

From Platform-based Design to Transaction Level Modeling

Franco Fummi



UNIVERSITÀ
di VERONA
Dipartimento
di **INFORMATICA**

Version 1.1

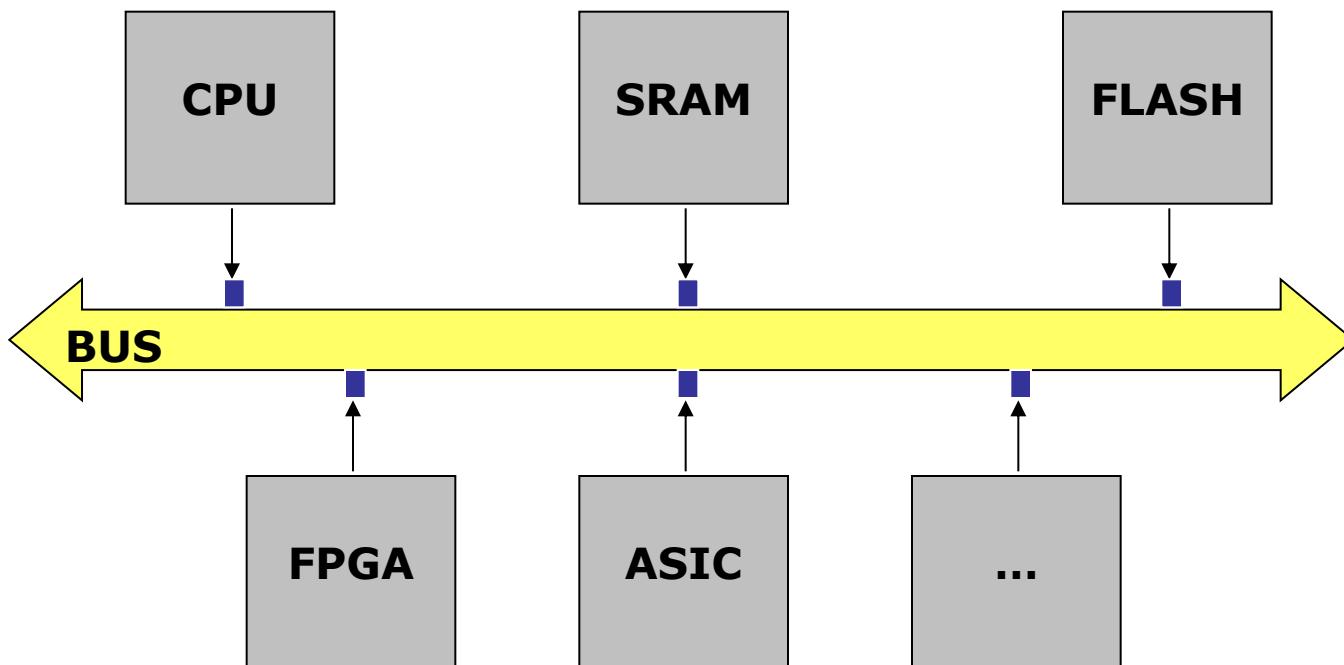
Contents

- Platform Based Design
- Transaction Level Modeling (TLM)
 - Key concepts
 - First standardization
- TLM-based Design Flows
 - Refinement and abstraction steps
 - IP reuse
 - Co-design
- A complete example

