



电子产品架构设计、性能仿真分析 系统解决方案 - VisualSim®

Mirabilis Design公司是业界领先的复杂电子系统与嵌入式软件性能分析和架构设计工具提供商，其产品VisualSim®是一款基于加州伯克利大学Ptolemy仿真内核的软件环境，能够同时支持数字、模拟、DSP及嵌入式软件等系统部件的建模与仿真。VisualSim专有的快速虚拟原型可使设计团队在项目开发早期即对系统的架构、性能、成本、功耗等指标进行研究，并对设计方案进行评价。VisualSim软件能够发布系统模型的Applet版本，这些模型包含了设计规约（Specification），并能在普通的Web浏览器或Word文档中查看和执行。借助Architecture Generators 和SmartBlocks等VisualSim的革命性技术，能够快速创建系统模型，并执行系统层次软硬件体系结构的“What if”的分析。

VisualSim® 仿真模型库介绍

在 VisualSim 中，系统模型的创建是通过在图形化的结构框图编辑器中组合不同的建模模块（图标）来完成的。若干建模模块被组合在一个文件夹中称之为库，每个库中的一系列模块具有同相似的功能类型。例如，资源（Resource）文件夹下的调度器（Scheduler）库包含三种调度模块，分别适用于模拟不同情况下的总线、处理器/DSP、ASIC、FPGA 和 RTOS 的调度操作。类似的，源（Source）文件夹中包含各种信号源、基于不同概率分布函数和记录驱动（Trace-driven）的事务/负载发生器。

用鼠标将库文件夹中的模块拖入框图编辑器进行模块的实例化，编辑、设置好各模块的参数表，再用类似导线的黑色连线描述各模块的端口连接（或采用虚拟连接）即可完成在 VisualSim 中的模型创建。每个模块拥有唯一的参数集，通过它能定制模块的功能来完成特定的任务。例如，数据流（Data Flow）库中的 DS_Expression 模块有四个参数，用户通过在其中一个参数域中输入 “Input.data = FFT ({1.0+2.0j, 3.0+4.0j})” 即可设定一种信号处理功能。

VisualSim 模型库

库是根据其操作特点来分类合并的。例如，用来计算处理延迟和在仿真过程中绘制处理延迟曲线的模块都归入 Result 库中。类似的，描述 DSP 通信算法的模块都归入 Application-Specific Digital Signal Processing 库中。

每个库中的所有模块都具备数据类型多态性，就是说待处理的数据类型取决于运行时刻。例如，单一的 FIR 滤波器模块可以同时支持浮点、定点或复数的输入数据类型。这种灵活性允许同一模块或功能应用于多种不同的情况下。在 VisualSim 中许多库模块还具有仿真域多态性，即同一模块可以在不同的仿真计算模型中表达同一个功能。例如，同一个正弦波发生器模块可以分别用在离散事件、连续时间和同步数据流的仿真引擎中。如数学计算功能模块之类的基本模块大概有 10 种不同的操作模式；而复杂一些的模块如带有优先级的多维队列模块就有 45 种以上不同的操作模式。模块的操作模式是建立在模块的属性设置之上的。

数据多态性和仿真域多态性的特点可以在很大程度上减少仿真库中模块的种类和数量，这同时也减少了需要实例化的模块的种类和数量，从而使创建大型、复杂的模型成为可能，并具有更高的仿真执行性能。因为 VisualSim 中不需要任何专门用于实现类型转换的模块，这一重要特性不仅减少了对仿真器的额外开销，同时降低了 VisualSim 用户的学习难度。

结构框图编辑器提供对每个模块的操作、参数和端口的说明文档。多种调试工具如侦听器用于监听端口信号和内部模块进程。此外，每个模块都可以设置断点，整个模型还可以以动画效果测试仿真流程的正确性。

