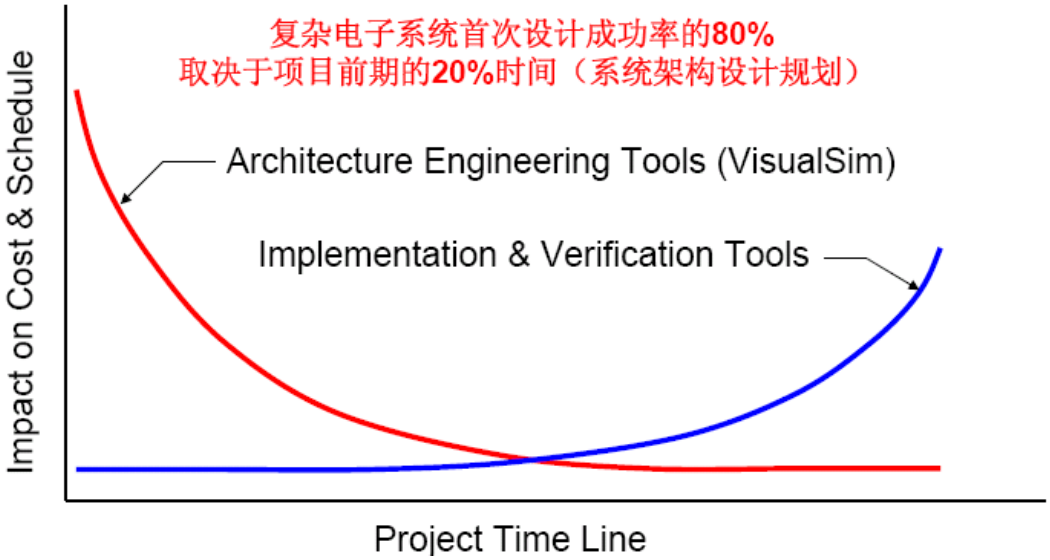
VisualSim®产品介绍

1. 技术背景

EDA 技术经过了二十几年的发展，针对电子设计流程中某一专门领域的设计与验证工具（例如 FPGA 设计、原理图与 PCB 设计、DSP 算法仿真等）已经发展得相当成熟，自动化程度越来越高，使用也变得越来越简便快捷。但与此形成对比的是，对于通信、多媒体处理等领域的 SoC/ASIC 或电子系统设计，由于可选择的 IP 或芯片以及相关的软件、硬件实现方案越来越复杂、同时设计约束条件（例如实时性、功耗、成本与物理尺寸等）越来越苛刻，项目研发团队开始面临芯片或系统样机首次设计完成后测试失败的可能。

随着单个硬件与软件部件设计能力和可靠性的提高，系统失效更多的是由于各子系统之间没有相互协调好所引起，而非单个部件的设计缺陷引起。这时，设计过程要求在其前期阶段，进行系统级别上的规划、分析与优化。为满足局部和全局的设计约定，通过大量的事务级（Transaction Level 或简称 TL）模型、用例流程和测试激励来验证系统规约是十分必要的。这里所指的系统规约包括系统架构设计方案、硬件平台选择、软件与数据流规划、硬件/软件任务协调与划分方案等。

因此，系统设计师已经开始把更多的注意力转向系统设计的方法学上，寻求真正面向电子系统级（ESL）设计的，能够为复杂电子系统的体系结构设计提供科学、有效手段的 EDA 工具。



2. VisualSim®概述

VisualSim®是一个用于系统规约设计、分析与验证的系统工程学解决方案。**VisualSim®**

使用图形化环境来构建诸如分布式、网络化处理系统，或 SoC/FPGA 这样的高性能半导体产品的事务级模型。借助 **VisualSim®**的快速虚拟原型开发技术，设计团队在项目开发的前期阶段即可对一个电子系统的不同硬件、软件实现方案进行快速的性能分析与研究评价，验证和优化系统设计构想，得到能够满足全部约束条件的最优系统实现方案（系统规约）。

作为一款完整的电子系统级建模、架构设计与性能分析工具，**VisualSim®**通过独特的参数化组件库建模技术以及多抽象层次模型混合仿真技术，在统一的图形化环境中快速实现对系统、子系统、元件或嵌入式软件的行为和性能建模，并据此进行事务级（TL）或时钟精度级（Cycle-accurate 或简称 CA/BCA）仿真。**VisualSim®**的仿真报告依照系统工程师的定制要求给出，借助这些报告，系统工程师可以确定所设计的系统是否按照设想的流程或方式工作，性能是否达到预计的要求，并可以通过仿真参数扫描来定位设计方案的瓶颈所在，以便进一步进行系统规约的优化。

