**HLS部分课堂展示稿（1段对应1页PPT）：**

1、电子设计自动化是指利用计算机辅助设计软件，完成超大规模集成电路芯片的功能设计、综合、验证、布局、布线、版图、设计规则检查等流程的设计方式。EDA技术的基本特征是采用硬件描述语言描述，具有系统级仿真和综合能力。电子设计自动化的发展可以分为三个阶段：计算机辅助设计阶段、计算机辅助工程设计阶段、电子设计自动化阶段。电子设计自动化工具软件可大致可分为芯片设计辅助软件、可编程芯片辅助设计软件、系统设计辅助软件等三类。EDA软件在EDA体系中占据重要地位。利用EDA软件,电子设计师可以从概念、算法、协议等开始设计电子系统，大量工作可以通过计算机完成，并可以将电子产品从电路设计、性能分析到设计出IC版图或PCB版图的整个过程的计算机上自动处理完成。EDA工具不仅能够帮助设计人员完成电子系统设计，并且能提供独立于工艺和厂家的系统级设计与仿真。

2、硬件描述语言是一种用于电子系统硬件行为描述、结构描述、数据流描述的语言。目前，任何一种EDA工具都离不开硬件描述语言。目前主流的硬件描述语言主要有两种：Verilog HDL与VHDL。二者的共同点在于：它们都可以形式化地抽象表示电路的行为和结构；支持逻辑设计中层次与范围的描述；可以简化电路行为的描述；具有电路仿真和验证机制；支持电路描述由高层到低层的综合转换；与实现工艺无关；便于管理和设计重用。但它们也有一定的不足之处，例如难于理解，入门时间较长，对于软件开发者来说不友好等等，即传统的软件工程师很难使用 Verilog 和 VHDL 语言在 FPGA 上实现高效的数字硬件电路或硬件加速算法。

3、高层次综合则很好地解决了上述难题，HLS 是一种代码综合工具，它能将 C/C++等高级编程语言描述的逻辑、算法和功能转 换成 Verilog 和 VHDL 等硬件编程语言描述的寄存器传输级（Register Transfer Level，RTL）数字硬件电路，这不仅提升了开发速度，降低了开发难度，缩短了开发周期，更重要的是让软件工程师也能方便地进行基于 FPGA 等具备高并行处理能力设备的硬件电路或硬件算法的开发。

4、Vivado HLS 除了对常用的 C/C++标准库进行了兼容和优化，还添加了针对图像和视频处理算法的优化库。此外，Vivado HLS 中的可用数据类型十分丰富，此外还可以通过相关库来定义定点数据类型。更重要的是，Vivado HLS为了更好地支持 FPGA 硬件IP的开发，还融入了多种硬件接口封装、数据封装和优化指令。因此，相对于 Verilog 和 VHDL 的开发方式而言，Vivado HLS 极大地降低了在 FPGA 上实现硬件算法和硬件功能的难度。因此，相对于 Verilog 和 VHDL 的开发方式而言，Vivado HLS 极大地降低了在 FPGA 上实现硬件算法和 硬件功能的难度。特别是例如图像处理、视频流处理和卷积神经网络等复杂算法 在 FPGA 上的硬件设计实现，Vivado HLS 将是更好的选择，也是未来硬件开发的趋势。

5、利用高级综合工具进行系统开发的流程如下：第一步：使用 Vivado HLS 提供的数据类型、标准库和附加库等编写所需实现逻辑、算法或功能的 C/C++代码，同时编写用于测试的 TestBench 脚本； 第二步：使用 TestBench 测试脚本和算法 C/C++代码进行 C 功能仿真， 验证算法的功能正确性； 第三步：使用 Vivado HLS 提供的控制约束、接口封装和优化指令等功能，对通过 C 功能仿真验证的算法代码进行约束、封装和优化，包括并行优化和流水线优化等； 第四步：对优化后的算法代码进行 C 综合，即 HLS 高层次综合，转换成 RTL 级的 Verilog 或 VHDL 代码； 第五步：结合 TestBench 测试脚本对高层次综合后的算法代码进行 RTL 联合仿真，即 C/C++和 RTL 联合仿真，确保算法功能和 C 功能仿真的结果一致； 第六步：将通过 RTL 联合仿真的算法代码打包成硬件 IP 包并输出，后 续可在 Vivado 设计开发套件中调用该硬件算法 IP 用于系统集成设计。

6、基于 FPGA 的神经网络硬件加速器设计方案发展至今总共可归纳为四种，分别 是 1）在 FPGA 中实现一个完整的 神经网络网络模型；2）对 神经网络的各个神经元层 进行软硬件划分，一部分在软件处理器（例如 ARM）上以软件的形式实现，一部分在 FPGA 中以硬件的形式实现；3）在 FPGA 中实现 神经网络的一部分计算， 通过软件对硬件的配置和调用实现完整的 神经网络推理；以及 4）在 FPGA 中设计 实现一个基于特定指令集的通用型 神经网络硬件处理器。上述四种设计方案从不同的角度解决了 神经网络硬件设计时的速度、灵活性和 FPGA 硬件资源消耗量的难题。但是方案 1 的 FPGA 硬件资源消耗量较大，方案 2 的 神经网络执行速度和灵活性较差，方案 4 的设计成本、周期和难度均太高。因此方案3在一定 程度上平衡了速度、灵活性和 FPGA 硬件资源消耗量，更适合用于设计嵌入式 神经网络系统。