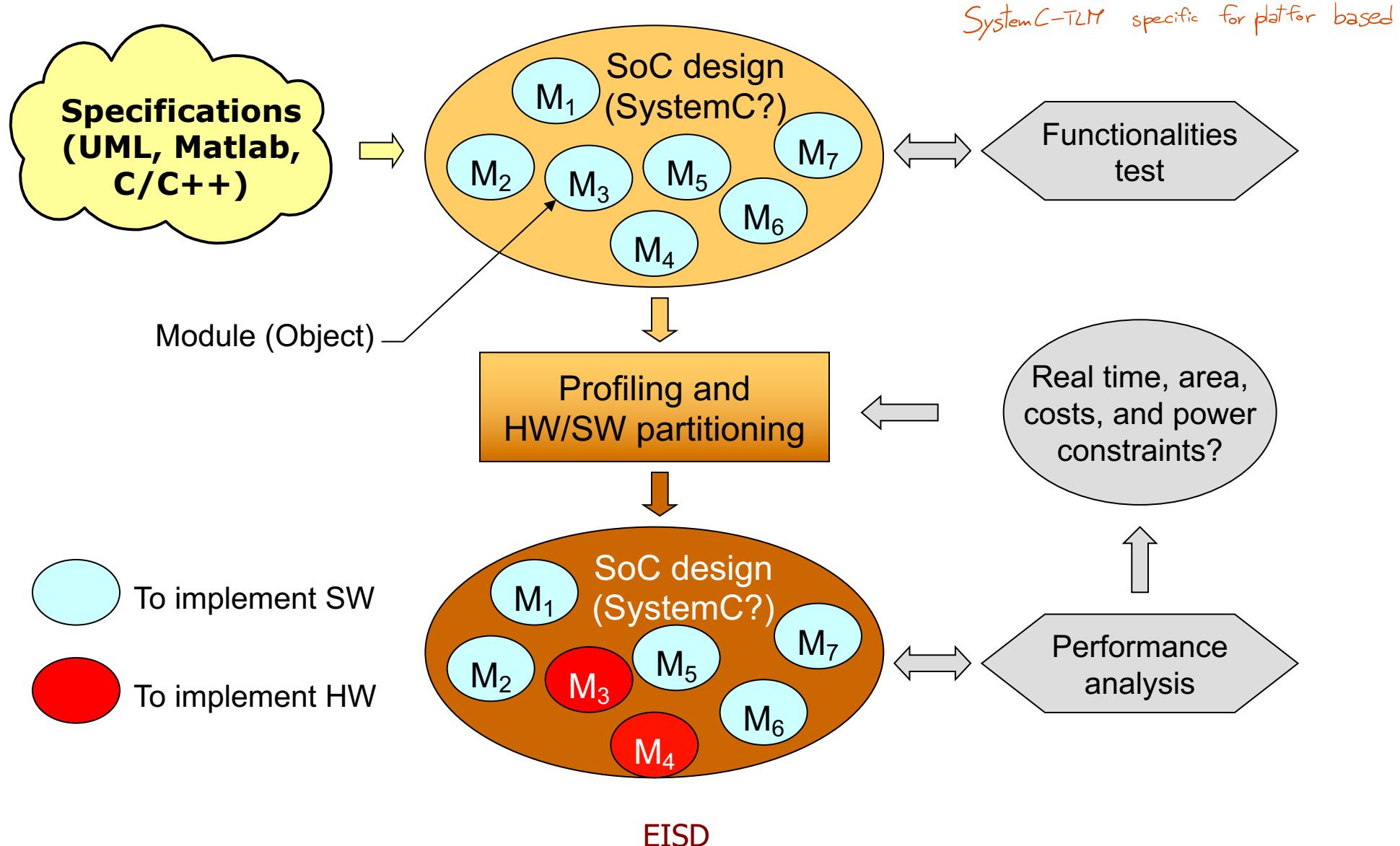
Platform Based Design

Can be applied to:
 • SOC
 • SoB = System on a Board

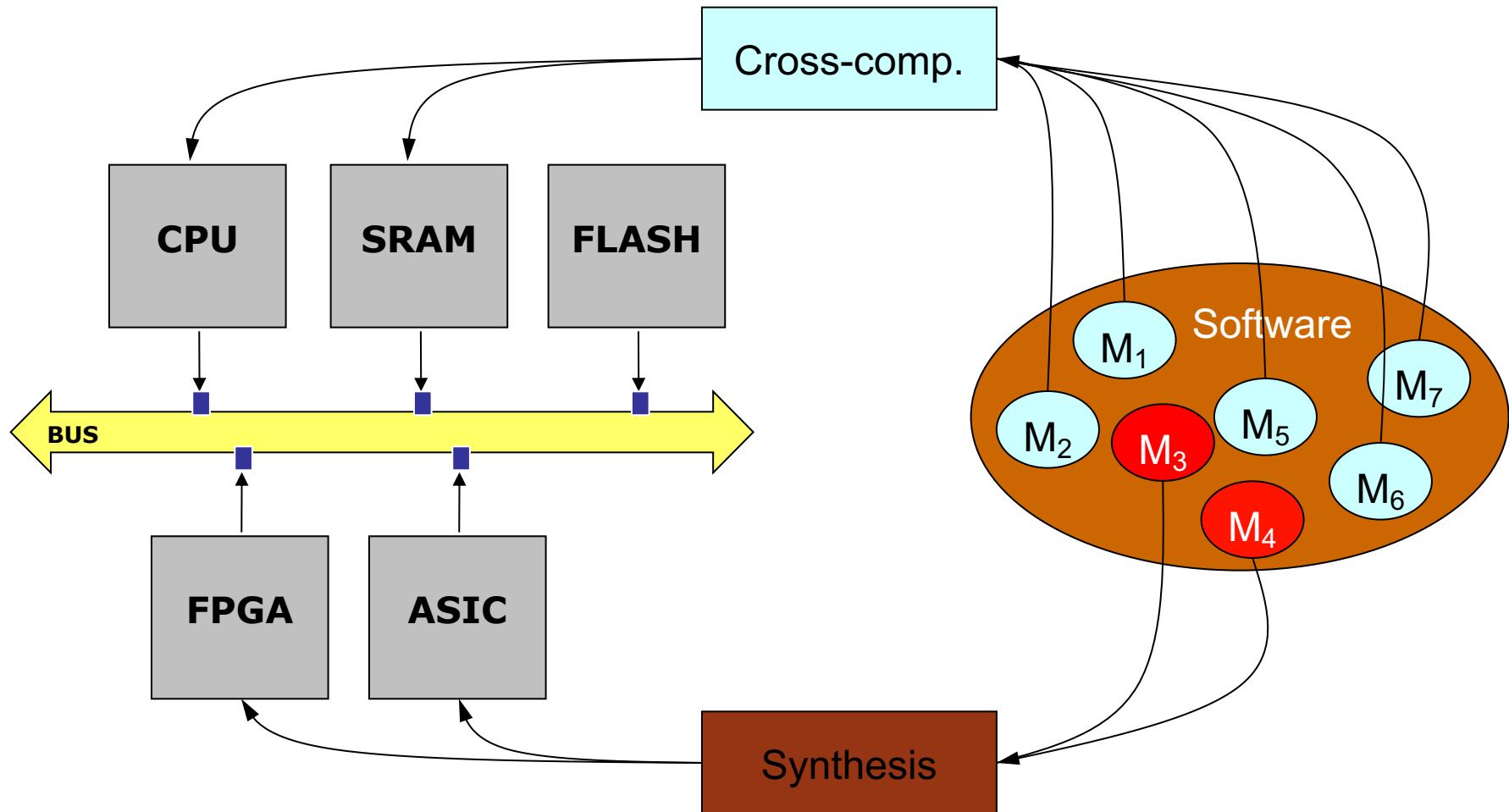
- **Definition:** *platform-based design is the creation of a stable microprocessor-based architecture that can be rapidly extended, customized for a range of applications and delivered to customers for quick deployment. (J.M. Chateau – STMicroelectronics)*



