EE382V: Embedded System Design and Modeling

Lecture 9 – System Modeling & Virtual Prototyping

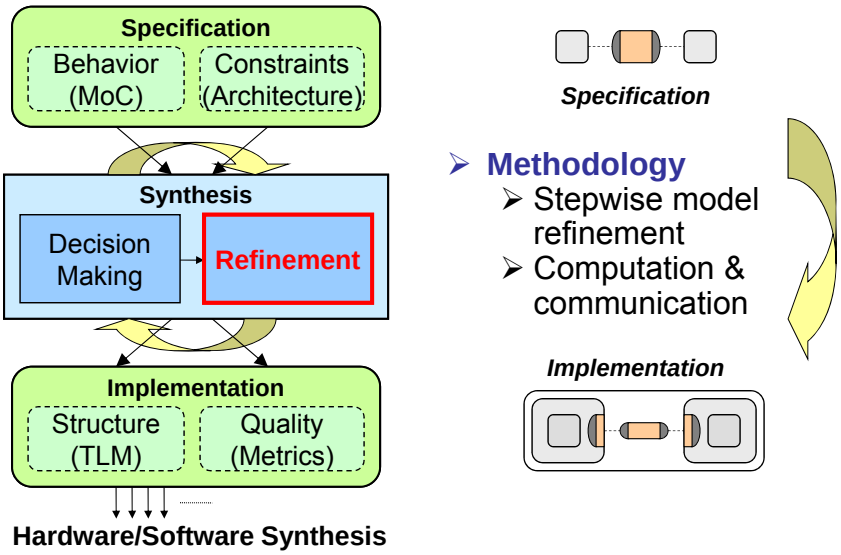
Andreas Gerstlauer
Electrical and Computer Engineering
University of Texas at Austin
gerstl@ece.utexas.edu

Lecture 9: Outline

- **System refinement**
  - Automatic model generation
  - Refinement flow
- **System modeling**
  - Virtual platform prototyping
  - Modeling levels

System-Level Synthesis

- **X-Chart**



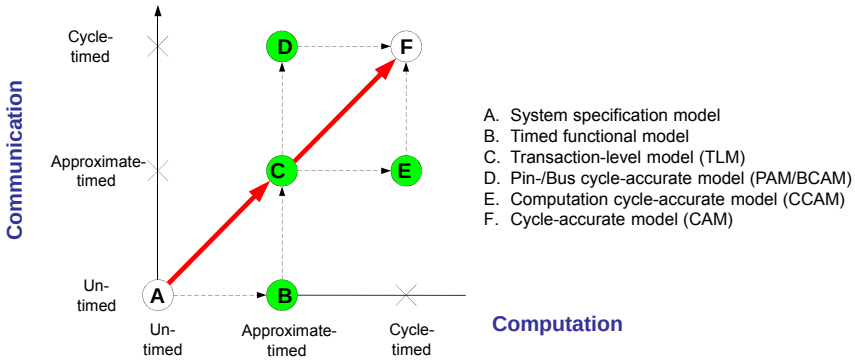
- **Methodology**
  - Stepwise model refinement
  - Computation & communication

Automatic Model Generation

- **Problem: Writing of system models is**
  - Time consuming, error-prone, tedious
- **Solution:**
  - Automatic model generation
- **Refinement-based approach**
  - System designer or tool : makes design decisions
  - Refinement tool : automatically generates the model
- **Benefits**
  - No manual model writing, focus on design decisions
  - Low error rate by automating error-prone tasks
  - Easy change/upgrade for incremental/derivative design
  - No change in basic design methodology
- **Automated path to implementation (synthesis)**
- **Automated design quality evaluation (exploration)**

Recall: System Refinement Flow

- **Abstraction based on level of implementation detail**
  - Computation and communication granularity
- ➢ **Path from model A to model F**

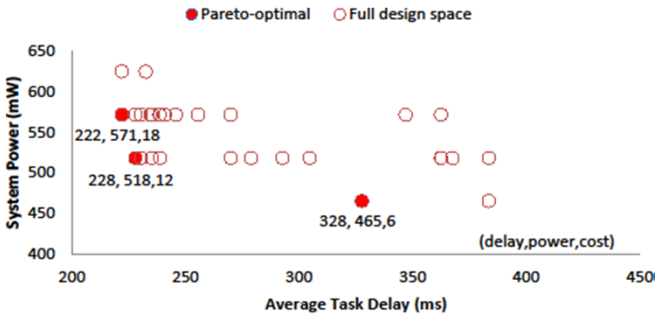


Source: L. Cai, D. Gajski. "Transaction level modeling: An overview", ISSS 2003

- ➢ **Design methodology & flow**
  - ➢ Set of models and transformations between models

