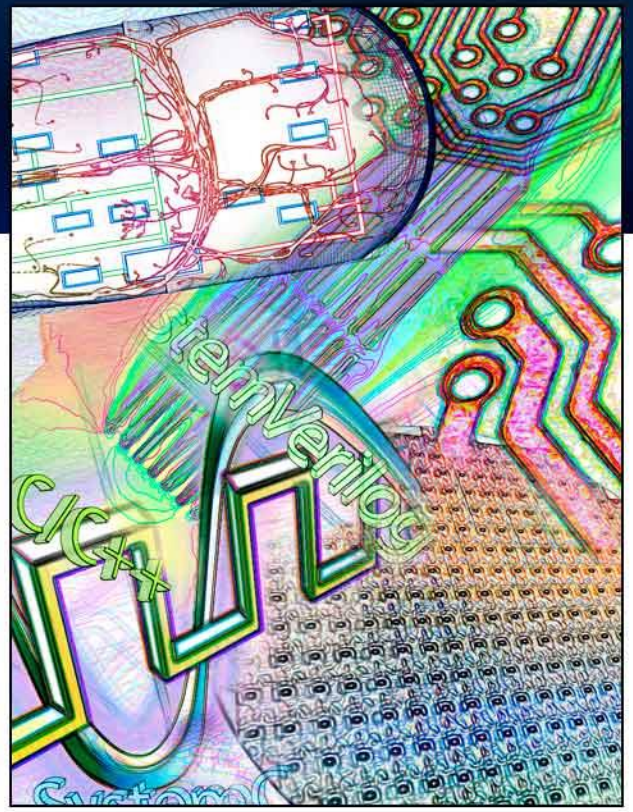



# The Veloce Emulator

## and its Use for Verification and System Integration of Complex Multi-node SOC Computing System

Laurent VUILLEMIN

Platform Compile Software Manager

Emulation Division

Mentor Graphics®

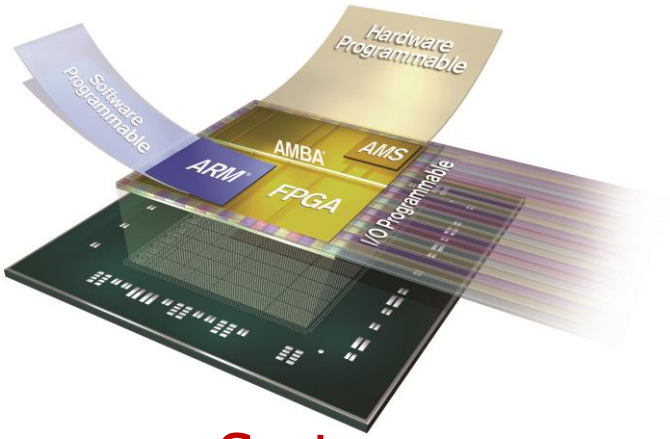
# Agenda

- What is Emulation
- Use models
- Veloce Architecture Overview
- Veloce Software
- Practical : Use Veloce to verify Veloce

# Agenda

- What is Emulation
- Use models
- Veloce Architecture Overview
- Veloce Software
- Practical : Use Veloce to verify Veloce

