New Strategies
for
System Level Design

Daniel Gajski

Center for Embedded Computer Systems (CECS)

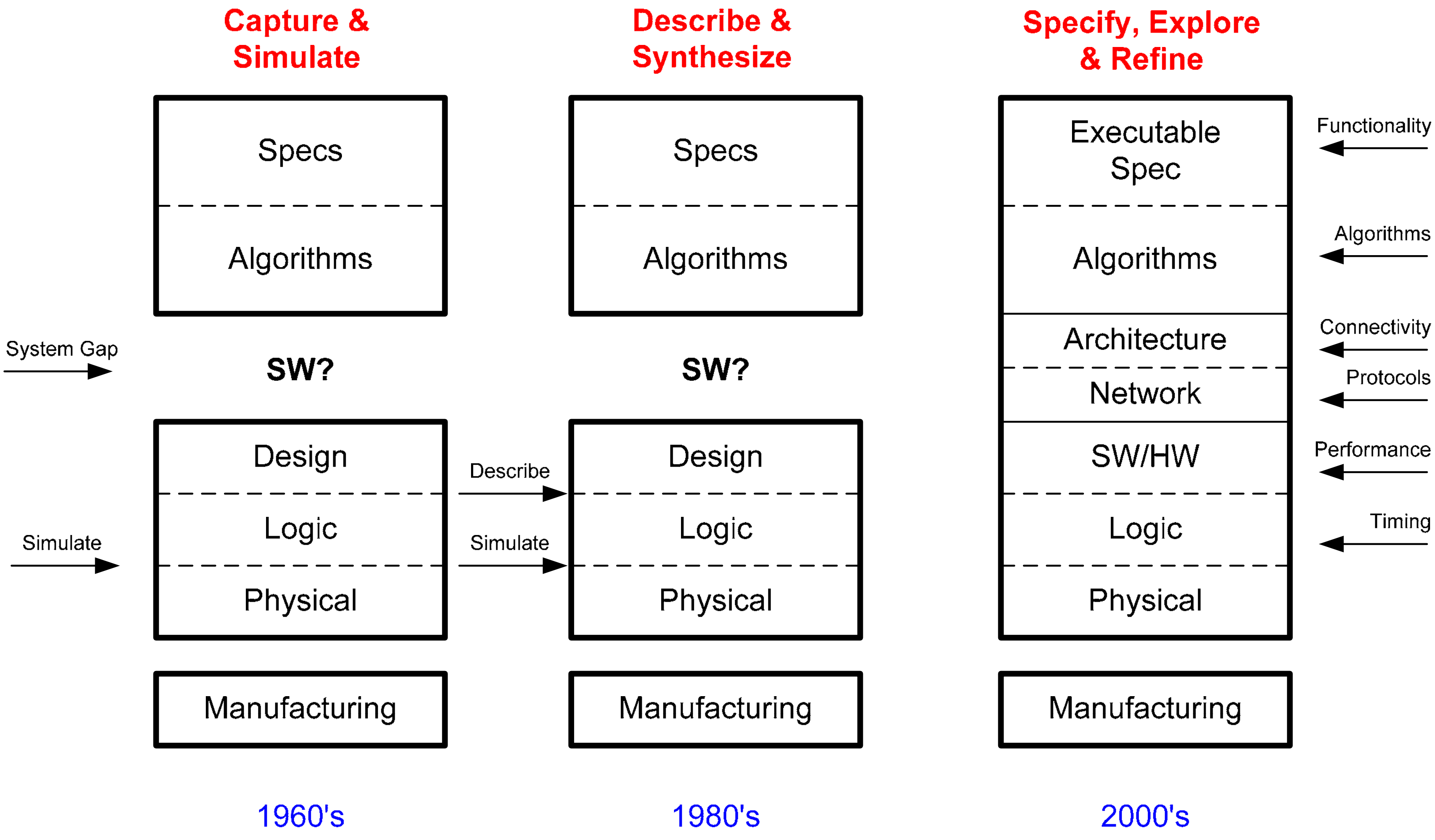
University of California, Irvine

gajski@uci.edu

Overview

- Introduction
- Issues
- Models
- Platforms
- Tools
- Benefits
- Conclusion

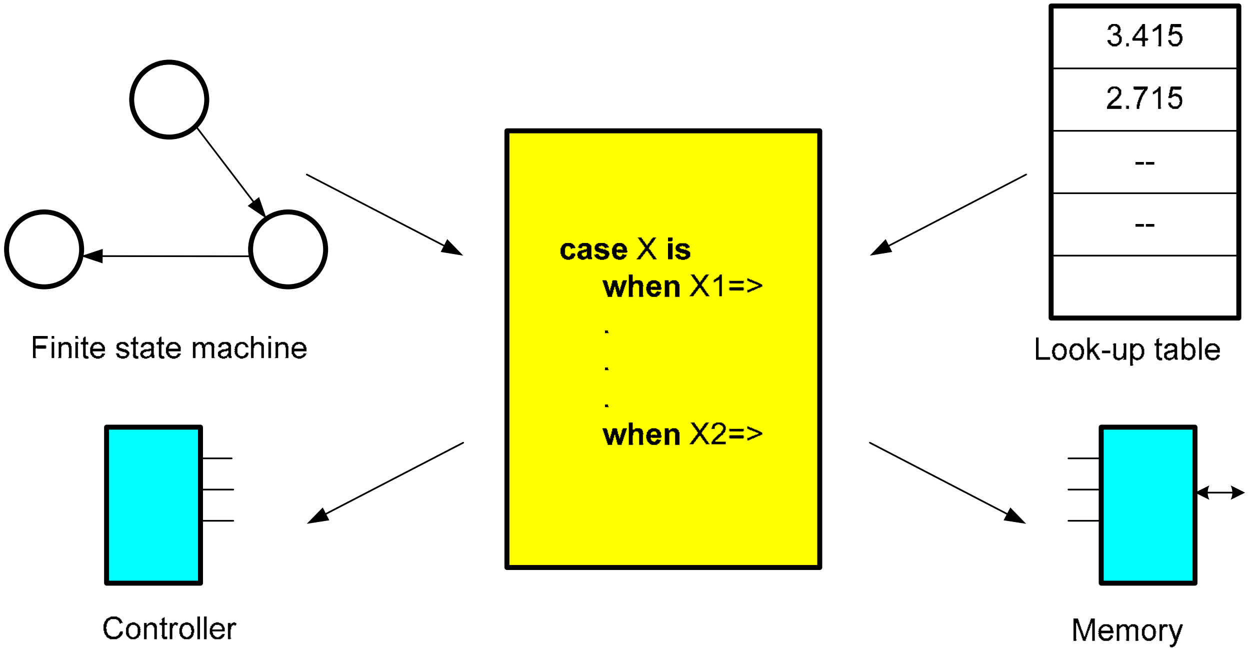
Closing the System Gap



Real gap: behavior and structure (semantics and syntax)

Simulation Based Methodology

Ambiguous semantics of hardware/system level languages



```
case X is
    when X1=>
    .
    .
    .
    when X2=>
```

