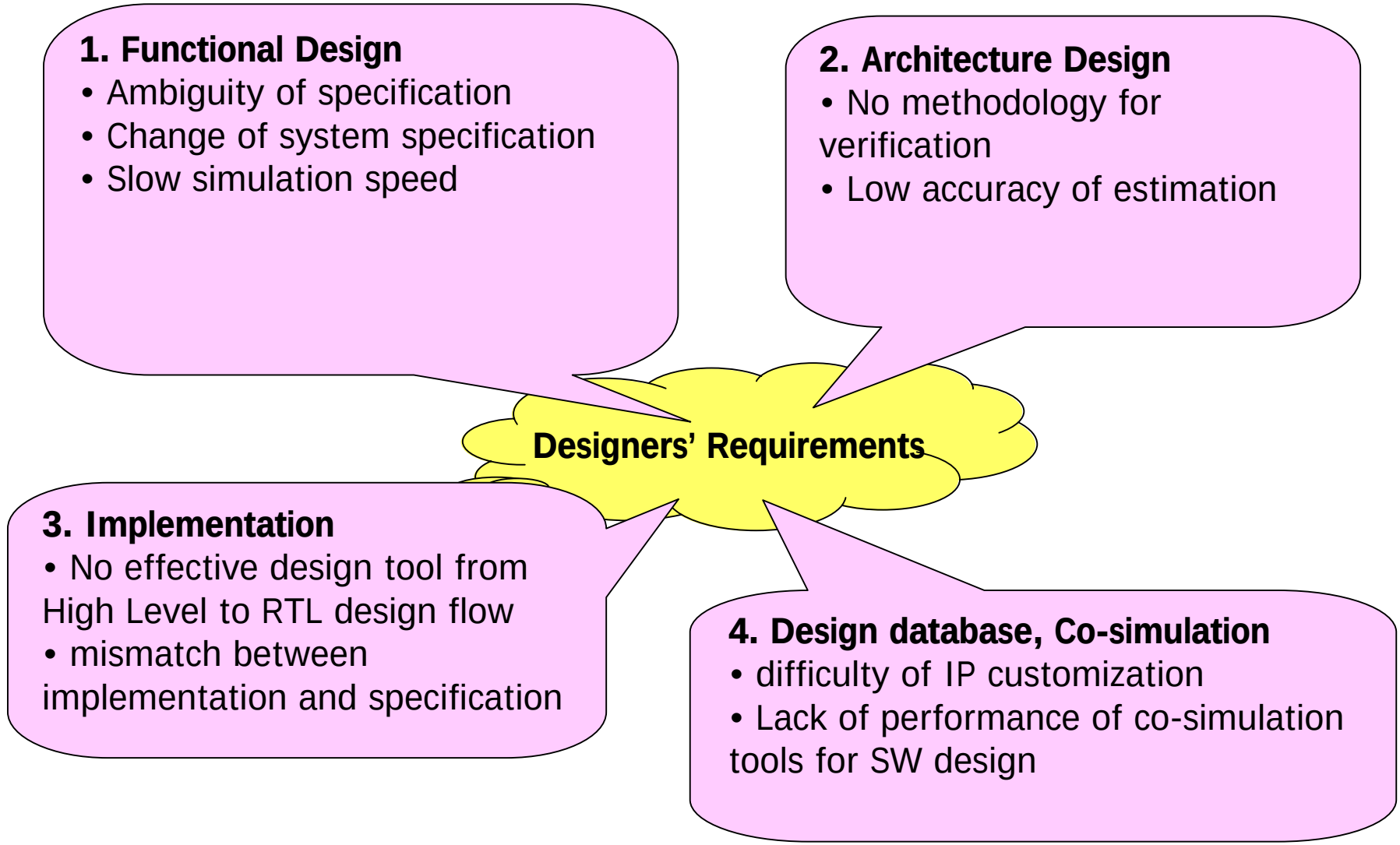
System Level Design Technologies and System Level Design Languages

SLD Study Group

EDA-TC, JEITA

http://eda.ics.es.osaka-u.ac.jp/jeita/eda/english/project/sld/index.html

Problems to Be Solved



Solutions

1. Functional Design
For handling the ambiguity of specification ...

- Methodology to define functional specification
- Executable SLD language

2. Architecture Design
For earlier estimation of cost, performance, etc. ...

- Fast exploration method of finding optimum architecture
- Estimation model and design database

Designers' Requirements

3. Implementation
For easy link to implementation phase ...

- Interface synthesis of HW/SW
- Equivalency check

4. Design database, Co-simulation
For easiness of IP use, high speed co-simulation ...

Standardized I/F of IP, IP synthesis, Highly abstracted HW model

System Level Design Technologies

- Research Target
  - “EDA Technology Roadmap Toward 2002”
    EIAJ/EDA-TC/EDA Vision Study Group （1998）
  - Universities/Research Centers
    - POLIS — http://www-cad.eecs.berkeley.edu/Respep/Research/hsc/abstract.html
      “Hardware-Software Co-Design of Embedded Systems, The POLIS Approach” Kluwer Academic Publishers(1997)
    - OCAPI-xl — http://www.imec.be/ocapi
    - lpChinook — http://www.cs.washington.edu/research/chinook/index.html
  - Vendors
    - CoCentric — http://www.synopsys.com/
    - N2C — http://www.coware.com/
    - VCC — http://www.cadence.com/

Roadmap of EDA Technologies

In the prediction, almost all technologies are in the trial phase in 2000 !!
How in the real world?