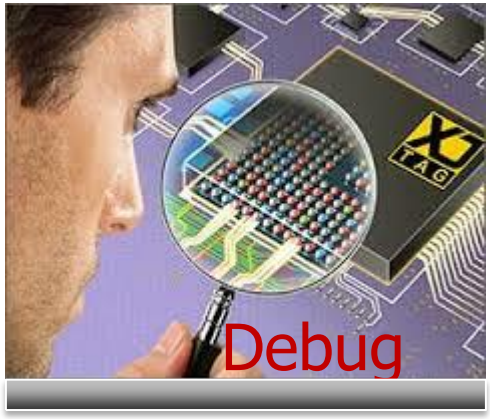
# Verification Challenges




Systems

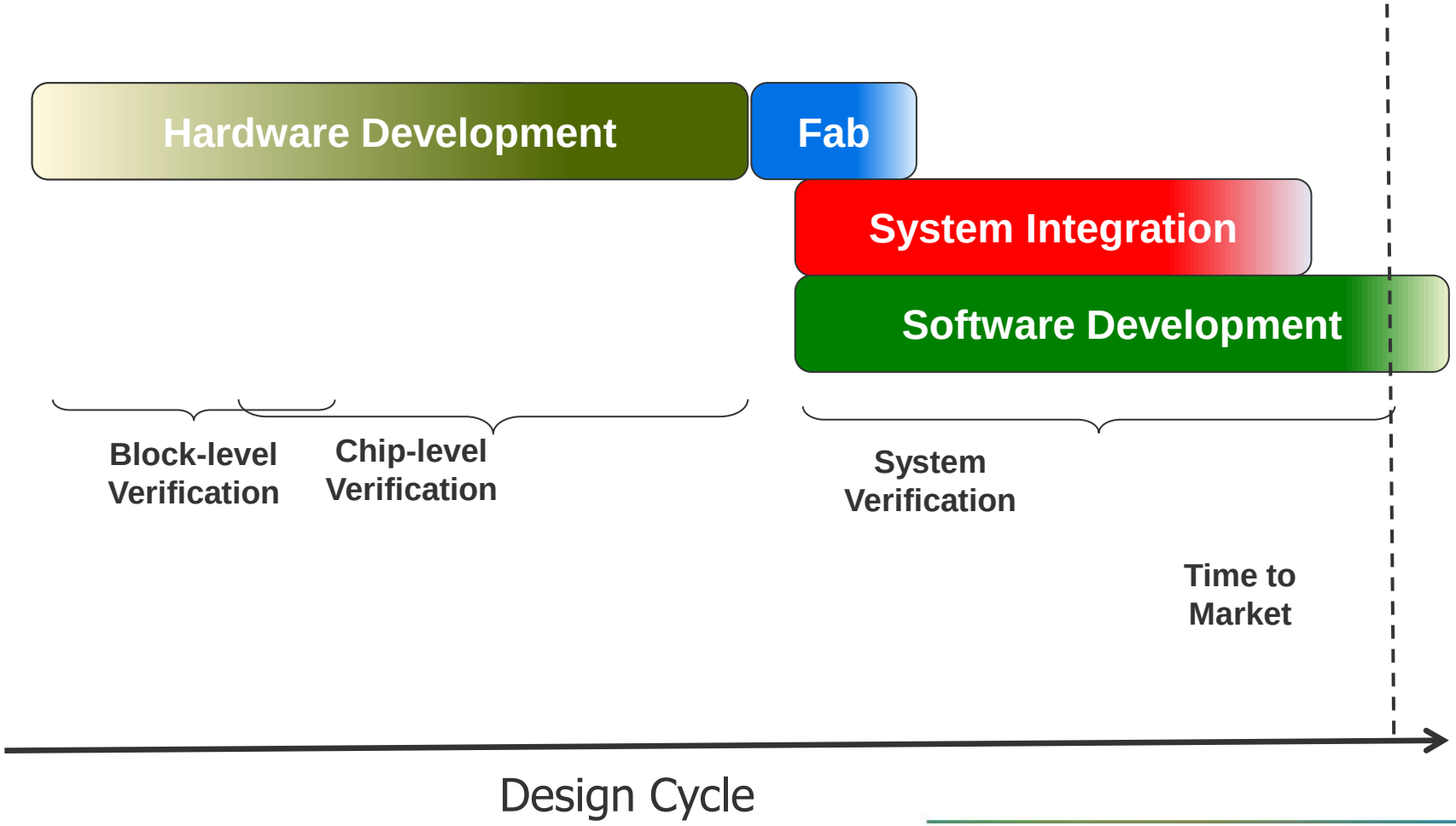
Software

SPEED

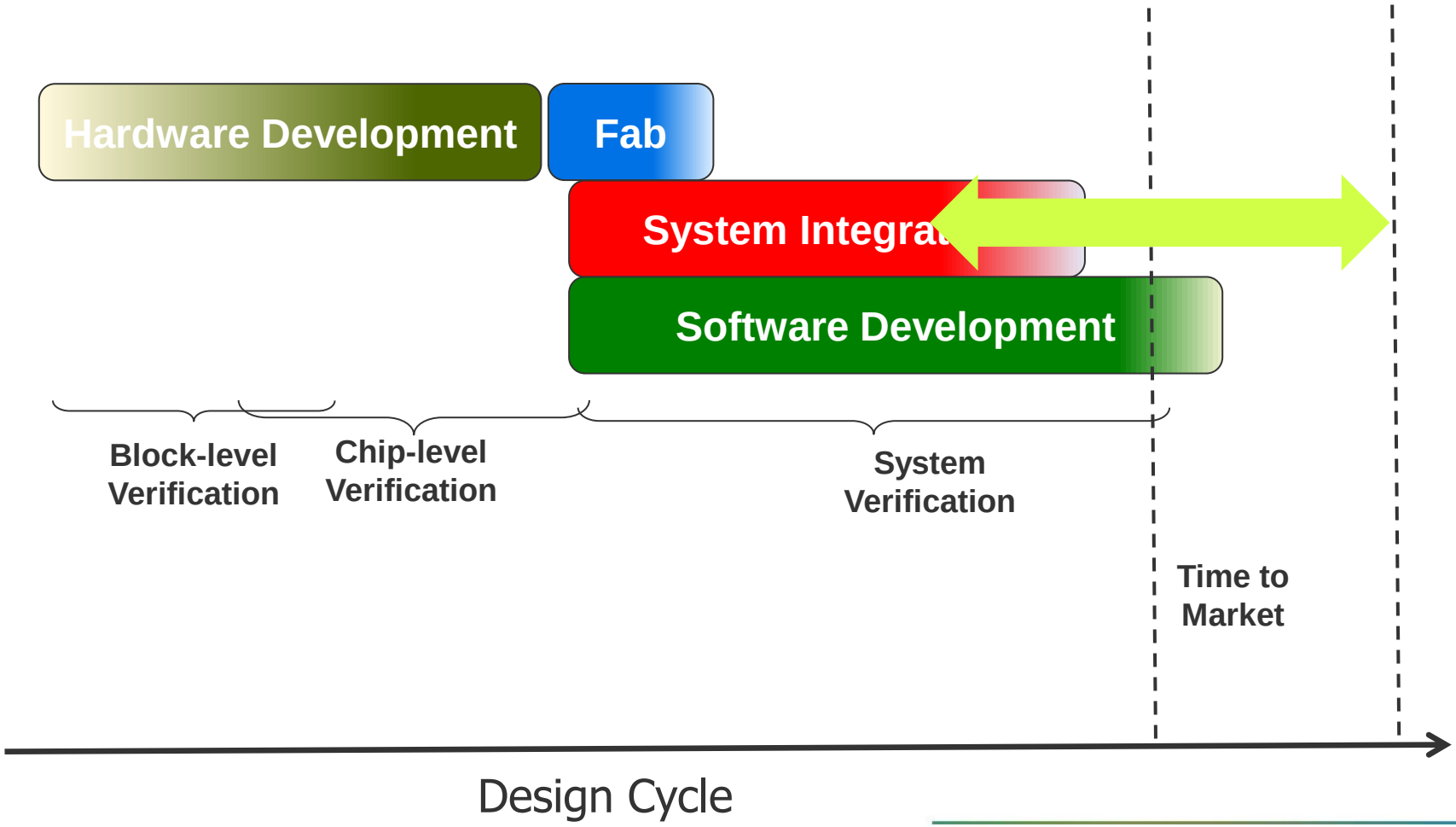



Debug

# Typical Development Cycle



# Typical System Development

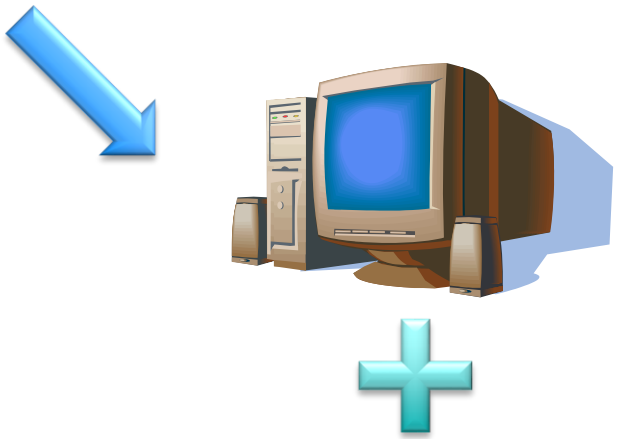


# Software Simulation

```
module counter (ck, en,
step, dout);
    input ck, en;
    input [3:0] step
    output [3:0] dout
    reg [3 : 0] dout;


    always @ (posedge ck)
    begin
      if ( en ==1 )
        dout = dout+step;
    end
endmodule
```

RTL Model



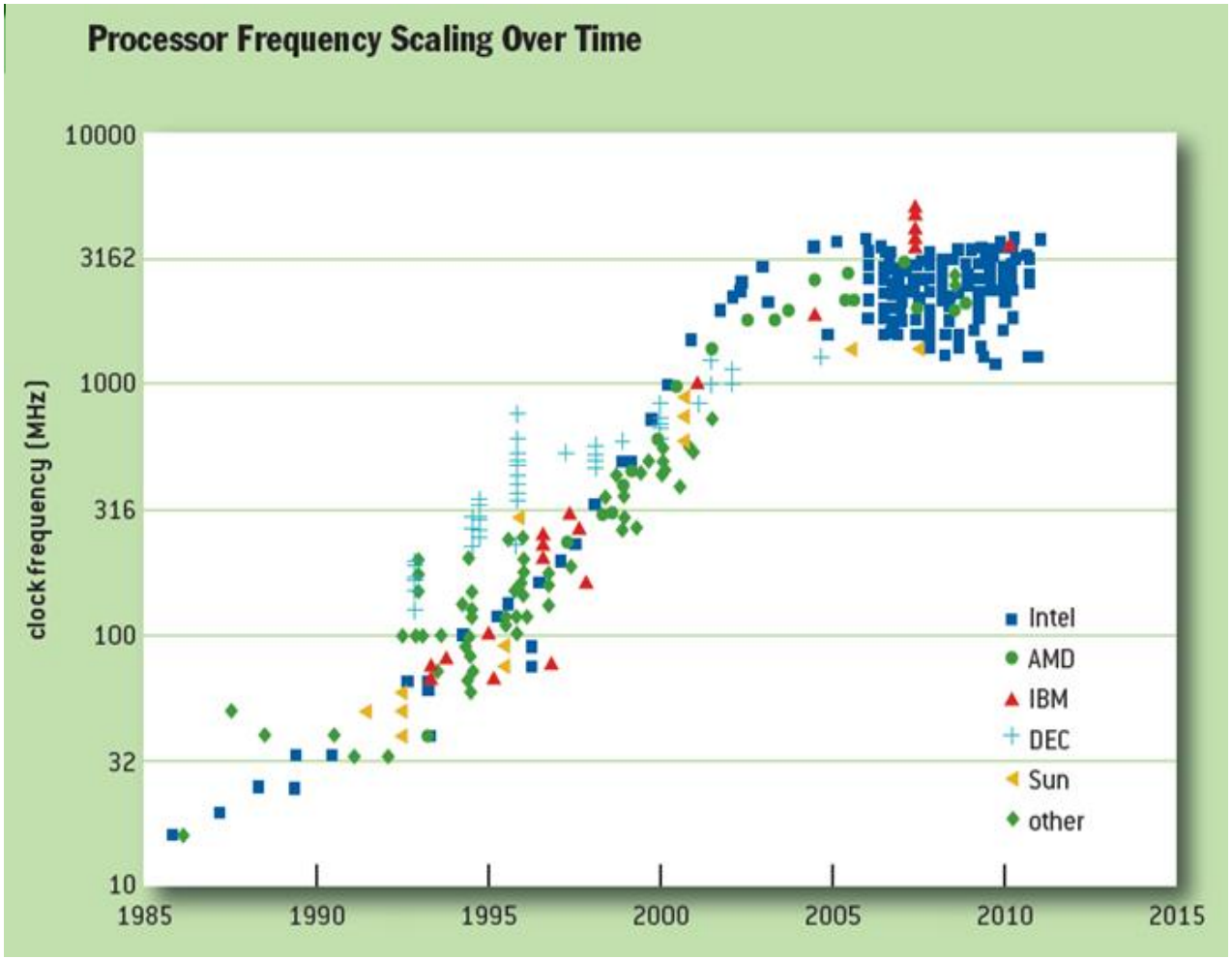
```
module counter (ck, en, step,
dout);
    input ck, en;
    input [3:0] step
    output [3:0] dout
    reg [3 : 0] dout;


    always @ (posedge ck)
    begin
      if ( en ==1 )
        dout = dout+step;
    end
endmodule
```

Test bench

- Model is represented in Data Structures
- Compute and propagate signal values

# Clock Speed Scaling Stalls




**Emulation Required to Extend Performance**

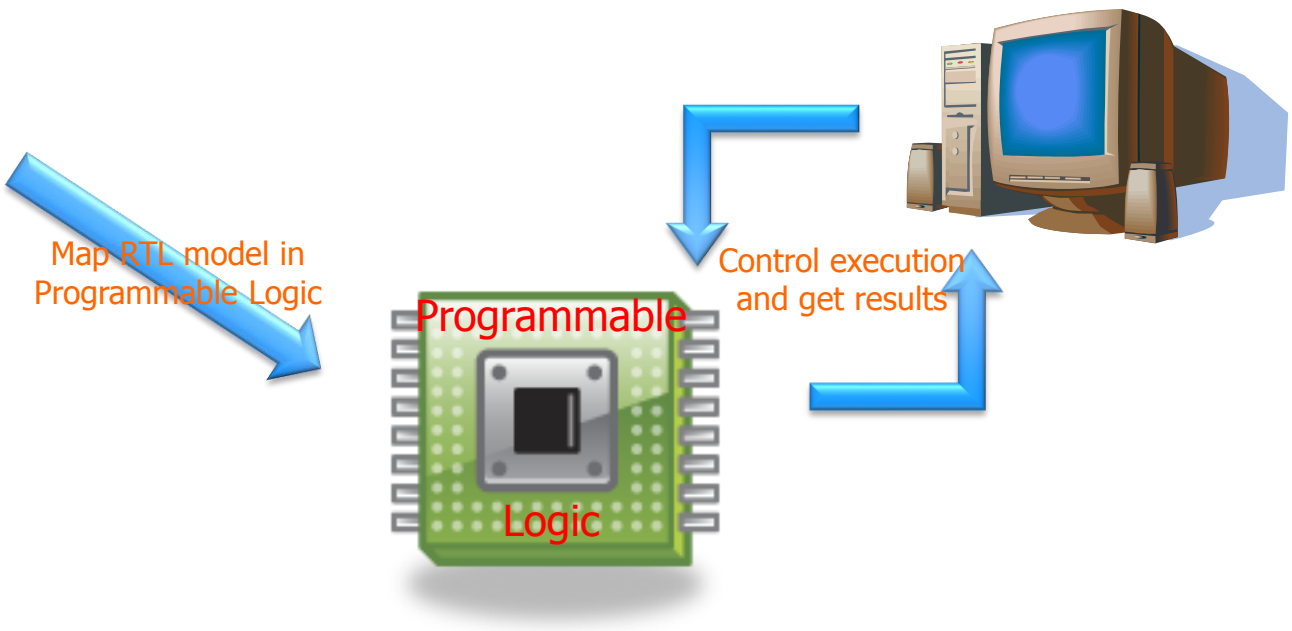
*Source:* Recording Microprocessor History *4/6/2012 Andrew Danowitz, Kyle Kelley, James Mao, John P. Stevenson, Mark Horowitz http://queue.acm.org/detail.cfm?id=2181798*

# Emulation

```
module counter (ck, en,
step, dout);
    input ck, en;
    input [3:0] step
    output [3:0] dout
    reg [3 : 0] dout;


    always @ (posedge ck)
    begin
      if ( en ==1 )
        dout = dout+step;
    end
endmodule
```

