PaSiVe: Tool for Matlab/Simulink Design Flow for Fast Prototyping Signal Processing Systems on FPGA

Zhongren Cao Joshua Ng

Senior Research Engineer

California Institute for Telecommunications & Information Technology

University of California, San Diego

zcao@soe.ucsd.edu

Bhaskar Rao

Professor

Electrical and Computer Department

University of California, San Diego

Outline

- Motivation
- Fast Prototyping Design Flow
- PaSiVe: Tool for Matlab/Simulink Flow
- Example Project

Sponsor:
Ericsson Research

EXperimental reConfigurable radIo TEstbed

A joint project between Ericsson Research and Calit2@UCSD

Motivation

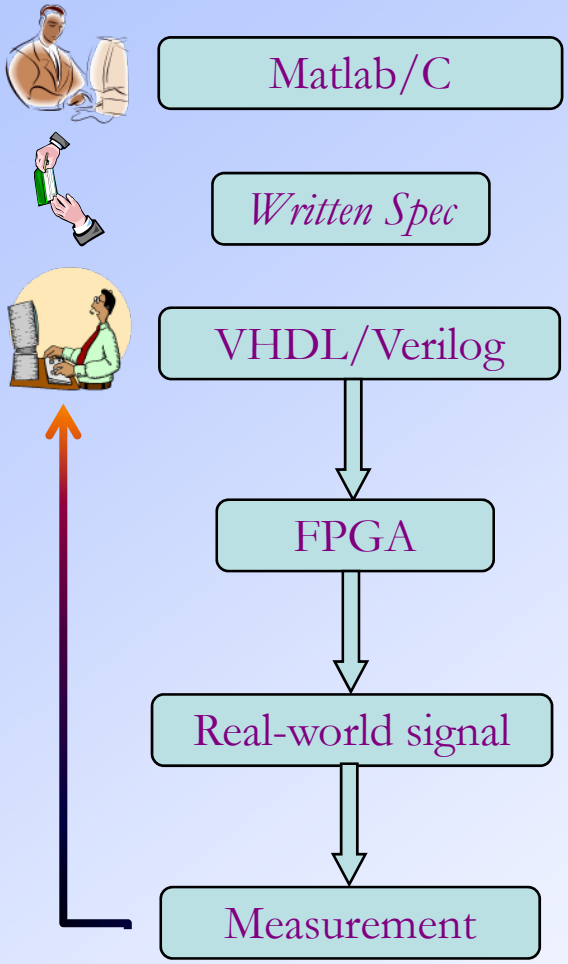
- **Fast Prototyping**
  - Implement new ideas in a real scenario quickly
  - Verify innovations and identify problems early in the design phase
  - FPGA is widely adopted for fast prototyping projects
    - Programmability
    - Computing parallelism
- **Limitation of the Tradition FPGA Design Flow**
  - Team is separated into system/algorithm and hardware groups.
    - Use different languages → Communication problem within the team
  - Different expertise require different development environments
    - Time consuming and error-prone for translation
  - Very difficult for joint algorithm-architecture-circuit optimization
  - Unsuitable for fast prototyping projects

Motivation

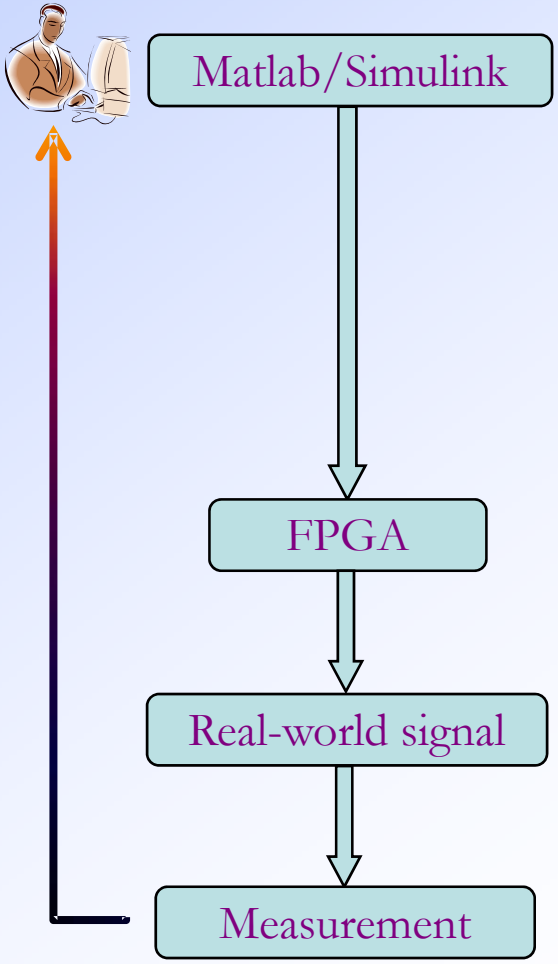
- **Matlab – The De Facto Tool for Signal Processing**
  - Widely adopted by signal processing system and algorithm designers
  - Providing abstractions for
    - Complex numbers, matrices and vectors
  - Supporting arithmetic operation with multi-dimension data structure
  - Preserving parallelism of the algorithm
  - Agnostic to hardware implement architecture
- **Simulink**
  - Smoothly integrated with Matlab
  - Graphic Interface that captures addition system aspects information
    - Implementation architecture
    - Timed data flow
  - Ideal for fast system prototyping

