


Platform-based Design and the First Generation Dilemma

By Jiang Xu and Wayne Wolf
Dept. of ELE Princeton University
April, 2002



Outline

- ◆ Background
- ◆ Typical system-level design methodology
- ◆ Platform-based design methodology
- ◆ First generation dilemma
- ◆ Conclusions

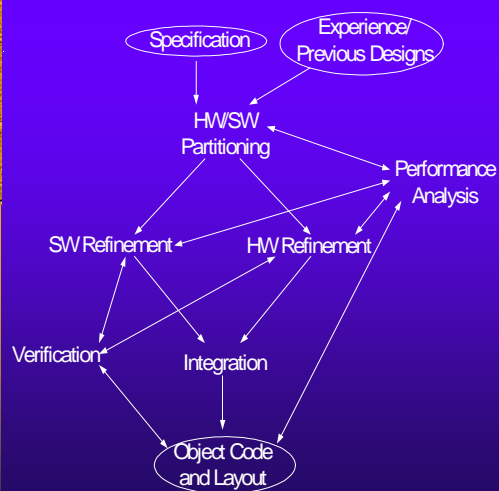


Background

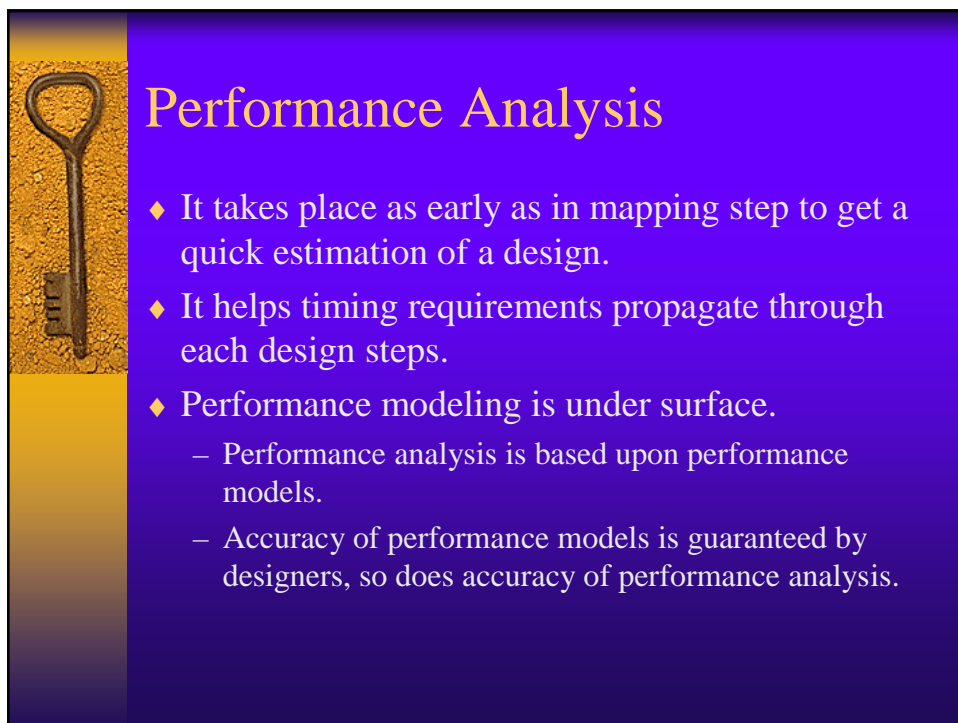
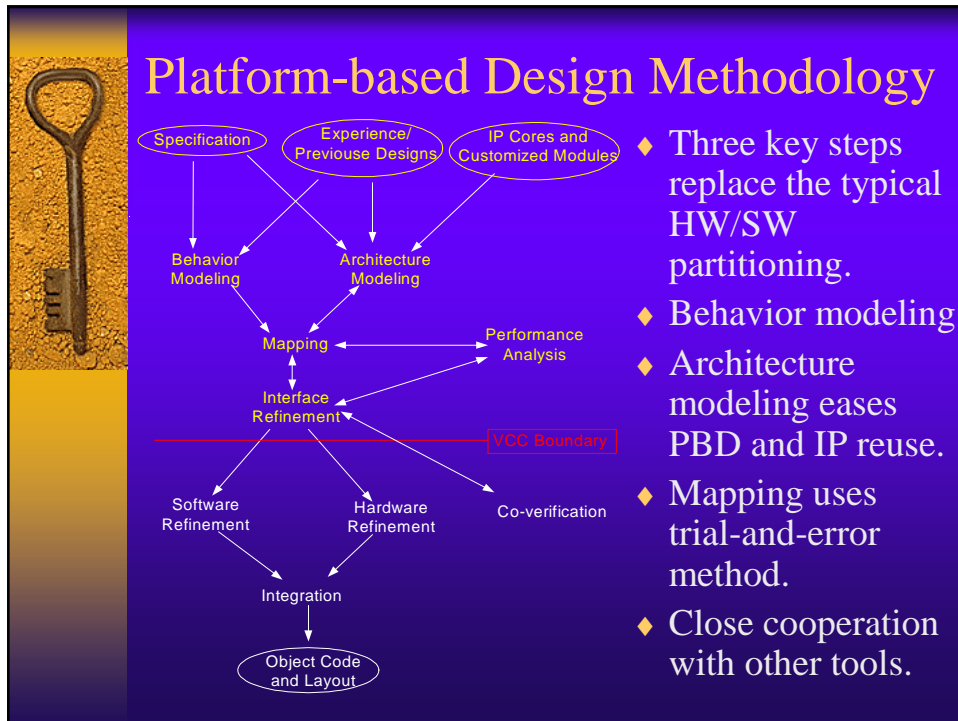
- ◆ Growing gap between design complexity and productivity.
- ◆ System-level design tool is a solution.
- ◆ Some system-level design tools.
 - Cadence Virtual Component Co-design(VCC).
 - CoWare N2C.
 - Synopsys CoCentric System Studio.
 - Innoveda Visual Elite.
 - Elanix SystemView.



