Platform Modeling for Exploration and Synthesis

Andreas Gerstlauer

Electrical and Computer Engineering

University of Texas at Austin

`http://www.ece.utexas.edu/~gerstl`

Gunar Schirner

Electrical and Computer Engineering

Northeastern University

`http://www.ece.neu.edu/~schirner/`

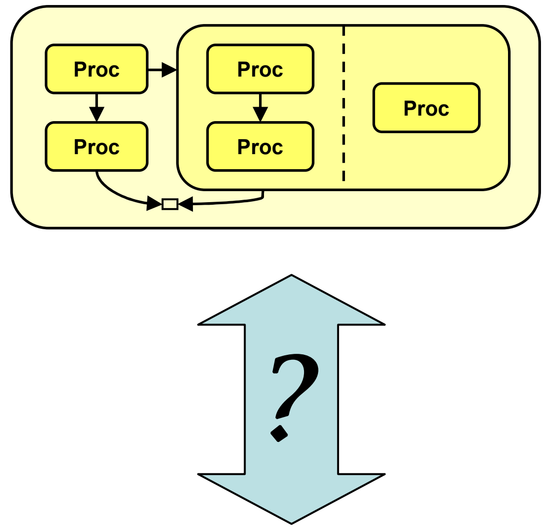
With source material from the book:
D. Gajski, S. Abdi, A. Gerstlauer, G. Schirner. "Embedded System Design: Modeling, Synthesis, Verification," Springer 2009.

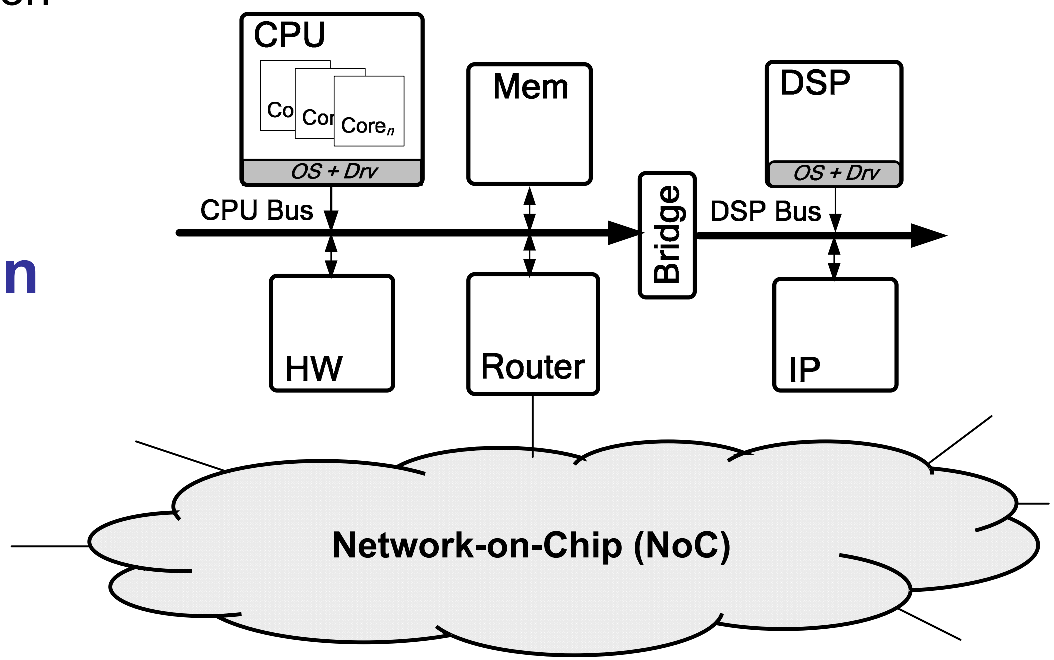
System Design

- **Challenges**
  - Complexity and heterogeneity
    - Application demands
    - Technological advances
  - ➢ Multi-Processor/Multi-Core System-on-Chip (MPCSoC)
    - ➢ Parallelism and communication
    - ➢ Hybrid and hierarchical

➢ **Higher levels of abstraction [ITRS07]**

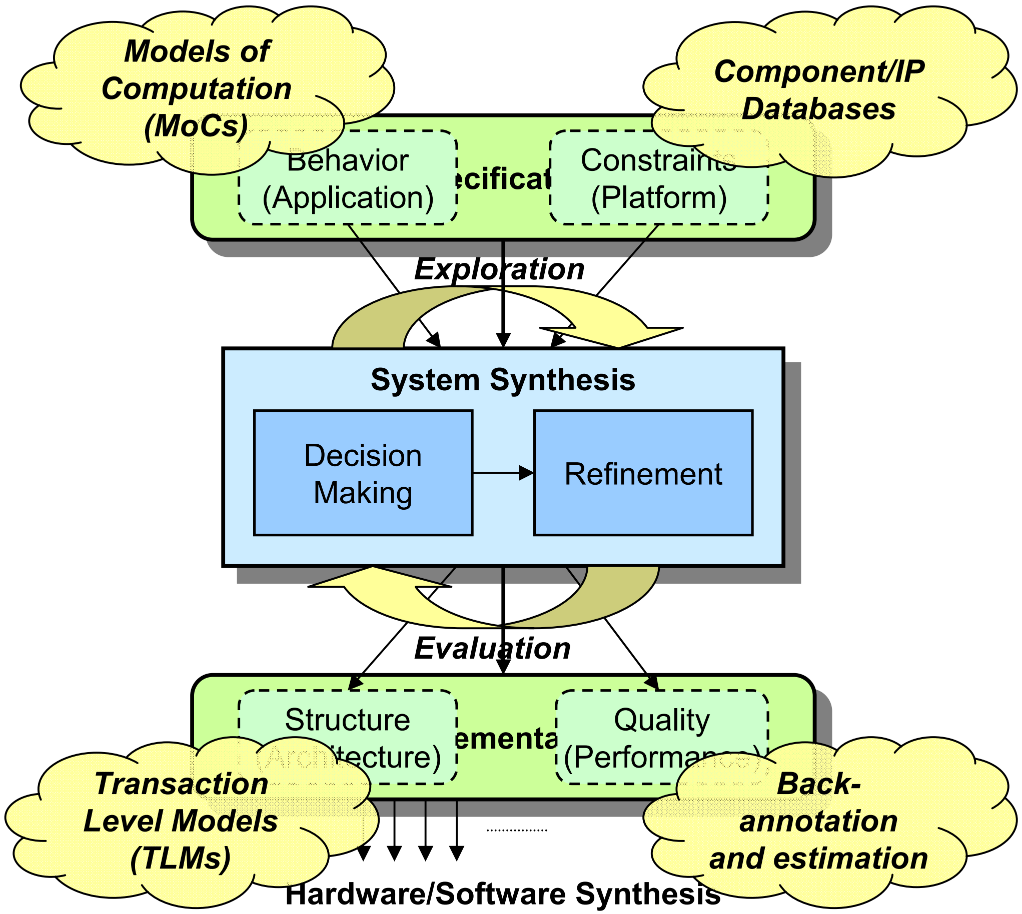
- ➢ System-level design





Electronic System-Level (ESL) Design

- **From specification**
  - Functionality, behavior
    - Application algorithms
  - Constraints
    - Platform components
- **To implementation**
  - Structure
    - Spatial and temporal order
    - Components & connectivity
    - Hardware & software
  - Quality metrics
    - Performance numbers