利用 VisualSim 库可以在使用相同模块的情况下定义不同抽象层的模型，进行不同类型的分析。几个例子如下所示：

1. 一个卫星系统可以应用网络库进行建模，分析其与外部的网络通信及飞行器之间的通信互联。STK 接口模块可以被用来接收电池状态、地理位置和太阳耀斑的动态轨道数据。通信链路可以定义为一定通信速率的信道连接以进行性能分析，或用通信（Communication）库中的编码、解码、纠错和 Gaussian 噪声模块进行功能考察。类似的，飞行器的硬件和软件在建模之初可以被定义为一个具有处理延迟（时间消耗）的功能模块，就象应用类似统一建模语言（UML）来描述一样。之后，算法或行为可以用表达式语言或 VisualSim SmartMachine 脚本语言来定义；而实现结构则应用资源（Resource）库来定义。系统实现的测试和验证工作可以通过 VisualSim 标准外部接口如半实物仿真（HIL）、网络套接字（Socket）、Verilog、VHDL 和 C/C++ 进行数据输入。对上述系统的不同层次的分析可以包括：优化 64 QAM 调制的波形，端到端数据传输延时，整个传输网络的有效吞吐量，处理器、FPGA 和 ASIC 之间的资源共享，以及器件在总线上的最低利用率等。

2. 一个使用 PCI-Express 或 Rapid I/O 的数据交换机可以看做是对输入的一个序列的并行队列数据进行输出重定向和分配仲裁，这可以用资源（Resource）库、控制流（Control Flow）库和 SmartMachine 快速进行建模。之后，使用 Traffic 库模块及表达式语言可以快速生成大量的系统负载（输入激励），这些输入事件的发生时刻可以定义为基于某种概率分布，也可以是由一个数据序列驱动。按照设计规约给出系统负载后，通过 VisualSim 中的仿真分析得到交换机的具优先级的输出处理延时、协议处理开销对交换性能的影响、根组件（Root Complex）所需高速缓冲的

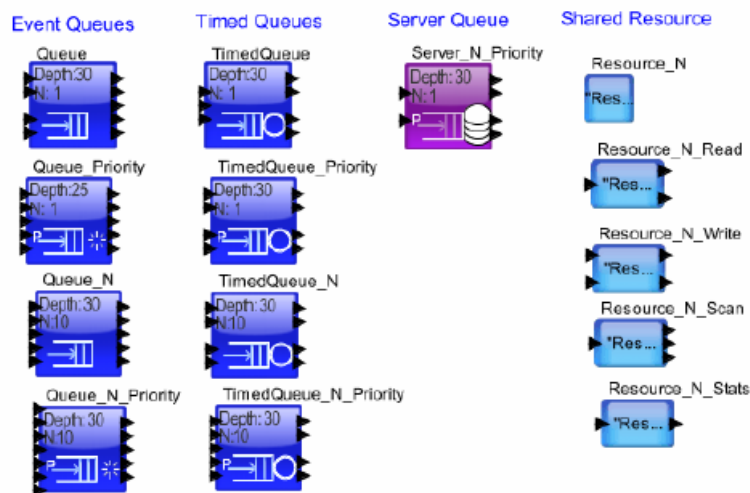
大小、性能要求以及交换系统的有效吞吐率。

标准库

Source: 提供输入激励，工况负载和各种事务触发发生器。库中的模块可以是信号或时钟发生器，按照预设的时间间隔（正态、高斯、循环、均匀分布等）生成事件。也可以用表达式语言和Traffic模块定义特殊事件分布函数。**File**模块可以从Excel文件、XML文件和文本文件中导入数据，例如预先设计好的软件线程执行列表，或是仿真程序运行时得到的任务、指令跟踪记录文件。

Result: 这些模块作为其它模块输出数据的接收器，提供对数据的实时动态统计及统计结果的可视化输出。这些模块可以将接收到的数据写入文本、XML或Excel文件中，或者接收图像、视频或音频等不同标准的文件，如BMP、TIFF、GIF、MPEG、PNG和WAV。此外，图形观察器支持XY轴、条状图、直方图和分布图等显示格式。该库中还包含有标准统计结果计算器和累加器等。