Typical System-level Design Methodology



- ◆ Besides specification, previous design experience is an important input for system-level design.
- ◆ HW/SW partitioning is converted in function mapping.
- ◆ Performance analysis plays a more important role in PBD.
 - The accuracy of early performance analysis significantly affects following design steps.





Behavior Modeling

	Black Box	White Box	Clear Box
Language	C++, SPW, SDL, OMI	WhiteBox C	STD, Textual SDL
Simulated	Yes	Yes	Yes
Analyzed	No	Yes	Yes
Synthesizing	No	No	Yes

- ◆ Behavior model is used to capture specification details and develop HW and SW.
- ◆ Different languages are used.
 - Choosing programming style is an old and unsolved problem.
- ◆ White-Box C is preferable to others.



Outline

- ◆ Background
- ◆ Typical system-level design methodology
- ◆ Platform-based design methodology
- ◆ **First generation dilemma**
- ◆ Conclusions



Another View of the PBD Methodology

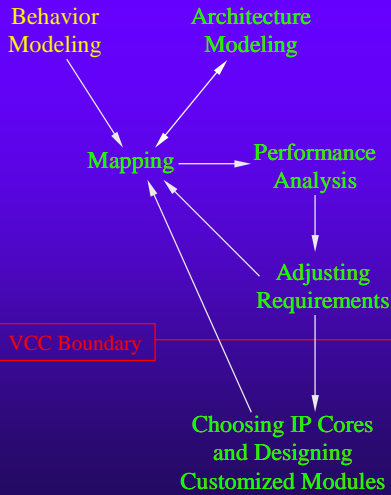
- ◆ Two steps for architecture: choosing and mapping.
- ◆ Decisions of choosing platform and IP cores and designing customized modules are based upon performance analysis results.
- ◆ Performance analysis can not be made without IP cores or customized modules.



First Generation Dilemma

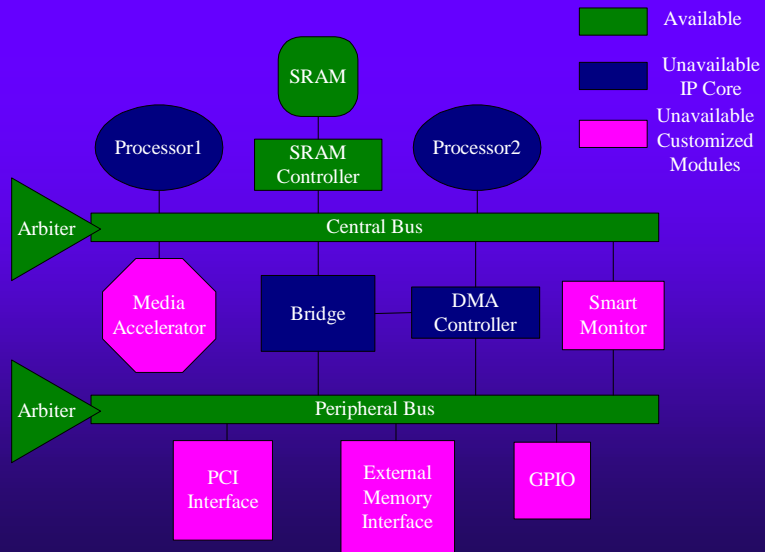
- ◆ PBD assumes designers have enough IP cores and customized modules to build a platform.
- ◆ In many first generation designs, only a few IP cores and customized modules are in libraries.
- ◆ In the original PBD methodology, we can not:
 - Get performance results without enough IP cores and customized modules.
 - Choose IP cores and design customized modules without performance results.

A Way Out of the Dilemma



- ◆ First decide an architecture, and assign estimated requirements to unavailable modules.
- ◆ Adjust the requirements using performance analysis in a trial-and-error fashion.
- ◆ Based upon the requirements purchase IP cores and design customized modules.
- ◆ May need several iterations to reach a final design.
- ◆ It is very helpful, if designers can get performance models of IP cores before buy them.

Example -- A Multimedia Embedded Chip





Conclusions

- ◆ First generation dilemma
 - PBD is not purely choosing-and-mapping, it still includes architecture design and module design.
 - Easily accessed performance models will be very helpful.
- ◆ Behavior modeling is a critical step in PBD, and modeling language is still an important issue.
- ◆ PBD needs a systematic method to guarantee performance analysis accuracy.