flows through them, and the kind of information that flows through them. In addition, the interface description may contain parameter declarations and assertion definitions. Parameter declarations allow making instances of library definitions. These parameters may describe characteristics of the entity such as the initial values, the no-connect values, the timing data, or the environmental characteristics. Assertion definitions allow the user to specify information about the intent of a design entity so errors can be detected closer to their source. Assertions can be used to check setup and hold times, electrical characteristics, and operating conditions.

Body description. The body description of a design entity specifies the operation and internal organization of the entity. The body declarations provide the capability to describe a design entity in structural or behavioral terms, or in a combination of the two.

A structural description is a description of a design entity purely in terms of its subcomponents, parts, and their interconnections, nets. The parts are modeled through the recursive process of hierarchical decomposition, and nets are modeled through the specification of their characteristics.

There are two types of nets: physical and conceptual. A physical net is composed of a single physical line, and its value is 0, 1, Z (high impedance), or U (unknown). A conceptual net is composed of a number of physical lines, and its value can be an integer. A merge/decomposition construct to support the interconnection between conceptual and physical nets is required.

With a purely structural description, nothing is known about how the design entity behaves unless the behavior of its parts is known. Any delays associated with the design entity are not specified here but in the descriptions of the contained parts.

A behavioral description is a language form which describes the behavior of a design entity. Behavioral descriptions should have conditional control and iteration primitives, logic and arithmetic operators, and hardware data types (register and memory, for example); support procedural (sequential) and non-procedural (concurrent) assignments; be directly executable as a simulation program and be adequate for synthesis purposes. The language should lend itself to the process of step-wise refinement by which a behavioral description is translated into a structural description.

Sometimes it is convenient to describe the behavior of a design entity in terms of queuing models. We still can use the structural or behavioral descriptions to model a queuing system by simply mapping events into nets and entities into parts. However, the delay time is the major difference between a queueing model and a hardware model: delays in a queuing model are random; delays in a hardware model are constant. This requires a special construct to inform the simulator events can be scheduled correctly. In addition, built-in random number generators and statistics-gathering routines are needed to facilitate model building.

Packet description. For any given design entity, there may be many classes of information which augment it as the design proceeds. We should not integrate those classes of information into one amorphous mass, but rather keep them in separate packets. For example, while a structural description is captured through an interactive graphical editor, the symbol names and symbol locations of parts will be stored in a schematic packet. Based on the schematic packet, a schematic can be semi-automatically redrawn. Similarly, a physical packet can be used to store manufacturing codes, placements and other allied physical design information.

D&T Education

CAD/CAM and CIM: A Management Overview, October 16-17, Minneapolis, MN; October 21-22, Houston, TX; November 7-8, Boston, MA; November 18-19, San Francisco, CA; $585. Introduction to the philosophy and benefits of an integrated CAD/CAM process. Contact: Institute for Advanced Technology, 6003 Executive Blvd., Rockville, MD 20852; (800) 638-6590.

III-V Semiconductor Materials and Devices, October 21-22, Palo Alto, CA. Overview of current status of III-V semiconductor technology for scientists and engineers; speakers from industry and universities. Contact: Continuing Education in Engineering, University of California Extension, 2223 Fulton St., Berkeley, CA 94720; (415) 642-4151.

Implementing Local Area Networks, October 29-November 1, Boston, MA; November 19-22, Washington, DC and Los Angeles, CA; December 10-13, Baltimore, MD; $995. Designing Digital Communication Systems, October 22-25, Baltimore, MD; October 29-November 1, Palo Alto, CA; November 19-22, Washington, DC; December 10-13, San Diego, CA; $995. Fiber Optic Communications for Engineers, November 19-22, San Diego, CA; December 3-6, Toronto, Canada; December 10-13, Washington, DC; $995. 16- and 32-bit Microprocessors, October 29-November 1, Washington, DC; December 3-6, Boston, MA; December 10-13, Los Angeles, CA and Ottawa, Canada; $995. Contact: Integrated Computer Systems, 6305 Arizona Pl., PO Box 5055, Los Angeles, CA; outside California, (800) 421-8166; inside California, (800) 352-8251.

Concepts of CAE Technology, November 4-5, December 2-3; $500. The Art of Finite Element Modeling and Analysis, October 21-23, December 16-18; $750. Composite Modeling for Design and Analysis, October 29-November 1; $1000. Contact: Kathy Dockendorf, Seminar Coordinator, Software Products Division, PDA Engineering, 1560 Brookhollow Dr., Santa Ana, CA 92705-5475; (714) 556-2800; Telex 683392.

Implementing Database Systems, November 13-15, Boston, MA; $895. Seminar on how centralized and distributed database management systems work. Contact: Technology Transfer Institute, 741 10th St., Santa Monica, CA 90402; (213) 394-8305.

Design and Drafting of Printed Circuits, November 20-22, Milwaukee, WI. Three-day seminar on printed-circuit board techniques for designers and drafters. Contact: Peter L. Tocups, Program Director, University of Wisconsin at Milwaukee Extension, 929 N. Sixth St., Milwaukee, WI 53202; (414) 224-3952.