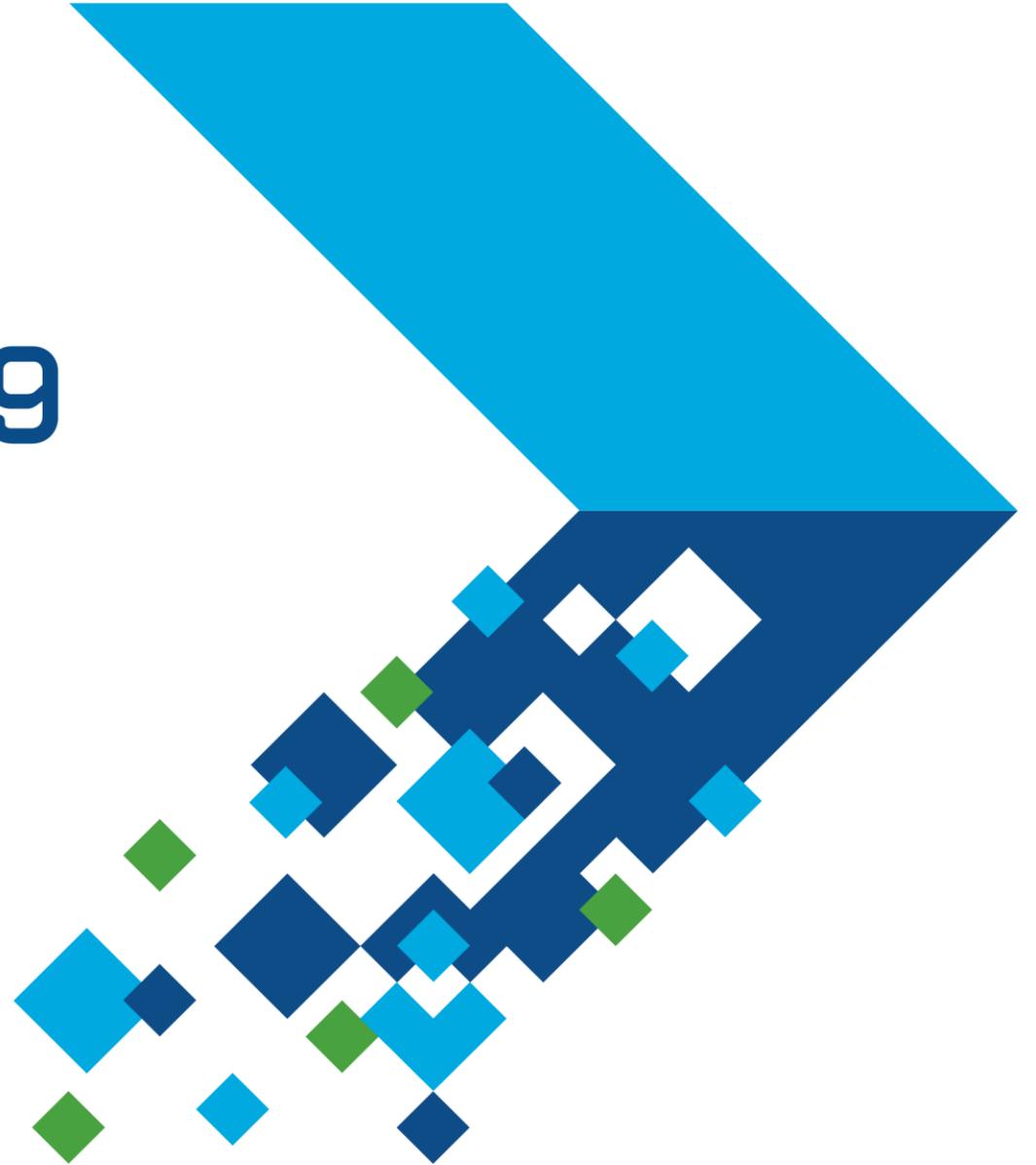


MATLAB EXPO 2019

SoC Blockset 소개

정승혁

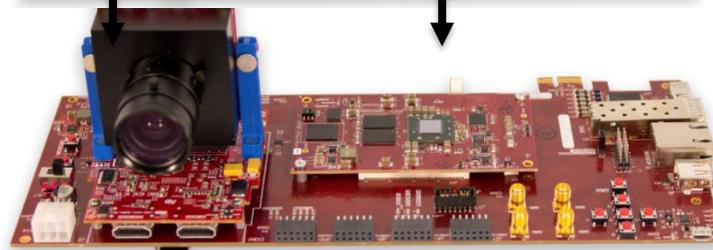
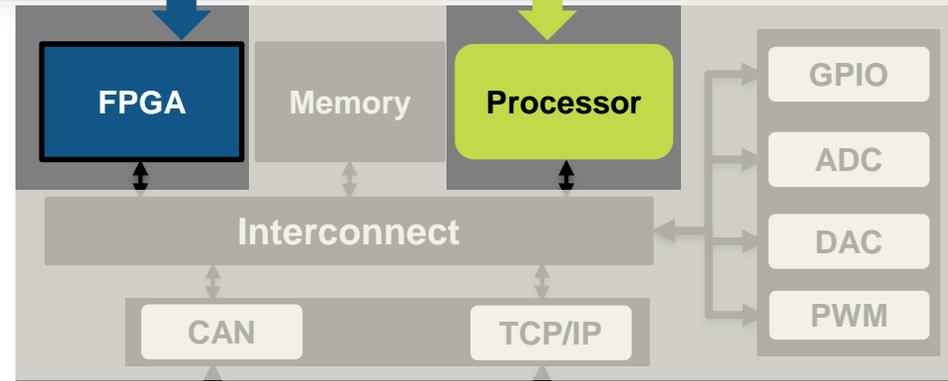
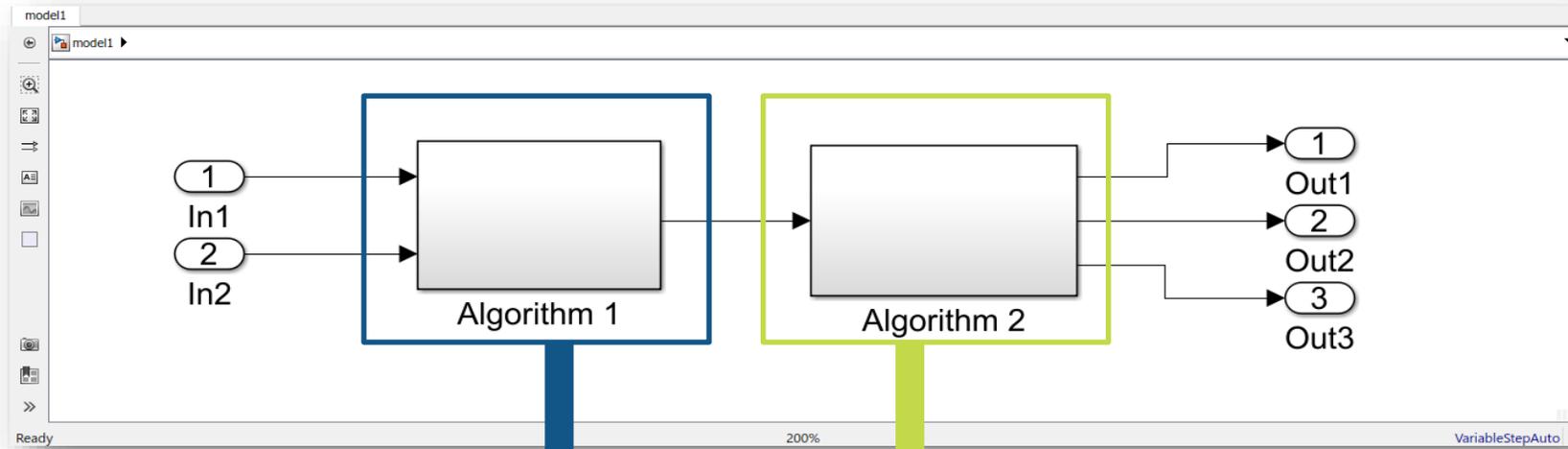


Agenda

- 
- What is SoC Blockset?
 - Traditional Workflow and New Workflow for SoC Design
 - Features of SoC Blockset
 - **Simulate** SoC Architectures
 - **Analyze** System Performance
 - **Deploy** to SoC and FPGA Devices