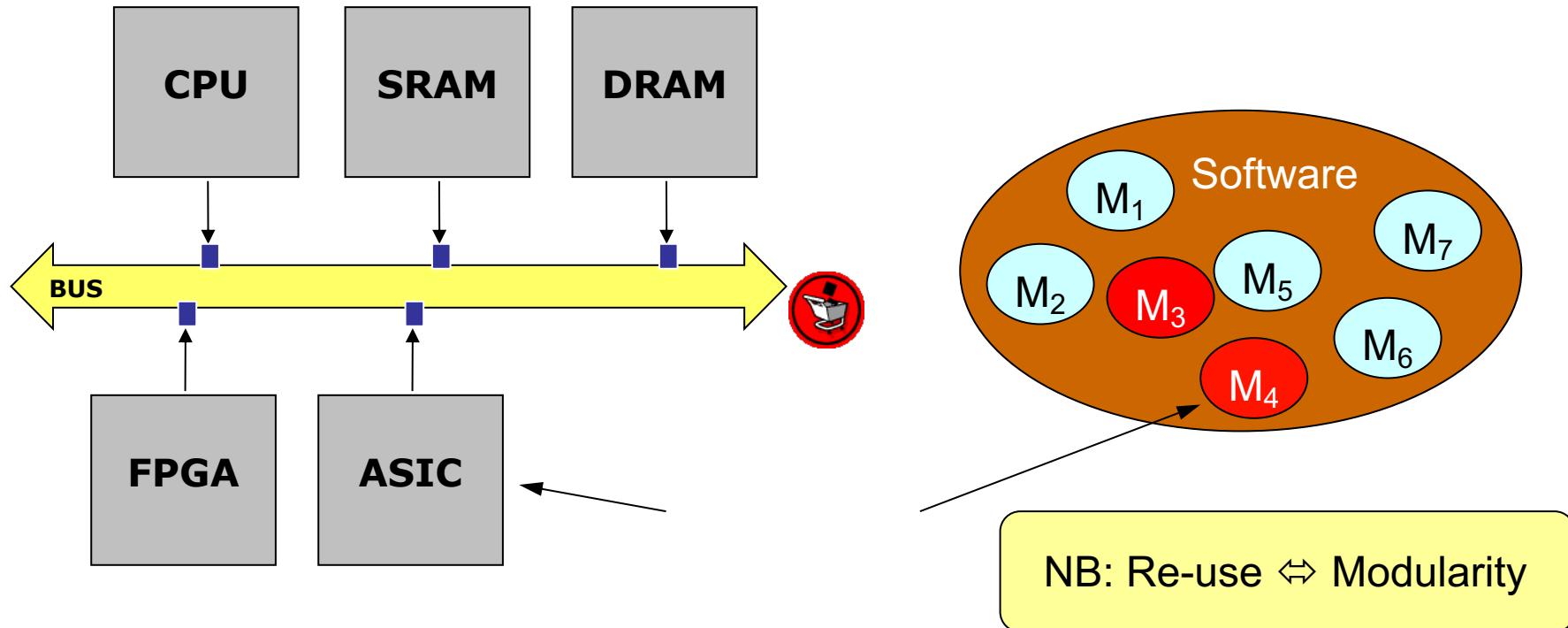
Platform based design flow



Platform based design flow



Platform based design: IPs re-use



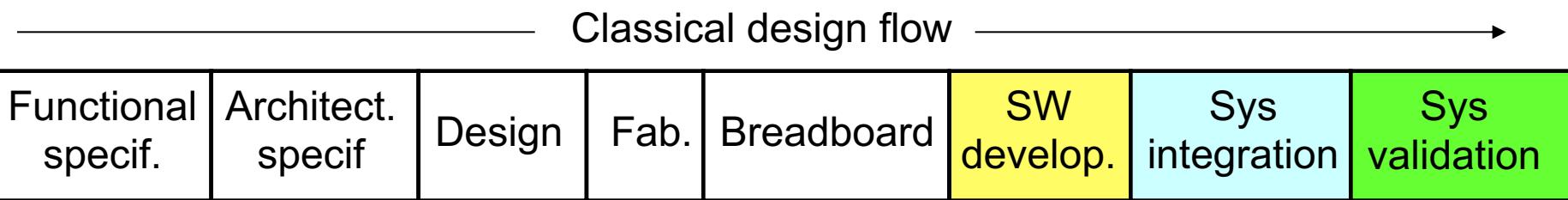
i.e., Digital Signal Processor (DSP)
MPEG, Coder, Decoder, Filters, etc.

Platform based design: summary

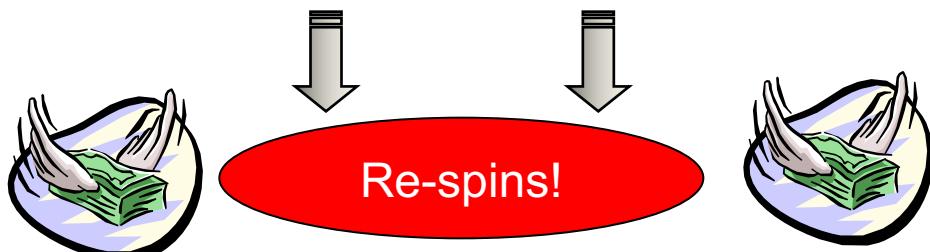
- High-level description for functional verification
- Timing estimation for performance analysis
- Hierarchy for HW/SW partitioning
- Modularity for IPs re-use and/or refinement
- Simulation speed vs. implementation details
- Synthesizable descriptions for HW modules
 - **Transaction Level Modeling (TLM)**
 - **Virtual Platform (VP)**