Design/Implementation Workflow

Traditional Design Flow



Matlab / Simulink Design Flow



Fast Prototyping Design Flow

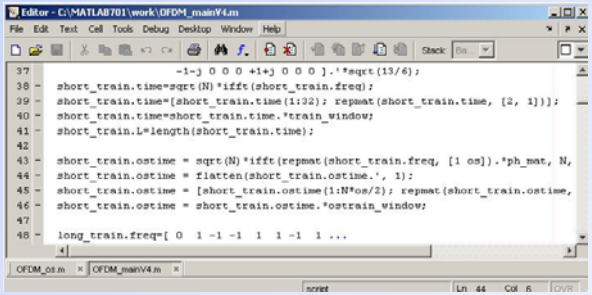
- **Simulink – Unified Design Interface**
  - Hierarchy layered abstraction for a signal processing system
  - Intuitive for both system engineers and hardware designers
    - Algorithm development and modeling
    - Architecture and circuit optimization
  - A golden reference is still needed – Matlab description
    - Generate verification data sets
    - Verify operation accuracy
- **Three Major Phases in the Matlab/Simulink Flow**
  1. Executable Matlab system description
  2. Bit-true and cycle-true Simulink model
  3. Real-time hardware execution

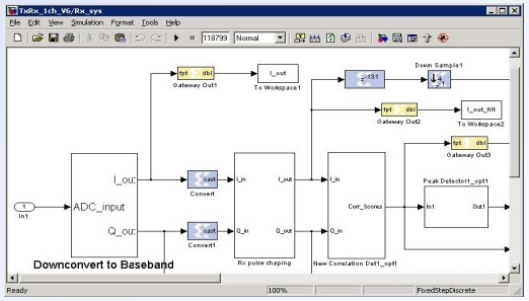
Matlab and Simulink Design Flow

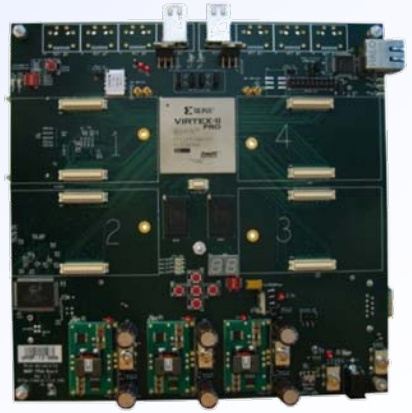
System and Algorithm Description in Matlab

Bit True and Cycle True Simulink Model

Real-Time FPGA Execution

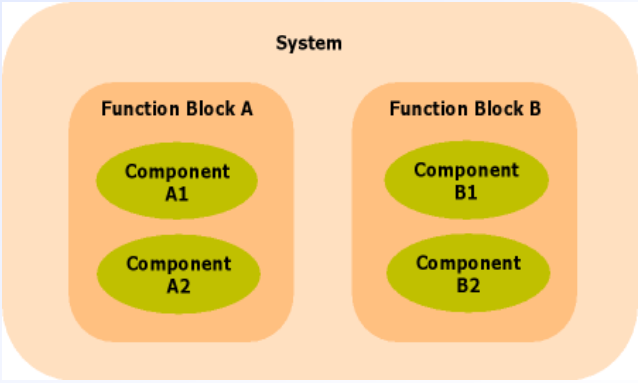






Executable System Description in Matlab

- **System Specification**
  - Hierarchy system partitioning
  - Algorithms used by each system module
- **Generates Test Data Set for Verification**
- **System Partitioning**
  - Based on data flow and functionalities
  - Considering the expertise and skills of designers
  - Looking for reasonable hierarchy levels
  - Frequent modification
    - Typical for prototyping projects
    - Good system partitioning
      - Localize modification within the hierarchy structure

