降低动态功耗与静态功耗。多电源电压技术是静态的电压调整方法，不同的电源电压代表着不同的电路工作功耗和性能。

多电源电压、动态电压调节（Dynamic Voltage Scaling, DVS）以及动态电压频率

调节技术均是通过调节电源电压来实现低功耗。

动态电压频率调整技术 (DVFS) 是根据芯片处于不同工作场景时, 所需要运行的应用程序对计算能力的不同需要, 动态的调整芯片的不同模块的工作频率和工作电压, 从而达到降低功耗的目的[11]。

DVFS 和 DVS 都是具有自适应性的技术, 它们通过统计处理器的利用率来进行工作: 当处理器的负载较小的时候, 会降低工作频率和工作电压; 当处理器处于额定负荷工作状态时, 系统就会处于最大的工作频率和工作电压条件下。这些低功耗技术已被广泛应用于商用微处理器中, 并有着显著的节能特性[6-10]。

文献 12 中 3D 图形处理中有三个独立的模块, 每个模块有不同的性能需求, 将不同的三个模块应用 DVFS 技术可以实现每个功率区域的高效的能耗减少。最终, GPU 系统产生相同输出结果的同时, 应用 DVFS 技术的电源管理比正常工作的电源管理实现节省功率约 65%。

## 1.2 SRAM 仿真工作存在的优化可能

在各种层次的低功耗技术广泛应用在 SRAM 设计的时候, 在带来显著减少功耗的同时, 也给 SRAM 设计提出了相应的附属要求, 以 DVFS 为例, 不同模块动态调整工作电压和工作频率, 对于设计的要求, 尤其是仿真的要求提高了很多。通常 SRAM 编译器都有几千个实例, 对于某个实例都要在不同的 PVT (Process、Voltage、Temperature) 进行仿真。DVFS 使得 SOC 的电压节点增加了很多。以台积电 28nm HPCPLUS 为例, 标准电压是 0.9v, 相对应的 SRAM 的签出需要 6 个 PVT 的仿真。电压节点从 0.72v 到 1.08v 时, 以 90mv 为间隔, 一共需要 30 个 corner 的签出。

以 50 个 Intel (R) Xeon® 的 4 核 CPU, 其频率为 3.60GHZ, 以及足够的 EDA licenses, Memory Compiler 的表征仿真需要约 340 小时, 基于机器资源和时间成本的考虑, 以及更多优化技术的加入, 未来 SRAM Compiler 表征所需的仿真将会呈现阶梯式的增长。本课题的出发点就是提出利用机器学习算法, 通过已知的仿真数据构建模型, 去预测未知的仿真数据, 从而实现节省机器资源和减少时间成本的目的, 进而优化整个 SRAM Compiler 的设计效率。



### 1.3 论文的主要内容

本文主要基于当今微电子行业对于机器学习的应用现状与前景，总结了机器学习在存储器设计中应用的可行性与可靠性，确立了本文的课题：基于机器学习的 28nm 工艺 SRAM Compiler 特征化。围绕这一主题，基于 SRAM 编译器的设计现状，针对仿真过程的耗时长、占用资源比重大等不足，提出了应用机器学习算法简化仿真，并先后在标准单元库和 SRAM Compiler 的设计中应用，以期对未来的机器学习应用于集成电路设计的重要环节中有一定的指导意义。

本文的主要内容如下：

第一章为绪论，主要介绍了 SRAM Compiler 的研究背景、意义，系统阐述了机器学习应用于 SRAM Compiler 的应用现状和发展前景，提出了 SRAM Compiler 仿真工作方面存在的耗时长、资源占比大等问题，并针对这一问题提出了优化方法。

第二章首先简单的介绍了机器学习及其发展现状与应用前景，主要阐述了回归算法，着重分析了最小二乘法的算法原理和应用场景。

第三章介绍了标准单元库及标准单元时序，并通过最小二乘算法，拟合已有的时序数据，预测未知的时序数据。

第四章介绍了 SRAM Compiler 的设计流程，着重描述了表征的三个模块：时序、功耗和电容，并对三个模块分别使用最小二乘法进行数据的预测。

第五章为总结与展望，对本文的工作进行了总结，并对本课题的研究进行了展望。

## 第二章 最小二乘法

### 2.1 机器学习

#### 2.1.1 机器学习的定义

机器学习 (Machine Learning) 是一门人工智能的学科, 机器学习主要讨论的内容是在“从有限观察发现观测数据中暗含的各种关系的数据分析”和“从有限观察概括特定问题世界模型的机器学习”的方法。机器学习领域的主要研究对象是人工智能。其研究热点是在经验学习中改进具体算法的性能。

机器学习主要应用于语音和手写识别、数据挖掘、生物特征识别、计算机视觉、医学诊断、自然语言处理、医学诊断、生物特征识别、搜索引擎、检测信用卡欺诈、证券市场分析、DNA 序列测序、战略游戏和机器运用等领域。

#### 2.1.2 机器学习算法分类

机器学习有很多算法, 分类方式也有多种。主要的分类方式有基于学习策略的分类、按应用领域分类、基于所获取知识的表示形式分类、学习形式分类和综合分类等。其中最为普遍接受的是按学习形式分类。

通常的, 机器学习按学习形式分类: 监督学习 (Supervised Learning)、非监督学习 (Unsupervised Learning) 和强化学习等几种学习形式。具有应用场景较多的是监督学习和非监督学习。

监督学习主要利用一组已知类型的训练数据样本, 通过对样本数据的特征 (Feature) 和已知目标 (Target) 来推断一个功能的机器学习任务, 并在未知的输入样本时也给出正确的输出结果, 从而达到预测未知的目的; 非监督学习是指数据样本中没有给出正确的输出结果信息。无监督学习希望从数据中挖掘的正是这一类信息, 常见的例子有关联规则挖掘、聚类、离群点检测等等; 强化学习是智能系统从环境到行为映射的学习, 以使奖励信号 (强化信号) 函数值最大, 强化学习不同于连接主义学习中的监督学习, 主要表现在强化学习中由环境提供的强化信号是对产生动作的好坏作一种评价 (通常为标量信号)。

其中根据输出结果是离散还是连续, 监督学习可以分为分类问题和回归问题两大

类。他们在文字、语音、图像识别、垃圾邮件分类与拦截、网页检索、股票预测等方面有着广泛应用。本课题使用监督学习中的回归算法：最小二乘法。回归分析中，如果只包括一个自变量和一个因变量，且二者的关系可用一条直线近似表示，这种回归分析称为一元线性回归分析。如果回归分析中包括两个或两个以上的自变量，且因变量和自变量之间是线性关系，则称为多元线性回归分析。

## 2.2 最小二乘法回归

### 2.2.1 线性回归模型

考虑到本课题中数据的多维性，这里用多元回归方程表示：

$$h_{\theta}(x) = \theta_0 + \theta_1 X_1 + \theta_2 X_2 + \cdots + \theta_n X_n \quad (2.1)$$

式中： $\theta_i$ 称为假设模型即映射输入特征 $x$ 与结果 $y$ 的线性函数 $h$ 的参数(Parameters)，为了简化表示，我们在输入特征中加入 $X_0=1$ ，于是得到：

$$h(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x \quad (2.2)$$

参数 $\theta$ 和输入特征 $x$ 都为矢量， $n$ 是输入的特征 $x$ 的个数（不包含 $x_0$ ）。

使得预测值 $h(x)$ 尽可能接近 $y$ ，为了达到这个目的，我们对于每一个参数 $\theta$ ，定义一个代价函数（Cost Function）用来描述 $h(x^{(i)})$ 与对应的 $y^{(i)}$ 的接近程度：

$$J(\theta) = \frac{1}{2} \sum_{i=0}^m (h_0(x^{(i)}) - y^i)^2 \quad (2.3)$$

于是我们的目标就变为了调整 $\theta$ 使得代价函数 $J(\theta)$ 取得最小值，方法有梯度下降法，最小二乘法等。

### 2.2.2 最小二乘法

最小二乘法通过最小化误差的平方和寻找数据的最佳函数匹配。利用最小二乘法可以简便地求得未知的数据，并使得这些求得的数据与实际数据之间误差的平方和为最小。最小二乘法还可用于曲线拟合。其他一些优化问题也可通过最小二乘法来表达。最小二乘法是一种直观的直接利用矩阵运算可以得到 $\theta$ 值的算法。

假设函数 $f$ 是将 $m*n$ 维矩阵映射为一个实数的运算，即

$$\text{Function } f : \mathbb{R}^{m \times n} \mapsto \mathbb{R} \quad (2.4)$$

并且定义对于矩阵 $A$ ，映射 $f(A)$ 对 $A$ 的梯度为：

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix} \quad (2.5)$$

因此该梯度为  $m \times n$  的矩阵。

另外，对于矩阵的迹的梯度运算，有如下规则：

$$\nabla_A \text{tr} AB = B^T \quad (2.6)$$

$$\nabla_{A^T} f(A) = (\nabla_A f(A))^T \quad (2.7)$$

$$\nabla_A \text{tr} ABA^T C = CAB + C^T AB^T \quad (2.8)$$

$$\nabla_A |A| = |A| (A^{-1})^T \quad (2.9)$$

下面，我们将测试集中的输入特征  $x$  和对应的结果  $y$  表示成矩阵或者向量的形式，有：

$$X = \begin{bmatrix} (x^{(1)})^T \\ (x^{(2)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \quad (2.10)$$

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad (2.11)$$

对于预测模型有：

$$h(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x, \quad (2.12)$$

即：

$$h_x(x^{(i)}) = (x^{(i)})^T \theta, \quad (2.13)$$

于是可以得到：

$$\begin{aligned} X\theta - \vec{y} &= \begin{bmatrix} (x^{(1)})^T \theta \\ \vdots \\ (x^{(m)})^T \theta \end{bmatrix} - \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \\ &= \begin{bmatrix} h_\theta(x^{(1)}) - y^{(1)} \\ \vdots \\ h_\theta(x^{(m)}) - y^{(m)} \end{bmatrix} \end{aligned} \quad (2.14)$$

所以可以得到：

$$\frac{1}{2} (X\theta - \vec{y})^T (X\theta - \vec{y}) = \frac{1}{2} \sum_{i=1}^n (h_\theta(x^{(i)}) - y^{(i)})^2 = J(\theta) \quad (2.15)$$

于是，我们就将代价函数  $J(\theta)$  表示为了矩阵的形式，就可以用上述提到的矩阵运算来得到梯度：

$$\begin{aligned}
\nabla_{\theta} J(\theta) &= \nabla_{\theta} \frac{1}{2} (X\theta - \vec{y})^T (X\theta - \vec{y}) \\
&= \frac{1}{2} \nabla_{\theta} (\theta^T X^T X \theta - \theta^T X^T \vec{y} - \vec{y}^T X \theta + \vec{y}^T \vec{y}) \\
&= \frac{1}{2} \nabla_{\theta} \text{tr}(\theta^T X^T X \theta - \theta^T X^T \vec{y} - \vec{y}^T X \theta + \vec{y}^T \vec{y}) \\
&= \frac{1}{2} \nabla_{\theta} (\text{tr} \theta^T X^T X \theta - 2 \text{tr} \vec{y}^T X \theta) \\
&= \frac{1}{2} (X^T X \theta + X^T X \theta - 2 X^T \vec{y}) \\
&= X^T X \theta - X^T \vec{y}
\end{aligned} \tag{2.16}$$

令上述梯度为 0，得到等式：

$$X^T X \theta = X^T \vec{y}, \tag{2.17}$$

于是得到  $\theta$  的值：

$$\theta = (X^T X)^{-1} X^T \vec{y}. \tag{2.18}$$

这就是最小二乘法得到的假设模型中参数的值。

### 2.2.3 算法实现

本课题所有工作都是在 CentOS release 6.8 (Final) 平台完成，Linux 内核版本为 2.6.32。本文中 SRAM 批量仿真工作由 C shell 语言完成，仿真数据的采集与计算由 Python 语言编写实现，最小二乘法算法的回归与预测由 Python 和 C++ 语言实现。

## 2.3 本章小结

本章首先介绍了机器学习的应用现状，并针对按学习方式指出了监督学习的回归算法。从解决多元回归问题的方式入手，阐述了最小二乘法用于回归问题的原理，并进行了详细的推理。最后说明了本课题中所有使用的算法实现方式。

## 第三章 28nm 标准单元库时序预测

### 3.1 标准单元库

#### 3.1.1 标准单元方法

在半导体设计中，标准单元方法（Standard Cell Methodology）是一种用于专用集成电路（Application Specific Integrated Circuits, ASICs）的数字逻辑的设计方法。标准单元方法通过将一个低级的集成电路逻辑单元的版图封装成一个抽象的逻辑表示形式，比如非门等。这就可以让电路设计师更专注于数字设计，而其他的设计师更专注于物理设计。随着半导体制造业的进步，标准单元方法使得成千上万的逻辑功能相对简单的电路，得以缩放到复杂的百万门级的 SOC 上。

标准单元是通过晶体管通过组合互联构成具有布尔逻辑函数，比如 AND、OR、XOR、XNOR 等，存储单元，比如 Flip-Flop 和 Latch。最简单的标准单元比如 NAND、NOR、XOR 等布尔逻辑功能，尽管更为常用的是一些复杂的布尔逻辑单元：全加器，D 触发器等。标准单元的布尔逻辑功能就是它的逻辑视图，其逻辑行为通常表现为真值表、布尔代数方程和状态转换表。其中布尔代数方程通常用于组合逻辑，状态转换表通常用于顺序逻辑。

通常，标准单元的初始设计开发在晶体管级，以晶体管网表或电路的视图的形式。网表是晶体管、晶体管连接、端口与外部电路连接的节点描述。电路图一般都是通过一些计算机辅助设计（Computer aided design, CAD）和电子设计自动化（Electronic design automation, EDA）的程序提供的图形化界面（Graphical User Interface, GUI）来表示。

标准单元除了有数字逻辑的表示，版图（layout）作为最接近实际产品的视图，是由基础层（layer）组成的，基础层对应晶体管的不同结构，布线层（wiring layers）和通过层（via layer）相互连接，然后连接晶体管端口形成的。

最后，布局和线路（PNR）工具可以用来组合的一切，合成的超大规模集成电路的布局，以自动化的方式，从更高层次设计的网表和平面图。

此外，还可以使用许多其他 CAD 工具来验证单元格视图和模型的其他方面。和其他文件可以创建支持各种工具，利用标准细胞过多的其他原因。所有支持标准单元变化而创建的所有文件被统称为标准单元库。

### 3.1.2 标准单元库

表 3-1 标准单元库的分类

标准单元库	组合逻辑单元	Inverter 反相器 NAND、NOR 与非、或非门 AND、OR 与、或逻辑 BUFFER 缓冲器 MUX 多路选择器 XOR、XNOR 异或、同或逻辑 AOI、OAI 与或非、或与非 ADD 加法器包括全加器、半加器 Clock Cell 包括 Clock BUFFER、Clock Inverter、Clock NAND 等
	时序逻辑单元	Flip-Flop 触发器 LATCH 锁存器
	特殊类型单元	FILL cells 填充单元 Tie High、Tie Low cells 电源连接单元 Delay cells 信号延迟单元 二极管单元 Tap cells 阱连接单元 FDCAP cells 去耦合单元
	低功耗辅助设计单元	Clock Gating 单元 Power Gating 单元 Isolation Cell 隔离单元 Retention Cell 信号保持单元 Level Shift 电平转换单元 Always-on 单元 Well Bias Cell 阱电位偏置单元

一般地，标准单元库通常会包含以下四类单元：组合逻辑单元、时序逻辑单元、特殊类型单元和 PMK（Power Management Kit）低功耗辅助设计逻辑单元。

除了不同工艺尺寸会有不同的标准单元库之外，相同工艺尺寸还会有不同的多套标准单元库用来供用户选择，它们在不同的电路密度和速度上有着相应的优势。例如，中芯国际（SMIC）深亚微米的工艺制程包括了三套标准单元库：一套高性能的超高速 VHS（Very High Speed）单元库、一套密度和速度优化的高速 HS（High Speed）单元库和一套高密度 HD（High Density）单元库。

### 3.1.3 标准单元库流程

传统标准单元库建立的流程如图：

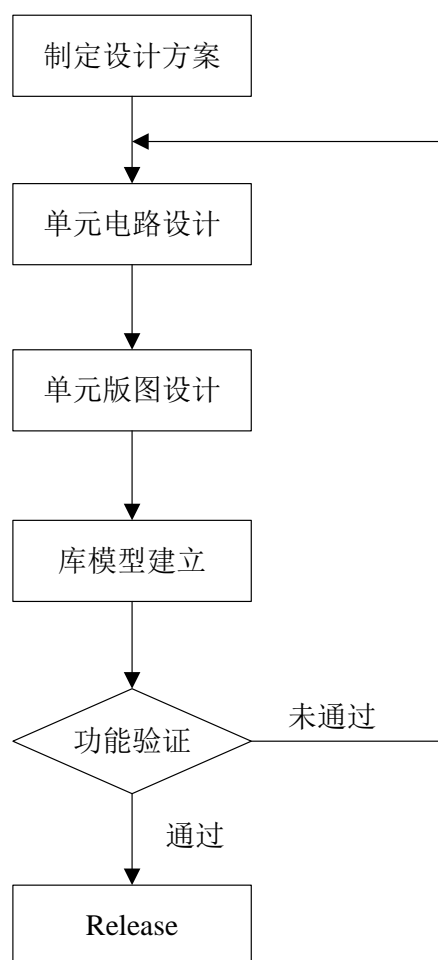


图 3-1 标准单元库建立流程

- (1) 制定设计方案：在设计设计 28nm 标准单元之前，我们首先要制定设计方案。主要包括：标准单元库的 Track 数、需要设计的单元库器件、MOSFET 中 PMOS 与 NMOS 的宽度比、在版图中版面两单元间的中心距距离（Pitch）、单元库器



件的驱动能力等。

- (2) 单元电路设计：合理地设计出所需要的全部标准单元的器件尺寸和电路结构，同时单元的时序和功能必须通过 SPICE 等仿真工具的相关验证。
- (3) 单元版图设计：绘制出所有库单元的版图，并使其满足面积最小，同时所有的电路单元保持高度一致。版图验证主要通过 DRC (Design Rule Check) 和 LVS (Layout Versus Schematic) 这两步的验证。其次，还需要根据 Foundry 所提供的 Design Rule 进行物理特性的验证。
- (4) 库模型建立：一套完整的标准单元库不仅需要包含电路图和版图，还需要包含各种符合国际标准认证的 EDA (Electronics Design Automation) 工具与可识别的库模型文件。后端设计中，需要使用各种 EDA 工具以及利用 Foundry 提供的文件，对标准单元库进行仿真和验证。通常，这些库文件主要描述了单元的时序信息、网表信息和版图信息三个方面。

表 3-2 库模型文件分类

时序信息文件	.lib	标准单元的 Synopsis 的时序模型，描述了标准单元的时间、功率特性
	.db	.lib 的二进制格式文件
	.tlf	标准单元的 TT、FF、SS 工艺角的时序库文件
版图信息文件	.lef	供布局布线使用的版图文件格式，具有工艺信息和各个单元的几何特性的文件，是生成 milkyway 库所必需的
	.gdsii	包含版图信息的文件
网表信息文件	.cdl	标准单元网表，用于 LVS 检查
	.v	Verilog 语言格式文件
	.vhd	VHDL 语言格式文件
符号库文件	.slib	标准单元的符号库文件

- (5) 库验证：标准单元库文件必须通过 Foundry 可制造性的验证。与此同时还必须确保标准单元库模型的正确性，还需要确保应用库模型设计出来的电路在制造出来后时序、功耗、面积等都能满足要求。
- (6) Release：标准单元库在设计和验证之后，还需要撰写标准单元库的设计报告、数据手册、使用手册等相关的文档，以方便满足 Foundry 的使用，并将相关资

料整理打包，最终发布。

## 3.2 时序

### 3.2.1 Liberty 库

LIB 时序库主要用于描述每个标准单元的传输延迟的时间参数、基本的逻辑功能等，其中传输延迟的时间参数是通过测试或仿真工作的提取获得。LIB 库文件主要是由 Synopsys 公司开发的，专门用于对物理单元的时序以及功耗方面的信息进行描述的关键库文件。一般情况下，LIB 文件由标准单元的具体信息和库的基本属性两个部分构成。

基本属性主要描述了标准单元库的基本信息，主要包含 PVT corner (Process、Voltage、Temperature)、文件的版本号、生成日期、电路属性、器件类型、以及时序功耗的查找表等。主要用于标准单元的功能参数库可以进行电路的 Verilog/VHDL 的数字仿真，从而确定所设计电路的功能正确与否，并检查异步时序电路中的竞争和冒险等工作。

```
internal_power() {
    related_pin : "A" ;

    fall_power(pwr_tin_oload_7x7) {
        index_1("0.0014, 0.00717851, 0.027357, 0.0658909, 0.125925, 0.210159, 0.321");
        index_2("0.0001, 0.0005063, 0.00193291, 0.00465706, 0.00890115, 0.0148561, 0.022692");
        values("-9.01904e-05, -8.17841e-05, -7.54719e-05, -7.33688e-05, -7.26228e-05, -7.23099e-05, -7.22542e-05,\
            -0.000120066, -0.000104731, -8.73979e-05, -7.96142e-05, -7.61866e-05, -7.46777e-05, -7.39607e-05,\
            -0.000130273, -0.00012325, -0.00010486, -9.07016e-05, -8.31107e-05, -7.91606e-05, -7.69348e-05,\
            -7.68705e-05, -9.25756e-05, -0.000100105, -9.41088e-05, -8.69738e-05, -8.23151e-05, -7.92124e-05,\
            5.60702e-05, 8.84662e-06, -4.70009e-05, -6.8807e-05, -7.44396e-05, -7.52156e-05, -7.48211e-05,\
            0.000277816, 0.000196678, 7.35128e-05, 3.12551e-06, -3.03089e-05, -4.65244e-05, -5.50157e-05,\
            0.000594049, 0.000480416, 0.000279068, 0.000139033, 6.06426e-05, 1.67003e-05, -9.53282e-06");
    }

    rise_power(pwr_tin_oload_7x7) {
        index_1("0.00143, 0.00717851, 0.027357, 0.0658909, 0.125925, 0.210159, 0.321");
        index_2("0.0001, 0.000506391, 0.00193291, 0.00465706, 0.00890115, 0.0148561, 0.022692");
        values("0.000759953, 0.000766216, 0.000757426, 0.000727301, 0.000676787, 0.000598788, 0.000496922,\
            0.000727172, 0.000740239, 0.000743616, 0.000720167, 0.000672315, 0.000594296, 0.000497029,\
            0.000730773, 0.000734448, 0.000726281, 0.000708882, 0.00066513, 0.000594751, 0.0004841,\
            0.000798609, 0.000780644, 0.000764309, 0.000708032, 0.000663557, 0.000588023, 0.000493243,\
            0.000945226, 0.000900931, 0.00083953, 0.000801242, 0.000686766, 0.000606463, 0.000503606,\
            0.00117914, 0.00110752, 0.000987935, 0.000905395, 0.000837522, 0.000650749, 0.000534238,\
            0.00150742, 0.00140885, 0.00122503, 0.00107831, 0.00098204, 0.000866048, 0.00059579");
    }
}
```

