

T/ACCEM

团 体 标 准

T/ACCEM XXXX—XXXX

企业级芯片设计技术平台管理规范

Management specification for enterprise chip design technology platform

征求意见稿

XXXX—XX—XX 发布

XXXX—XX—XX 实施

中国商业企业管理协会 发布

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中茵微电子（南京）有限公司提出。

本文件由中国商业企业管理协会归口。

本文件起草单位：中茵微电子（南京）有限公司、XXX、XXX。

本文件主要起草人：XXX、XXX、XXX。

企业级芯片设计技术平台管理规范

1 范围

本文件规定了企业级芯片设计技术平台的总体要求、芯片设计流程管理、人员与团队管数据管理、项目进度与质量管理、文档管理以及变更管理的内容。

本文件适用于企业内芯片设计技术平台相关活动的管理和指导。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

- GB/T 35010.2 半导体芯片产品 第2部分：数据交换格式
- GB/T 35010.5 半导体芯片产品 第5部分：电学仿真要求
- GB/T 35010.6 半导体芯片产品 第6部分：热仿真要求

3 术语和定义

下列术语和定义适用于本文件。

3.1

系统级 system - level

芯片设计的最高抽象层次，从整体系统的角度描述芯片功能和性能要求，包括芯片在整个电子系统中的角色、与其他组件的接口关系、系统规范和总体架构设计等内容。

4 总体要求

4.1 管理目标

4.1.1 保障设计质量

应确保芯片设计在功能、性能、可靠性等方面满足预定的技术指标。从系统级功能的准确实现到电路级的电气特性达标，减少设计缺陷，提高芯片的良品率和使用寿命。

4.1.2 提高设计效率

优化设计流程和资源分配，减少不必要的设计迭代和返工。通过标准化设计步骤、合理使用设计工具以及高效的团队协作，缩短芯片设计周期。

4.1.3 确保设计过程可追溯性

建立完善的文档管理和版本控制系统，记录设计过程中的每一个决策、修改和操作。从需求分析、设计文档、代码编写到验证结果，每一个环节应有清晰的记录，以便在出现问题时能够迅速定位原因，同时也便于对设计过程进行审查和改进。

4.1.4 增强设计过程可控性

对设计流程、人员、工具、数据等各个方面进行有效管理和监控。明确各个设计阶段的输入、输出和关键控制点，通过进度管理、发现并解决潜在的风险和问题。

4.2 基本原则

4.2.1 遵循标准化流程

企业级芯片设计应依据统一、规范的设计流程进行操作。从设计的最初阶段到最终的测试环节，每个步骤都应有明确的工作内容、方法和标准。

4.2.2 保证数据安全与完整性

4.2.2.1 应采取严格的数据安全措施，如加密存储、访问控制、数据备份等，防止数据泄露和损坏。

4.2.2.2 数据交换格式应符合 GB/T 35010.2 的规定。

4.2.3 促进团队协作

建立良好的团队协作机制，明确各成员的角色和职责，加强沟通与协调。

5 芯片设计流程管理

5.1 设计过程

5.1.1 设计过程及系统需求分析

5.1.1.1 芯片设计从系统需求出发，贯穿物理域、结构域和行为域，在不同设计层次上逐步细化，直至完成芯片制造和测试

5.1.1.2 从用户需求和应用场景出发，确定芯片的功能、性能、功耗、成本等总体要求。

5.1.1.3 在系统级层面，应考虑芯片在整个电子系统中的角色，如与其他芯片或模块的接口、通信协议、数据处理能力等。这些需求将指导后续的设计决策，包括算法选择、架构设计和模块划分。

5.1.2 不同设计层次的关系

5.1.2.1 系统级到算法级

依据系统需求，将芯片功能分解为具体的算法模块，确定各个算法模块的功能和相互关系。

5.1.2.2 算法级到RTL级

将算法模块进一步转化为RTL级的硬件描述，通过寄存器、逻辑单元和数据传输路径来实现算法功能。每个算法模块在RTL级设计为一个或多个模块，描述数据在寄存器之间的传输和处理过程。