RTL Model




- Multiples FPGA reproduce Model behavior from high level description

# Start Early – Continue for Entire SoC Life

Architecture Phase

Block-level Verification

Fullchip/SoC Verification

Software, Firmware & Device Drivers

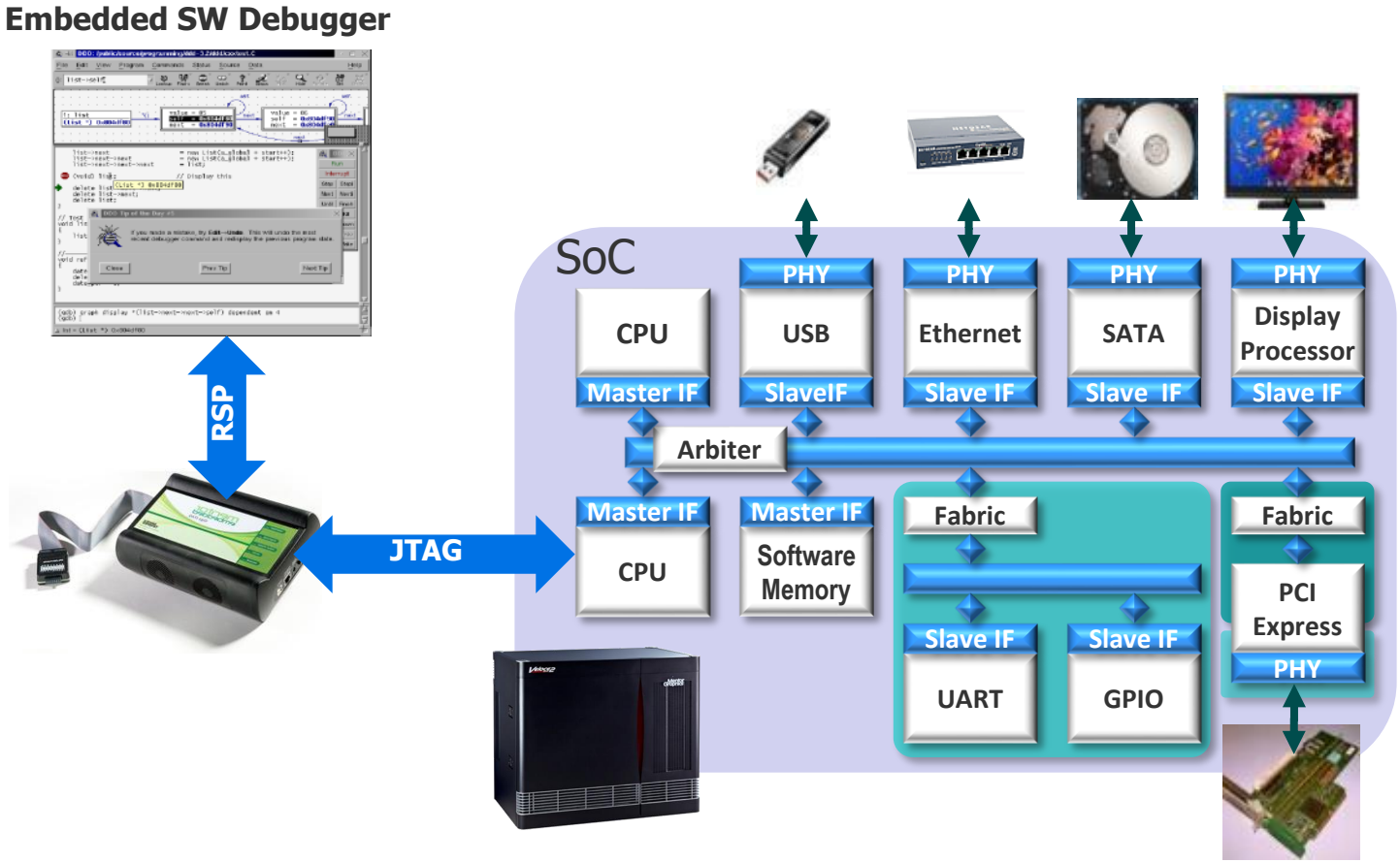
Systems Validation

Post Silicon Bringup and Validation

# Agenda

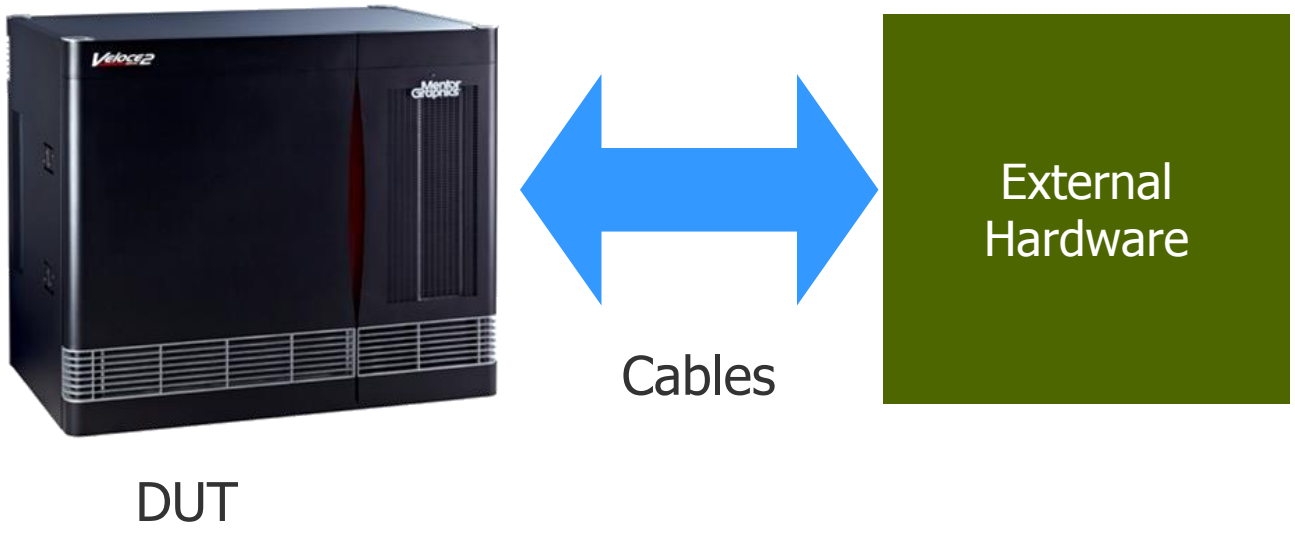
- What is Emulation
- Use models
- Veloce Architecture Overview
- Veloce Software
- Practical : Use Veloce to verify Veloce