Mix of these two => Platform based design

TLM: Motivations

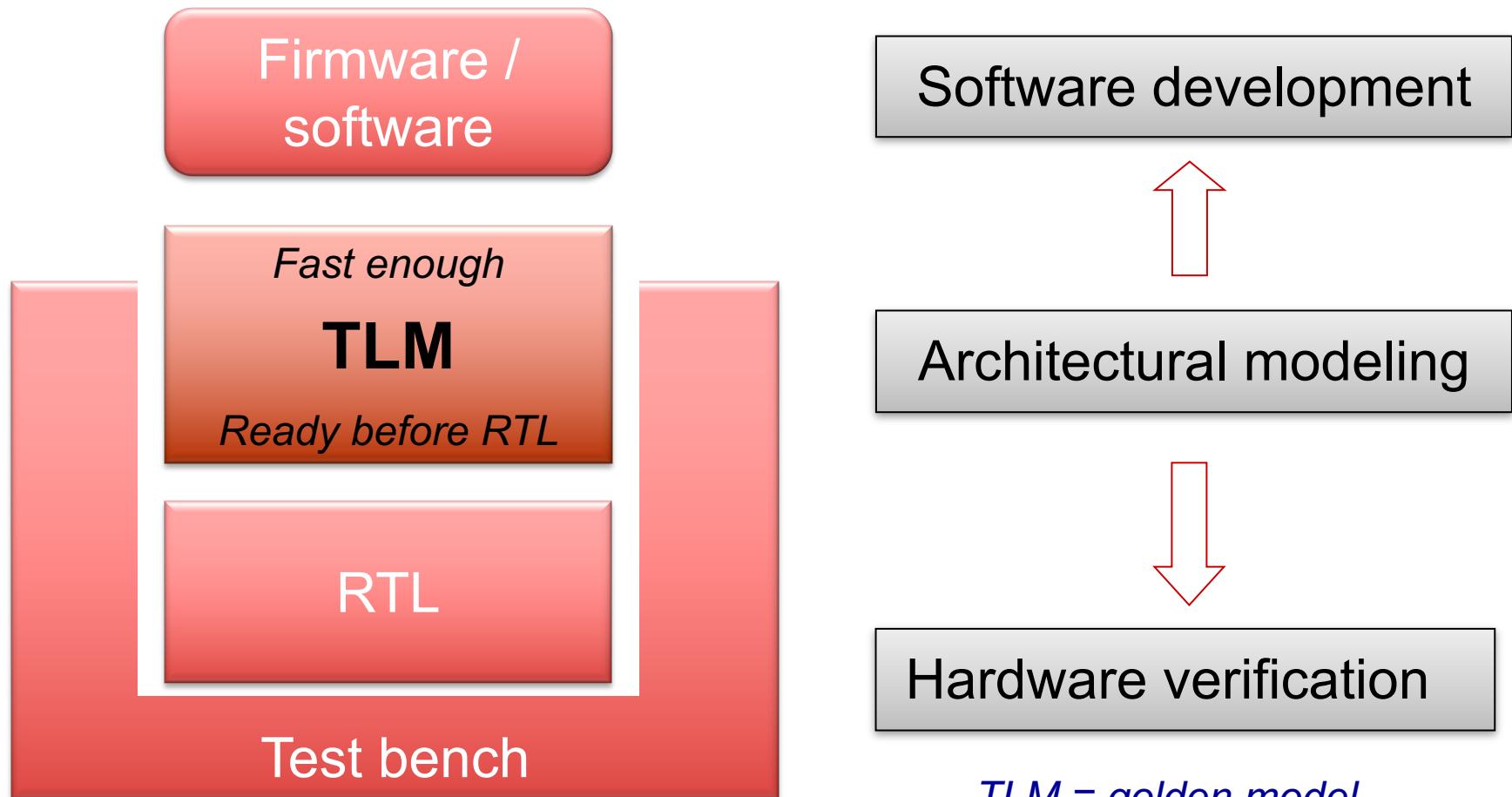


Functional specifications failures?
Performance expectations missing?



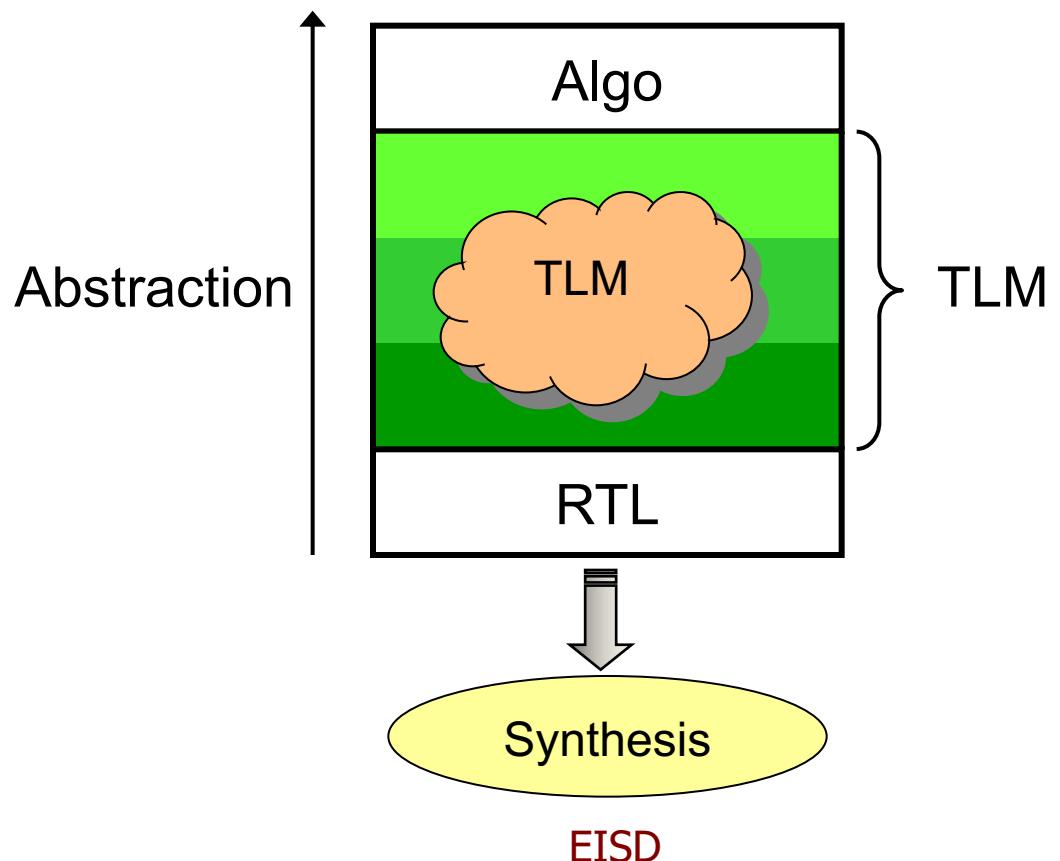
Main reason:
lacking of a concretely usable view
of the complete system before the
tape-out phase!

Reasons for Using TLM



TLM: Definition

- Transaction Level is the new design and verification abstraction above RTL



TLM: Abstraction Levels

Level	Use	Features	Abstraction
PV (Programmer's View) (Loosely-timed)	<ul style="list-style-type: none"> Executable specification Proof of concepts 	<ul style="list-style-type: none"> Event-driven Untimed 	<ul style="list-style-type: none"> Data types Time Resource sharing
PVT (Programmer's View + Time) (Approximately-timed)	<ul style="list-style-type: none"> HW/SW partitioning Performance estimation 	<ul style="list-style-type: none"> Event-driven with time estimation 	<ul style="list-style-type: none"> Clocks Protocols
CA (Cycle Accurate)	<ul style="list-style-type: none"> Detailed modeling Cycle accurate testbenches 	<ul style="list-style-type: none"> Cycle accurate 	<ul style="list-style-type: none"> Wires Registers
RTL	<ul style="list-style-type: none"> Signal level modeling for synthesis 	<ul style="list-style-type: none"> Pin accurate 	<ul style="list-style-type: none"> Gates Delays