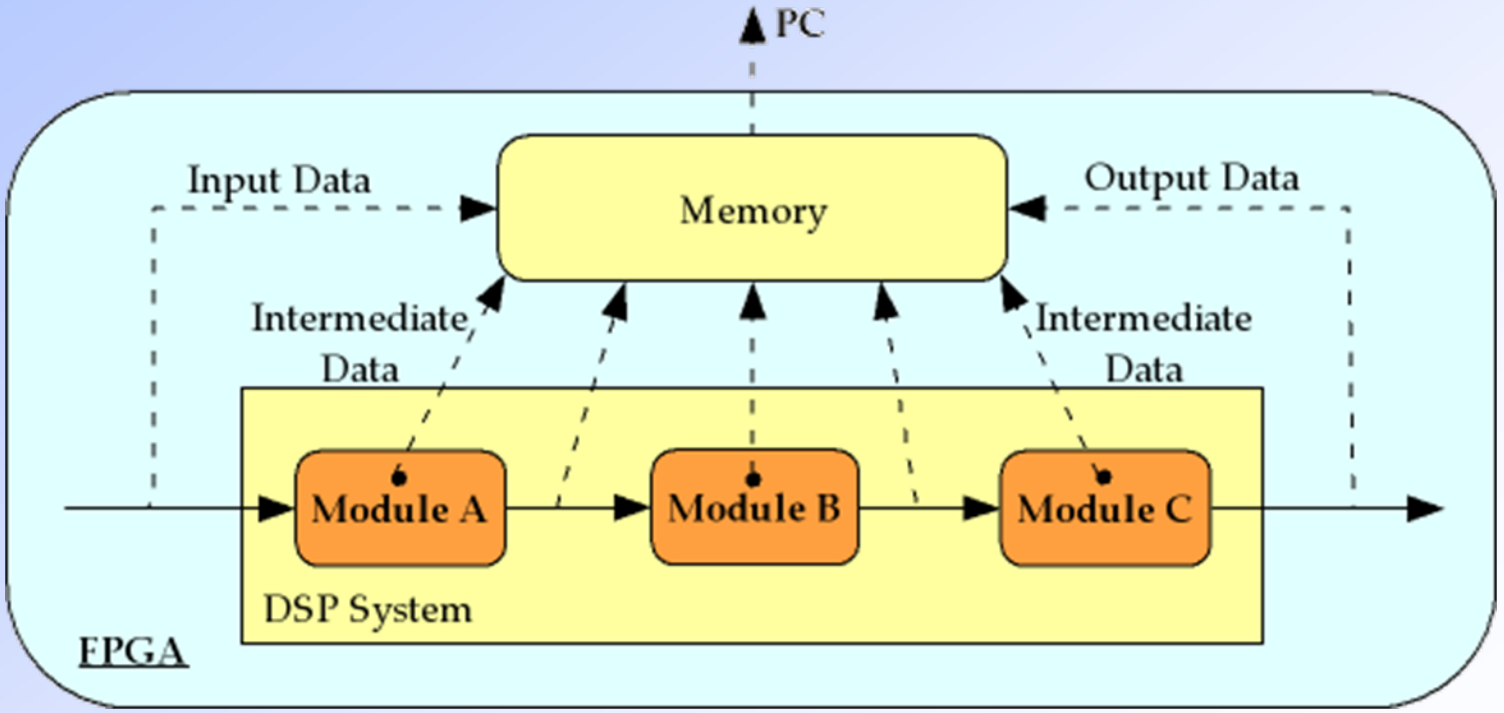


Bit-True And Cycle-True Simulink Model

- **Magic of Hardware Equivalent Blocks**
  - Support fixed point simulation in Simulink
  - Implanted with verified and optimized technology-specific HDL
    - Xilinx FPGA – System Generator
    - Altera FPGA – DSP Builder
    - ASIC – Synplify DSP or UC Berkeley INSECTA
    - User customized blocks
  - Early stage power and area estimation
- **Simulink Model vs. Matlab Description**
  - Data in Simulink is timed with explicit sample rate information
    - Matlab's data are untimed.
  - Simulink has control logic and delay pipelining
    - No need to consider in Matlab
  - Fixed point arithmetic in Simulink