图 3-2 LIB 文件中功耗数值查询表

LIB 文件的主要部分是每个标准单元的功能和有关的延迟参数，主要包括亚阈值电流、输入输出负载以及时序功耗的数值查询表。

如图是联华电子 28nm 工艺的 TT1.05V25C 下的标准单元库的 LIB 文件的 INV 的

功耗查询表。

其中 `rise_power` 表示管脚 A 从 0 变为 1 的内部功耗，但不包含开关功耗，开关功耗有电容乘以电压的平方单独运算；`index_1` 表示输入的传输时间；`index_2` 表示输出端电容；`values` 就是当输入输出取不同值时，相对应的内部功耗值。其中横坐标是 `index_1`，纵坐标是 `index_2`。

```
timing() {
    related_pin : "A" ;
    timing_sense : negative_unate ;
    timing_type : combinational ;

    cell_fall(tmg_ntin_oload_7x7) {
        index_1("0.00143, 0.00717851, 0.027357, 0.0658909, 0.125925, 0.210159, 0.321");
        index_2("0.0001, 0.000506391, 0.00193291, 0.00465706, 0.00890115, 0.0148561, 0.022692");
        values("0.00700959, 0.0106597, 0.0231308, 0.0468026, 0.0836437, 0.135322, 0.203321", \
            "0.00903949, 0.0130079, 0.0256715, 0.0494028, 0.0862693, 0.137963, 0.205969", \
            "0.012613, 0.018432, 0.0336777, 0.0582654, 0.09514, 0.146842, 0.214853", \
            "0.0151835, 0.0233638, 0.0439112, 0.0729309, 0.111877, 0.163858, 0.231816", \
            "0.0164862, 0.0270146, 0.0534445, 0.0894918, 0.134054, 0.189362, 0.258534", \
            "0.0162875, 0.0291974, 0.0615071, 0.1054, 0.158011, 0.219818, 0.293214", \
            "0.0145145, 0.0296991, 0.0679218, 0.11978, 0.181465, 0.252041, 0.332668");
    }

    fall_transition(tmg_ntin_oload_7x7) {
        index_1("0.00143, 0.00717851, 0.027357, 0.0658909, 0.125925, 0.210159, 0.321");
        index_2("0.0001, 0.000506391, 0.00193291, 0.00465706, 0.00890115, 0.0148561, 0.022692");
        values("0.00550394, 0.0103103, 0.0271865, 0.0594104, 0.109624, 0.180066, 0.272772", \
            "0.00630141, 0.0107975, 0.0272522, 0.0594143, 0.109623, 0.180072, 0.272776", \
            "0.0107904, 0.015474, 0.0305614, 0.0606207, 0.109839, 0.180063, 0.272761", \
            "0.0174097, 0.0233866, 0.0400217, 0.0685642, 0.114331, 0.181895, 0.273226", \
            "0.0261785, 0.0334558, 0.0532065, 0.0835544, 0.127805, 0.191341, 0.278587", \
            "0.0374124, 0.0459518, 0.0689266, 0.103113, 0.148999, 0.210944, 0.294051", \
            "0.0513997, 0.0610865, 0.0873152, 0.125814, 0.175641, 0.238911, 0.320344");
    }
}
```

图 3-3 LIB 文件中时序数值查询表

如图是联华电子 28nm 工艺的 TT1.05V25C 下的标准单元库的 LIB 文件的 INV 的时序查询表（部分）。

其中 `timing` 就是标准单元的时序描述；`cell_fall` 是下降时间；`fall_transition` 是传输时间；`index_1` 是输入的传输时间；`index_2` 是输出端的电容；`values` 是 7×7 的时序查询表，横坐标是 `index_1`，纵坐标是 `index_2`。

LIB 文件包含的时序信息是非常繁多的，详见公式 3.1：

$$\text{LIB 数} = \text{阈值电压} \times \text{沟道长度} \times \text{时序模型} \times \text{工艺角} \times \text{电压} \times \text{温度} \quad (3.1)$$

以联华电子 28nmHLP 工艺的标准单元库而言，LIB 数由以下元素组成。

表 3-3 LIB 文件数的影响因素详表

阈值电压	沟道长度 (纳米)	时序模型	工艺角	电压 (V)	温度 (°C)
RVT	30	NLDM	FF	0.855	-40
LVT	35	CCS	FFG	0.90	0
HVT	40	ECSM	TT	0.945	25
			SS	1.05	85
			SSG	1.15	125

其中，时序模型分为三种：

- 1) NLDM: 非线性延时电压源，适用于 40nm 以上，LIB 文件较小；
- 2) CCS: 复合电流源模型，由 Synopsys 公司开发，适用于 40nm 工艺以下，时序更为精准，LIB 文件较大；
- 3) ECSM: 有效电流源模型，有 Cadence 公司开发，适用于 40nm 工艺以下，时序更为精准，LIB 文件中等。

由此可以看出，一套标准单元库会包含数量繁多的 LIB 文件，大约有 1425 个 LIB 文件。与此同时，单个 LIB 文件中具有整个标准单元库的所有电路的时序信息，当阈值电压为 HVT，沟道长度是 30nm 时，就有 939 个标准单元电路。

由此可见，对于标准单元库而言，LIB 文件的数量是非常多的。繁多的 LIB 文件数也就代表着仿真工作的繁多，也就意味着繁多的仿真时长和需求很大的机器资源。本文以标准单元为例，提出了通过机器学习算法对部分标准单元的时序进行回归，并对未知标准单元的时序进行预测的方法。

### 3.2.2 数据收集

由于制造工艺的偏差，同一晶圆的芯片在同一环境下都有不同的工作状态。以 MOSFET 晶体管为例，尽管他们也有类似的电阻和电容，会有不同的状态，如 FNFP、SNSP、TNTP、FNTP、SNFP，不同的阈值电压等参数。除了制作工艺的偏差，MOSFET 晶体管的性能也受到电压和温度的影响。一般来说，仿真的时候会选择不同的温度节点和工作电压节点。

表中有 325 套 Process Corner。

表 3-4 PVT 的选择范围

工艺	FNFP、SNSP、TNTP、FNTP、SNFP
电压 (V)	0.84 0.875 0.91 0.945 0.98 1.015 1.05 1.085 1.12 1.155 1.19 1.225 1.26
温度 (°C)	-40 -0 25 85 125

本课题选取了十个电路分别是：INV、AND2、AND3、NOR2B、NOR2、AOI211、XOR3、ADDF、DFFNQ、DFFNSRPQ。

本章仿真均使用基于 Bsim4 仿真模型的 H-Spice 仿真工具，Slew 设置为 1ns。

以 D 触发器为例，输入端是 D，输出端是 Q。路径延时 (Path Delay) 定义为从输入波形的 50% 到输出波形的 90% 或者从输入波形的 50% 到输出波形的 10%，以确保输出波形的电势能稳定达到目标电位。

图 3-5 描述了 setup time 是时钟达到之前波形需要保持一段时间，hold time 是时钟达到之后波形需要保持的一段时间。

以 DFFs 为例，setup time 是从 D 端到节点 m 的传输路径时间，hold time 有两个值：一个是节点 m 到节点 s 的传输路径时间，另一个是 D 端到节点 nm 的传输路径时间。Hold time 选取二者的最大值。

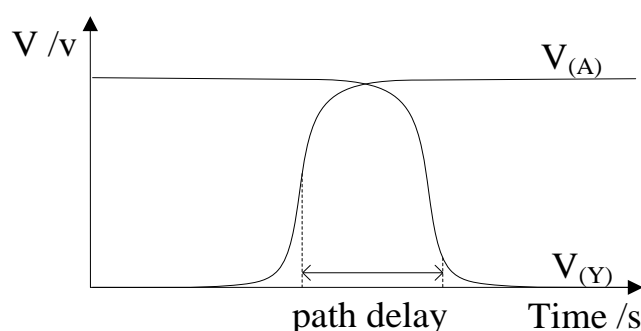


图 3-4 传输路径延时：从输入波形 50% 到输出波形 10% 的传输路径延时

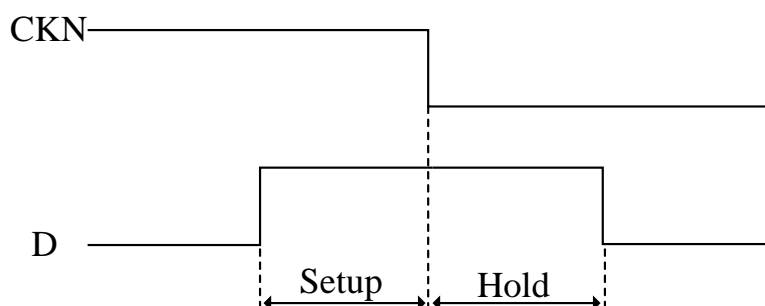


图 3-5 D 触发器的 setup time 和 hold time

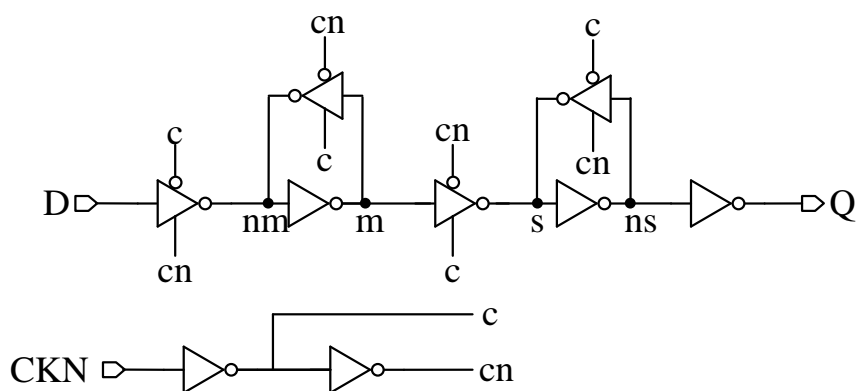


图 3-6 D 触发器的逻辑电路图

本课题中的 10 个标准单元电路分别有不同数量的路径延迟，此外 D 触发器还拥有建立时间和保持时间。

对于单输入电路而言，以 INV 为例，当输入端是上升沿时，输出端是下降沿，具有一个路径延时；当输入端是下降沿时，输出端是上升沿，还具有一个路径延时。对于多输入电路而言，当某一输入端翻转，而其他输入端保持电平不变，且输出端也翻转的时候，才具有一个路径延时。

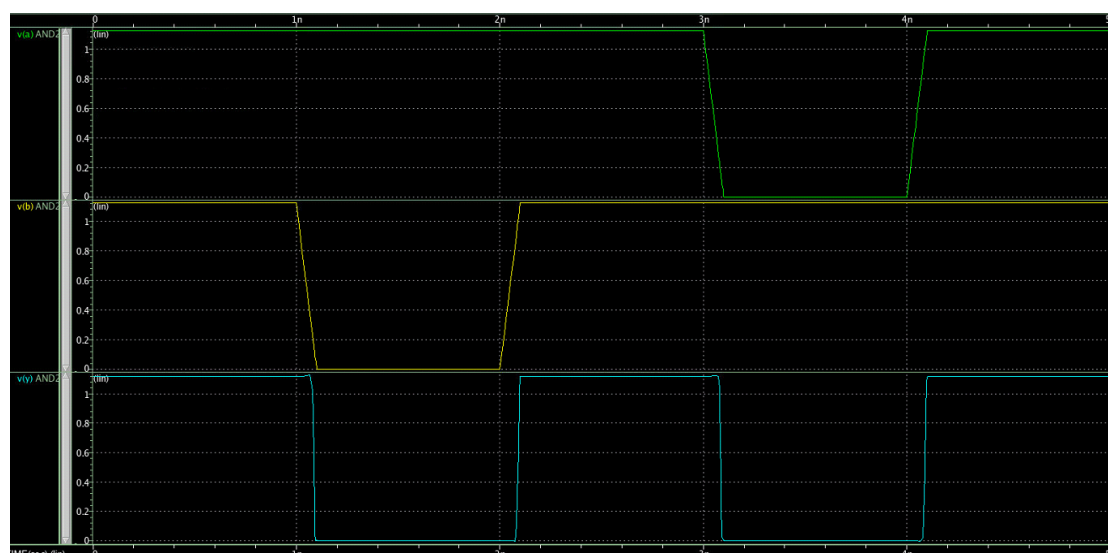


图 3-7 AND2 输入输出波形示意图

以 AND2 为例，时序包含四种时序：

1. 当输入 A 为 1 时，输入 B 从 0 翻转到 1 时，输出 Y 从 0 翻转到 1。输入 B 的 50% 到输出端 Y 的 90% 的传输路径延时。
2. 当输入 A 为 1 时，输入 B 从 1 翻转到 0 时，输出 Y 从 1 翻转到 0。输入 B 的 50% 到输出端 Y 的 10% 的传输路径延时。
3. 当输入 B 为 1 时，输入 A 从 0 翻转到 1 时，输出 Y 从 0 翻转到 1。输入 A 的 50% 到输出端 Y 的 90% 的传输路径延时。
4. 当输入 B 为 1 时，输入 A 从 1 翻转到 0 时，输出 Y 从 1 翻转到 0。输入 A 的 50% 到输出端 Y 的 10% 的传输路径延时。

下表是本课题中所选用的 10 个电路相对应的传输路径延时的数量：

表 3-5 10 个电路的传输路径延时

INV	AND2	AND3	NOR2B	NOR2	AOI211	XOR3	ADDF	DFFNQ	DFFNSRPQ
2	4	6	4	4	16	24	36	3	3

### 3.3 时序预测

本课题提出了两种方法实现预测标准电路的路径延时：电路参数方法和时序方法。

### 3.3.1 电路参数方法

作为集成电路的最小组成单元，金属氧化物半导体效应晶体管（MOSFET）的各种电路特性表现在电路的特性上。MOSFET 根据载流子的极性不同，可以分为 NMOSFET 和 PMOSFET。

MOSFET 晶体管都是有漏极(Drain)、栅极(Gate)、源级(Source)与衬底(Body)。由于每个终端都影响着 MOSFET 的电路特性，且不同 MOSFET 的逻辑组合又具备不同的电路特性，所以本课题选择了 MOSFET 的一些电路参数作为最小二乘法回归学习的特征。

表 3-6 电路参数表及其含义

$I_{vin}$ (A)	信号上升或下降时 VIN 电压的电流的峰值
$I_{vdd}$ (A)	信号上升或下降时 VDD 电压的电流的峰值
$I_{vss}$ (A)	信号上升或下降时 VSS 电压的电流的峰值
$I_{on}$ (A)	信号上升或下降后 MOS 管的电流的平均数值
$I_{peak}$ (A)	信号上升或下降后 MOS 管的电流的峰值
$V_{th}$ (V)	信号上升或下降时 MOS 管的阈值电压

以 INV 为例，共有 325 组仿真数据，每组数据有 9 个电路参数作为拟合的特征 (Feature) 和一个路径延时作为目标 (Target)。表中所示是五组 INV 的特征和目标值。

表 3-7 五组 INV 的电路参数与延时

$I_{vin1}$ ( $10^{-6}$ A)	$I_{vdd}$ ( $10^{-6}$ A)	$I_{vss}$ ( $10^{-6}$ A)	$V_{th1}$ (V)	$V_{th2}$ (V)	$I_{on1}$ ( $10^{-9}$ A)	$I_{on2}$ ( $10^{-9}$ A)	$I_{peak1}$ ( $10^{-5}$ A)	$I_{peak2}$ ( $10^{-6}$ A)	Delay ( $10^{-11}$ s)
5.868	1.345	4.226	0.5129	0.5063	4.357	0.9851	1.650	4.074	3.146
4.419	1.020	3.226	0.2629	0.3713	1.663	2.142	1.064	3.672	2.527
7.076	9.112	11.97	0.2456	0.3713	5.097	4.475	2.466	13.06	1.859
6.802	1.614	5.830	0.4511	0.5922	2.508	0.6069	1.996	6.239	1.456
5.461	1.422	3.938	0.5768	0.4257	15.85	4.057	1.358	3.789	4.058

本次学习使用 10 折交叉验证 (10-Fold Cross Validation)，每次学习使用 32 组仿真数据作为训练集，另外的 293 组仿真数据作为测试集。

本次学习选择了拟合优度作为拟合程度的度量，拟合优度 (Goodness of Fit) 是指



回归模型对观测值的拟合程度。度量拟合优度的统计量是可决系数（determination coefficients） $R^2$ 。可决系数的值越接近 1，说明回归直线对观测值的拟合程度越好；反之，可决系数的值越小，说明回归直线对观测值的拟合程度越差。

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (3.2)$$

可决系数的表达式如上，其中 $SS_{res}$ 是回归平方， $SS_{tot}$ 是总平方和。是表示因变量与所有自变量之间的总体关系。拟合优度的可决系数是无量纲系数，有确定的取值范围（0-1），便于对不同资料回归模型的拟合优度进行比较。

表 3-8 十个标准电路延时训练集与测试集的拟合优度

$R^2$	INV	AND2	AND3	NOR2B	NOR2	AOI211	XOR3	ADDF	DDFNQ	DDFNRPQ
Train	0.968	0.973	0.968	0.982	0.989	0.960	0.989	0.995	0.996	0.971
Test	0.962	0.930	0.952	0.973	0.983	0.924	0.962	0.962	0.965	0.940

本次学习还选择了准确度（Accuracy）作为预测时序准确度的评价标准。准确度和相对误差一般由公式表示：

$$\delta = \Delta \div L \times 100\% \quad (3.3)$$

$$\alpha = 1 - \delta \quad (3.4)$$

其中， $\alpha$ 是准确度， $\delta$ 是实际相对误差， $\Delta$ 是绝对误差， $L$ 为真实值。相对误差一般表示为百分数。相对误差越接近 0，表示预测的时序的准确度越好，预测值与真实值越接近。同样的，准确度越接近 1，表示预测的时序的准确度越好，预测值与真实值越接近。

表 3-9 十个标准电路延时训练集与测试集的预测准确度

$\delta$	INV	AND2	AND3	NOR2B	NOR2	AOI211	XOR3	ADDF	DDFNQ	DDFNRPQ
Train	0.836	0.941	0.968	0.985	0.935	0.887	0.940	0.991	0.993	0.982
Test	0.859	0.796	0.952	0.958	0.771	0.779	0.754	0.954	0.955	0.958

电路参数方法可以取得很好的预测效果，与此同时，所有的电路参数中， $I_{vdd}$ 、 $I_{vss}$  在电路中只有一个； $I_{vin}$  的数量对应输入端的数量； $V_{th}$ 、 $I_{on}$ 、 $I_{peak}$  每一个 MOS 管都有。对于小电路，电路参数不是很多，然而，对于 DFF 这种复杂一些的电路，则有三十多个 MOS 管，电路参数方法会有大概一百个电路参数作为学习的特征。

电路参数方法对于标准单元仿真周期，我们采用时间成本和机器资源成本来进行评价。

以联华电子 28nmHLP 为例，通常以 50 个核来进行仿真，大致需要 24 小时。采

用本方法后，以 50 个核来进行仿真数据的采集以及时序数据预测，大致需要 17 小时。

### 3.3.2 时序方法

从 Process corner 来看，一般而言，时序都随着 Process、Voltage、Temperature 的变化呈现相对应的变化。

以 DFF 为例，本方法以其他标准单元电路的时序作为学习的特征，DFF 的时序作为目标。其中其他标准单元电路分别是 INV、AND2、AND3、NOR2B、NOR2、AOI211、XOR3 和 ADFF。如图是 5 个 process corner 下的 DFF 的时序，分别是传输路径延时、setup time 和 hold time。

本学习也是用 10 折交叉验证，表列出了 DFF 的时序拟合优度。表列出了 DFF 的准确度。可决系数的值越接近 1，说明回归直线对观测值的拟合程度越好，同样的，准确度越接近 1，表示预测的时序的准确度越好，预测值与真实值越接近。

表 3-10 DFFs 的传输路径延时、setup time 和 hold time 在 5 个 corner 下时序

Process corner	DFFNQ			DFFNSRPQ		
	Path delay	Setup time	Hold time	Path delay	Setup time	Hold time
	[10 <sup>-11</sup> s]	[10 <sup>-11</sup> s]	[10 <sup>-11</sup> s]	[10 <sup>-11</sup> s]	[10 <sup>-11</sup> s]	[10 <sup>-11</sup> s]
tt1p05v0c	7.766	8.880	5.857	9.717	11.92	8.879
ff1p26v125c	4.378	4.760	3.634	5.059	6.276	5.396
ss0p84vn40c	72.50	65.69	49.26	95.01	92.43	72.91
sf1p05v25c	7.999	9.969	5.928	9.793	12.46	8.312
fs1p05v85c	7.915	7.918	6.556	10.05	12.24	10.67

表 3-11 DFFs 的传输路径延时、setup time 和 hold time 的拟合优度

R <sup>2</sup>	DFFNQ			DFFNSRPQ		
	Path delay	Setup time	Hold time	Path delay	Setup time	Hold time
Training set	0.9991	0.9996	0.9989	0.9990	0.9996	0.9990
Testing set	0.9984	0.9987	0.9980	0.9979	0.9992	0.9983