Simuletable but not synthesizable or verifiable

In Search of a Solution

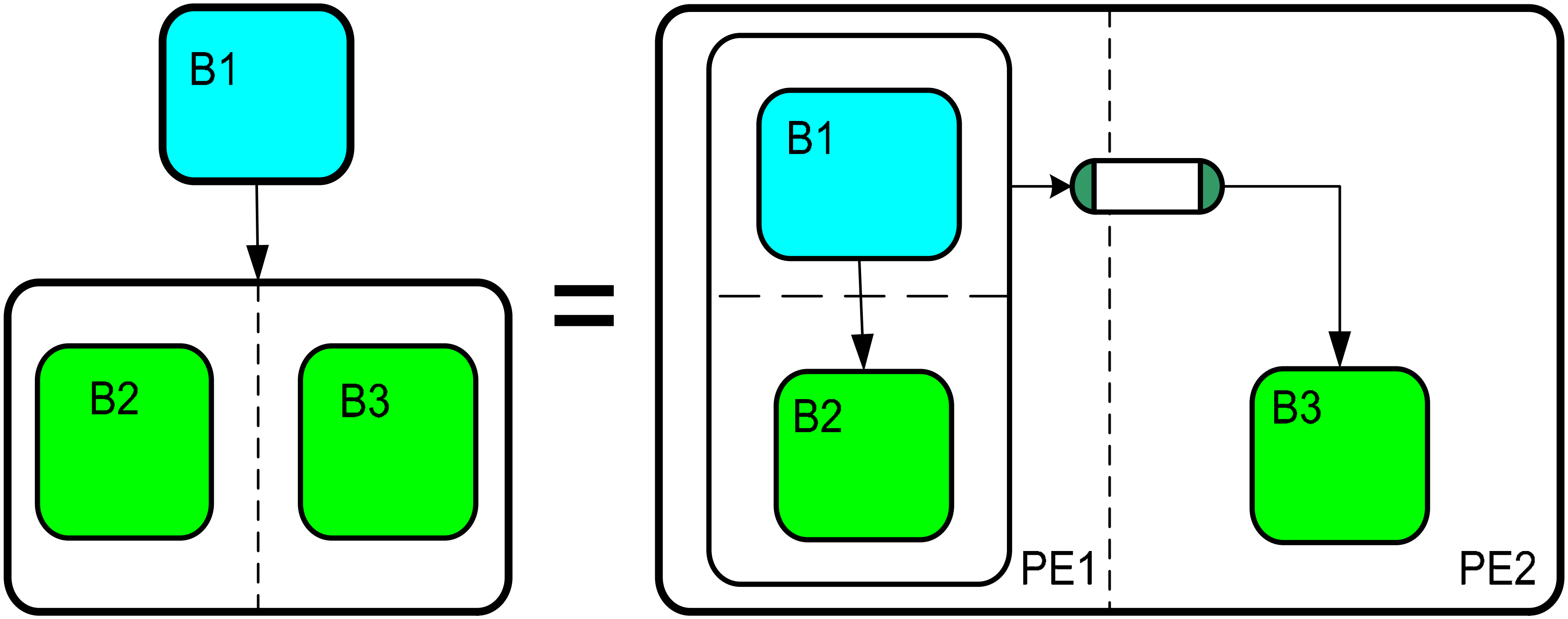
Algebra: < objects, operations>

$$a*(b+c) = a*b + a*c$$

Arithmetic algebra allows creation of expressions and equivalences

Model Algebra

Model algebra: <objects, compositions>



Model algebra allows creation of models and model equivalences

Specify-Explore-Refine Methodology

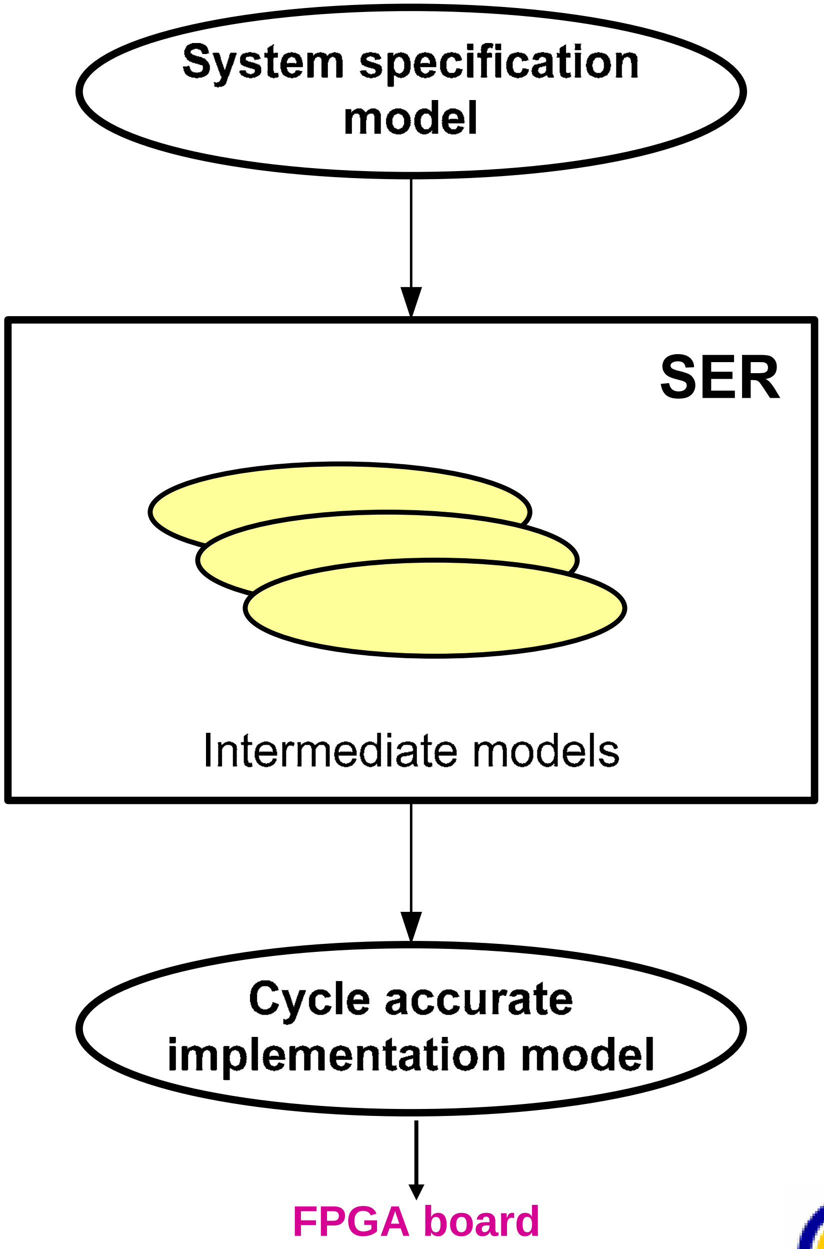
Design decisions

↓

Model refinement

↓

Replacement or re-composition



System specification model

SER

Intermediate models

Cycle accurate implementation model

FPGA board

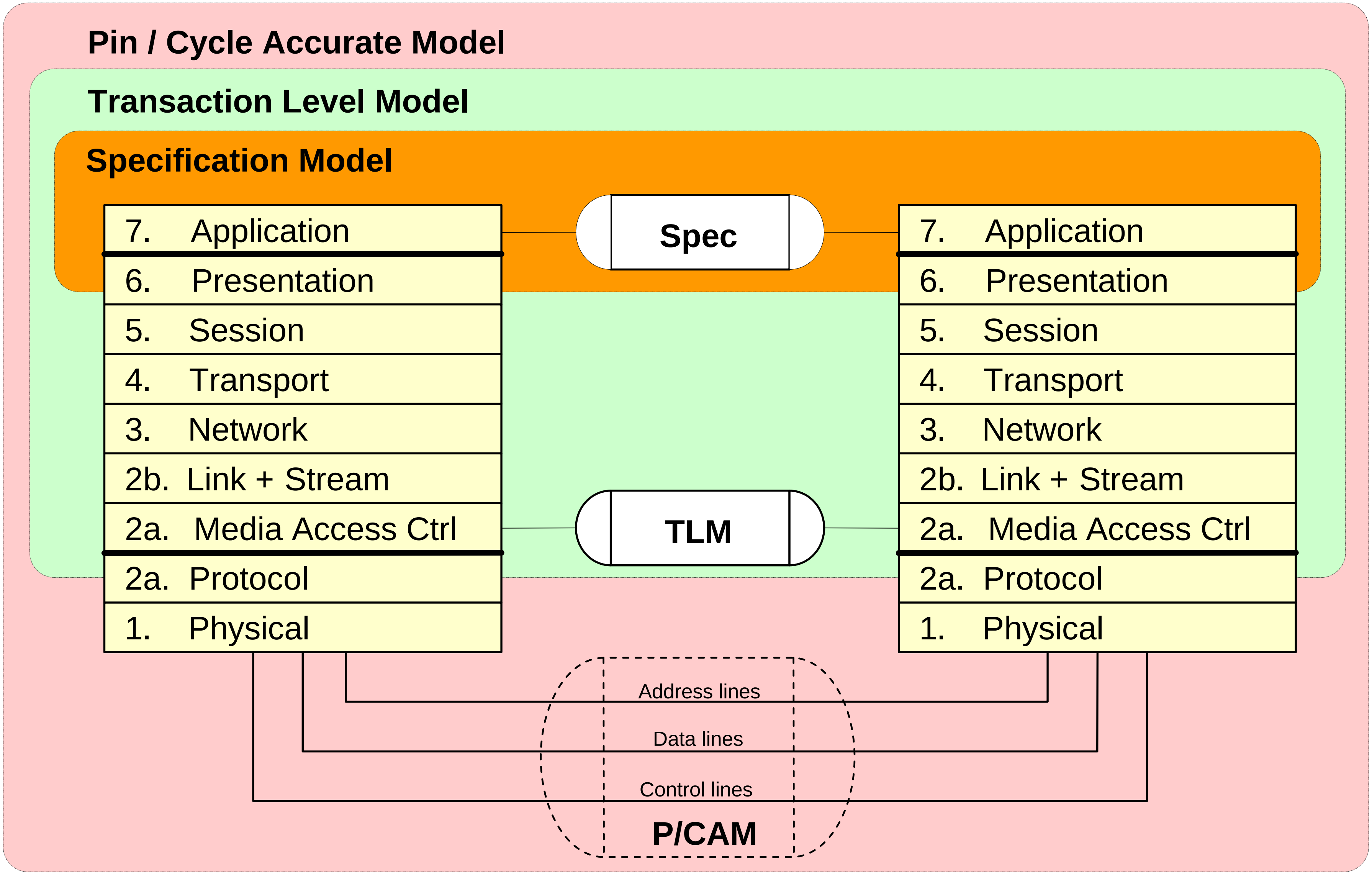
How many models?

Minimal set for any methodology

(3 is enough)

- System specification model (application designers)
- Transaction-level model (system designers)
- Pin&Cycle accurate model (implementation designers)

Three Models (with Respect to OSI)



Pin / Cycle Accurate Model

Transaction Level Model

Specification Model

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2b. Link + Stream
2a. Media Access Ctrl
2a. Protocol
1. Physical

Spec

TLM

7. Application
6. Presentation
5. Session
4. Transport
3. Network
2b. Link + Stream
2a. Media Access Ctrl
2a. Protocol
1. Physical