Items	1997	1998	1999	20
[System Level Design]				
Standardization of system model				
Standardization of system level design language(SLDL)				
Simulation of SLD Language				
Performance estimation in system level design (application soft, compiler, performance estimation of HW, optimization)				
High speed prototyping				
Formal verification (Specification - System)				
HW/SW partition				
System level library (IP core, middle ware, etc.)				
Decision support of system level test strategy				
Tradeoff of analog/digital				
[Architecture Design]				
Standardization of architecture model				
Standardization of architecture description language				
Estimation in architecture level (area, delay, power,..)				
Decision of RTL constraint (area, timing, power, budgeting of floor plan)				
Formal verification (System - Architecture)				
Validation/Simulation				
Architecture synthesis				
Co-synthesis				
Decision support of architecture level test strategy (test method/area/length of test)				
DFT in architecture level				

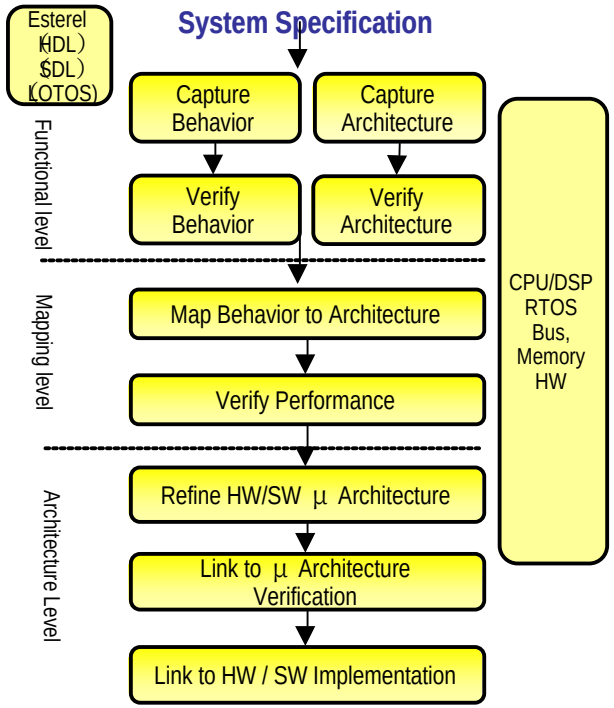
:trial use :preceding use :practical use

Cited from EIAJ/EDA Technology Roadmap Toward 2002

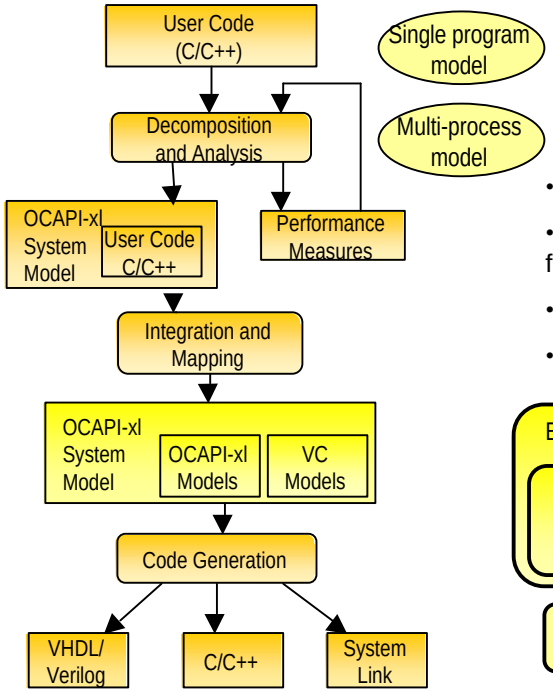
Design Tools (1) by Universities and Research Centers

POLIS (UCB, U.S.A.)

- Co-design Environment for embedded system based on the CPU and DSP
- Supporting functional design to prototyping
- Architecture exploration by mapping behavior to architecture
- Accurate and high-level performance estimation using a fuzzy instruction set



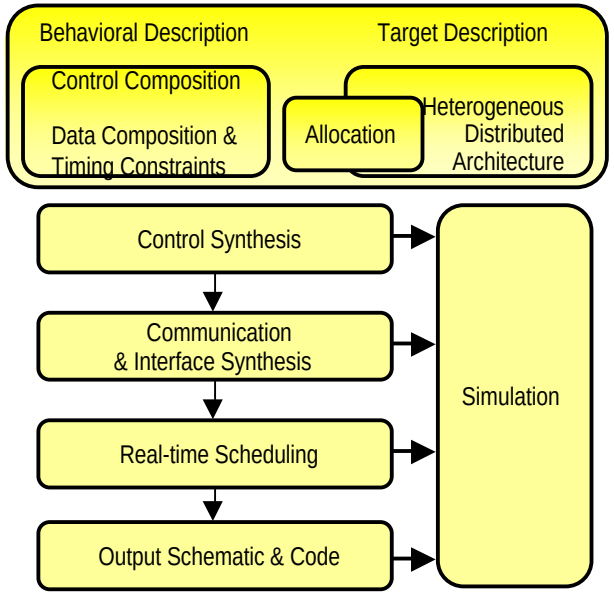
OCAPI-xl (IMEC, Belgium)



- C++ model which can describe HW and SW uniformly
- Optimal block partition using a performance analysis
- Automatic generation of VHDL / Verilog-HDL / C

IpChinook (U of Washington, U.S.A.)

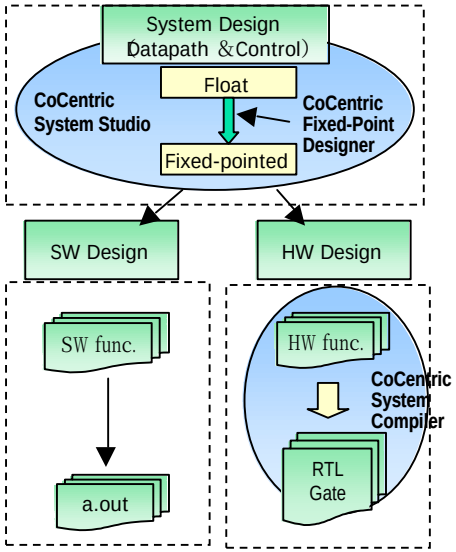
- Based on IP-based/Reused-Design methodology
- Automatic generation of control and I/F between functional blocks
- Lots of communication protocol libraries
- Support of Java, Pia, C