表 3-12 DFFs 的传输路径延时、setup time 和 hold time 的预测准确度

$\delta$	DFFNQ			DFFNSRPQ		
	Path delay	Setup time	Hold time	Path delay	Setup time	Hold time
Training set	0.9923	0.9907	0.9772	0.9981	0.9946	0.9763
Testing set	0.9912	0.9821	0.9874	0.9901	0.9846	0.9854

从上表可以看出，时序方法可以取得相当好的预测效果。

与电路参数方法相比较，时序方法所需要的特征数目较少，只需要其他电路的时序即可。不需要每个电路的每个 MOS 管的电路参数。

时序方法对于标准单元仿真周期，我们采用时间成本和机器资源成本来进行评价。

以联华电子 28nmHLP 为例，通常以 50 个核来进行仿真，大致需要 24 小时。采用本方法后，以 50 个核来进行仿真数据的采集及时序数据预测，大致需要 12 小时。

### 3.4 本章小结

本章详细阐述了标准单元方法和标准单元库流程，着重讲述了时序库 LIB 文件，分析了标准单元库仿真的周期时长。并通过电路参数方法和时序方法两种方法，使用最小二乘回归算法预测了标准单元电路的时序，其中电路参数方法的预测时序的准确度能达到 94.58% 和 87.36%；时序方法的预测时序的准确度能达到 98.82% 和 98.68%。并从时间成本和机器资源成本对两种方法进行了评估，电路参数方法节约时间大致有 30%，时序方法节约时间大致有 50%。

## 第四章 SRAM 编译器

本章以台积电 28nm SRAM 为例，选择六管单元作为设计方案，介绍了 SRAM 编译器，然后详细分析 SRAM 编译器的表征（Characterization），提出预测三种仿真数据的方法，最后对预测的结果和预测方法进行了评价。

### 4.1 SRAM 基本原理

#### 4.1.1 SRAM 架构

台积电 28nm SRAM 架构如图 4-1。

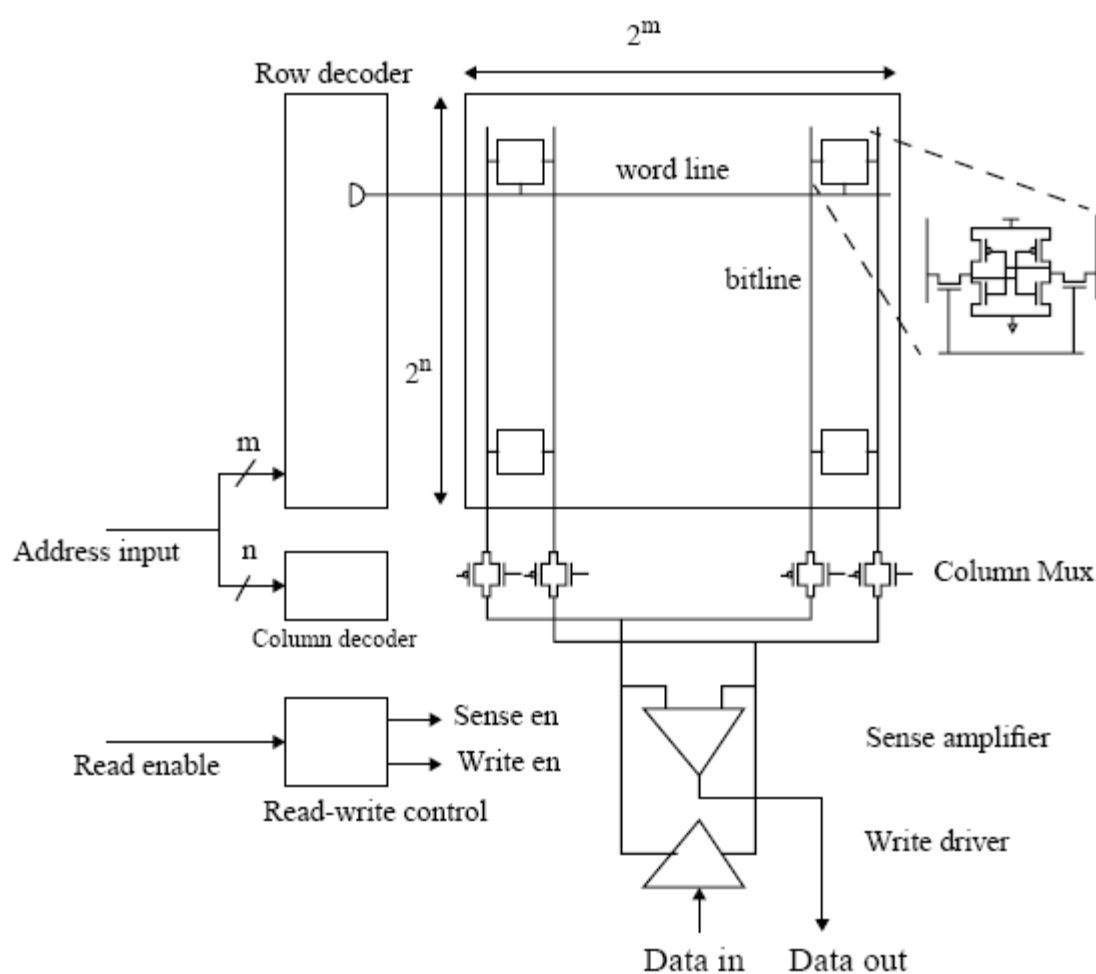


图 4-1 SRAM 电路框架图

SRAM 即静态随机存取存储器，SRAM 主要由存储单元阵列（Core Cell Array）、地址译码器（Decoder）、灵敏放大器（Sense Amplifier）、控制电路（Control circuit）和缓冲/驱动电路（FFIO）。

如上图 4-1，存储单元阵列有  $2^m$  行和  $2^n$  列，分别表示存储阵列有  $2^m$  根字线（Word line）构成，每个字深存储  $2^n$  根位线（Bit line）。每个 SRAM 包含  $2^m \times 2^n$  个存储单元。每个存储单元与一根字线和一组反相的位线相连接，并通过它们进行对存储单元的读写操作。

译码器分为行译码器（Row decoder）和列译码器（Column decoder）。行译码器接收地址信号，选中一根字线触发阵列的一行，在进行读操作的时候，字线驱动位线，反相的位线通过灵敏放大器读出存储单元的数据。

通常为了节省功耗和减少数据读取时间，存储器中的字数会远远多于每个字存储的位数，这样会使得存储器变得长而窄，因为位线过长会增加数据访问的速度。

### 4.1.2 SRAM 存储单元

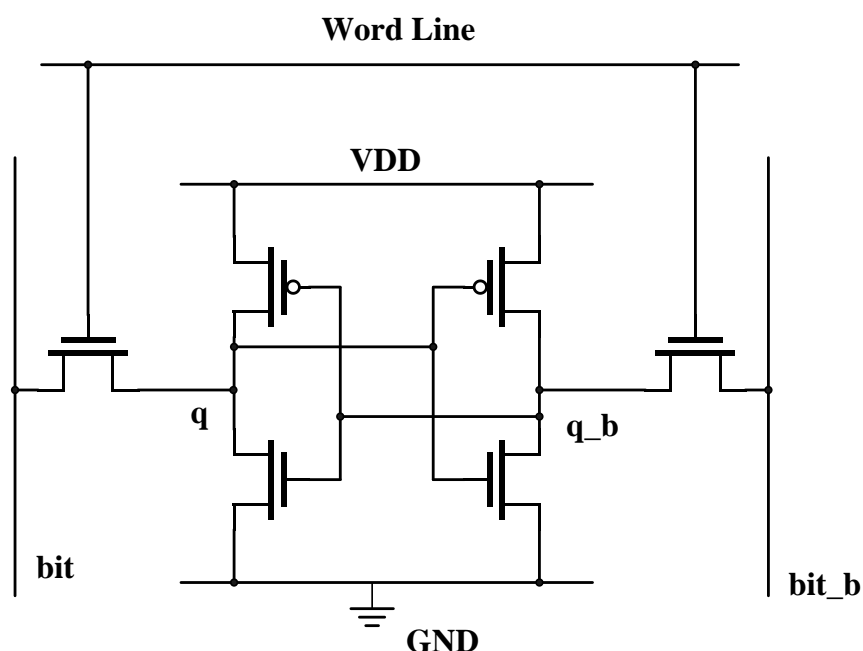


图 4-2 SRAM 6 管单元原理图

图中所示为 6 管 SRAM 单元，存储单元使用一根字线（Word Line）、两根反相的位线（bit 和 bit\_b）。单元中包括了一对交叉耦合的反相器，并且每根位线链接了一个

存取晶体管。一对互补的数据存储在交叉耦合的反相器上。如果数据受到轻微的干扰，由回路构成的正反馈将使数据恢复到 VDD 或 GND。选中这个单元的字线就可以对这个单元进行读出或写入。

### 4.1.2 SRAM Compiler

一般而言，Memory Compiler 属于 IPs 的一种，目前部分 Design House 和 Foundry 都可以向客户提供配套的 Memory Compiler。Memory Compiler 是设计一套完整的存储器单元模块（SRAM、Flash 等）。设计中包含不同大小的存储器单元模块，尽量做到单元模块的面积和速度最优。用户可以通过简单的界面定义实现目标要求的存储器单元模块的 GDSII 文件和网表文件等。由于用户不能更改电路的内部结构和版图，所以面积和速度就是 Memory Compiler 的竞争力所在。

常用 SRAMs 是全定制设计，全定制设计以存储器密度、速度、功耗和良率为主要设计目标，全定制设计需要相应较长的设计周期。能在较短时间内完成满足设计需求的 SRAM，就意味着在高度竞争的市场中具有强有力的竞争力。SRAM Compiler 就是一种解决方法，SRAM Compiler 不仅能节约成本，而且能在很大程度的缩短设计周期。

SRAM Compiler 是一种能生成不同存储器配置的计算机智能化程序。SRAM Compiler 主要功能包括：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、建立电源栅结构等。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。

### 4.1.2.1 SRAM Compiler 设计流程

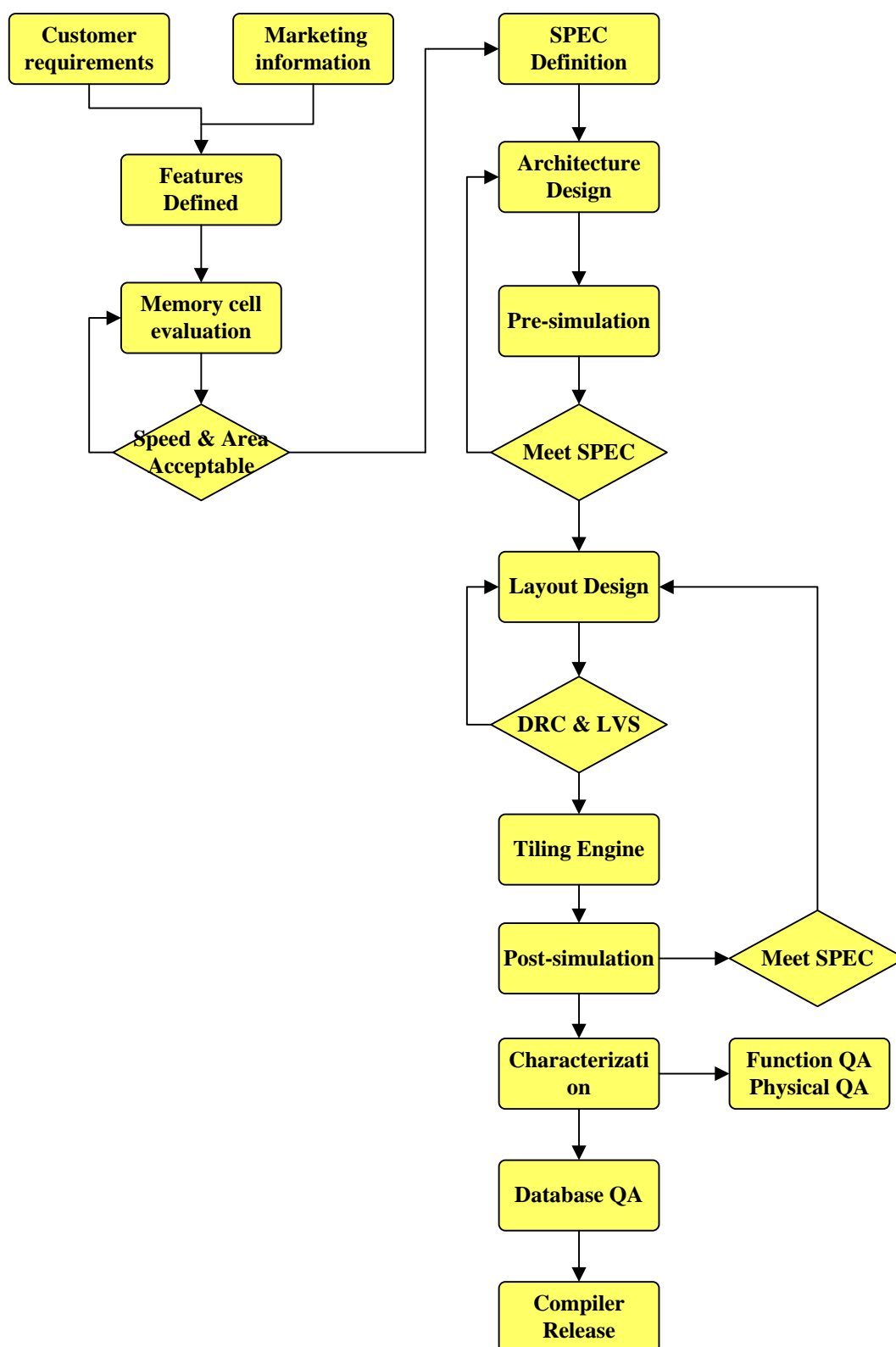


图 4-3 SRAM Compiler 设计流程图

SRAM Compiler 通过调用设计电路结构中的子模块的版图来产生存储器模块版图,所以不同的电路设计结构得到的版图会对 SRAM Compiler 的面积有很大的影响。

如图 4-4 是 SRAM Compiler 的设计流程,发开一套 SRAM Compiler 的主要流程如下:根据客户的需求以及市场信息,提出对 SRAM 的设计需求,主要是 SRAM 所具备的一些功能和特征,并对存储器单元进行评估,并检查是否满足存储器设计的速度和面积要求;然后对 SPEC 的配置进行电路图设计及预仿真,其次是页元(Leaf cell)的版图设计,通过 DRC 规则对版图设计是否符合设计规则进行检查,以及通过 LVS 规则对电路图和版图检查是否一致;使用 SRAM Compiler 的 Tiling Engine 来产生存储器模板 (Memory Instance) 版图,最后是对不同配置的 Instance 进行表征 (Characterization),主要通过时序 (Timing)、功耗 (Power) 和电容 (Capacitance) 来体现。

在 Memory Characterization 环节中,主要是仿真工作和仿真数据收集。实际设计中,仿真工作的时间几乎占据了全部的 Memory Characterization 环节。因为仿真工作的数量由 SPEC 配置和 PVT 共同决定。而 SPEC 与 PVT 参数包含了特征、MUX、Bank、Word、Bit、Input Slew、Output loading、Process Corner、Voltage 以及 Temperature 等。参数的繁多决定了仿真工作的数量的繁多。



## 4.1.2.2 SRAM Compiler 整合流程

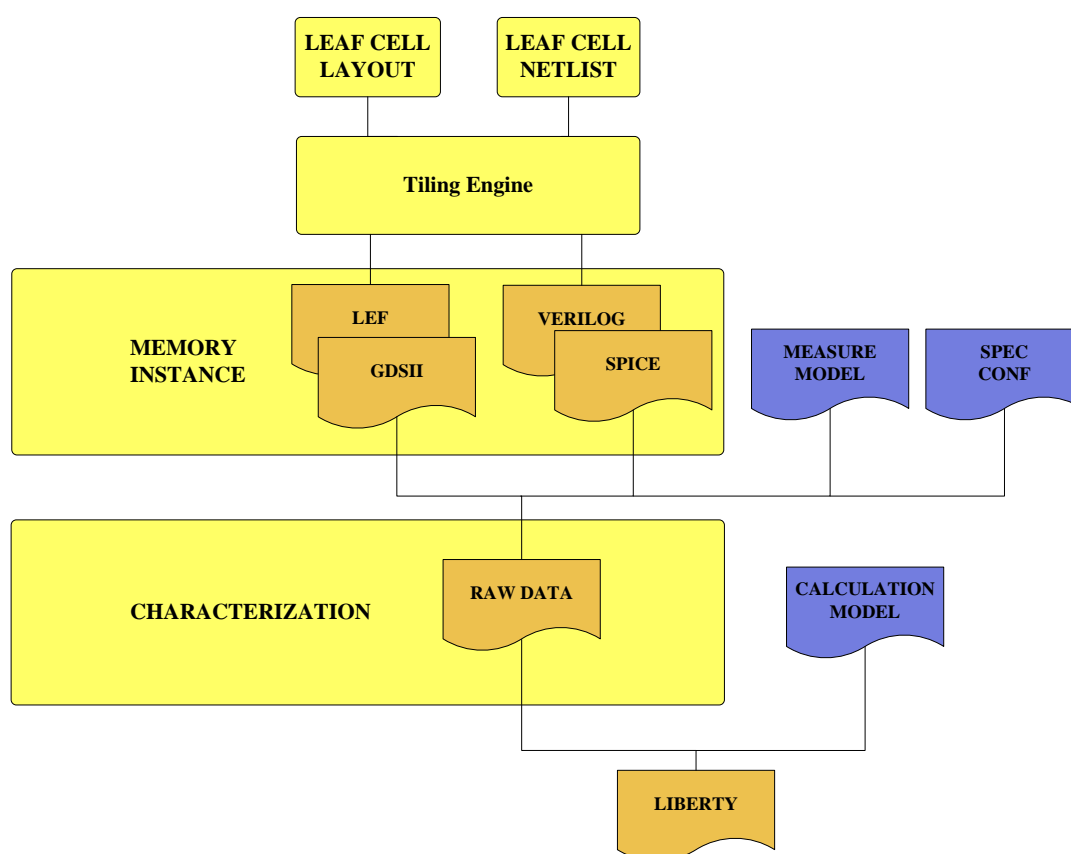


图 4-4 SRAM Compiler 整合流程图

1. Leafcell 的版图和电路在通过 DRC 和 LVS 之后，通过 Tiling Engine 的脚本生成不同 instance 以及 instance 的版图文件（LEF、GDSII 文件）和电路描述文件（Verilog 语言文件、SPICE 文件）；
2. Characterization 环节通过设计者给出的 Instance 列表对 instance 抽象成 CP（Critical Path），通过 Word、Bit，Feature 的变化来改变 CP 中的各个参数，模拟对应 Instance 的行为，节省了仿真时间，同时还不失准确性。针对不同 instance 的 SPEC 配置文件的变量参数进行配置，通过脚本自动生成仿真文件，并进行仿真；
3. 仿真完成后，通过测量描述语言对仿真结果文件进行数据抽取，生成 Rawdata 文件，并通过计算文件生成最后的 LIB 文件。
4. 最后将 LIB 文件、LEF 文件、GDSII 文件、Verilog 文件和 SPICE 文件整合，编写 Compiler 实现程序，使得用户可以通过自定义 instance 的大小与功能自动生成 SRAM 及其网表文件、版图文件与表征数据库等。

通过 Compiler 工具中的各种引擎模块调用设计中相应的设计输入文件，如 SPEC 配置文件、Compiler 的物理单元库 (GDS)，网表单元库 (Netlist)、Memory Development Language (MDL) 语言脚本等文件，具体需要的文件见图；Memory Compiler 的输出文件结构如图，大致有 GDSII、LEF、SPICE 和 Verilog 文件等。

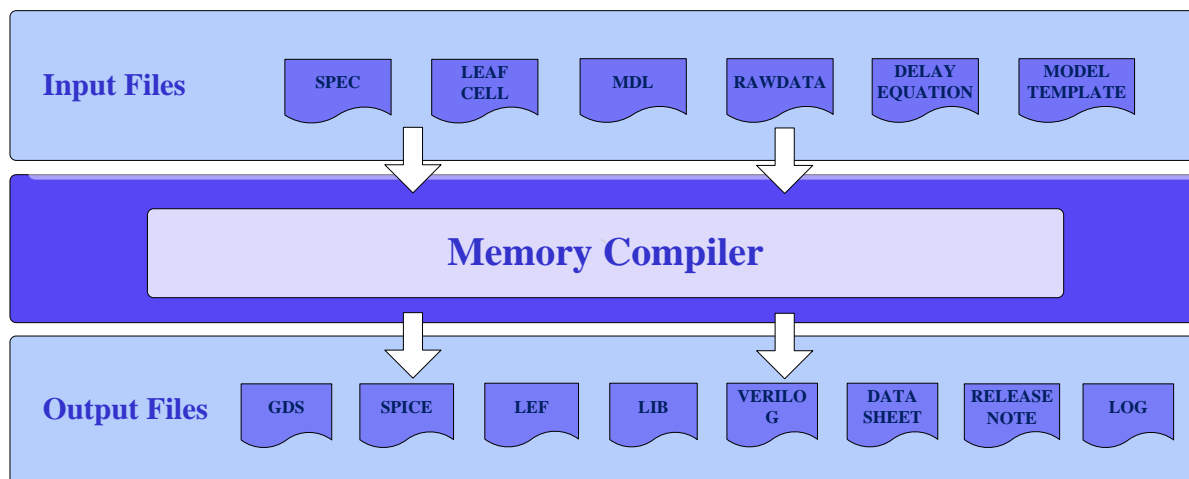


图 4-5 Memory Compiler 输入输出文件结构

## 4.2 存储器表征

Memory Characterization 是 SRAM Compiler 设计中非常重要的环节，主要是 instance 的仿真工作以及仿真数据的收集。其中仿真数据主要是时序和功耗。

Characterization 的仿真工作具有一定的重复性，可以通过脚本的实现来生成 Test bench，进行仿真、数据检查以及数据收集。Characterization 减少了设计者的仿真时间，同时也可以减少一些人为的错误。