5.1.2.3 RTL级到逻辑级和电路级

在RTL级设计完成后，通过综合工具将其转换为逻辑级的门电路表示，再进一步细化到电路级的晶体管设计。

5.1.3 物理域、结构域和行为域的贯穿

5.1.3.1 物理域

在设计过程中，从电路级开始考虑芯片的物理实现，包括晶体管的版图设计、布线规则、芯片面积、功耗等物理因素。随着设计层次的上升，物理域的约束和要求将影响到更高层次的设计决策，如模块布局、信号完整性等。

5.1.3.2 结构域

从系统级的芯片整体架构设计开始，确定功能模块的划分、层次结构和接口关系。在各个设计层次中，结构域的设计应不断细化。

5.1.3.3 行为域

从系统级的功能行为描述开始，通过模拟和验证芯片在不同输入条件下的行为反应来指导设计。。

5.2 设计阶段管理

5.2.1 预研阶段

5.2.1.1 对芯片市场的需求趋势、竞争态势、潜在应用领域进行广泛调查。收集同类型芯片的性能指标、功能特点、价格范围等信息，分析市场的空白点和发展机会，为芯片设计定位提供依据。

5.2.1.2 对芯片设计所涉及的关键技术进行评估。包括对新算法、新架构、新工艺的研究和分析。评估技术的成熟度、可获取性以及与企业技术路线的兼容性。

5.2.1.3 初步估算芯片设计、制造、测试和封装的成本。

5.2.2 顶层设计阶段

5.2.2.1 根据预研阶段确定的芯片功能和性能目标，设计芯片的整体架构。

5.2.2.2 将芯片的整体功能细化为相对独立的功能模块。明确每个模块的功能边界和接口定义，确保模块之间的低耦合和高内聚。

5.2.2.3 详细定义功能模块之间以及芯片与外部环境的接口。

5.2.3 模块级设计阶段

5.2.3.1 依据顶层设计阶段确定的模块功能，进一步详细描述每个模块的内部功能。

5.2.3.2 对模块间接口在顶层设计的基础上进行细化。确定接口信号的具体含义、信号的有效电平、时序关系等。

5.2.3.3 根据细化后的模块功能，设计模块内部的逻辑电路。

5.2.4 模块实现阶段

5.2.4.1 依据模块级设计的 RTL 描述，进行代码的实际编写工作。

5.2.4.2 特定电路实现的模块，除了 RTL 代码编写外，应进行电路设计。

5.2.5 子系统仿真阶段

5.2.5.1 根据设计的规模和复杂程度，选择合适的仿真工具（如 ModelSim、VCS 等）。

5.2.5.2 设置仿真的时间单位、精度等基本参数。根据设计所使用的工艺库和实际工作条件，配置信号延迟、电源电压等参数，使仿真环境尽可能接近实际芯片的工作环境。

5.2.5.3 将子系统的设计文件加载到仿真工具中，建立完整的仿真模型。

5.2.5.4 电学仿真应符合 GB/T 35010.6 的规定。

5.2.5.5 热仿真应符合 GB/T 35010.7 的规定。

5.2.6 系统仿真、综合和版图设计前门级仿真阶段

5.2.6.1 系统仿真范围

5.2.6.1.1 将各个子系统集成在一起，进行全芯片的功能仿真。验证芯片在系统级的功能是否完整和正确，包括各个子系统之间的协同工作、数据在整个芯片中的传输和处理是否符合设计要求。

5.2.6.1.2 分析芯片在系统级的性能指标。应考虑不同子系统之间的交互对性能的影响，评估芯片是否满足系统级的性能目标。

5.2.6.1.3 检查各个子系统之间以及芯片与外部接口的一致性。确保数据在不同子系统之间传输时没有丢失、错误或时序问题，芯片与外部设备的通信正常。

5.2.6.2 综合工具的使用

5.2.6.2.1 根据芯片的制造工艺要求，选择合适的工艺库。

5.2.6.2.2 为综合过程设置约束条件，包括时序约束（如时钟频率、建立时间、保持时间等）、面积约束（如芯片的最大面积限制）、功耗约束（如芯片的功耗上限）等。

5.2.6.2.3 根据芯片的复杂度和设计目标，选择合适的综合策略。

5.2.6.3 门级仿真的目标和质量标准

5.2.6.3.1 对综合生成的门级网表进行功能仿真，验证其功能是否与 RTL 级设计一致。检查在综合过程中是否引入了新的功能错误，确保门级网表在逻辑功能上的正确性。

