

电子科技大学示范性微电子学院

实验课程及实验室安全预习报告

(实验) 课程名称 IC 综合实验 2

学号: 2022020915006

姓名: 邓锦琪

实验地点: 清水河校区国际创新中心 B523

实验日期: 2024.11-2025.01

一、 实验目的与要求

本次 IC2 实验课程（数字方向）的设计内容为一个已知结构的 AI 加速器的 ASIC 全流程设计并通过评估和流片。已知结构的 AI 加速器具有 8bit 输入和 8bit 输出，可以工作在至少 50Mhz 的时钟下，能够完成给定卷积神经网络的运算。总体的面积要求为在不带 I/O 的情况下通过 DC 综合得到的网表面积为 $24 \times 10^4 \mu\text{m}^2$ ，版图的面积最大限制为 $900\mu\text{m} \times 900\mu\text{m}$ ，可以使用的最大的引脚数量为 28 个（含 4 到 6 个电源引脚）。

实验目的：通过 AI 芯片设计驱动，完成前端设计、仿真与验证、综合、后端布局布线，物理验证、一致性检查等相关的流程，学习整个 IC 数字芯片设计的全流程，对未来的科研和工作打下坚实的基础。

二、 实验相关原理预习

2.1 芯片设计与前仿真

在芯片设计与前仿真阶段，VCS（Verilog Compiled Simulator）软件和 DVE（Debugging and Visualization Environment）是两个重要的工具，它们协同工作以实现高效的仿真和调试。

VCS 是 Synopsys 公司提供的高性能 Verilog 仿真软件，它支持从行为级、RTL（寄存器传输级）到门级等多种抽象层次的仿真。VCS 能够分析和编译 HDL 设计代码，并内置了仿真和调试功能，同时支持 SystemVerilog、验证规划、覆盖率分析等高级验证功能。

DVE 是 VCS 的图形用户界面（GUI），提供了交互式波形查看和调试功能。用户可以通过 DVE 直观地查看仿真波形，添加断点，单步执行代码，以及进行信号的实时监控和分析。

芯片设计与前仿真流程包括一下 4 个阶段：

- 1、编译：首先，使用 VCS 对 Verilog 或 SystemVerilog 代码进行编译，生成仿真所需的可执行文件。这个过程中，可以指定编译选项，如+timescale、+define 等，以控制编译行为。

- 2、仿真：编译完成后，运行仿真可执行文件（通常是 simv），进行实际的仿真。这个步骤可以是命令行方式，也可以通过 DVE 的 GUI 界面进行。

- 3、波形查看与调试：仿真运行后，可以使用 DVE 打开波形文件，查看仿真波形。DVE 允许用户添加波形视图，设置断点，以及进行信号追踪，从而深入分析和调试设计中的问题。

- 4、调试：在 DVE 中，用户可以利用各种调试工具，如信号赋值、断点管理等，来交互式地调试设计。这些工具帮助用户快速定位和解决设计中的错误。

VCS 负责仿真的编译和执行，而 DVE 则提供了一个直观的界面来查看和分析仿真结果。两者的结合使得芯片设计的前仿真阶段既高效又易于管理，特别是在复杂设计中，图形化的调试工具可以显著提高调试的效率和准确性。

2.2 芯片综合

利用 DC 软件和 DC 脚本进行的综合流程可以将 RTL 代码转换成门级网表，同时优化以满足特定的时序和面积要求。首先，DC 软件通过读取设计文件和库文件来准备综合环境。随后，通过 DC 脚本自动化执行综合命令，包括设置顶层模块、定义时钟约束和进行面积优化。综合过程中，DC 软件会分析设计并提供时序和资源利用的报告，以便设计师对设计进行调整。最终，综合结果被保存为门级网表，并生成包含时序、面积和功耗信息的综合报告，为后续的物理设计和验证提供基础。通过 DC 脚本，这一流程可以实现自动化，提高综合的效率和准确性。

随后可以使用 Formality 工具进行形式验证，通过比较 RTL 设计和综合后的网表，确保两者在功能上的等价性。Formality 验证流程包括以下几个步骤：

1、**Guidance**: 添加 DC 综合产生的.svf 文件，该文件记录了综合过程中的优化信息，这对于 Formality 验证非常重要，因为它包含了对象名称变更、常量寄存器优化等信息。

2、**Reference**: 读入 RTL 设计文件，并设置顶层模块。同时，读入设计库文件，并设置参考设计的顶层。

3、**Implementation**: 读入综合后的网表文件，并设置实施设计的顶层。

4、**Setup**: 在 Formality 中设置常量和和其他相关约束，为匹配和验证做准备。

5、**Match**: 运行匹配操作，比较参考设计和实施设计，查看匹配点和不匹配点。

6、**Verify**: 执行验证操作，确认 RTL 设计和门级网表在功能上是否相同。

7、**Debug**: 如果验证结果不一致，使用 Formality 的调试功能找出差异原因，并进行修正。

2.3 芯片的后端设计

后端设计流程开始前需要从前端获取综合得到的门级网表和时序约束文件等必要的设计数据。接着，使用 ICC 软件进行设计和布局、时钟树、布线的优化。在 ICC 中，通过编写 TCL 脚本来自动化 Data Setup 步骤，该步骤主要目的是将门级网表、约束文件、单元库、物理库和寄生参数模型等输入 ICC。