Address lines

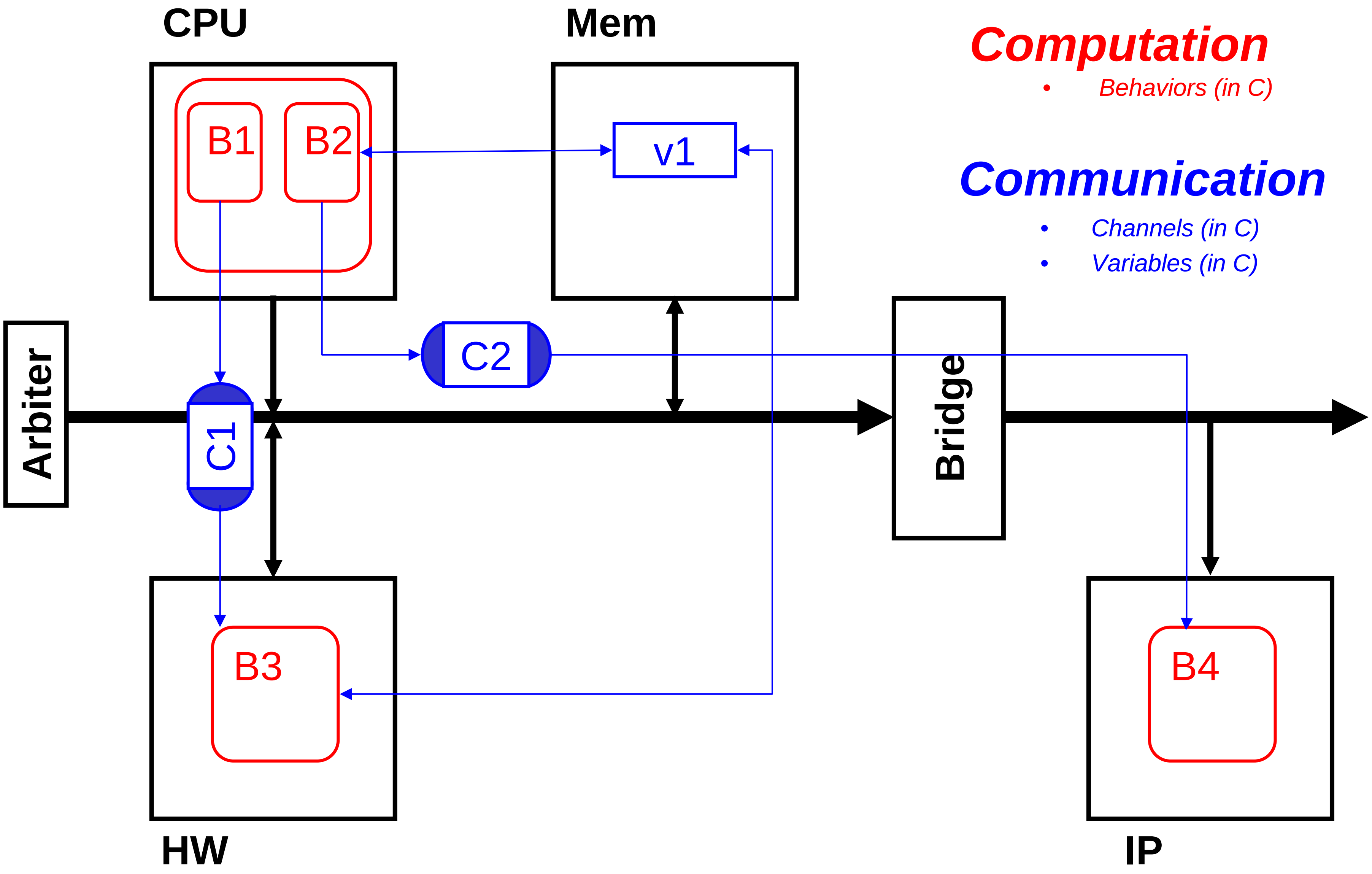
Data lines

Control lines

P/CAM

Source: G Schirner

System Specification



Computation

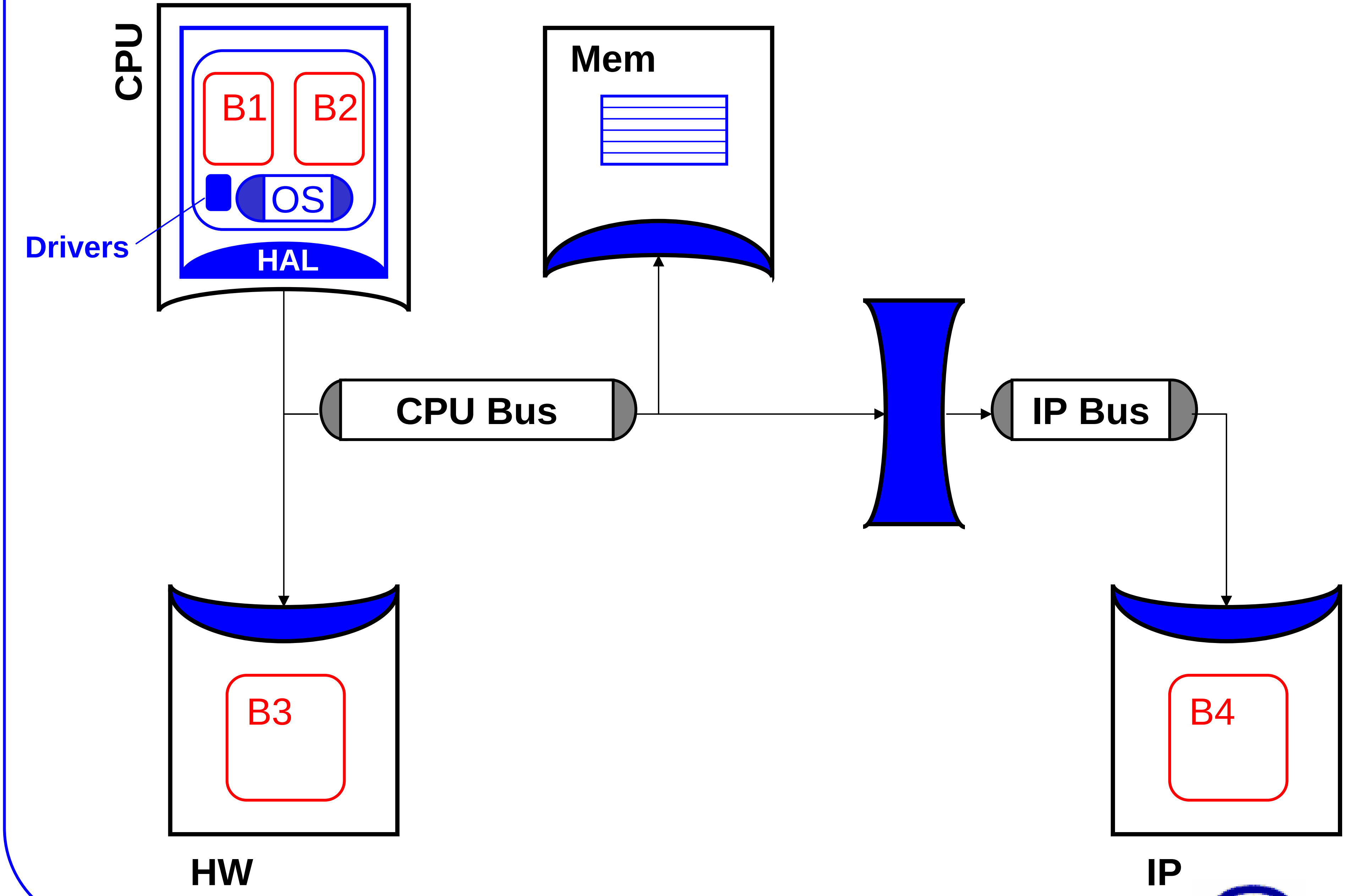
- *Behaviors (in C)*

Communication

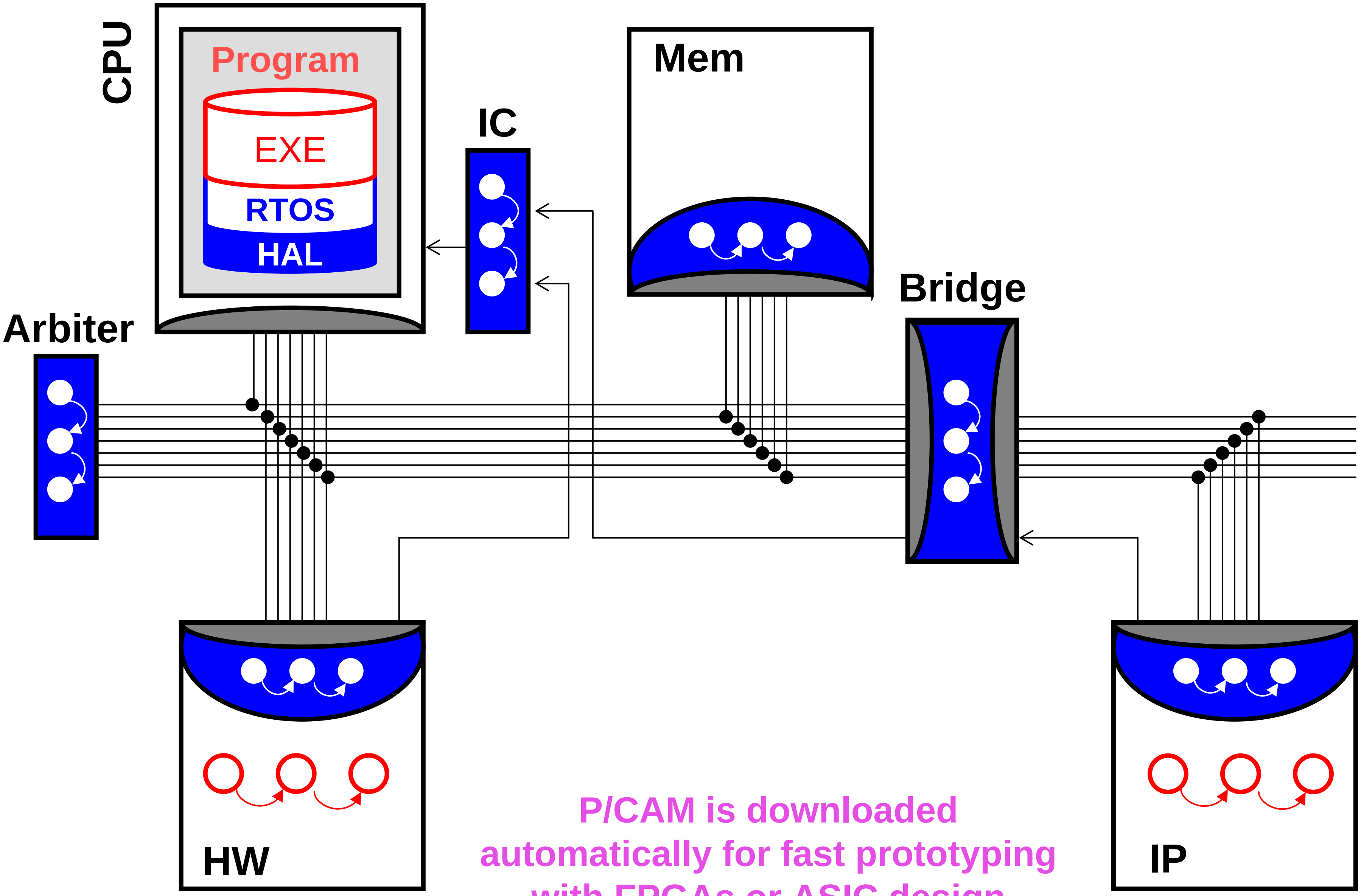
- *Channels (in C)*
- *Variables (in C)*

System Definition* = *(Partial) Platform* + *(Partial) Spec

Transaction-Level Model (TLM)



Pin/Cycle Accurate Model (P/CAM)



P/CAM is downloaded automatically for fast prototyping with FPGAs or ASIC design

Source: D. Gajski et al.

