ENST Sophia - LabSoC

Design Space Exploration
Refinement
Security

Context

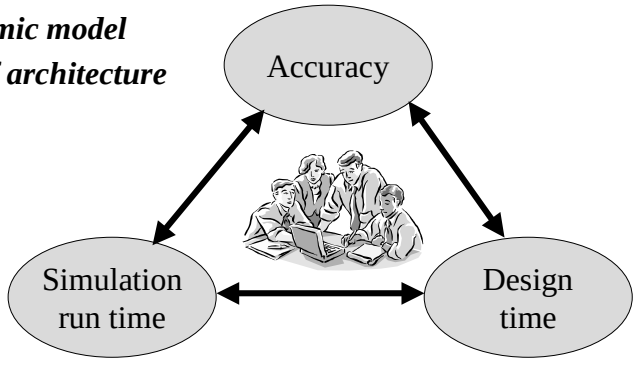
- **Why?**
  - ***Industrial and public labs in Sophia-Antipolis***
  - ***Potential research partnerships***
  - ***Education needs (Eurecom new curriculum / "Real Time and Embedded Systems" track)***
  - ***SoC and Security are key components of Information Technologies***
- **Who?**
  - ***Sophie Coudert (formal methods)***
  - ***Ludovic Apvrille (real time systems)***
  - ***Renaud Pacalet (integrated systems)***

LabSoC Research Themes (I)

- **Integrated systems complexity increases**
- **Design productivity => design reuse (IP blocks)**
- **Functional verification is a bottleneck**
- **Face the challenge**
  - ***New design space exploration methods***
  - ***Help the refinement from abstract to concrete models***
- **How?**
  - ***Mix formal methods and simulation***
  - ***Separate function and performance***
  - ***Use stochastic bounded performance models when needed (data dependant control)***

Design Space Exploration

- **Traditional flow:**
  - ***Algorithmic model***
  - ***Model of architecture***
  - ***Mapping***
  - ***Simulate***
  - ***Analyze***



DSE Outputs

- **HW/SW partitioning**
- **Specification of HW and SW modules**
- **CPU choice, clock frequency, caches, …**
- **Interconnect architecture**
- **Memory architecture**
- **Shared resource dimensioning (memory, bus)**
- **Potential dead locks identified and fixed**
- **Real time constraints checked**
- ***Analog/Digital partitioning***
- ***Power management, …***

DSE Today: The Models

- **Hardware models**
  - ***Algorithmic (bit accurate or not)***
  - ***Untimed Transaction Level Model***
  - ***Timed TLM***
  - ***Bus Cycle Accurate***
  - ***Register Transfer Level***
- **Software models**
  - ***Algorithmic (bit accurate or not)***
  - ***Source code***
  - ***Binary***
- **CPU models**
  - ***Instruction Set Simulator***
  - ***Cycle true ISS***
  - ***RTL***

Full function plus more or less accurate performance

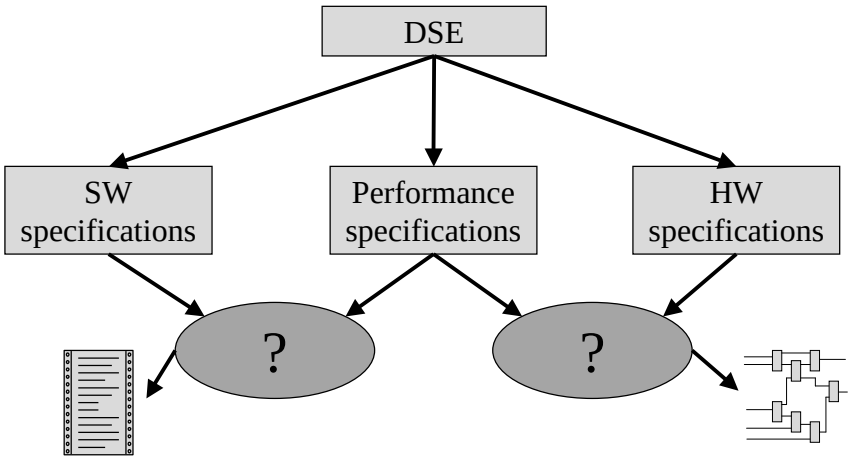
DSE Key Modeling Idea: Abstract Function