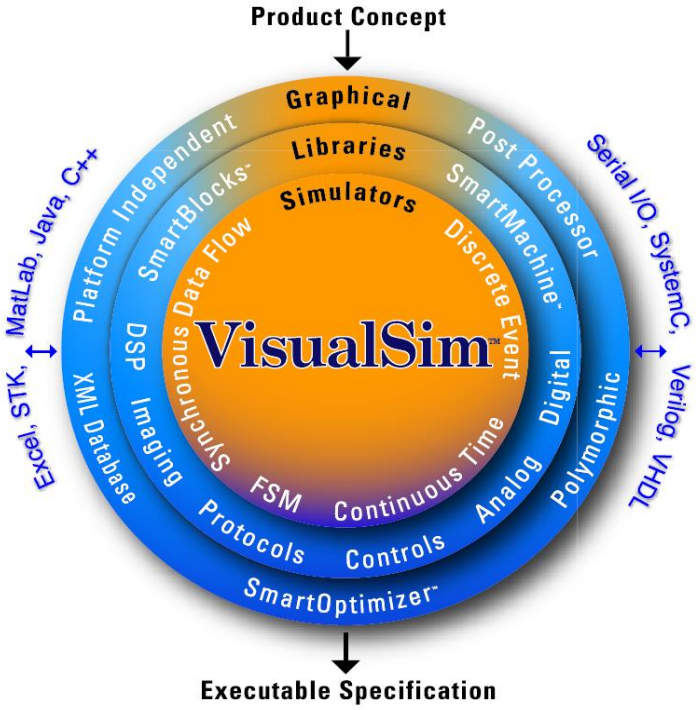
VisualSim®组件库中的模型组件是专门真对事务级系统建模而抽象定义出来的，所有组件均已经预先编译好，并经过面向仿真的全面优化以提高仿真性能。**VisualSim®**组件库中提供预定义的参数化模型组件，包括事务激励发生器（Traffic Generator）库、事务流程定义与处理库、通道与队列资源库、时钟精度级硬件模型模板库、数学运算库、脚本与接口库、结果输出与统计分析库、应用算法库等。

VisualSim®不仅支持直接使用组件库中各种组件进行快速系统架构建模，同时允许用户自己开发定制组件或者导入已有的 C/C++/Java/SystemC/Verilog/VHDL 模型。在 **VisualSim®**环境中允许创建和仿真所有层次的事务级模型（TLM3, 2, 1）。

VisualSim®环境中创建的模型直接在经过优化的类似 SystemC 的 **VisualSim®**仿真器中进行事务级或时钟精度级仿真。**VisualSim®**的仿真结果或报告完全依照模型设计者的需要定制给出，例如实时系统的全系统处理耗时、多任务调度方案形成的任务延时（Latency）、流水线工况、（多）处理器实际利用率、总线（网络）仲裁方案造成的总线（网络）传输延时、总线（网络）冲突与利用率、缓冲利用率、元件或系统功耗等。

通过调整系统模型的全局参数、改变输入激励方案等，系统工程师或架构师能够在 **VisualSim®** 模型的帮助下研究和评价设想方案的功能和性能特性。对于仿真结果不满足要求的设计方案，通过在图形化环境中更改模型组件的拓扑结构或参数，快速实现更“剧烈”的软硬件架构变化，以“What if”分析方式进行系统性能、功能、功耗等设计要求之间的平衡折衷（Trade-off）与优化。

确认满足设计要求的 **VisualSim®**模型即是 **VisualSim®**的输出结果，这是一个可视化、可执行的系统规约。**VisualSim®**可执行规约能够确认设计需求、设计约束的合理性，证明设计方案的正确性，将既往模糊的系统设计经验表述为数字化模型，并为后续系统方案的扩展升级或新方案设计提供科学的、可重用的参考依据。此外，**VisualSim®**模型还能够生成用于实现阶段的验证用测试向量集与断言。



3. VisualSim®核心价值

借助于 **VisualSim®**，设计者可以得到“**正确且适用**”的产品架构，亦即避免由于前期系统设计阶段错误而导致的产品实现失败，避免盲目的超裕量设计（Over-design）或者性能不达标设计（Under-design）。**VisualSim®**通过彻底改变传统建模方法来加速概念工程，将模型开发周期从数月缩短至数天，保证了产品从需求分析到设计实现的连贯性，将总体项目开发时间缩减 25%以上。

我们的用户集中于计算机、半导体、网络和空间技术领域。**VisualSim®**解决方案为客户提供的核心价值：将既往模糊的系统设计经验表述为可视化的可执行系统规约；在系统总体架构层面上进行快速的产品成本、性能和功耗的综合评估；为优化和差异化产品设计创新提供便利。