Model Based Design Workflow for SoC

Deploy to Hardware with Coders and Hardware Support Packages

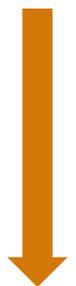


Algorithmic Model



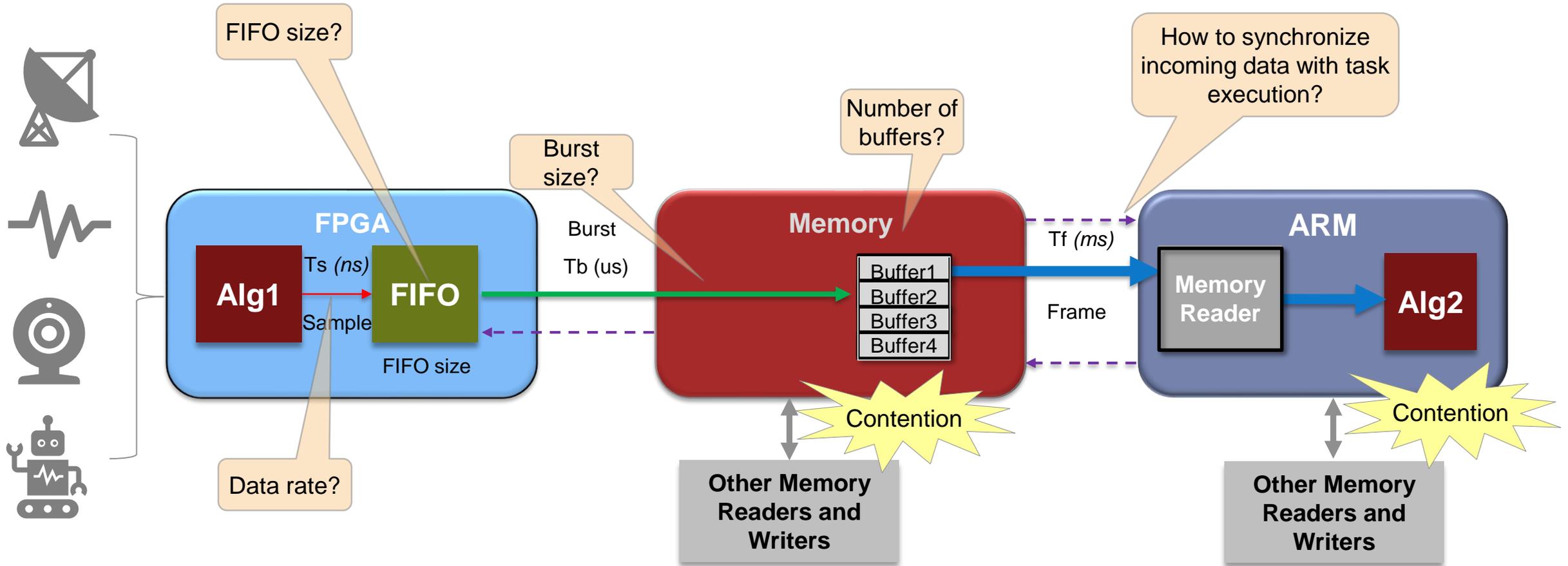
Algorithmic Code

HSP (Reference Design)