Design Tools (2)

by Vendors

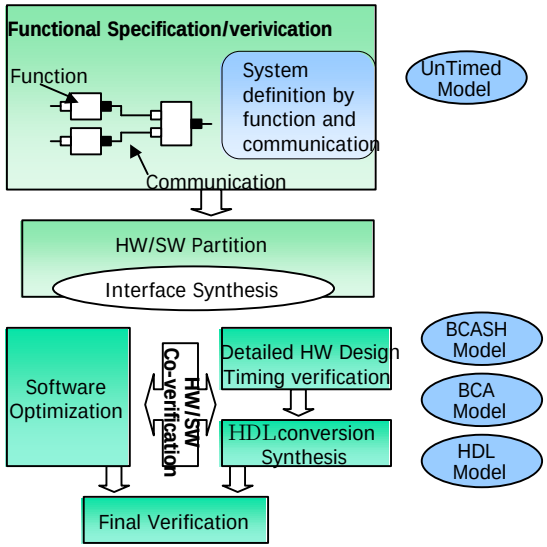
CoCentric (Synopsys)



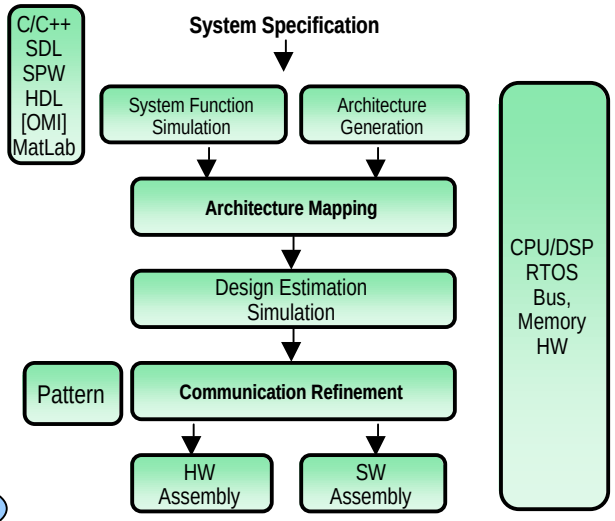
- Support HW side and co-verification of SOC Design
- Conversion from float to fix-pointed
- HW Synthesis from behavior (Verilog,VHDL)
- Supporting SystemC

N2C (CoWare)

- Supporting functional level to implementation
- System Definition in Untimed, BCASH, BCA abstraction levels by original C-based language (CoWareC)
- Communication refinement
- Interface synthesis between HW and SW
- Supporting SystemC (in future)



VCC(Cadence)



- Architecture exploration by mapping function to architecture
- Performance estimation using a fuzzy instruction set
- Supporting IP reuse
- Communication synthesis
- Interface to implementation

Summary of EDA Trend

- May be solved in near future…?
  - Function Verification
  - Architecture Mapping
  - Performance Estimation
  - Interface Synthesis
  - HW/SW Co-Verification

- Unsolved Problems
  - Function Partition
  - High Speed Estimation
  - Estimation of Area, Cost, and Power
  - Standardized Description of system specification, constraints

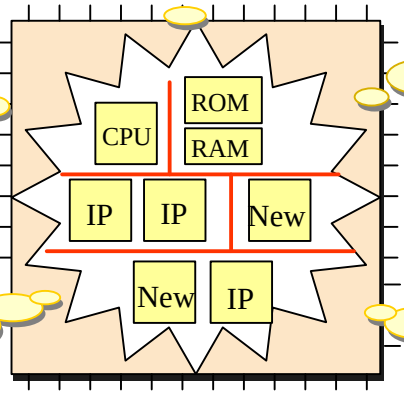
  etc.. . .