4. VisualSim 方法学

4.1 Learning by Doing

复杂的系统架构设计必然是一个从模糊到具体的过程，即从初步的方案设想演进为可信的系统规约，这个过程需要花费时间。在 **VisualSim®**中，系统模型设计（建模）的过程本身实际上就是对设想架构的研究、思考和细化的过程，这是一个交互的过程：在 **VisualSim®**

中用建模的方式描述你的架构方案设想；**VisualSim®**模型仿真结果回答你所关注的问题；分析这些结果，了解系统特性及其影响因素；评估是否需要对模型进一步改进、优化或细化，再次执行仿真…

在 **VisualSim®**中，这种交互式的过程本身就是对系统架构进行定义、设计和研究的过程。系统设计师正是借助这种演进式的建模与仿真方法来得到明确可信的系统规约。

4.2 Spending Time Designing Rather than Coding

与经典的使用 C/SystemC 编程进行事务级建模的方法不同，**VisualSim®**通过预定义的参数化组件库来加速建模过程：VisualSim 提供通用组件库和模型生成模板，建模工程师在图形化的环境中选择需要的模型组件，通过组装与配置这些组件来快速构建系统模型。**VisualSim®**组件库中很多模型的复杂程度与数万行的 C/SystemC 编码相当，并且这些模型都已经过充分的验证。

VisualSim®让用户将更多的时间用于设计、研究和尝试各种可能的系统实现方案，而不是把这些时间花费在编写、调试和验证 C/SystemC 模型代码的工作上。

4.3 Go Complex Only if Needed

VisualSim®定义了三种不同的建模层次，在设计的最初阶段，系统模型可以被抽象成为仅由特定时间分布的事务激励以及处理这些事务所需的延迟功能或资源来描述。之后这些延迟功能或资源可以被时钟精度级模型所替换，实现更细节层次的建模。上述替换是根据设计师的意愿而有选择地进行的。**VisualSim®**强调建模的目的是回答真正关注的系统架构问题（Model Questions），根据 Model Questions 来合理选择建模层次，**VisualSim®**允许用户将不同层次的模型混合使用，只在需要时再进行模型细化。

VisualSim®建议用户以解决真正关注的问题为重心，从简单的模型层次开始，只细化关注的部分，节省系统建模时间。

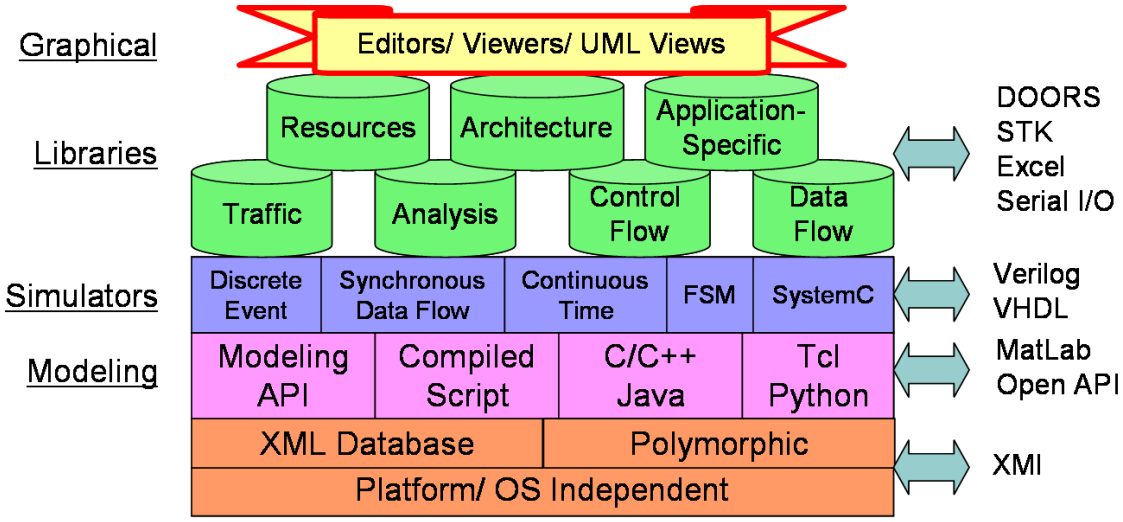
5. VisualSim®产品

5.1 VisualSim® Architect

VisualSim® Architect 是一个用于层次化系统建模、交互式仿真执行和分析的图形化工具环境。借助 **VisualSim®**的模型组件库和用户定制模型，用户可以在短时间内构造复杂的系统模型。