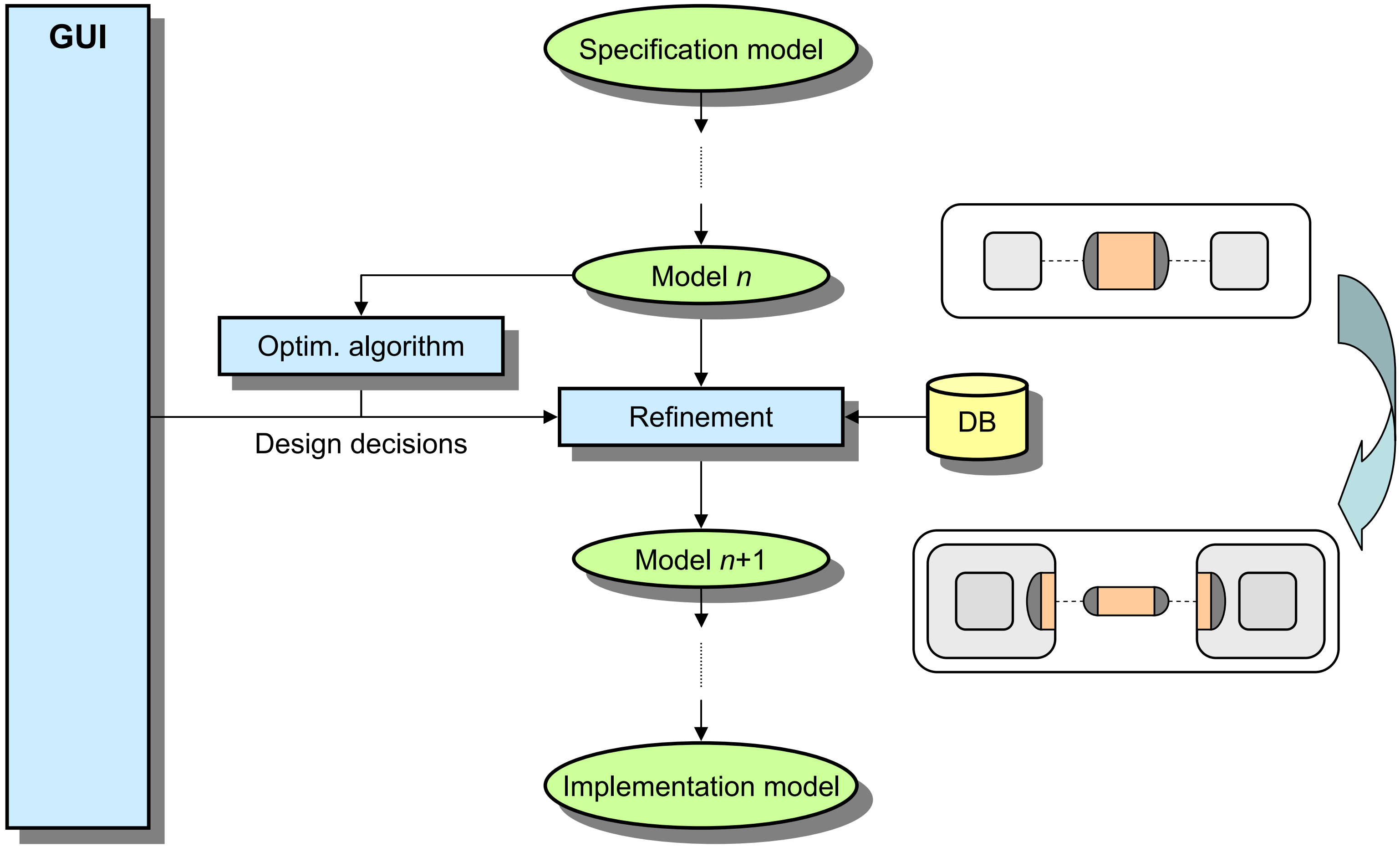
➢ **ESL design automation**
  - Synthesis and exploration
  - ➢ Modeling for representation, simulation and analysis

ESL Design Today

- **Simulation-centric system modeling**
  - Algorithmic specification [KPN, SDF, StateCharts]
    - Models of Computation (MoCs) [Ptolemy, MILAN]
    - Model-Based Design (MBD) [UML/MARTE, Matlab/Simulink]
  - Virtual system prototyping [TLM]
    - System-level design languages (SLDLs) [SpecC, SystemC]
    - IP integration [OCP, SPIRIT/IP-XACT]
  - ➢ *Horizontal integration of different models / components*
  - ➢ *Lack of vertical integration for synthesis and verification*

- **High-level synthesis**
  - C-to-RTL [Forte, Mentor Catapult]
    - Interface specification [SystemVerilog]
  - ➢ Single hardware unit only

Synthesis Flow



➤ **Successive, stepwise, layer-based model refinement**

System Design Needs

- **Design models as abstraction of a design instance**
  - Representation for validation and analysis
  - Specification for further implementation
  - ➢ Documentation & specification

➢ **Systematic modeling flow and methodology**
  - Set of models and design steps
  - ➢ From specification to implementation

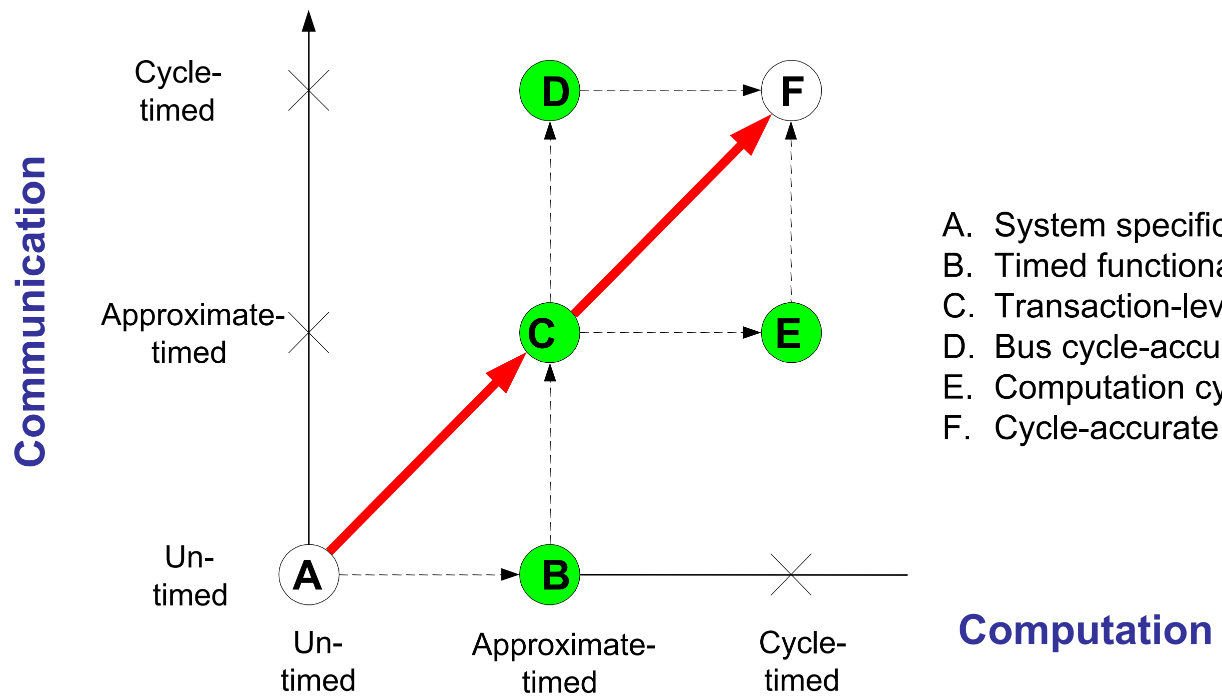
➢ **Well-defined, rigorous model semantics**
  - Fast and accurate
  - Unambiguous, explicit abstractions
    - Objects and composition rules
  - ➢ Exploration, synthesis and verification

Outline

- ✓ **Introduction**
- **Component modeling**
  - Communication modeling
  - Computation modeling
- **Platform modeling**
  - System models
  - Design example
- **Summary and conclusions**

Platform Models

- **Abstraction based on level of detail & granularity**
  - Data & timing
  - Computation & communication



A. System specification model
B. Timed functional model
C. Transaction-level model (TLM)
D. Bus cycle-accurate model (BCAM)
E. Computation cycle-accurate model (CCAM)
F. Cycle-accurate model (CAM)

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

➢ **Design methodology and modeling flow (A → F)**
  ➢ Set of models and transformations between models

Communication Modeling

- **Basic system communication component is a *bus***
  - Master/slave (AMBA) to network (Ethernet, CAN)
    - Read and write transactions
    - Address and data
    - Protocol timing

➢ **Transaction-Level Modeling (TLM)**
  - Function calls above pins and wires
  - ➢ Various transaction styles and granularity levels [PVT, CCATB]
  - ➢ Various bus architectures [AHB]
  - ➢ Standardization efforts [SystemC TLM 2.0]
    - ➢ General payloads and coding style guidelines
    - ➢ Flexible and widely applicable, but no well-defined semantics