Resource: 该库中的模块用来描述系统中的处理资源，从简单队列模型到复杂的处理器和ASIC。这个库是构成VisualSim架构建模与仿真的基础。**Resource**库中的模块支持共享式或分布式应用，并且包含复杂的调度算法如先到先处理（First-Come-First-Serve）、轮询调度（Round-Robin）或优先级抢占式调度。库中的所有模块都能自动生成与处理性能相关的各种统计数据，诸如队列利用率、缓冲使用率、数据进入和离开队列的速率、数据在队列中转换和处理延时等。**Resource**库中另外一个重要的模块类型是调度器（Scheduler），软件或硬件的调度器可以模拟RTOS、处理器或总线控制器的工作，将来自不同位置（多线程、任务或总线端设备）的请求按指定算法进行调度排队处理，软件调度器同时支持多层次嵌套的抢先式调度机制。将**Resource**库中的队列模型与调度器模型相结合可以对各种复杂的硬件和软件资源进行建模。



Processing: 该库提供对系统行为、处理流程、数据流、控制流和算法的的定义的基础框架。此外，

Processing库中的模块包含独特的用于加速仿真执行速度，减少复杂仿真流程模型配线的技术。这些模块与VisualSim表达式语言完全集成，可以通过对参数和端口配置进行充分扩展。

Finite State Machine: 是一个类似于统一建模语言（UML）的图形化的状态机描述。状态图支持层次结构设计。此外，状态机的每个状态可以附加与执行性能有关的信息，因此可以对每个状态进行独立的时控。

Virtual Prototyping: 该库可以将VisualSim模型进行动态化和可视化，用作设计三维的动态图形。该库使用Java-3D来最大程度的减少对图形场景的重复计算。一个模型可以包含一系列三维图形元素，如球、气缸等。VisualSim模型控制在虚拟样机环境中所定义三维图形的位置、旋转和尺寸，形成一个三维动画场景。通过三维可视化虚拟样机，可以提供了对系统功能模型和性能模型的更直观、深入的了解，这是非常有趣和有用的。

接口

VisualSim 提供了大量标准应用程序的接口，包括开放 API 以便于集成第三方应用程序。

1. 数据输入输出：支持 Excel 和其它电子制表软件，文本文件。
2. 行为仿真分析：将 MATLAB 功能嵌入在表达式语言中，所有的 MATLAB 函数在 VisualSim 中均可以通过表达式语言模块实现。例如，在 VisualSim 表达式模块中，一个 FFT 函数可以通过 VisualSim 表达式语言格式定义，也可以使用 MATLAB FFT 格式定义。
3. 动态轨道数据：VisualSim 对 STK（卫星工具包）的接口提供了高性能、多种查询方式、双向存取轨道数据。可以籍此研究指定的卫星和已存在的卫星通讯网络的相互作用，确定对硬件系统的能量和重量的要求。
4. Verilog/VHDL 仿真器：协同仿真接口将 VisualSim 集成于硬件设计流程中，用于对 FPGA 和 ASIC 系统的设计验证。该接口可以在结构框图中生成用户模块，使用 VPI（Verilog）或 FLI（VHDL）实现 VisualSim 模块与 HDL 仿真器模型的直接接口。
5. SystemC：SystemC 导入模块允许用户导入已经存在的 SystemC 模型而不需进行任何更改，因此可以方便在 VisualSim 中添加第三方提供的 SystemC 处理器模型。另一方面，VisualSim 提供一个 SystemC 模块（SC_Module）模板发生器，可以直接生成具有全部所需端口定义和参数的 SystemC 模块框架文件。

上述两种 SystemC 模块可以和其它 VisualSim 基础模块一起在 VisualSim 结构框图编辑器中使用。这些基础模块包括：定义 RTOS 的调度器，定义控制流程的 FSM 编辑器，可以进行混合信号仿真的连续时间仿真器，各种激励发生器和分析工具等。另外，一些 VisualSim 工具可以用

来加速 SystemC 模型的开发。例如：在 VisualSim 环境中可以在 SystemC 模型中调用 MATLAB 方法 (Methods)；以 VisualSim 集中管理 SystemC 模型库；在结构框图编辑器中以图形化的方式表达和管理各 SystemC 模型之间的互连。

OSCI SystemC v2.1 仿真器已经是 VisualSim 的标准产品包，所以不需要额外的安装步骤。为了保持完全符合 OSCI 规范，SystemC 库模块是与其运行平台相关的，这与其它的 VisualSim 库模块有所不同。VisualSim SystemC 库模块支持众多编译器 (gcc / Borland C++ / Visual Studio) 和操作系统 (Windows / Solaris / Linux)。