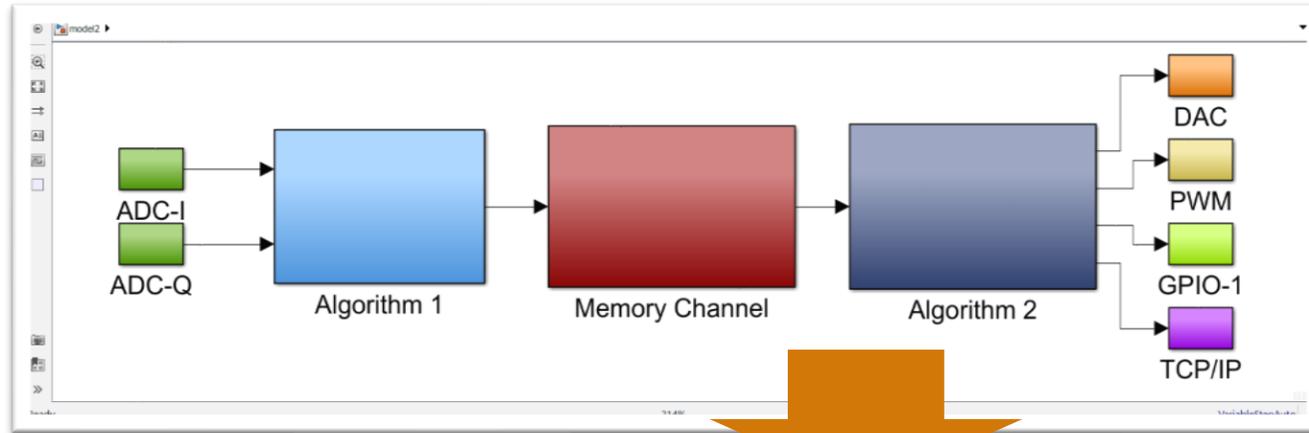
Hardware Platform

Actual Data Exchange Between FPGA and Processor

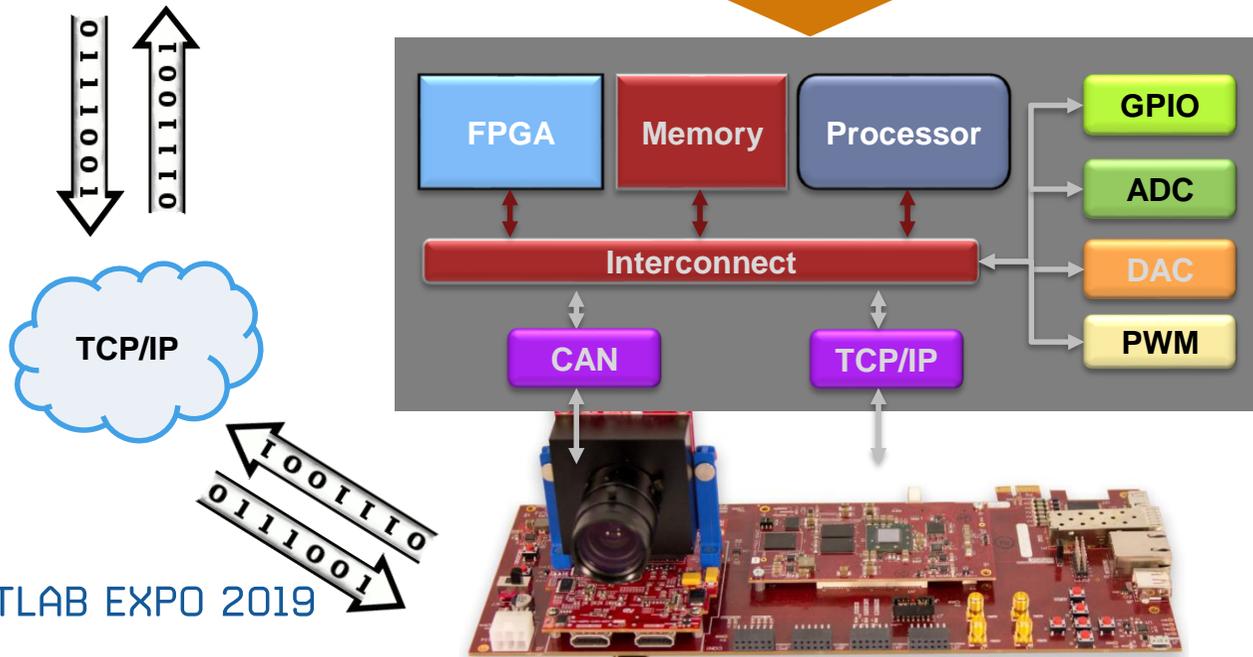


Model Based Design Workflow for SoC

Hardware Architecture Simulation with SoC Blockset

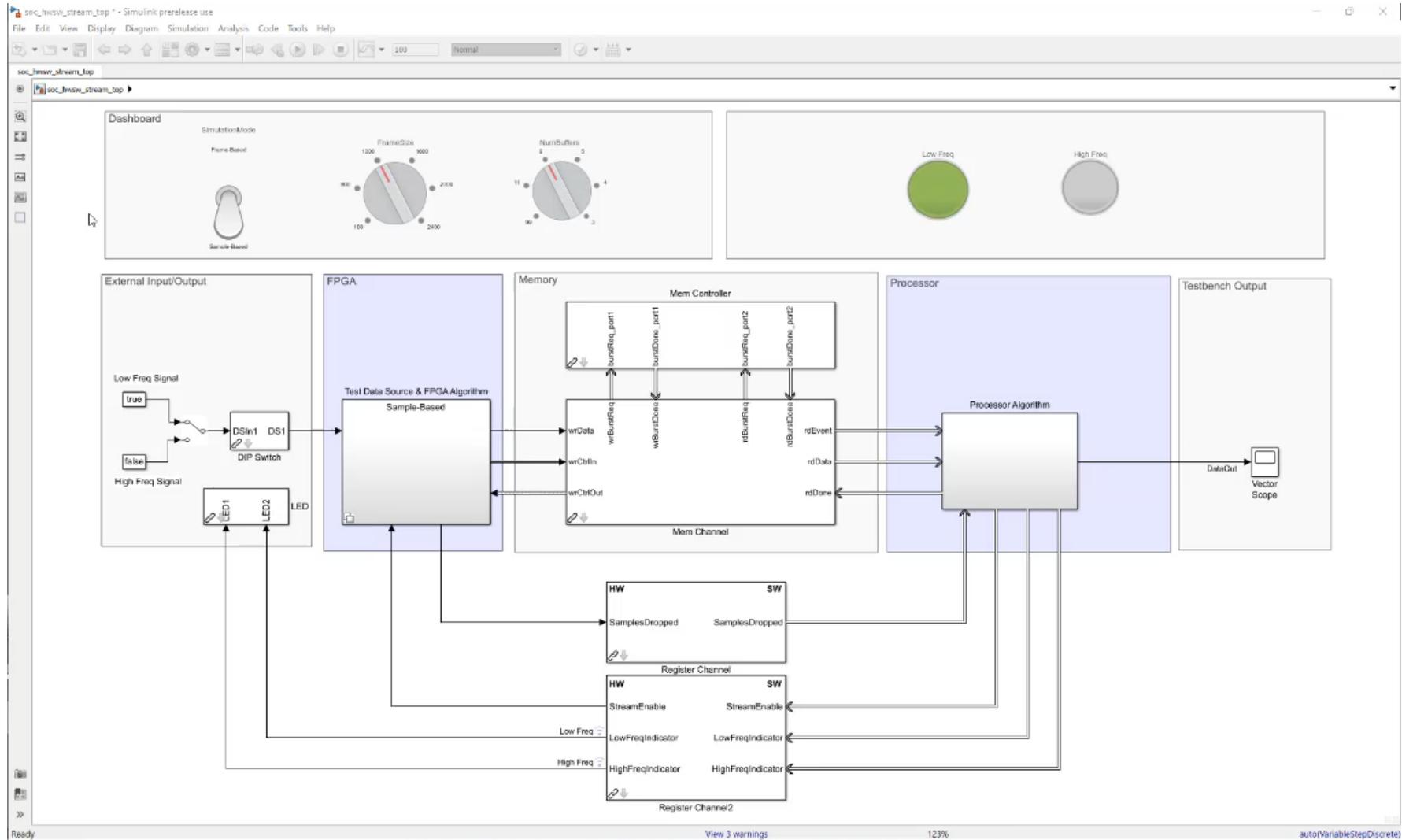


- Simulate algorithms as well as hardware/software architecture
 - ✓ Memory
 - ✓ Internal/external connectivity
 - ✓ I/O
 - ✓ Task scheduling



- Deploy on support hardware
- Profile performance using external mode

Short Demo Video



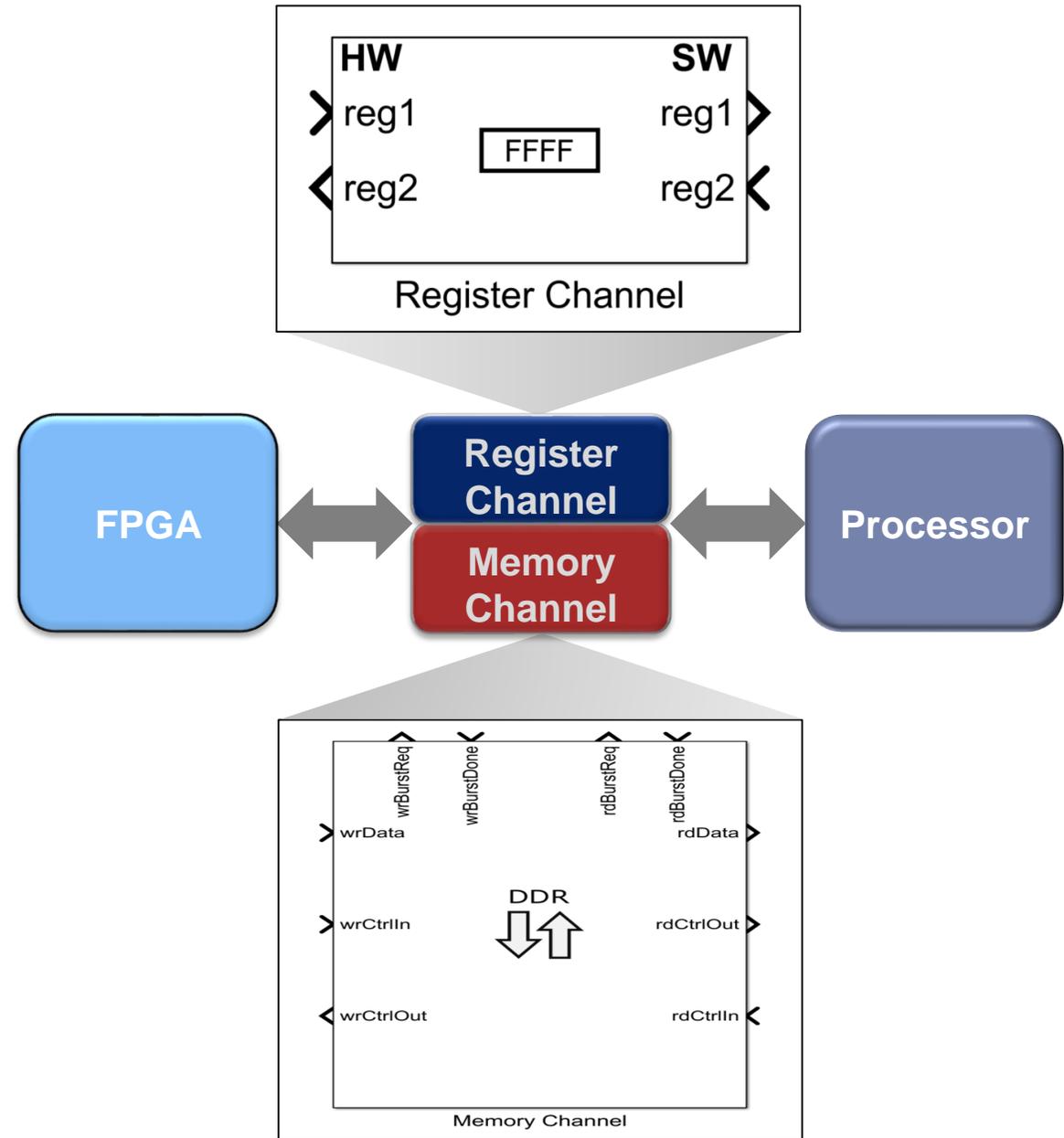
Agenda

- What is SoC Blockset?
 - Pains and the new solution
- » ▪ Features of SoC Blockset
 - **Simulate** SoC Architectures
 - **Analyze** System Performance
 - **Deploy** to SoC and FPGA Devices

Simulate SoC Architectures

Modeling Data Transfer

- **Memory channel blocks**
 - Register
 - Shared memory
- **Multiple protocols**
 - AXI4-Stream to Software via DMA
 - AXI4-Stream FIFO
 - AXI4-Stream Video FIFO
 - AXI4-Stream Video Frame Buffer
 - AXI4-Random Access