Communication Layers

- **ISO/OSI 7-layer model**

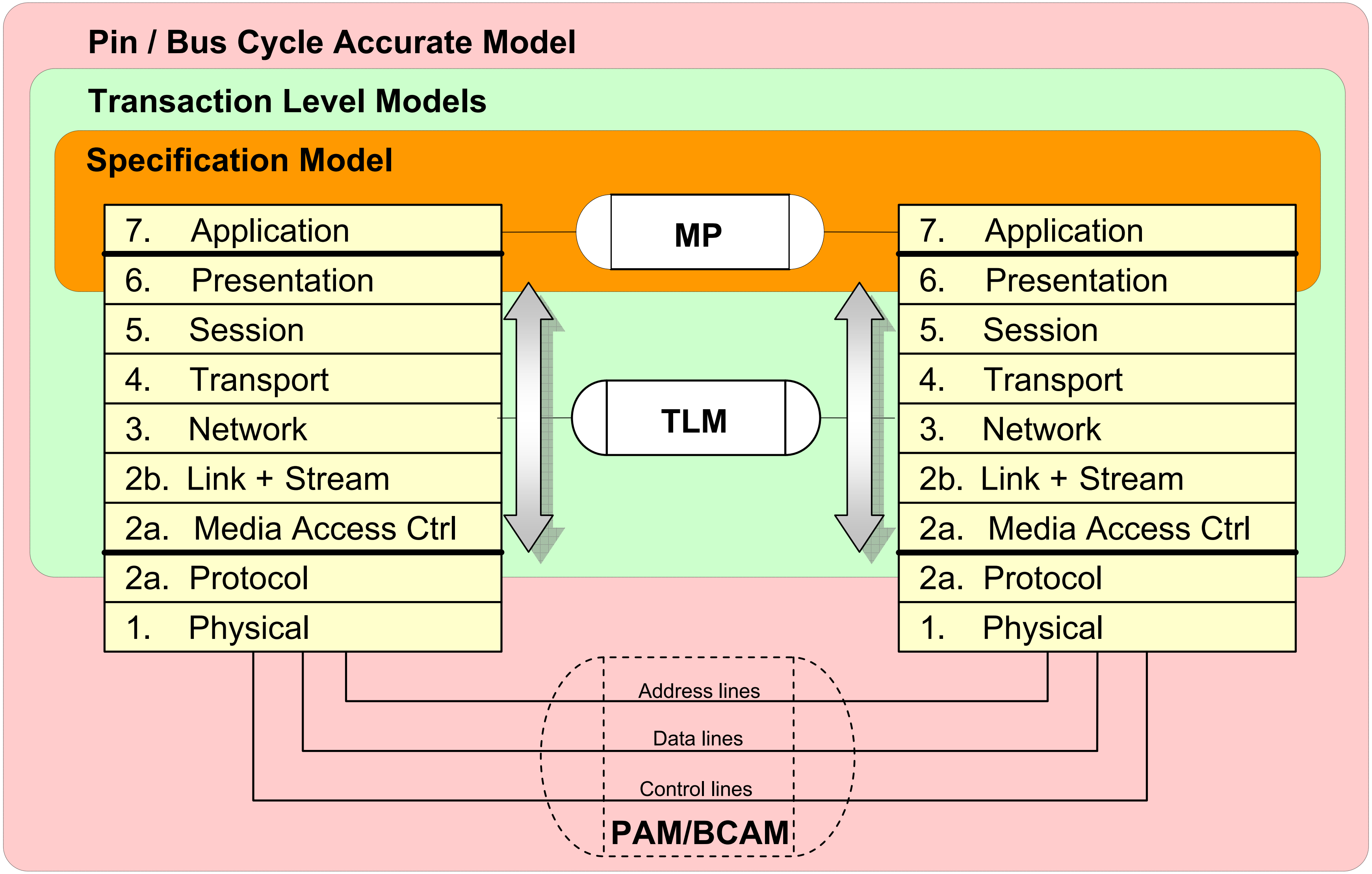
Layer	Semantics	Functionality	Implementation	OSI
Application	Channels, variables	Computation	Application	7
Presentation	End-to-end typed messages	Data formatting	Middleware	6
Session	End-to-end untyped messages	Synchronization, Multiplexing	Middleware	5
Transport	End-to-end data streams	Packeting, Flow control	Middleware	4
Network	End-to-end packets	Subnet bridging, Routing	Middleware	3
Link	Point-to-point logical links	Station typing, Synchronization	Driver	2b
Stream	Point-to-point control/data streams	Multiplexing, Addressing	Driver	2b
Media Access	Shared medium byte streams	Data slicing, Arbitration	HAL	2a
Protocol	Media (word/frame) transactions	Protocol timing	Hardware	2a
Physical	Pins, wires	Driving, sampling	Interconnect	1

➢ **A *model*, not an implementation !**

Source: A. Gerstlaeur, D. Shin, J. Peng, R. Doemer, D. Gajski. "Automatic, Layer-Based Generation of System-On-Chip Bus Communication Models," TCAD, 2007.

Communication Models

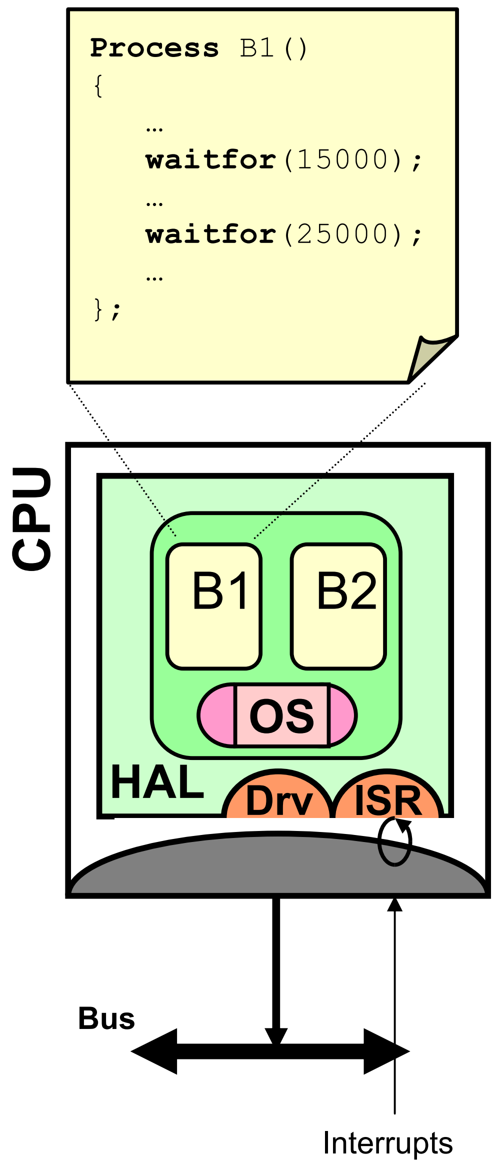
- **From layers to models…**



Computation Modeling

- **Basic system computation component is a *processor***
  - General-purpose programmable to full custom hardware
    - – Controller and datapath
    - – Functionality and execution time
- **Processor simulation**
  - Mircoarchitecture or instruction-set simulation (ISS)
    - – Standard (TLM) integration [CoWare, OVP]
    - – Prioprietary (CPU-centric) backplane [MPARM, VaST, Virtutech]
    - ➢ Limited scalability and slow in multi-processing context
  - ➢ Host-compiled or hybrid simulation
    - ➢ Native execution of functionality
    - ➢ Back-annotation of timing
    - ➢ Lightweight, well-defined model of execution environment

