Typical Embedded Signal Processing System

- control panel
- ASIC
- microcontroller
- real-time operating system
- controller process
- user interface process
- system bus
- host port
- programmable DSP
- memory interface
- host port
- programmable DSP
- memory interface
- dual-ported memory
- CODEC
- analog interface
- DSP assembly code
Heterogeneity in System-Level Design

System-level modeling:
- imperative
- FSM
- dataflow
- discrete event

Synthesis:
- partitioning
  - compiler
  - software synthesis
  - ASIC synthesis
  - logic synthesis

Detail modeling and simulation:
- execution model
  - execution model
  - ASIC model
  - logic model

Cosimulation:
- symbolic
  - cosimulation
Ptolemy Project

Design Methodologies for Heterogeneous Systems

- Formal models of computation
- Hierarchical compositions of models form complex systems
- Synthesis and partitioning algorithms
- Laboratory to test design methodology is the Ptolemy software environment

Personnel

- Directors: Profs. Edward Lee and David Messerschmitt
- Staff: 4 post-doctoral, 1 software manager, 2 administrative
- Students: 13 graduate and 3 undergraduate
Hierarchical Graphs As Underlying Abstract Syntax

Discrete-Event

Attach semantics

Dataflow

Finite State Machine
Computational Models (Domains) in Ptolemy

Code generation domains:
- C
- 56000
- 96000
- Silage
- VHDL
- VHDLB
- Sproc
- CG

Process networks
- dynamic dataflow
- Boolean dataflow
- synchronous dataflow

Untimed domains:
- SDF
- BDF
- DDF
- PN

Timed domains:
- CP
- DE
- FSM
- IPUS

Integrated proc./understanding of signals

Communicating processes
- discrete-event

Control

Finite state machine
- timed
Heterogeneous System-Level Design in Ptolemy

Mixing Models

- ATM network with three 4x4 switches
- Detailed model of each switch with queueing and routing protocols.
- Dummy traffic (Poisson arrivals) to create congestion.
- Test traffic (video and audio) to measure subjective performance.

Multiple models of computation may be used in the same system. Here, dataflow is used for signal processing, while a timed discrete-event system models a communication network.
# Open Research Issues in System-Level Design

<table>
<thead>
<tr>
<th>Specification</th>
<th>Topic</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Integrated documentation</td>
<td>Parameter relationships</td>
</tr>
<tr>
<td></td>
<td>System optimization</td>
<td>System rearrangement</td>
</tr>
<tr>
<td></td>
<td>Converting graphical specifications into block diagrams</td>
<td>Multidimensional compression systems</td>
</tr>
<tr>
<td></td>
<td>Optimizing algebraic specifications with conversion into block diagrams</td>
<td>Analog filter design</td>
</tr>
<tr>
<td>Simulation</td>
<td>Models of computation</td>
<td>Multidimensional dataflow</td>
</tr>
<tr>
<td></td>
<td>Cosimulation of diverse models of computation</td>
<td>Mixed signal</td>
</tr>
<tr>
<td></td>
<td>Cosimulation of diverse implementation technologies</td>
<td>DSP core</td>
</tr>
<tr>
<td>Synthesis</td>
<td>Partitioning</td>
<td>Hardware/software codesign</td>
</tr>
<tr>
<td></td>
<td>Scheduling</td>
<td>Minimizing data memory in DSP assembly code generation</td>
</tr>
</tbody>
</table>