VisualSim® Architect 提供对基于模型的系统设计方法的全面支持，包括事务级、时钟精

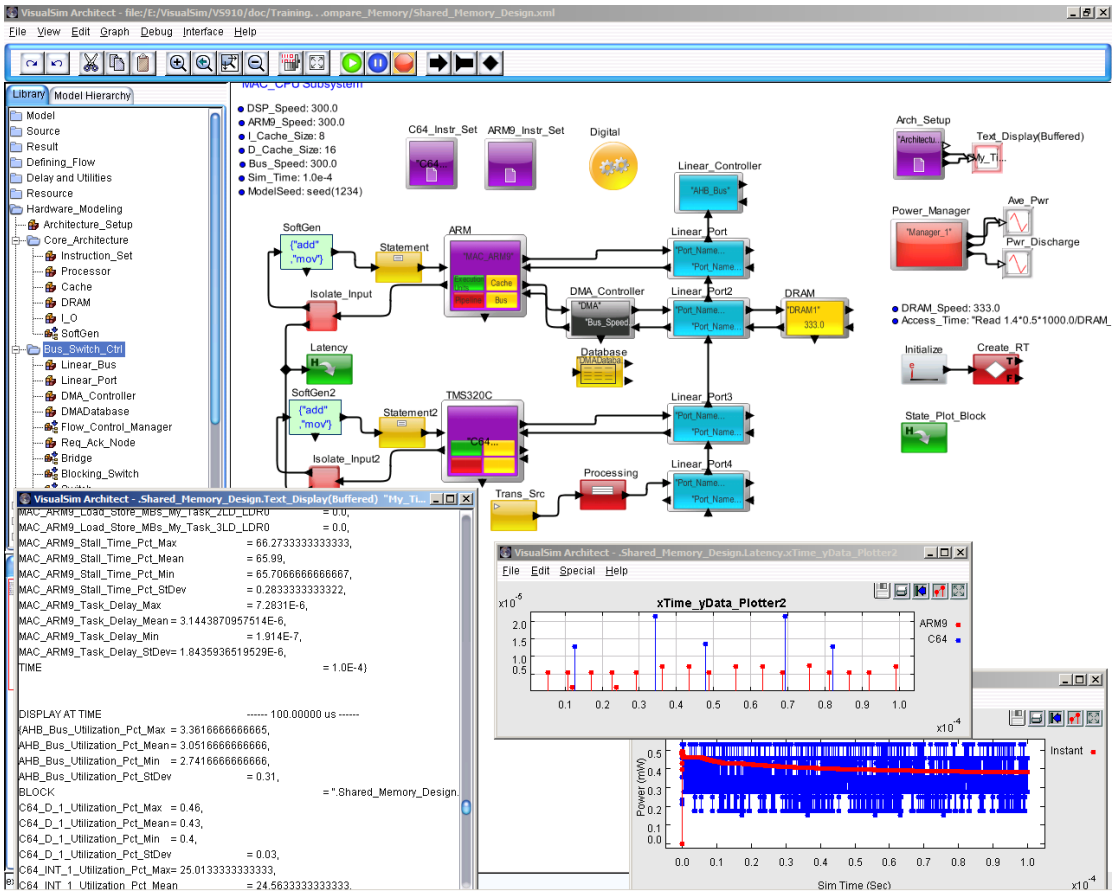
度级建模和设计架构验证。**VisualSim® Architect** 包含完整的建模组件库、仿真引擎、XML 数据库、报告生成工具和图形化调试工具，其软件与接口构成如下图所示。



VisualSim® Architect 具有优化的多域（Multi-domain）仿真引擎，可以实现同步数据流、事件触发、SystemC、模拟量代数/微分方程、有限状态机等不同类型计算模型的仿真。**VisualSim®**仿真引擎全面兼容 OSCI SystemC 内核。

在 **VisualSim® Architect** 的图形化框图编辑环境中，用户利用预先编译好的参数化组件库，或者已有的 C/C++/Java/SystemC/Verilog/VHDL 模型对设想的系统方案进行架构建模。参数化的组件库中包括各种用于事务级激励发生、事务流程定义与控制处理、资源与架构建模、结果输出与统计报告、应用算法处理等功能的组件。每种组件根据其不同特性而具有特定的配置参数集，用户通过更改这些参数来定义模块的功能特性。

VisualSim® Architect 允许在其模型的任何节点放置探针，用以收集模型信息、生成用户定制的图表或统计报告，回答用户关注的系统架构问题。通过在图形化环境中更改模型组件的拓扑结构、调整系统的全局参数或模型组件的配置参数、改变输入激励方案等，用户在**VisualSim®**模型的帮助下研 究和评价设想方案的特性，进行系统性能、功能、功耗等设计要求之间的平衡折衷。



在 **VisualSim® Architect** 中，设计优化可以通过对全局参数和模型组件参数的扫描实现。仿真输出结果则可以保存为文本或 Excel 表格用于后续处理，例如生成验证用测试向量集与断言。**VisualSim® Architect** 输出的模型是一个满足设计约束要求的系统实现方案，同时也是一个可执行的系统设计规约，交付给实现工程师作为设计依据。