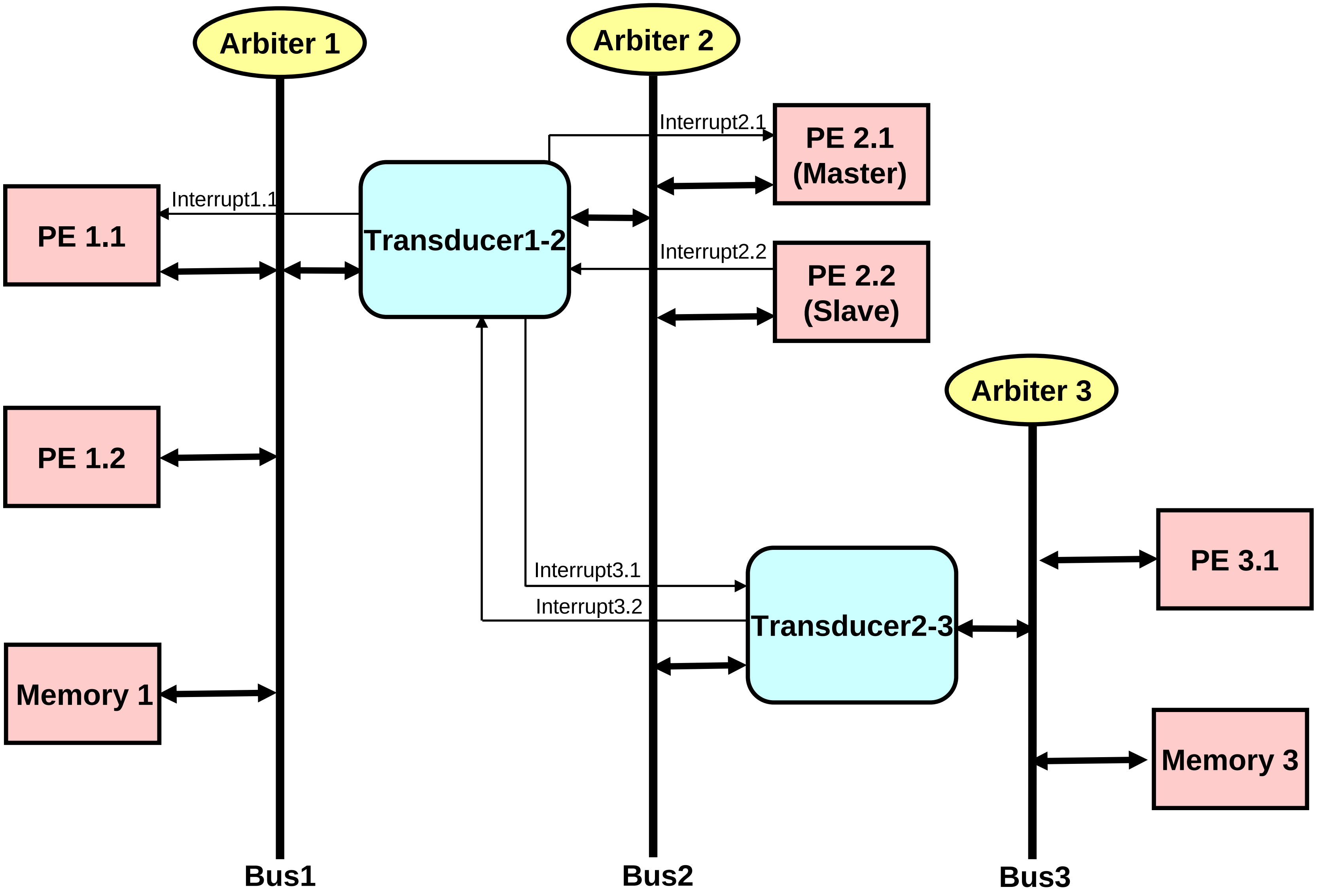
How many components?

Minimal set for any design

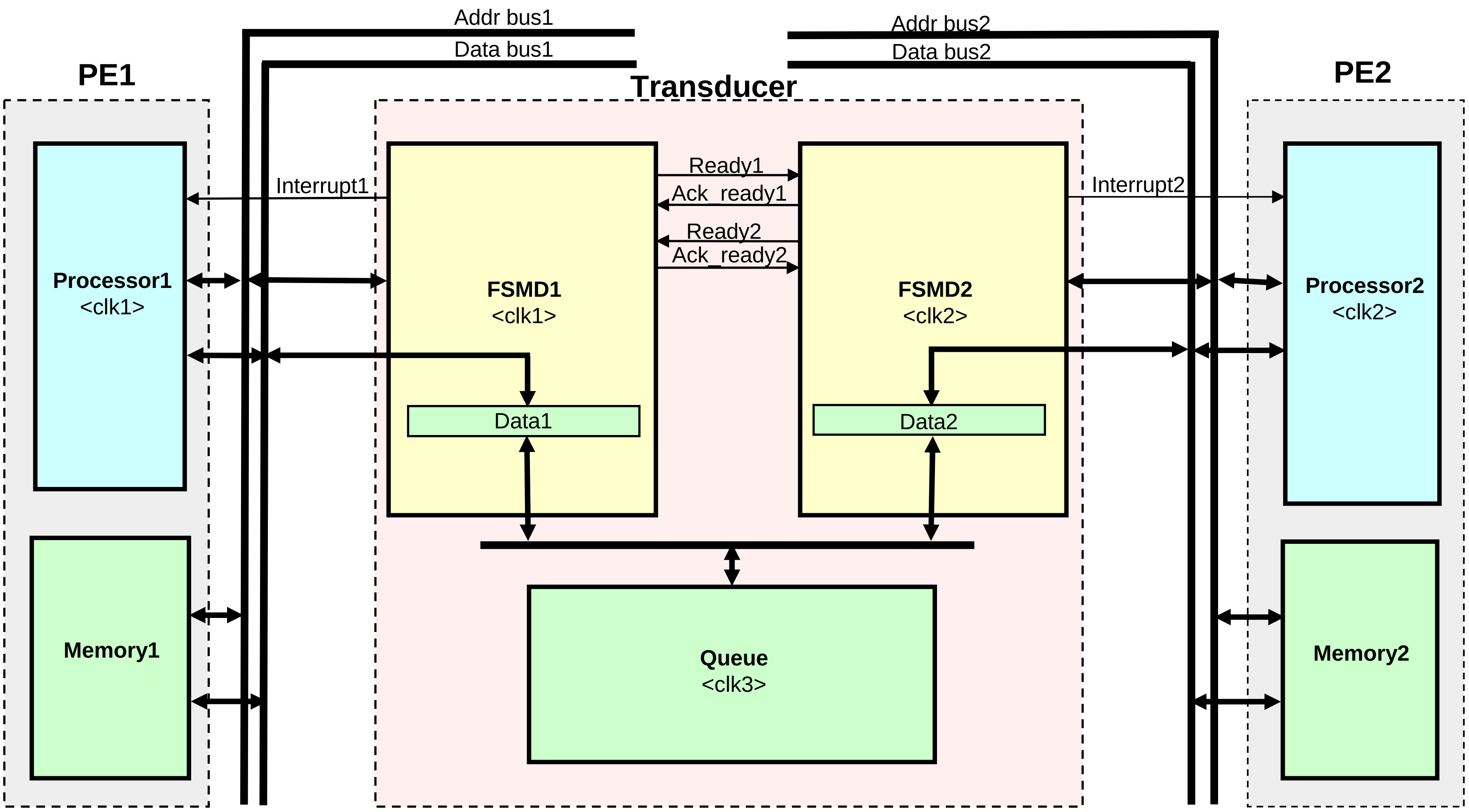
(4 is enough?)