仿真的工作量主要由 SPEC 配置和 PVT 共同决定，SPEC 配置有 FEATURE、MUX、BANK、WORD、BIT、INPUT SLEW 和 OUTPUT LOADING 等。

### 4.2.1 特征

本课题 SRAM Compiler 所采用的工艺是台积电 28nm 的 HPC+工艺版本，本课题 SRAM Compiler 所采用的功能如下：

1. Synchronous read and write operations
2. Column Mux options for the best aspect ratio

3. Fully customized layout density available in 0.9V  $\pm 10\%$
4. Three power management modes: Light Sleep, Deep Sleep and Shut Down to reduce static power
5. Bit write and global write operations available
6. Extra margin control option
7. Special test modes externally bypass read and write self-timing circuits
8. Redundancy inside to repair defective cell

表 4-1 台积电 28nm SRAM Compiler 具备的特征

1	具有 deep sleep 模式
2	具有 shut down 模式
P	具有 light sleep 功能
B	具有 Bit write 功能
I	具有 IO redundancy 功能
M	具有 BIST 功能
E	具有 HVT Periphery 功能
L	具有 LVT Periphery 功能

用户可以通过定义不同的 word、bit 和 mux 来优化存储器，从而实现面积、速度和功耗的最佳平衡。以上的特征都被集成到 Memory Compiler 中，用来实现系统应用和用户需求。在设定一些参数之后，Compiler 就可以在很短的时间内生成 memory instance 的 data sheet、GDSII Layout、LVS netlist 和 PR model 等。

通常，Memory Compiler 所提供的全部功能，用户在自定义生成 SRAM 的时候，根据需求选择其中几个或者全部的功能。这就要求在进行仿真的时候，针对不同的特征组合进行不同的仿真，这也是仿真工作量繁多的重要原因。

以下为需要进行仿真的特征组合（字母组合）：

无特征：-

一个特征：1, 2, b, i, m, p

两个特征：12, 1b, 1i, 1m, 1p, 2b, 2i, 2m, 2p, bi, bm, bp, im, ip, mp

三个特征：12b, 12i, 12m, 12p, 1bi, 1bm, 1bp, 1im, 1ip, 1mp, 2bi, 2bm, 2bp, 2im, 2ip, 2mp, bim, bip, bmp, imp, 12bi, 12bm, 12bp, 12im, 12ip, 12mp, 1bim, 1bip

四个特征：1bmp, 1imp, 2bim, 2bip, 2bmp, 2imp, bimp

五个特征: 12bim, 12bip, 12bmp, 12imp, 1bimp, 2bimp

六个特征: 12bimp

共计 64 组特征。针对每一组特征都有相应的仿真工作需要完成。

## 4.2.2 实例

用户可以通过定义不同的 word、bit、mux 和 bank 来定义需求的存储器。

通常随着存储阵列容量的变大, 那么字线和位线就会变得更长。增加字线和位线的长度会使电容变大, 从而导致延时和功耗的增加。因此, 通常会将大容量的存储阵列分割成为多个较小的存储阵列, 这样的较小的存储阵列就被称为体 (Bank)。以图 4-7 为例, 将一个完整的存储阵列分成 2 个 bank 之后, 其位线的长度也变成了原来的 1/2, 相应的负载也变成了原来的 1/2, 从而提升了速度。

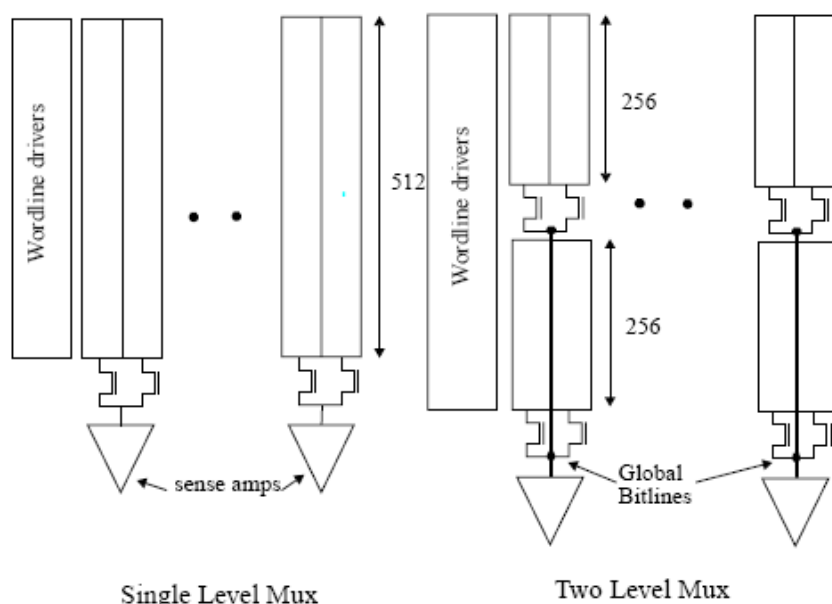


图 4-6 Bank 示意图

通常为了节省功耗和减少数据读取时间, 存储阵列中的字数会远远多于每个字存储的位数, 这样会使得存储器变得长而窄, 因为位线过长会增加数据访问的速度。

通常采用的另一个做法就是 Column Mux。将原来每列的存储单元的位线变为若干列 (记为  $k$ ) 的存储单元共用, 这样可以将原来的  $2 \times n$  根位线减少为  $n/k$  根位线。

其中 Mux 就等于 k，且 Mux 总是偶数。在具体的 SRAM Compiler 设计中，会提供很多不同的字线、位线和 Mux 来供用户进行选择。

本课题的 SRAM Compiler 是小面积的存储器，设计中不分 Bank，因此 Bank 的值为 1。

Mux 根据 SPEC 文件定义为 2 和 4。当 Mux 取 2 时，Word 最小取 32，最大取 512，步长为 4，Bit 最小取 8，最大取 144，步长为 2；当 Mux 取 4 时，Word 最小取 64，最大取 1024，步长为 8，Bit 最小取 8，最大取 72，步长为 1。

表 4-2 字长根据 Mux 不同的选择范围

Word Range	
Column Mux=2	32,36,40,...508,512
Column Mux=4	64,72,80,...1016,1024

表 4-3 位线根据 Mux 不同的选择范围

Bit Range	
Column Mux=2	8,10,12,...142,144
Column Mux=4	8,9,10,...71,72

当 Mux 选定 2 或者 4 的时候，选择不同的 Word 和 Bit 的值即可确定 Memory Instance 的大小。从上表可以看出，当 Mux=2 的时候，Word 可取 120 个值，Bit 可以取 140 个值；当 Mux=4 的时候，Word 可以取 120 个值，Bit 可以取 64 个值。由此可计算出本课题的 SRAM Compiler 的 Instance 个数共有 24480 个。理论上，用户自定义生成的 Memory Instance 的可以是 24480 中的任何一种，这就要求每一个 Memory Instance 都有相应的仿真数据，供用户查询。仿真阶段完成所有的 Instance 的仿真是不可能实现的，通常设计公司会提供比较典型的几个 Instance 的仿真数据。而客户需求所提供的 Instance 之外的仿真数据的时候，就需要设计公司重新进行仿真。

### 4.2.3 输入压摆和输出负载

在 IO 部分中，电路的端口主要有地址信号 A、时钟信号 CK、控制信号 CEB，WEB、数据输入信号 DI 和数据输出信号 DO。输入端口和时钟端口的 slew 的不同对电路的表征有不同的影响，在输出端口的负载的大小对电路的表征也有不同的影响。

在 SPEC 文件中，input slew 和 output loading 都设置七个值。其中 A 作为数据输

入的端口，采用 input slew 的值，称之为 Data slew，CK 作为时钟，也采用 input slew 的值，称之为 CK slew，DO 作为输出端，选有 output loading 的值，称之为 load。

本课题中，input slew 和 output loading 的值如下表：

表 4-4: input slew 和 output loading 数值表

Input slew	Data slew	0.004, 0.050, 0.100, 0.200, 0.500, 0.750, 1.000
	CK slew	0.004, 0.050, 0.100, 0.200, 0.500, 0.750, 1.000
Output loading	Load	0.001, 0.020, 0.050, 0.100, 0.250, 0.300, 0.350

对于 instance 而言，Data slew、CK slew 和 Load 选择不同的数值的时候，都有不同的电路表征。也就是说对于某个确定的 Instance 而言，在 IO 中就有 343 种状态。

#### 4.2.4 工艺角

在不同批次的晶圆和同批次的不同晶圆之间，MOSFETs 的电路性能的变化都很大。为了规范电路设计，需要从晶片中提取相对应的电路器件模型，比如：片上 NMOS 和 PMOS 的测试会显示出不同的延时。在确保有效的工作状态下，对各种工艺角和各种温度条件下的电路进行仿真。工艺工程师们会在保证器件在电路性能处于在某个范围内，对有效芯片进行严格控制预期的参数变化。最终从仿真结果来对工艺角的选取制定标准。并根据这一标准对电路设计进行规范。这一做法可以在很大程度上减轻电路设计的困难。

通常提供给设计师的性能范围只适用于数字电路并以工艺角（Process Corner）的形式给出。通常表示为工艺、电压和温度的连续组合形式，比如 TT0P9V125C。

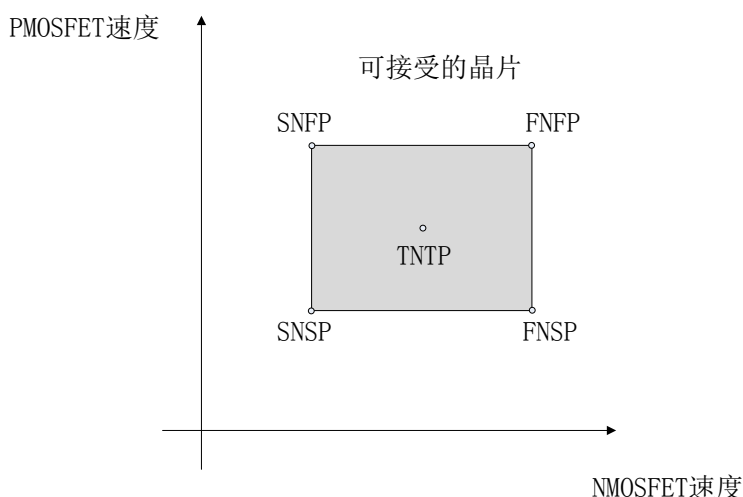


图 4-7 PVT Corners 的示意图

图 4-8 表示了 PVT Coners 的思想：把有效的 NMOS 和 PMOS 晶体管的速度范围限制在由四个角所确定的矩形内。这四个角分别是：快速 NMOSFET 和快速 PMOSFET（FNFP），慢速 NMOSFET 和慢速 PMOSFET（SNSP），快速 NMOSFET 和慢速 PMOSFET（FNFP），慢速 NMOSFET 和快速 PMOSFET（SNFP）。比如具有较薄的栅氧和较低阈值电压的晶体管在图中就落在快速角的附近。

从载流子迁移率（Carrier Mobility）的角度来看，载流子迁移率指的是载流子在单位电场的作用下漂移的平均速度。工艺角的命名方式是两个字母组合，前一个字母表示 NMOS 的工艺角，后一个字母表示 PMOS 的工艺角。工艺角的类型通常有：快速（Fast）、慢速（Slow）和正常（Typical）。以 TT 为例，就是指正常速度的 NMOS 管和正常速度的 PMOS 管。

通常的，单一器件的载流子迁移率分布呈现正态分布，均值为 TT，最小极值为 SS，最大极值为 FF。一般而言，在 NMOS 和 PMOS 管的迁移率分布范围中，TT、SS、FF 这三个工艺角能覆盖大约  $\pm 3$  Sigma，即约 99.73% 的范围。

然而，不同的工艺偏差也有很多情况。比如：掺杂浓度、制造时的温度控制、刻蚀程度等情况。再考虑到同一个晶圆上不同区域的非一致性，以及不同晶圆之间不同物理环境的因素。这些随机性事件的发生，只能使用统计学的方法才能够对工艺偏差进行有效的评估，才能覆盖到合理的范围。

本课题采用 3 Corner Model，它们是 TNTP，SNSP 和 FNFP。

设计除了要满足上述的 Process Corner 外，还需要满足电压与温度等条件，形成的组合称为 PVT (process, voltage, temperature)。

表 4-5: PVT 配置表

Process	TT、SS、FF
Voltage/V	0.72、0.81、0.9、0.99、1.08
Temperature/°C	-40、25、125

如表，电压设定数值分别是中位电压的 80%、90%、100%、110% 和 120%。设计时还需要找到最好最坏情况，时序分析中将最好的条件(Best Case)定义为速度最快的情况，而最坏的条件(Worst Case)则相反。

根据不同的需要，会有不同的 PVT 组合，以下是一些标准的分析条件：

1. WCS (Worst Case Slow): slow process、high temperature、lowest voltage
2. TYP (Typical): typical process、nominal temperature、nominal voltage
3. WCL (Worst Case Cold): slow process、lowest temperature、lowest voltage
4. BCF (Best Case Fast): fast process、lowest temperature、high voltage
5. ML (Maximal Leakage): fast process、high temperature、high voltage
6. TL (Typical Leakage): typical process、high temperature、nominal voltage

前四个的 PVT 组合主要用于时序分析，后两者 PVT 组合通常用于功耗分析。与前面说设计师通常只会提供一部分 Instance 的仿真数据一样，设计师通常也只会提供小部分 PVT 的仿真数据。当用户需求超出了最初的设计范围时，需要重新进行仿真工作。

### 4.3 本章小结

本章主要介绍了 SRAM 的基本框架和基本电路结构，然后描述了 SRAMCompiler 的设计流程与主要文件类型；着重对 Compiler 的表征环节进行了阐述，主要分析了 Instance 生成所受的因素影响。并详细对特征、word、bit、mux、bank、input slew、output loading 和 PVT corners 进行详细解释，间接地表达了 memory instance 数量与仿真工作量的繁多。



## 第五章 SRAM 表征数据预测

### 5.1 时序

#### 5.1.1 时序数据

##### 5.1.1.1 读操作时序

SRAM 有三种工作状态：保持（Standby）、写入（Write）和读出（Read）。

SRAM 的数据保持时，芯片使能信号 CEB 处于无效状态，无论其他信号是什么状态，都不会对存储单元进行任何操作。

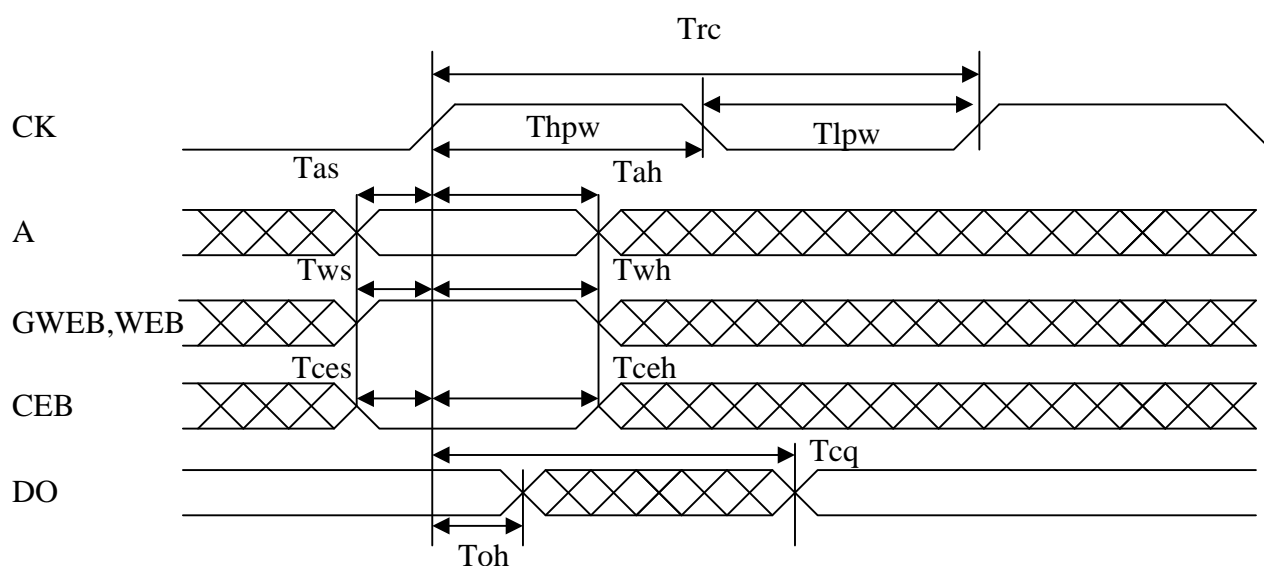


图 5-1 读操作波形示意图

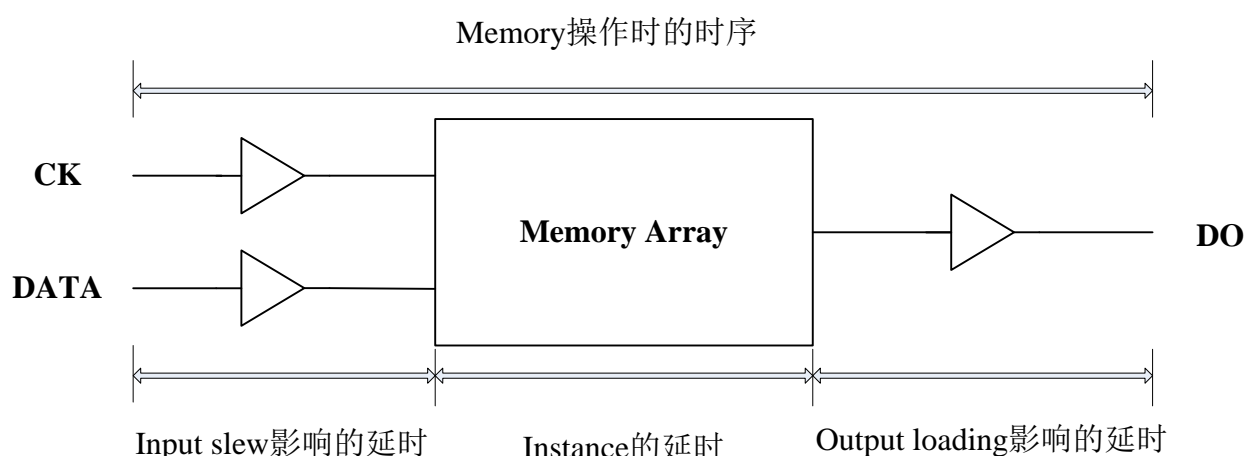
SRAM 读操作时序如图，其中 WEB 为写使能，CK 是时钟信号，CEB 为片选中信号，A 为地址信号，DO 为数据输出信号。

表 5-1 读操作时序

Parameter	Symbol
Read cycle time	Trc
Data access time at the CK rising edge	Tcq
Output data hold time after the CK rising edge	Toh
Address setup time before the CK rising edge	Tas

Address hold time after the CK rising edge	Tah
GWEB,WEB setup time after the CK rising edge	Tws
GWEB,WEB hold time after the CK rising edge	Twh
CEB setup time before the CK rising edge	Tces
CEB setup time after the CK rising edge	Tceh
Clock high/low pulse widths	Thpw/Tlpw

SRAM 读取数据的时候，芯片使能信号 CEB 处于低电平（有效状态），写使能信号处于高电平（无效状态），芯片处于读状态，地址信号 A 在时钟信号 CK 的上升沿到来的前后都保持一段时间，输出信号 DO 在延时之后会得到相应的数据读出。其中



在时钟信号上升沿到来的前后都需保持一段时间的信号分别有地址信号 A、写操作使能信号和芯片使能信号 CEB。这三个信号在时钟信号上升沿到来之前保持的时间就是建立时间（Setup time），比如：Tas、Tws 和 Tces。相应的在时钟信号上升沿到来之后信号保持的时间就是保持时间（Hold time），比如 Tah、Twh 和 Tceh。除此之外时钟信号 CK 处于高电平的时间为 Thpw，处于低电平的时间为 Tlpw，时钟信号的周期为 Trc。输出信号 DO 从时钟信号上升沿到输出信号开始翻转的时间为 Toh，而从时钟信号上升沿到输出信号翻转结束的时间为 Tcq。其中 Tcq 越小说明 SRAM 工作越快。

图 5-2 Tcq 组合示意图

对于 SRAM 的延时 Tcq，它主要分布在三个方面：首先是译码延时，其次是存储阵列上的延时，另外还有时钟电路、放大电路、输出驱动等延时。

$$T_{cq} = T_{cq}(\text{input slew}) + T_{cq}(\text{instance}) + T_{cq}(\text{loading}) \quad (5.1)$$

由公式可以看出， $T_{cq}$  主要有三部分组成：输入的延时、阵列的延时和读出路径的延时，其中 Input slew 是 clock slew rate 和 data slew rate，loading 是 output loading。

其中  $tcq(\text{input slew})$  只取最大 instance 和最小 loading 时的值； $tcq(\text{instance})$  只取最小 Input slew 和最小 loading 时的值， $tcq(\text{load})$  取最小 cksr 和最大 inst 的值。

因为 Input Slew 和 output loading 都有 7 个数值，所以每一个 instance 的每一个时序都有  $7 \times 7$  的查询表。