5.2 VisualSim® Explorer

VisualSim® Explorer 是一个 Web 服务器程序，它允许用户将 **VisualSim®**模型嵌入文档中，供其他（远程）人员以 Web 页面的方式查看、执行仿真和分析。在该方式下，直接嵌入文档中的仿真模型的查看与执行是通过访问 **VisualSim® Explorer** 服务器实现的，不要求在本地下载、安装任何 **VisualSim®**软件。

借助这种方式，模型设计者可以非常方便的进行仿真演示，或与其他人员远程交流、讨论使用 **VisualSim® Architect** 创建的模型。该技术可用于设计团体沟通，设计方案评审以及早期客户需求沟通、客户支持工作等。

5.3 VisualSim®通用组件库

在 **VisualSim® Architect** 中，使用类似 UML 的图形化框图编辑环境进行建模。大部分模型可以使用一系列预先编译的模型组件构建：从模型组件库的文件夹中选择需要的组件；拖

放到框图编辑器中完成实例化；配置组件实例的参数；用连线联接各组件实例的端口、或以虚联接动态映射的方式实现拓扑结构设计。

VisualSim®组件库采用目录式的分类管理，一系列具有相似功能的组件被归入一个文件夹或子级文件夹中。例如，Resource 文件夹下的 Scheduler 子文件夹中包含三种类型的调度器组件，分别用于不同情况下的总线仲裁、处理器或 RTOS 的调度操作建模。一个模型组件可以是一个基本结构，例如激励发生器或调度器，也可以是一个复杂的元件，例如一个处理器。每个模型组件包含一组控制自身执行的参数集，例如，事务激励发生器组件包含启动时间，事件概率分布函数类型和概率分布函数参数等。模型组件库构成了 **VisualSim®**快速建模能力的基础。

VisualSim®组件库中的所有组件都支持事务级建模（TLM），并且其数据类型是多态的（Polymorphic），即同一个组件可以处理多种不同类型的数据，数据类型仅在运行时刻决定。这种灵活性允许同一组件或模型在多个用例或想定（Scenarios）中重用。此外，一个组件可能有多种仿真执行模式，以其参数设置为基础来确定。例如，某些数学功能类的组件有 10 余种不同的操作模式，而像多维优先级队列和处理器组件这样的复杂组件，则具有多达 45 种不同的操作模式。

VisualSim®模型组件的数据类型多态性、多操作模式特性在很大程度上减少了组件库中所需要的组件类型的数量，同时也减少了建模过程中需要实例化的模型组件的数量，因此可以用较少的内存占用实现大型、复杂系统的建模，并具有更高的仿真执行性能。这些重要特性不仅减少了仿真器的额外开销，也降低了新用户学习 **VisualSim®**的难度。

Source 库

Source 组件库为用户提供了可以在不同抽象层次上使用的事务级激励模型，例如激励发生器组件，可以按照固定时钟、一定概率分布（正态、指数、均匀分布等）、或由某个事件触发而产生激励。File 组件可以从文本、Excel 或 XML 文件中导入需要的数据，例如预设的事务激励序列和时间间隔序列，或是以不同途径得到的任务、指令记录（Trace）文件等。

Analysis and Result 库

在 **VisualSim®**组件库（特别是资源库和架构建模工具箱）的模型组件中，预定义了超过 3000 种统计信息报告（如 Cache 命中率、总线利用率等）。Analysis and Result 库中的组件作为模型中其它组件输出数据流的接收器，提供对事务数据的动态统计和结果的可视化输出。数据结果可以以条状图、直方图或分布图等方式显示，也可以写入文本、XML 或 Excel 文件中。从外，该库允许用户自定义其所关注的仿真结果或统计，例如某处理环节的中间延迟，吞吐量或数据包大小与传输延迟之间的比值等。

Resource 库

Resource 组件库中包含队列（Queue），通道（Channel），流水线（Pipeline），调度器

（Scheduler）和服务器（Server）等。这些组件用于定义系统资源，即那些在系统性能模型中会被消耗的时间和数量资源。例如，任何数据处理和数据传输过程都会消耗时间资源，而数据传输过程对总线（或传输信道）的占用则是对有限的数量资源的消耗。“资源”是**VisualSim®**模型中一个非常重要的概念，Resource 组件库中主要定义有两类资源：时间资源和数量资源。两类资源的消耗是相互联系的，例如，数量资源的限制使得多个设备同时访问总线时需要仲裁机制，而仲裁机制的实现必然同时带来时间资源的消耗。