TLM: Advantages

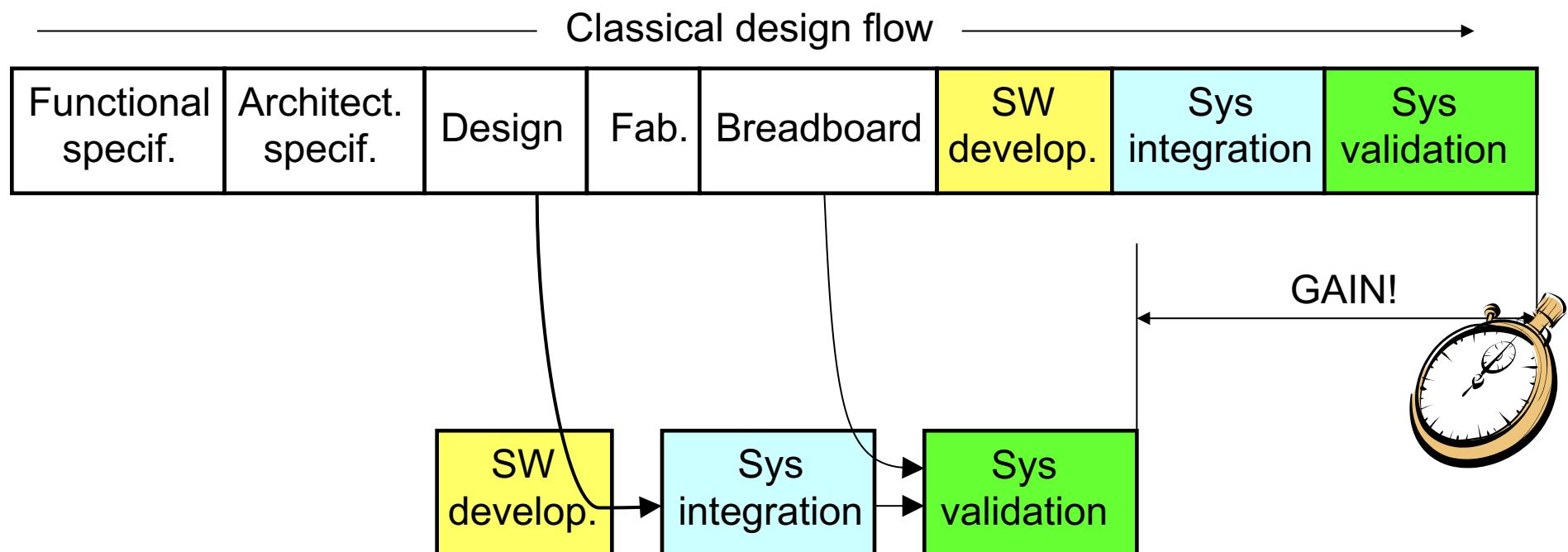
- Implementation details are abstracted while preserving the behavioral aspects of the system
 - this allows a faster simulation (up to 1,000x) than at RTL
- System level design exploration and verification are simplified
 - IP components and buses can be modified and replaced in an easier way than at RTL
- An early platform for SW development can be quickly developed

TLM: Typical Use Cases

- Represents key architectural components of hardware platform
- Architectural exploration, performance modeling
- Software execution on virtual model of hardware platform
- Golden model for hardware functional verification
- Available before RTL
- Simulates much faster than RTL

TLM: Main Advantage

- Enabling Software development to start very early in the design flow



TLM: Modeling Comparison

- More emphasis on the data transfer functionality
 - less on their implementation details at the early design stage

RTL

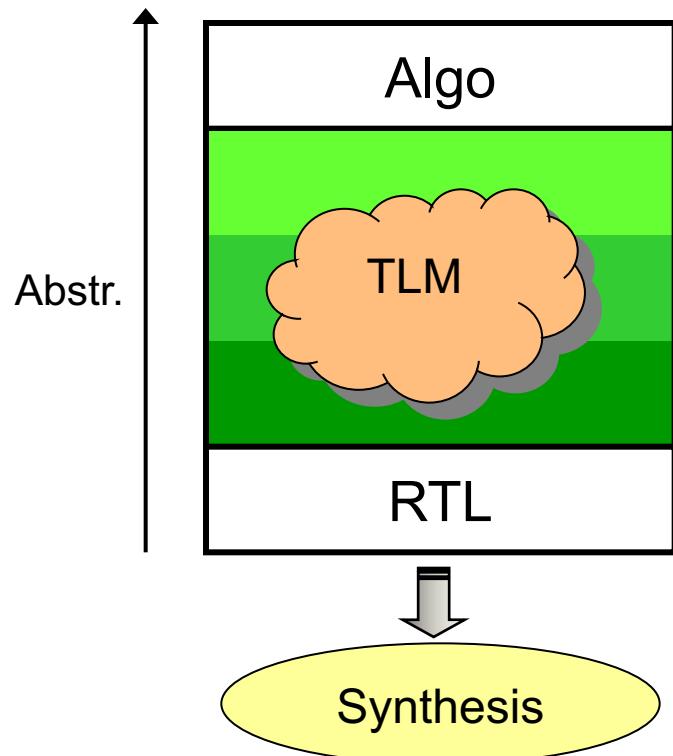
TLM

```

process(clock)
IF (clock'event and clock = '1')
THEN
CASE fsm_state IS:
  WHEN s0 =>
    request_port <= '1';
    fsm_state := s1;
  WHEN s1 =>
    IF (grant_port = '1')
      THEN
        fsm_state := s2;
  WHEN s2 =>
    data_port <= data;
    addr_port <= addr;
...
  