### 5.1.1.2 写操作时序

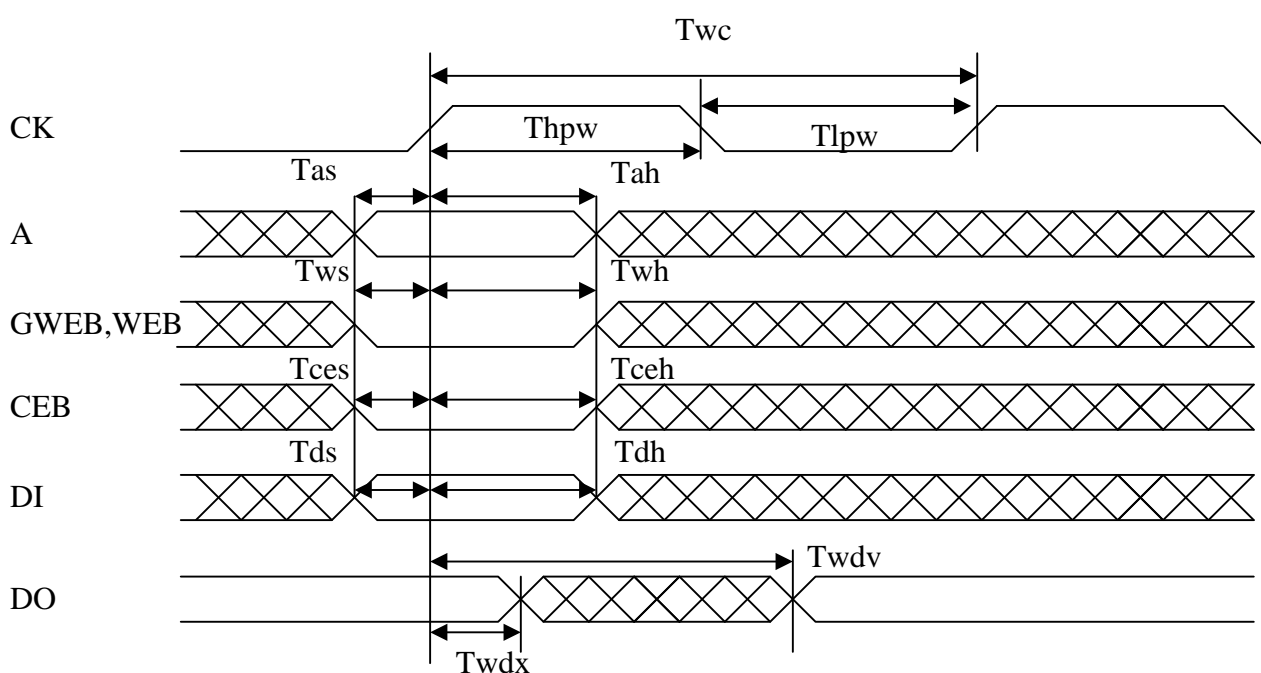


图 5-3 写操作波形示意图

表 5-2 写操作时序表

Parameter	Symbol
Write cycle time	$T_{wc}$
Address setup time before the CK rising edge	$T_{as}$
Address hold time after the CK rising edge	$T_{ah}$
GWEB, WEB setup time after the CK rising edge	$T_{ws}$
GWEB, WEB hold time after the CK rising edge	$T_{wh}$
Input data setup time before the CK rising edge	$T_{ds}$

Input data hold time after the CK rising edge	Tdh
CEB setup time before the CK rising edge	Tces
CEB setup time after the CK rising edge	Tceh
Clock high/low pulse widths	Thpw/Tlpw

SRAM 写操作时序如图，其中 WEB 为写使能，CK 是时钟信号，CEB 为片选中信号，A 为地址信号，DI 为数据输入信号，DO 为数据输出信号。

SRAM 写数据的时候，芯片使能信号 CEB 处于低电平（有效状态），写使能信号处于低电平（有效状态），地址信号 A 在时钟信号 CK 的上升沿到来的前后都保持一段时间，输出信号 DO 在延时之后会得到相应的数据读出。其中在时钟信号上升沿到来的前后都需保持一段时间的信号分别有地址信号 A、写操作使能信号、数据输入信号 DI 和芯片使能信号 CEB。这四个信号在时钟信号上升沿到来之前保持的时间就是建立时间（Setup time），比如：Tas、Tws、Tds 和 Tces。相应的在时钟信号上升沿到来之后信号保持的时间就是保持时间（Hold time），比如 Tah、Twh、Tdh 和 Tceh。除此之外时钟信号 CK 处于高电平的时间为 Thpw，处于低电平的时间为 Tlpw，时钟信号的周期为 Twc。

### 5.1.1.3 电源管理模式时序

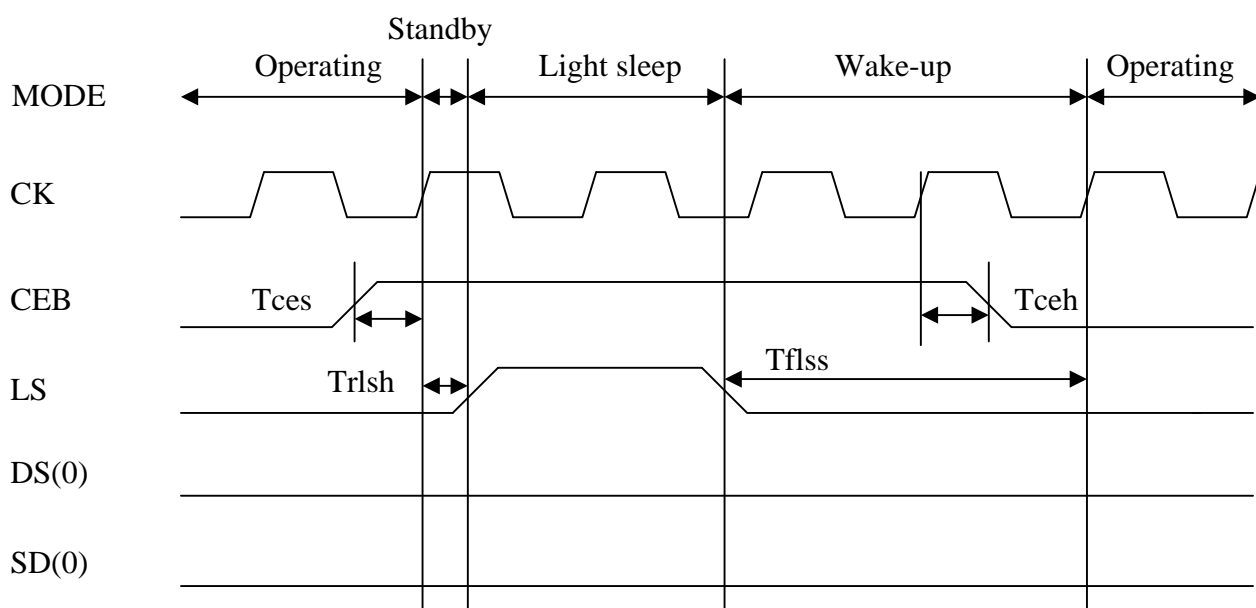


图 5-4 Light sleep 模式下电路各端口的波形示意图

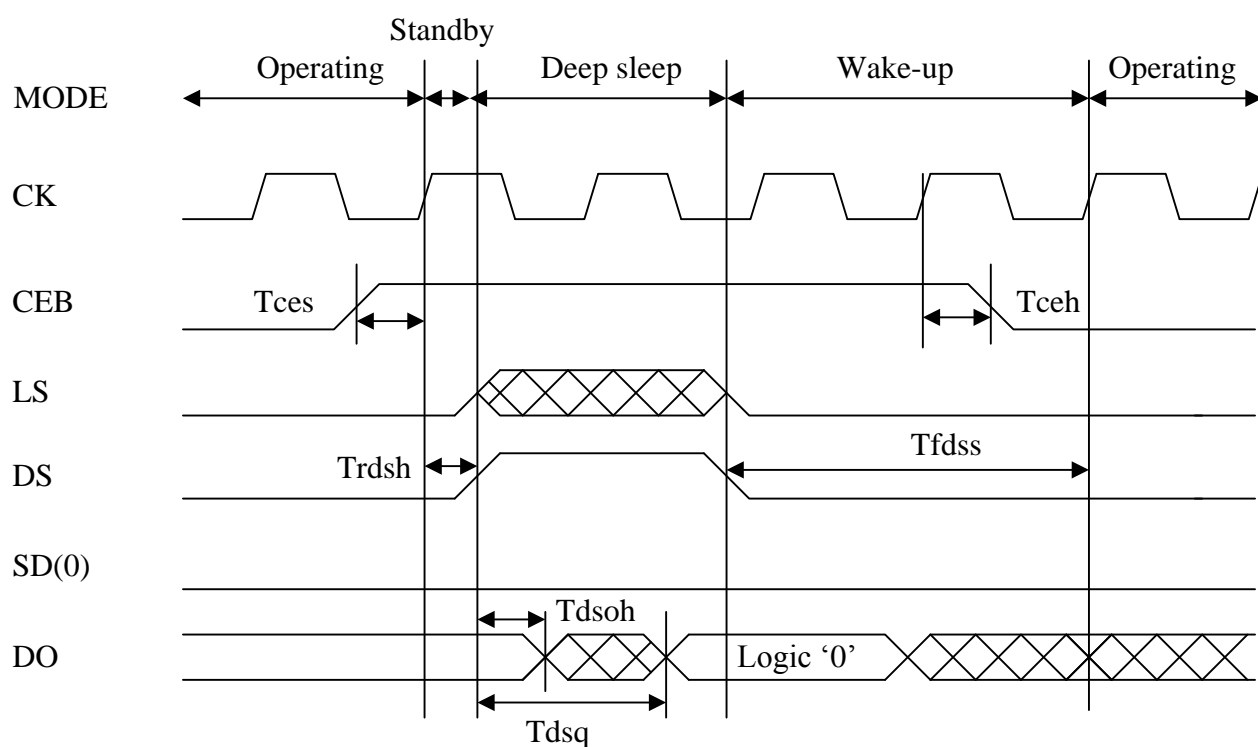


图 5-5 Deep sleep 模式下电路各端口的波形示意图

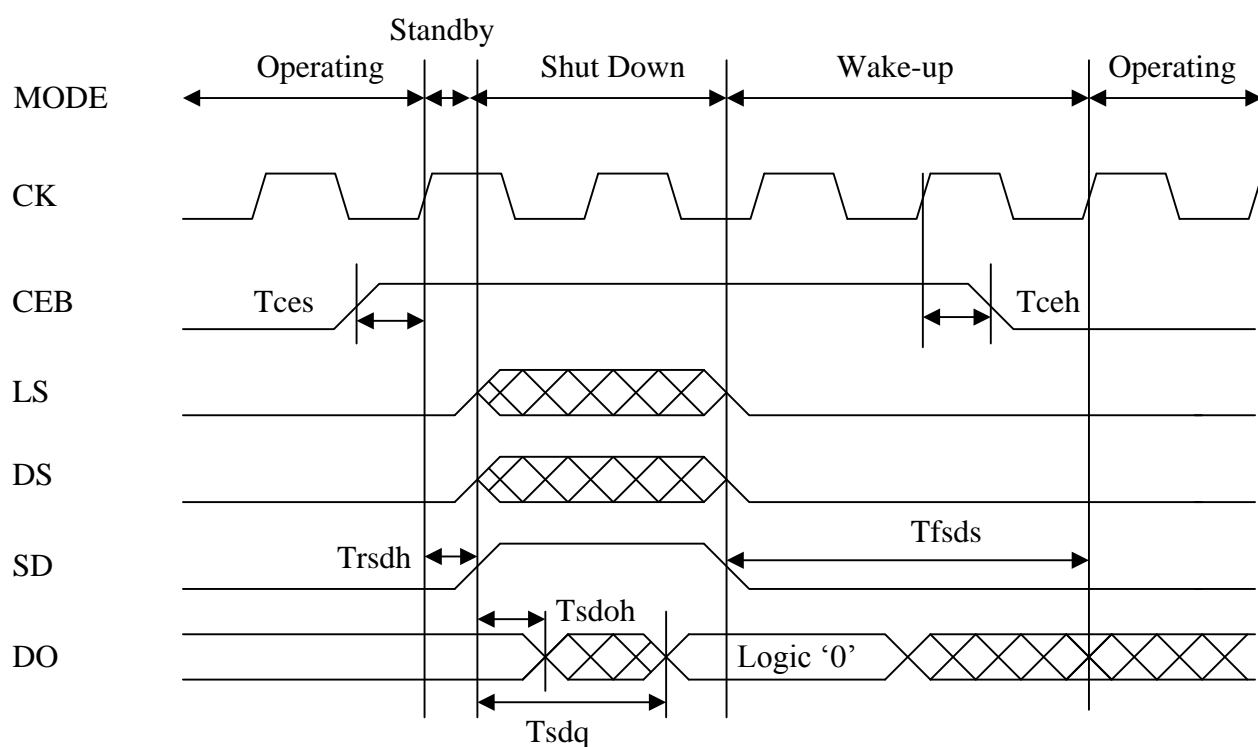


图 5-6 Shut down 模式下电路各端口的波形示意图

在 Light sleep 模式, SRAM 可以在不到一个完整的时钟周期之内恢复(wake up), 但其静态功耗也是三种模式中最大的; Shut down 模式具有最少的静态功耗, 但其恢

复时间是三种模式中最长的；Deep sleep 模式的静态功耗和恢复时间都是介于两者之间。

表 5-3 电源模式时序表

Parameter	Symbol
LS fall setup time before CK rises	Tflss
LS rise setup time before CK rises	Trlsh
DS fall setup time before CK rises	Tfdss
DS rise setup time before CK rises	Trdsh
DS rise to output hold previous data time	Tdsoh
DS rise to all output fall to logic '0' delay	Tdsq
SD fall setup time before CK rises	Tfsds
SD rise setup time before CK rises	Trsdh
SD rise to output hold previous data times	Tsdoh
SD rise to all output fall to logic '0' delay	Tsdq

以 W32B8M2B1 为例，当  $ck\ slew = 0.004\ ns$ ， $load = 0.001\ pF$ ， $data\ slew = 0.004\ ns$  时， $T_{cq}$  在 45 个 corners 下的值。

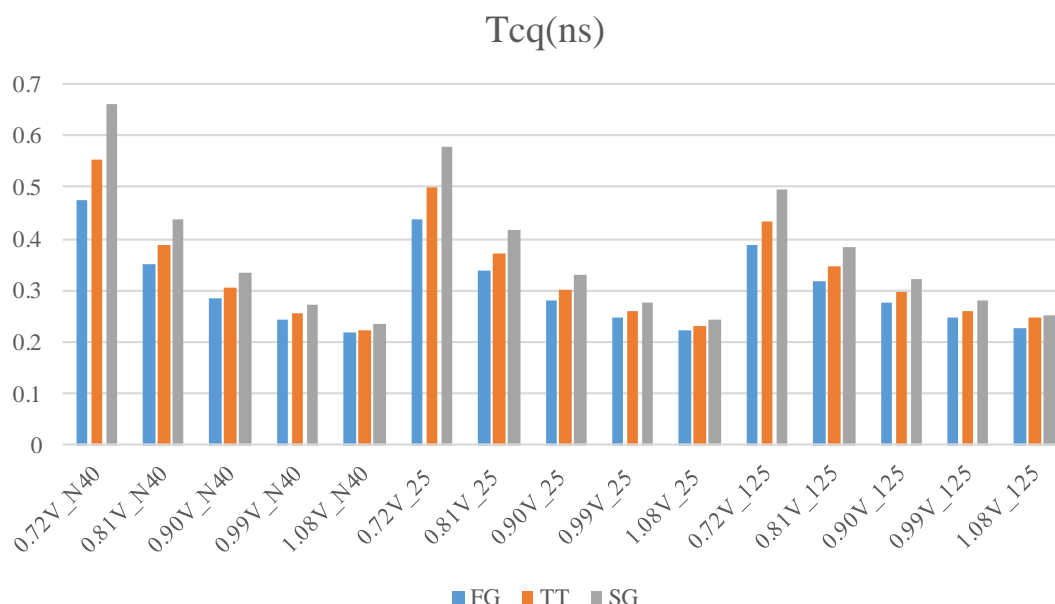


图 5-7 全 corners 组合下 Tcq 的时序柱状图

本课题分别从读操作时序、写操作时序和电源模式中选择 20 个时序，分别是 Tcq、Toh、Troslp、Tfoslpl、Tsdq、Tsdoh、Tdsq、Tdsch、Tas、Tah、Tds、Tdh、Tceh、Tces、Tws、Twh、Thpw、Tlpw、Trc、Twc。

## 5.1.2 时序预测

### 5.1.2.1 相同实例预测

本方法可以实现通过某一确定 Instance 的 CK Slew、DATA Slew 和 LOAD 的时序预测其他所有的 CK Slew、DATA Slew 和 LOAD 组合的时序。

本方法使用 ck slew=0.004 ns, data slew=0.004 ns, load=0.001 pF (组合 1) 的时序，去预测了其他不同 ck slew、data slew 和 load 组合的时序。目标组合一共有 7 组，分别是：

组合 2: ck slew=0.004 ns, load=0.001 pF, data slew=0.500 ns;

组合 3: ck slew=0.004 ns, load=0.500 pF, data slew=0.004 ns;

组合 4: ck slew=0.004 ns, load=0.500 pF, data slew=0.500 ns;

组合 5: ck slew=0.500 ns, load=0.001 pF, data slew=0.004 ns;

组合 6: ck slew=0.500 ns, load=0.001 pF, data slew=0.500 ns;

组合 7: ck slew=0.500 ns, load=0.500 pF, data slew=0.004 ns;

组合 8: ck slew=0.500 ns, load=0.500 pF, data slew=0.500 ns。

本次预测采用 8 组 corners 作为训练集: FG1P08VN40C、FG0P9V25C、FG1P08V25C、TT0P9VN40C、TT0P9V125C、SG0P81VN40C、SG0P99VN40C 和 SG0P9V125C。其他的 37 组 Corners 作为测试集。

以 CKslew=0.500 ns, load=0.500 pF, data slew=0.500 ns 为目标组合, 并对数据集进行最小二乘回归拟合, 如下式 5.2 和 5.3:

$$h(x) = \sum_{i=0}^n \theta_i x_i = \theta^T x \quad (5.2)$$

$$\theta = (X^T X)^{-1} X^T \vec{y} \quad (5.3)$$

其中 $x$ 是训练集的特征: 组合 1 的 8 corners $\times$ 20 timing 的数据集,  $\vec{y}$ 是训练集的目标: 组合 8 的 8corners $\times$ 20timing 的数据集。通过上述公式求得 $\theta$ 。并通过式 5.2 得到的回归模型, 对测试集数据进行预测, 并对测试集的目标数据与预测数据进行对比, 其测试集的预测时序与仿真真实时序的对比图如图 5-8。

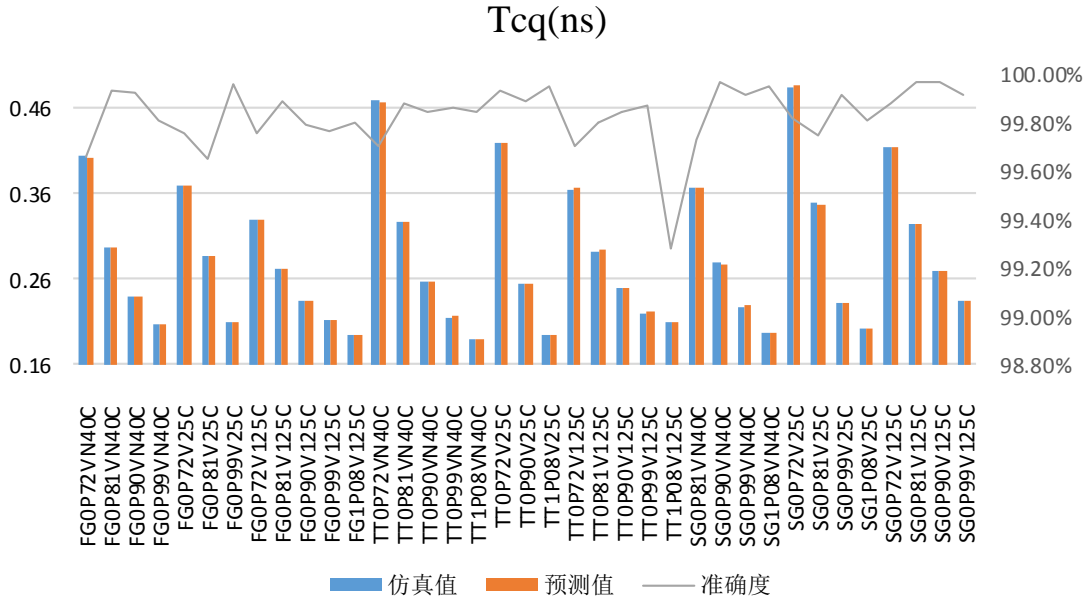


图 5-8 ck slew=0.500 ns, load=0.500 pF, data slew=0.500 ns 时的测试集 corners 的 Tcq 的仿真真实值与预测值及其预测准确度

从图中可以看出 37 组 Corners 下 Tcq 的仿真值和预测值相差很小, 可以从预测的准确度来观察仿真值和预测值的差异。



例中对测试集的 37 组数据的准确度求平均，得到本组 Tcq 预测的平均准确度为 99.83%。

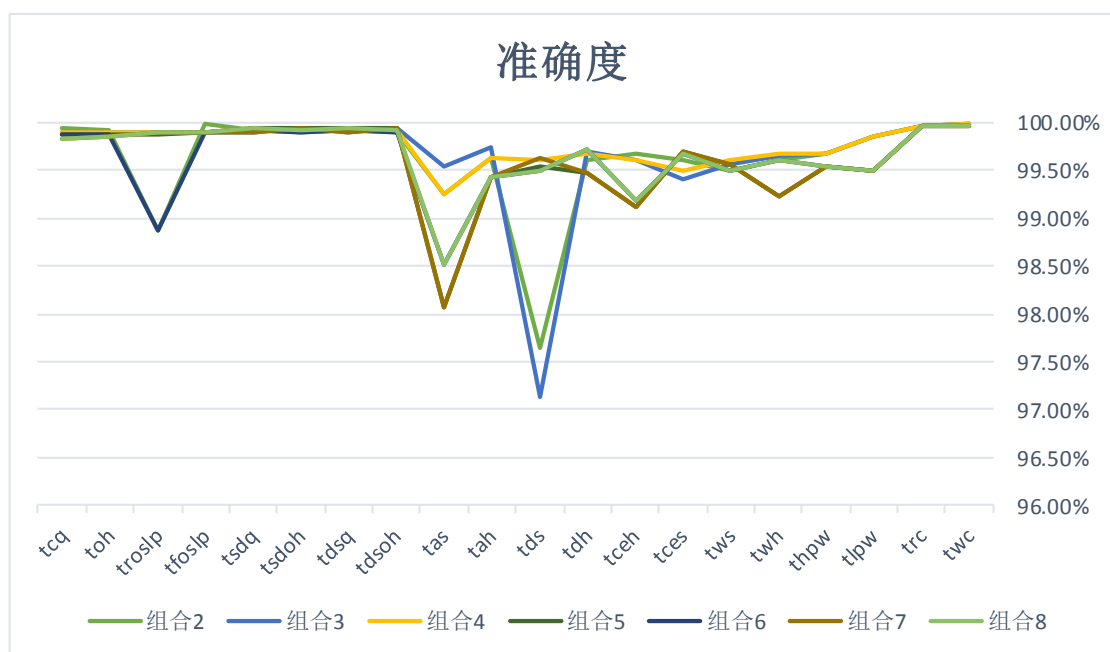


图 5-9 测试集 corners 的全时序仿真数据的预测平均准确度

上图为目标组合的测试集所有时序的平均准确度。可以从图中观察到七个目标组合的测试集在 37 组 PVT Corners 下的时序的平均准确度都处于 97% 以上，其中组合 3 和组合 5 的 Tds 的预测准确度约为 97%，除此之外所有的平均准确度都达到 98% 以上，而且大部分的时序的预测平均准确度可以达到 99%。由此可以得出结论：本方法在台积电 28nm 工艺下的同一 Instance，通过某一确定的 Input slew 和 output loading 的时序仿真数据，实现其他 Input slew 和 output loading 组合的时序仿真数据的预测，可以预测得到准确度很高的时序仿真数据，同时可以很有成效的减少仿真工作量，节省仿真时间和仿真资源。