VisualSim®模型能够帮助设计师分析系统“资源”与其多个“消耗者”之间的复杂关系，而正是这种关系决定了系统的性能。因此，Resource 库构成了 **VisualSim®**的事务级建模和性能建模等功能的基础。在 **VisualSim®**性能模型中，资源组件可以用于表示无线信道、总线、网络、处理器甚至是像卫星一样的大型系统等，资源的分配方式可以是共享式应用，也可以是分布式应用。Resource 库中的调度器组件用于处理共享资源的冲突与仲裁，将来自不同消耗者（如任务线程或总线访问设备）对同一资源的请求按指定算法进行调度处理。调度器支持多层次嵌套，基本的调度算法包括先到先处理（First-Come-First-Serve）、分时轮询（Round-Robin）或基于优先级的抢占式调度等，用户还可以使用脚本语言创建自己的仲裁算法或方案。

Resource 库中的所有组件都能生成与性能相关的各种统计数据，如缓冲使用率、队列利用率、数据进入和输出速率、事务和处理延时等。在 **VisualSim®**性能模型层，通过 Resource 库中的组件能够实现对各种复杂的硬件和软件资源的建模，在不需要更多细节实现信息的情况下做到相当精确的系统性能分析。之后，可以再依需要将性能模型进一步细化为时钟精度级模型。

Processing 库

Processing 组件库用于构建 **VisualSim®**模型的事务流框架，即对系统资源、行为和仲裁之间逻辑关系的描述。库中包括基本的条件判断、选择等控制组件，能够简化模型复杂配线的动态映射和虚联接组件，还可以方便地使用类似 C 语言的 **VisualSim®**脚本语言和相应的建模 API 函数来进行更为复杂的逻辑关系描述。

Finite State Machine

该组件库支持类似 UML 的图形化的状态图描述，并支持层次化的状态定义与转换条件定义。在 **VisualSim®**的状态机设计环境中，状态机的每个状态都可以附加与性能模型有关的信息，因此每个状态都可以独立地进行同步控制。

6. VisualSim®工具箱与接口

以下工具箱或接口模块均需要 **VisualSim®** Architect 软件支持。

6.1 VisualSim® SmartMachine 脚本（SmartMachine® Script）

VisualSim®脚本是一种语法类似 C 语言的描述语言，可用于构造算法、定义复杂的逻辑关系或协议状态图、功耗分析、定制统计数据等。**VisualSim®**脚本的主要特点包括：类似 C 语言的语法风格、易于学习和部署、本地代码（Native code）以提高执行性能、内置 JIT 编译器、基于文本的调试器、很小的内存资源占用等。

VisualSim®脚本的完整解决方案称为 SmartMachine®技术，包括 uEngine 等可执行脚本组件、JIT 编译器及内建的专用建模 API 方法。**VisualSim®**脚本在运行时刻即时编译，在模型仿真过程中即时执行。脚本组件支持单步调试、代码跟踪和语法检查等功能。

在 **VisualSim®**模型中使用 SmartMachine®的三个主要目的是：加速处理密集型任务的仿真执行、对抽象组件进行功能细化调整、压缩繁冗的图形化描述。SmartMachine®技术可以实现对软件指令、算法和资源处理等的快速建模，以及对类似于交换矩阵、分布式处理器或图形加速控制器等这样复杂的硬件系统的快速原型开发。

6.2 VisualSim®处理器与外设建模工具箱（Processor and Peripheral Modeling Toolkit）

VisualSim®处理器与外设建模工具箱提供指令精度级和时钟精度级的通用硬件模型的生成组件，包括单核或多核处理器，存储器和其它外设设备等。工具箱包括能够生成标准、专用、或用户定制处理器的架构模型，CPU 总线、高速缓存（Cache）、SDR/DDR 存储器控制器、SRAM、DRAM、SDRAM、和 FLASH 等通用架构模型。借助该工具箱，用户可以在 4 小时内完成一个标准或者专有处理器的指令集仿真（ISS）模型，全部工作仅需要使用 Excel 表格和该处理器的数据手册，而无需任何编程。生成的处理器模型能够直接在 **VisualSim®**环境中执行指令代码，用于进一步的系统架构研究工作。

VisualSim®的处理器 ISS 模型能够支持现代 GPP/DSP 的诸多特性，如流水线操作、寄存器文件、并发多执行单元、片内数据和指令 Cache 等。处理器模型的主要配置参数包括处理器资源（Cache 和执行单元）、流水线结构和指令集等，通过不同的参数配置方案可以仿真多种处理器特性：如冒险（Hazard）、排空（Flush）、指令跳转预测、停滞（Stall）等流水线操作，以及上下文切换（Context-switching）、中断指令和数据 Load/Store 指令模式等。