6. 半实物仿真：使用 datagram 套接字接口。

7. 串行 I/O：接口用来访问可编程板或操作控制设备。

8. X10：该库通过串口模拟 X10 接口设备。包括 CM11A 串行模块 (有线) 和 CM17A 串行模块 (无线) 通信模型。无线串行模块是单向的，只用来发送命令给 X10 网络，不从其接收数据。

8. 其它标准计算机外设：操纵杆、键盘、音频和视频设备。

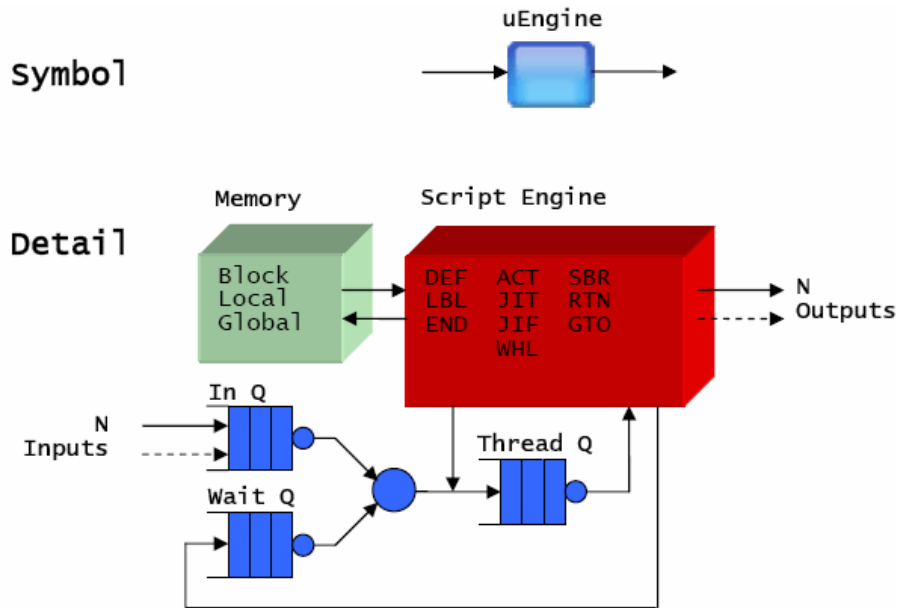
表达式语言

VisualSim 表达式语言模块规定了如何以文本形式描述代数表达式及对表达式进行评估的方法。表达式的计算结果可以用来对一个模块的某个参数值进行赋值，或者用来确定一个状态机是否允许发生状态转换。表达式语言是标准数学分析工具箱一种高级形式，实际上，表达式语言是 VisualSim 基础构架的一个通用组成部分，程序员可以通过它来扩展 VisualSim。表达式语言具有仿真域多态性，可以应用在需要利用数学方程式的任何地方。所有标准的算术、逻辑、三角学、数组、矩阵、字符串、图像处理、DSP 等操作运算都随 VisualSim 库一起提供。此外，该语言具有数据类型多样性，所有单一表达式可以实现不同的数据类型运算。VisualSim 提供内置的表达式解释和错误提示机制以便于用户快速调试表达式语言的功能。

VisualSim SmartMachine

SmartMachine 是 VisualSim 专用的用户开发脚本语言，用于对软件指令、算法和资源处理进行快速原型开发。SmartMachine 的优点是易学易用，由于内嵌支持所有可以提高仿真性能，用少量的代码就可以实现灵活的语言功能。应用 SmartMachine 主要有如下三种目的：显著的加速处理密集型的任务，对抽象模块进行细节性设计，或是希望以少量的代码代替大量的图形描述。

使用 SmartMachine 的通用指令和语法可以对硬件系统、嵌入式软件和复杂事务激励进行建模。例如用来进行复杂系统的功耗模型分析，协议的状态图表示，RTOS 性能评估和对复杂硬件系统 (如宽带交换结构、分布式处理器等) 进行准确建模。