```


write (data, addr);

TLM: Design Languages

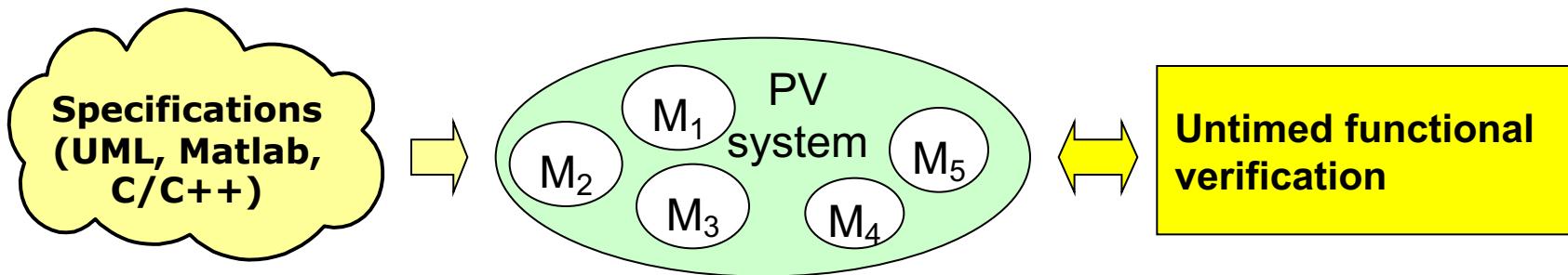


Simulink, C/C++	SystemC	VHDL, Verilog	System Verilog
★★★★★	★★★★★	--	★
--	★★★★★	--	(★★) ↪
--	★★★	★★★★★	★★★★★

⇒ Extension of Verilog

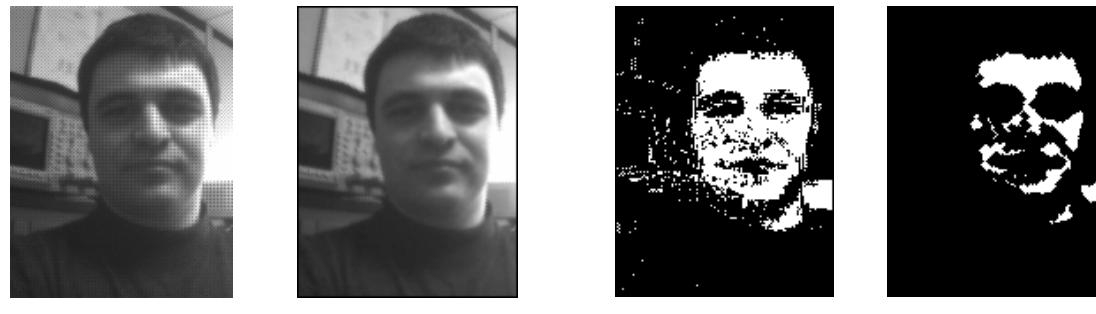
AN EXAMPLE: THE FACE RECOGNITION

TLM-based Design Flow



Example: Face recognition system

(STMicroelectronics)