# Modern SOC environment

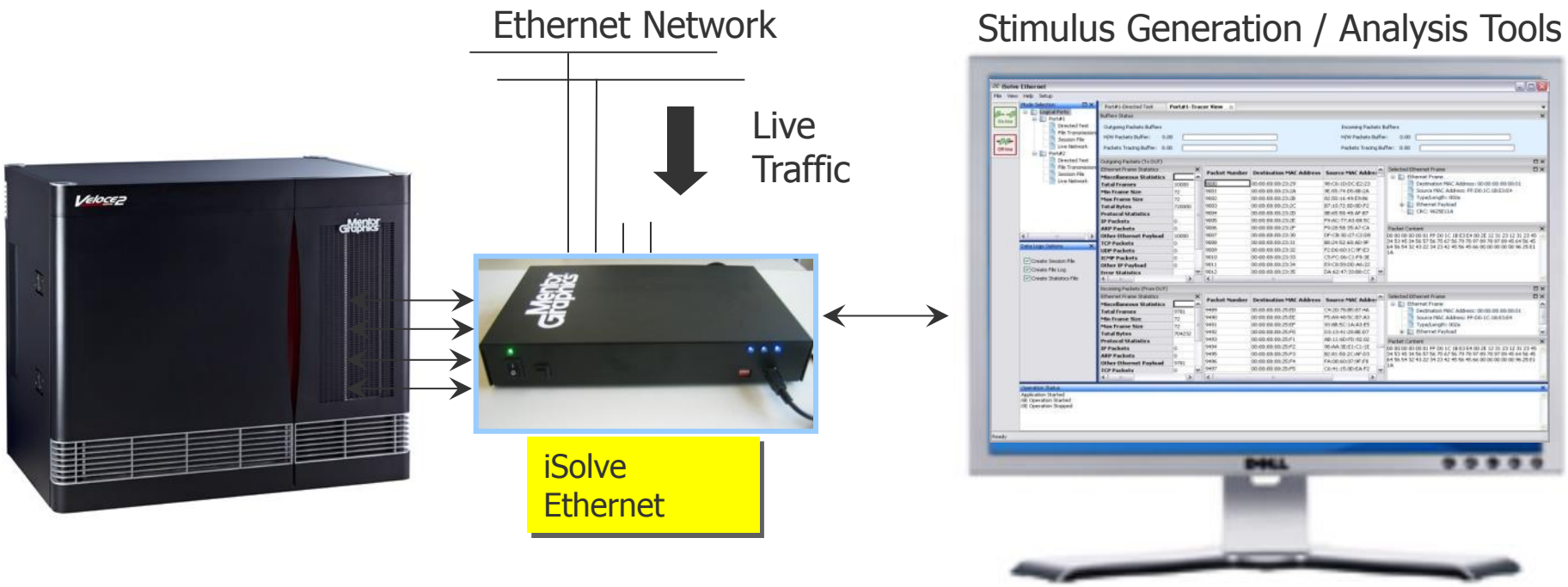


# ICE use Model

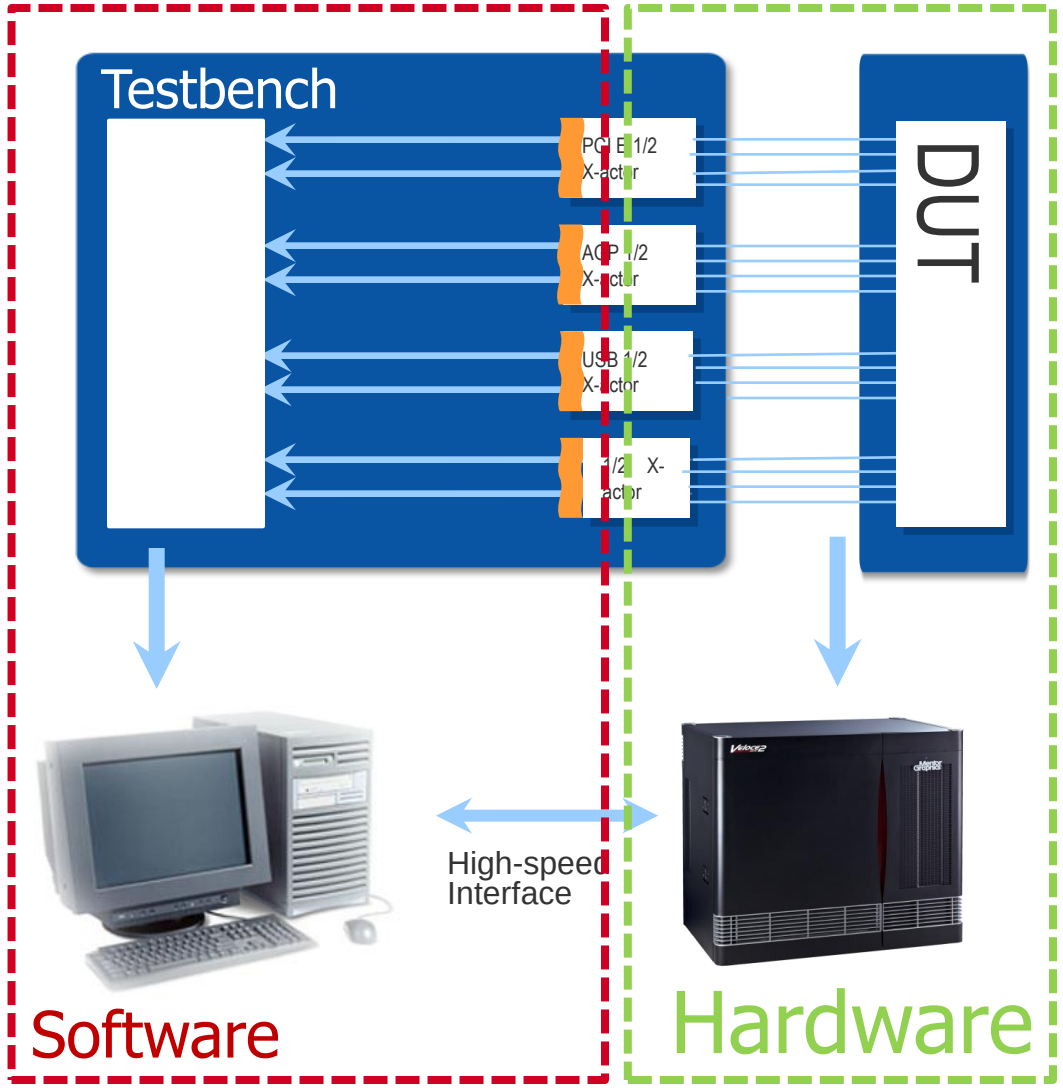
- The emulator is connected to actual Hardware



# Ethernet Verification With ICE

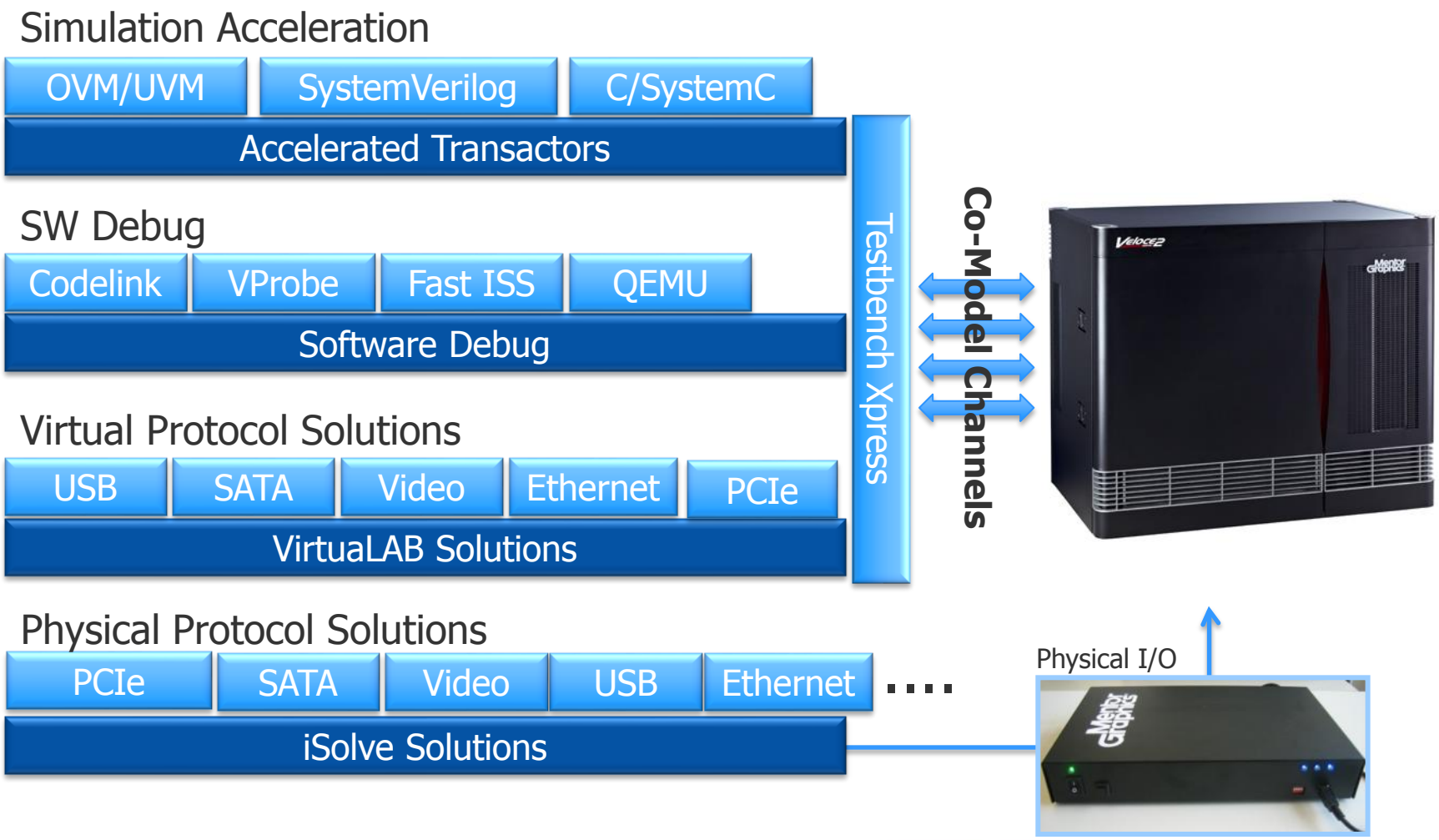


# Co Emulation
# Testbench Xpress (TBX)




- Testbench divided in 2 parts one in emulator the other in Station
- Communication via transactor
- Software sends commands that are interpreted by transactor to generate DUT stimulus

# Veloce Use Models



# Agenda

- What is Emulation
- Use models
- Veloce Architecture Overview
- Veloce Software
- Practical : Use Veloce to verify Veloce

# Veloce 2 Architecture

Crystal2 Chip


X 16

AVB2

X 64

+ 40 SXB (Switch boards)

+ 4 CXB (clock board)

Maximus2

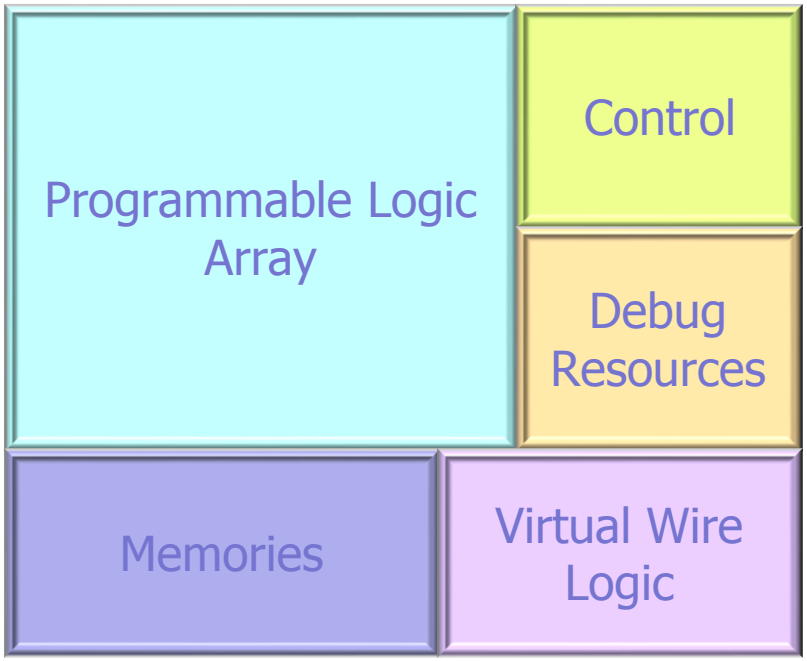

16 AVB2

+ 6 SXB (Switch boards)

+ 1 CXB (clock board)