System LSI

Multi-function/
High performance
Requirement of system level optimization

Large/Complex
Heterogeneous components

High IP Utilization

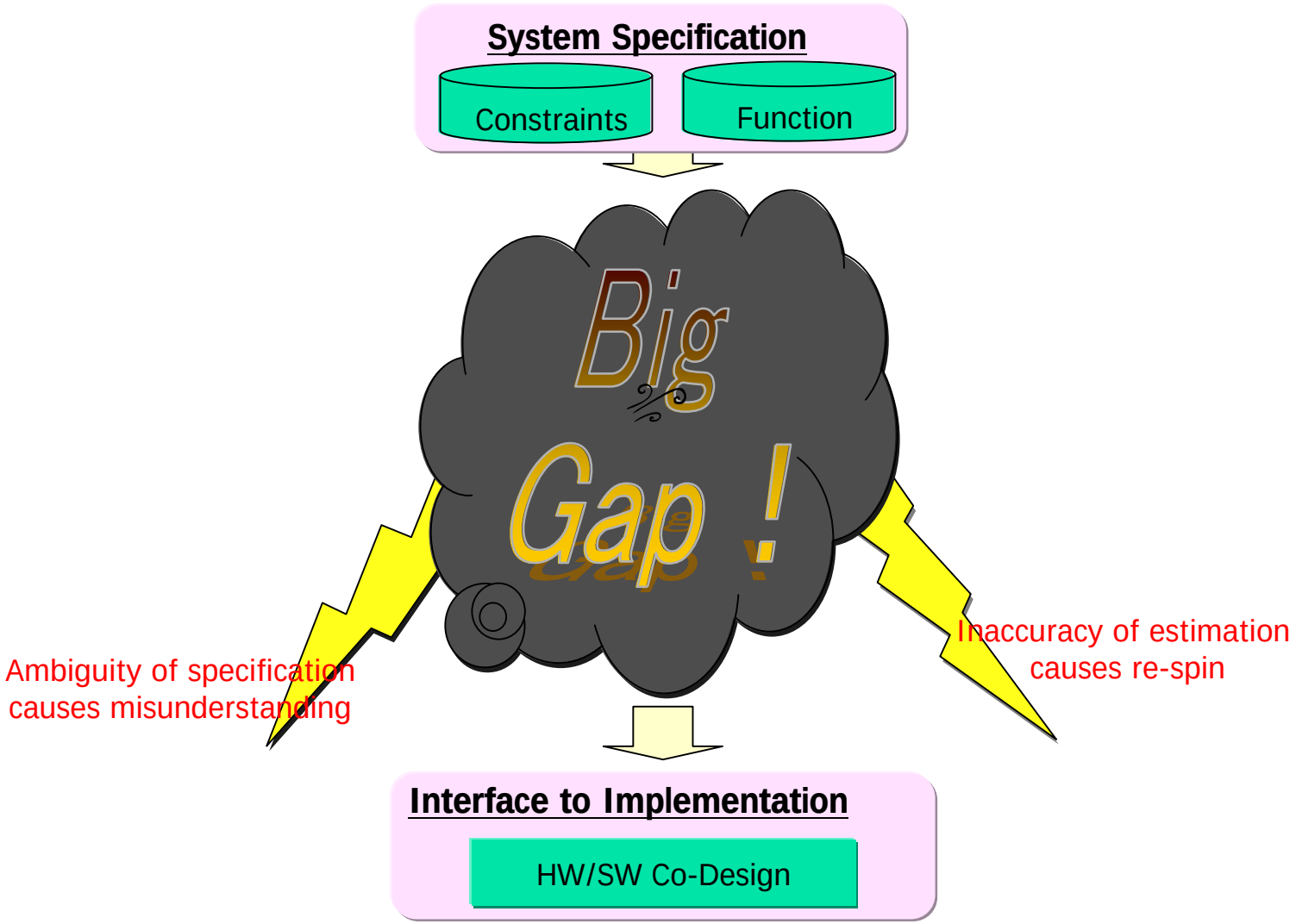


System LSI

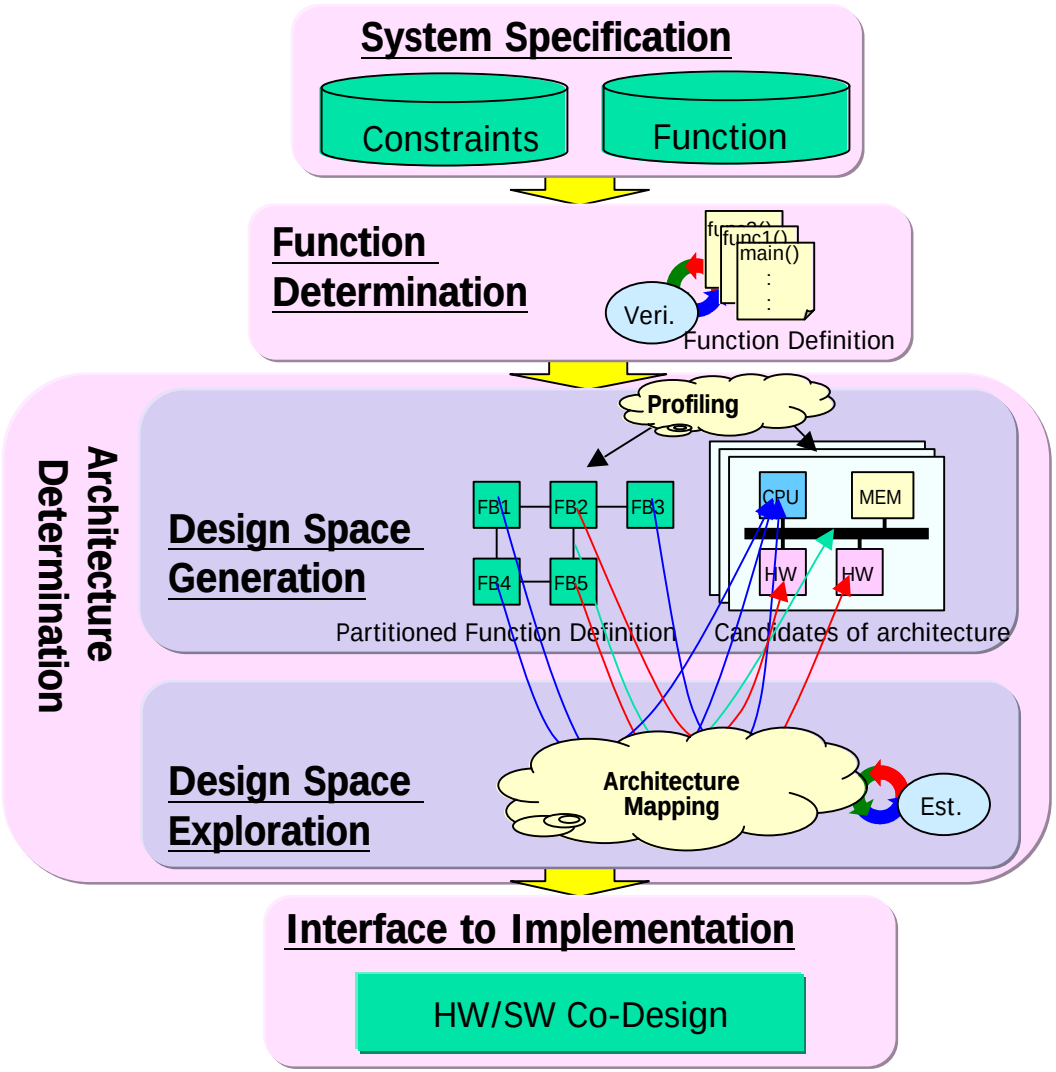
Multi-product, Small-lot Production

Short TAT

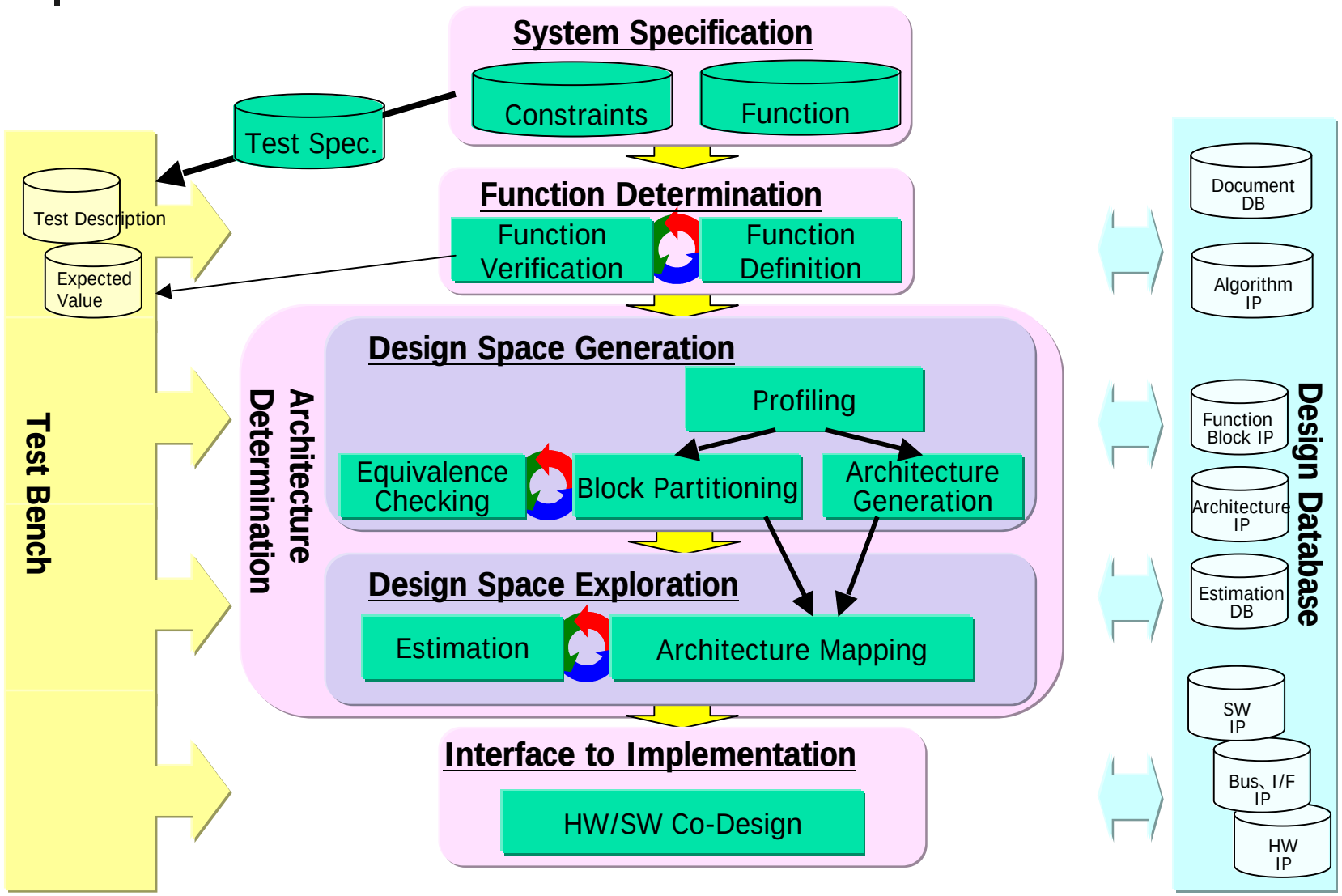
Conventional Design Flow



Proposed Design Flow

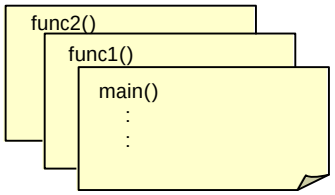


Proposed Design Flow



Design Space Generation

Block partitioning and architecture generation from function model



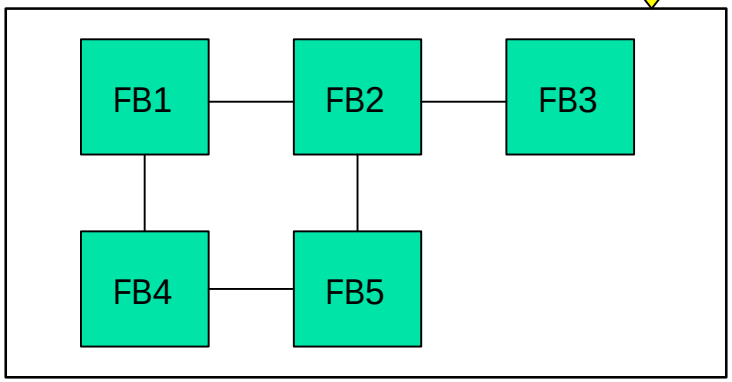
Function Model