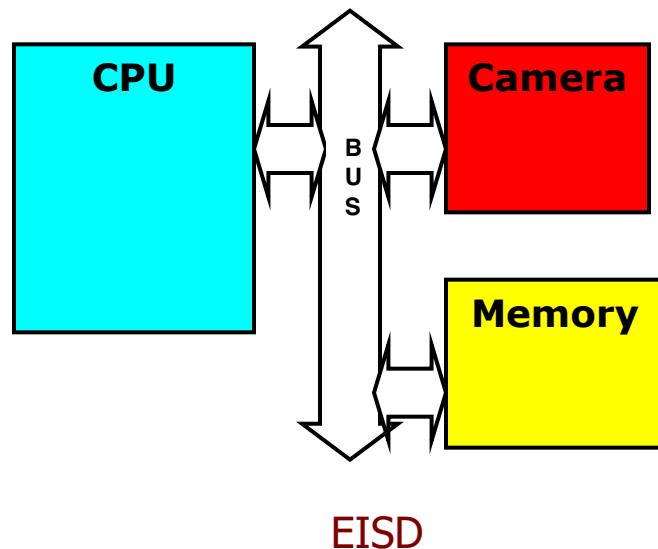


Captured Image

1st step

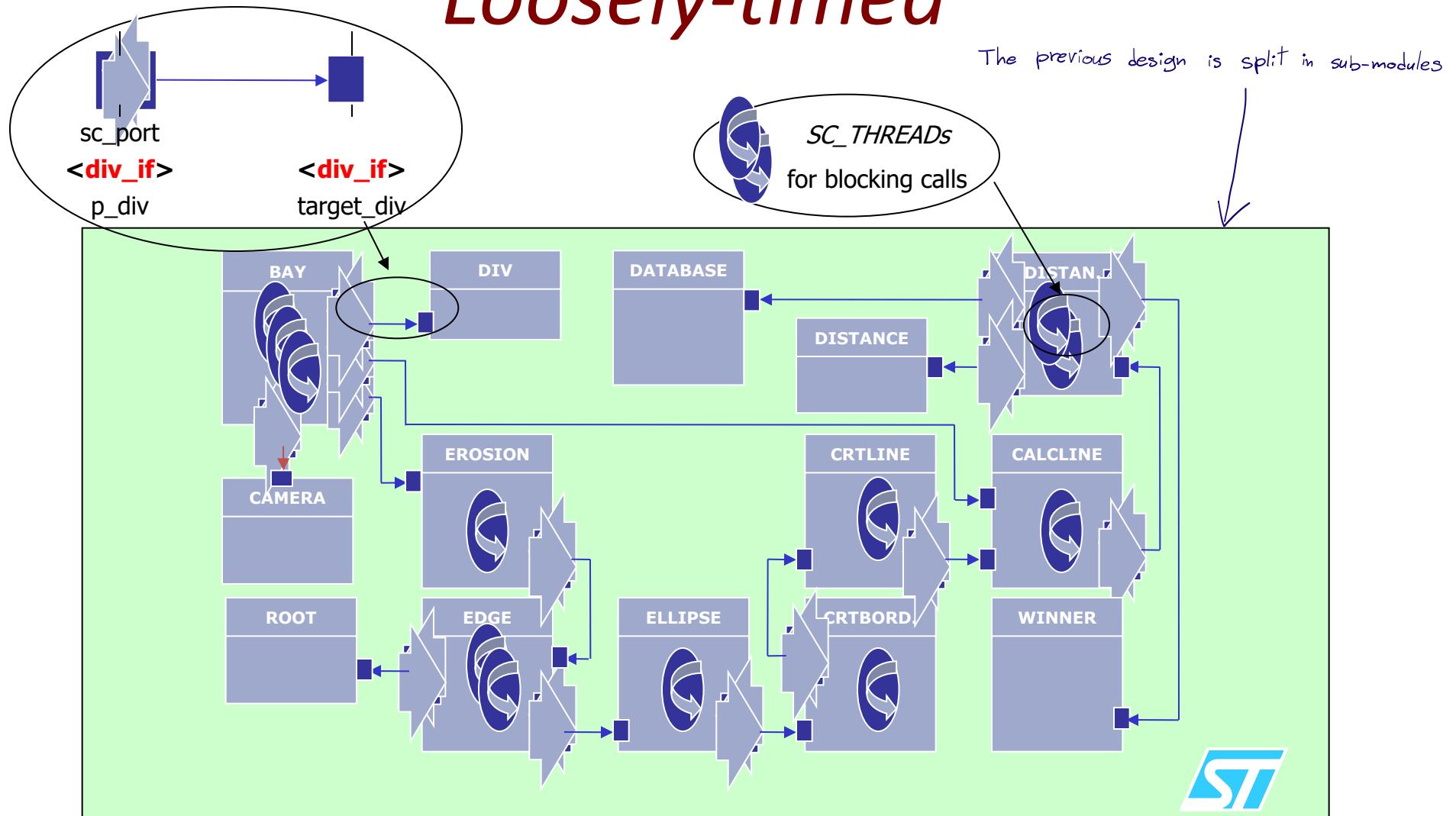
2nd step

3rd step

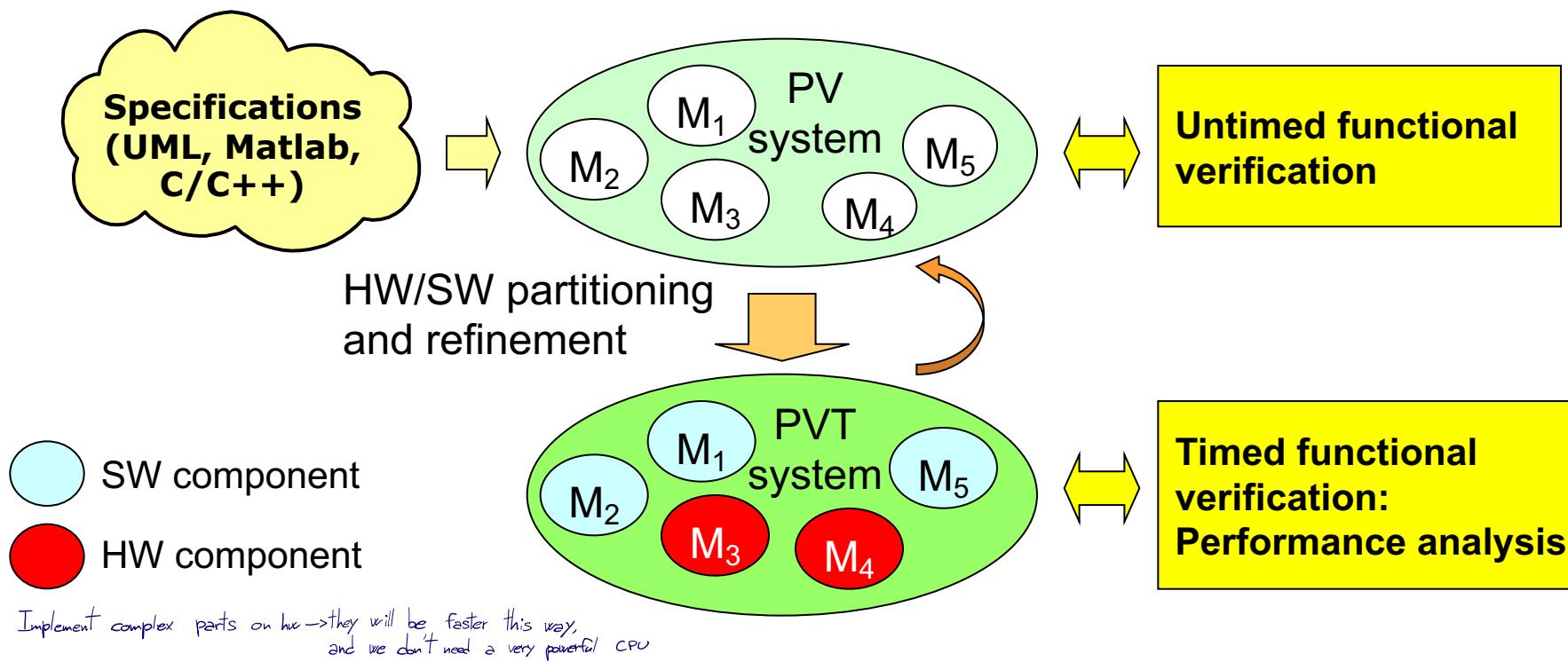


Face recognition system: PV level