- Processing element (PE)
- Memory
- Transducer / Bridge
- Arbiter

General System Model



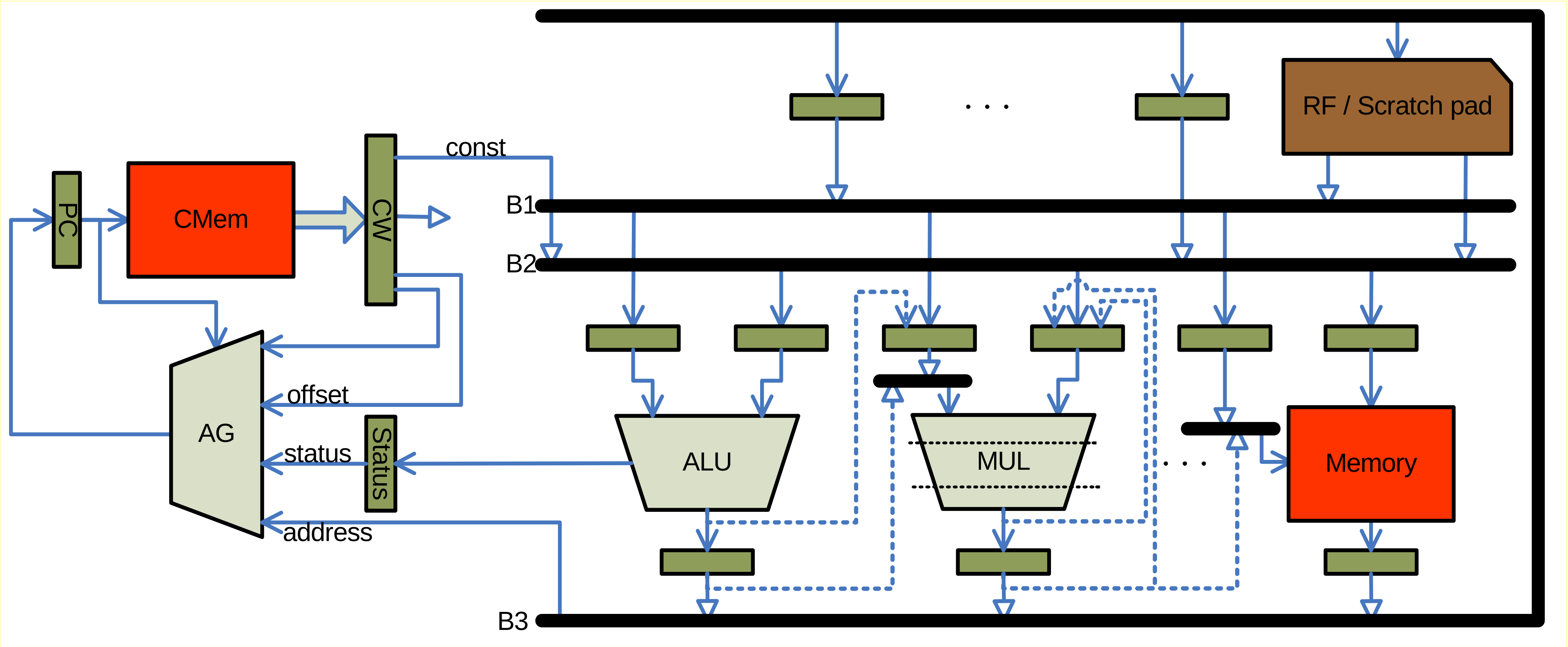
Transducer Model



Source: H. Cho

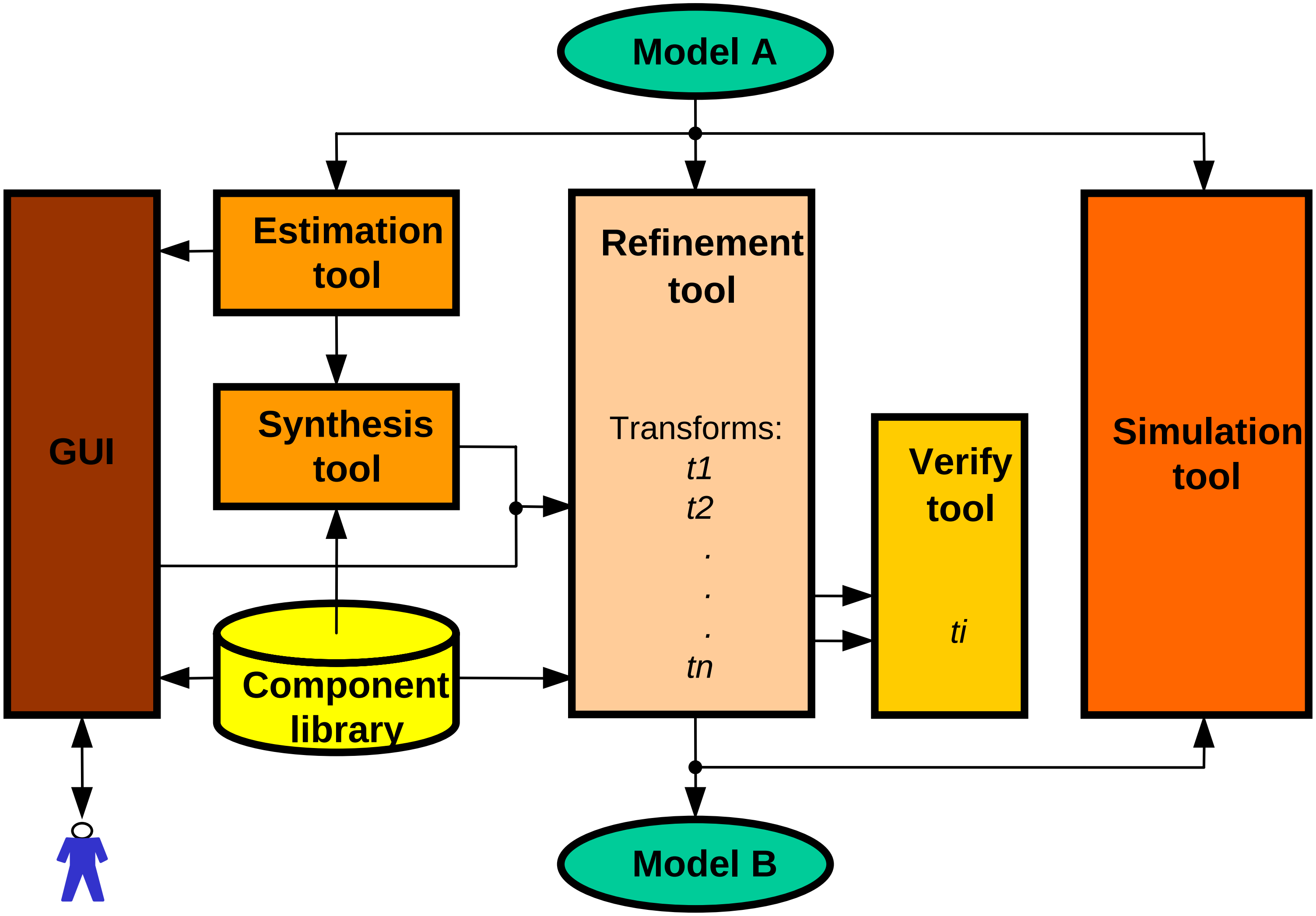
Processing Element: NISC technology

- Direct compilation of C to HW (fastest possible execution)
- Statically and dynamically reconfigurable (anytime, anywhere)
- Designed for manufacturability (solving timing closure)



Programmable controller | Datapath | Multi-cycle units | Pipelined units | Controller pipelining | Datapath pipelining | Data forwarding

General System Design Environment



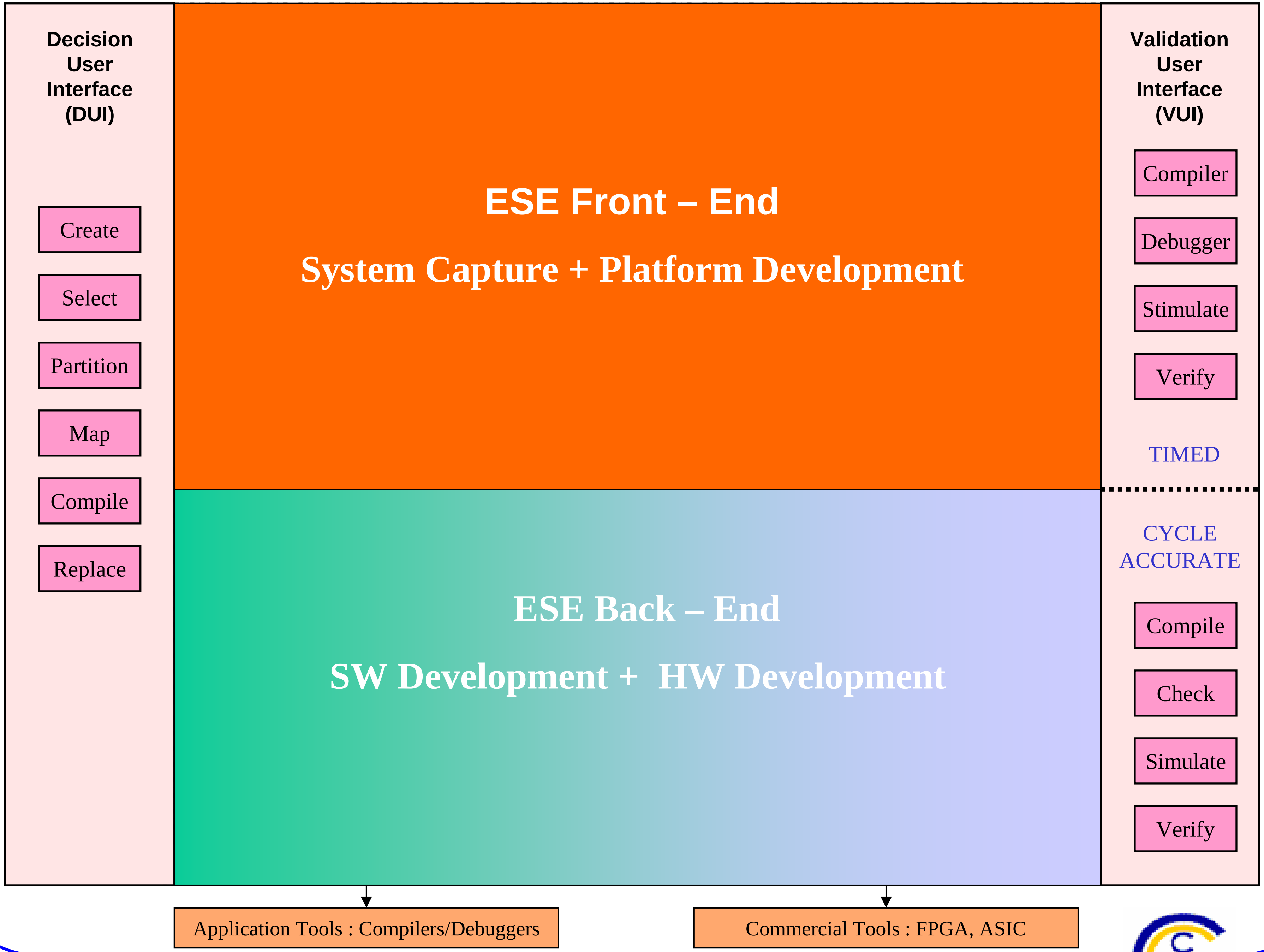
How many tools?

Minimal set for any methodology

(2 is enough?)

- Front-End (for application developers)
  - **Input: C, C++, Mathlab, UML, …**
  - **Output: TLM**
- Back-End (for SW/HW system designers)
  - **Input : TLM**
  - **Output: Pin/Cycle accurate Verilog/VHDL**

ES Environment



Decision User Interface (DUI)

Create

Select

Partition

Map

Compile

Replace

ESE Front – End

System Capture + Platform Development

ESE Back – End

SW Development + HW Development

Validation User Interface (VUI)