5.2.6.3.2 门级仿真的结果应与 RTL 级仿真结果在功能上完全一致，且初步的时序检查应满足综合过程中设置的时序约束的一定比例（如 80% 以上），以确保综合过程的质量。

5.2.7 后端版面设计阶段

5.2.7.1 布局规划

- 5.2.7.1.1 根据芯片的功能模块划分和预估的模块面积，对芯片的总面积进行合理分配。
- 5.2.7.1.2 将各个功能模块放置在芯片上的合适位置。
- 5.2.7.1.3 设计芯片的电源和接地网络，确保每个模块都有稳定的电源供应和良好的接地。

5.2.7.2 布线规则

- 5.2.7.2.1 根据芯片的制造工艺和设计要求，确定布线的层次结构。
- 5.2.7.2.2 依据芯片制造工艺的设计规则，确定布线的线宽和间距。
- 5.2.7.2.3 根据信号的特点和设计要求选择合适的布线拓扑结构。

5.2.7.3 时钟树设计

- 5.2.7.3.1 确定时钟信号的来源和在芯片中的分布方式。
- 5.2.7.3.2 在时钟树中设置适当的时钟缓冲器，以增强时钟信号的驱动能力，确保时钟信号能够稳定地到达各个模块。

5.2.8 测试向量准备阶段

- 5.2.8.1 基于芯片的功能规格说明书，采用手动编写和自动生成相结合的方式创建功能测试向量。
- 5.2.8.2 如芯片设计中采用了可测试性设计技术（如扫描插入），则生成相应的结构测试向量。
- 5.2.8.3 功能测试向量应覆盖芯片的所有功能模块、功能点和操作模式。
- 5.2.8.4 对于结构测试向量，应确保对芯片中所有可测试的结构元素进行全面覆盖。

5.2.9 后端仿真阶段

5.2.9.1 后端仿真工具选择

根据芯片后端设计的特点和需求，选择合适的后端仿真工具（如 PrimeTime、Calibre 等）。

5.2.9.2 设置仿真参数

根据芯片的制造工艺、设计要求和实际工作条件，设置后端仿真的参数。

5.2.9.3 仿真流程

- 5.2.9.3.1 对芯片进行电气特性仿真，分析版图中布线的电阻、电容、电感等参数对信号传输的影响。
- 5.2.9.3.2 对芯片的时序性能进行仿真分析。
- 5.2.9.3.3 对芯片的可靠性（如需要）进行评估，可在仿真过程中进行故障注入仿真。

5.2.10 生产签字

- 5.2.10.1 组织由设计团队、质量控制团队、项目管理团队等相关人员组成的审核小组，对芯片设计的全过程进行全面审核。
- 5.2.10.2 对于审核过程中发现的问题，由设计团队负责解决。问题解决后，需要重新进行相关的验证和检查，确保问题得到彻底解决。审核小组对问题的解决情况进行确认，保证设计没有遗留的隐患。
- 5.2.10.3 在设计审核通过且所有问题都得到妥善解决后，由授权的相关人员进行生产签字。生产签字表示芯片设计已经完成，可以进入制造环节。

5.2.11 硅片测试阶段

- 5.2.11.1 根据芯片设计的功能和性能要求，确定硅片测试的目标。包括功能测试、性能测试、可靠性测试等。
- 5.2.11.2 针对不同的测试目标，选择合适的测试方法。如对于功能测试，可采用自动测试设备加载测试向量进行测试，也可结合软件驱动程序在实际应用环境中进行测试。
- 5.2.11.3 规划硅片测试的详细流程，包括测试的先后顺序、每个测试环节的时间安排、不同测试之间的衔接。

