Extending Model-Based Design for HW/SW Design and Verification in MPSoCs

Jim Tung
MathWorks Fellow
jim@mathworks.com
Model-Based Design:
From Concept to Production

- Automate regression testing
- Detect design errors
- Support certification and standards
- Generate efficient code
- Explore and optimize implementation tradeoffs
- Model multi-domain systems
- Collaborate across teams and continents
- Automate regression testing
- Detect design errors
- Support certification and standards
MBD for Embedded Software Development

A widely used engineering approach for designing, analyzing, and implementing embedded software

- Automotive OEMs and Suppliers
- Aero & Defense
- Industrial automation & machinery
- Robotics, and more
“Help me design, validate, and verify my algorithms for deployment on a distributed processing platform.”
Support for Symmetric/Asymmetric Multicore

Tightly coupled
Symmetric / Asymmetric Multicore
(e.g., ARM, Simulink Real-Time)

Loosely coupled
Software+Hardware
Multiprocessor
Compute Clusters

Concurrent Execution Dialog
Core occupancy report
Support for Processor+FPGA Architectures

Tightly coupled

Symmetric / Asymmetric Multicore
(e.g. ARM, Simulink Real-Time)

Software+Hardware
(e.g. Xilinx Zynq, Simulink Real-Time w/ FPGA)

Multiprocessor
(e.g. ARM + DSP, multiple Simulink Real-Time)

Loosely coupled

Compute Clusters

R2011b xPC Target + FPGA
R2013a Xilinx Zynq
MBD for Hardware Development
From Concept to Production

- Includes newer capabilities
- Built on the same MATLAB and Simulink foundation
- Has established success
  - Automotive OEMs and Suppliers
  - Aero & Defense
  - Communications and electronics
FLIR Accelerates Development of Thermal Imaging FPGA

Challenge
Accelerate the implementation of advanced thermal imaging filters and algorithms on FPGA hardware

Solution
Use MATLAB to develop, simulate, and evaluate algorithms, and use HDL Coder to implement the best algorithms on FPGAs

Results
- Time from concept to field-testable prototype reduced by 60%
- Enhancements completed in hours, not weeks
- Code reuse increased from zero to 30%

“With MATLAB and HDL Coder we are much more responsive to marketplace needs. We now embrace change, because we can take a new idea to a real-time-capable hardware prototype in just a few weeks. There is more joy in engineering, so we’ve increased job satisfaction as well as customer satisfaction.”

Nicholas Hogasten
FLIR Systems

Link to user story
Wolfson Microelectronics Accelerates Audio Hub Design Verification

Challenge
Develop a multipath, multichannel audio hub for smartphones

Solution
Use Simulink to model and simulate the DSP design and use HDL Coder to generate bit-true Verilog models for verification of the digital implementation

Results
- Months of hand-coding eliminated
- Datapath verification coverage increased to 100%
- Debugging process accelerated by 20%

“For development of the world’s first highly optimized digital audio hub solution, Simulink and HDL Coder were the best options. The design and verification flow we applied using MathWorks tools scales well and provides the route to build more complex DSP and signal mixing paths.”

Brian Paisley
Wolfson Microelectronics

Link to user story
Support for Processor+FPGA Architectures

- **Tightly coupled**
  - Symmetric/Asymmetric
    - Multicore
      - (e.g. ARM, Simulink Real-Time)
  - Software+Hardware
    - (e.g. Xilinx Zynq, Simulink Real-Time w/ FPGA)

- **Loosely coupled**
  - Multiprocessor
    - (e.g. ARM + DSP, multiple Simulink Real-Time)
  - Compute Clusters

- **xPC Target + FPGA**
- **ZedBoard**
- **Xilinx Zynq**
Model-Based Design for Zynq: New Challenges

- How do I model interconnect?
- How do I partition my design?

- How do I generate IP Core with AXI interface?
- How do I write the driver for the IP Core?

- How do I easily integrate the IP Core with the synthesis tool like Vivado?
Partitioning and mapping for distributed execution

1. Express concurrency (target independent)

2. Express target architecture and configuration

3. Map & evaluate

Diagram showing the flow of tasks and components, including F1, F2, F3, F4, F5, F6, Scheduler, Core, FPGA, CPU, and Channel.
Example: Design a motor controller

- Processing System
  - 1 kHz
  - Cortex™-A9
  - Velocity Control
  - Velocity Estimate
  - Open-source LINUX

- Programmable Logic
  - 10 MHz
  - Six-Step Commutation
  - PWM
  - Hall
  - Period