Real-time Hardware Execution

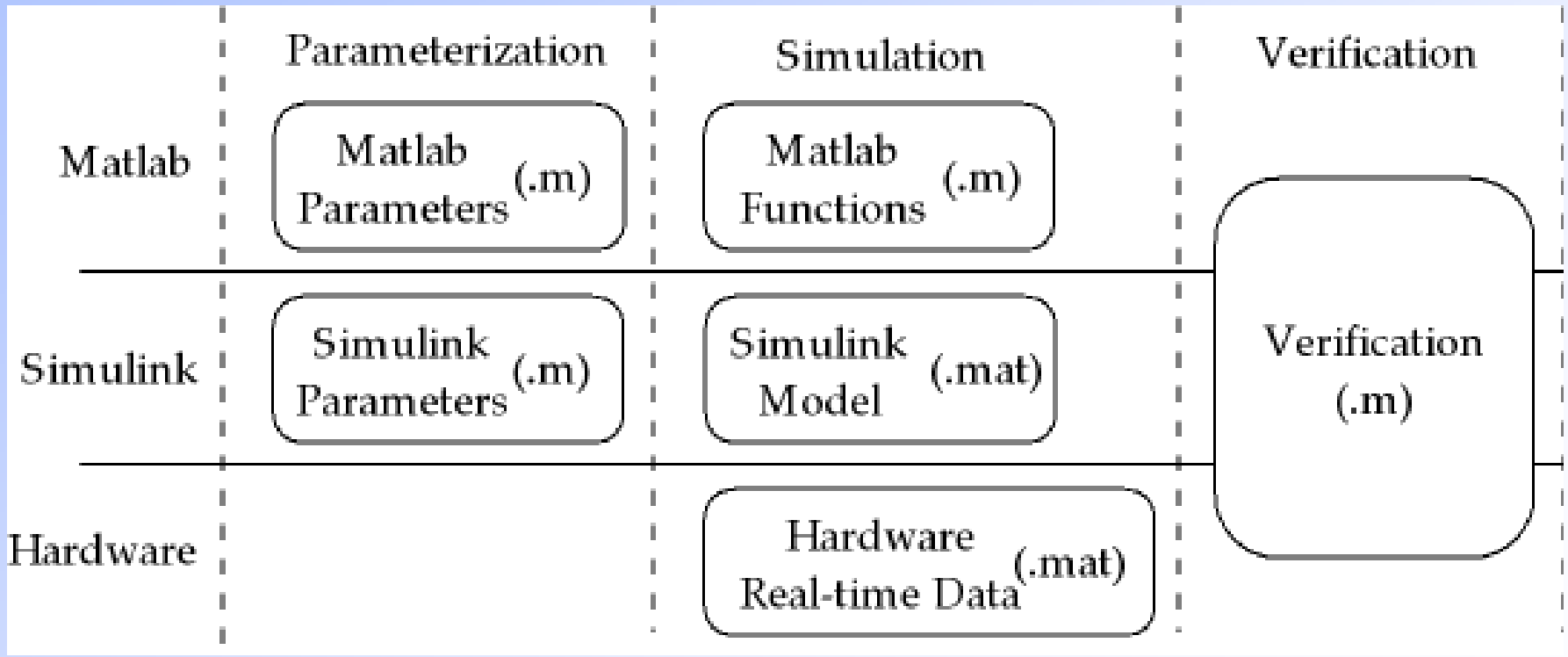
- **Verification With Real Data**
  - Additional data save scheme is built into the FPGA design.



PaSiVe Overview

- **Collaborative Prototyping is Necessary**
  - Design becomes more and more complex
  - FPGA capacity grows significantly
- **PaSiVe – Parameterization, Simulation and Verification**
  - A tool for facilitate collaborative fast prototyping
  - Using the Matlab / Simulink design flow
  - PaSiVe manages multiple design files
  - PaSiVe supports modular design
  - PaSiVe organize simulations
  - PaSiVe streamlines verifications