Executable model which represents the function of system independent of HW/SW implementation (created in the function determination phase)

Design Space Generation

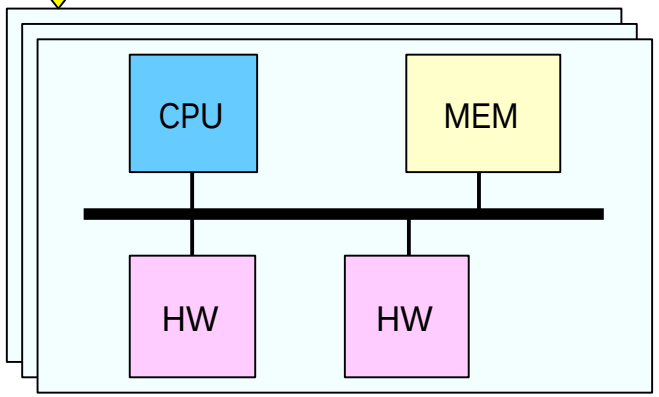
Profiling

Block Partitioning

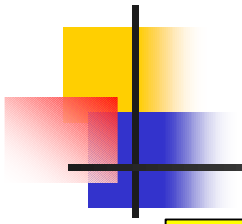


Process Model
represents system by functional blocks and communications between them

Architecture Generation



Architecture Model
represents system by components



Profiling

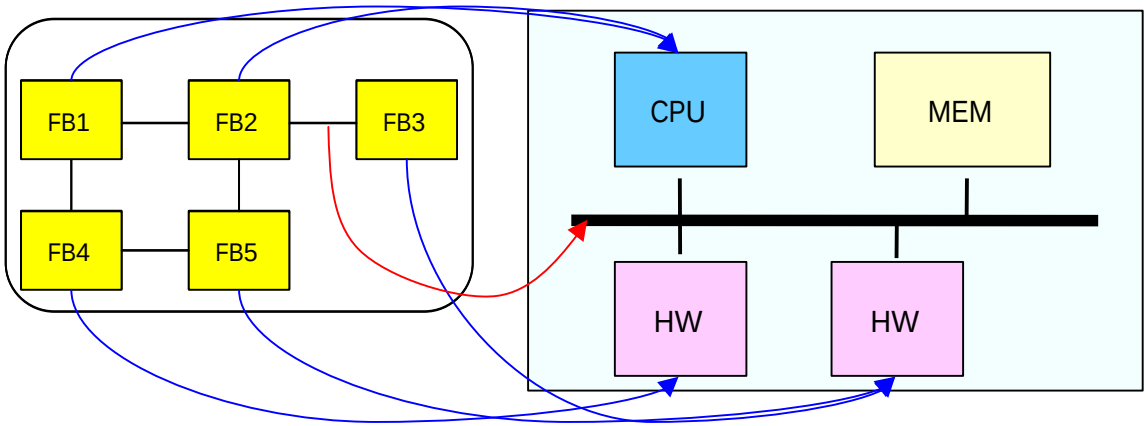
Analyze function models statically/dynamically and extract information for block partitioning and architecture generation