SmartMachine 由可执行模块、即时（Just-In-Time）编译器和只有 10 个指令的脚本语言构成。模块脚本在运行时刻即时编译，在 VisualSim 仿真过程中时即时执行。脚本支持 16 线程的多线操作和并发事件处理。脚本执行模块可以访问 VisualSim 的图形资源模块如队列、存储器和调度器等。模块脚本可以进行单步调试，代码跟踪和语法检查。

应用、专用库

Image Processing: 该库提供对视频和图像处理系统进行设计、仿真和验证所需要的工具模块。图像库模块具有数据类型多态性，因此同一模块可以接受整数、数组、浮点数和定点数等不同输入数据类型。图像处理库提供全面的图像处理函数，庞大数量的可编程函数，可扩展的类似于 MATLAB 的表达式语言以及视频/音频设备接口。图像处理库的建立是基于业界标准的、开源 Java Advanced Imaging 和 Media Framework 组件，可以实现高性能的图像处理，如图像读写访问，图像操作、修正，图像的矩阵、数组操作和显示，DCT, FFT 等。表达式语言包含许多标准函数，可以直接在表达式中调用，并可以通过可编程接口进行扩展。

Networking and Protocols: 该库用来对网络元件、协议进行建模，或评估不同设备间的数据路由。芯片、电路板或网络中的计算机都可以被定义为一个网络节点。库中的 5 个模块分别用来定义网络节点、路由表、协议堆栈层、协议参数集以及网络控制器。用户路由算法可以由可编程接口提供，同时网络结构可以通过 Node Master 模块或由动态编辑路由表来重构。

Controls and Analog: 该库用来对模拟电路，机电一体化系统和控制系统等连续时间系统进行建模。该库的核心模块为一个积分器，在常微分方程（ODE）解析中起特殊作用。模拟控制库中包含波形发生器、步长控制器、事件发生器、微分和积分模型，以及采样保持系统模型、零阶及一

阶保持模型、过零监测器和阈值监视器等。

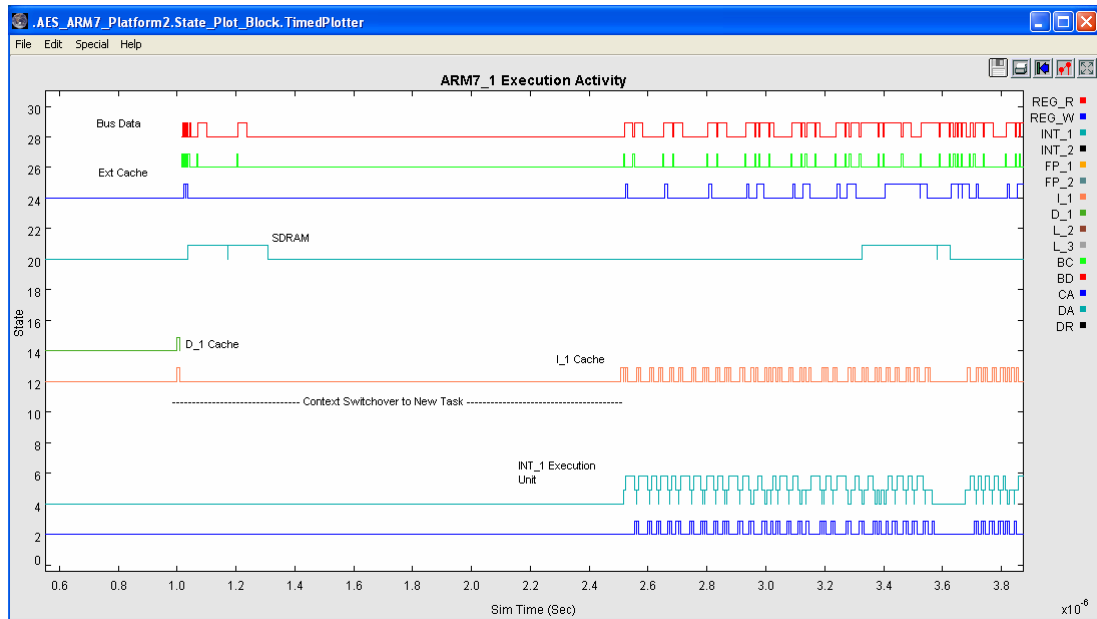
DSP and Communications: 该库提供基本的DSP函数和标准的通信函数。DSP库包含各种信号源模型, 量化模型, 滤波器(FIR、IIR、欠采样、过采样、内插等), 变换(FFT、IFFT、Levinson-Durbin等)和统计分析工具等。通信库提供的卷积编码、Viterbi和Hammar编码等通信系统物理层处理模型。其它一些模块如自相关处理、直方图和最大熵分析等便于对DSP系统模型进行分析和评估。所有的库模块均具有数据类型多态性, 支持整形、双精度形、复数形、定点形和浮点形。