Design File Organization in PaSiVe

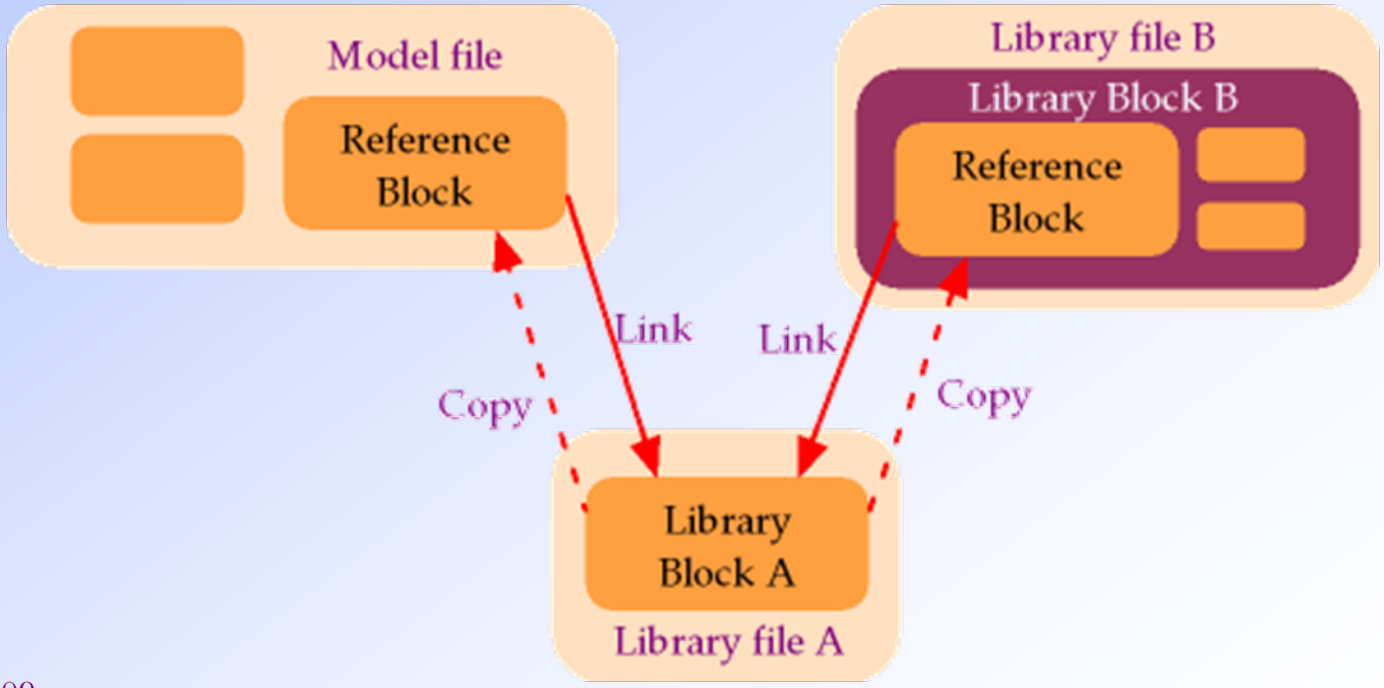


Modular Design in PaSiVe

- **Modular Design in Fast Prototyping**
  - Enable parallel development on multiple modules
  - Simplify verification, debug and correction
  - Flexible to add, delete and change module functionalities
- **Realize Modular Design in Matlab**
  - Straightforward with Matlab functions
  - Each module is implemented in a separate Matlab function file
  - Each module has its own parameter files and verification files
  - Hierarchy among modules is reflected in nested function calls
- **Implement Modular Design in Simulink**
  - Not obvious
  - Simulink simulation requires one file to contain the full system

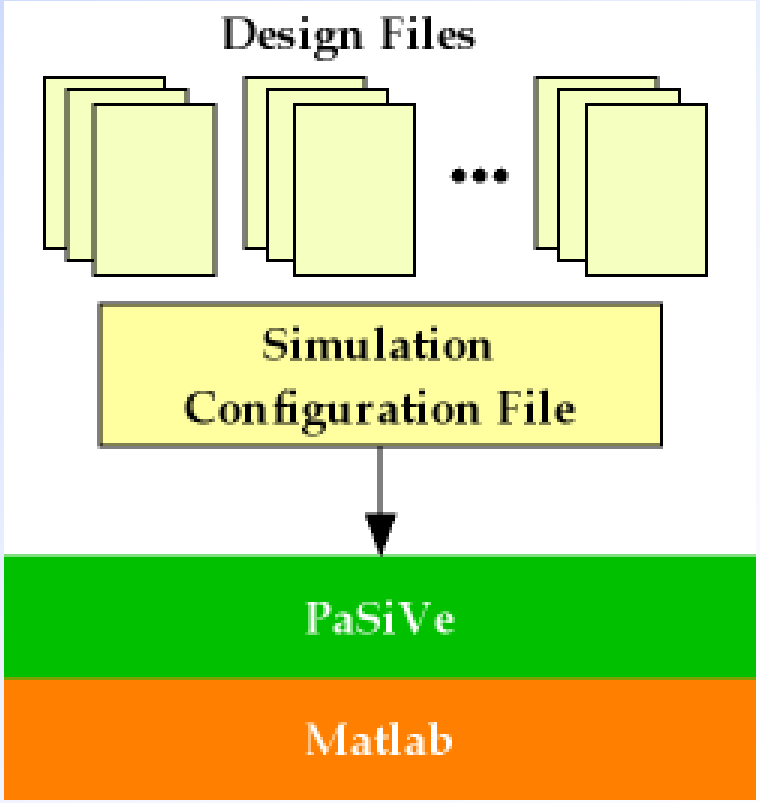
Modular Design in PaSiVe

- **Using Simulink Library Link for Modular Design**
  - Each module is implemented as a Simulink library block
  - A library block can include links to other library blocks
    - Reflecting modular hierarchy
  - The full system Simulink model includes links to all modules' libraries