Class	Information	Usage (in design space generation)
Number of Variable Access	•Number of read/write •Access position •Read-write distance	•Choice of algorithm •Decision of block candidates •Selection of memory/register
Amount of Data Transfer	•Number of (bits)*(number of exec.) of read/write •Amount of parameters	•Decision of partitioning units •Decision of HW/SW partitioning •Selection of comm. protocol
Amount of Calculation	•Execution count of lines •Execution count of blocks •Execution count of functions	•Decision of partitioning units •Performance estimation on HW/SW
Statistical Analysis	•Type and count of arithmetic/control instructions	•Choice of processor •Refinement of instruction set

Design Space Exploration

Mapping each functional block to a component and select the optimum architecture which satisfies constraints

Architecture Mapping



Transaction Model

represents only the processing cost for estimation

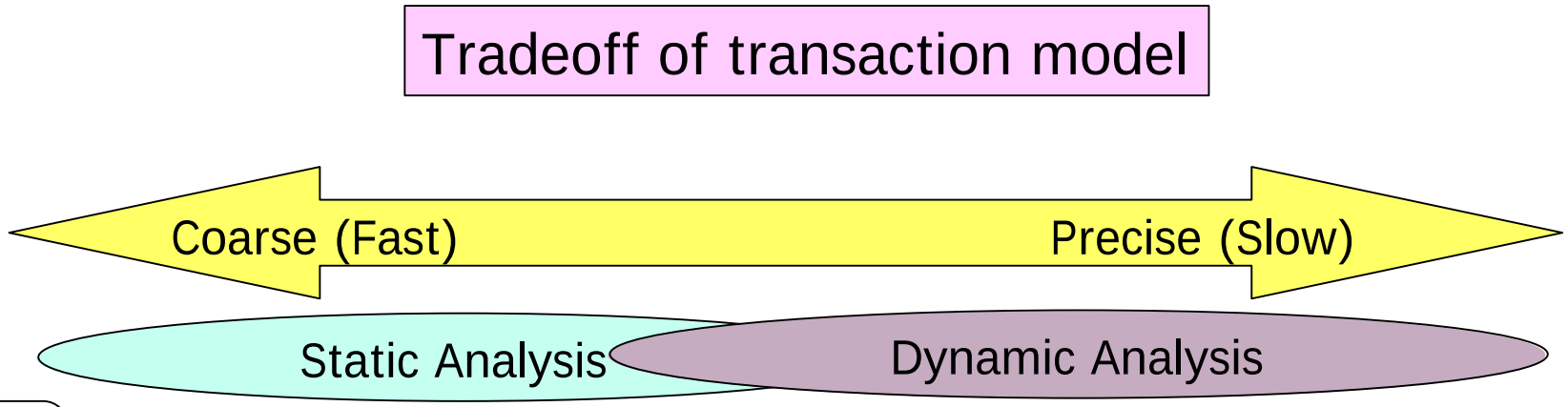
Architecture Model

(Example) Processing time = (process cost) / (process capability per units)