5.3 设计一般步骤管理

5.3.1 第一阶段设计应包括如下步骤

- a) 依据企业内部既定的编码标准进行结构及电学特性编码；
- b) 遵循良好的 RTL 编码风格，如使用合适的模块划分，每个模块功能单一且明确；
- c) 应严格遵守硬件描述语言（如 Verilog、VHDL）的语法规则；
- d) 建立代码评审机制，在 RTL 编码完成后，组织相关的设计人员进行评审；
- e) 为包含存储单元的设计插入 DFT memory BIST；
- f) 验证设计功能，进行详尽的动态仿真；
- g) 实际环境设置，包括将使用的工艺库及其他环境属性；
- h) 使用 design compiler 工具对具有扫描插入的设计进行约束和综合设计；
- i) 使用 design compiler 的内建静态时序分析机进行模块级静态时序分析；
- j) 设计的形式验证，使用 formality 将 RTL 和综合后的网表进行对比；
- k) 使用 primetime 进行整个设计布图前的静态时序分析。

5.3.2 第二阶段设计步骤应包括如下步骤

- a) 对布图工具进行时序结束的前标注；
- b) 具有时序驱动单元布局，时钟树插入和全局布线的初始布局划分；
- c) 将时钟树转换到驻留在 design compiler 中原始设计（网表）；
- d) 在 design compiler 中进行设计的布局优化；
- e) 使用 formality 在综合网表和时钟树插入的网表之间进行形式验证；
- f) 在全局布线后，从版图提取估计的延时；
- g) 从全局布线得到的估计时间数据反标注到 prime time；
- h) 使用在全局布线后提取的估计延时数据在 prime time 中进行静态时序分析。

5.3.3 第三阶段的设计应包括如下步骤

- a) 设计的详细布局；
- b) 提取来自详细布局设计的实际时间延迟；
- c) 实际提取时间数据反标注到 prime time；
- d) 使用 prime time 进行布图后的静态时序分析；
- e) 布图后的门级功能仿真（如需要）；
- f) 在 LVS 和 DRC 验证后定案。

6 人员与团队管理

6.1 人员资质与培训

规定芯片设计相关人员的专业资质要求，以及入职培训、持续培训的内容和计划。

6.2 团队组织与职责

明确设计团队的组织结构，包括系统设计、模块设计、仿真、验证等不同角色的职责和协作关系。

6.3 团队沟通与协作机制

建立团队内部沟通渠道、会议制度、问题反馈与解决机制等，确保设计过程中的信息流畅和高效协作。

7 数据管理

7.1 数据存储与备份

规定设计数据的存储格式、存储位置、备份策略和存储介质管理，确保数据的安全性和可恢复性。

7.2 数据安全和保密

建立数据安全管理制度，包括用户权限管理、数据加密、访问控制、防止数据泄露和非法访问的措施等。

7.3 数据质量与完整性

制定数据质量标准，对数据的准确性、一致性、完整性进行检查和维护，保证设计数据的可靠性。

8 项目进度与质量管理

8.1 项目进度管理

8.1.1 依据设计流程和项目要求，制定详细的芯片设计项目进度计划，明确各阶段的起止时间和关键里程碑。

8.1.2 建立进度跟踪机制，定期对项目进度进行检查和评估，制定应对进度偏差的调整措施。

8.2 质量管理

8.2.1 明确芯片设计各阶段的质量标准，包括功能正确性、性能指标、时序要求、可靠性等方面的质量要求。

8.2.2 建立质量控制流程，包括设计评审、代码审查、仿真验证、测试等环节的质量控制措施，确保设计质量符合标准。

9 文档管理

9.1 规定芯片设计过程中各类文档（设计规格书、测试计划、用户手册等）的内容、格式、编写规范和详细程度要求。

9.2 建立文档审核流程，明确审核人员的职责和审核标准，确保文档的准确性和完整性，规定文档的批准程序。

9.3 制定文档的存储管理规定，包括存储位置、版本控制、更新通知等，保证文档的可获取性和时效性。

10 变更管理

10.1 规定变更请求的提出方式、内容要求，包括变更原因、影响范围、建议解决方案等。

10.2 建立变更评估机制，对变更请求进行技术、进度、质量、成本等方面的评估，确定变更的必要性和可行性。

10.3 明确变更审批流程和权限，确保变更经过适当的审批。

10.4 规定变更实施的步骤和方法，以及对变更后的设计进行验证的流程，确保变更不会引入新的问题。