### 5.1.2.2 不同实例预测

本课题中采选用了 4 个 instance，他们分别是 W32B8M2B1、W32B144M2B1、W512B8M2B1 和 W512B144M2B1。Instance 的形状与阵列的大小密切相关，从版图我们可以看出存储阵列几乎占据了 SRAM 版图 98% 的面积。4.1 中讲到通常阵列会被设计成行数较少、列数较多的结构，因为位线的负载是通过并联的，过长的位线会使得 SRAM 读写的速度变得很慢。相同的存储容量的 SRAM 会因为其字线和位线的

不同具有不同的读写速度，也就是时序。

Instance 的形状与字数和位数线性相关，从版图上看 Instance 总是矩形，其宽通常为 Word/Mux，其长通常为 Bit/Mux。

表 5-4 4 个 Instance 的特征、存储容量

Inst	W32B8M2B1	W32B144M2B1	W512B8M2B1	W512B144M2B1
Word×Bit	16×16	16×288	256×16	256×288
特征	面积最小	低长	高窄	面积最大
存储容量	0.4MB	4.5MB	4MB	72MB

本时序预测方法通过某个 Instance 的全 corner 来预测另一个 instance 的时序。可以实现相同 CK slew、Data slew、Load 的时候，通过单个 instance 的全 corner 时序以及目标 instance 的 6、8 组 corners 的时序，可以实现预测其他 corners 的时序。

## 方法一：

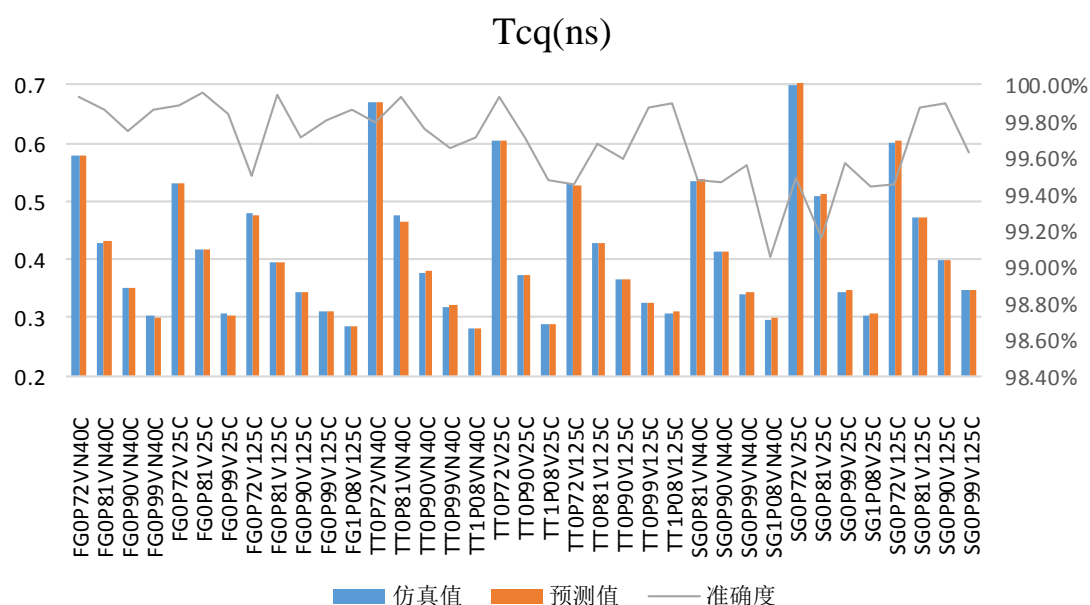


图 5-10 W32B144M2B1 的测试集目标 Tcq 的仿真值、预测值以及预测准确度

通过 W32B8M2B1 和 W32B144M2B1 的 8 组 corners 时序数据作为训练集，其中 W32B8M2B1 的 8 组 corners 时序数据作为训练集的特征(Feature)，而 W32B144M2B1 的 8 个 corners 时序数据作为训练集的目标(Target)。W32B8M2B1 和 W32B144M2B1 的剩余 37 组时序数据作为测试集，其中 W32B8M2B1 的 37 组时序数据作为测试集的特征，W32B144M2B1 的 37 组时序数据作为测试集的目标。使用最小二乘法对训

训练集特征和目标进行拟合，然后测试集的特征作为输入，得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的 8 组 Corners 分别是 FG1P08VN40C、FG0P9V25C、FG1P08V25C、TT0P81V25C、TT0P99V25C、SG0P72VN40C、SG0P9V25C 和 SG1P08V125C。

如图是本次预测的 W32B144M2B1 的测试集的 Tcq 的仿真值、预测值以及预测准确度。对 39 组 Tcq 的预测准确度进行求平均值，得到本次学习的平均预测准确度为 99.69%。从表中也可以看出所有 Tcq 的预测准确度均处于 99% 以上，预测效果很好。

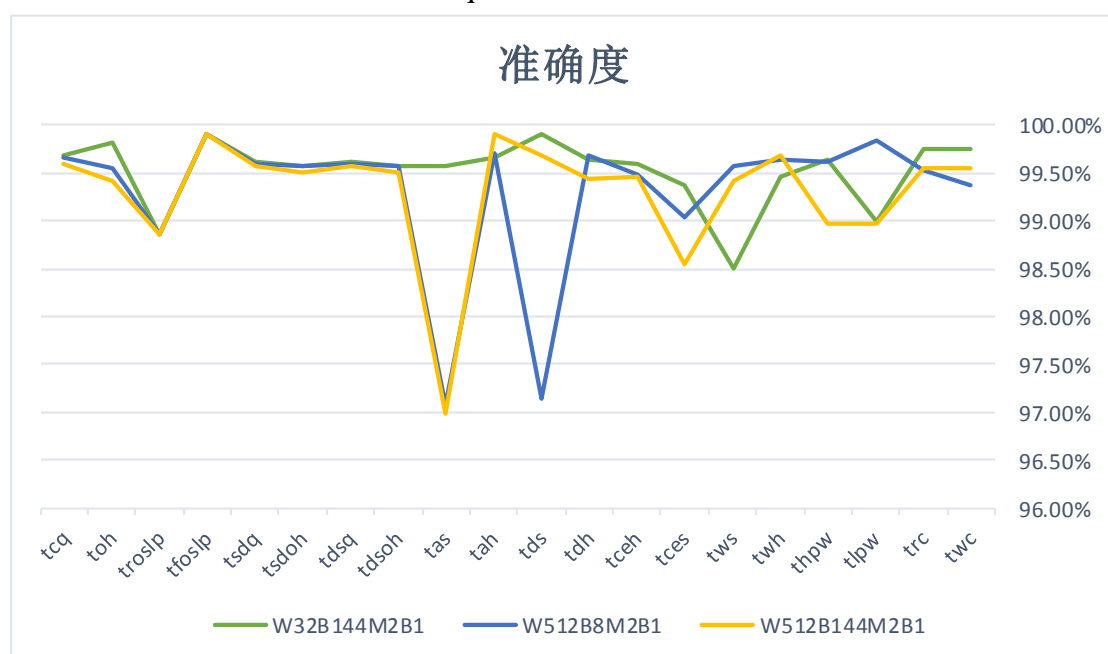


图 5-11 三个目标 Instances 测试集的全时序预测平均准确度

上图为 W32B144M2B1、W512B8M2B1 和 W512B144M2B1 的测试集的所有时序的平均准确度折线图。从折线图中可以观察到三组 Instance 的测试集在 37 组 PVT Corners 下的时序的平均准确度都处于 97% 以上，其中 W32B144M2B1 和 W512B144M2B1 的 Tas 的预测平均准确度约为 97%，W512B8M2B1 的 Tds 的预测平均准确度约为 97%，除此之外所有的平均准确度都达到 98.5% 以上，而且大部分的时序的预测平均准确度可以达到 99%。

## 方法二：

本方法通过选择 W32B8M2B1 和 W32B144M2B1 的 6 组 corners 时序数据作为训

练集，其中 W32B8M2B1 的 6 组 corners 时序数据作为训练集的特征，而 W32B144M2B1 的 6 个 corners 时序数据作为训练集的目标。W32B8M2B1 和 W32B144M2B1 的剩余 39 组时序数据作为测试集，其中 W32B8M2B1 的 39 组时序数据作为测试集的特征，W32B144M2B1 的 39 组时序数据作为测试集的目标。使用最小二乘法对训练集特征和目标进行拟合，然后测试集的特征作为输入，得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的 6 组 Corners 分别是 FG1P08VN40C、FG1P08V25C、TT0P81V25C、TT0P99V25C、SG0P72VN40C 和 SG0P72VN40C。

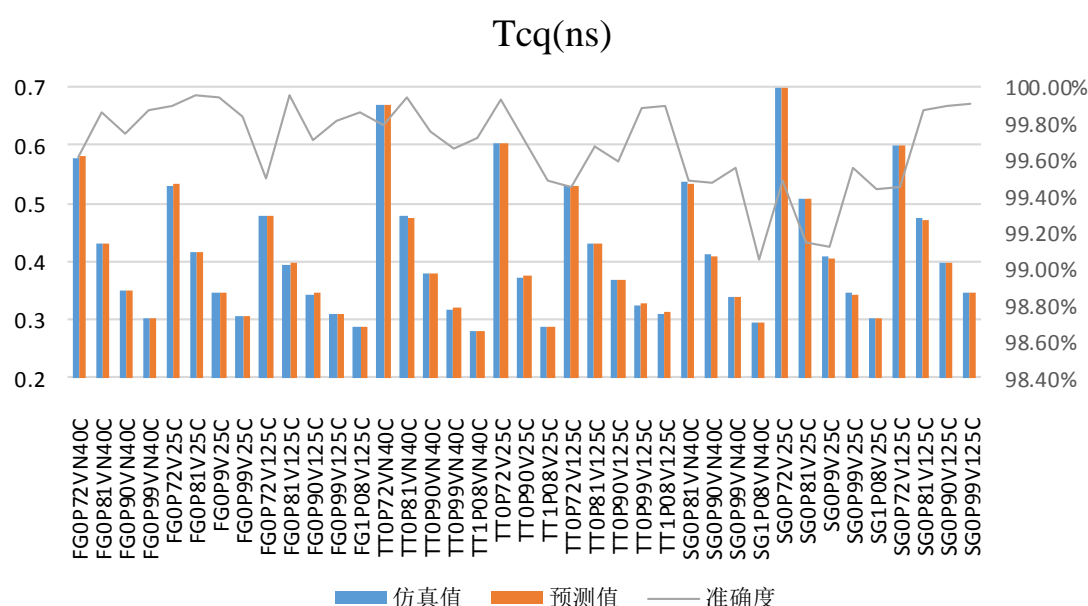


图 5-12 W32B144M2B1 的测试集目标 Tcq 的仿真值、预测值以及预测准确度

如图是本次预测的 W32B144M2B1 的测试集的 Tcq 的仿真值、预测值以及预测准确度。对 39 组 Tcq 的预测准确度进行求平均值，得到本次学习的平均预测准确度为 99.68%。从表中也可以看出所有 Tcq 的预测准确度均处于 99% 以上，预测效果很好。

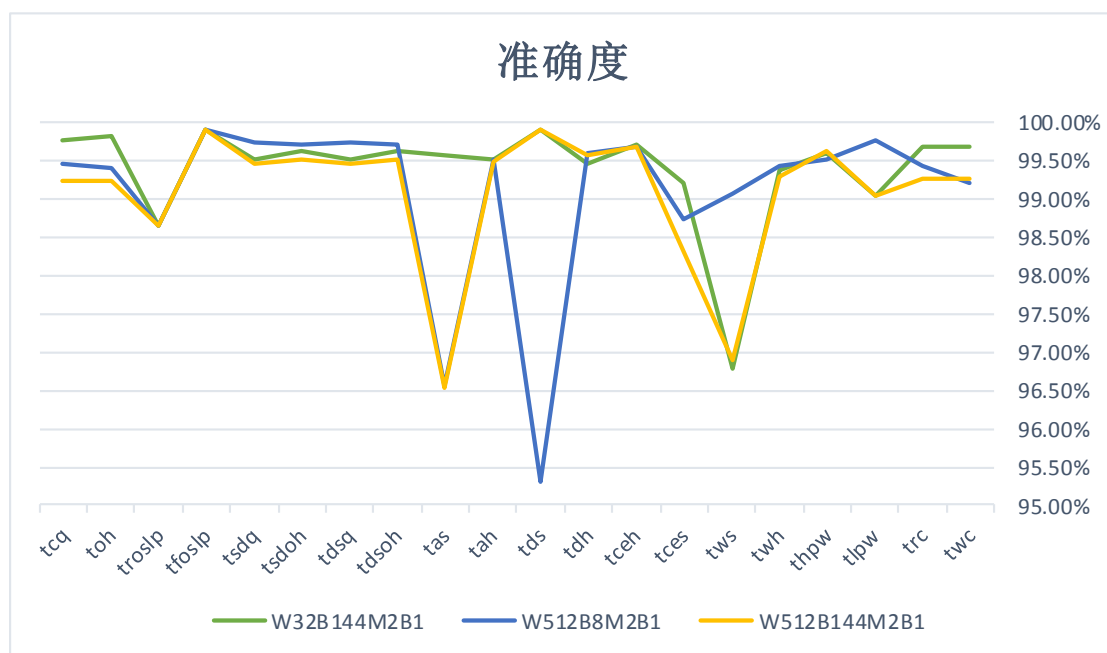


图 5-13 三个目标 Instances 测试集的全时序预测平均准确度

上图为 W32B144M2B1、W512B8M2B1 和 W512B144M2B1 的测试集的所有时序的平均准确度折线图。从折线图中可以观察到三组 Instance 的测试集在 39 组 PVT Corners 下的时序的平均准确度都处于 95% 以上，其中 W32B144M2B1 和 W512B144M2B1 的 Tas、Tws 的预测平均准确度约为 97%，W512B8M2B1 的 Tds 的预测平均准确度约为 95%，除此之外所有的平均准确度都达到 98.5% 以上，而且大部分的时序的预测平均准确度可以达到 99%。

### 方法三：

通过 W32B8M2B1 和 W32B144M2B1 的 8 组 corners 时序数据作为训练集，其中 W32B8M2B1 的 8 组 corners 时序数据作为训练集的特征，而 W32B144M2B1 的 8 个 corners 时序数据作为训练集的目标。W32B8M2B1 和 W32B144M2B1 的剩余 37 组时序数据作为测试集，其中 W32B8M2B1 的 37 组时序数据作为测试集的特征，W32B144M2B1 的 37 组时序数据作为测试集的目标。使用最小二乘法对训练集特征和目标进行拟合，然后测试集的特征作为输入，得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的 8 组 Corners 分别是 FG0P81VN40C、FG0P99VN40C、FG0P9V25C、TT0P9VN40C、TT0P9V125C、SG0P81VN40C、SG0P99VN40C 和 SG0P9V125C。

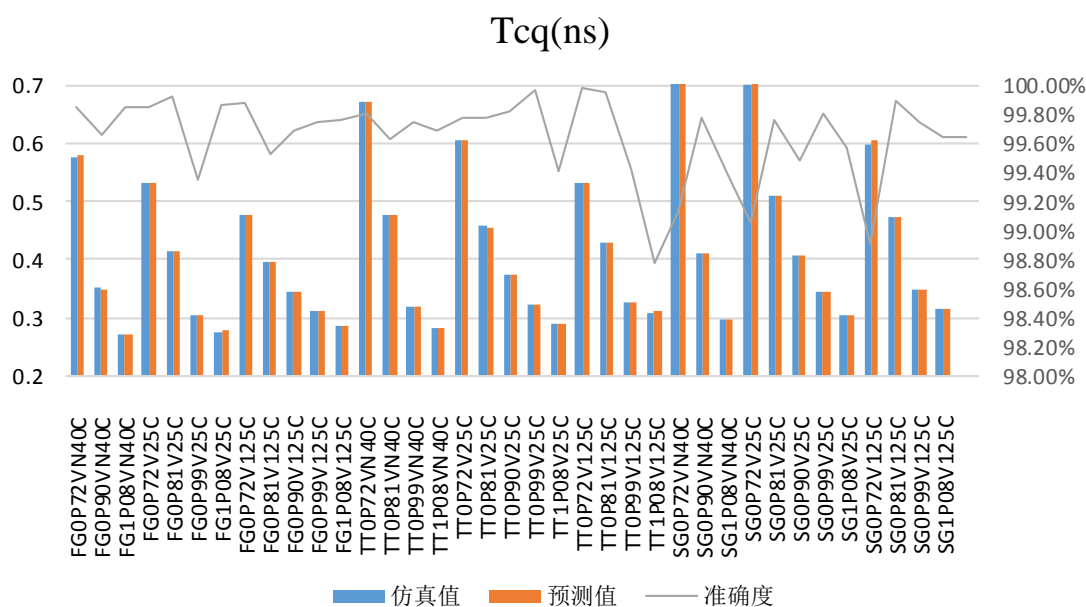


图 5-14 W32B144M2B1 的测试集目标 Tcq 的仿真值、预测值以及预测准确度

如图是本次预测的 W32B144M2B1 的测试集的 Tcq 的仿真值、预测值以及预测准确度。对 37 组 Tcq 的预测准确度进行求平均值，得到本次学习的平均预测准确度为 99.64%。从表中也可以看出所有 Tcq 的预测准确度均处于 98.8% 以上，预测效果很好。

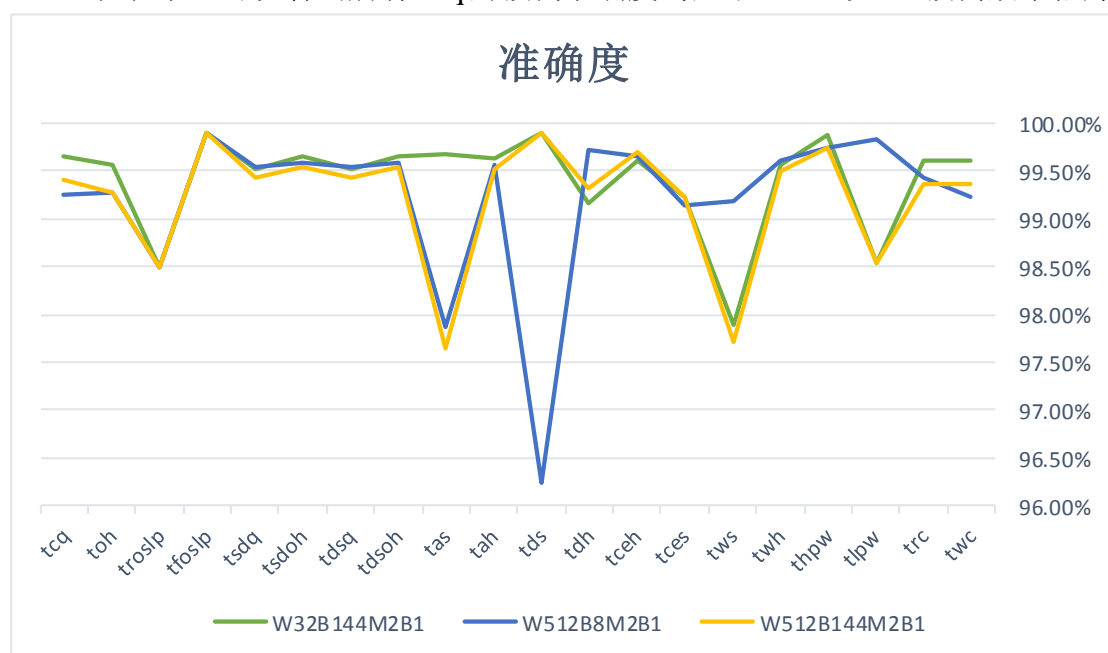


图 5-15 三个目标 Instances 测试集的全时序预测平均准确度

上图为 W32B144M2B1、W512B8M2B1 和 W512B144M2B1 的测试集的所有时序的平均准确度折线图。从折线图中可以观察到三组 Instance 的测试集在 37 组 PVT Corners 下的时序的平均准确度都处于 96% 以上，其中 W32B144M2B1 和 W512B144M2B1 的 Tas 的预测平均准确度约为 97.5%，W512B8M2B1 的 Tds 的预测平均准确度约为 96%，除此之外所有的平均准确度都达到 98.5% 以上，而且大部分的时序的预测平均准确度可以达到 99%。

## 方法比较

在以上三个方法中，都是通过某一确定 Input slew 和 output loading 的 Instance 的时序仿真数据，以及目标 Instance 的部分 corners 的时序仿真数据，通过最小二乘法进行回归训练，并通过训练模型进行测试集的预测。