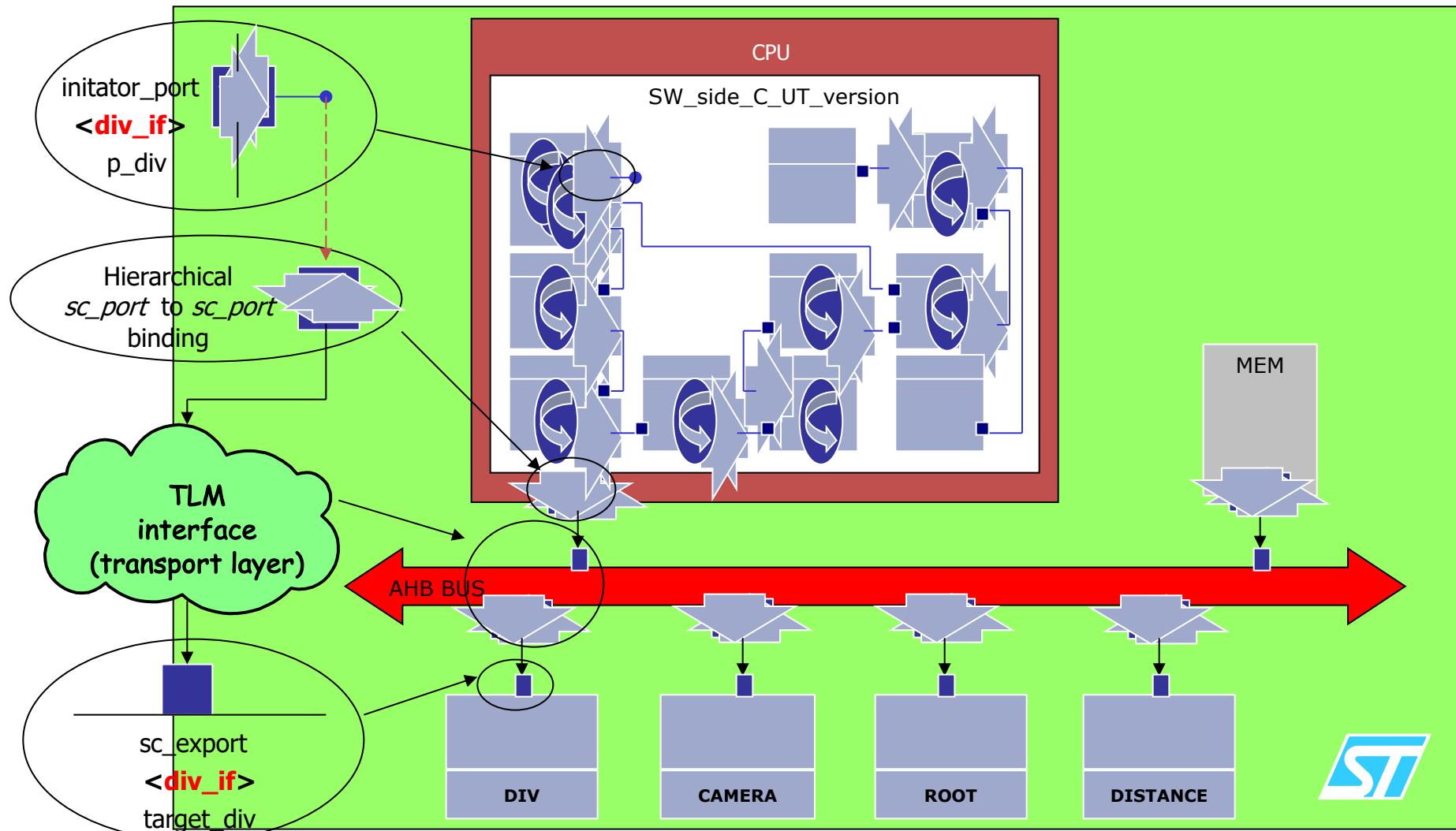
Loosely-timed



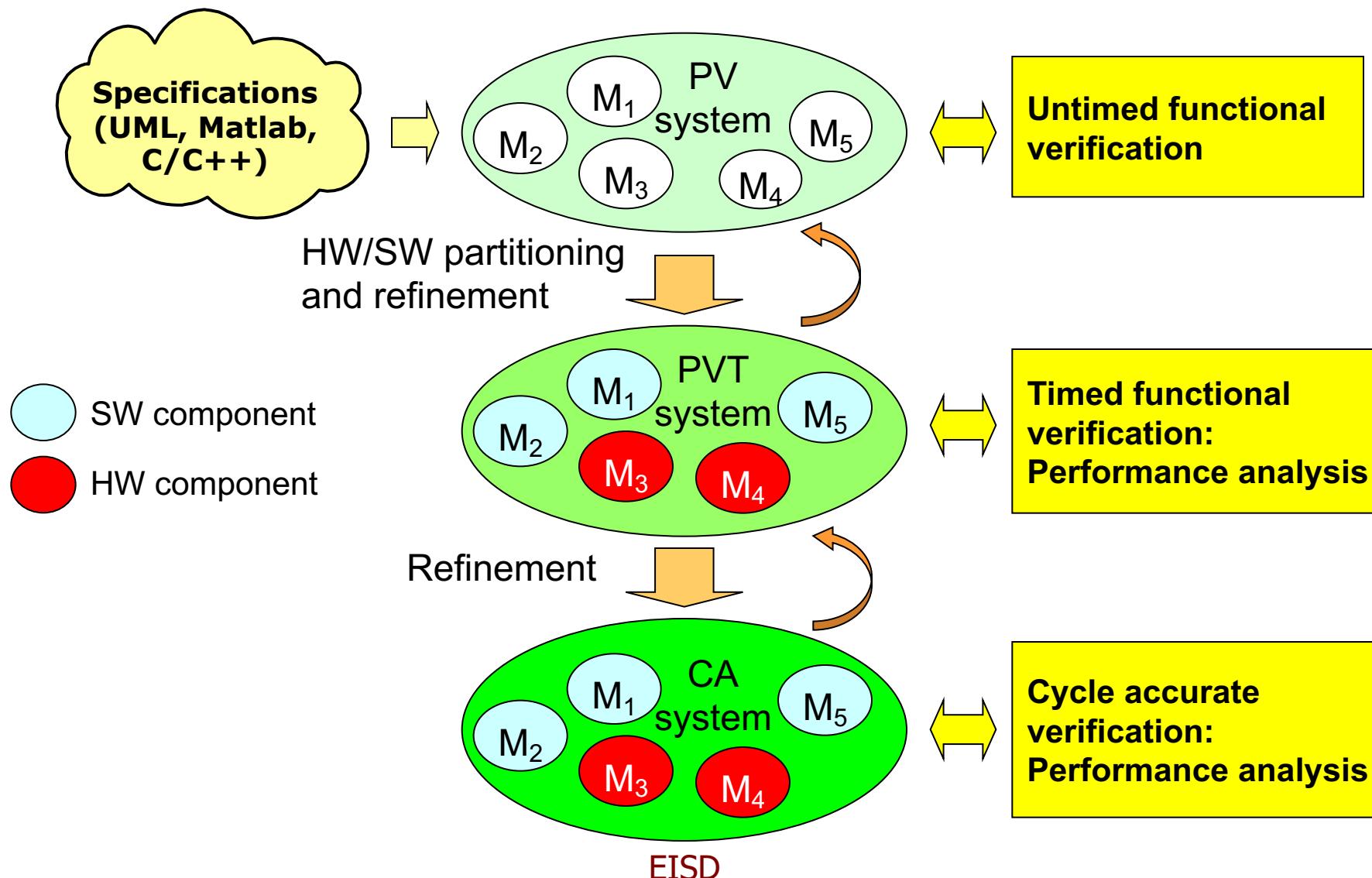
TLM-based Design Flow



PVT level: *Approximately-timed*



TLM-based Design Flow



Cycle-Accurate