Quattro2

# Crystal 2 IC

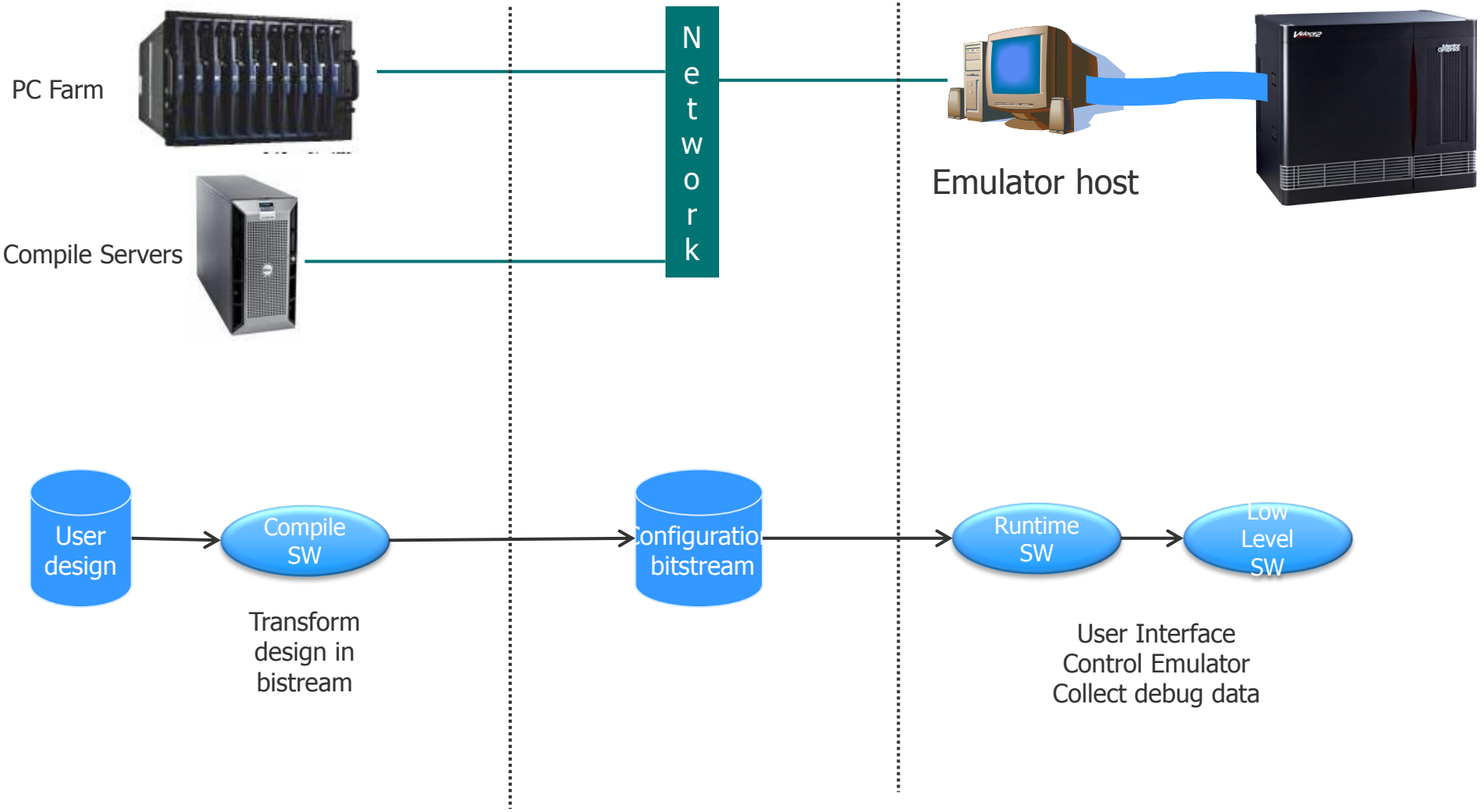



- Programmable Logic Array
  - Set of LUT an Sequential elements
  - Interconnect Network
- Memories :
  - User Memories model
- Virtual Wire Logic
  - Transport signals between chips
- Debug Resources
  - Trace every Sequential element and memories
  - Triggers
- Control
  - Load configuration
  - Control emulation

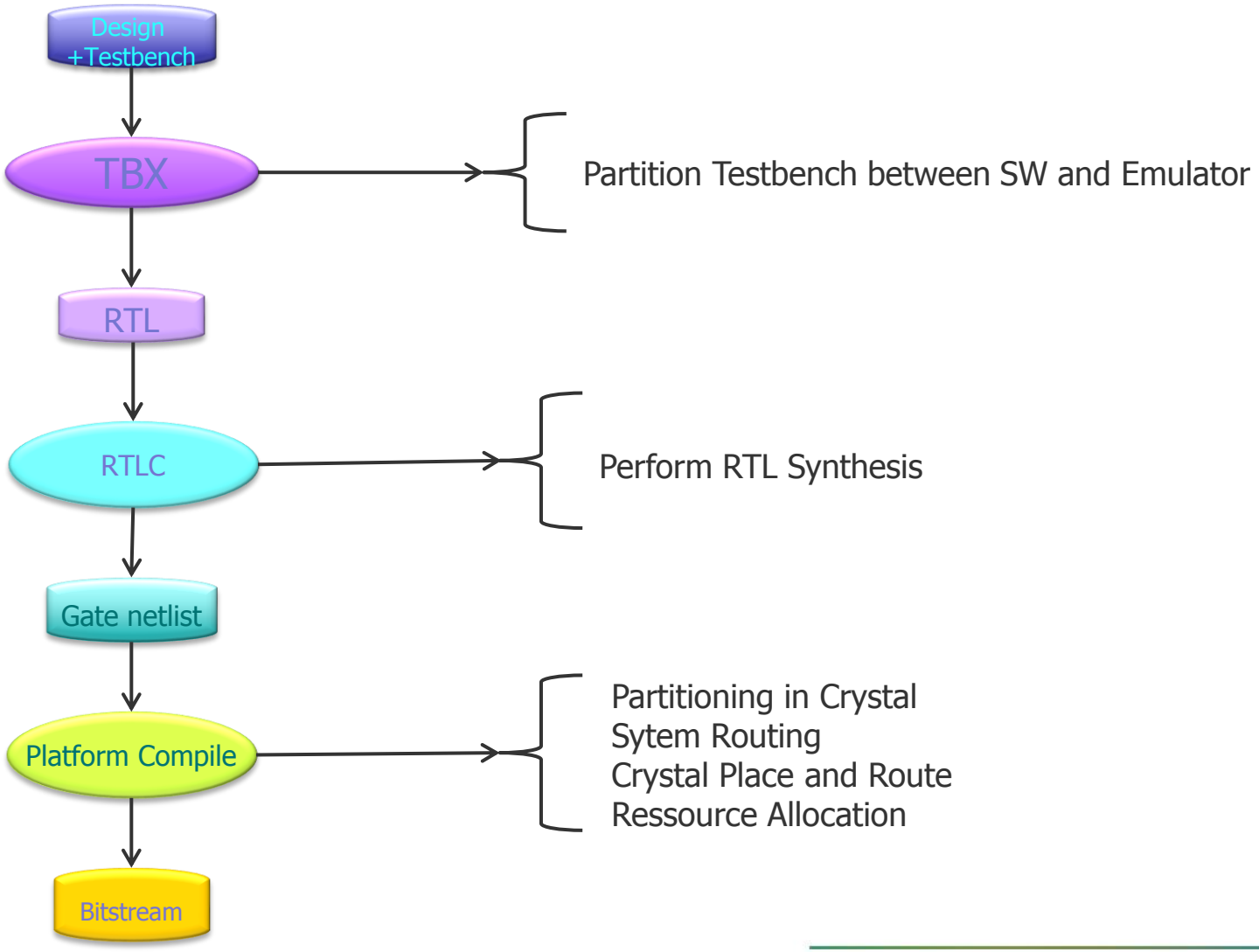
# Agenda

- What is Emulation
- Use models
- Veloce Architecture Overview
- Veloce Software
  - Compile Software
  - Runtime Software
- Practical : Use Veloce to verify Veloce

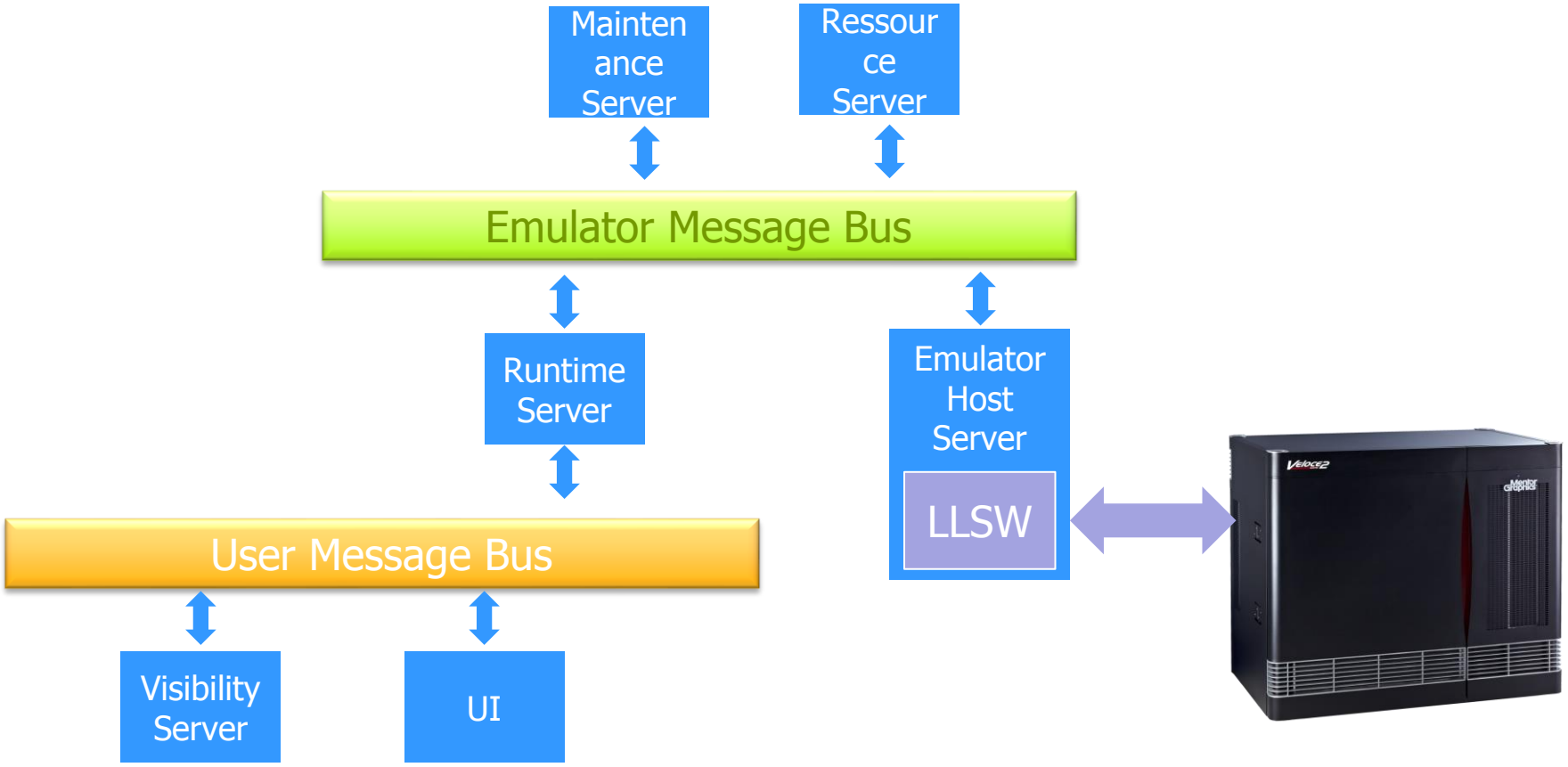
# Software Overview



# Compile software



# Runtime Software



# Agenda

- What is Emulation
- Use models
- Veloce Architecture Overview
- Veloce Software
- Practical : Use Veloce to verify Veloce
  - Challenges
  - The verification infrastructure
  - Verifying ASIC
  - Verifying the Compilation software
  - Low level software integration
  - Firmware validation debug
  - Runtime software integration