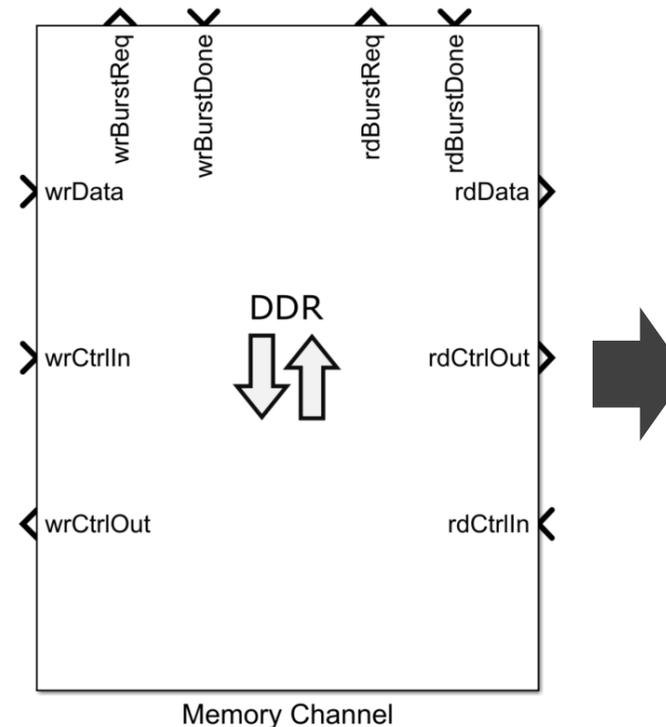
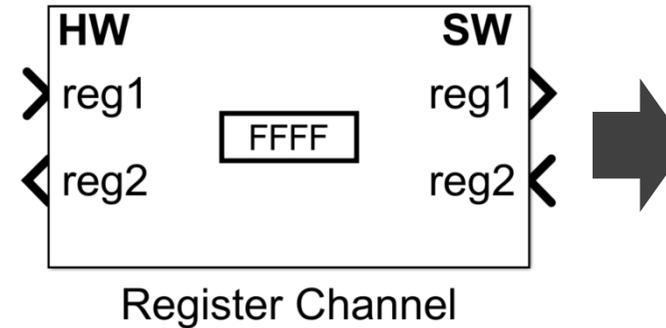
Modeling Data Transfer

- Memory channel blocks

- Register
- Shared memory

- Multiple protocols

- AXI4-Stream to Software via DMA
- AXI4-Stream FIFO
- AXI4-Stream Video FIFO
- AXI4-Stream Video Frame Buffer
- AXI4-Random Access



Block Parameters: Register Channel

Register Channel

Model transfer of register values between processor and hardware logic.

Add or remove ports using New and Delete buttons, respectively. Arrange the order of ports using Move Up and Move Down buttons.

Parameters

Registers

Filter table contents

Register	Direction	Data Type	Dimension
reg1	Read	uint32	1
reg2	Write	uint32	1

New Edit Move Up Move Down Delete

Register write sample time: -1

OK Cancel Help Apply

Block Parameters: Memory Channel

Memory Channel (mask) (link)

Model memory accesses as read and write bursts.

Data, CtrlIn and CtrlOut ports read and write data. burstReq and burstDone ports manage requests to the Memory Controller.

Burst length, Bandwidth, FIFO depth and other parameters can be changed in the Advanced section. In the Performance tab, metrics can be visualized after running a simulation.

Hardware board: [ZedBoard](#)

[Show implementation info](#)

Main Signal Attributes Performance

Channel type: AXI4-Stream to Software via DMA

Region size (bytes): 8192

Buffer size (bytes): 1024

Number of buffers: 8

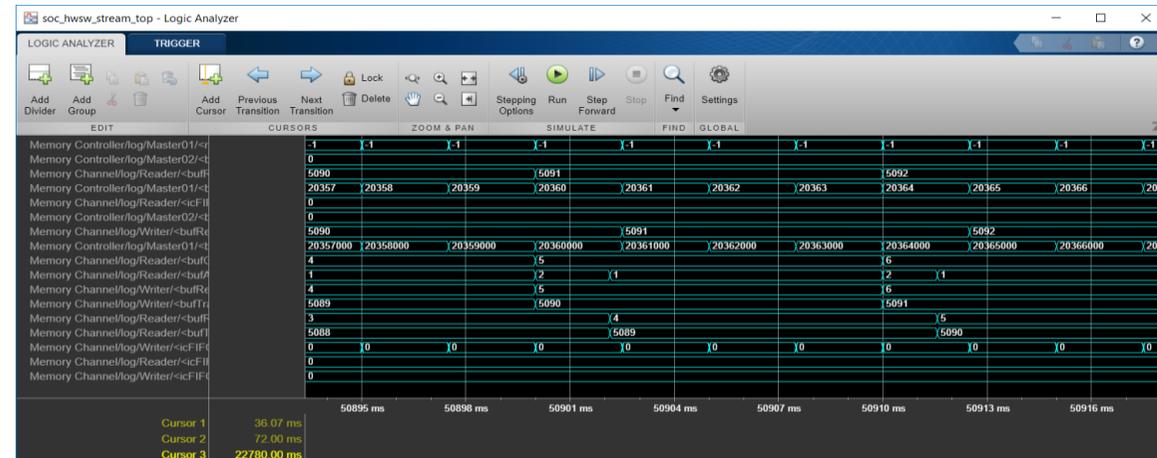
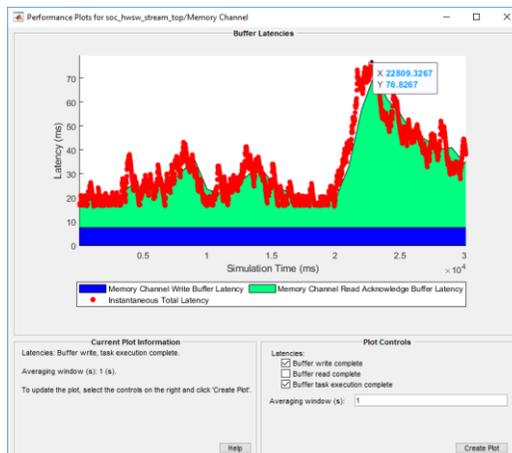
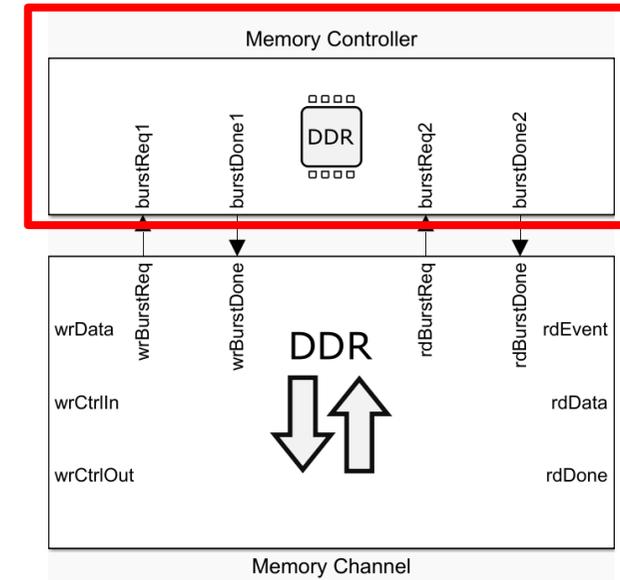
Advanced

OK Cancel Help Apply

Modeling Data Transfer

Memory controller block

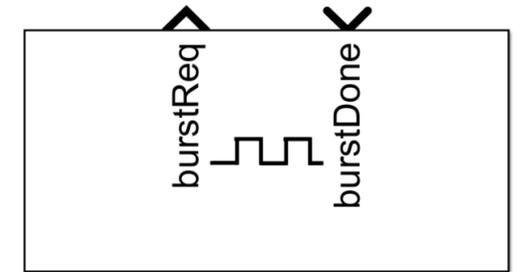
- Arbitrate access to shared memory
- Support multiple channels
- Support various arbitration protocols
- Log and display performance data
 - Latency, Burst, Bandwidth
- Visualize internal state via Logic Analyzer



