



# Godson-T16核 处理器的可测试 性设计

汇报人：

2024-01-16



# 目录

- 引言
- Godson-T16核处理器概述
- 可测试性设计理论与方法
- Godson-T16核处理器可测试性设计实现
- 实验验证与结果分析
- 结论与展望





# 01

## 引言



# 研究背景与意义

## 国产处理器发展现状

随着信息技术的迅猛发展，处理器作为计算机系统的核心部件，其性能和可靠性对整个系统至关重要。近年来，国产处理器在设计和制造方面取得了显著进步，但在测试领域仍面临诸多挑战。

## Godson-T16核处理器的特点

Godson-T16是一款高性能、低功耗的16核处理器，采用先进的微架构设计和制造工艺，具有高速缓存、多核并行处理、硬件虚拟化等特性，适用于云计算、大数据等领域。

## 可测试性设计的必要性

随着处理器复杂度的不断提高，传统测试方法已无法满足高质量、高效率的测试需求。可测试性设计（Design for Testability, DFT）通过优化处理器结构、提高故障覆盖率、降低测试成本等手段，为处理器测试提供有力支持，对提升国产处理器整体性能和可靠性具有重要意义。



# 国内外研究现状及发展趋势

## 国外研究现状

国外在处理器可测试性设计方面起步较早，已形成较为完善的理论体系和产业应用。例如，Intel、AMD等公司采用先进的DFT技术，实现了处理器的高故障覆盖率 and 低测试成本。同时，学术界也在不断探索新的测试方法和技术，如基于机器学习的故障预测、自适应测试等。

VS

## 国内研究现状

国内在处理器可测试性设计领域的研究相对较晚，但近年来发展迅速。一些国内企业和研究机构在DFT技术方面取得了一定成果，如华为海思、龙芯等公司的处理器产品已具备较高的可测试性。然而，与国外先进水平相比，国内在处理器测试技术、故障模拟与仿真等方面仍存在差距。



# 论文研究目的和内容概述

## 研究目的

本文旨在针对Godson-T16核处理器的特点和应用需求，研究其可测试性设计方法和技术，提高处理器的故障覆盖率和测试效率，降低测试成本，为国产处理器的研发和应用提供有力支持。

## 内容概述

(1) 分析Godson-T16核处理器的结构特点和测试需求；(2) 研究可测试性设计的关键技术，包括故障模拟与仿真、测试激励生成、故障覆盖率评估等；(3) 设计并实现针对Godson-T16核处理器的可测试性方案；(4) 对所提出的方案进行实验验证和性能评估；(5) 总结研究成果并展望未来发展方向。