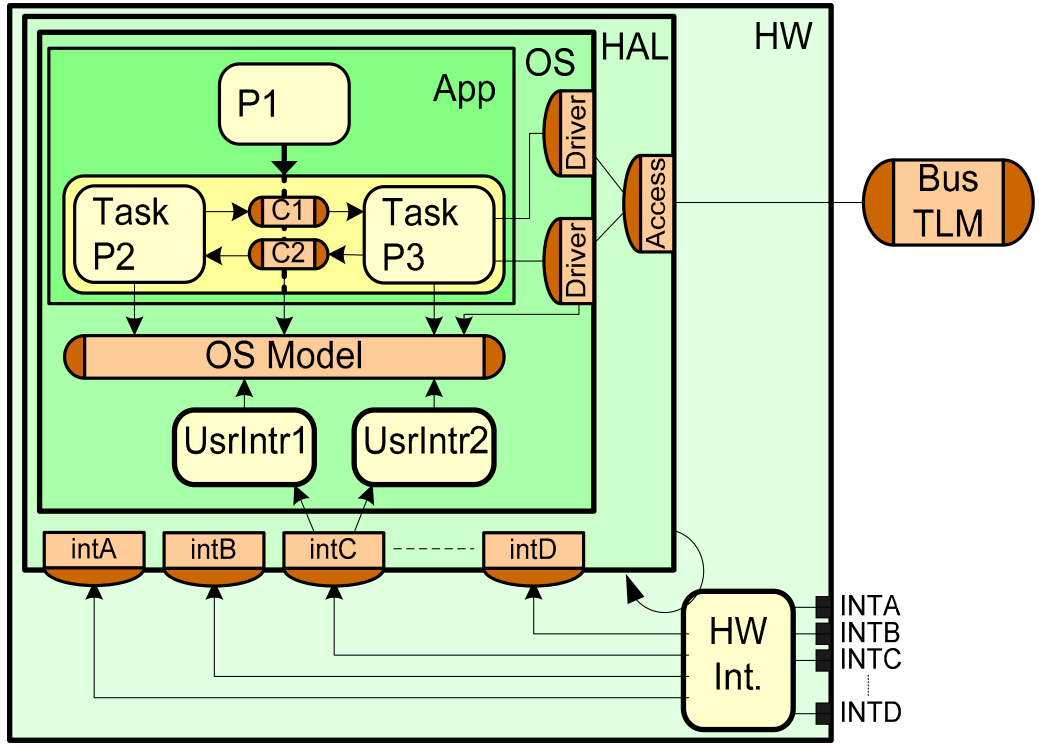
Computation Layers



- **Application model**
  - Model of Computation (MoC)
    - Process-/state-based [KPN, SDF, FSM, …]
  - Back-annotated execution timing
    - Timing granularity (basic block level)
- **Processor model**
  - Operating system
    - Middleware (MoC runtime)
    - Real-time multi-tasking (RTOS)
    - Bus drivers
  - Hardware abstraction layer (HAL)
    - Interrupt handlers
    - Media accesses
  - Processor hardware
    - Bus interfaces (I/O state machines)
    - Interrupt suspension and timing

Processor Model

- **Processor layers**
  - Application
    - Native, host-compiled C
    - Back-annotated timing
  - OS
    - OS model
    - Middleware, drivers
  - HAL
    - Firmware
  - Processor hardware
    - Bus interfaces
    - Interrupts handling & suspension



Features	
Target approx. computation timing	Appl.
Task mapping, dynamic scheduling	
Task communication, synchronization	OS
Interrupt handlers, low level SW drivers	HAL
HW interrupt handling, int. scheduling	HW-TLM
Cycle accurate communication	HW-BFM
Cycle accurate computation	BFM - ISS

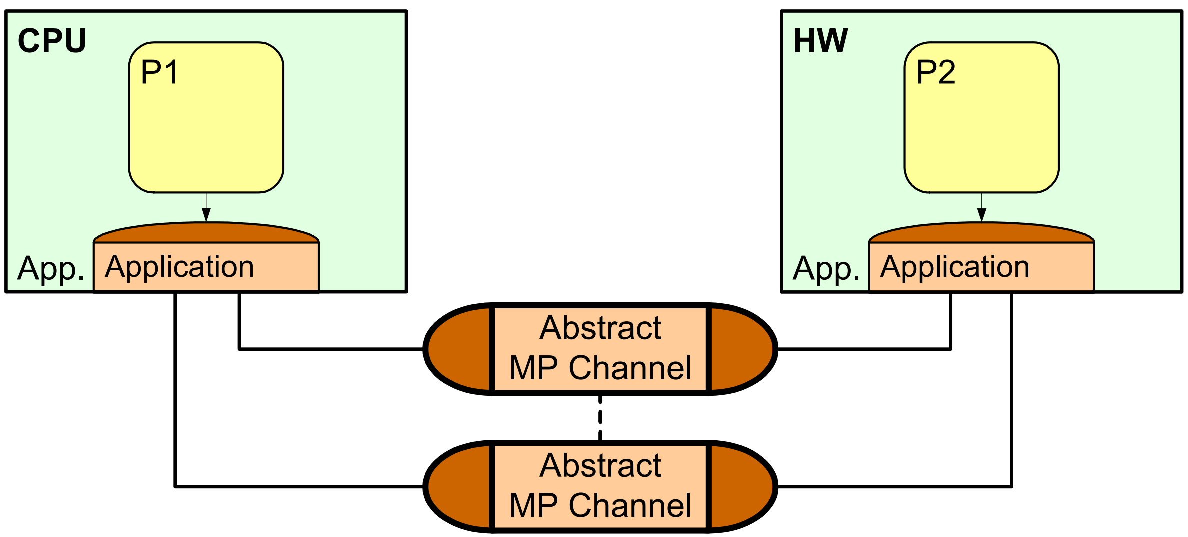
Source: G. Schirner, A. Gerstlauer, R. Doemer. "Fast and Accurate Processor Models for Efficient MPSoC Design," TODAES, to appear 2010.

Outline

- ✓ Introduction
- ✓ Component modeling
  - ✓ Communication modeling
  - ✓ Computation modeling
- **Platform modeling**
  - System models
  - Design example
- Summary and conclusions

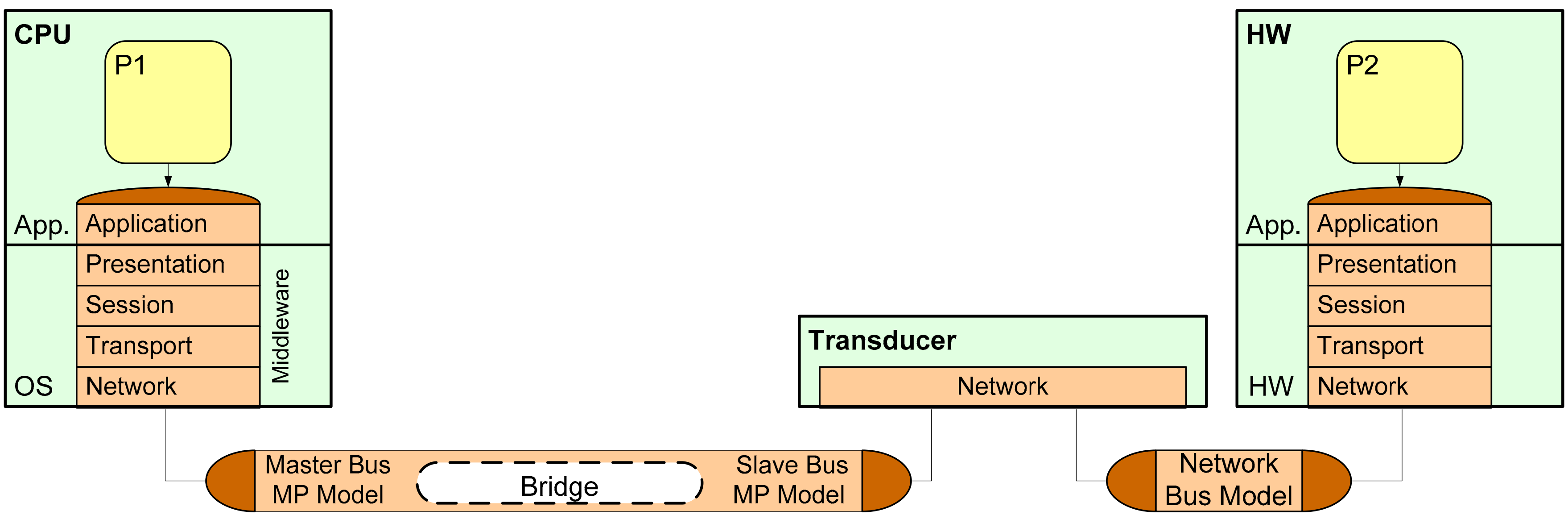
Specification Model

- **Application layers**
  - Computation
    - Processes
    - Variables
  - Communication
    - Sync./async. message-passing
    - Memory interfaces
    - Events