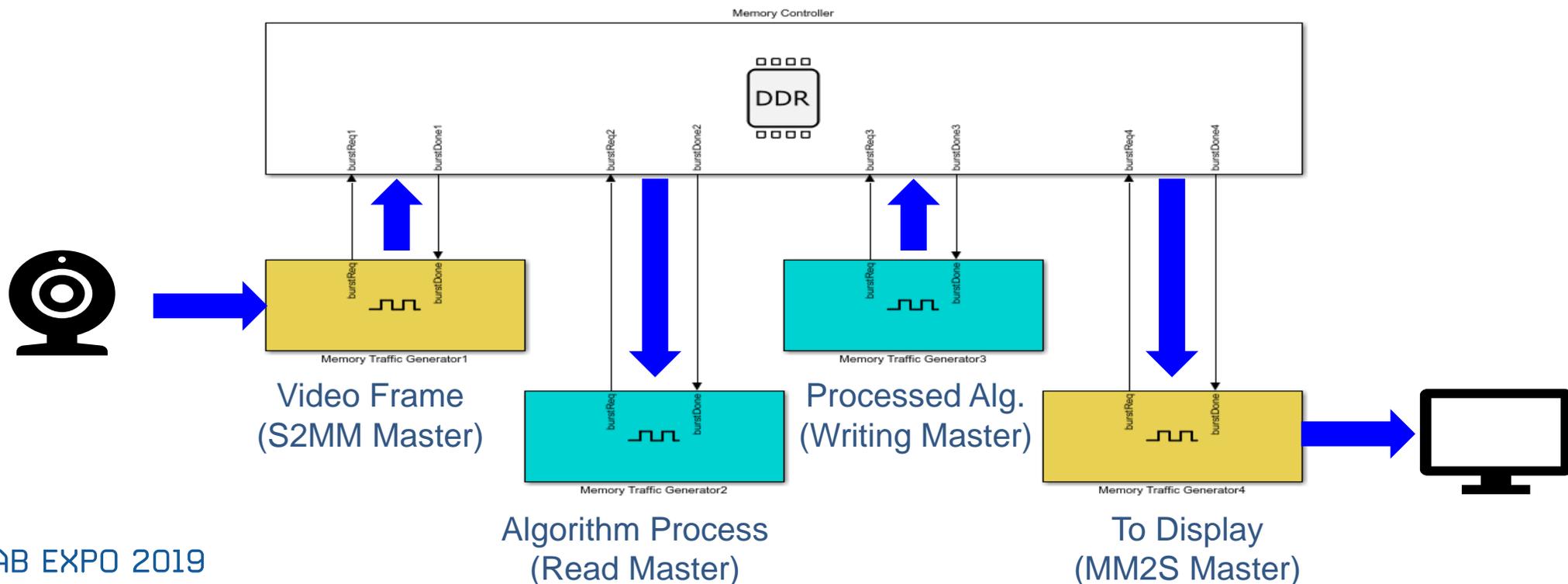
Modeling Data Transfer

- ## Memory traffic generator

- Generate read or write requests to the memory
- Model the impact of a master's memory accesses
- Characterize performance of memory subsystem under contention



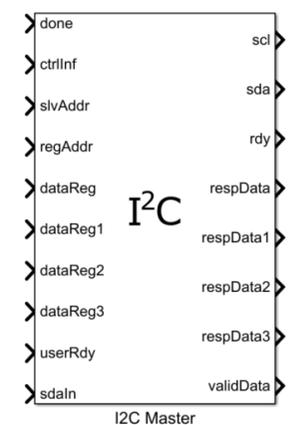
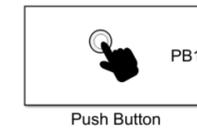
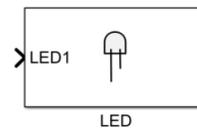
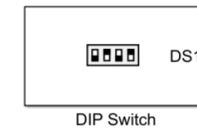
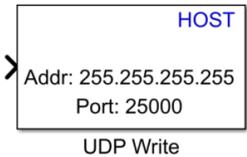
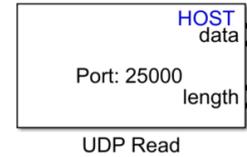
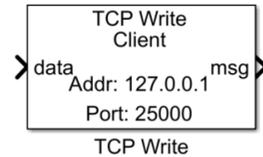
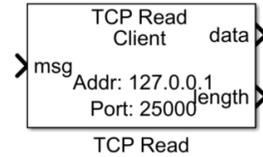
Memory Traffic Generator



Modeling I/Os

- **Processor I/O**

- TCP Read/Write
- UDP Read/Write
- Register Read/Write



- **Hardware Logic I/O**

- DIP Switch, LED, Pushbutton
- I2C Master

- **Simulation with real I/O data**

- Record real I/O data
- Simulate with record data



Block Parameters: AXI4-Stream Read

AXI4-Stream Read
Read AXI4-Stream Data using IIO

Use the Samples per frame parameter to determine the number of samples read for each DMA transfer.

Main Advanced **Simulation**

Simulation output: **From recorded file**

Dataset name: Browse...

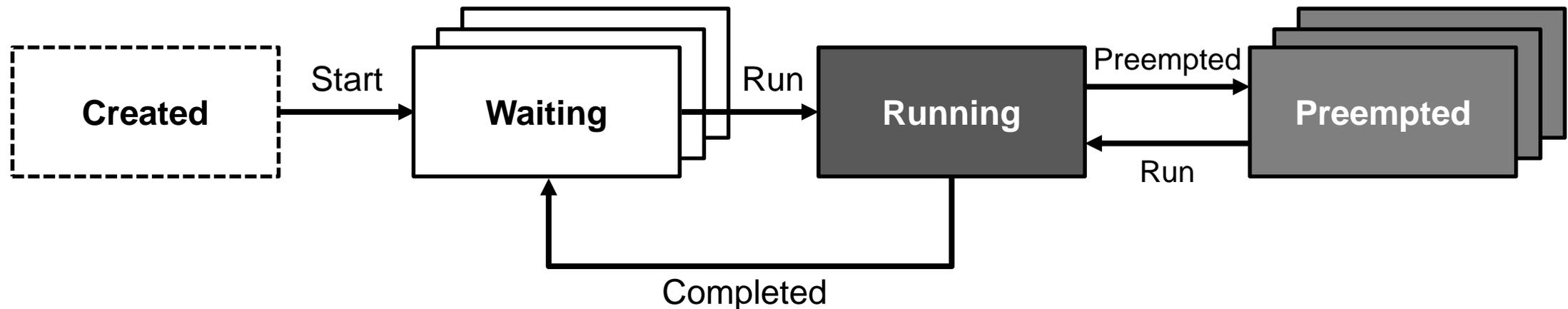
Source name: Select...

Buffer Read Event:

OK Cancel Help Apply

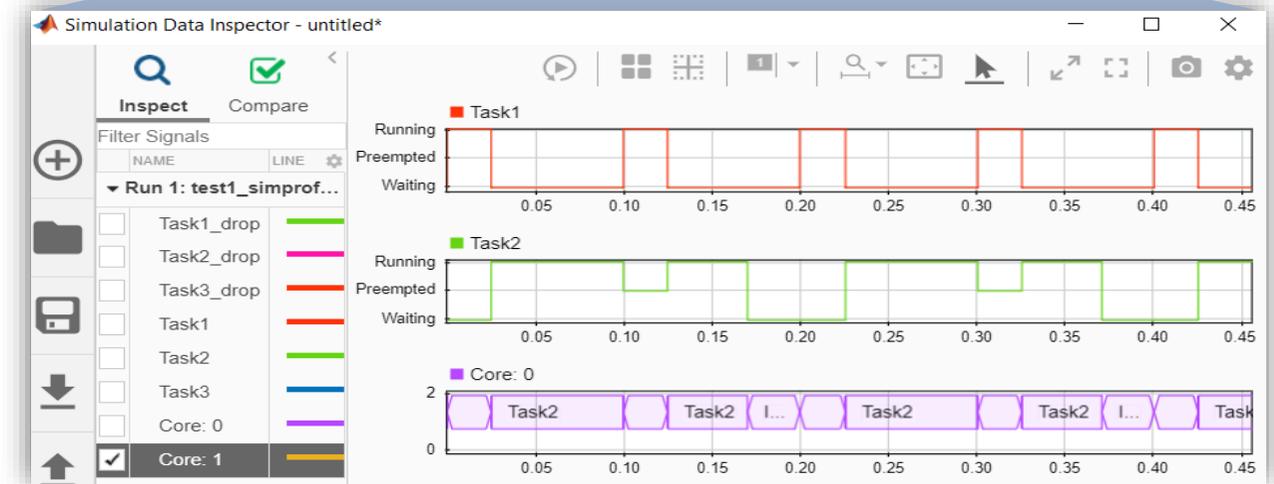
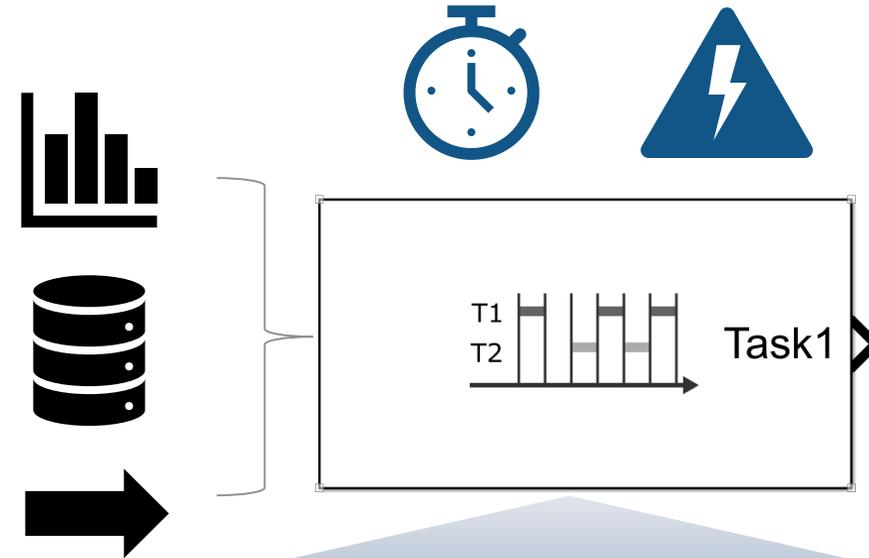
Modeling Task Execution