Compiler

Debugger

Stimulate

Verify

TIMED

CYCLE ACCURATE

Compile

Check

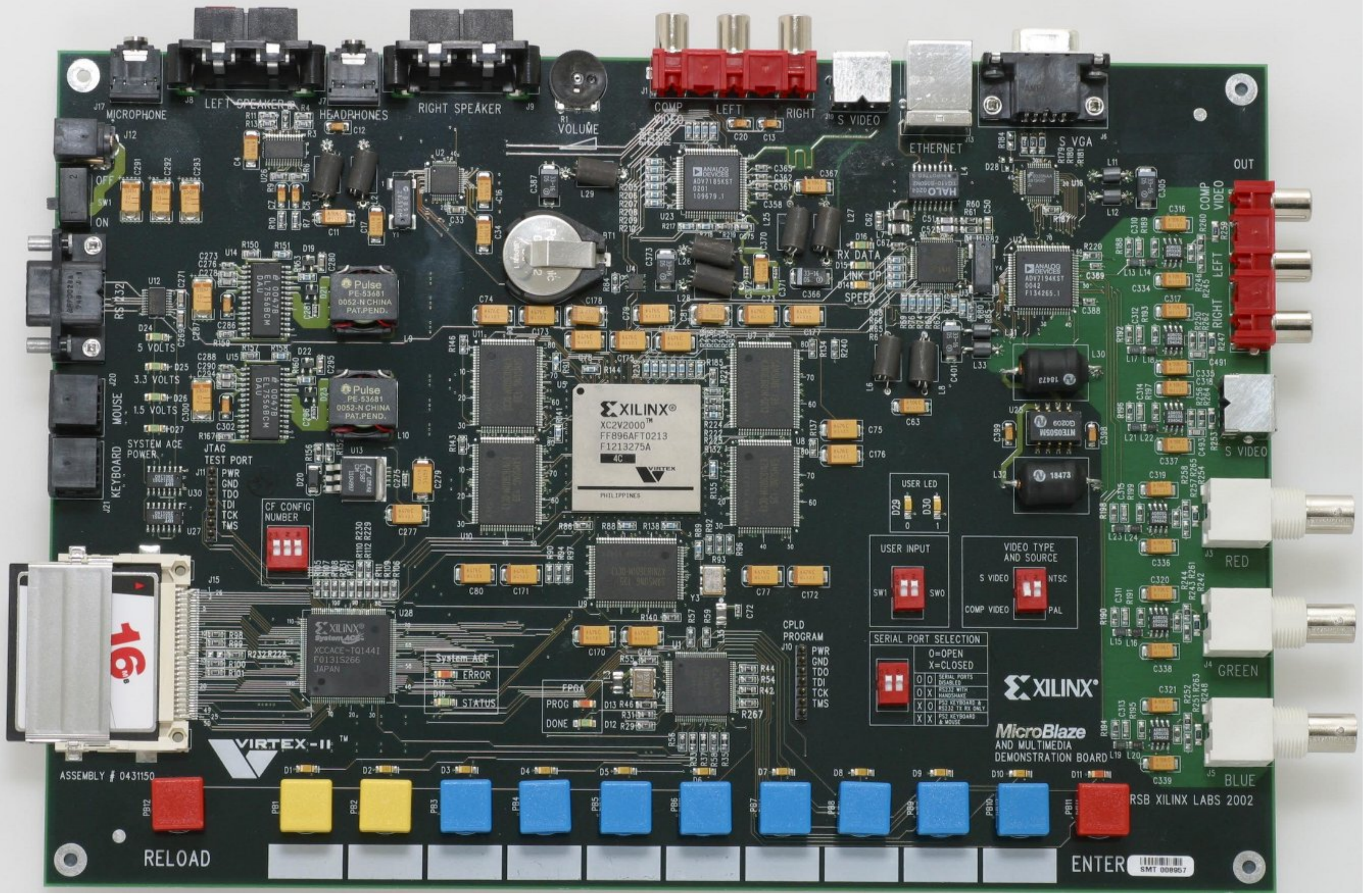
Simulate

Verify

Application Tools : Compilers/Debuggers

Commercial Tools : FPGA, ASIC

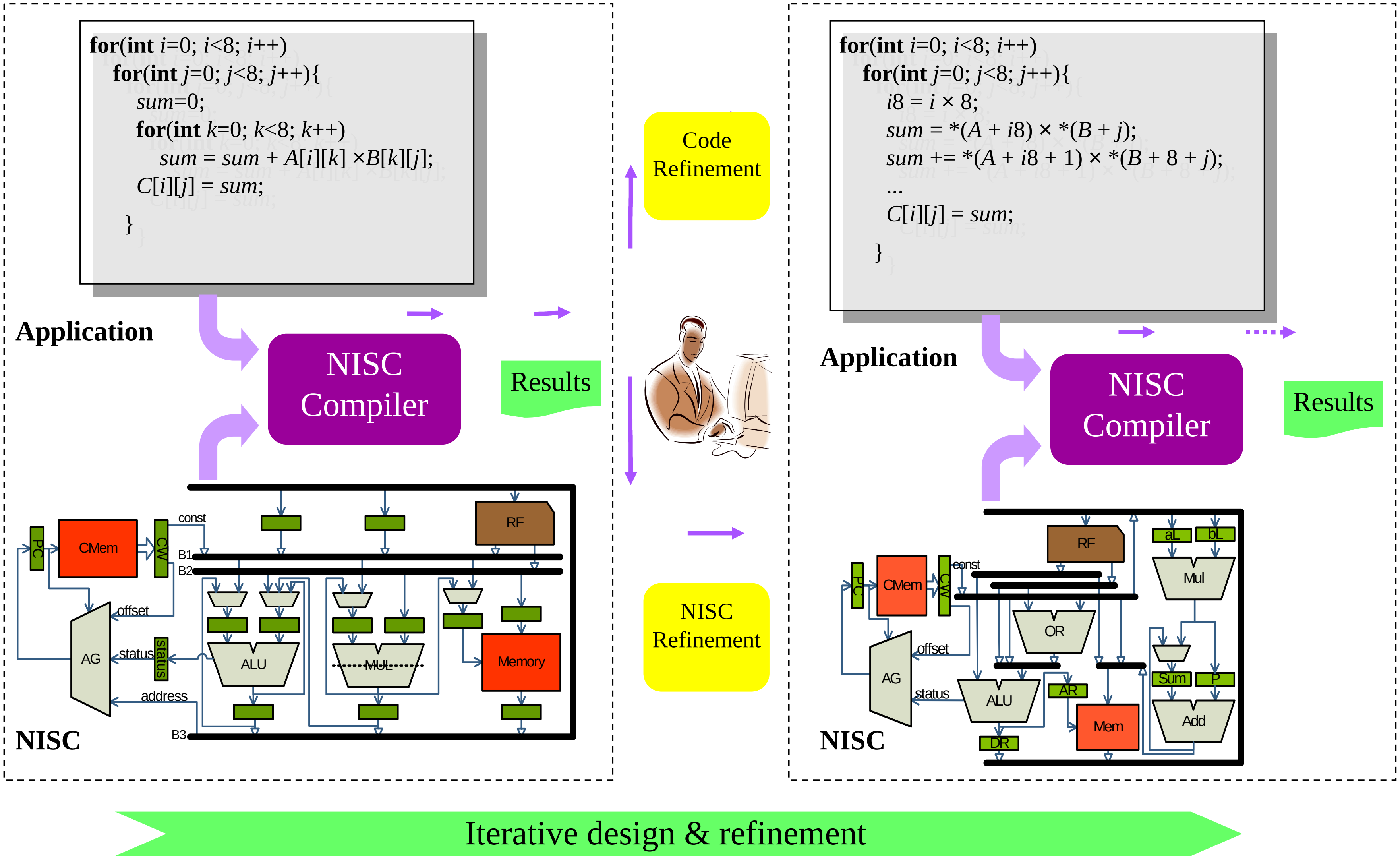
Benefit: Spec-to-Prototype in 1 Week



Does it work?

- Intuitively it does
  - Well defined models, rules, transformations, refinements
  - Worked in the past: layout, logic, RTL?
  - System level complexity simplified
- Proof of concept demonstrated
  - Embedded System Environment (ESE)
  - Automatic model generation
  - Model synthesis and verification
  - Universal IP technology (NISC)
  - Productivity gains greater then 1000
- Benefits
  - Large productivity gains
  - Easy design management
  - Easy derivatives
  - Shorter TTM

Design flow with NISC technology

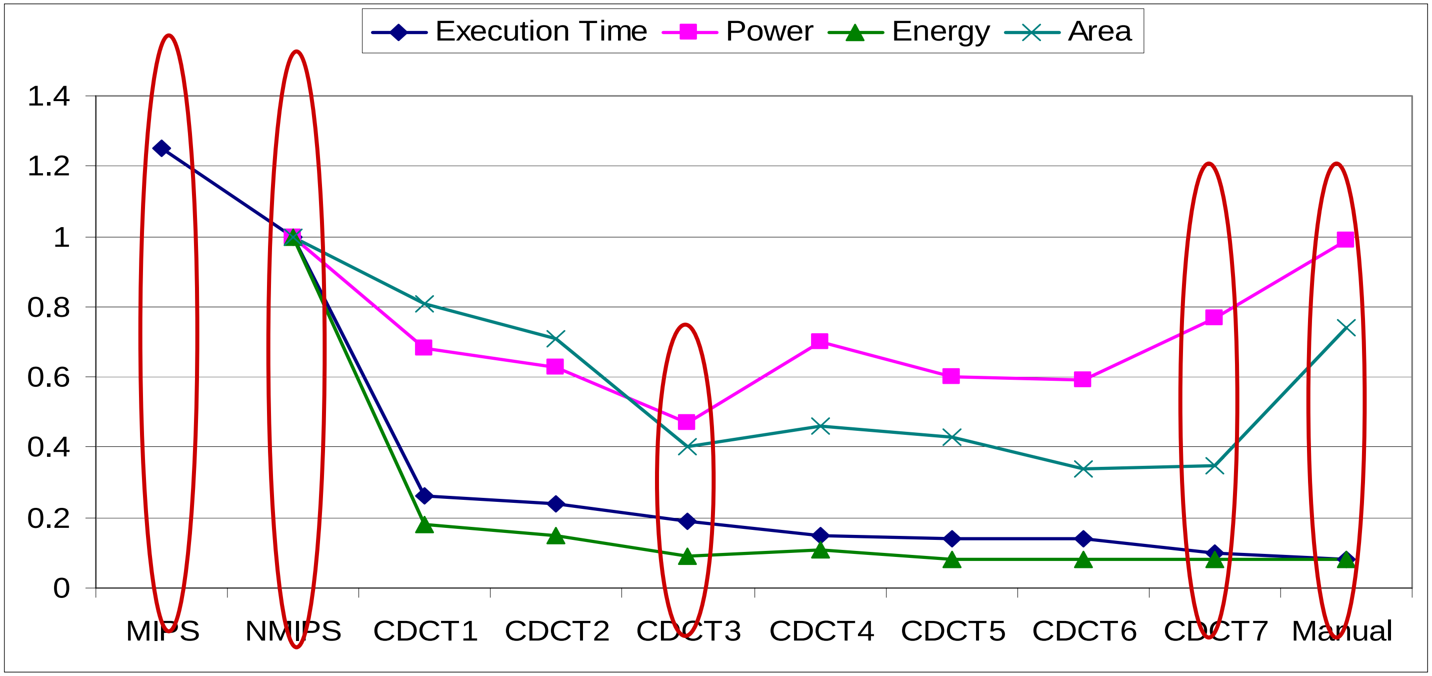


```
for(int i=0; i<8; i++)
   for(int j=0; j<8; j++){
      sum=0;
      for(int k=0; k<8; k++)
         sum = sum + A[i][k] ×B[k][j];
      C[i][j] = sum;
   }
```

```
for(int i=0; i<8; i++)
   for(int j=0; j<8; j++){
      i8 = i × 8;
      sum = *(A + i8) × *(B + j);
      sum += *(A + i8 + 1) × *(B + 8 + j);
      …
      C[i][j] = sum;
   }
```