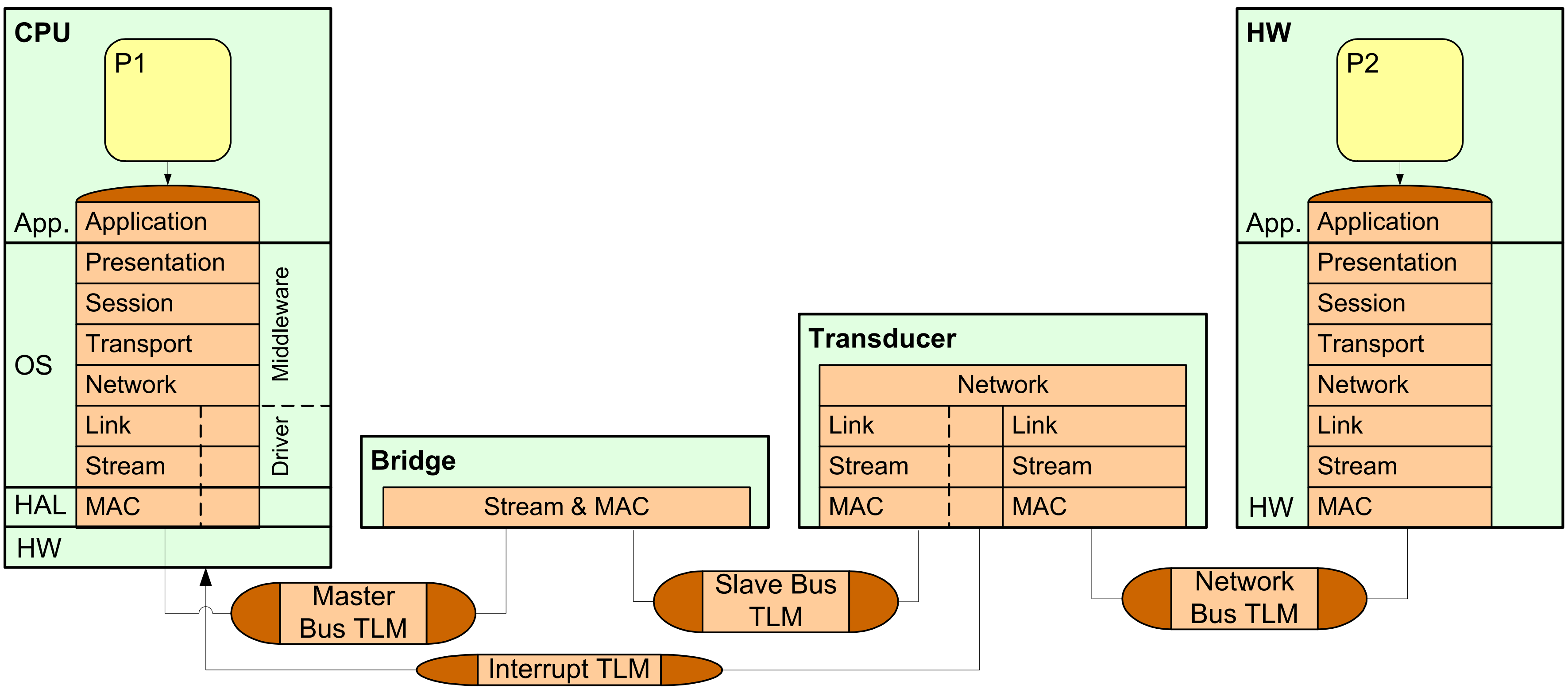
Network TLM

- **Topology of communication architecture.**
  - PEs + Memories + CEs
  - Upper protocol layers inserted into PEs/CEs
  - Communication via universal bus channels
    - Point-to-point logical links, synchronous packet transfers (data transfers)
    - Memory accesses (shared memory, memory-mapped I/O)
    - Events (control flow)



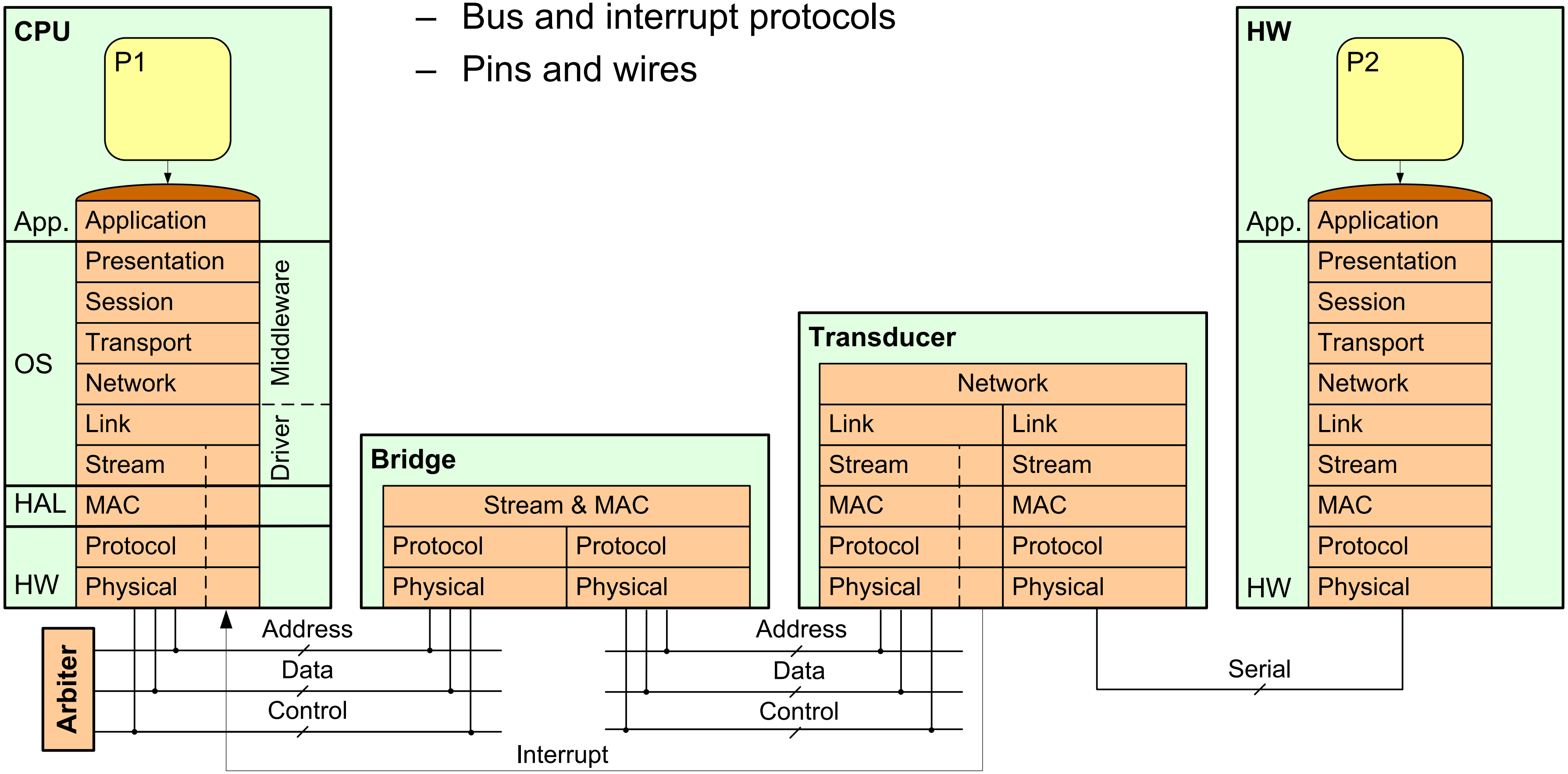
Protocol TLM

- **Abstract component & bus structure/architecture**
  - PEs + Memories + CEs + Busses
  - Communication layers down to protocol transactions
  - Communication via transaction-level bus channels
    - Bus protocol transactions (data transfers)
    - Synchronization events (interrupts)