VisualSim®的处理器 ISS 模型能够反映处理器的性能、功耗两方面信息。生成的处理器模型可以与工具箱中的其它外设模型，以及系统资源模型组合在一起，构成完整的处理器系统模型。软件指令可以在该处理器模型中执行，并生成多种统计信息。处理器与外设建模工具箱中的组件内建对超过 250 种标准与定制统计信息的支持，典型的统计信息包括利用率、处理器停滞时间、上下文切换时间、Cache 命中率、吞吐率、任务处理耗时、空闲时间等。在 **VisualSim®**的处理器 ISS 模型中，每个时钟周期的流水线和执行单元等的工作状态对用户都是可见的，而状态窗口可以给出处理器内部以及各种外设资源的活动情况指示图。

VisualSim®处理器与外设建模工具箱中的处理器生成组件是一个通用的模型框架，通过不同的参数配置方案能够支持绝大多数的商业处理器。如 ARM 系列的 ARM7TDMI，ARM920T，

ARM946/966/968E-S，ARM926EJ-S，ARM11xx，Cortex-A8，M3；PowerPC 系列的 7410，750，405/603e；TI 的 TMS320C6x 系列 DSP；Intel 的 Xeon；Renesas 的 SH4/5；Tensilica 的 Xtensa LX2 等。

VisualSim®处理器与外设建模工具箱极大地减少了系统架构师在构建 ESL 模型时所花费的时间。借助该工具箱，硬件开发人员可以构建商用的或自有的处理器、指令集、Cache、存储器、和总线模型。软件开发人员则可以利用生成的系统模型来验证实现设想、线程分配方案和最佳的代码流/数据流调度方案，这些工作可以在很短的时间内完成，甚至是在着手写具体代码之前。

6.3 VisualSim®总线交换与控制器工具箱（Bus Switch and Controller Toolkit）

VisualSim®总线交换与控制器工具箱提供总线、总线控制器、交换结构和其它时控主导（Control-dominated）的硬件结构的生成组件。该工具箱能够生成的模型包括共享式、点对点式、请求-回答式和环形总线；DMA、存储器和 I/O 控制器；总线桥接、交换和网格结构模型等。通过更改组件参数，用户可以利用这些通用组件来生成各种标准的和自有的总线模型。典型的总线模型配置参数包括仲裁算法、通道数量、速度、位宽、猝发操作长度等。

随着系统硬件开发逐步向平台化设计方法的过渡，设计的差异化和特色将主要取决于用户定制 IP 以及在拓扑结构、数据流规划方面的创新。**VisualSim®**总线交换与控制器工具箱让设计人员能够快速创建用户定制的片上或系统总线、DMA、存储器控制器等硬件模型，之后可以在 **VisualSim®**环境中快速构建复杂的拓扑结构、定义专有的数据传输协议、尝试不同的数据流方案和仲裁算法。模型会给出不同方案的性能，如数据吞吐能力、总线利用率和缓冲占用情况等，帮助系统架构师对不同的设计方案进行评估和选择。

VisualSim®总线交换与控制器工具箱生成的总线或控制器模型可以与处理器 ISS 模型结合使用，也可以在性能建模层单独或与其它资源组件结合使用。该工具箱能够加速用户构建通用或专有高性能总线、交换结构和控制器的模型，例如把 PCI-Express 或 AXI bus 的建模时间由 4 周缩短至 1 周。**VisualSim®**总线与控制器模型能够提供高于 95%的性能分析精度和高于 85%的功耗分析精度。

6.4 VisualSim®标准总线工具箱（Bus Standard Toolkit）

VisualSim®标准总线工具箱提供多种在工业、网络、航空和汽车领域广泛应用的标准总线的 **VisualSim®**组件库。组件库中有超过 30 种总线，包括 PCI，PCIx，PCIe，Rapid I/O, AMBA 2.0，AMBA 3.0，CoreConnect（PLB 和 OPB），FlexRay，CAN，以太网端口和路由结构，VME, SPI 3.0 等。

标准总线工具箱（组件库）是在 **VisualSim®**总线交换与控制器工具箱基础上扩展而成的，提供经过验证的业界各种标准总线的 **VisualSim®**组件模型。模型中的控制器算法、仲裁机制和时序控制等都遵循相应的规约或标准的规定。标准总线组件库中的组件由 **VisualSim®**通用组件和定制的 **VisualSim®**脚本共同构成。这些标准总线组件可以和其它架构模型，如处理器、

存储器和 RTOS 一起构成完整的系统模型。

6.5 VisualSim® Xilinx FPGA 建模工具箱（Xilinx FPGA Modeling Toolkit）

VisualSim® Xilinx FPGA 建模工具箱包括 Xilinx FPGA 平台中的各种硬 IP 或软 IP 核的模型，主要有 PowerPC，MicroBlaze，DMA，SDR/DDR 存储器控制器，CoreConnect 总线（PLB 和 OPB），FSL 总线，块 RAM 等。