- Motor FMC Card
  - Isolation
  - Inverter Module
  - Hall Interface

- Ethernet

- C code
- HDL code
Design with simulation

Simulation

Prototype

Production

Simulink

Algorithm Model

Algorithm Model

Motor Model
Design with simulation

System Simulation Testbench Model:
Provides test stimulus, integrates controller with physical model of plant, predicts system dynamics with continuous time solver, instruments/logs signals

Algorithm C Specification Model

Algorithm HDL Specification Model
Reuse components to prototype on hardware

Simulation
- Simulink
  - Algorithm Model

Prototype
- ARM
  - Algorithm C
  - Linux Driver
  - AXI Bus
  - AXI Interface
  - Algorithm HDL
  - Prog. Logic

Production
- Motor
  - Motor Model

Graphical representation with arrows indicating:
- Embedded Coder
- HDL Coder
Generate bitstream and interface

Simulation

- Simulink
  - Algorithm Model
    - Motor Model

Prototype

- ARM
  - Algorithm C
    - Linux Driver
  - AXI Bus
    - AXI Interface
    - Algorithm HDL
      - Prog. Logic

Production

- HDL Coder
- Embedded Coder

Model

Algorithm

Prog. Logic

Motor

Motor Model
Generate bitstream and interface

**Algorithm HDL Specification Model + Zynq Support Package:**
Generates algorithmic code, wraps into an IP Core, and integrates into MathWorks provided project for programmable logic.

---

**HDL Coder + Zynq Support Package**

---

**Algorithm HDL Bitstream**

---

**Algorithm HDL Interface Model**
Integrate Xilinx Vivado IP Integrator tool flow into HDL Workflow Advisor

Insert the generated IP core into Vivado Zynq system design

Build and Program Zynq board
Target Independence AND Optimization Options

- Expose vendor-provided attributes
- Currently implemented for some blocks
- Infrastructure to support more, available to block authors

```vhDL
ARCHITECTURE rtl OF symmetric_fir IS
    ATTRIBUTE use_dsp48 : string;
    ATTRIBUTE use_dsp48 OF rtl : ARCHIT

    -- Signals
    SIGNAL x_in : signed;
    SIGNAL h_in1, h_in2, h_in3, h_in4 : signed;
    SIGNAL y_out, delayed_xout;
```

```verilog
(* use_dsp48 = "yes" *) module symmetric_fir
    (clk, reset, clk_enable, x_in,
    h_in1, h_in2, h_in3, h_in4,
    ce_out, y_out, delayed_xout);
```
Generate ARM executable

Simulation
- Simulink
  - Algorithm Model
  - Algorithm Model
  - Motor Model

Prototype
- ARM
  - Algorithm C
  - Linux Driver
  - AXI Bus
  - AXI Interface
  - Algorithm HDL
  - Prog. Logic

Production
- Embedded Coder
- HDL Coder
- AXI Interface
- Algorithm HDL
- Prog. Logic
- Motor Model
- Motor
Generate ARM executable

ARM Prototype Specification Model + Zynq Support Package
Generates algorithmic code and automates integration with MathWorks project for
ARM. Model acts as graphical user interface to hardware (switches, sliders, scopes)

Embedded Coder +
Zynq Support Package

Algorithm C
Specification Model

Algorithm HDL
Interface Model
C Code Generation for ARM Cortex-A9

- Automatic AXI Interface generation
- Multi-threaded Linux code
- Code optimizations using ARM Neon intrinsics
Performance Profiling

- On desktop profiling
  - Tasks execution time on desktop machines

- On target profiling
  - Tasks execution time when running on target

- SIL/PIL – based profiling
  - Functions and tasks execution time when running in-the-loop type of simulations
Performance Measurements and Resource Usage

**CPU**
Execution profiler

**FPGA**
Resource report and critical path highlighting
Processor-In-the-Loop (External Mode)

- Control Algorithm
- Tune parameters
- Gather real-time results
- Control IP Core from Simulink
From simulation to prototype to production
And more?

Tightly coupled

Symmetric / Asymmetric Multicore

(e.g., ARM, Simulink Real-Time)

Loosely coupled

Software+Hardware

(e.g., Xilinx Zynq, Simulink Real-Time w/ FPGA)

Multiprocessor

(e.g., ARM + DSP, multiple Simulink Real-Time, other MPSoC platforms)

Compute Clusters

Target authoring
Tooling for MPSoC Development

Opportunities and Questions
Summary

- Model-Based Design, using MATLAB and Simulink, is well-established for embedded software development
- MBD use is growing for FPGA prototyping and development
- MATLAB and Simulink now have fundamental capabilities for hardware/software co-design

- We seek your input:

What do you need in a MPSoC development platform?