From transaction model

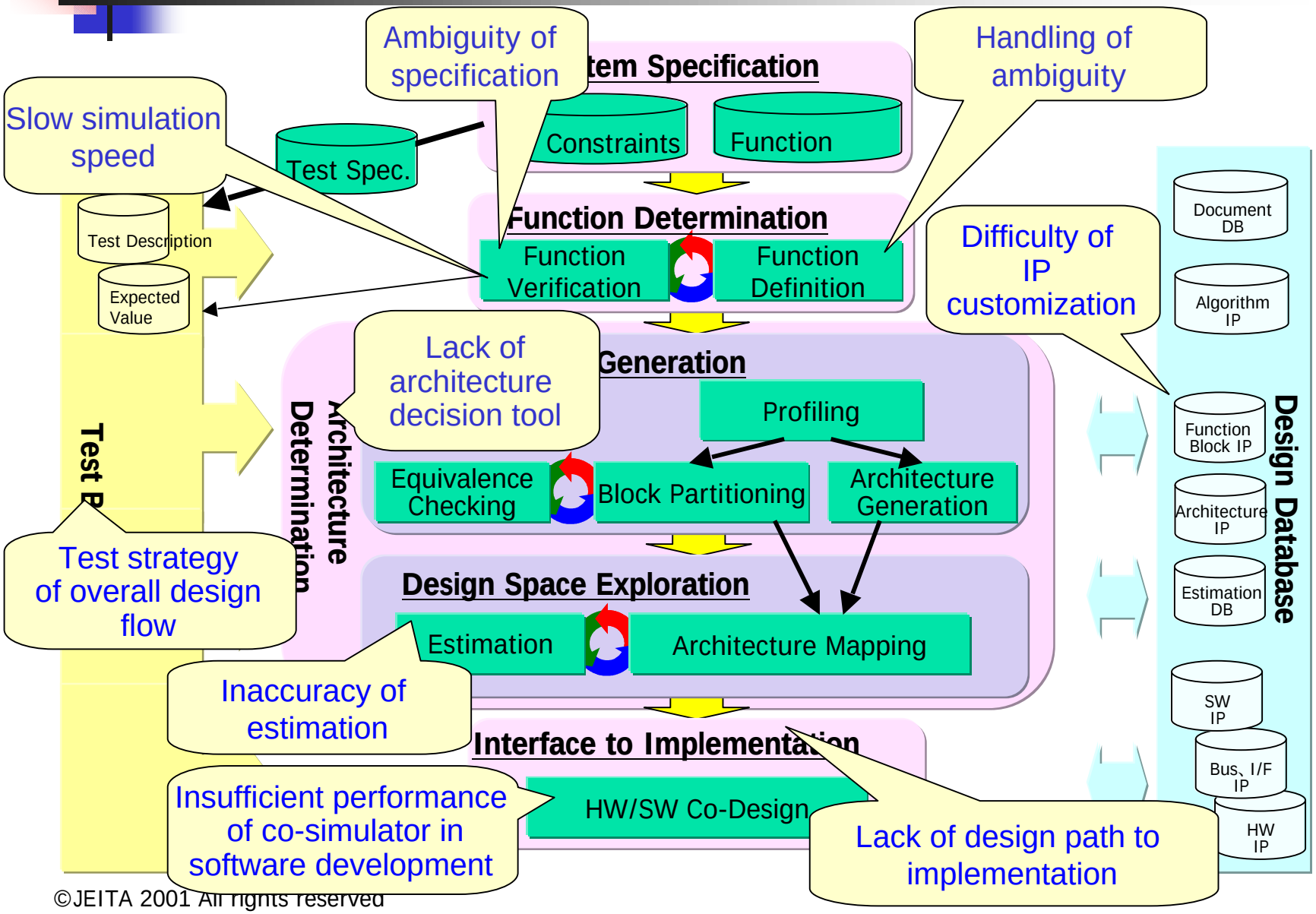
From architecture model

Transaction Model and Accuracy



	Static Analysis		Dynamic Analysis
Method	・Simple summation of average properties (design data)	・Considering algorithm (branch, loop) ・Instruction set	・Dynamic simulation (using constraints)
Advantages and Disadvantages	・High speed/Low accuracy ・Need DB of past data ・Impossible to critical analysis	・Middle speed/Middle accuracy ・Estimation of average property/range	・Low speed/High accuracy ・Need detailed analysis of implementation

Solution to Problems



Technology Map of Tools

Only the verification is supported by all tools

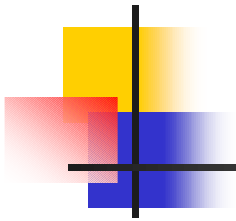
No tool support

Design Phase		Tech	Tool1	Tool2	Tool3	Tool4	Tool5	Tool6
System Specification		Function	×	×	×	×	×	×
		Constraints	×	×	×	×	×	×
Function Determination		Function Definitio	○	△	○	△	△	△
		Function Verificatio	◎	◎	◎	◎	◎	◎
Architecture Determination	Design Space Generation	Profiling	×	×	×	×	◎	△
		Block Partitioning	△	△	△	△	△	△
		Architecture Generati	△	△	○	○	△	△
		Equivalence Check	◎	◎	◎	◎	◎	◎
	Design Space Exploration	Architecture Mapping	△	○	○	○	○	△
		Performance Estimati	×	○	○	○	○	×
Interface to Implementation		Interface Synthesi	×	◎	◎	◎	◎	◎
		HW/SW Co-D	◎	◎	×	○	◎	◎
Test Bench		Test Ben erat	△	△	△	△	△	△

◎ : automated by the tool
○ : tool has supporting function
△ : tool does not have supporting function
× : not covered by the tool

Some tools support this phase but speed/accuracy of the estimation is insufficient

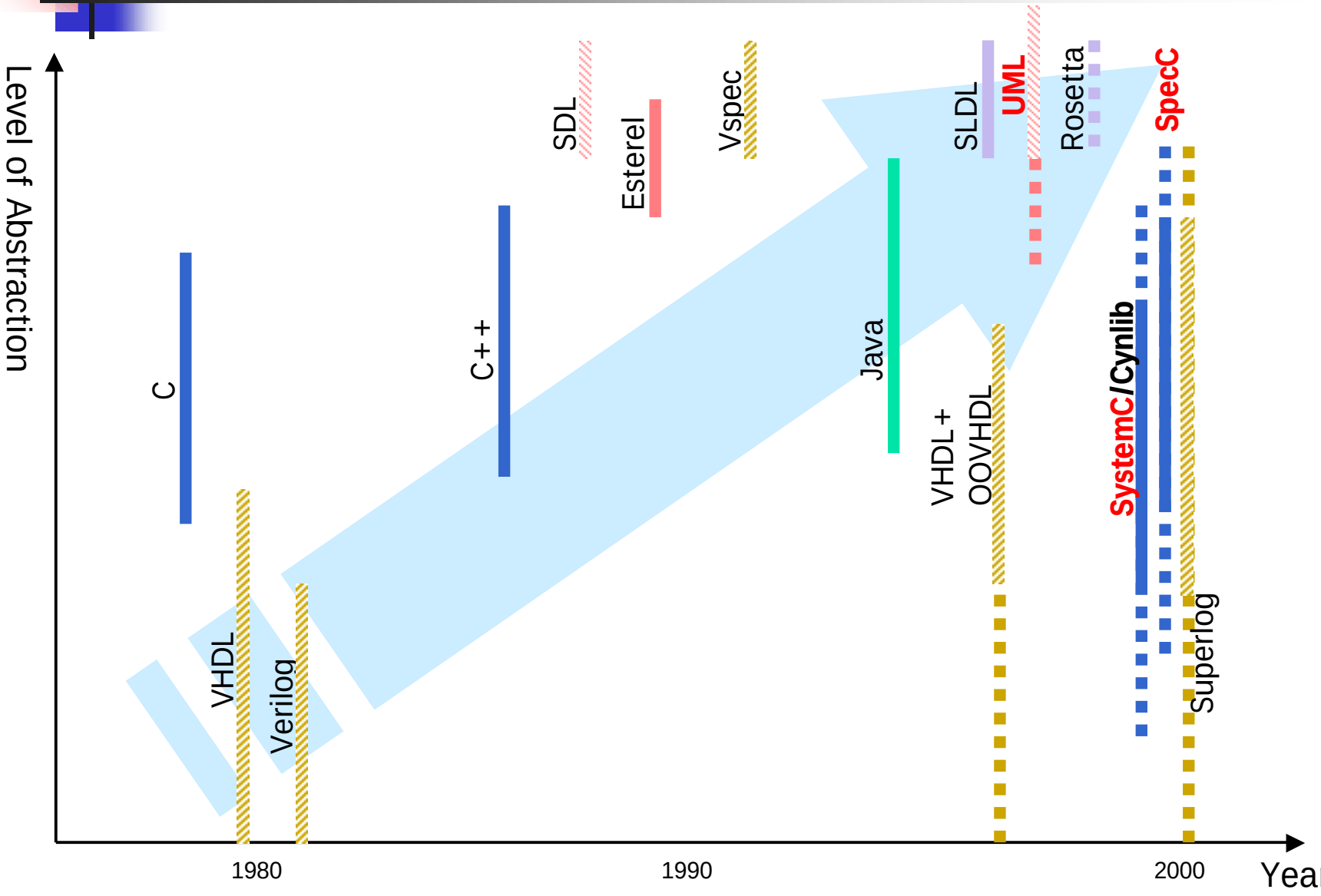
No tool supports the test bench generation