在 ICC 中完成布局布线后，得到的版图文件（GDSII 格式）需要被导入到 Cadence Virtuoso 中进行进一步的分析和验证。在 Virtuoso 中，可以通过 Library

Manager 将 ICC 产生的 GDSII 版图文件与标准单元库的版图文件以及 IO 库的版图文件进行合并。

最后，利用 Calibre 工具进行 DRC（设计规则检查）和 LVS（版图与原理图对比）检查。Calibre 作为物理验证工具，提供了有效的 DRC/LVS/ERC 解决方案，特别适合超大规模 IC 电路的物理验证。在 Virtuoso 中调用 Calibre 进行 DRC 检查，需要使用工艺库中提供的 RULE 文件。检查完成后，Calibre 生成的结果文件（如 drc_err.sum）会列出所有 DRC 违规项，设计者需要根据这些信息在版图中进行相应的修改。Calibre 的 RVE（Results Viewing Environment）界面可以将验证错误反标到版图工具中，大大提高了改错的效率。

三、 实验内容与步骤预习纲要

专用集成电路（ASIC）的设计阶段包括需求分析与规格说明、前端设计、逻辑综合、形式验证与静态时序分析、后端的布局布线、设计规则检查（DRC）和布局与原理图对比（LVS）。最终，完成了验证的可制造性检查的 ASIC 设计进行流片和制造，成功制造和封装后，ASIC 可以进行电气测试并集成到原型中进行实验室和现场测试。

3.1 需求分析

本次 IC2 实验课程（数字方向）的设计内容为一个已知结构的 AI 加速器的 ASIC 全流程设计并通过评估和流片。已知结构的 AI 加速器具有 8bit 输入和 8bit 输出，可以工作在至少 50Mhz 的时钟下，能够完成给定卷积神经网络的运算。总体的面积要求为在不带 I/O 的情况下通过 DC 综合得到的网表面积为 $24 \times 10^4 \mu\text{m}^2$ ，版图的面积最大限制为 $900\mu\text{m} * 900\mu\text{m}$ ，可以使用的最大的引脚数量为 28 个（含 4 到 6 个电源引脚）。

给定的卷积核的结构包含卷积层、池化层、全连接层各一层，共计 3 层。各层的详细介绍如下。

卷积层：在 11×11 的矩阵数据外添加 Padding 后，对对应的矩阵进行卷积操作，其中，卷积核大小为 3×3 ，步长为 2，共有 3 个卷积核；

池化层：对输出的数据进行最大池化，其中池化核大小为 2×2 ，步长为 2；

全连接层：全连接层对应的权重矩阵大小为 $3 \times 3 \times 3$ ，与池化输出一一对应，需要完成一次乘加操作。

芯片的输入数据为 1×1 的矩阵，是位宽为 8bit 的有符号数补码，按照卷积核、全连接权重、100 组 11×11 的矩阵的数据依次输入。

芯片的输出数据为 1×1 的矩阵，位宽为 8bit，输出的为全连接的结果通过量化所得到的内容。

由于在数据处理过程中得到的乘积数据为 16bit，加和后需要用 20bit（全连接 21bit）的数据记录才能保证数据不发生溢出，而后续进行池化、全连接和输出所需要的数据均为 8bit，因此需要一种量化规则来进行 20bit 或 21bit 的到 8bit 的量化处理。给定的卷积核的量化规则为：在保证符号正确的情况下拓展到 20 位来表示，然后将低 8 位抹除，剩余的 12 位若超过 int8 的表示范围，则取 int8 范围的最大值，反之，取这 12 位的低 8 位。由于数据均由补码表示，因此所得的结果只需要将最高位符号位向前扩展即可得到扩展数据。

3.2 系统架构设计

这一阶段的主要目的是确定 ASIC 的整体结构和功能模块。在这个阶段，设计师根据需求分析的结果，定义 ASIC 的顶层架构，包括数据路径、控制逻辑、接口规范以及各个子模块的划分。架构设计的目标是确保 ASIC 能够高效地实现预定的功能，同时满足性能、功耗和成本等约束条件。此外，架构设计还需要考虑可扩展性和可维护性，以适应未来可能的需求变化。

3.3 RTL（寄存器传输级）设计

RTL 设计是将系统架构转化为具体的硬件描述语言代码的过程。在这个阶段，设计师使用 Verilog 或 VHDL 等硬件描述语言，详细描述 ASIC 中每个模块的功能和行为。RTL 代码是 ASIC 设计的核心，它直接影响到 ASIC 的最终实现和性能。因此，RTL 设计需要精确地反映架构设计的要求，并考虑到时序、资源利用和功耗等因素。