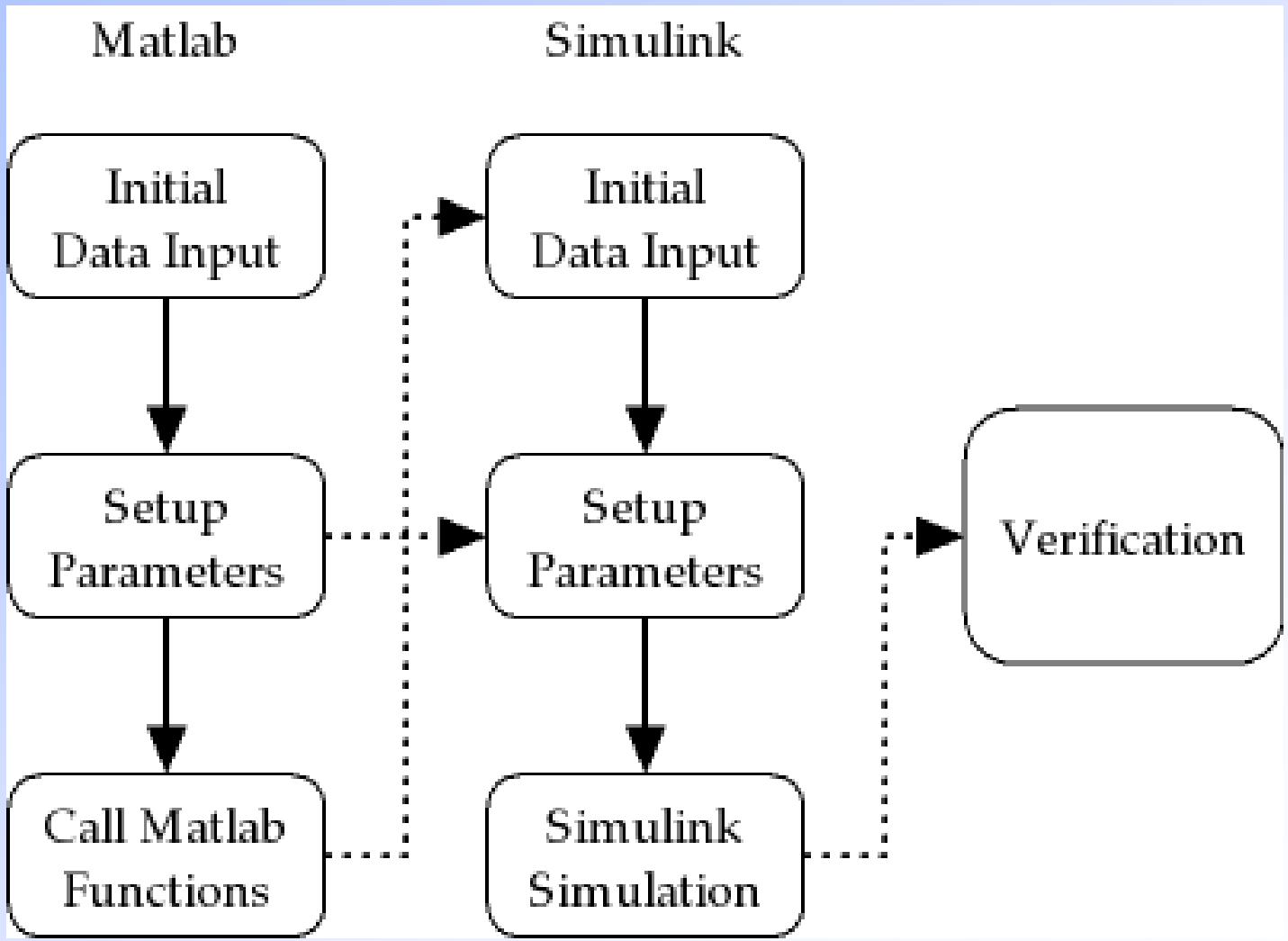


Simulation in PaSiVe

- **Modes of Simulation in PaSiVe**
  - Based on prototyping phases
    - Matlab only simulation
    - Simulink only simulation
    - Matlab-Simulink joint simulation
  - Based on simulated system scope
    - Full system simulation
    - Partial system simulation
    - Single module simulation
- **PaSiVe Simulation Mechanism**
  - Overlay on top of vender's simulation
  - Use 'Simulation Configuration file'
    - psvset.m
  - Support Monte Carlo simulation