VisualSim® Architecture Generator Toolkit

VisualSim 的硬件架构模型生成器 (Hardware Architecture Generators) 工具包能够生成复杂的 DSP 和通用/专用处理器模型, 该模型支持指令集仿真且具有时钟周期级的仿真精度。应用该架构模型生成器以及相关的硬件结构库, 可以在 VisualSim 框图环境中快速定义完整的硬件平台结构, 而无需为此编写大量 C/ SystemC 代码或是为描述指令集创建复杂的电子数据表格。

在定义明确的硬件平台模型后, 软件结构工程师就可以在其中运行所设计的软件框架、任务流程, 用于评估软件规约的正确性及软件执行(处理)的性能。而当这个硬件平台模型上运行了软件后, 就可以看作是一个虚拟的设计平台, 系统工程师可以据此来选择关键元件(如处理器), 优化器件的大小和速度(如存储器或总线桥), 以及定义仲裁算法等。因此, 这个以处理器为核心的虚拟平台可以保存为一个 VisualSim 模型, 供设计团队中的处理器架构设计师、系统工程师和软件结构设计师协同使用。

这项技术极大的减少了架构设计师和系统工程师创建电子系统级(ESL)模型所需的时间。借助 VisualSim 硬件架构模型生成库, 用户可以对通用商业处理器系统设计以及自定制处理器系统创建处理器、指令集、高速缓存、存储器以及总线的模型。VisualSim 的处理器模型的独特之处在于它能将独立的结构性能建模和行为建模统一在同一个模型中。例如, 完成对一个基于 DSP 的 FIR 处理系统的建模, 允许随意调整定点 FIR 滤波器系数的比特位宽, 或者是滤波器抽头数。通过运行仿真模型可以同时评估设计参数调整对两个方面的影响: FIR 滤波器截止频率、频响跌落等滤波质量的变化(即行为或功能分析), 以及系统执行 FIR 程序所需处理时间、系统功耗等的变化(即性能分析)。应用这种新的技术方法, 架构设计师和高性能计算系统工程师可以在短于一周的时间内验证他们设想的系统架构, 专注于特定处理事务对系统实现的影响。在设计最新的多处理器系统或者多处理内核 Soc 结构时, 这种设计方法就显得尤为重要。软件设计人员能在很短的时间内验证有关线程分配、最佳任务调度与加载等设想, 同时系统工程师可以依据虚拟系统平台进行性能和功耗等的评估。