Source: M. Reshadi

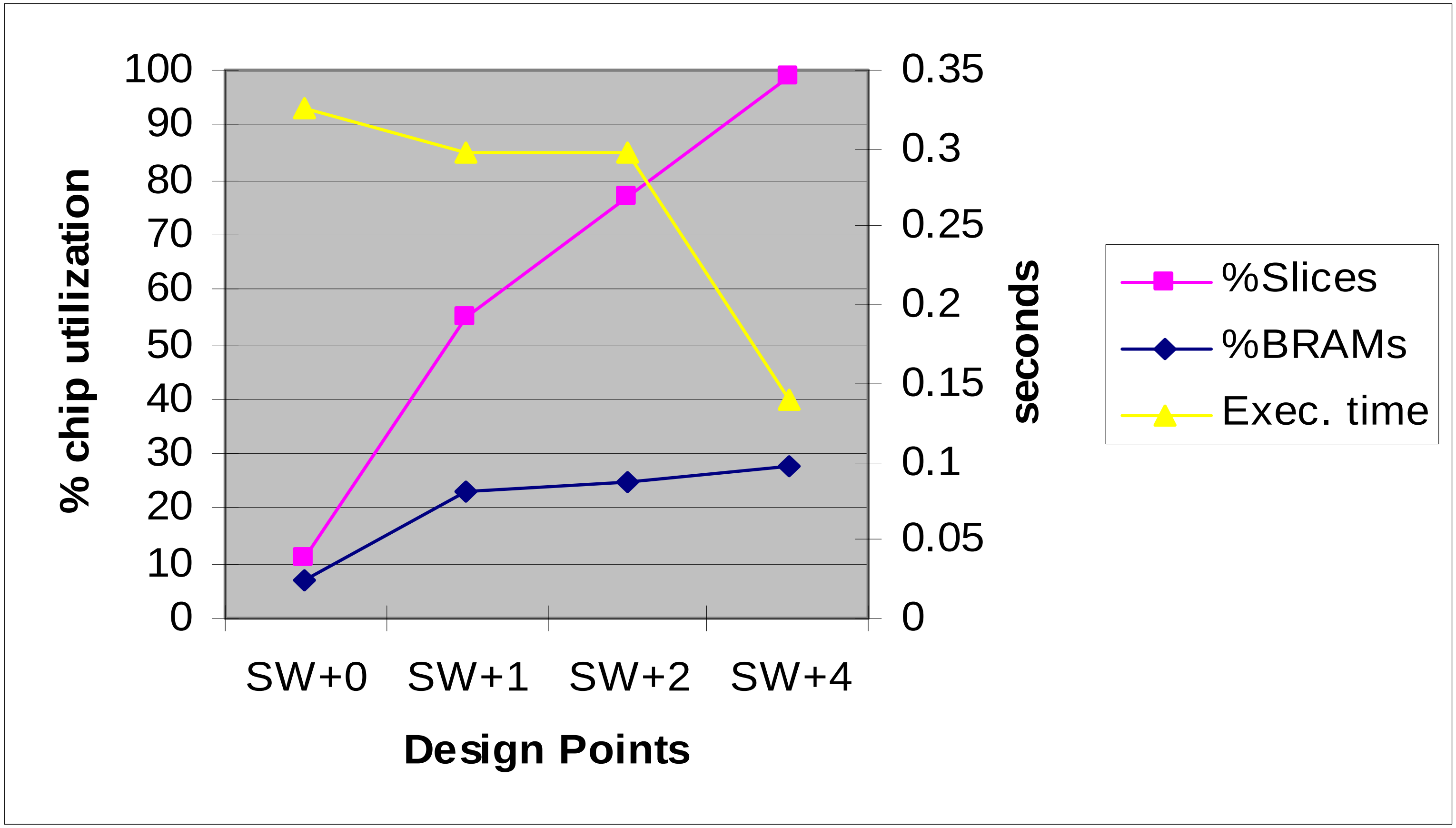
DCT with NISC technology



	Performance	Power saving	Energy saving	Area reduction
NMIPS vs. MIPS	1.25X	NA	NA	NA
CDCT3 vs. NMIPS	5.3X	2.1X	11.6X	2.5X
CDCT7 vs. NMIPS	10X	1.3X	12.8X	3X
CDCT7 vs. Manual	0.83X	1.3X	0	2.1X

Source: B. Gorjiara

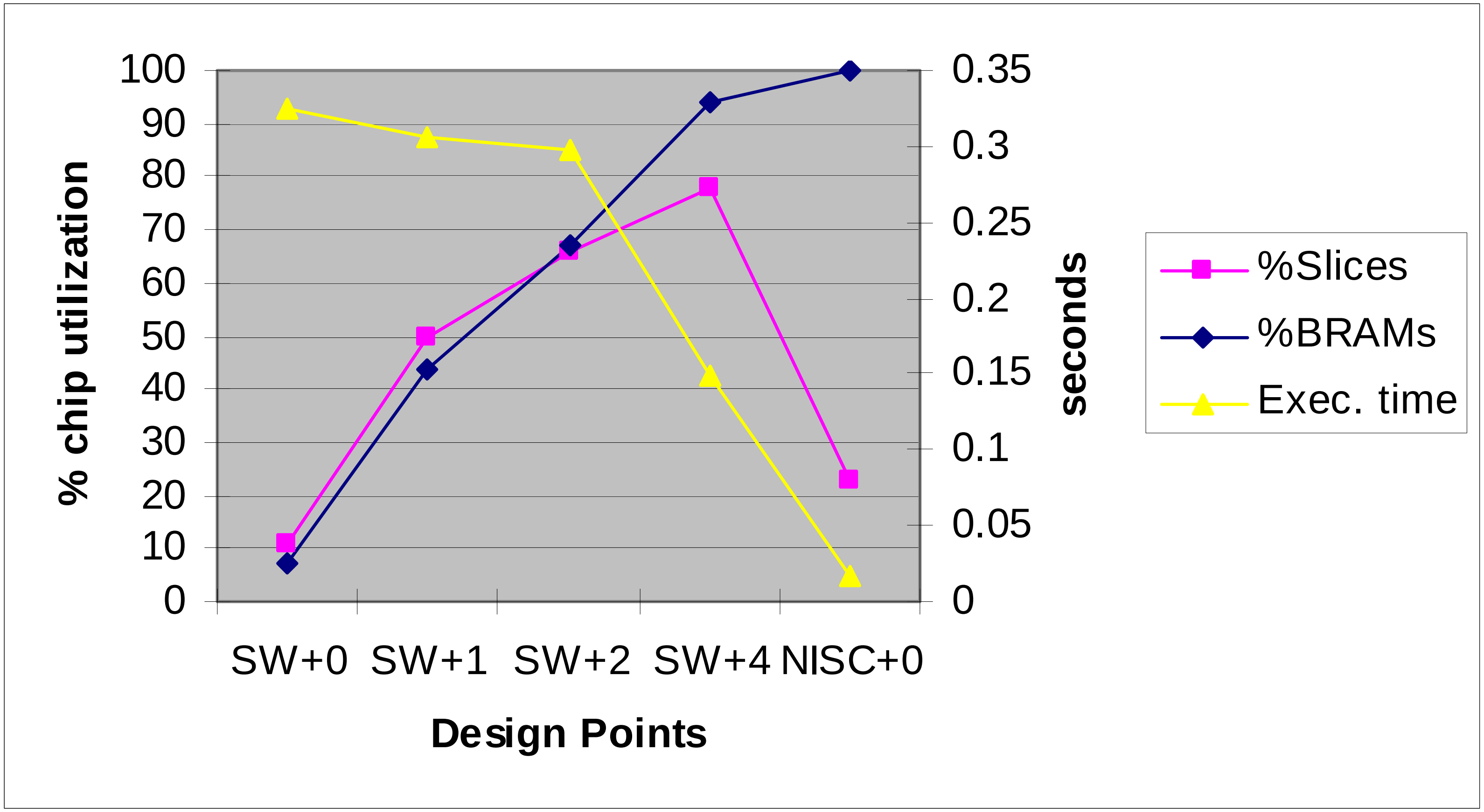
MP3 on Xillinx with ESE



- **Area**
  - % of FPGA slices and BRAMS
- **Performance**
  - Time to decode 1 frame of MP3 data

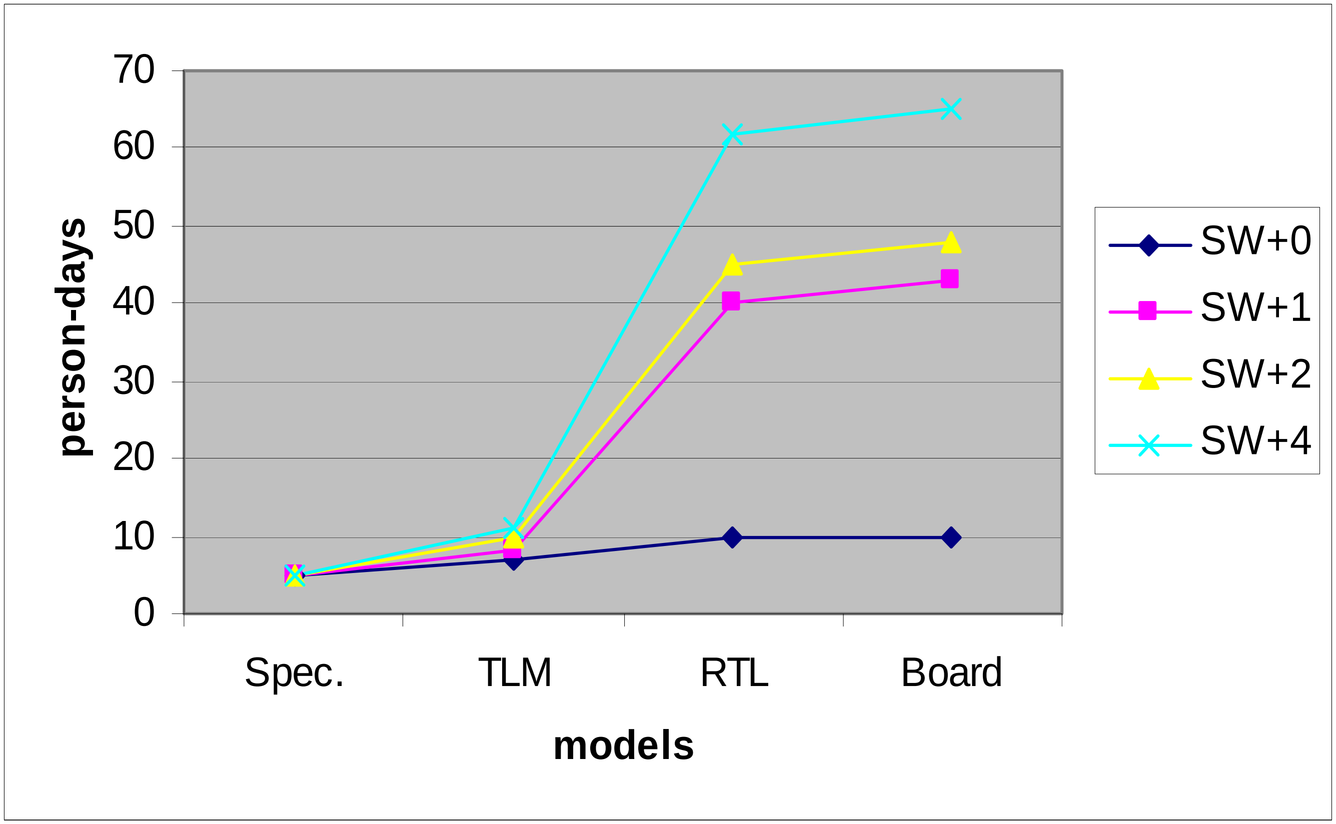
Source: S. Abdi

MP3 on Xillinx with ESE using NISC



- **Area**
  - NISC uses fewer FPGA slices and more BRAMs than manual HW
- **Performance**
  - NISC comparable to manual HW and much faster than SW

