Embedded System Design

- Increasing system complexity / heterogeneity
- No single simulator can model the entire system
- Trade-off estimation at system level vs. component level
- Exhaustive exploration of design choices is not possible
  - Choice of node architectures for multi-node systems
  - Choice of algorithm implementations
  - Choice of task-to-node mappings

"System design" in MILAN is the process of determining the right architectures, task implementations, mapping and scheduling, so that the desired functionality is provided and performance requirements are satisfied.

Key Elements of the MILAN Approach

- Multi-granularity simulation
  - Use coarse system models that estimate performance using a few key parameters
  - Reduce (large) number of initial design choices to a manageable set
  - Use low-level simulators with detailed performance models to analyze the reduced number of design options
  - Choose one (or more) designs for implementation

- Challenges in implementing a framework
  - Drive all tools from a single system specification
  - Make diverse simulators "talk" to each other (horizontally/vertically)
  - Make the framework modular and extensible

MILAN Architecture

Design Flow Using MILAN

Case Study: 7-node ATR

Scenario
- Collaborative computation and short range communication between small size sensor networks.
- 7 sensor nodes in the ATR application
- Designer wants to model
  - Entire wireless network topology
  - Individual node architectures
  - Application executing on each node
- Multi-level energy estimation
  - Low-level simulation using ns2 for the network and EnergyWatch for a node
  - High-level energy estimation based on a user-specified analytical model

Our Accomplishments

- A modeling and simulation environment for power-aware design of a multi-node sensor network
- Multi-granularity simulation
  - Accurate energy estimates from ns2 and EnergyWatch.
- Simulator integration
  - Results from EnergyWatch simulation are used to automatically configure ns-2 parameters
  - Results from EnergyWatch/ns-2 are used to automatically refine parameters for high level estimator
Simulation Parameters

- Configuring and executing simulators is transparent to the user.
- Following parameters can be varied by designer (through the GUI).

<table>
<thead>
<tr>
<th>Module</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio</td>
<td>transmission/receive/radio power, receive threshold, working frequencies, maximum packet size, etc.</td>
</tr>
<tr>
<td>Propagation model</td>
<td>FreeSpace, TwoRayGround, Shadowing</td>
</tr>
<tr>
<td>Routing protocol</td>
<td>Dynamic Source Routing, AODV, DSDV</td>
</tr>
<tr>
<td>Processor</td>
<td>Frequency, cache configuration, energy per operation</td>
</tr>
<tr>
<td>Memory</td>
<td>size, storage power per byte</td>
</tr>
<tr>
<td>Network</td>
<td>number of nodes, locations</td>
</tr>
<tr>
<td>Application</td>
<td>Number of tasks, implementations, sample size</td>
</tr>
</tbody>
</table>

The Designers’ Perspective

1. Graphically instantiate the system
   - Draw network topology
   - Specify required parameter values (both low and high level)
2. Click “i” – choose high-level estimation
3. View results
4. Click “i” – choose low-level simulation
5. View results

Energy-Efficient Datapath Synthesis for FPGAs*

Traditional Approach
- Choose a specific implementation
- Parameterized model for that implementation
- Low-level simulation and synthetic tools

Our Approach using MILAN
- Algorithm
  - Define high-level performance model for the domain
- Architecture
  - Low-level simulation and synthetic tools
- Automatic coefficient estimation using sample low-level simulation results

Matrix Multiplication on Xilinx Virtex-II

<table>
<thead>
<tr>
<th>Matrix Size</th>
<th>Xilinx</th>
<th>ULP</th>
<th>Linear Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 × 3</td>
<td>0.2</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>6 × 6</td>
<td>4.1</td>
<td>3.6</td>
<td>3.1</td>
</tr>
<tr>
<td>9 × 9</td>
<td>468.5</td>
<td>39.2</td>
<td>32.7</td>
</tr>
<tr>
<td>15 × 15</td>
<td>10063.3</td>
<td>839.77</td>
<td>580.0</td>
</tr>
</tbody>
</table>