- Task execution on hardware is managed by Operating System
- Task is a portion of Simulink model contained within a sample rate or function-call subsystem



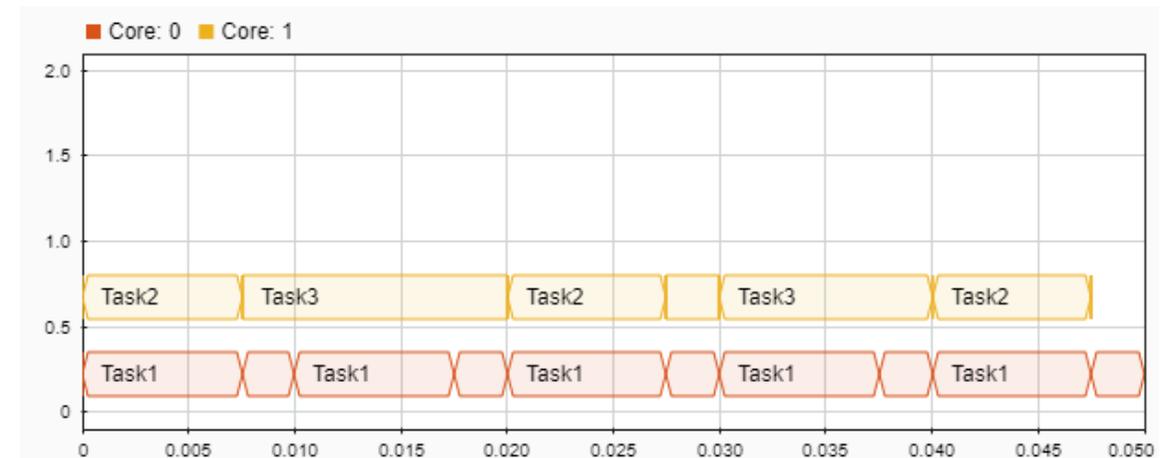
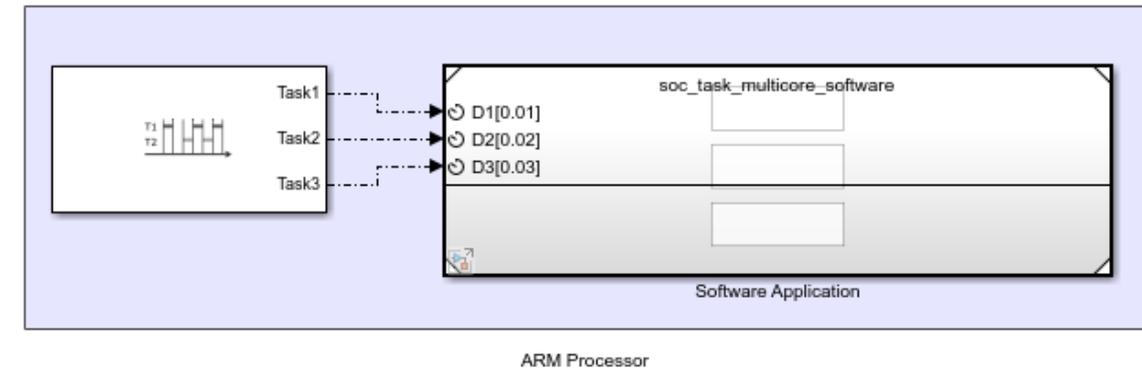
Modeling Task Execution

- Task manager block
 - Timer-Driven, Event-driven
 - Probability model
 - From a data file recording
 - Input ports on the block
 - Parameters
 - Task Period, Task Duration
 - Priority
 - Processor core
 - Task Visualization in SDI*
- *SDI: Simulation Data Inspector



Modeling Software Algorithm

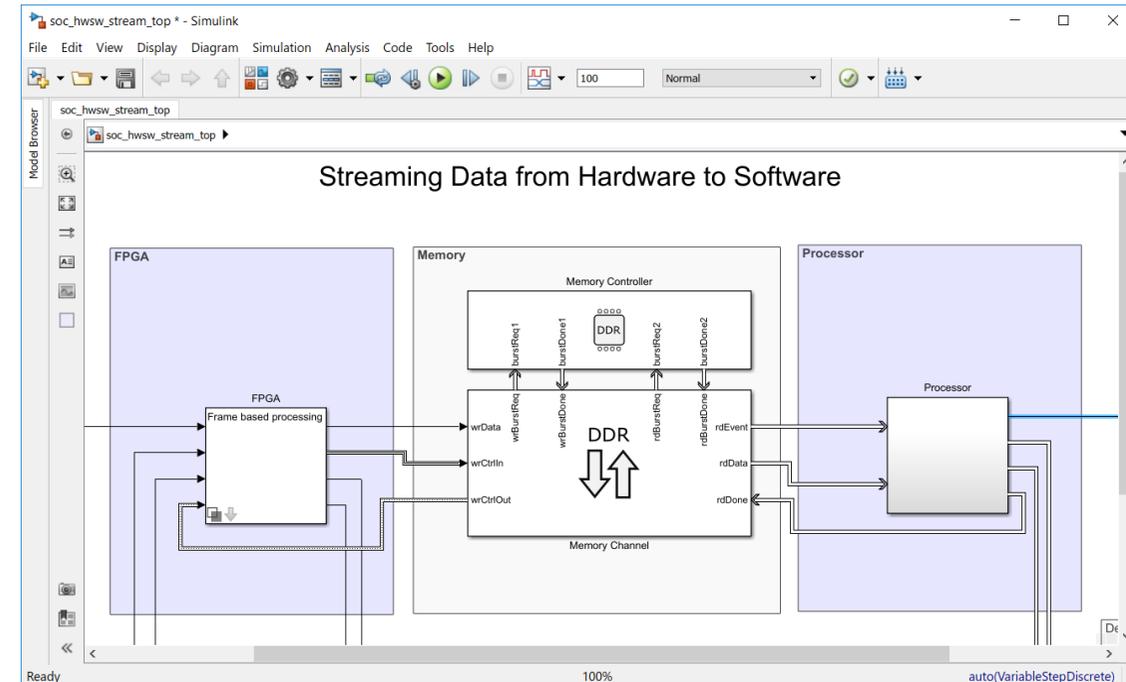
- Detect task overruns and implement countermeasures
- Visualize task priority and preemption
- Simulate multicore task execution
- Record and playback task execution in simulation



Deploy to SoC and FPGA Devices

Deploying Architecture on Hardware Boards

- Generate reference designs for FPGAs and SoCs from Xilinx and Intel
- Generate HDL for hardware algorithm (require HDL Coder)
- Generate C/C++ for processor algorithm (require Embedded Coder)