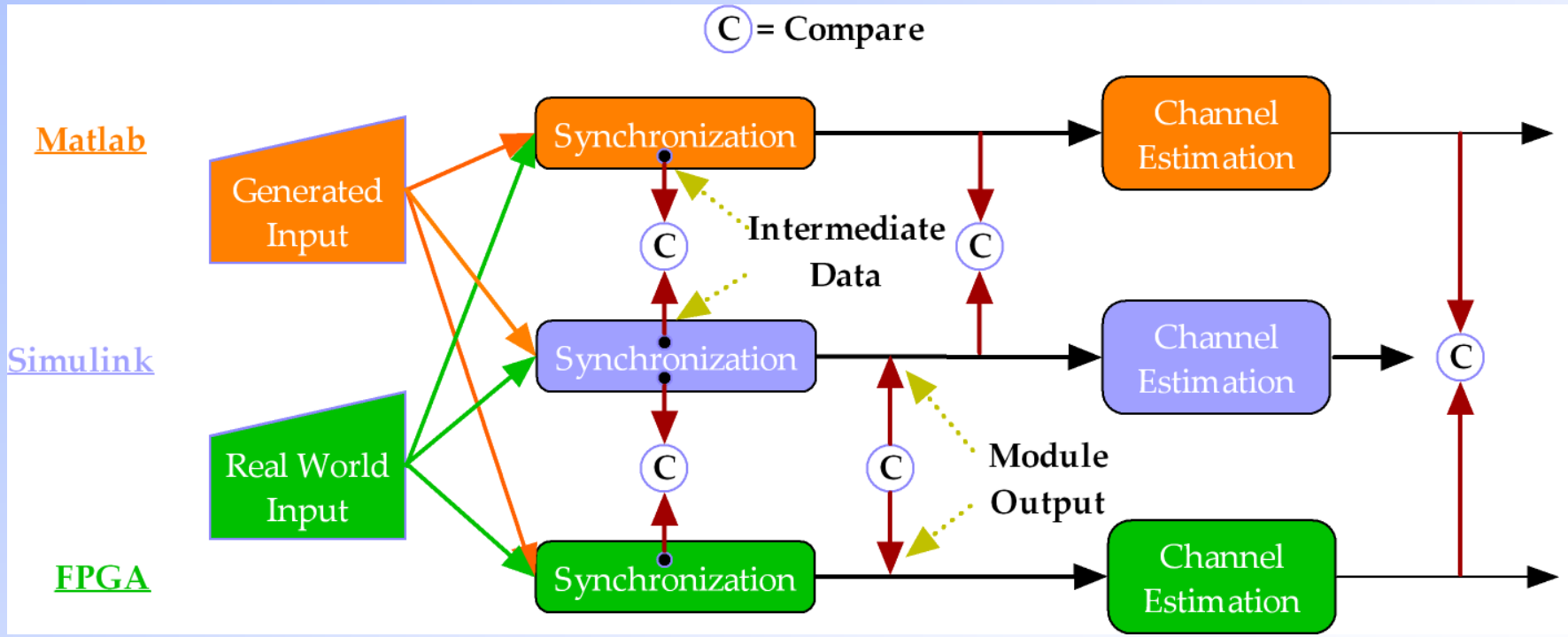
Simulation in PaSiVe



Verification in PaSiVe

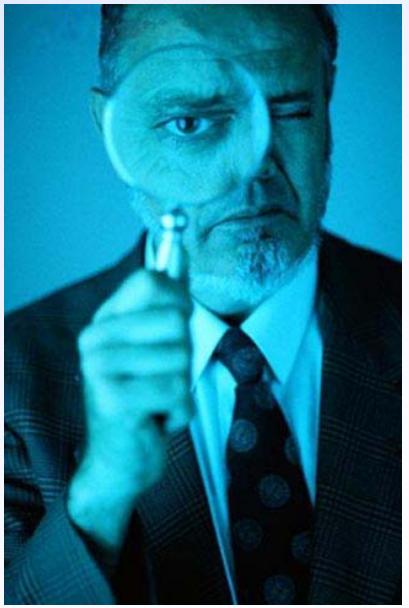
- **Pre-Generation Verification**
  - Happen before generating the hardware bitstream
  - Verify bit-true cycle-true Simulink model against the Matlab description
  - Use Matlab and Simulink joint simulation
    - Same parameter set
    - Same initial inputs and conditions
- **Post-Generation Verification**
  - Happen after real-time FPGA execution
  - Check the FPGA operation accuracy
  - Rely on real data collected from FPGA operation
  - Use Matlab simulation to speed up verification
    - Simulink simulation is slow
    - Simulink is invoked when Matlab simulation shows errors