在以上三个方法中，方法一是用 8 个 corners 进行训练集范围内部预测，方法二使用 6 个 corners 进行训练集范围内部预测，方法三通过 8 个 corners 进行全 corners 范围预测。三种方法的 Tcq 预测准确度的最小值分别是 99.05%、99.05%、98.77。三种方法的所有时序的预测平均准确度的最小值分别是 96.99%、95.31%、96.24%。

从 Tcq 的准确度和全时序的平均准确度来看，方法一适用于有较多 corners 仿真数据的目标 Instance，且已有 corners 是处于 PVT corners 的边缘，并对预测时序数据有非常高准确度和精确度的情况；方法二适用于有较少 corners 仿真数据的目标 Instance，且已有 corners 是处于 PVT corners 的边缘，并对预测时序数据有比较高的准确度和精确度的情况；方法三适用于有较多 corners 仿真数据的目标 Instance，且已有 corners 不是处于 PVT corners 的边缘，并对预测时序数据有非常高准确度的情况。

由此可以得出结论：本方法在台积电 28nm 工艺下某一确定的 Input slew 和 output loading 的某一 Instance 的时序仿真数据，实现其他 Instance 的时序仿真数据的预测，提供了三种使用不同情况的预测方法，且三种方法都可以得到准确度很高的时序仿真数据。

以本套 SRAM Compiler 为例，以 50 个 Intel (R) Xeon® 频率为 3.60MHZ 的 4 核 CPU，其全 corners 时序表征仿真工作量需要大致 340 天。在本时序预测方法应用于仿真工作中，本套 SRAM Compiler 全部仿真数据获得大概为 120 小时。缩减了约 65% 的仿真时间。由此可见，本时序预测方法可以很有成效的减少仿真工作量，同时节省仿真时间和仿真资源。

## 5.2 功率

### 5.2.1 功率数据

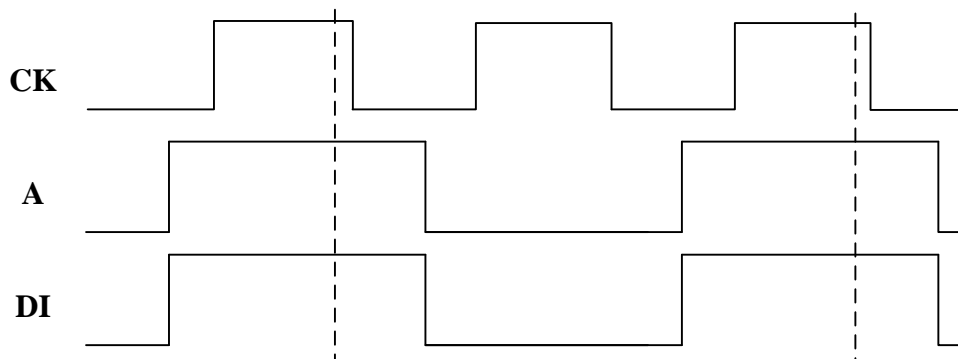


图 5-16 CK、A、DI 的波形示意图

如图 5-16 所示，读电流的测量时间从 CK 的下降沿到下一个下降沿，在读周期中只有 DI、A、CK 在翻转，其他的 Pin 都是 standby，且 A 和 DI 应该变化两次。

写电流的测量时间从 CK 的下降沿到 CK 的下一个下降沿，在写周期中只有 DI、A、CK 在翻转，其他的 Pin 都处于 standby，且 A 和 DI 都翻转两次。

在 light sleep mode、deep sleep mode 和 shut down mode 下，所有的 Pin 都保持原值，然后在分别在 light sleep、deep sleep 和 shut down 的最后几个周期，选择两个周期测量。

在功耗中主要需要采集的功率数据主要有：

1. dc: standby 状态下的电流的平均值
2. ac\_read: 读操作电流平均值
3. ac\_write: 写操作电流平均值
4. ls: light sleep 模式下电流平均值
5. ds: deep sleep 模式下电流平均值
6. sd: shut down 模式下电流平均值

通常为了节省功耗和减少数据读取时间，存储器中的字数会远远多于每个字存储的位数，这样会使得存储器变得长而窄，因为位线过长会增加数据访问的速度。字深的长度和位长的大小对功率也有同样的影响。本方法考虑到样本的多样性，以及数据的差异性，采用 4 个形状各异的 instance 作为训练集，如图上上：W32B8M2B1、



W32B144M2B1、W512B8M2B1 和 W512B144M2B1 分别是字长的极大值和极小值、位深的极大值和极小值的组合，也就代表了不同字长、位深对于功耗大小响应的极致体现。

本方法共采集了 18 个 instance 的全 corner 仿真数据，通过程序处理和运算得到每组仿真数据分别为 6 个参数。共得到 18 组 45×6 的数据集。鉴于功率参数的复杂性，本方法的训练集采用 4 个 instance 的数据集，以其他 14 个 instance 的数据作为测试集，分别进行了 14 次回归学习和预测。分别是 W80B8M2B1、W128B8M2B1、W192B8M2B1、W256B8M2B1、W320B8M2B1、W384B8M2B1、W448B8M2B1、W80B144M2B1、W128B144M2B1、W192B144M2B1、W256B144M2B1、W320B144M2B1、W384B144M2B1、W448B144M2B1 等 14 个 Instances。

因为功率参数受驱动电压波动的影响会有很明显的波动，通常在仿真阶段 SRAM Compiler 会选择三种电压作为仿真基准电压，分别是 HVT (High Voltage Threshold)、SVT (Standard Voltage Threshold) 和 LVT (Low Voltage Threshold)。通常功耗会随着阈值电压的增加而减少，就是相同条件下，HVT 的功耗最小，LVT 的功耗最大。

## 5.2.2 功率预测

以 SVT 为阈值电压进行台积电 28nm 工艺 SRAM Compiler 全特征的功耗仿真。

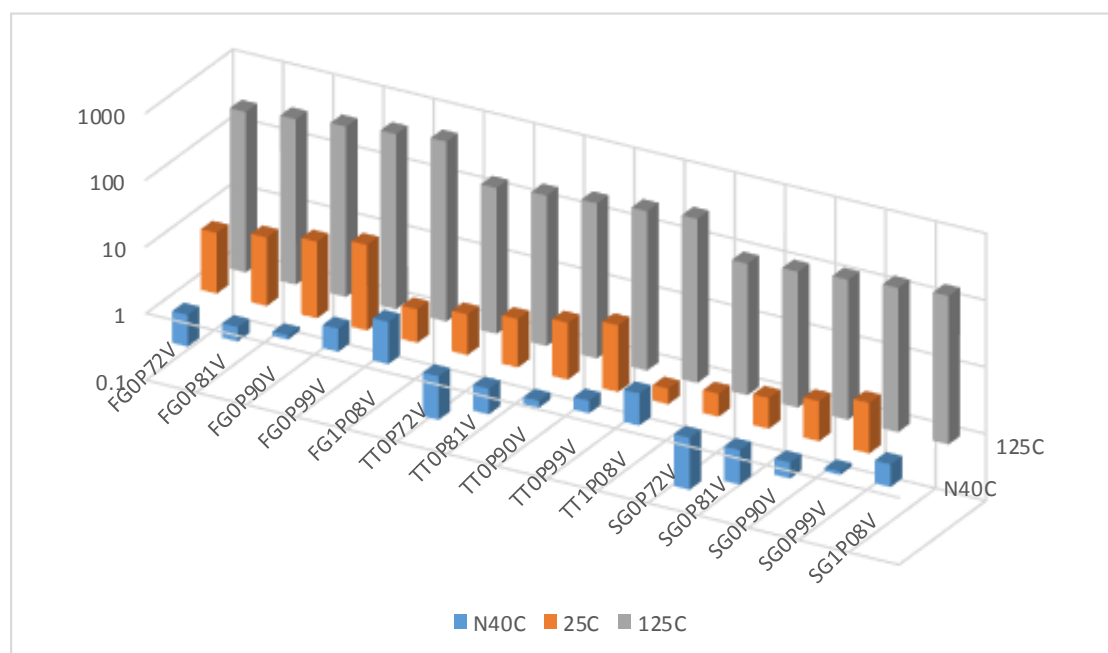


图 5-17 SVT 下 W512B8M2B1\_12BIMP 的全 corners 的 DC 电流值分布图

从图中可以看出图表中根据不同温度的 DC 数值在不同的量级，变化差异较大，故在此预测方法中对数据进行按温度分组进行回归学习与预测。

本方法将不同的温度分开进行学习。以 W80B8M2B1 为目标例，

### 5.2.2.1 125°C

本次回归学习中，训练集 Instances 的功率数据作为特征，测试集 W80B8M2B1 的功率数据作为目标。其中训练集 Instances 的 5 组 corners 的数据作为训练集的特征，另外 10 组 corners 的数据作为测试集的特征；测试集 W80B8M2B1 的 5 组 corners 的数据作为训练集的目标，另外 10 组数据作为测试集的目标。使用最小二乘法对训练集特征和目标进行回归拟合，然后将测试集的特征作为输入，得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的 5 组 Corners 分别是 FG0P72V125C、FG1P08V125C、TT0P9V125C、SG0P72V125C、SG1P08V125C。

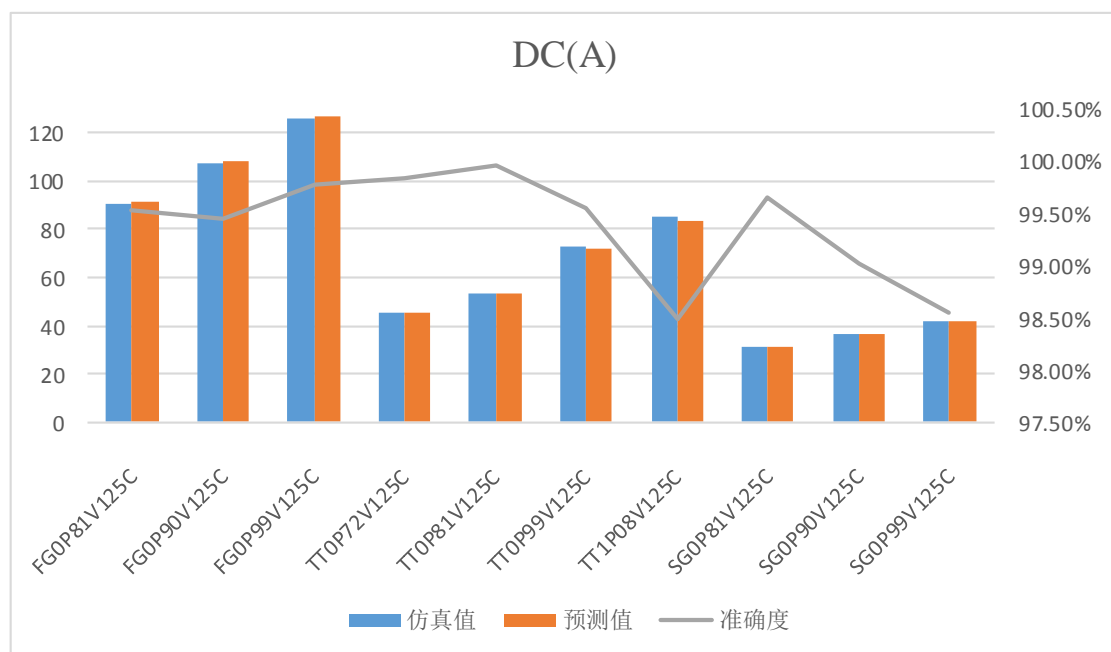


图 5-18 W80B8M2B1 的测试集目标 DC 的仿真值、预测值以及预测准确度

如图是本次预测的 W80B8M2B1 的测试集的 DC 的仿真值、预测值以及预测准确度。对 10 组 DC 的预测准确度进行求平均值，得到本次学习的平均预测准确度为 99.38%。从表中也可以看出所有 DC 的预测准确度均处于 98.5% 以上，预测效果很好。

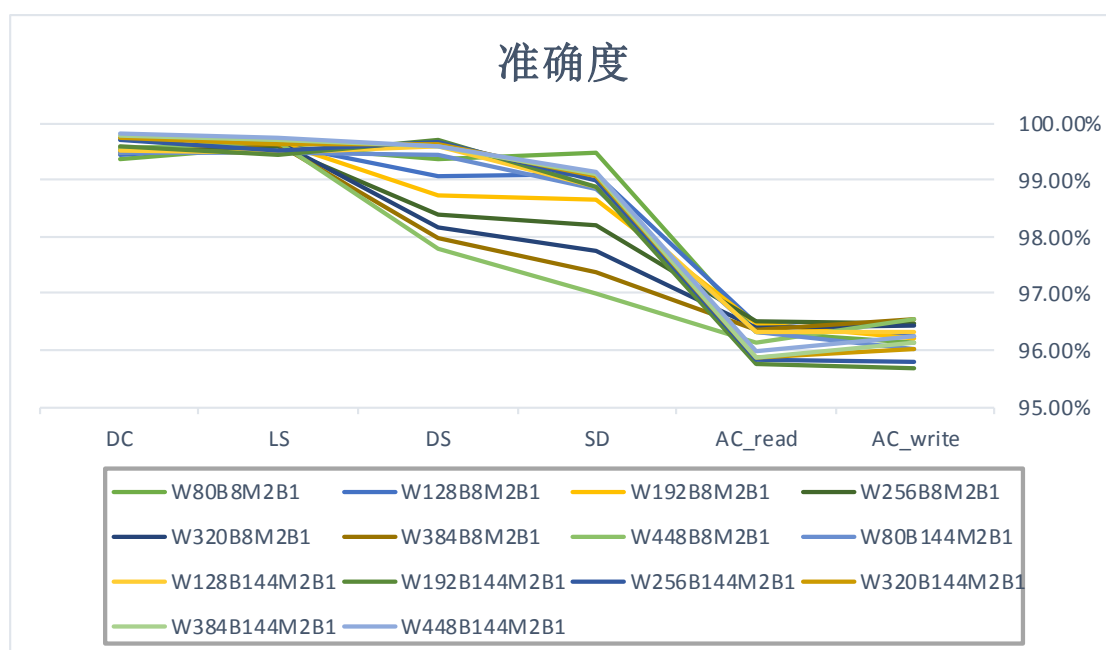


图 5-19 14 个目标 Instances 测试集的全功率预测平均准确度

上图为目标 Instances 测试集所有功率数据的平均准确度折线图。从折线图中可以观察到 14 组 Instances 的测试集在 10 组 PVT Corners 下功率数据的平均准确度都处于 95.5% 以上，其中所有 Instances 在 AC\_read 和 AC\_write 处的预测平均准确度约为 95.5%-96.5%，在 DC 和 LS 处的预测平均准确度约为 99.6%，在 DS 和 SD 处的预测平均准确度分布在 97%-99.5% 之间。

## 5.2.2.2 25°C

本次回归学习中，训练集 Instances 的功率数据作为特征，测试集 W80B8M2B1 的功率数据作为目标。其中训练集 Instances 的 5 组 corners 的数据作为训练集的特征，另外 10 组 corners 的数据作为测试集的特征；测试集 W80B8M2B1 的 5 组 corners 的数据作为训练集的目标，另外 10 组数据作为测试集的目标。使用最小二乘法对训练集特征和目标进行回归拟合，然后将测试集的特征作为输入，得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的 5 组 Corners 分别是 FG0P72V25C、FG1P08V25C、TT0P9V25C、SG0P72V25C、SG1P08V25C。

如图是本次预测的 W80B8M2B1 的测试集的 DC 的仿真值、预测值以及预测准确度。对 10 组 DC 的预测准确度进行求平均值，得到本次学习的平均预测准确度为

98.44%。从表中也可以看出所有 DC 的预测准确度均处于 97.5% 以上, 预测效果很好。

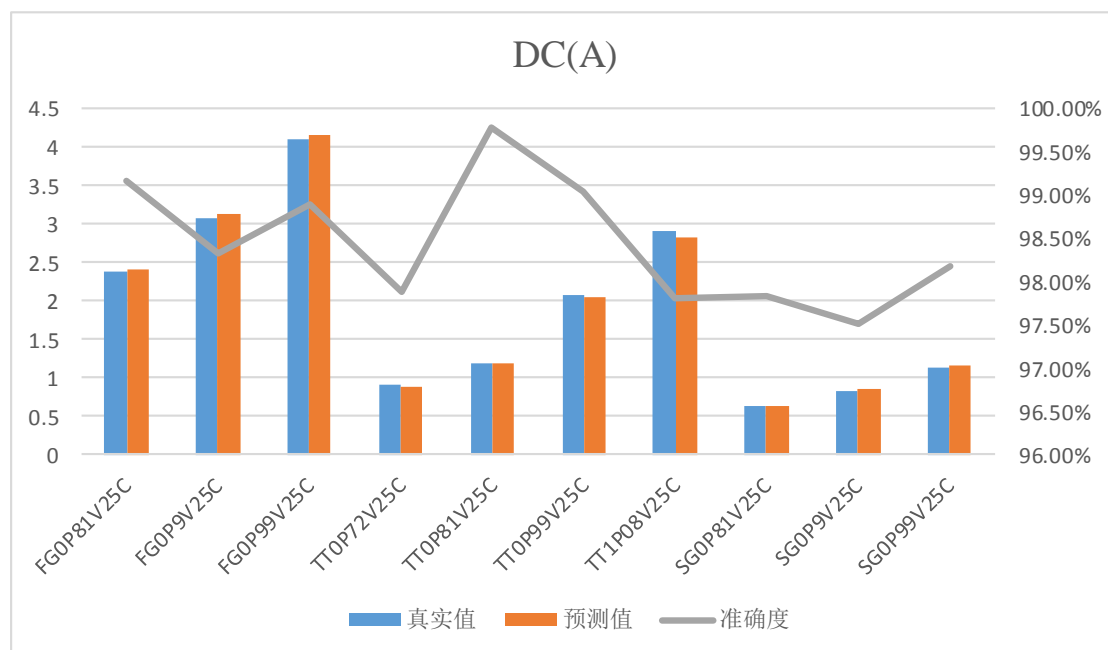


图 5-20 W80B8M2B1 的测试集目标 DC 的仿真值、预测值以及预测准确度

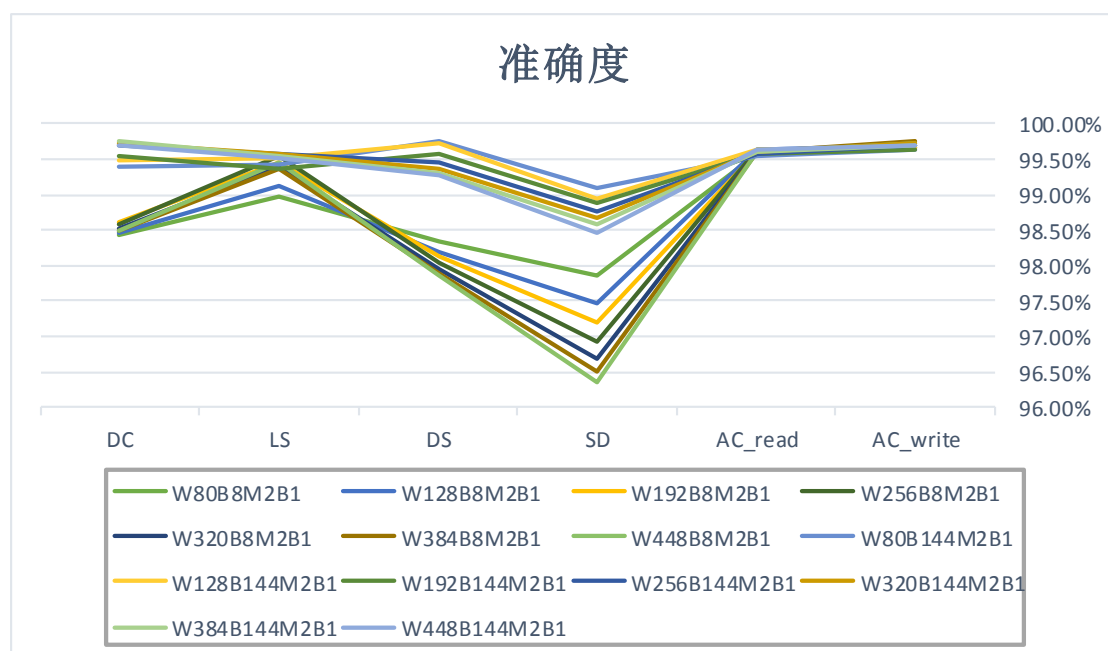


图 5-21 14 个目标 Instances 测试集的全功率预测平均准确度

上图为目标 Instances 测试集所有功率数据的平均准确度折线图。从折线图中可以观察到 14 组 Instances 的测试集在 10 组 PVT Corners 下功率数据的平均准确度都处于 96% 以上, 其中所有 Instances 在 AC\_read 和 AC\_write 处的预测平均准确度约为 99.6%, 在 DC 和 LS 处的预测平均准确度约为 98.5%~99.5%, 在 DS 和 SD 处的预

测平均准确度分布在 96%~99.5% 之间。

### 5.2.2.3 -40°C

本次回归学习中，训练集 Instances 的功率数据作为特征，测试集 W80B8M2B1 的功率数据作为目标。其中训练集 Instances 的 5 组 corners 的数据作为训练集的特征，另外 10 组 corners 的数据作为测试集的特征；测试集 W80B8M2B1 的 5 组 corners 的数据作为训练集的目标，另外 10 组数据作为测试集的目标。使用最小二乘法对训练集特征和目标进行回归拟合，然后将测试集的特征作为输入，得到测试集的目标预测值，并与测试集的目标真实值进行对比。

本方法中的训练集的 5 组 Corners 分别是 FG0P72VN40C、FG1P08VN40C、TT0P9VN40C、SG0P72VN40C、SG1P08VN40C。

