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

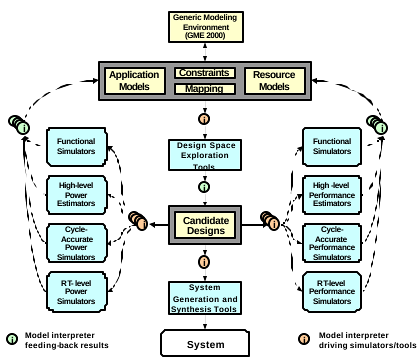
"System design" in MILAN is the process of determining the right architectures, task implementations, mapping and scheduling, so that the desired functionality is provided and performance requirements are satisfied

Key Elements of the MILAN Approach

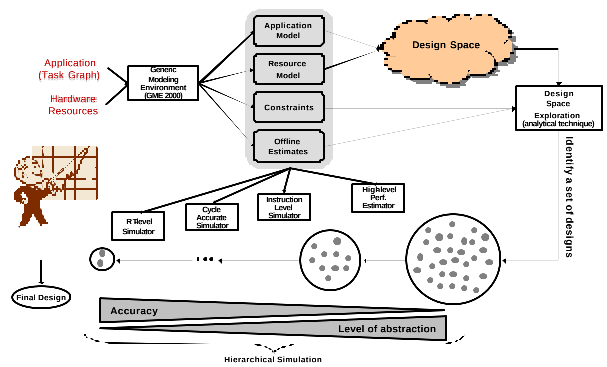
- Multi-granularity simulation
  - Use coarse system models that estimate performance using a few key parameters
  - Reduce (large) number of initial design choices to a manageable set
  - Use low-level simulators with detailed performance models to analyze the reduced number of design options
  - Choose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

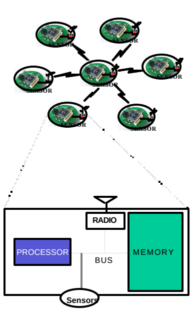
MILAN Architecture



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 Wattch for a node
  - High-level energy estimation based on a user-specified analytical model

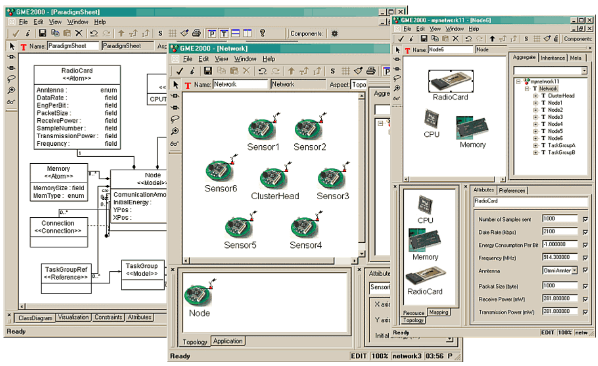
Our Accomplishments

- A modeling and simulation environment for power-aware design of a multi-node sensor network

- Multi-granularity simulation
  - High-level estimates using the system-wide energy model outlined in *"System Level Energy Trade-offs for Collaborative Computation in Wireless Networks"*, M. Singh and V. Prasanna, IMPACCT 2002.
  - Accurate energy estimates from ns-2 and Wattch

- Simulator integration
  - Results from Wattch simulation are used to automatically configure ns-2 parameters
  - Results from Wattch/ns-2 are used to automatically refine parameters for high-level estimator

7-node ATR: Modeling and Simulation



Simulation Parameters

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

Module	Parameters
Radio	transmission/receive/idle power, receive threshold, working frequencies, maximum packet size, etc.
Propagation model	FreeSpace, TwoRayGround, Shadowing
Routing protocol	Dynamic Source Routing, AODV, DSDV
Processor	frequency, cache configuration, energy per operation
Memory	size, storage power per byte
Network	number of nodes, locations
Application	Number of tasks, implementations, sample size

The Designers' Perspective

USER

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

Automatic Parameter Refinement

MILAN

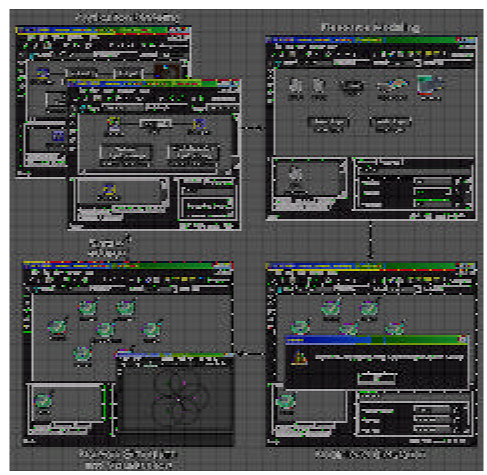
1. Read all system models
2. Generate configuration file for high-level estimation tool
3. Execute tool and update MILAN database

MILAN

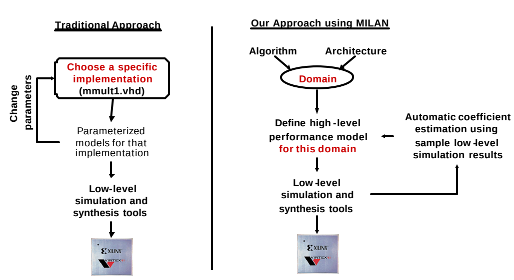
1. Read all system models
2. Generate task executables using SimpleScalar on Linux
3. Configure and execute Wattch on Linux – obtain computation latency and energy consumption for the node
4. Update MILAN database
5. Configure ns-2 using system models, and latency values generated by Watch
6. Execute ns-2 on Linux – obtain network energy consumption; calculate total energy consumed
7. Feed results back to MILAN database

NOTE

- User does not need to know ns-2 or Wattch
- Exchange of simulation results between simulators is transparent to user
- Cross-platform execution using ssh

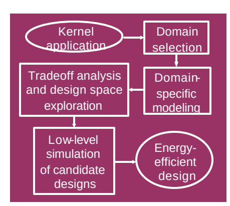


Energy-Efficient Datapath Synthesis for FPGAs*



* S. Choi, R. Scrofano, and V.K. Prasanna, "*Energy-Efficient Design of Kernel Applications for FPGAs through Domain-Specific Modeling*," MAPLD 2002.

Matrix Multiplication on Xilinx Virtex-II



EAT (million nJ × slices × cycles)			
Matrix Size	Xilinx	Uni-proc.	Linear Array
3 × 3	0.2	0.1	0.1
6 × 6	41.2	3.9	3.5
9 × 9	469.5	39.2	32.5
15 × 15	10063.3	839.77	580.0