VisualSim® Xilinx FPGA 建模工具箱提供对 Xilinx Virtex 系列 FPGA 硬件平台的快速原型建模和架构设计优化。利用预先编译好的标准 IP 的事务级模型，用户可以图形化的原型开发环境中快速构建设想的 FPGA 硬件架构方案，在添加适当的数据流量和软件任务激励后，就可以实现对 FPGA 片上系统的 ESL 仿真，得到系统性能特性与设计架构之间的关系矩阵。

不同于 Verilog/VHDL 这类需要细节实现信息的 RTL 仿真，**VisualSim®**将复杂电子系统 ESL 设计方法学中的 TL 建模与仿真技术应用于 FPGA 片上系统设计领域。借助 **VisualSim®** Xilinx FPGA 建模工具箱，高性能 FPGA 片上系统的设计人员在产品定义阶段就可以开始进行快速和深入的性能－架构平衡折衷研究，发现系统性能的瓶颈所在，从而正确、合理地选择 IP 资源和系统实现架构。

6.6 VisualSim®功耗建模工具箱（Power Modeling Toolkit）

VisualSim®功耗建模工具箱是 **VisualSim®**平台下的系统级功耗估算解决方案，用于动态计算某一器件或整个系统的瞬间、平均及累计功率消耗。目标模型可以是一颗 SoC，也可以是一个分布式网络系统。功耗估算结果在规约阶段就可以得到，用户不需要任何编程就可以看到这个估算结果。该解决方案可以帮助系统工程师在统一的架构模型设计环境中对系统的功耗和性能进行平衡折衷。

VisualSim®的功耗管理组件根据模型中每个器件的工作状态来动态更新其瞬间、平均及累计功耗。为已有的 **VisualSim®**模型增加功耗估算功能非常方便，只需在模型上增加功耗管理组件、并对每个需要分析的器件添加确定的功耗属性（参数）即可。用于功耗分析的参数值既可以是示意性的相对值，也可以是从数据手册中得到的准确值。功耗分析不需要细节的软件代码、RTL 或布局布线信息。

VisualSim®的功耗管理组件不仅支持其它 **VisualSim®**工具箱中的架构层组件，同时支持调度器这样的资源组件，以便在系统设计初期就能够进行性能层的功耗估算，进行不同架构方案的功耗对比评估。此外，功耗管理组件还可以与 **VisualSim®**脚本结合，为 **VisualSim®**环境中的非 **VisualSim®**组件（如外部 ISS 或 SystemC 模块）提供功耗分析功能支持。

6.7 VisualSim® SystemC 接口

VisualSim® SystemC 接口组件是在 **VisualSim® Architect** 工具环境中扩展 SystemC 模块的接口工具。该接口允许用户充分利用 VisualSim®中预先编译好的丰富的 TL 模型库，将其与

用户定制的 SystemC 模块很好地结合起来，为开发基于 TL 的复杂系统模型提供高效、统一的图形环境。根据用户的实际经验反馈，在一个大型开发项目中使用这种方案可以节省将近 2 人年的工作量。

在使用 **VisualSim®** SystemC 接口建模方案时，用户仅需要对专有 IP 或核心部件进行 SystemC 模型编码，其余外围系统环境模型，例如网络和信道、事务激励（流量）生成、 通用队列和统计分析报告等，则直接从 **VisualSim®**通用组件库中选择相应组件并进行实例化即可。系统顶层的建模、仿真仍在 **VisualSim®**的图形化环境中进行，以图形化方式维护模块间的拓扑连接关系，并可以充分利用 **VisualSim®**的图形化侦听、动画执行和系统级断言等调试特性。通过 SystemC 接口组件可以导入用户创建的 SystemC 模块或已存在的 IP 或其它源文件。

6.8 VisualSim® HDL 接口

VisualSim® HDL 接口用于实现与业界标准的 Verilog 仿真器进行协同仿真。

VisualSim®提供基于 VPI/PLI 的接口，以实现与符合标准的 Verilog 仿真器进行协同仿真。VHDL 用户则可以使用 SystemC 或 C 编程接口的方式实现与 **VisualSim®**的协同仿真，例如，用户需要时我们可以提供用于与 ModelSim VHDL 仿真器接口的定制 FLI 接口。

6.9 VisualSim® STK 接口

VisualSim® STK 接口实现 VisualSim 与 Satellite Toolkit 的协同仿真，以满足动态轨道研究领域资源架构建模分析的需要。如果需要这方面的进一步的信息，请与我们联系。