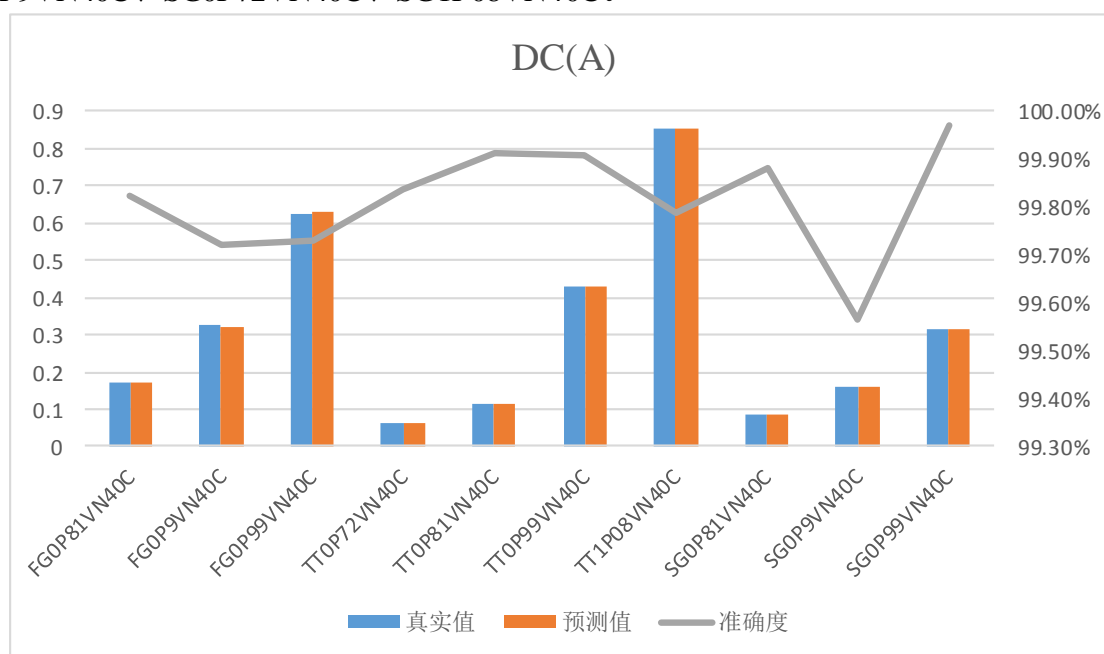


图 5-22 W80B8M2B1 的测试集目标 DC 的仿真值、预测值以及预测准确度

如图是本次预测的 W80B8M2B1 的测试集的 DC 的仿真值、预测值以及预测准确度。对 10 组 DC 的预测准确度进行求平均值，得到本次学习的平均预测准确度为 99.81%。从表中也可以看出所有 DC 的预测准确度均处于 99.5% 以上，预测效果很好。

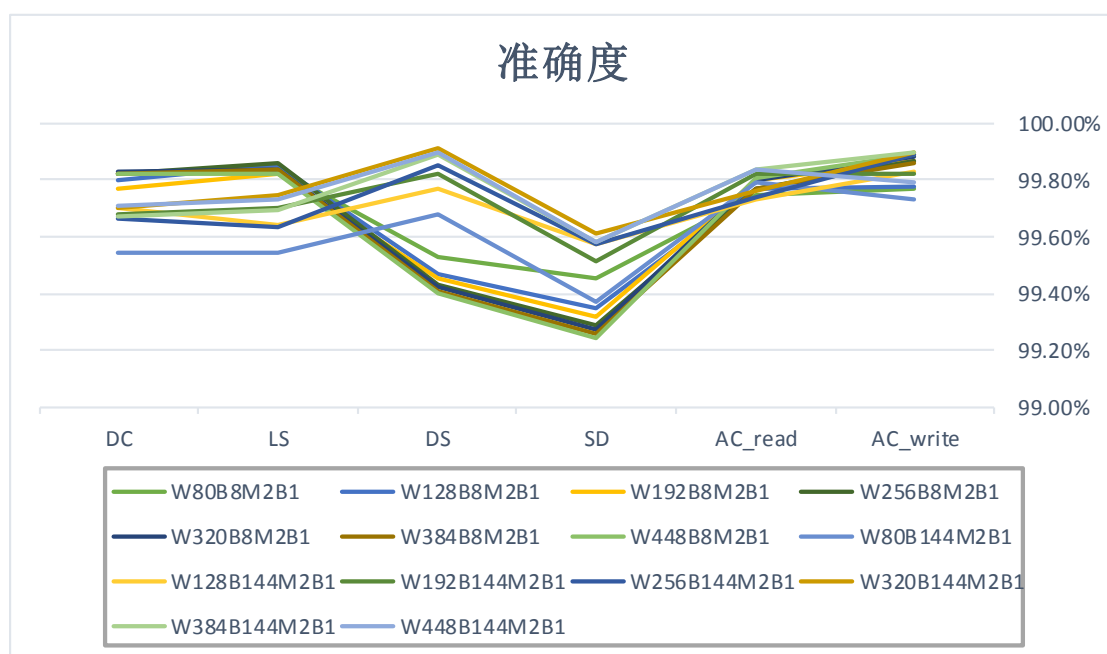


图 5-23 14 个目标 Instances 测试集的全功率预测平均准确度

上图为目标 Instances 测试集所有功率数据的平均准确度折线图。从折线图中可以观察到 14 组 Instances 的测试集在 10 组 PVT Corners 下功率数据的平均准确度都处于 99.2% 以上，其中所有 Instances 在 AC\_read 和 AC\_write 处的预测平均准确度约为 99.7%，在 DC 和 LS 处的预测平均准确度约为 99.5%~99.9%，在 DS 和 SD 处的预测平均准确度分布在 99.2%~99.9% 之间。

从以上 SVT 作为阈值电压的功耗数据预测中，针对 Process corners 的温度不同进行了分组学习。W80B8M2B1 中功耗 DC 在 125℃、25℃、-40℃时的最小预测准确度分别是 98.5%、97.5%、99.5%；由此我们可以推测本预测方法在预测 W80B8M2B1 的 6 个功耗参数的准确度都很高，并以此为据，进行了 14 个目标 Instances 的仿真，分别得到 125℃、25℃、-40℃时每组的 6 个功耗数据的平均预测准确度，其最小平均预测准确度分别是 95.5%、96%、99.2%。从预测平均准确度的结果来看，本方法应用于台积电 28nm 工艺的 SRAM Compiler 的表征功耗数据的预测具有很高的准确度，表明了通过最小二乘法进行回归学习与预测功耗数据的可行性与准确性。

以本套 SRAM Compiler 为例，以 50 个 Intel (R) Xeon® 频率为 3.60MHZ 的 4 核 CPU，其全 corners 功耗表征仿真工作量需要大致 720 小时。在本功耗预测方法应用于仿真工作中，本套 SRAM Compiler 全部仿真数据获得大概需要 200 小时。缩减了约 72% 的仿真时间。由此可见，本时序预测方法可以很有成效的减少仿真工作量，同时节省仿真时间和仿真资源。

### 5.3 本章小结

本章首先介绍了读操作、写操作和电源模式的时序含义，提出了两种时序预测方法：相同 Instances 预测、不同 Instances 预测。其中相同 Instances 预测可以实现预测相同 Instances 不同 Input slew 和 output loading 的时序数据；不同 Instance 预测提供三种方法实现相同 Input slew 和 output loading 的不同 Instances 的时序数据预测。且以上方法都能达到 97% 和 95.5% 以上的预测平均准确度。并且分析了应用本方法后本 SRAM Compiler 的时序仿真能节省 65% 的仿真时间。

本章的第二部分介绍了功率参数的含义，提出了基于温度分类的功率参数分类方法，且本方法能达到至少 95.2%、96%、99.2% 的预测平均准确度。并分析了应用本方法后本 SRAM Compiler 的功耗仿真能节省 72% 的仿真时间。

## 第六章 总结与展望

### 6.1 论文总结

现代 SOC 应用场景的不断拓展,对 SOC 不同的需求被提出,存储器作为 SOC 的重要部分,也对存储器提出了极高的要求,其中使用的 SRAM 在此背景下需要不断提升以适用于不同的需求,在各种高速、低功耗技术不断被提出的状态下,其对本身就复杂耗时的 SRAM 表征环节提出了更高的要求。

本文以联华电子 28nmHLP 工艺的标准单元库为例,提出了两种应用于标准单元库的时序预测方法:电路参数预测方法和时序预测方法。使用最小二乘回归算法预测了标准单元电路的时序,其中电路参数方法的预测时序的准确度能达到 87.36%;时序方法的预测时序的准确度能达到 98.68%。并从时间成本和机器资源成本对两种方法进行了评估,电路参数方法节约时间大致有 30%,时序方法节约时间大致有 50%。

以标准单元库时序预测的可行性为出发点,本文还提出了针对 SRAM Compiler 表征的数据预测方法。

本文还介绍了 SRAM 的基本框架和基本电路结构,针对影响表征仿真工作量的所有参数特征、word、bit、mux、bank、input slew、output loading 和 PVT corners 等进行了详细的阐述,针对不同仿真的需求:相同 Instance 不同 Input slew 和 output loading 之间的预测、相同 input slew 和 output loading 下不同 Instance 之间的预测。对两种需求分别提出了一种和三种预测方法,其中针对不同 instance 之间预测的三种方法分别具有不同的应用场景。从预测结果来看,相同 Instance 预测方法能达到最低 97% 的平均准确度;不同 Instance 预测方法分别能达到最低 96.99%、95.31%、96.24% 的平均准确度。从时间成本个机器成本进行了评估,本时序预测方法能节省仿真时间大约 67%。

最后对 SRAM Compiler 表征的功耗数据也提出了一种基于温度分类的功率参数分类方法,三个温度段的预测方法能达到至少 95.2%、96%、99.2% 的预测平均准确度。并分析了应用本方法后本 SRAM Compiler 的功耗仿真能节省 73% 的仿真时间。

### 6.2 工作展望

本文主要的工作集中在 SRAM Compiler 表征的时序与功耗的预测,本文的不足



以及后续工作包括：

- 一、本课题时序数据都是选取全特征的 Instance，未对单个或多个 Instance 特征进行仿真与预测工作，接下来会对这部分进行研究。
- 二、本课题功耗数据中只针对 SVT 下的功耗数据进行了预测工作，将来需要对 HVT 和 LVT 下的功耗数据进行预测研究。
- 三、本文采用台积电 28nmHPC+工艺的 SRAM Compiler 进行了表征数据预测，接下来还需要针对不同工艺的 SRAM Compiler 进行表征数据的预测。

## 参考文献

- [1] 移动智能终端暨智能硬件白皮书. 中国信息通信研究院, 2016.
- [2] 关立军. 基于 28nm 工艺低电压 SRAM 单元电路设计[D].安徽大学,2017.
- [3] 冯海玉. 中国集成电路市场的结构性分析[J].中国集成电路,2016,25(08):14-16+20.
- [4] Zhang K. Embedded Memories for Nano-Scale VLSIs [M]. Springer US, 2009.
- [5] Moore G E. The microprocessor: engine of the technology revolution [M]. ACM, 1997.
- [6] Narendra S, De V, Borkar S, et al. Full-chip subthreshold leakage power prediction and reduction techniques for sub-0.18- $\mu\text{m}$  CMOS[J]. IEEE Journal of Solid-State Circuits, 2004, 39(3):501-510.
- [7] De V, Borkar S. Technology and design challenges for low power and high performance [microprocessors][C] International Symposium on Low Power Electronics and Design, 1999. Proceedings. IEEE, 1999:163-168.
- [8] Burd T, Brodersen R. Design Issue for Dynamic Voltage Scaling [J]. In Proceedings of International Symposium on Low Power Electronics and Design (ISLPED, 2001:9 - 14.
- [9] Zhai B, Blaauw D, Sylvester D, et al. The limit of dynamic voltage scaling and insomnia dynamic voltage scaling [J]. IEEE Transactions on Very Large Scale Integration Systems, 2005, 13(11):1239-1252.
- [10] Burd T D, Pering T A, Stratakos A J, et al. Dynamic voltage scaled microprocessor system[J]. Solid-State Circuits, IEEE Journal of, 2000, 35(11):1571-1580.
- [11] Cho Y. Simultaneous optimization of battery-aware voltage regulator scheduling with dynamic voltage and frequency scaling[C] ACM/IEEE International Symposium on Low Power Electronics and Design. IEEE, 2010:309-314.
- [12] Lee J, Nam B G, Yoo H J. Dynamic Voltage and Frequency Scaling (DVFS) scheme for multi-domains power management[C] Solid-State Circuits Conference, 2007. ASSCC '07. IEEE Asian. IEEE, 2008:360-363.
- [13] Dutta U, Soni M K, Pattanaik M. A Review of NBTI Degradation and its Impact on the Performance of SRAM [J]. International Journal of Modern Education & Computer Science, 2016.

- [14] Liu Z, Kursun V. Characterization of a Novel Nine-Transistor SRAM Cell [J]. IEEE Transactions on Very Large Scale Integration Systems, 2008, 16(4):488-492.
- [15] Jagasivamani M, Dong S H. Development of a low-power SRAM compiler [M]. 2000.
- [16] El-Thakeb A T, Elhamid H A, Mostafa H, et al. Performance evaluation of finFET based SRAM under statistical VT variability[C] International Conference on Microelectronics. IEEE, 2015:88-91.
- [17] Hentrich D, Oruklu E, Saniie J. Performance evaluation of SRAM cells in 22nm predictive CMOS technology[C] IEEE International Conference on Electro information Technology. IEEE, 2009:470-475.
- [18] Gupta S, Gupta K, Pandey N. Performance evaluation of SRAM cells for deep submicron technologies[C] Computational Intelligence on Power, Energy and Controls with Their Impact on Humanity. IEEE, 2017:292-296.
- [19] 丁海涛. SRAM 设计和 Compiler 技术[D]. 合肥工业大学, 2003.
- [20] 谢凯翔. SRAM 失效分析[J]. 中国集成电路, 2017, 26(10):70-73.
- [21] She Y Q, Zhang L J, Zheng J B, et al. Standard Cell Library Characterization of 28nm Process Based on Machine Learning [J]. 2017(CST).
- [22] 柏娜. 超低能耗亚阈值 SRAM 电路设计[D]. 东南大学, 2011.
- [23] 朱贾峰. 低电压 SRAM 关键技术研究 with 实现[D]. 东南大学, 2014.
- [24] 黎轩. 低电压 SRAM 关键模块研究与编译器设计[D]. 中国科学技术大学, 2015.
- [25] 王旭. 低功耗 SRAM 电路的设计技术研究[D]. 上海交通大学, 2015.
- [26] 李二亮. 基于 Mix-IS 算法的 SRAM 设计及良率分析[D]. 苏州大学, 2015.
- [27] 胡玉青. 应用于 Cache 的 65nm 高速 SRAM 设计[D]. 苏州大学, 2016.
- [28] Hocking R R. A Biometrics Invited Paper. The Analysis and Selection of Variables in Linear Regression [J]. Biometrics, 1976, 32(1):1-49.
- [29] Wei L, Chin H C, Cheng S L, et al. Performance evaluation of 14 nm FinFET-based 6T SRAM cell functionality for DC and transient circuit analysis[J]. Journal of Nanomaterials, 2014, 2014(9):105.
- [30] Cook R D. Detection of Influential Observation in Linear Regression [J]. Technometrics, 2000, 42(1):65-68.
- [31] Montgomery D C, Peck E A, Vining G G. Introduction to linear regression analysis. 4th

- ED [J]. Journal of Applied Statistics, 1992, 40(12):2775-2776.
- Tanaka H, Uejima S, Asai K. Linear Regression Analysis with Fuzzy Model [J]. Systems Man & Cybernetics IEEE Transactions on, 1982, 12(6):903-907.
- [32] Wold S, Ruhe A, Wold H, et al. The Collinearity Problem in Linear Regression. The Partial Least Squares (PLS) Approach to Generalized Inverses [J]. Siam J.sci.comput, 1984, 5(3):735-743.
- [33] Elkan C. The Foundations of Cost-Sensitive Learning[C] Seventeenth International Joint Conference on Artificial Intelligence. 2001:973--978.
- [34] Morifuji E, Yoshida T, Kanda M, et al. Supply and threshold-Voltage trends for scaled logic and SRAM MOSFETs [J]. IEEE Transactions on Electron Devices, 2006, 53(6):1427-1432.
- [35] Morifuji E, Patil D, Horowitz M, et al. Power Optimization for SRAM and Its Scaling [J]. IEEE Transactions on Electron Devices, 2007, 54(4):715-722.
- [36] Cao Y, Sato T, Orshansky M, et al. New paradigm of predictive MOSFET and interconnect modeling for early circuit simulation[C] Custom Integrated Circuits Conference, 2000. Cicc. Proceedings of the IEEE. IEEE, 2000:201-204.
- [37] 李端梁. SRAM 编译器的设计研究[D]. 北京大学, 2009.
- [38] Zhao W, Cao Y. New Generation of Predictive Technology Model for Sub-45 nm Early Design Exploration [J]. IEEE Transactions on Electron Devices, 2006, 53(11):2816-2823.
- [39] Rao R M, Burns J L, Devgan A, et al. Efficient techniques for gate leakage estimation[C] International Symposium on Low Power Electronics and Design. IEEE, 2003:100-103.
- [40] Baker R J. CMOS: Circuit Design, Layout, and Simulation, Third Edition [M] CMOS Circuit Design, Layout, and Simulation, Second Edition. Wiley-IEEE Press, 2004:1231 - 1232.
- [41] 余晓文. SRAM 的设计验证[J]. 中国集成电路, 2004(11):16-21.
- [42] Weste N H E, Eshraghian K. Principles of CMOS VLSI design: A systems perspective [M] Principles of CMOS VLSI design: a systems perspective. Addison-Wesley Longman Publishing Co. Inc. 1985:520.
- [43] Rabaey J M, Chandrakasan A, Nikolic B. Digital Integrated Circuits [M]. 清华大学出版社, 2004.

- [44] 徐雅男. 90nm 工艺高速低功耗 SRAM 的设计[D]. 复旦大学, 2010.
- [45] 叶亚东. 低电压 SRAM 时序控制电路技术研究[D]. 安徽大学, 2015.
- [46] Caravella J S. A low voltage SRAM for embedded applications [J]. IEEE Journal of Solid-State Circuits, 2002, 32(3):428-432.
- [47] 黄义定, 李鉴, 李天阳,等. 高速低功耗 SRAM 中灵敏放大器的设计[J]. 电子器件, 2008, 31(5):1650-1653.
- [48] 李列文, 桂卫华. 面向 FPGA 的低泄漏功耗 SRAM 单元设计方法研究[J]. 高技术通讯, 2012, 22(12):1292-1298.
- [49] Tran H. Demonstration of 5T SRAM and 6T dual-port RAM cell arrays[C] VLSI Circuits, 1996. Digest of Technical Papers. 1996 Symposium on. IEEE, 1996:68-69.
- [50] Elrabaa M S, Elmasry M I. Low-power BiCMOS/ECL SRAM: US, US 5602774 A[P]. 1997.
- [51] Wang J S, Yang P H, Tseng W. Low-power embedded SRAM macros with current-mode read/write operations[C] International Symposium on Low Power Electronics and Design. ACM, 1998:282-287.
- [52] 阎辉, 张学工, 李衍达. 支持向量机与最小二乘法的关系研究[J]. 清华大学学报(自然科学版), 2001, 41(9):77-80.
- [53] Morimura H, Shibata N. A 1-V 1-Mb SRAM for portable equipment[C] International Symposium on Low Power Electronics and Design. IEEE, 2002:61-66.
- [54] Alowersson J, Andersson P. SRAM cells for low-power write in buffer memories[C] Low Power Electronics, 1995. IEEE Symposium on. IEEE, 1995:60-61.
- [55] 王晓媛. 1Mb 高速低功耗 SRAM 的设计[D]. 西安电子科技大学, 2008.
- [56] 吴秋雷. 低功耗 SRAM 存储单元关键技术研究及电路设计[D]. 安徽大学, 2013.
- [57] Morimura H, Shigematsu S, Konaka S. A shared-bitline SRAM cell architecture for 1-V ultra low-power word-bit configurable macrocells[C] International Symposium on Low Power Electronics and Design, 1999. Proceedings. IEEE, 1999:12-17.
- [58] Nambu H, Kanetani K, Yamasaki K, et al. A 1.8-ns access, 550-MHz, 4.5-Mb CMOS SRAM [J]. Solid-State Circuits, IEEE Journal of, 1998, 33(11):1650-1658.

## 攻读学位期间本人出版或公开发表的论著、论文

She Y Q, Zhang L J, Zheng J B, et al. Standard Cell Library Characterization of 28nm Process Based on Machine Learning[J]. 2017(CST).

## 致 谢

在此衷心感谢恩师张立军研究员！感谢张老师三年来的认真培养，张老师从论文课题的选定到论文初步完成，时刻关心着课题的进展，他在诸多方面都给予了极大的支持和启发。张立军老师的国际化视野，前沿而精髓的学术造诣，严谨勤奋的治学风格，都让我敬佩不已。

在此衷心感谢郑坚斌先生！在半导体存储器行业中非常资深的郑先生为我论文课题的践行提供了很好的平台，也提供了良多宝贵的意见和建议。

在此衷心感谢张爱林先生、诸月平先生和吴守道先生等多位共事于紫光展锐（苏州）通信有限公司 **Physical IP** 部门的各位前辈们！感谢张爱林先生在多种软件语言学习和课题探索过程中不厌其烦的指导和帮助；感谢诸月平先生在 **SRAM** 仿真工作方面给予的无私教学和帮助；感谢吴守道先生在 **SRAM Compiler** 设计方面的专业准确的辅导和帮助。

在此感谢 EDA 设计实验室的 10 位兄弟姐妹！感谢你们的相伴，感谢你们在我处于低潮的时候给予的安慰和鼓励。他们是李有忠、张其笑、胡玉青、姜伟、张炎峰、张振鹏、冯李、石文优、顾昌山和桑胜男。

在此感谢曾经在学院一起快乐生活的三位室友和两位同学！他们是徐浩然、吴楠、杨斌、郝欢和李炫。

记下你们的名字便不会忘。

在此衷心感谢十年好友孙嘉慷的陪伴！

在此衷心感谢多多的陪伴！

在此感谢父亲母亲无私的养育与包容！

最后，感谢时间！研究生攻读的三年对我而言，不只是学习与生活的三年，三年时间更让我深切的认知了人生轨迹的唯一与深刻。

还要感谢一个人，凌欢。

余一奇

2018 年 4 月于独墅湖畔