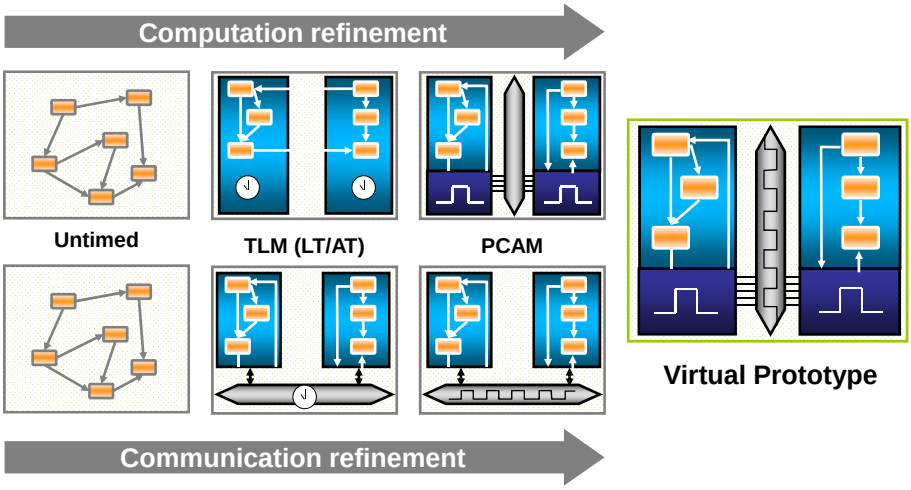
Exhaustive Design Space Exploration

- **Automated design space exploration**
  - Automotive task set [MiBench]
    - Two independent + two dependent tasks
  - PowerPC-based target platform
    - One to four cores, priority-based or round-robin scheduling

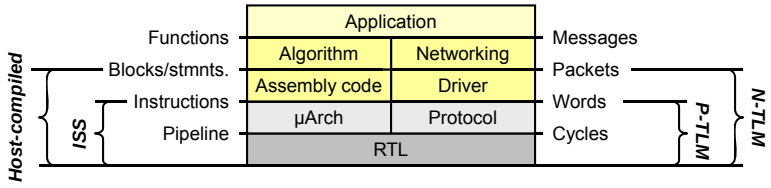


- ➢ **Exploration of 2000 design alternatives**
  - 47h of CPU time using exhaustive search
  - 8h of CPU time using hierarchical 2-step approach
    - First computation to prune design space, then communication to final Pareto front