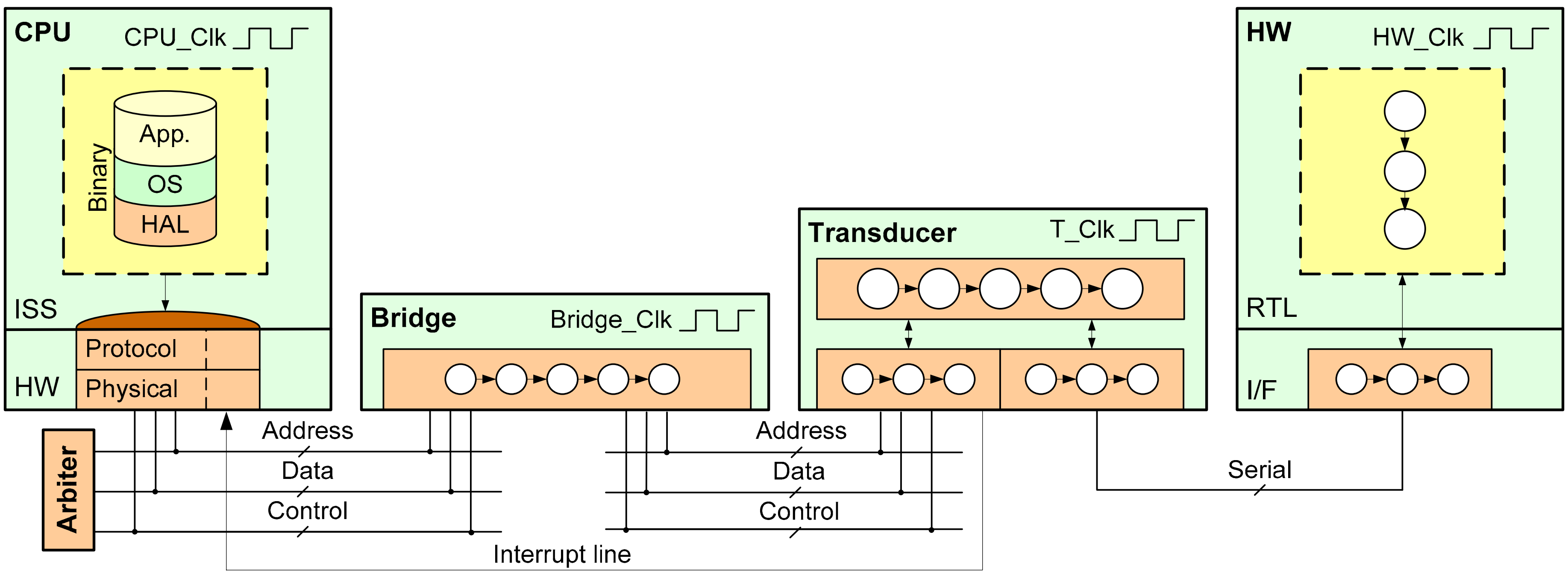
Bus Cycle-Accurate Model (BCAM)

- **Component & bus structure/architecture**
  - PEs + Memories + CEs + Busses (+ arbiters/muxes)
  - Pin-accurate, bus-functional components
  - Pin- and cycle-accurate communication
    - Bus and interrupt protocols
    - Pins and wires

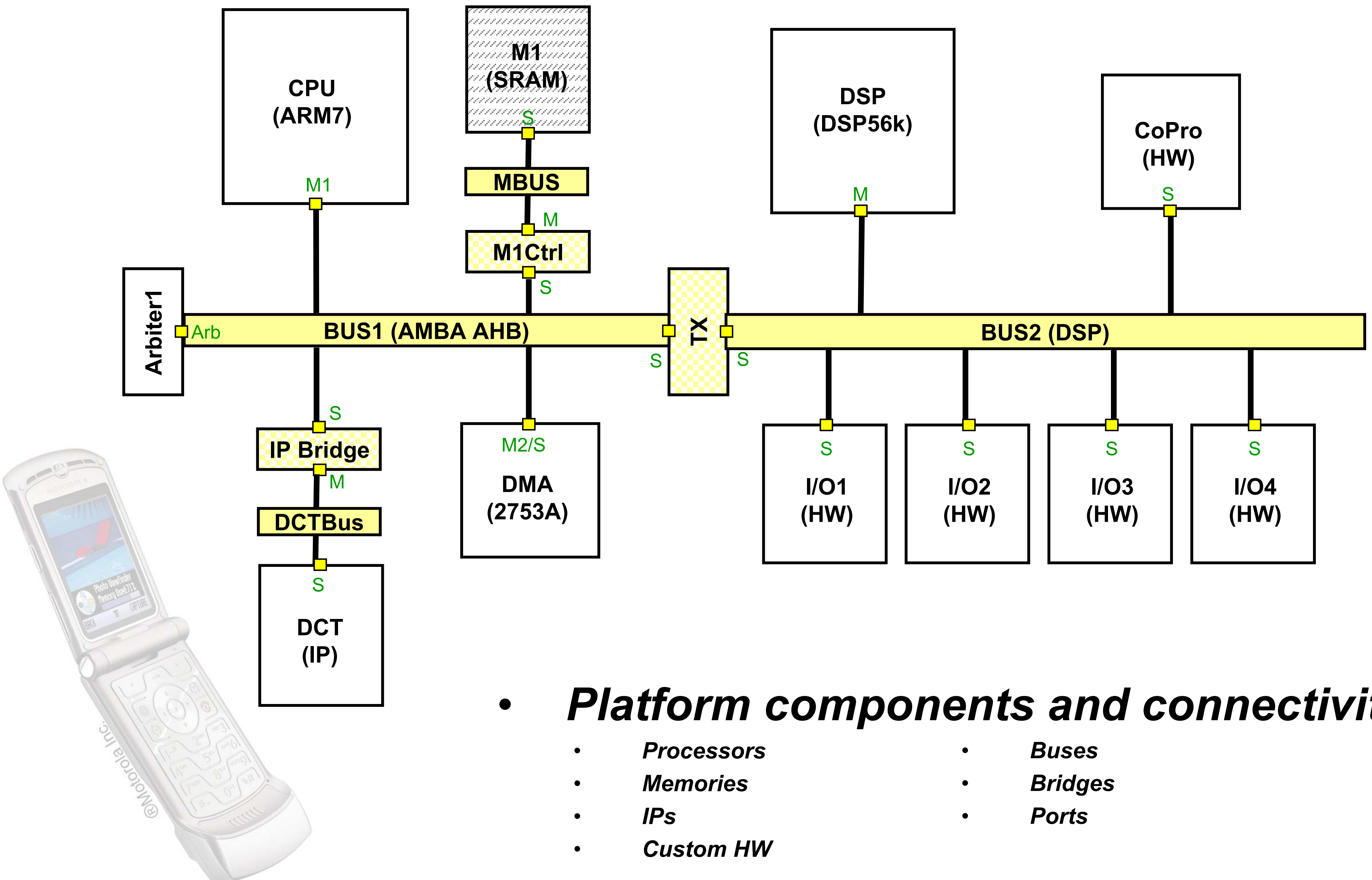


Cycle-Accurate Model (CAM)

- **Component & bus implemenation**
  - PEs + Memories + CEs + Busses
  - Cycle-accurate components
    - Instruction-set simulators (ISS) runing final target binaries
    - RTL hardware models
    - Bus protocol state machines