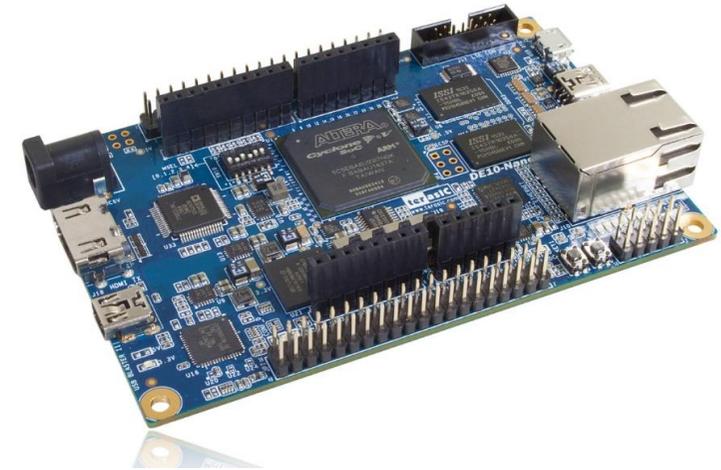
Implementing on Xilinx SoC and FPGA Platforms

- Tools
 - Vivado Design Suite 2018.2
- Boards
 - FPGA: Artix-7 35T Arty, Kintex-7 KC705
 - Zynq 7000: ZC706, ZedBoard
 - Zynq UltraScale+: ZCU102
- I/O modules
 - HDMI Tx/Rx
 - AD9361 Rx/Tx
 - ADAU17612 codec



Implementing on Altera SoC and FPGA Platforms

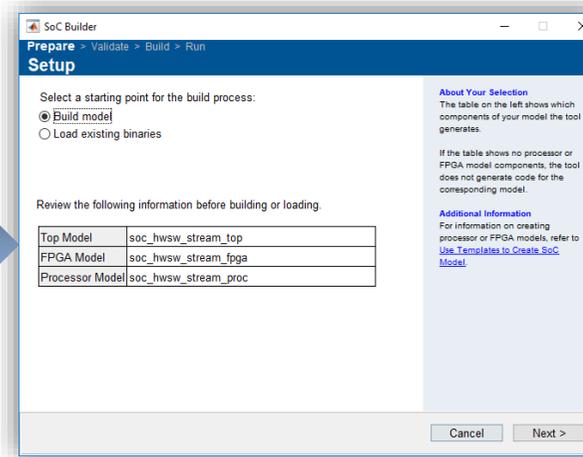
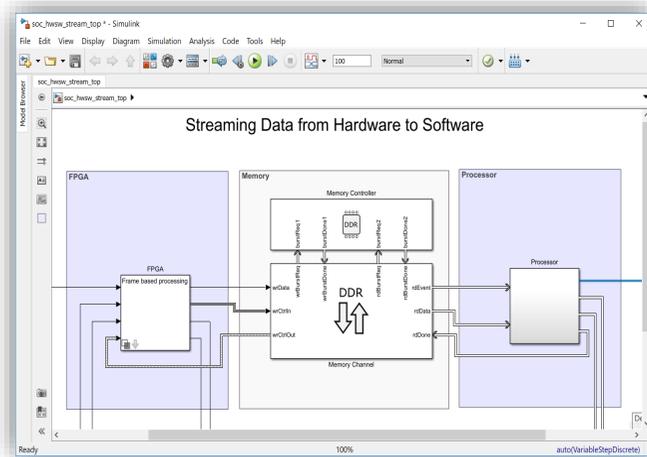
- Tools
 - Intel Quartus Prime Standard Edition 18.0
 - Intel SoC FPGA Embedded Development Suite (EDS) 18.0
- Boards
 - Arria 10 SoC Development Kit
 - Cyclone V SoC Development Kit
- I/O modules
 - None



socBuilder

SoC Builder tool steps through the various stages for building and executing an SoC model on FPGA/SoC

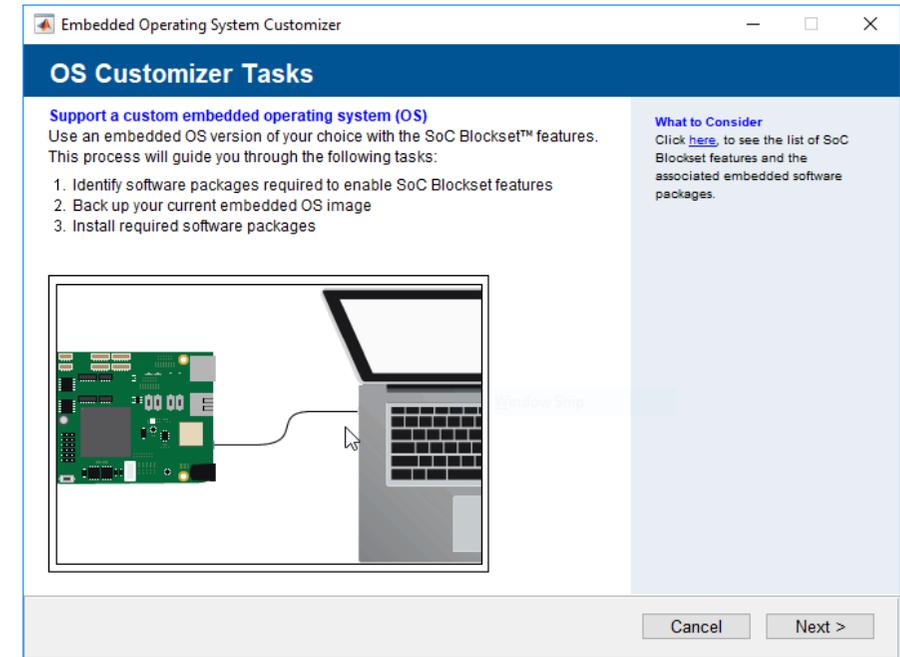
- Review the model information and memory map
- Choose build actions (Build, Load, Run)
- Build the model using Xilinx or Intel tools
- Configure the Ethernet connectivity
- Load the programming file to your FPGA board
- Run the application



socBuilder

Customizing OS on Embedded Processor

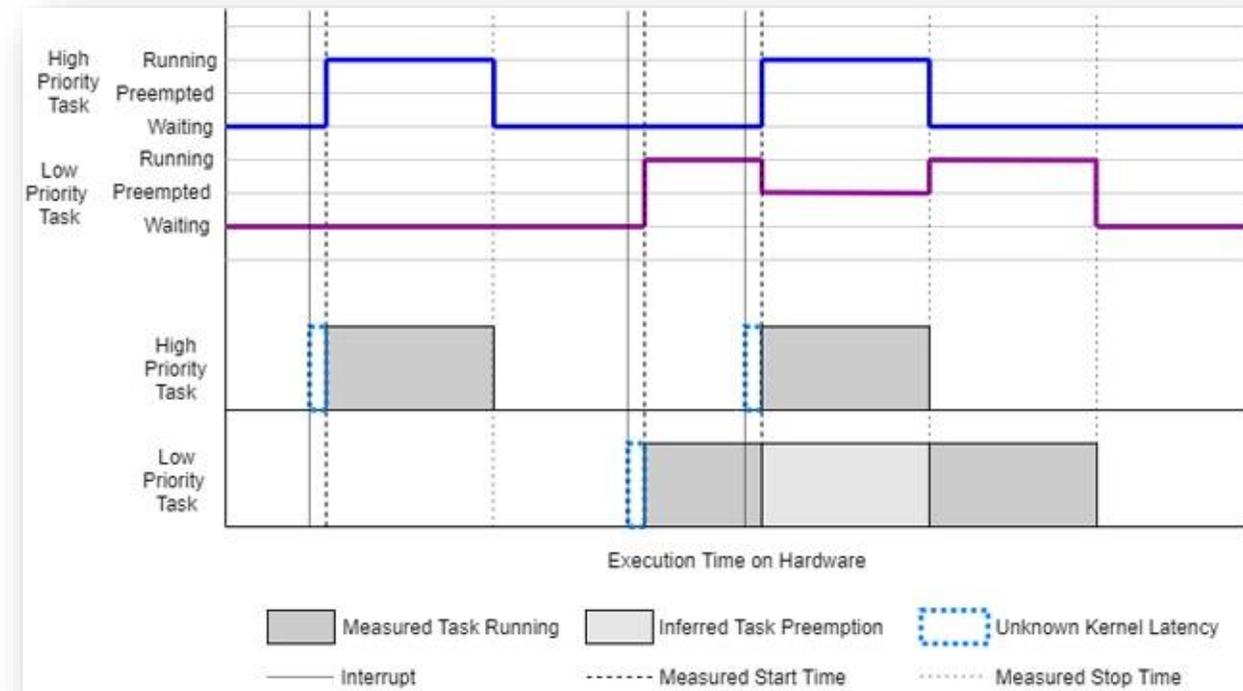
- Libraries added
 - Audio
 - Video capture
 - Data inspector
 - SDL Display
 - File transfer
 - Register read/write
 - AXI stream
- OS Customizer
 - Support for Debian, Yocto, PetaLinux, and BuildrootLinux distributions.
 - Support for OS with Package Management System
 - Support for external OS firmware image build systems (host or remote)