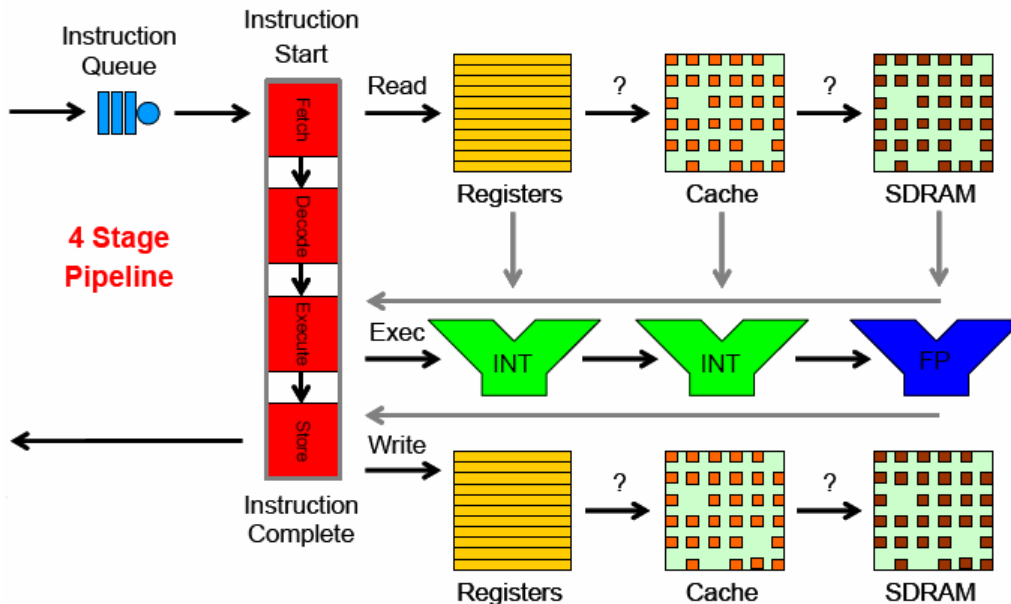
VisualSim 的处理器模型能够反映现代 GPP/DSP 的诸多特性，如流水线操作、寄存器文件、多执行单元、片上数据和指令 Cache（Harvard 体系）等。处理器模型的执行能够模拟实际处理器执行指令的并发流水线操作过程，以一个 4 级流水线处理器为例：第一阶段（Fetch）从最近的 Cache（如 L₁ 指令 Cache 和 D₁ 数据 Cache）中取出一个缓冲行；第二阶段（Decode）处理器等待对指令的译码和数据锁存；第三阶段（Execute）伪指令按照指定的执行周期数被执行；第四阶段（Store）处理器等待指令执行完毕，并保存操作结果。结果写入处理器寄存器还是 Cache 则取决于对处理器模型中寄存器（文件）数的设置以及当前运行任务所规定的寄存器数量。此外，当一个新任务或者外部中断输入处理器模型时，模型会按预置的任务切换时间（Context_Switch_Cycles）运行若干时钟周期，以模拟 RTOS 执行线程切换时所挂起的时间。

VisualSim 的处理器模型支持多执行单元（Execution Unit）仿真。流水线中的预取指令被划分为两个指令流：定点指令和浮点指令，并准备在流水线“执行”阶段送入相应的执行单元。执行单元是处理器模型的一种“资源”，因此可以任意设置处理器模型中定点和浮点执行单元的数量。当某条指令准备送入流水线的“执行”阶段时，模型将判断当前是否存在可用的执行单元资源，对单一执行单元系统即判断上一条指令是否执行完毕，对多执行单元系统而言则判断是否还有“空闲”的执行单元。如果没有可用的资源，模型中的流水线将暂停（Stall）执行直至所需资源有效。

在 VisualSim 的指令集模型中，每一条指令可以具有不同的执行周期数，且执行周期数可以作为一个范围内的随机数，以模拟某些位操作的实际执行情况。在指令集模型中还允许定义执行单元名称和指令集分组，用于约束某类指令只能在某个（类）执行单元中执行。

VisualSim 的这些模型机制非常适合于对一些新型处理器架构的建模，如 PowerPC 中的 AltiVec DSP 的 SIMD（单指令流多数据流）结构或 TI C6000 DSP 的 VLIW（甚长指令字）结构等。为满足对 DSP 处理器的特殊建模需要，VisualSim 的处理器模型支持跳转指令预测（Branch

Prediction), 可以指定跳转指令是否排空当前流水线操作。处理器模型还支持 DMA 方式, 允许某一任务申请 DMA 操作后挂起, 直至外部 DMA 模型完成操作后再继续执行; 或用于模拟 Load/Store 指令模式。