- **Model of application = network of tasks**
  - ***No processing***
  - ***As few control as possible***
  - ***I/O bandwidth and processing time modeled (statistical bounded models when needed)***
  - ***No difference between hardware and software tasks (partitioning not yet defined)***
- **Model of architecture made of a small library of generic components (CPU, memory, bus, …)**
- **Mapping of application on architecture**
  - ***Including arbitration schemes for multitask nodes***

DSE Challenges

- **Define suitable languages and formalisms**
- **Simulate several orders of magnitude faster % TLM**
- **Apply formal techniques (avoid state explosion)**
- **Mix simulation and formal techniques**
  - ***Languages, formalisms, user interfaces unified***
  - ***Pick the best of both***
- **Keep it simple (low cost design time) but realistic**
- **Keep it abstract (formal techniques and simulation speed) but accurate enough**
- **Keep it alive all the design flow long**
  - ***Golden performance model***
  - ***Fast back-annotation when accurate performance available***
  - ***Make it useful to quickly find the "less expensive modification"***

Refinement



Refinement Problems

- **Critical transformations, error prone**
  - ***TLM to BCA: introduce an exact timing***
  - ***BCA to RTL: identify, allocate, schedule resources***
- **Different languages**
  - ***Matlab, C++, SystemC, VHDL, Verilog***
  - ***Different semantics, even for a single language (simulation vs synthesis)***
- **Function and performance intricate**
- **No back-annotation to higher abstraction levels**
- **Optimized for target technology => not reusable**
- **Difficult to maintain**

Refinement Key Ideas: Formal Techniques

- **Formal bounded equivalence abstract / concrete models**
  - ***Unbounded liveness is often useless***
  - ***Bounded liveness = safety***
  - ***Concrete models $\subseteq$ abstract models***
- **DSE: abstract, performance-only, models available**
  - ***Easier to manipulate formally***
  - ***Should allow larger scale performance formal verification***
  - ***Separate functional and performance specifications available***
    - **Model checking for performance verification**
    - **Theorem proving for functional verification**

Refinement Key Ideas: Formal Techniques

- **Formal help to abstraction needed**
  - ***Black boxes IPs, COTS***
  - ***Reuse existing designs in this new flow***
  - ***Seamless transition from traditional approaches***
- **Integrated verification framework needed**
  - ***Simulation and formal verification interact (Assertion Based Verification)***
  - ***Maintain verification needs database***

LabSoC Research Themes (II)

- **Pervasive computing needs security**
- **e-Business needs security**
- **Strong algorithms and protocols are available**
- **The computing hardware is a weak element of the security chain**
- **Security hardware and hardware security**
  - ***Side channels attacks countermeasures (DPA, DEMA)***
  - ***On-the-fly deciphering of soft IPs***

Security Projects

- **Design of an experimental IC (Q2 2004) including DPA resistant cells**
  - ***Data independent power and EM***
  - ***Asynchronous delay insensitive computation***
  - ***Dual rail data representation***
- **Design of a publicly available smartcard-like crypto-processor (open hardware and software)**
  - ***Stimulate research in this area on both aspects***
  - ***Allow low cost redesigns for hardware research***
  - ***More attacks => improve countermeasures***

Security Projects

- **Optimize DPA / DEMA resistant structures**
  - ***Minimize the silicon area overhead***
  - ***Exploit the asynchronous paradigm***
    - **Variable speed / robustness depending on the sensitivity**
    - **Self power management**
  - ***Avoid SI problems with dual rail - event signaling***
  - ***Study manufacturing dispersion impact***
- **Investigate soft IP deciphering or/and integrity checking**
  - ***Silicon area / power consumption***
  - ***CPU performance vs algorithms robustness***

Partnerships with Eurecom labs

- **Eurecom labs provide excellent DSE-refinement test cases**
  - ***Mobile communications platform***
  - ***Multimedia applications***
  - ***Network and security applications***
- **Applications and architectures totally open**
- **Design flows available and customizable**
- **Designers available**

Partnerships with Eurecom labs

- **Mobile communications platform**
  - ***Modeling and design methods, hardware / software portioning***
  - ***Define a "universal" communication model between processing units***
  - ***Help the platform evolution***
    - **Smooth software <-> hardware shifts**
    - **Seamless functionality extensions**
  - ***UMTS TDD – 802.11 prototype project***
- **Mobile communications platform provides a simulation environment**
  - ***Enhance the simulation environment with different abstraction levels***
  - ***Hardware acceleration***