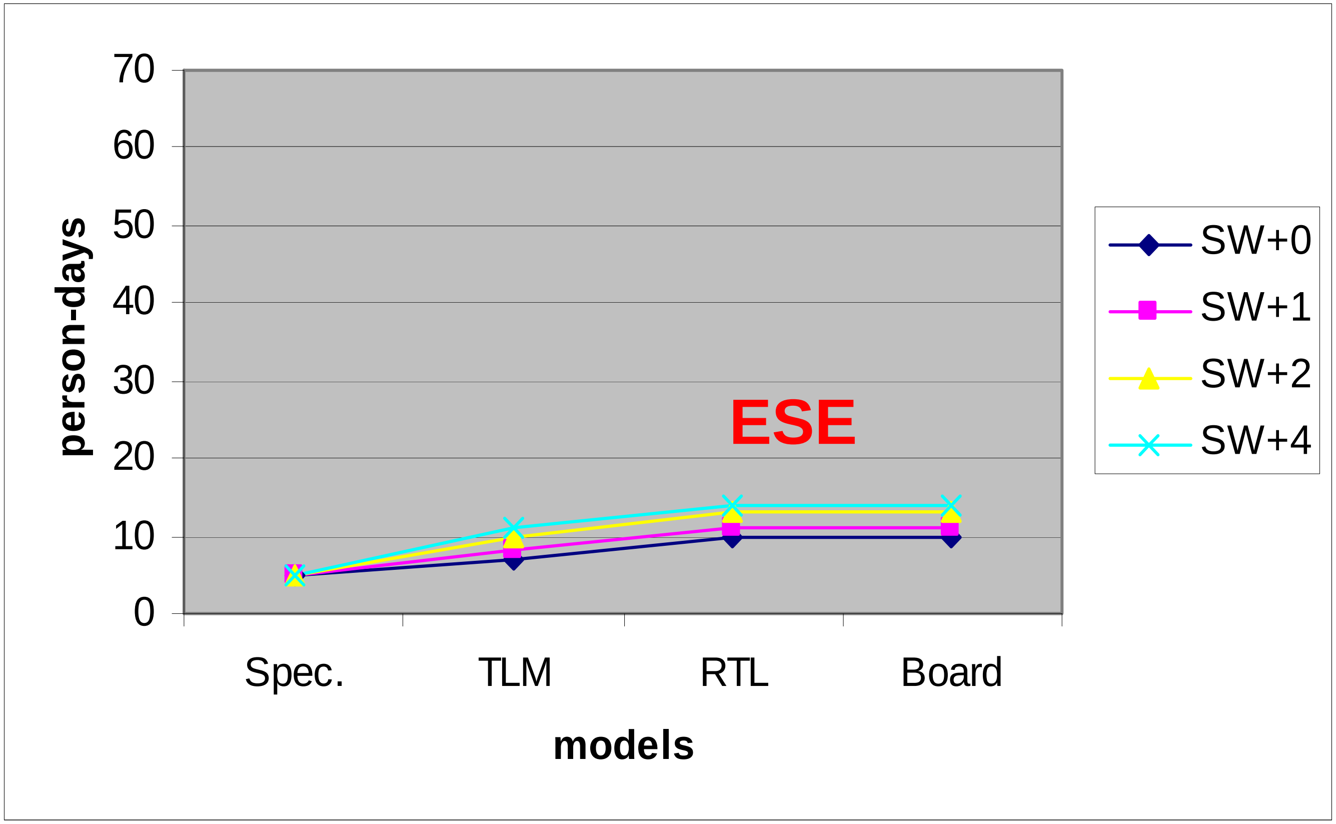
Manual Development Time



- **Model Development time**
  - Includes time for C, TLM and RTL Verilog coding and debugging

Source: S. Abdi

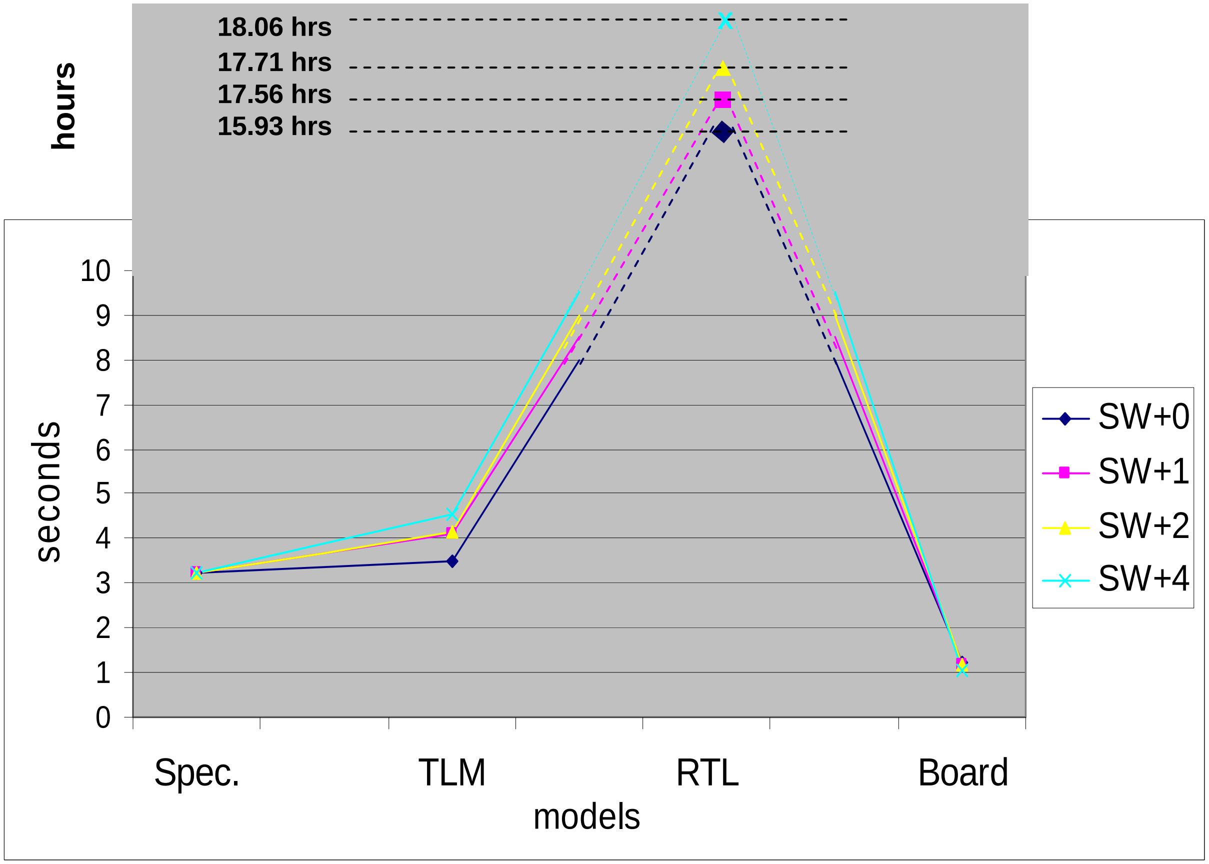
Development Time with ESE



- **ESE drastically cuts RTL and Board development time**
  - Models can be developed at Spec and TL
  - Synthesizable RTL models are generated automatically by ESE

Source: S. Abdi

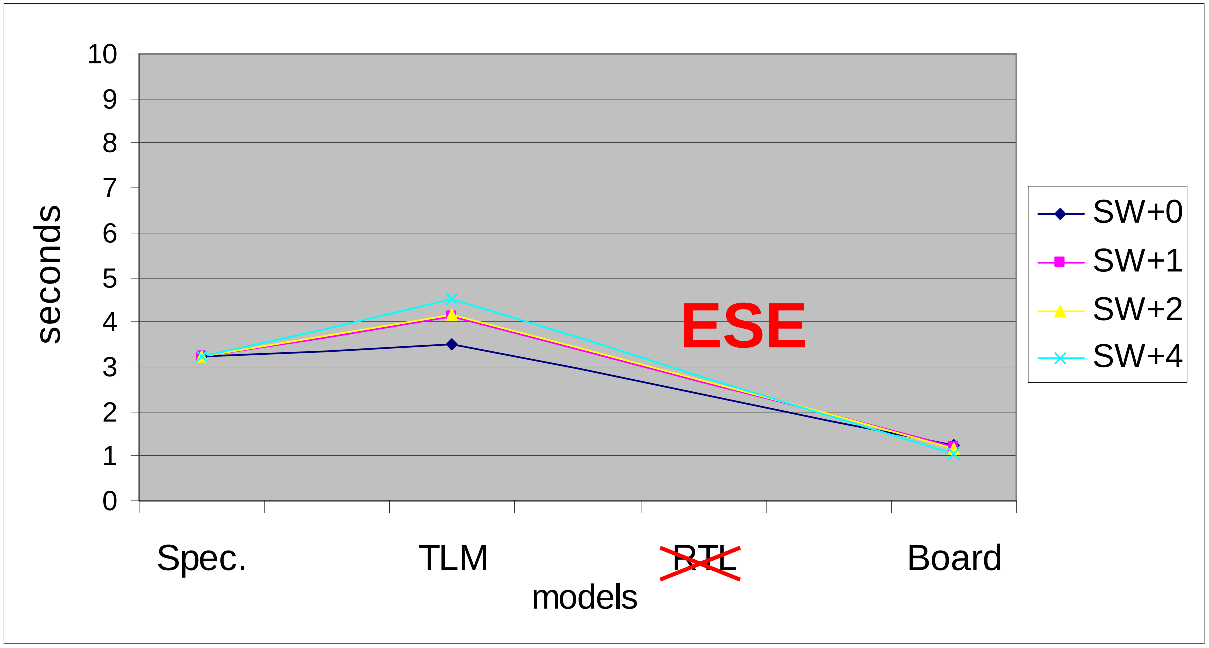
Validation Time



- Simulation time measured on 3.3 GHz processor
- Emulation time measured on board with Timer

Source: S. Abdi

Validation Time with ESE



- **ESE cuts validation time from hours to seconds**
  - No need to verify RTL models
  - Designers can perform high speed validation at TLM and board

Source: S. Abdi

Conclusions

- Extreme makeover is necessary for a new paradigm, where
  - SW = HW = SOC = Embedded Systems
  - Simulation based flow is not acceptable
  - Design methodology is based on scientific principles
- Model algebra is enabling technology for
  - System design, modeling and simulation
  - System synthesis, verification, and test
- What is next?
  - Change of mind
  - Application oriented EDA
  - Looking for early adapters

Thank You

Daniel Gajski

Center for Embedded Computer Systems (CECS)

www.cecs.uci.edu