# Challenges

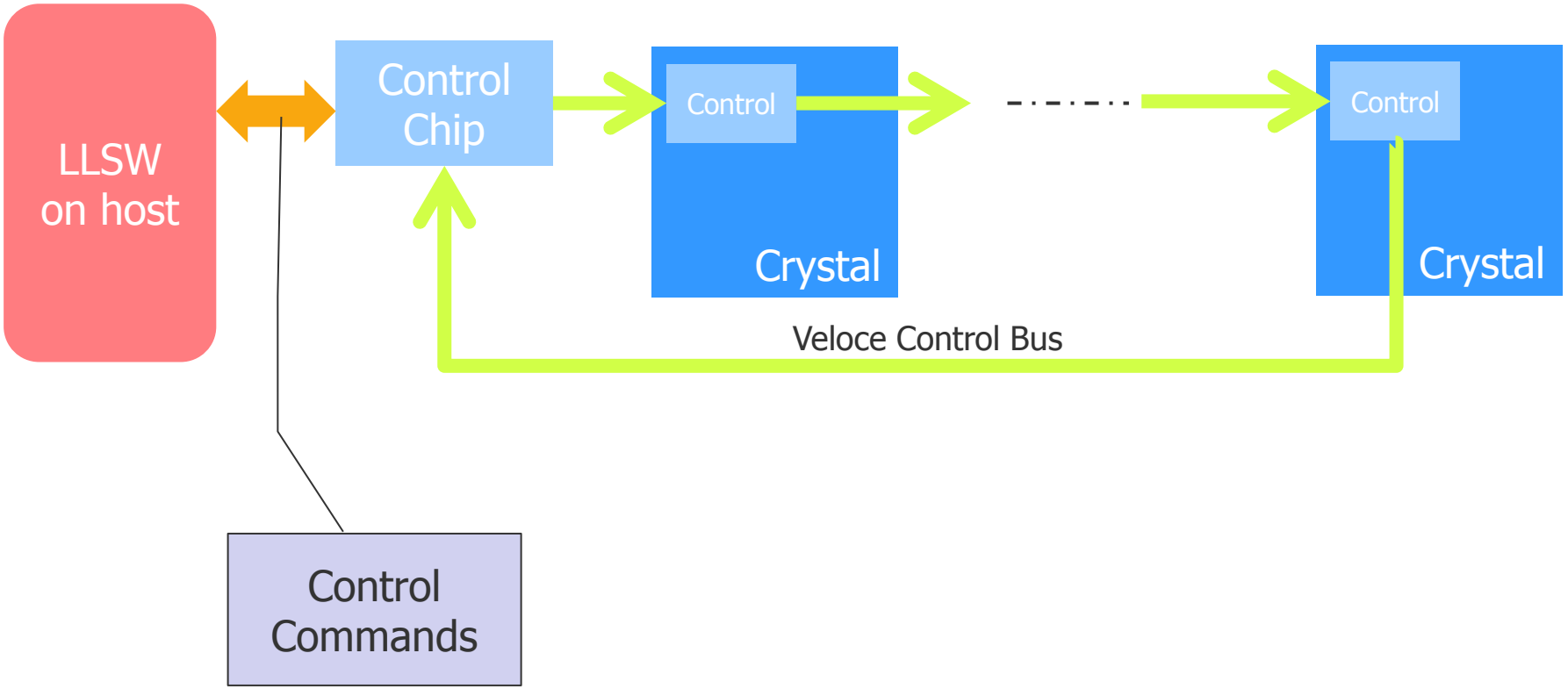
- Complex system
  - ASIC
  - FPGAs Firmware
  - Mutliple software components
- Verifying all component of the system and their interactions
- Time to get a bug
- Size :
  - Detailed model of a full Emulator of next generatio will not fit in current generation

# Addressing the challenges

- Use a comodel approach for more abstraction level and connection with the software
- Divide verification in steps based on functionality
- Simplify the model by using different abstraction level depending on what functionality is tested