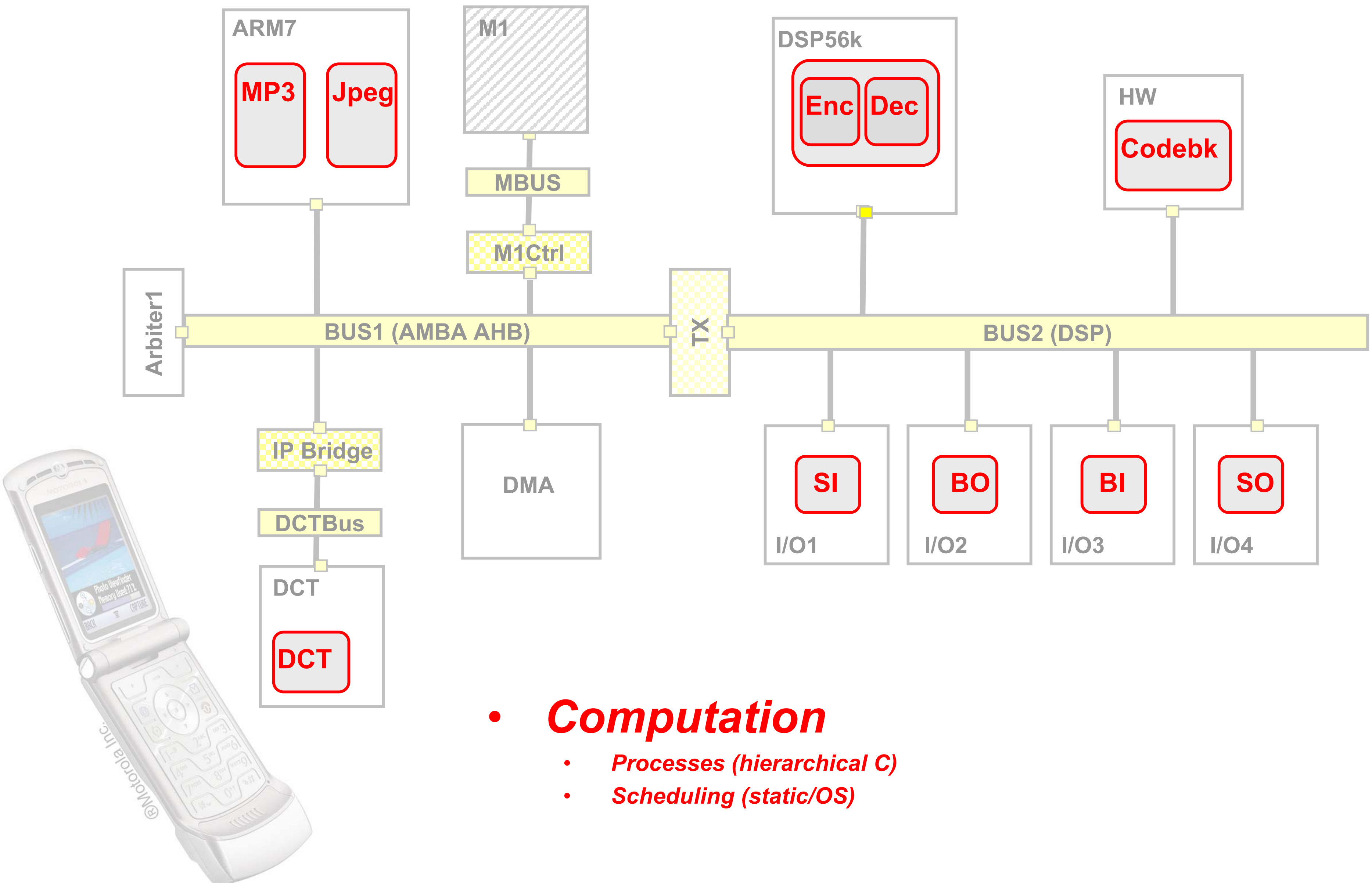


Cellphone Example: Specification



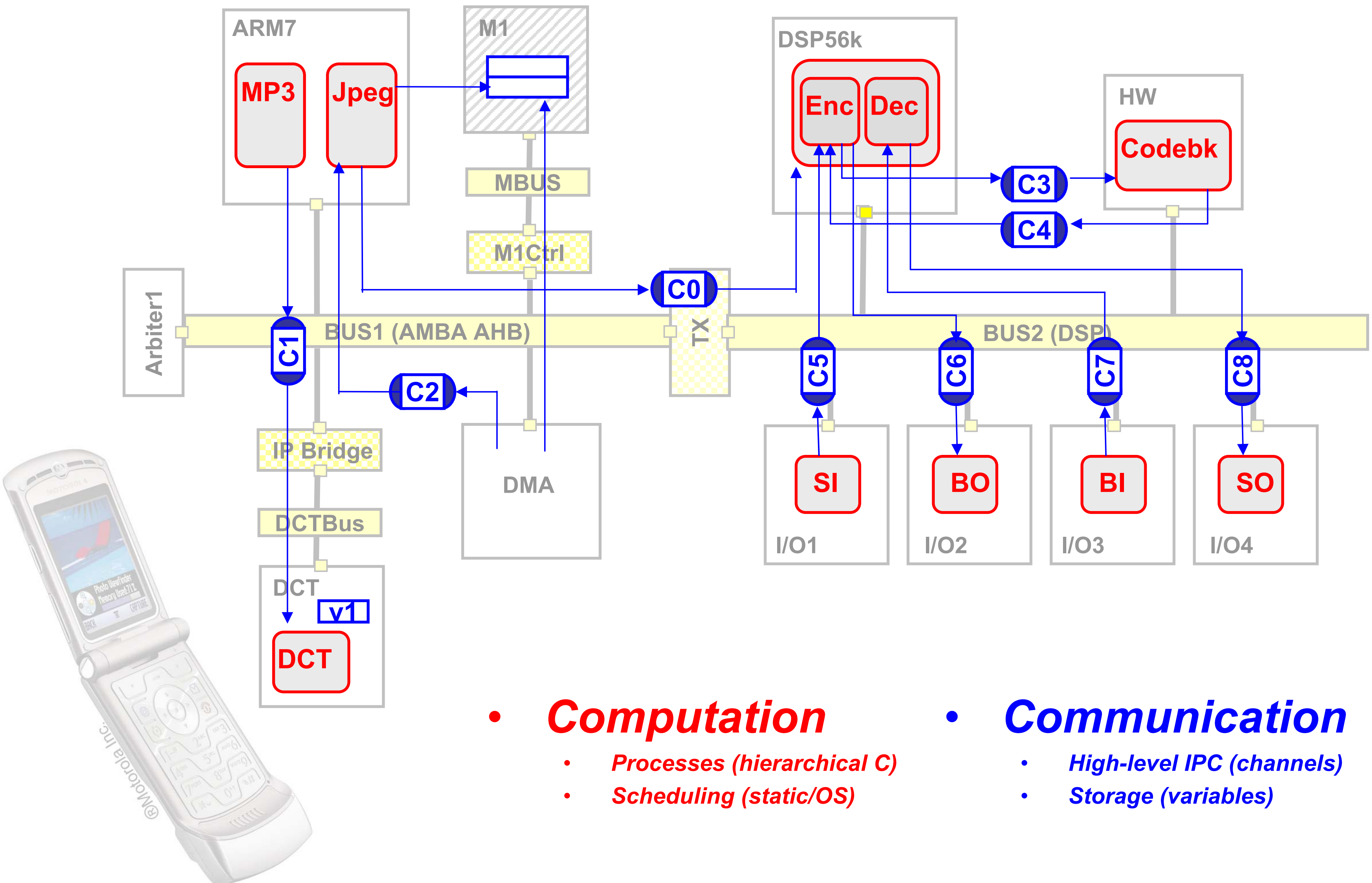
- ***Platform components and connectivity***
  - ***Processors***
  - ***Memories***
  - ***IPs***
  - ***Custom HW***
  - ***Buses***
  - ***Bridges***
  - ***Ports***

Cellphone Example: Specification



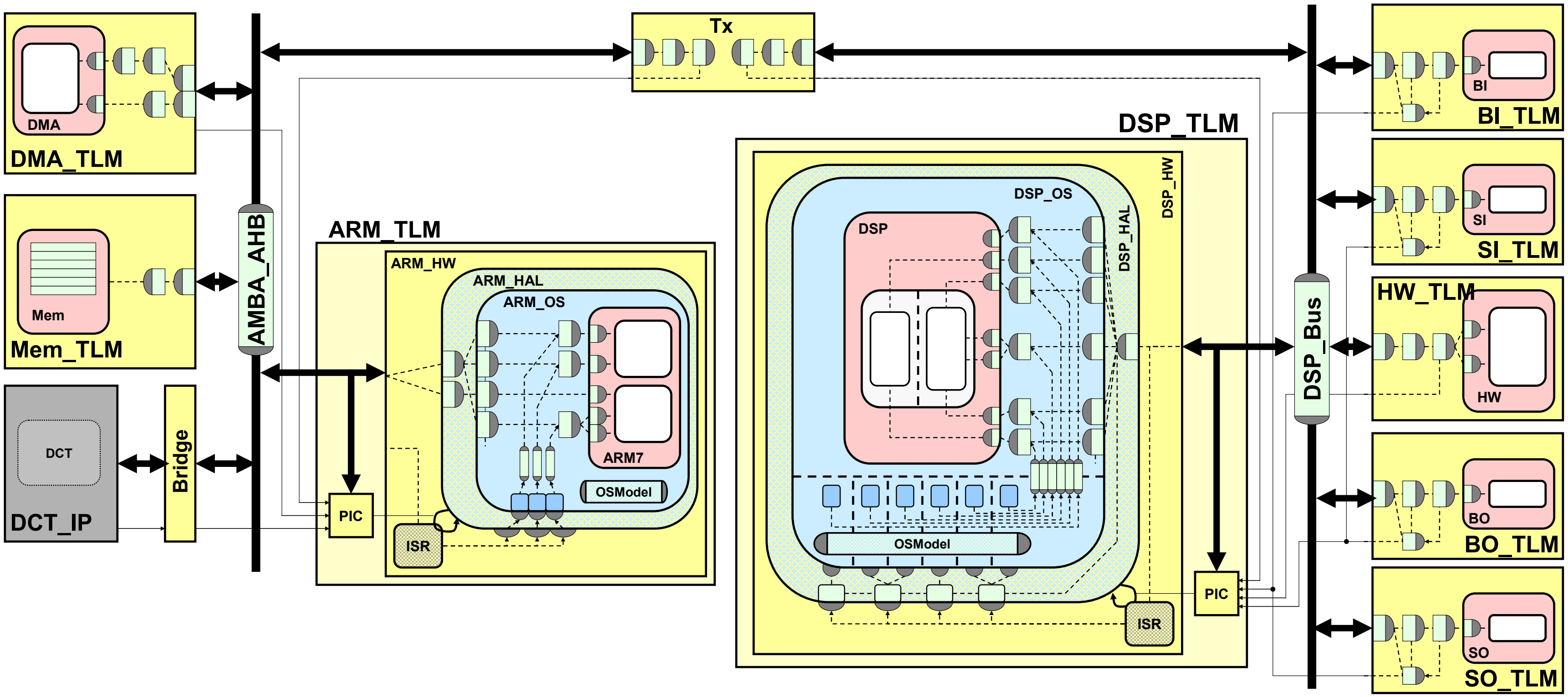
- ***Computation***
  - ***Processes (hierarchical C)***
  - ***Scheduling (static/OS)***