# 02

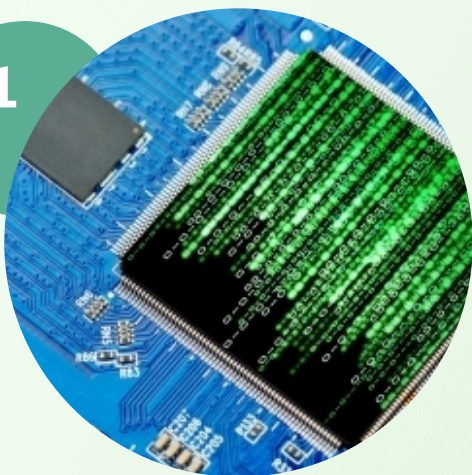
## Godson-T16核处理器 概述





# Godson-T16核处理器结构特点

01

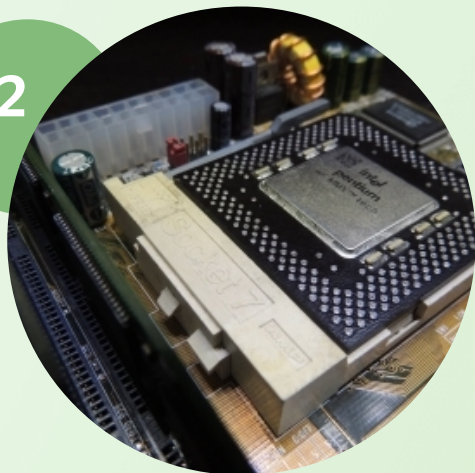


## 16核处理器



Godson-T16是一款拥有16个处理核心的处理器，采用先进的多核并行处理技术，实现高性能计算。

02



## 高速缓存



每个处理核心都配备有高速缓存，用于提高数据访问速度，减少处理器等待时间。

03



## 共享内存



Godson-T16采用共享内存结构，多个处理核心可以访问同一块内存空间，实现数据共享和协同工作。





# Godson-T16核处理器性能指标

## ● 主频

Godson-T16的主频达到了2.0GHz，保证了处理器的高速运算能力。

## ● 功耗

在保证高性能的同时，Godson-T16的功耗控制也非常出色，实现了高性能与低功耗的平衡。

## ● 兼容性

Godson-T16具有良好的软硬件兼容性，可以运行多种操作系统和软件应用。





# Godson-T16核处理器应用领域

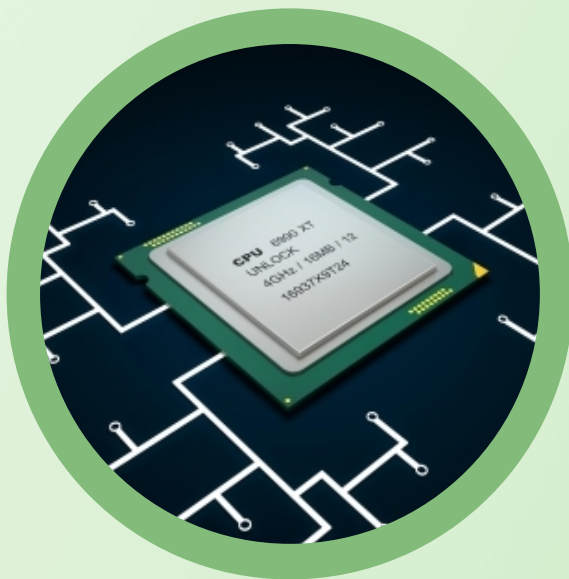
## 服务器领域

Godson-T16的高性能和多核并行处理能力使其成为服务器领域的理想选择，可以满足大数据处理、云计算等应用需求。



## 人工智能领域

Godson-T16可以应用于人工智能领域，如深度学习、神经网络等计算密集型任务。



## 嵌入式系统领域

Godson-T16的低功耗和出色性能使其也适用于嵌入式系统领域，如智能家居、物联网等应用场景。



# 03

## 可测试性设计理论与方法





# 可测试性设计基本概念及原理

## 可测试性定义

可测试性是指产品或系统能够被有效、高效地进行测试的程度，包括测试的容易性、准确性和完整性等方面。



## 可测试性设计原则

包括模块化、可观测性、可控制性、可重复性、故障隔离等原则，以确保设计的可测试性。



## 可测试性设计目标

通过设计使产品或系统更易于测试，提高测试的覆盖率、效率和准确性，降低测试成本和风险。





# 可测试性设计方法分类与比较

01

## 基于仿真的可测试性设计

通过建立仿真模型来模拟系统的行为，以验证设计的正确性和性能。这种方法适用于复杂系统的早期验证和测试。

02

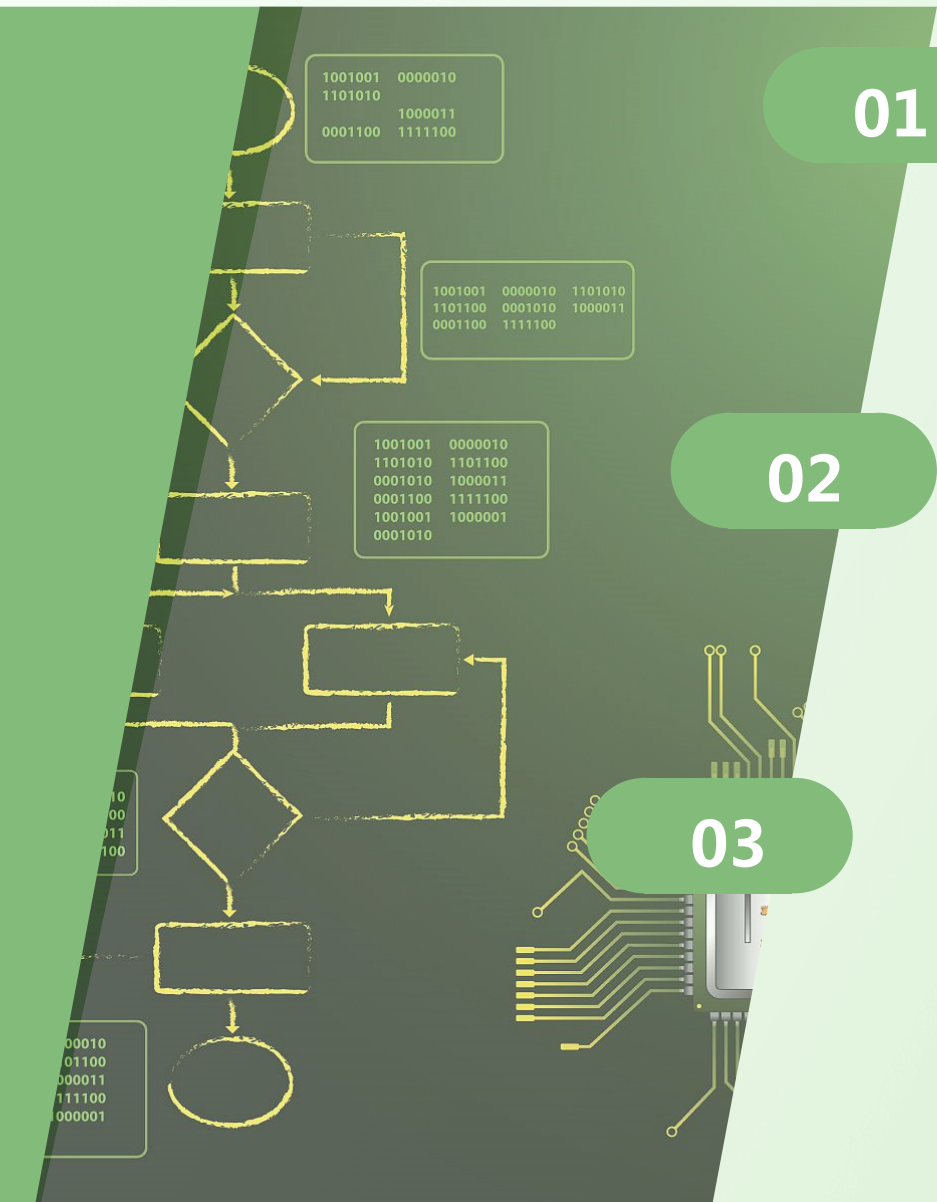
## 基于故障注入的可测试性设计

通过人为注入故障来观察系统的反应和恢复能力，以评估系统的可靠性和容错性。这种方法适用于对系统容错性和恢复能力的测试。

03

## 基于硬件辅助的可测试性设计

通过添加额外的硬件支持来提高测试的效率和准确性，例如使用内置的测试接口和调试工具。这种方法适用于对硬件进行深入测试和调试的情况。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：  
<https://d.book118.com/755024124001011222>