

MIRABILIS

design

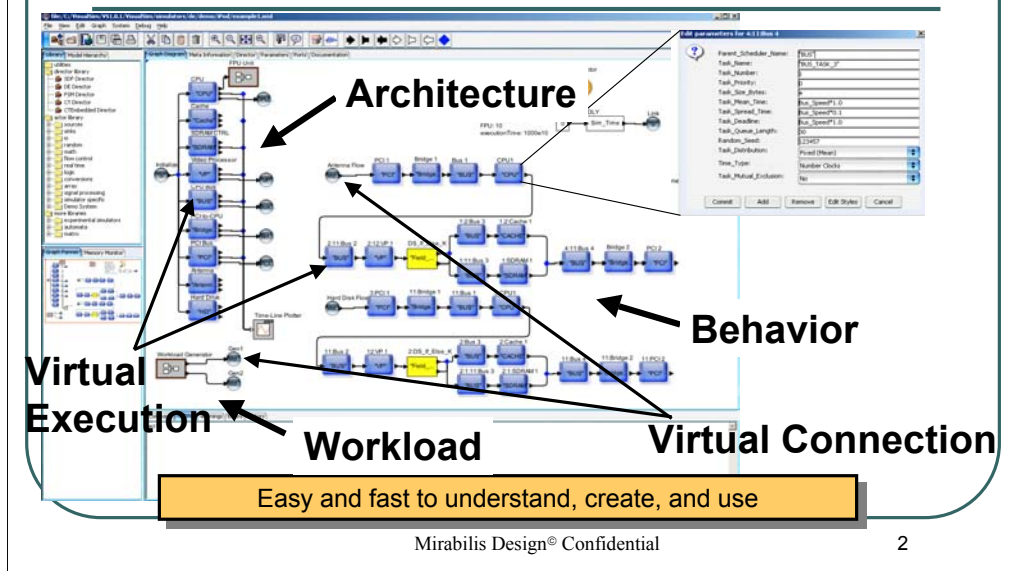
Implement Imaginations



How-to-Model VisualSim[©]

Welcome to our presentation of VisualSim from Mirabilis Design. In this presentation, we will cover the ability of VisualSim to model different aspects of a system including software, RTOS, hardware elements and communication protocols.

Block Diagram & Methodology



A VisualSim model is shown here.

- Each VisualSim building block has parameters that specify operation and performance of the block.
- A VisualSim system is described as a combination of architecture, behavior and workload.
 - Architecture. Architecture includes hardware, IP, blocks, RTOS, micro-engines and electromechanical.
 - Behavior. This describes the actions performed on the system. Example are activating the digital camera shutter, network traffic shaping or computer boot-up.
 - Workloads. Workloads are transactions that traverse the system and include network traffic, sensor signals etc. Each workload transaction in VisualSim is a data structure with multiple fields. The fields of the data structure carry modeling information such time started, priority, size, source and destination. In addition, trace files, XML data files or ISS can also be used.

Mapping between behavior and architecture is performed using Virtual Execution. These functions are important to reduce model clutter and increase visibility. The virtual execution capability makes remapping from hardware to software to ASIC by a simple change in the parameter.

In a standard model, the transactions are generated by the Workload. These transactions are connected to the behavior that defines the action. When data arrives at a particular behavior block, say the PCI, a request is sent to the equivalent block, in the Architecture. The PCI execution occurs in the architecture. The behavior block and the incoming data structure provide information to the architecture block. To refine the architecture, the output of the architecture block is connected to the detailed version. In this example, the CPU block has a detailed definition of the Floating Point Unit. Similar details can be added to the behavior.

Defining Workload

- Stochastic distributions
 - Standard: Poisson, Normal, Uniform & Fixed
 - Expression: Create custom distribution or feedback loops
- Trace-based
 - Use traces for enhanced modeling and accuracy
- File-based definitions
 - Define arrival rate, size, priority in .xls, .csv or .xml
- Connect to hardware using datagram

Use predicted or leverage existing workload

The workload defines all the scenarios and possible environmental conditions that the system-under-study will be operating. The quality of the workload determines the accuracy and completeness of the results. VisualSim provides an extensive workload generation tool for abstract and refined model analysis. For incremental software design and for field testing, the simulation can be connected to a hardware system that is on the network.