Virtual Platform Prototyping



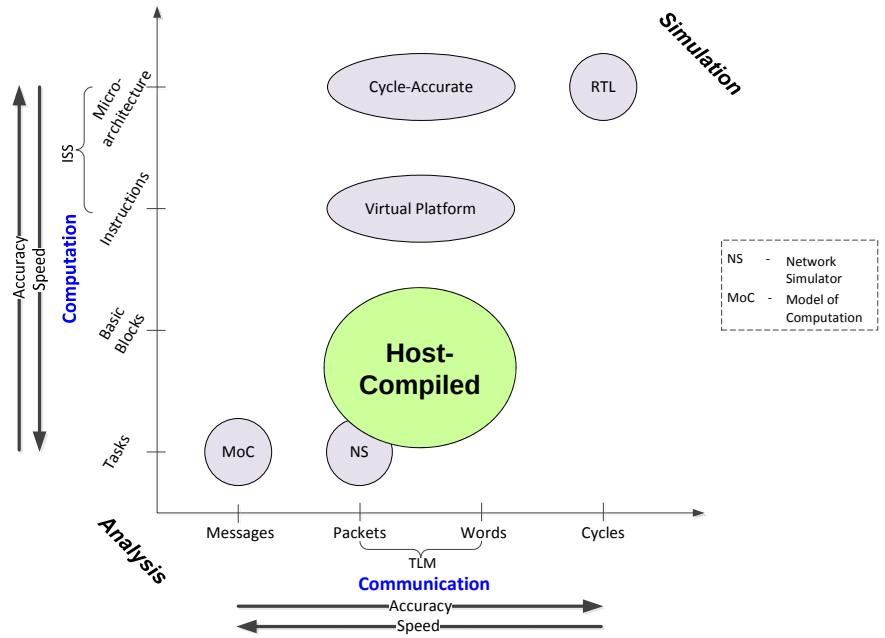
Modeling Levels



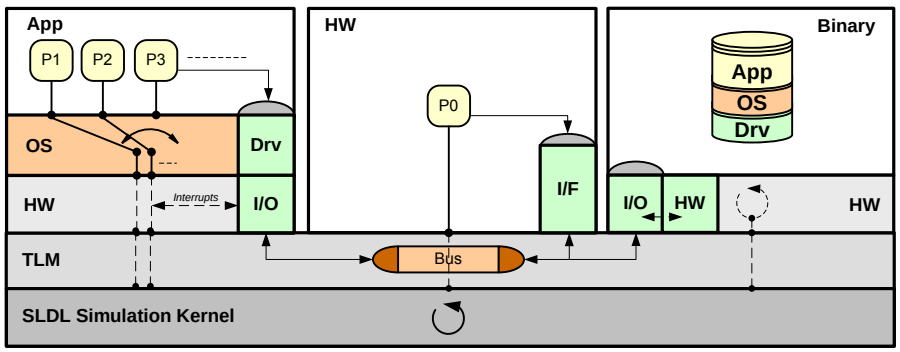
- **Computation**
  - Host-compiled modeling
    - Abstract execution above instructions
    - ➢ Native execution of functionality
    - ➢ Back-annotation
    - ➢ Models of execution environment (OS & processor)
- **Communication**
  - Transaction-level modeling (TLM)
    - Abstract transactions above pins and wires
    - Function calls for data transfer functionality
    - Back-annotation

➢ **Functionality & performance/power/energy/reliability/…**
- ➢ Varying levels of abstraction & granularity
- ➢ Fundamental speed vs. accuracy tradeoffs

System-Level Modeling Space



System Models

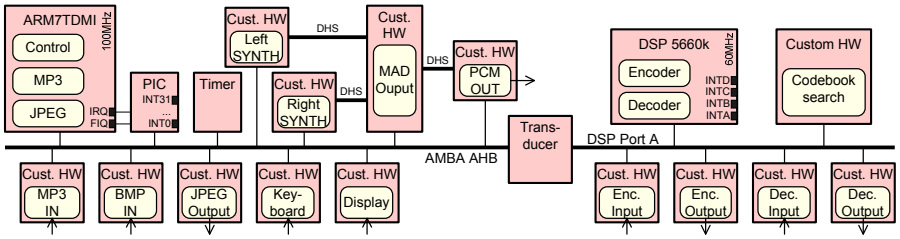


- **Host-compiled model**
  - Native functionality
  - Timing & power back-annotation
  - OS models
  - Processor models
- **SLDL & TLM backplane**
  - Discrete events [SpecC, SystemC]
  - Interconnect
- **Hardware model**
  - Functionality & timing
- **ISS model**
  - Functionality [QEMU, OVP]
  - Cycle-accurate [SimpleScalar]

Recall: SCE Cellphone Example

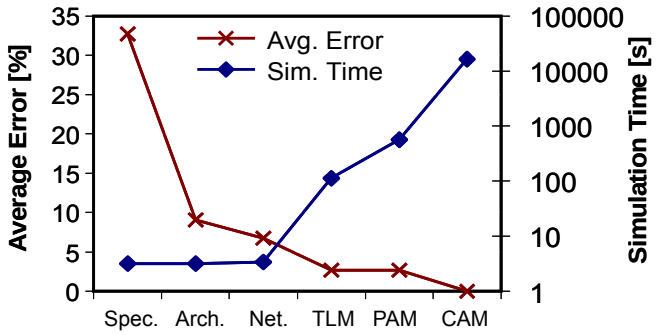
- **2 Subsystems**
  - ARM7TDMI
    - MP3 Decoding
    - Jpeg Encoding
  - Motorola DSP 56600k
    - GSM Transcoding

- **4 Accelerator HW blocks**
- **10 I/O HW blocks**
- **5 Busses**
  - AMBA AHB
  - DSP Port A bus
  - 3 Custom busses



Recall: Cellphone Results

- **Experimental setup**
  - 1.5 second MP3
  - 640x480 picture
  - 1.5 speech GSM
  - 3s / 300M ARM cycles / 180M DSP cycles
- **Simulation speed**
  - 300 Mcycles/s
- **Accuracy**
  - <3% error



- **Transaction-level modeling (TLM) of communication**
- **Host-compiled software, OS and processor modeling**

Lecture 9: Summary

- **System modeling and refinement**
  - From specification to implementation
  - Basis for automated synthesis and exploration
  - ➢ Automatic model generation
  - ➢ Layer-based successive refinement
- **Various levels of abstraction**
  - Computation and communication
  - Speed and accuracy tradeoffs
  - ➢ Quest for fast and accurate modeling techniques