Approaches to Modeling

Introduction

The purpose of system design is to identify the components of a system, how they are connected to each other and how they interact with each other to deliver the desired functionality. Traditionally, the design of a system was captured using diagrams and natural language descriptions. This informal description of a system has many disadvantages as described below:

- It is difficult to share, understand and reuse a design due to the non-standard methods used to describe a system.
- Automation of the system design process becomes difficult due to the non-formal nature of system description.
- It is very difficult to verify the correctness of a design that is specified in an informal way.

The use of informal notations for system design results in longer design cycles and is more error-prone. This has led to the development of formal languages for system modeling and design.

Characteristics

System Description Languages are used to describe a system for the purpose of system modeling, simulation, verification and synthesis. The following are some of the desirable characteristics of a language used for system design.

- **Declarative:** A declarative language is more suitable for describing system components and their interconnections. It is much more easier to synthesize a system for a declarative description than from a procedural description.

- **Support for Design Abstractions:** The language should provide support for describing common system design abstractions like system, component, resource, relation etc.

- **Support Hardware and Software Components:** A system contains both hardware and software components and so the language should allow both type of components to be specified in a uniform way.

- **Behavior Independent Modeling:** The language should allow modeling at a higher level of abstraction so that the design is not dependent on the behavior of the components of the system.

- **Implementation Language Independent Modeling:** It should be possible to use components in a system description irrespective of the language in which they are implemented.

Examples of System Description Languages

- **SystemC:** SystemC is a C++ library that provides support for describing a hardware system and simulating its behaviour. SystemC provides support for concurrent processes. The main abstractions provided include modules, ports, processes, channels, interfaces and events.

  - Modules are the basic blocks of a system. A module provides some useful functionality.
  - Ports enable communication between modules.
  - Processes provide the behaviour of a module.
  - Channels are the communication elements. These include signal, buffer, fifos etc.
  - Interfaces describe the services provided by a channel. These are used by the ports to communicate with the channnels.
  - Events are synchronization primitives.

  SystemC describes a system in a programmatic way. While this is useful for simulating the behaviour of the system, it presents difficulties when the system is to be synthesized into hardware.

- **SystemVerilog:** SystemVerilog is an extension to the Verilog Hardware Description Language that includes support for system verification and ability to link with SystemC functions. This adds higher level design to be linked with the lower level of abstractions supported by Verilog.

- **MATLAB/Simulink:** MATLAB is a high-level language for algorithm development. Simulink supports model-based design and simulation of embedded systems. Models of pre-existing algorithm and structural components can be quickly combined to create a system model, which can then be simulated.

- **SANKHYA System Description Language (SSDL):** SSDL is a system description language that allows a system to be described at a higher level of abstraction. SSDL is a declarative language and provides support for describing components and their mappings in a language independent and behavior independent fashion.

SSDL Example

Consider a system with a single CPU and a memory device. The CPU is based on the DLX architecture. The memory device is a RAM module of 0x5000 bytes. The memory is mapped to the processor's address space at address 0x1000. The components of this system are the CPU and the memory device. The component types are the DLX model and the memory device model. The relationship includes the mapping of the memory into the processor's address space. The SSDL description for the above system is shown here.

SSDL Description of a Simple System

```
; System with a processor and a memory device
; System Model
model system System XDKSTDSystem.dll

; Component Models
; The processor model.
model processor DLX dlx.md

; Memory device model.
model device SRAM tc_core_sram.dll

;Component Definitions
component System default

; Define core processor
component DLX dlx_core

; Define SRAM of 0x50000 byte size
component SRAM ram 0x50000 rw

; Define relationship between components
; Map SRAM in the processor's address space
map memory dlx_core:address.0x1000 ram:STDRAM rw
```

Technical Support

Contact support@sankhya.com for more information.

SANKHYA™

Sankhya Technologies Private Limited
Contact:
+1 408-556-9757 (USA)
+91 94449 72818 (India and South Asia)
http://sankhya.com/contact.html