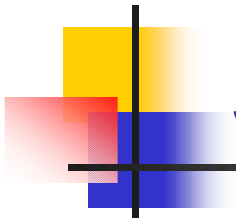


Investigation of System Level Design Languages

- Background
  - Important role for system level design flow
  - Standardization activities (OSCI, STOC, Accellera)
- Objectives
  - Understand standardization activities (SystemC, SpecC, UML, etc.)
  - Evaluation of the proposed design flow
  - Feedback to standardization groups

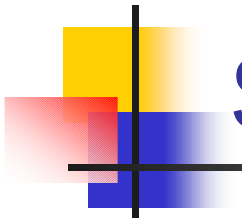
Design Languages





SystemC

- Background
  - Proposed based on the technologies of Synopsys Inc., CoWare Inc., Frontier Design Inc. (Sept 1999)
- Features
  - **Representation from system level to HW/SW by adding dedicated class libraries, and without enhancing C/C++ grammar**
    - Availability of standard C/C++ development environment
    - Less readability of description
  - **Developed from HW design using C/C++ to system level**
    - Easy to move from HDL environment
- Trends
  - Open SystemC Initiative (OSCI) promotes standardization
    - More than 70 of EDA vendors, IP vendors, semiconductor companies join to OSCI (Apr 2000)

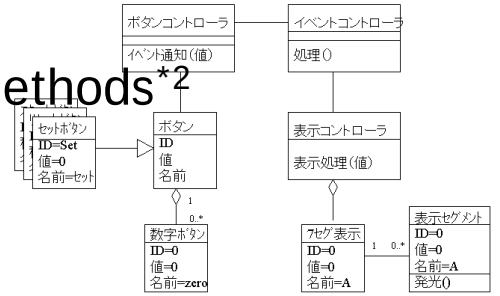


SpecC

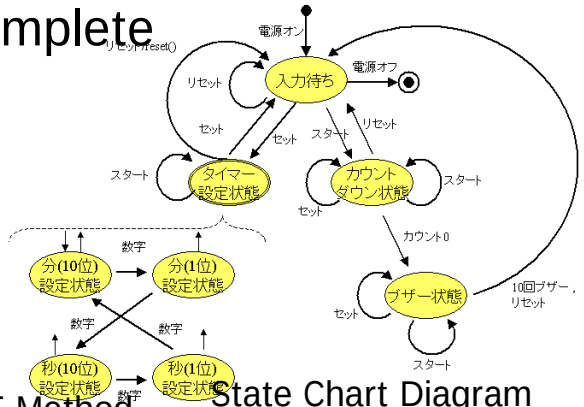
- Background
  - Developed by Prof. Gajski in UCI (1997)
- Features
  - **Unified description of HW/SW by enhancing ANSI-C grammar**
    - High readability
    - Needs special tools and environment
  - **Clear design methodology which covers over system modeling to architecture determination**
- Trends
  - SpecC Technology Open Consortium (STOC) promotes the standardization
  - System houses, embedded software tool vendors have strong interest

UML

- Background
  - Developed as modeling language for complicated large system
  - Started the standardization since 1996 by OMG *1
  - Unification of existing object-oriented modeling methods *2
- Features
  - **Visual language with high modeling ability**
    - Visual and multi-viewed system modeling
      use case, class, component, layout, state chart, sequence, activity, collaboration
    - Scheme for describing constraints
    - Methodology of applying HW design is incomplete
- Trends
  - Wide use in business software
  - Started to use in embedded software
    ⇒ trial phase for HW design



Class Diagram

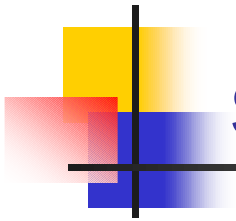


State Chart Diagram

[1] Object Modeling Group [*2] Booch Method, OMT Method, OOSE Method

Evaluation Results

Language / item	SystemC (v2.0)	SpecC (v1.0)	UML (v1.4)
System Specification			
Function	×	×	◎ visual description
Constraints	× not supported	× not supported	○ constraint language
Function Determination			
Function Definition	○ less readability	◎clear methodology	○ only for SW
Function Verification	◎ available C++ env.	○	△ only state chart
Architecture Determination			
Profiling	○	○ tool environment	× not supported
Block Partition	○	◎unified HW/SW desc.	×
Equivalence Check	○	○	×
Architecture Generation	△only structure	△only structure	×
Architecture Mapping	△transaction model NG	△transaction model NG	×
Estimation	△insufficient methodology	△insufficient methodology	×
Interface to Implementation			
HW/SW Co-Design	◎ support until RTL	△ planned to support Accellera's RTL model	× not supported
Test Bench Generation	○	○	△ only specification
IP Design	△	△	×



Summary of System Level Design Languages

- SystemC
  - Representation from system level to HW/SW by adding dedicated class libraries and without enhancing C/C++ grammar
  - Need to clarify the methodology
  - Strong support for implementation, but need to consider the architecture generation and the estimation
- SpecC
  - Unified description of HW/SW by enhancing ANSI-C grammar
  - A key to prevalence is to arrange special design environment to implementation
  - Methodology for system modeling exists, but weak for the architecture generation and later stages
- UML
  - Large ability of function/constraint description by visual modeling
  - Need to consider design phases after architecture design