任何产品都有窗口期，每家公司都希望自己的产品尽快上市(Time-to-Market)，更快更多地抢占市场。为了最大化地缩短开发和生产周期，每个行业都在使用新进的工具来进行并行开发。芯片设计也不例外，如果简单地将芯片设计划分为硬件和软件两部分，传统的设计流程是先硬件后软件，以串行方式进行。因为软件开发依赖于硬件平台，为了实现硬件与软件并行开发，功能模型使软硬件并行开发成为可能。功能模型(Function model)也称为虚拟原型(Virtual Prototyping VP)，虚拟平台(Virtual Platform VP)，它们的本质都是一样，如果说非要有些区别，虚拟原型更倾向于片上系统(System on Chip, SoC)，虚拟平台涉及更大的系统，除了包括SoC之外，还包括其它子模型如NoC，DDR，Peripherals等，在这里，可以理解为板级(Printed Circuit Board，PCB)虚拟系统。

软件设计左移(Shift-left)在芯片设计领域越来越被重视，左移就是为了实现软硬件并行开发。参考图7，如果将芯片设计简单分为架构，前端和后端设计三个阶段，使用传统设计流程，软件需要在RTL开发快结束的时候，才能在FPGA/Emulator平台开始软件开发和调试，参考图7中Emulation starting point。如果采用功能模型或虚拟原型(VP)，在时间上可以提前至少8~12个月，给软件团队足够的时间来开发和优化软件，利用功能模型，可以开发firmware/driver/API/boot flow/Linux/software stack等，如果功能模型增加了时序和追踪(Trace)功能，还可以运行应用程序，用来分析软件的性能。通过两种方式对比，在时间上有绝对的优势，因此获得了广泛应用，比如VDK、Simics、利用SystemC+TLM2.0开发的功能模型等。

日程表

描述已自动生成

芯片研发周期示意图

与性能模型相比，功能模型是程序员视角模型(Programmer View, PV)，只要保证寄存器精度和数据传输正确即可，无需关注时序以及链路上数据的具体传输过程。如果有额外的需求，比如分析软件性能，可以在模型里面增加时序(Timing)、事件(Event)等信息，建模非常灵活，完全可以根据需求增加相应的特征(Feature)或功能。

图示

描述已自动生成

功能模型内部结构

由于功能模型不需要考虑时序和传输过程等因素，与性能模型比起来，相对简单很多，参考图8，典型的功能模型用例主要包括输入/输出接口，数据队列，控制逻辑，回调函数，寄存器，状态信息，中断信号等等。

完整的ESL仿真器由4个层次构成：应用层、组件层、内核层和主机层。其中组件层和内核层构成仿真器的主体，应用层为仿真器提供输入，而主机层承载了仿真器的运行，如图所示：

图示

描述已自动生成

ESL模拟器基本组成

1. 应用层：负责芯片工作时的场景构建，用来产生仿真器的输入或仿真激励，包含三种应用模式：

a.执行驱动：运行业务软件，生成流量驱动。与目标系统程序的运行方式类似，都是执行编译好的可执行二进制代码；

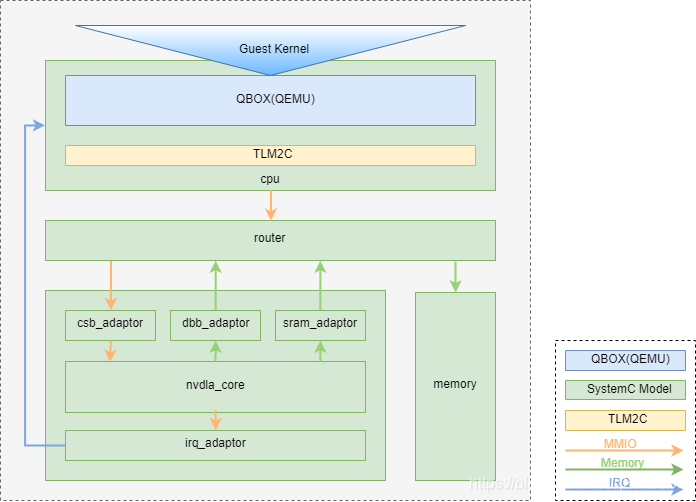
b.Trace驱动：解析其他平台生成的处理器核Trace轨迹，生成流量驱动。用trace代替实际CPU模型，将每条指令顺序执行所产生的所有信息作为仿真器的输入；

c.Traffic驱动：根据分解的业务动作配置表及驱动关系，生成流量驱动。给目标架构注入流量，实施载体为Traffic Generator(TG)。

1. 组件层：基于开源软件或工具生成的第三方IP或厂商提供的多精度模型搭建生成的待测芯片架构，使用TLM接口通信。组件层一般包括处理器组件、TG组件、接口转换组件、互联组件、DDR组件和Cache组件等。
2. 内核层：以事件队列为核心，控制仿真事件的正确调度、仿真时间的正确推进，控制整体仿真进程，支持组合目标硬件系统，支持串行、并行仿真和仿真加速手段。
3. 主机层：Linux多核工作站，承载架构仿真任务。

NVDLA VP虚拟平台架构：

NVDLA虚拟平台提供了一个可精确注册的系统，可以在该系统上快速开发和调试软件。该虚拟平台基于GreenSocs QBOX，它是与QEMU和SystemC共同仿真的解决方案。下面的图89显示了NVDLA虚拟模拟器的顶层示意图。包含ARMv8“虚拟” SoC板的QEMU仿真器，以提供高性能CPU仿真和通用设备。该仿真器包装在带有一组TLM-2.0接口的标准SystemC模块内，可以与其他SystemC模块进行交互。



图示

描述已自动生成