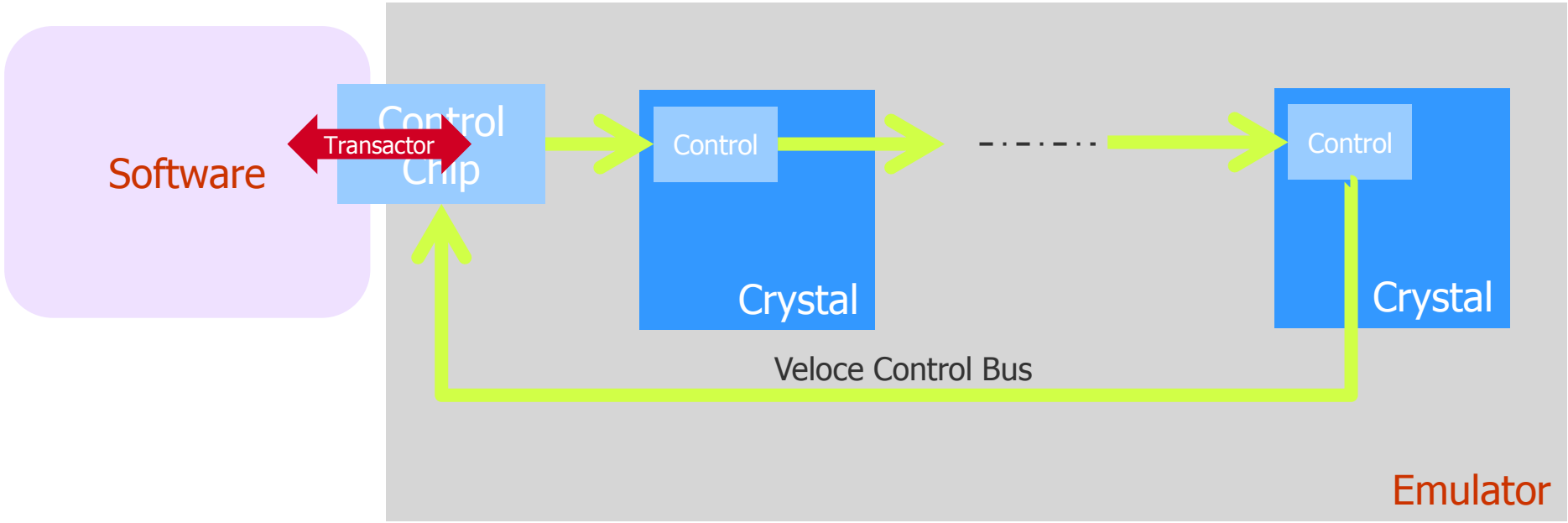
# Verification Infrastructure

More details on Veloce



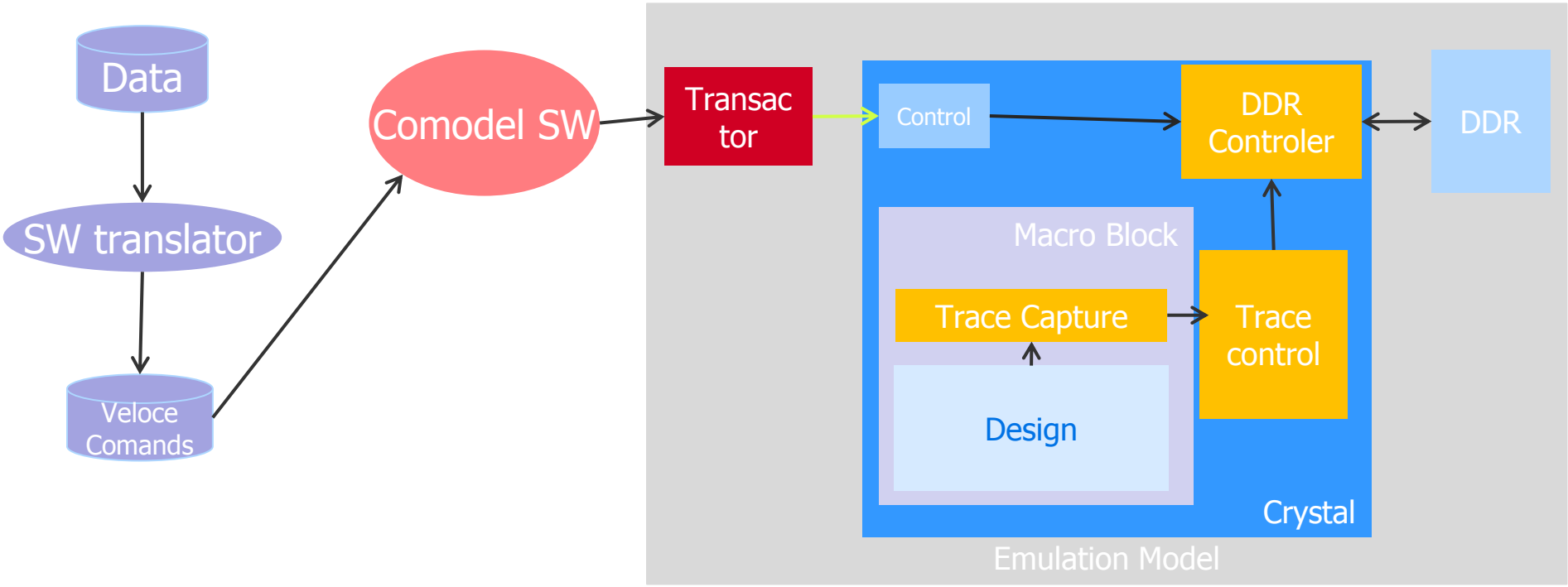
# Verification Infrastructure

## Emulation model



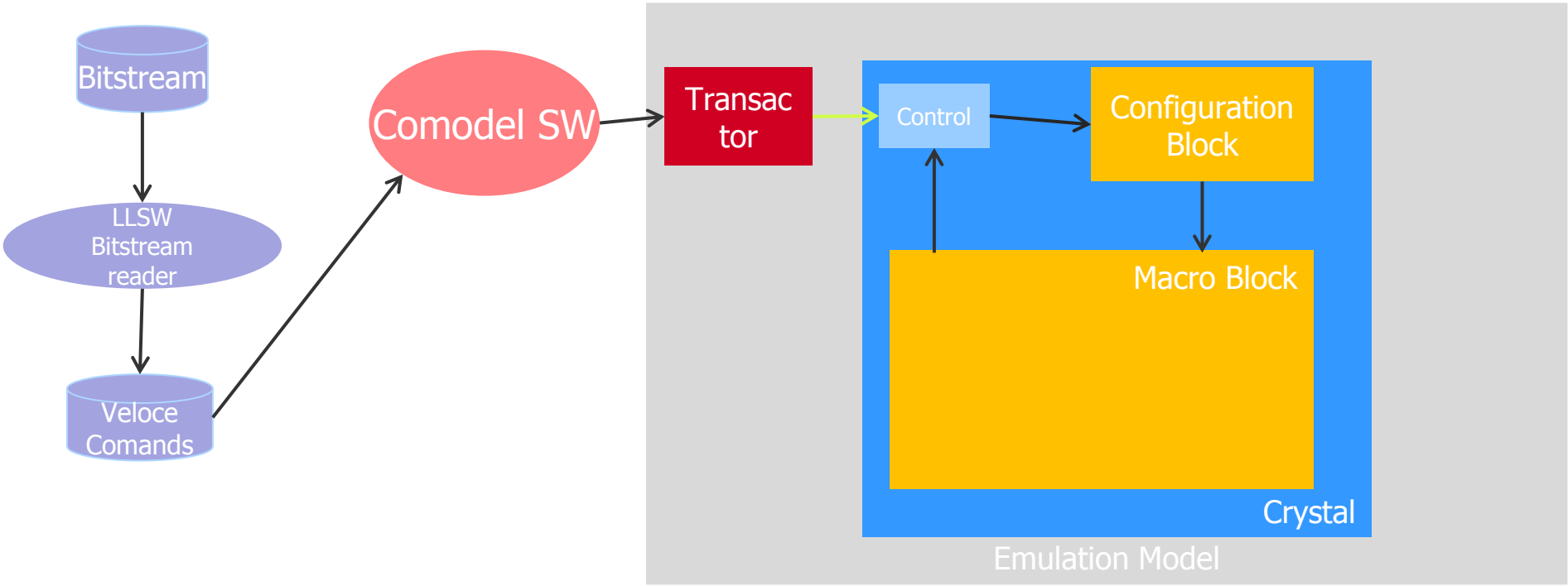

Nature of the Software, Transactor and Model in emulator depend on abstraction level

# ASIC verification : example of Trace




Trace Capture, Trace Control and DDR Controler are Accurate models
Design is a gate level netlist generating random data
Data are either manually generated or come from actual compile
Run million cycles on multiple designs

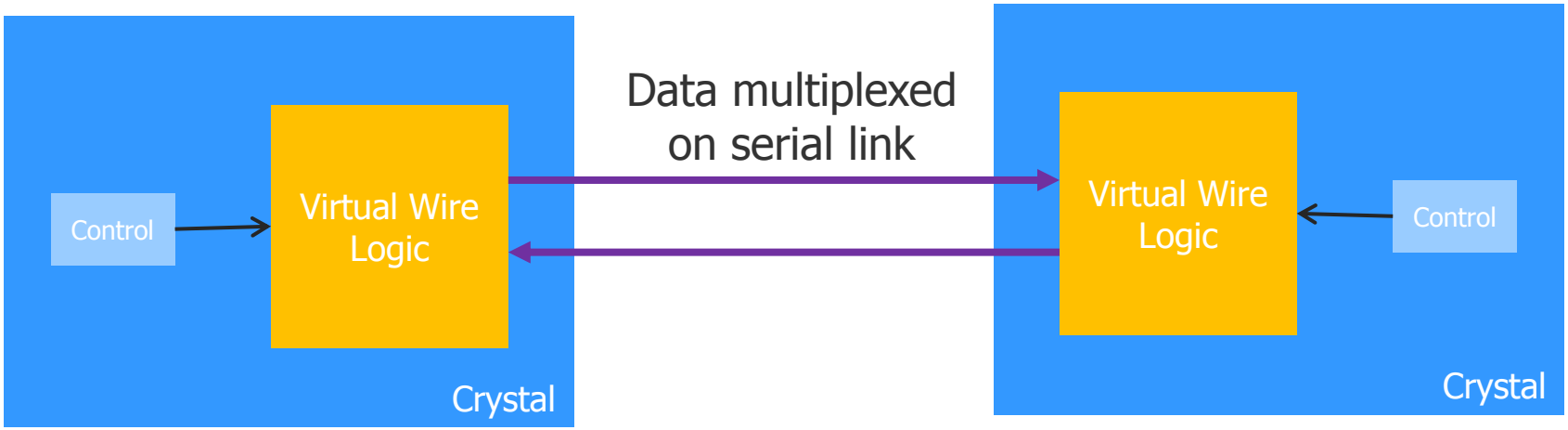
# Verification of compile SW




Macro and Configuration blocks are Accurate models
Bitstream is the output of actual compile flow
Verify behavior of design

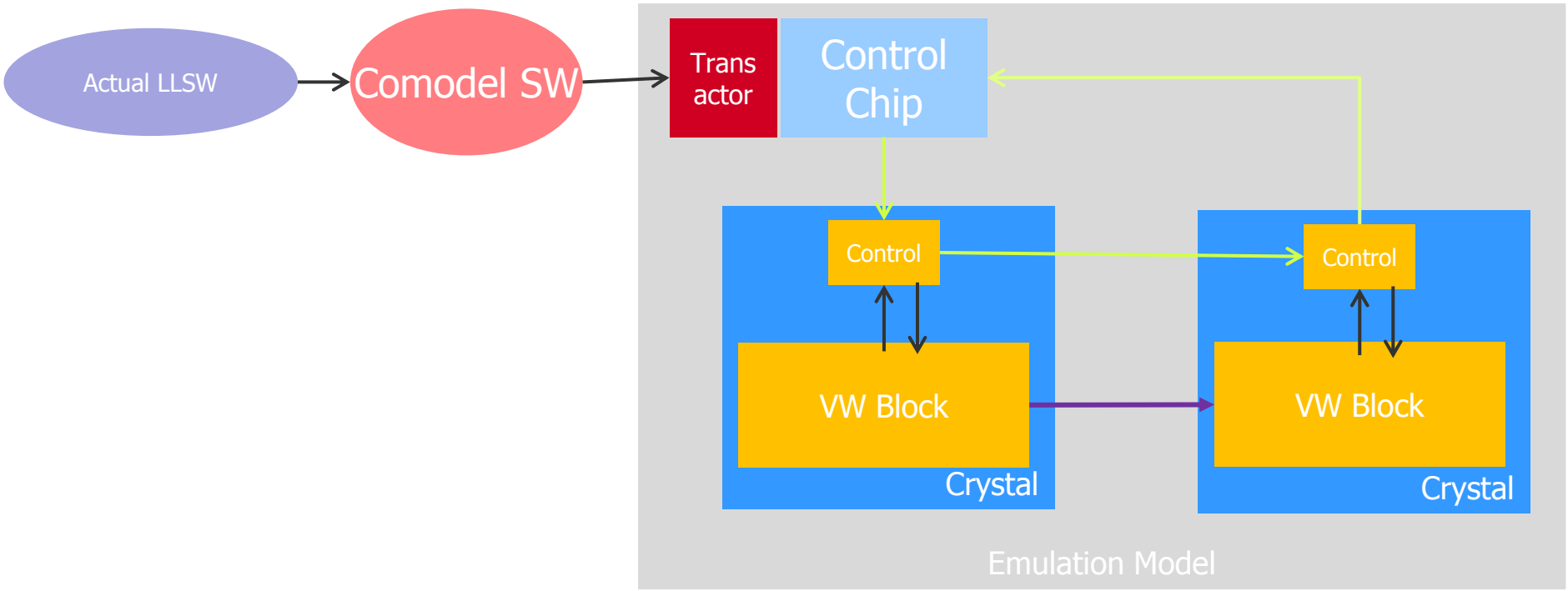
# Verification of Low Level Software 1/2

Example : Virtual Wire Synchronisation




Virtual Wire need a training/Calibration sequence
This sequence is controled by Low Level Software