Cellphone Example: Specification



- **Computation**
  - *Processes (hierarchical C)*
  - *Scheduling (static/OS)*
- **Communication**
  - *High-level IPC (channels)*
  - *Storage (variables)*

Cellphone Example: P-TLM



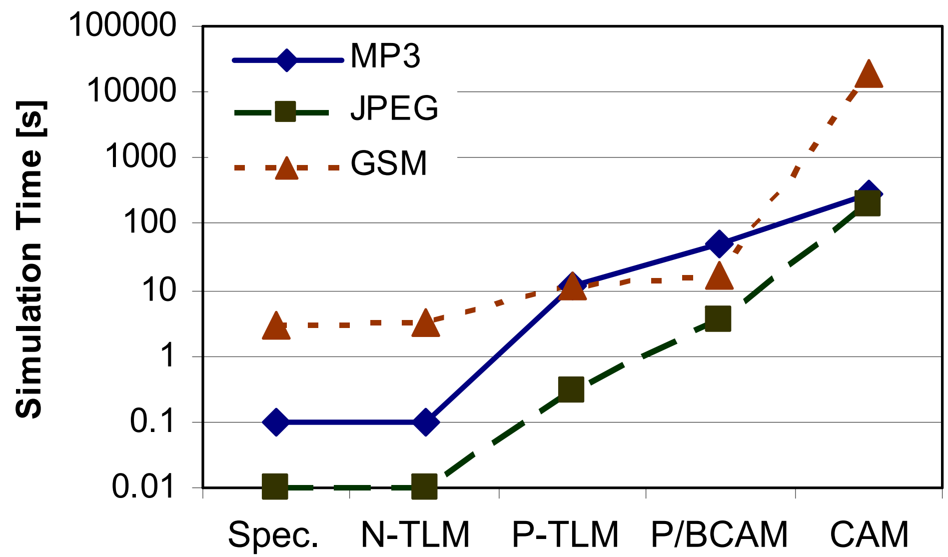
- **Implementation synthesis in backend tools**
  - Interface and high-level synthesis on hardware side
  - Firmware, RTOS and C synthesis on the software side

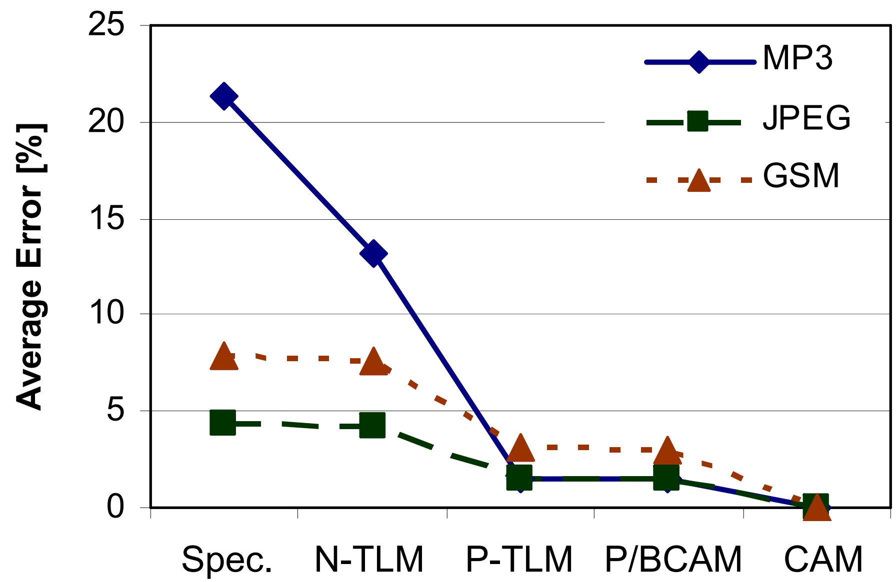
Cellphone Example: Subsystem Results

- **MP3 decoder on ARM**
  - 55 MP3 frames
- **JPEG encoder on ARM**
  - 30 116x96 pictures
- **GSM vocoder on DSP**
  - 163 speech frames encoded/decoded

➢ **Speed**
  - 2000 MIPS peak
  - 600 MIPS sustained

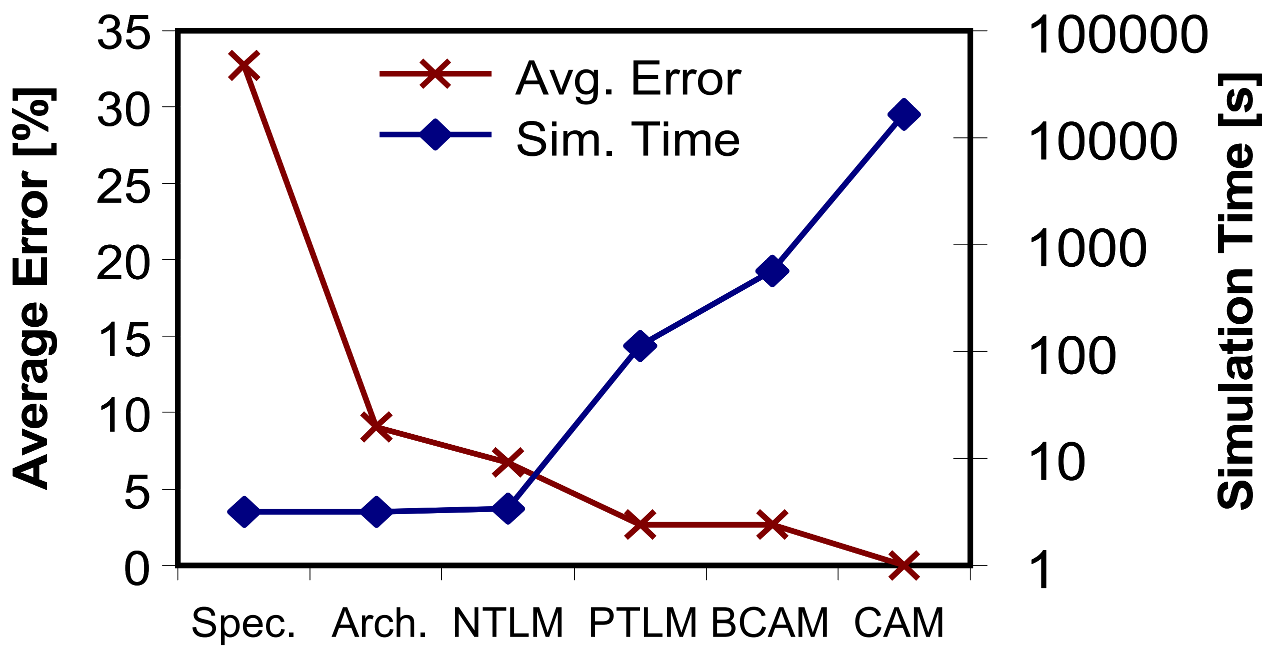
➢ **Accuracy**
  - <3% average frame timing error





Cellphone Example: Full-System Results

- **Experimental setup**
  - 1.5 second MP3
  - 640x480 picture
  - 1.5 speech GSM
  - ➢ 3s / 300M ARM cycles
- **Speed**
  - 2000 MIPS peak
  - 300-600 MIPS sustained
- **Accuracy**
  - <3% error



➢ **Prototyping and exploration with 100% fidelity**

Summary and Conclusions

- **Modeling of system computation and communication**
  - From specification to implementation
    - Data & timing granularity
  - ➢ Flow of well-defined, layer-based models
    - ➢ Design automation for model refinement and synthesis
- **Various level of abstraction, accuracy and speed**
  - Design space exploration and pruning
    - Design quality evaluation through model simulation or analysis
  - ➢ Transaction-level models (TLMs)
    - ➢ Fast and accurate virtual prototyping at varying levels

➢ **General modeling techniques for accuracy and/or speed**
  - ➢ Temporal decoupling [SystemC TLM 2.0]
  - ➢ Prediction with correction [ROM]