Verification in PaSiVe



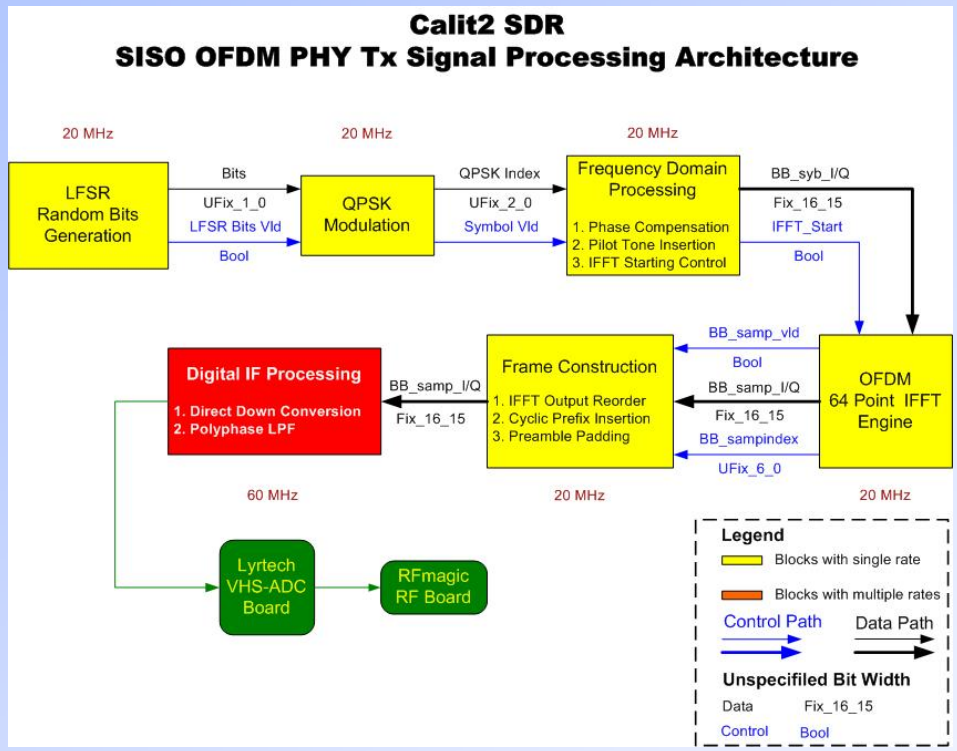
Verification in PaSiVe

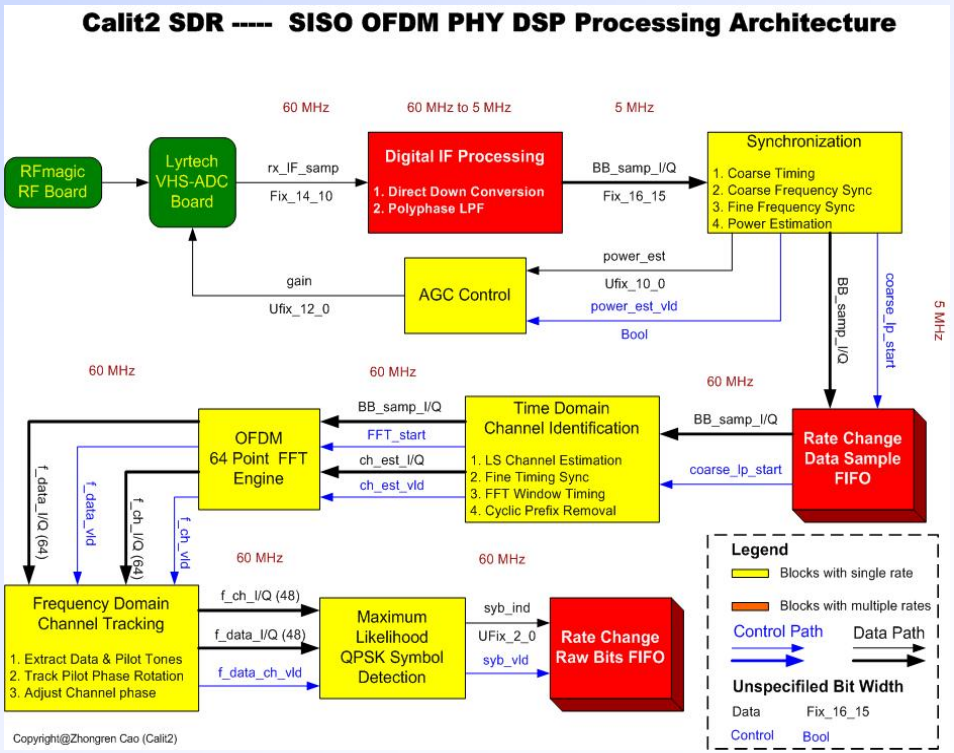
- **Verification Files**
  - Designed by developers
  - Called by PaSiVe after simulation
  - Data translation is needed
    - From multi-dimensional matrices to sequential arrays
    - From untimed to times
    - From floating point to fixed point
  - Data translation can
    - Verify control logic design
    - Match delay pipelining



Example Project

- Over-the-air SISO-OFDM packet transmission





Example Project