Modeling Behavior

- Behavior are the applications or instructions
- Modeled as a series of actions/ flow of data
- Execute on HW or SW through mapping
- Blocks provided
 - Task Issue, Data Structure Decision & Control
 - Model_List to create software profile (contains combination of source, destination, rate, priority & size)

Explore optimum dataflow
based on cost, power and performance

The behavior is key in providing a solution for the customer's requirements. Here the descriptions of the applications and operations are specified. The extent of these behaviors and the details provided is important in making accurate architectural decisions. These actions must be executed on hardware or software. The workload determines the key performance issues associated with implementing these actions. The implementation trade-offs can be made based on cost, power and performance criteria. Further optimization can involve consolidating some of these functions or rearranging them on different hardware and software. For example, to utilize simultaneous execution, the instructions can be separated between the CPU and DSP, thus reducing cost, averaging loading and increasing reliability.

Modeling Software

- Software definition uses
 - Model_List to define instructions and micro-code
 - Define behavior using Task_Issue and DS Control
 - Import C/C++/Java code using Wrapper
- Parameters includes
 - Source, destination, size, processing rate, priority & Mutex
- RTOS/ Task Overhead
 - Trace-based or context switching in scheduler
 - Delay based on Task_Size

Accelerate modeling by using existing intellectual property

In the current economy, new systems are released with software additions and no changes to the hardware architecture. If the architecture cannot support these new applications, the effects can be disastrous. The applications will not be able to maintain response time and quality perception will be greatly reduced. The entire software can be modeled in VisualSim or the new portion can be modeled and the rest can be abstracted as traffic. The third approach can be to connect the simulation to the hardware and model the new operations in VisualSim.

Modeling RTOS

- RTOS can be described as a state diagram
 - 6 states define the operation of an RTOS in VisualSim
 - (1) Dormant, (2) Waiting for event, (3) Running, (4) Delayed, (5) Ready, (6) Interrupted
- RTOS definition blocks
 - FSM, Model_List, Data Structure control
- RTOS parameters
 - Task Priorities
 - Task Properties (MUTEX, Fork, Join, etc.)
 - Task Dependencies (Events, Precedence, etc.)
 - Task Size (Memory requirements, # of accesses and hit ratio)
 - Task Times (Timeouts and deadlines)

Optimize RTOS performance & create implementation constraints

The RTOS is a critical link in the product development. The quality, reliability and performance of the software depends to a large extent on the tuning and correct selection of the RTOS. VisualSim provides capabilities to model and refine the RTOS. A number of parameters can be specified to analyze accurate operating conditions and provide software refinement.

Modeling Algorithms

- Architecture Algorithms
 - Type: Bus arbitration and memory controllers
 - Parameter: Clock, size, priority, rate & connections
- Functional Algorithms
 - Type: DSP, Imaging, MEMS and analog
 - Parameter: Bits, BER, cycles, amplification, harmonics, & power
- Model definition blocks
 - Basic blocks and DS control for architecture algorithms
 - Pre-defined DSP/Analog functions or custom-code
 - Link to MatLab or import C/C++/Java/SystemC

Combine algorithms with architecture & workload for accurate predictions

A system can have two different types of algorithms. One is the signal processing algorithms in the digital and analog domains. The other is the architecture-level algorithms such as memory controllers, bus arbitration and cache hit etc.

VisualSim has the capability to model both of these algorithms. Extensive capabilities are available to model the architectural algorithms including SmartBlocks, IO and control blocks. For the signal processing algorithms, limited number of commonly-used functions are provided. To have a complete system design analysis, link has been provided to include a MatLab simulation within the context of a system. Existing analog and digital algorithms modeled in MatLab can be imported into a VisualSim simulation without making any modifications.

Modeling Protocol and Network

- Networks can be on a chip, board or full-system
- Model definition blocks
 - Physical entity defined as a “Node Block”
 - Elements are attached to a bus on a IC or routers on a network
 - Connectivity information defined in Nodes
 - Protocol layers defined by “Layer Block”
 - Examples are PHY, MAC, IP, TCP
 - Functionality of each layer defined by “Layer Table”
 - Unique or consolidated for each layer and node combination
 - Routing algorithm & criteria defined by “Routing Table”
 - “Node Master” controls all network nodes
 - Modify connections & routing tables

Combine protocols with architecture

Today's complex systems have complex routing on the board and the chip. This requires modeling different devices on the chip or board as a node and evaluates the various operations between them. In addition, these have to function in a networking environment which requires an extensive analysis of the network protocols. VisualSim provides a comprehensive set of libraries for modeling networks and protocol stacks.

MIRABILIS

design

Implement Imaginations



How-to-Model VisualSim[©]

Thank you.