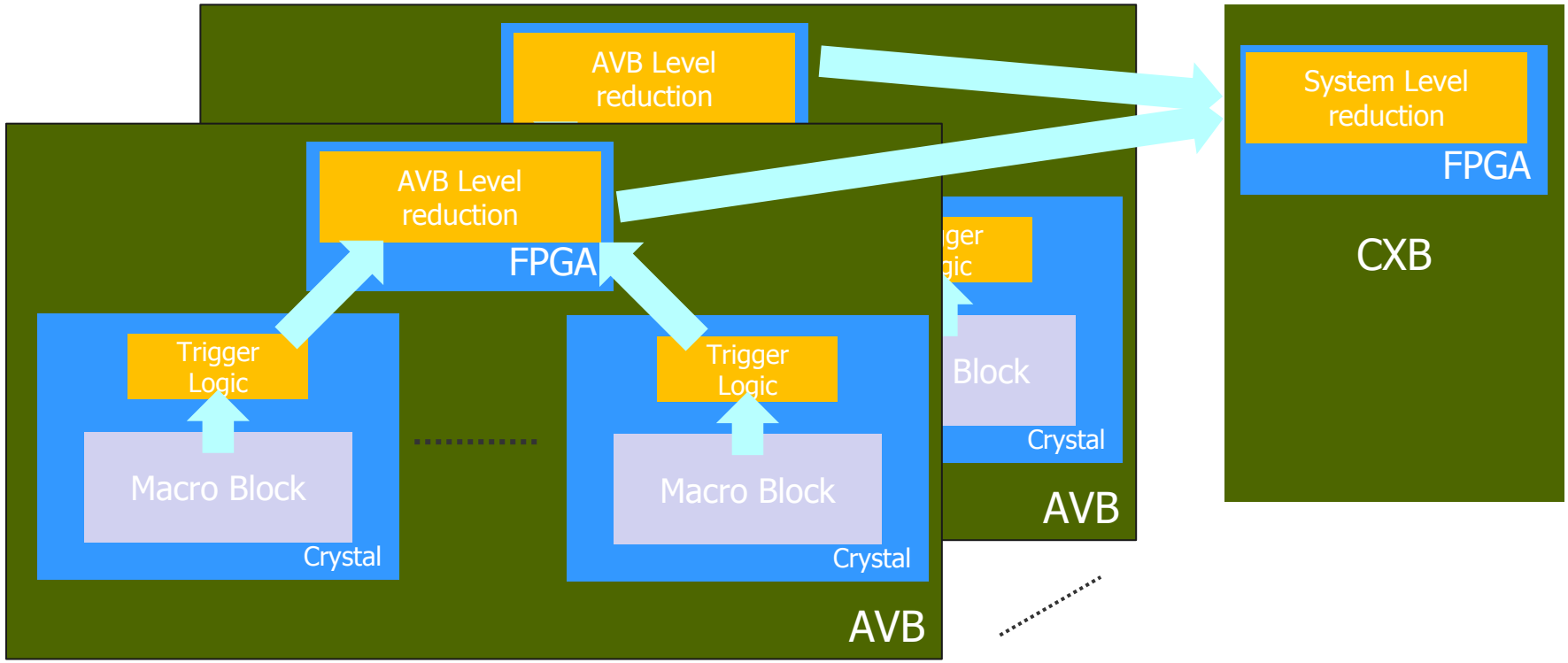
# Verification of Low Level Software 2/2




Control block in Crystal and VW block are accurate model
Actual LLSW communicate with the model through comodel SW and transactor

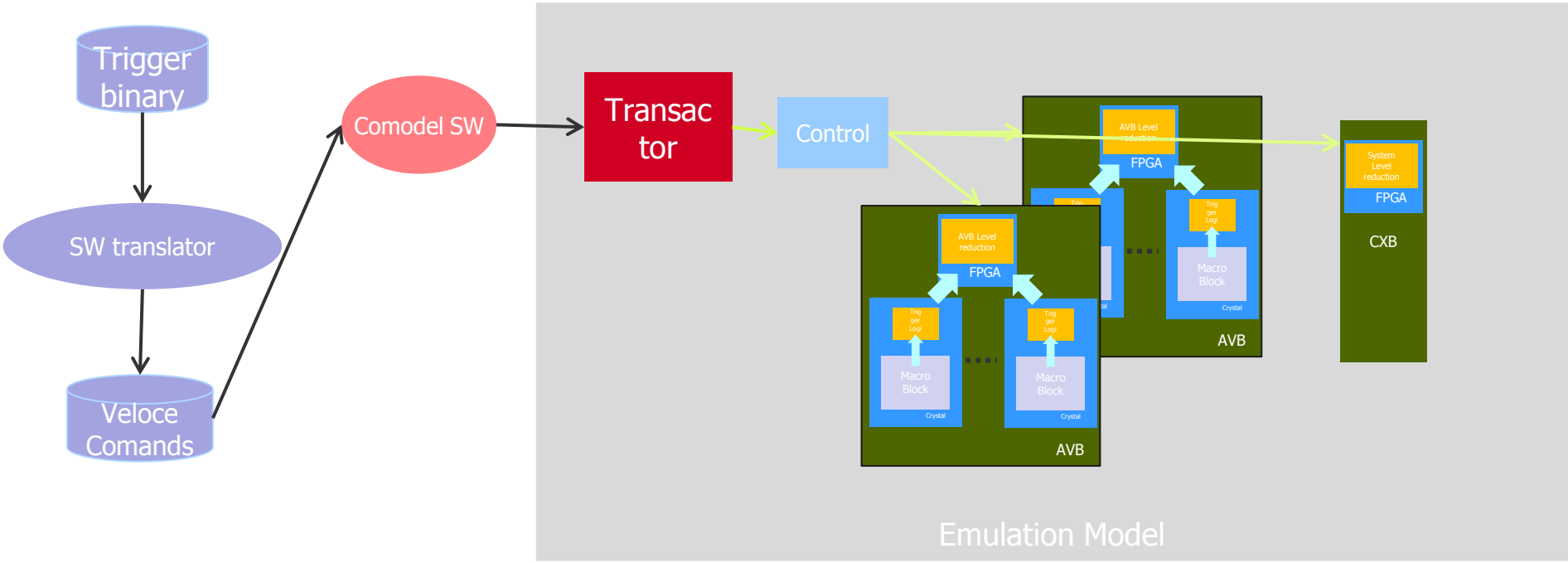
# Verification of Firmware 1/2

Example : Trigger Reduction



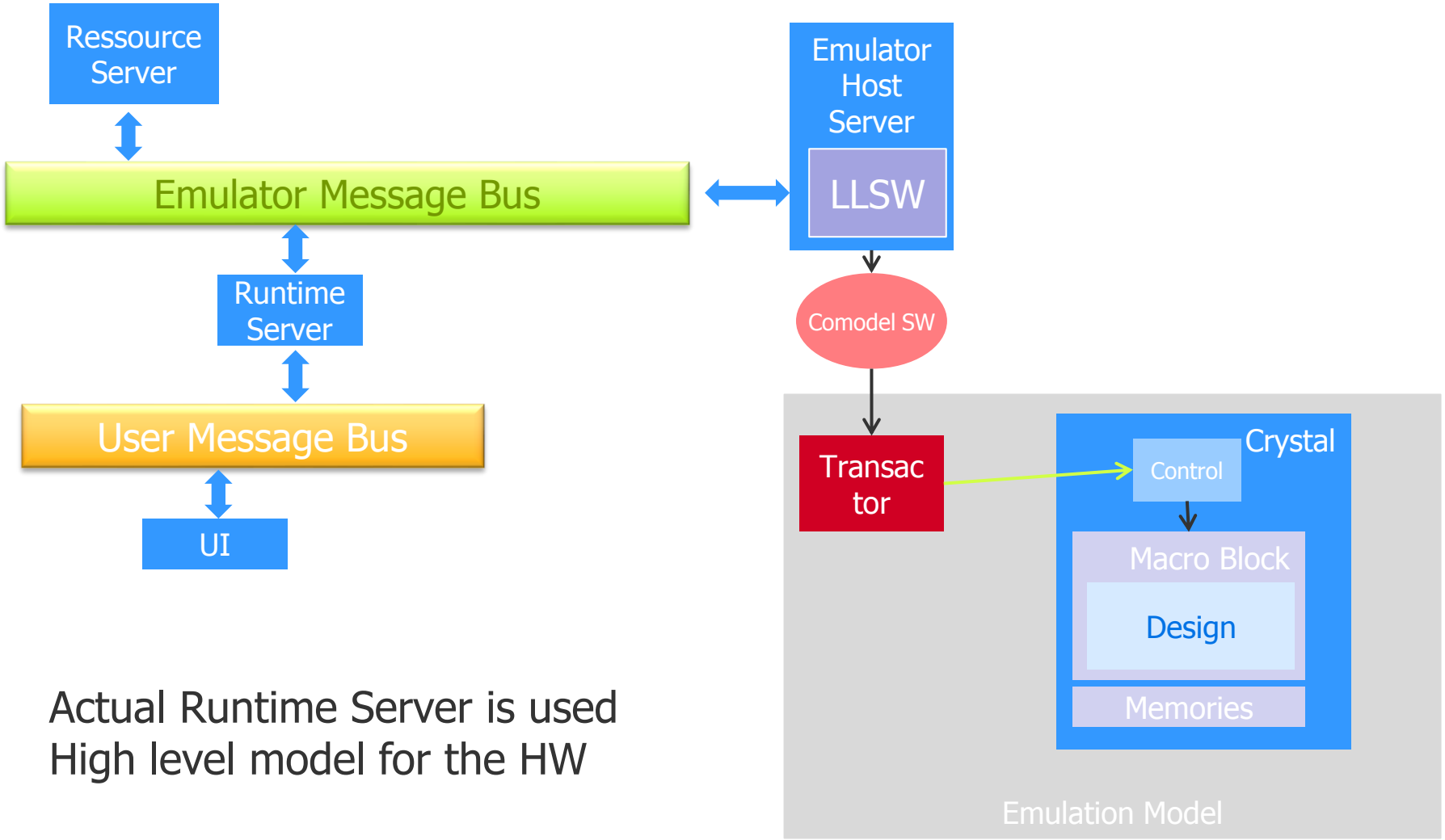

A trigger express a condition on values coming from the design
At AVB and System level it is implemented in FPGA
A binary is genrerated by runtime SW to express condition

# Verification of Firmware




Design is modeled as gate level netlist
Trigger binary is generated by actual runtime SW
Verify behavior of trigger in multiple design sequences

# Runtime SW Verification




Actual Runtime Server is used
High level model for the HW

# QUESTIONS