3.4 仿真与验证

仿真与验证是确保 ASIC 设计正确性的关键步骤。在这个阶段，设计师通过模拟 ASIC 在不同条件下的行为来验证其功能和性能是否符合预期。这包括对 RTL 代码进行单元测试、集成测试和系统测试，以及对时序、功耗和热特性等进行分析。仿真与验证的目的是尽早发现并修正设计中的错误，减少后期修改的成本和风险。

3.5 综合

综合是将 RTL 代码转换成门级网表的过程，这个网表描述了 ASIC 中所有逻辑门的连接和配置。综合工具会根据设计要求（如性能、面积和功耗）进行优化，以生成最优的门级实现。综合过程中，设计师需要考虑逻辑优化、时序约束和面积限制等因素，以确保 ASIC 设计满足所有规格要求。

3.6 布局与布线

布局与布线（Physical Design）是将门级网表转换成实际的物理版图的过程。

在这个阶段，设计师需要将逻辑门放置在芯片的特定区域，并连接它们的输入和输出，形成完整的电路。布局与布线需要考虑电路的性能、功耗、热分布和制造工艺等因素，以确保 ASIC 的可靠性和可制造性。

3.7 验证

在布局与布线之后，ASIC 设计需要经过一系列的验证步骤，以确保物理版图满足制造要求并且与逻辑设计一致。这包括设计规则检查（DRC）以确保版图符合制造工艺规范，以及布局与原理图对比（LVS）以确保版图与逻辑设计相匹配。此外，还需要进行后仿真验证，以确保布局布线后的 ASIC 在时序和功能上仍然正确。

3.8 流片

流片是将验证无误的 ASIC 设计提交给晶圆厂进行制造的过程。在这个阶段，晶圆厂会根据提供的 GDSII 文件在硅片上制造 ASIC。流片成功后，会进行芯片的封装和测试，以验证 ASIC 的实际性能和可靠性。

四、 使用的实验仪器（设备、元器件）

计算机终端、服务器终端、IC 设计工业软件，包括仿真软件 VCS(Verilog Compiled Simulator)、波形查看与电路调试软件 DVE(Debugging and Visualization Environment)、逻辑综合软件 DC (Design Compiler)、布局布线系统 ICC(IC Compiler)、后端版图设计软件 Cadence Virtuoso 和设计规则检查 DRC(Design Rule Check)和布局与原理图对比 LVS(layout versus schematic)的专业软件 Calibre

五、 实验室安全操作事项

IC2 实验中需要连接服务器，不要污染服务器中的文件
使用服务器时注意保持实验室电脑的安全

六、 思考题（至少 3 道）

6.1 在卷积神经网络运算中，如何优化 AI 加速器的架构设计以提高运算效率？

答：可以从数据流和控制流的角度出发，合理安排卷积层、池化层和全连接层的数据传输和处理顺序，减少数据搬运的开销，提高运算的并行度。例如，是否可以采用流水线结构，将不同层的运算重叠进行，以充分利用硬件资源，减少闲置时间。

另外，还可以考虑如何优化存储结构，高效地存储和访问卷积核、权重矩阵

以及中间数据，避免存储瓶颈对运算效率的影响。例如，是否可以采用片上存储和片外存储相结合的方式，将频繁访问的数据存储在片上，减少片外存储的访问次数。

6.2 在 ASIC 设计的全流程中，如何平衡设计的性能、面积和功耗等指标？

性能方面，在满足时钟频率要求的前提下，优化设计的时序和逻辑结构，提高运算速度。例如，是否可以采用时钟门控技术，减少无效时钟的切换，提高时钟的利用率；是否可以优化数据路径的设计，减少数据传输的延迟。

面积方面，在满足功能和性能要求的前提下，尽量减小芯片的面积。例如，是否可以采用共享资源的设计方法，减少重复模块的使用；是否可以优化布局布线，提高芯片的利用率。

功耗方面，在保证性能和面积的前提下，降低芯片的功耗。例如，是否可以采用低功耗设计技术，如动态电压调节和动态频率调节等，根据实际工作负载调整芯片的功耗；是否可以优化电源和地线的设计，减少功耗损耗。

6.3 如何在前端设计阶段准确地定义和划分 ASIC 的各个功能模块？

根据卷积神经网络的运算特点和需求，合理地将 ASIC 划分为卷积层处理模块、池化层处理模块、全连接层处理模块等子模块。需要考虑各模块之间的数据交互和控制逻辑，确保模块划分清晰、功能独立，便于后续的 RTL 设计和验证工作。

七、 预习遇到的问题：

- 1、DC 脚本中的时钟约束如何合适的添加；
- 2、DC 中通过的时序在 ICC 中无法通过，如何通过一些策略促进 ICC 布局布线得到比较好的结果；
- 3、在高速设计中如何设计计算单元可以得到比较好的面积和性能。