Analyze System Performance (On-Device Profiling)

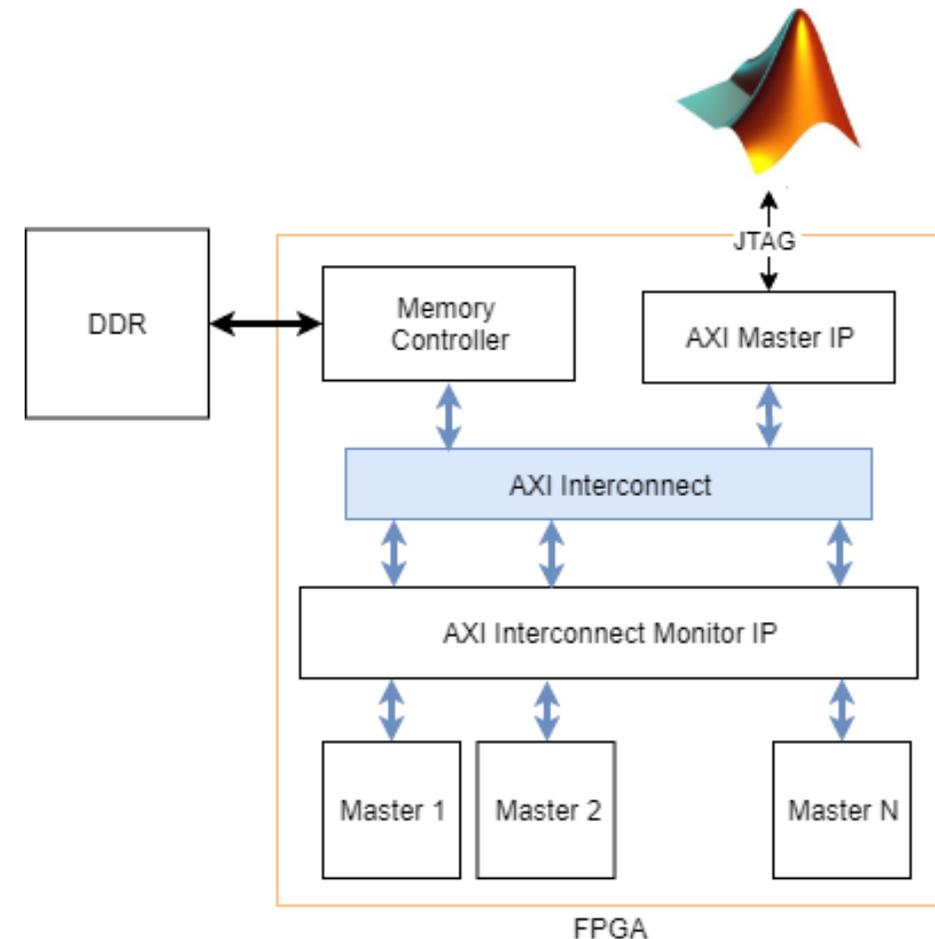
Profiling on Hardware Devices

- Code instrument profiler
 - Record the start and stop times of each task on the processor
 - Infer instantaneous state of each task
 - Does not record Kernel latency
- Light instrument code
 - Negligible impact on task execution
- Running in external mode
 - Task execution data and statistics are recorded in files



AXI Interconnect Monitor

- All memory masters in FPGA are connected to AXI Interconnect Monitor IP
- Data queried from MATLAB using JTAG
- Collect memory interconnect traffic
- Capture transaction information



Configuration for Profiling

Hardware Implementation Pane

Hardware board:

Code Generation system target file: [ert.tlc](#)

Device vendor: Device type:

▶ Device details

Hardware board settings

▼ Task profiling in simulation

- Show in SDI
- Save to file

▼ Target hardware resources

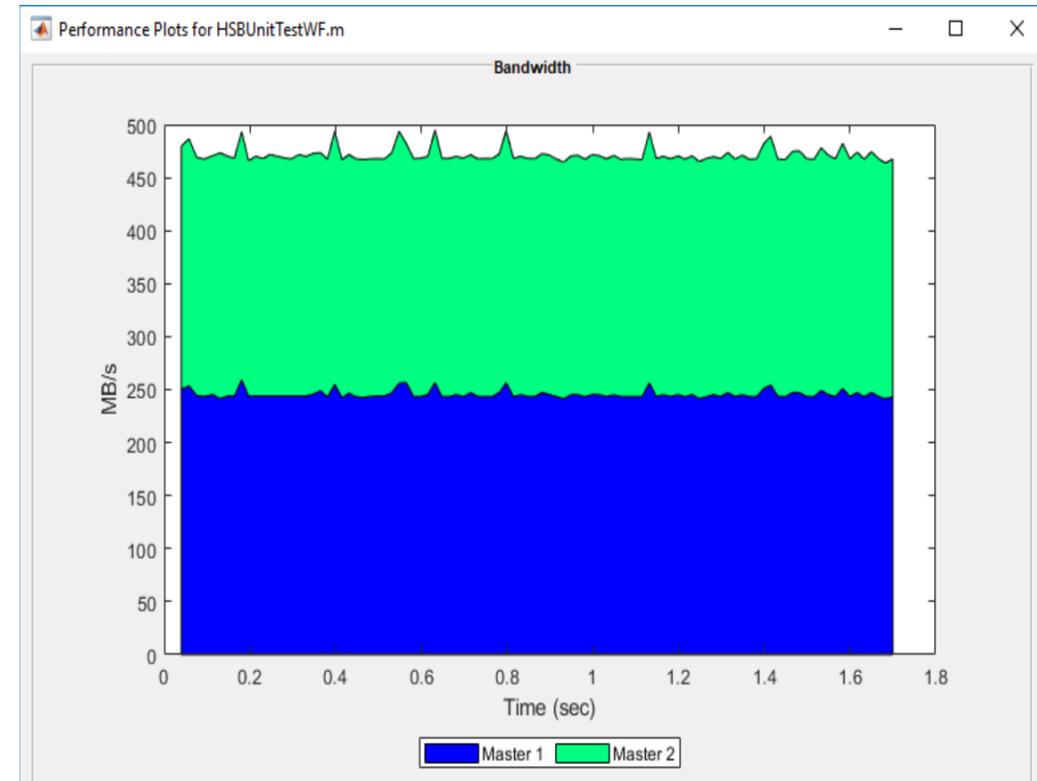
Groups

- Board Parameters
- Build options
- Clocking
- External mode
- FPGA design (top-level)
- FPGA design (mem controllers)
- FPGA design (mem channels)
- FPGA design (debug)

- Include 'MATLAB as AXI Master' IP for host-based interaction
- Include processing system
- Interrupt latency (s):
- Register configuration clock frequency (MHz):
- IP core clock frequency (MHz):

Profiling on Hardware Devices

- Bring real-time hardware diagnostics back to Simulink
 - Task execution profiling
 - Memory traffic monitoring
 - DMA buffer usage
 - CPU utilization
- Analyze and tune SoC model to meet your desired system performance
 - Run the SoC model in external mode
 - Interact in real time with an SoC device



Conclusion

With **SoC Blockset**, you can

- **Simulate** your hardware architecture with algorithms
- **Profile** software performance and hardware utilization on hardware devices
- **Deploy** on Xilinx and Intel devices

Learn More



SoC Blockset

Design, evaluate, and implement SoC hardware and software architectures

software architectures

- [SoC Blockset Webpage](#)
- [SoC Blockset Examples](#)
- [SoC Blockset Product Requirement](#)
- Supported Hardware Boards: [Xilinx](#), [Intel](#)

MATLAB EXPO 2019

데모부스와 상담부스로 질문 하시기 바랍니다.

감사합니다