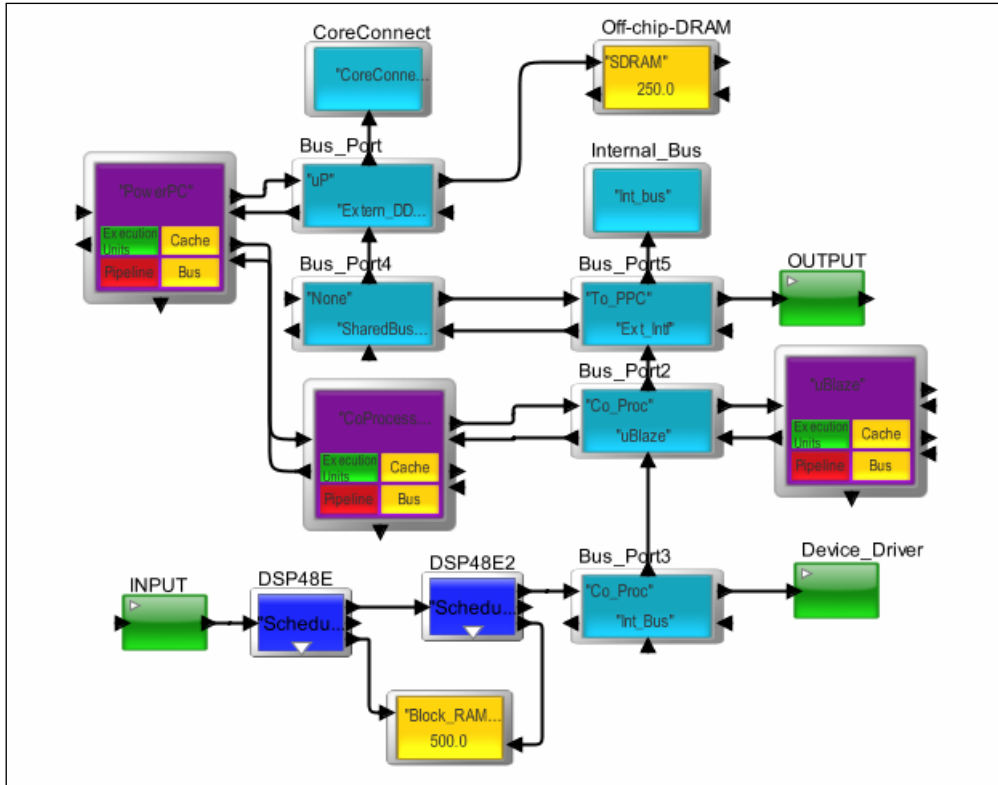


VisualSim 的硬件架构模型生成器目前可以提供的模块包括处理器与指令集、高速缓存、总线和 SDR/DDR RAM 模型。由于硬件架构模型生成器生成的处理器模型是“通用框架模型”，因此能够支持绝大多数的商业处理器。如 ARM 的 ARM Cortex-A8, ARM11xx, ARM9xx 及 ARM7 系列; PowerPC7410/750, 603e Core 以及 PowerQUICC 的 8xx; TI 的 TMS320C 系列 DSP 等。

VisualSim® FPGA Modeling Toolkit

随着商业 FPGA 的芯片规模和性能的急剧提升, 以超大规模 FPGA 进行 SoC 设计的技术趋势已经逐渐被系统工程师认可。VisualSim 即将推出的 FPGA Modeling Toolkit 能够帮助 FPGA 工程师快速构建图形化系统原型, 通过对 FPGA 系统设计结构的考察和性能分析, 确保在系统设计初期即可选择正确的 FPGA 平台或器件, 并得到优化的软件、硬件协同功能实现的划分方案。

借助 VisualSim 的 FPGA Modeling 工具, 设计师可以利用参数化模块库快速构建复杂的 FPGA 设计及周边系统的结构模型, 并根据仿真结果来评估系统设计的正确性、性能表现、布线面积及功耗等, 而这些分析、评估工作过去通常是在耗费大量时间、金钱得到实物硬件系统之后才可能进行。当一个 FPGA 设计中包含多个处理内核 (MicroBlaze、PowerPC 等) 或复杂外部交互接口时, 这种系统虚拟原型的设计方法显得尤为重要。



目前的 VisualSim Xilinx FPGA 模型库中包括: PowerPC 处理器硬核、MicroBlaze/PicoBlaze 处理器软核、CoreConnect 总线模型、DMA、中断控制器、DDR、Block RAM、LUT、DSP48E 等。