

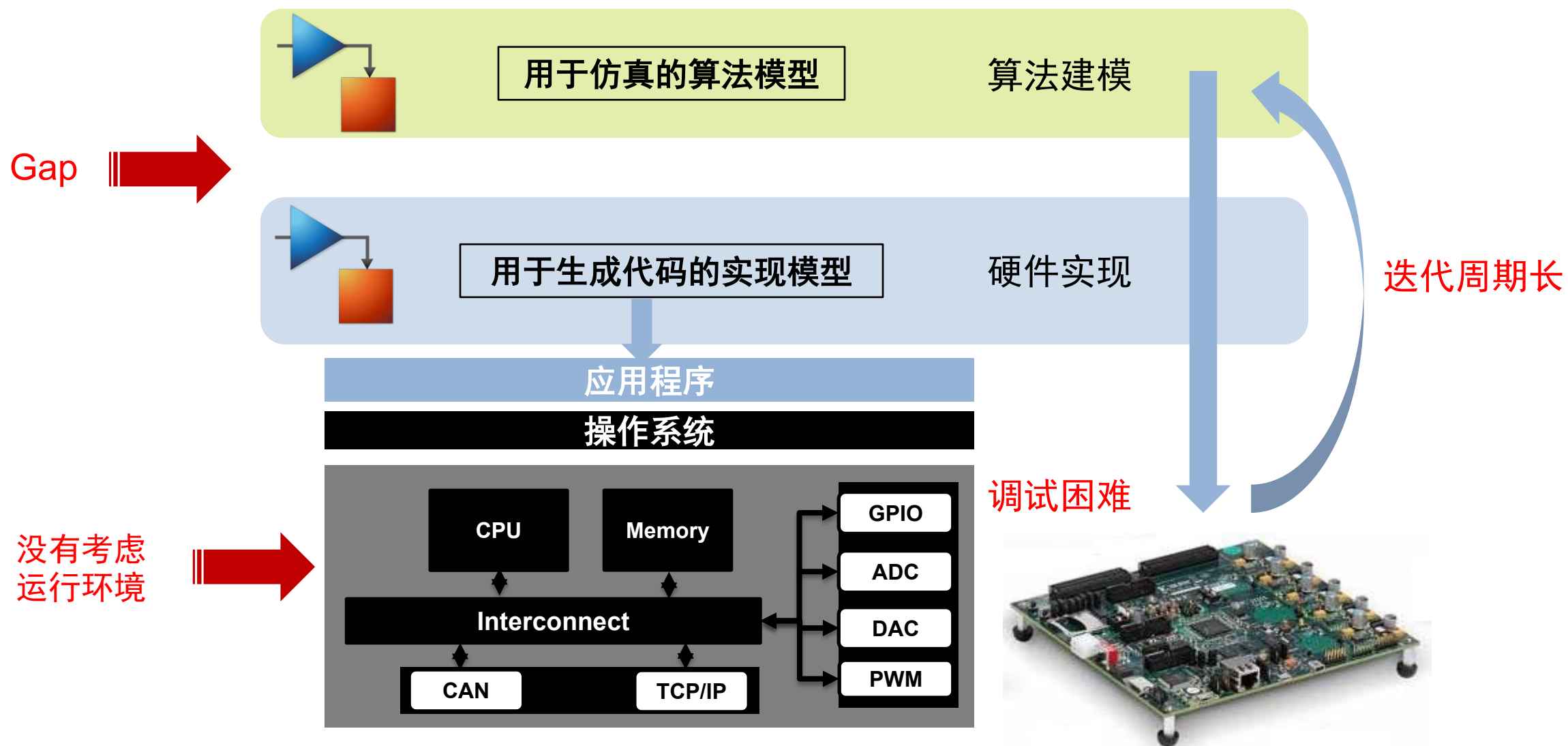
MATLAB EXPO

将算法模型快速部署到 DSP 和 FPGA

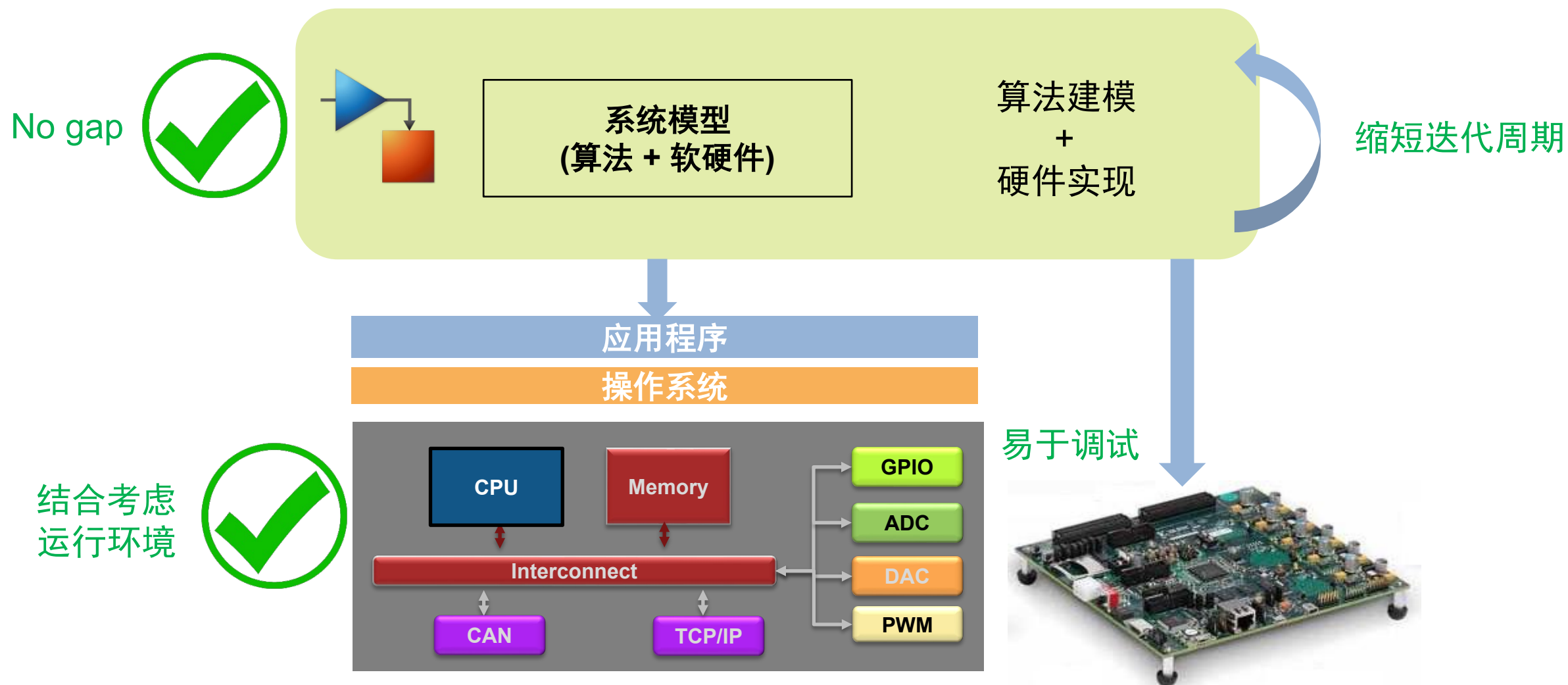
赵恒 MathWorks



算法和实现之间容易脱节



结合架构考虑算法的实现



架构仿真的核心功能

系统建模

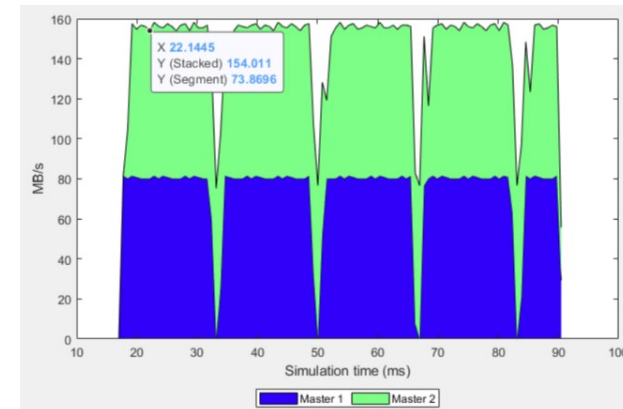
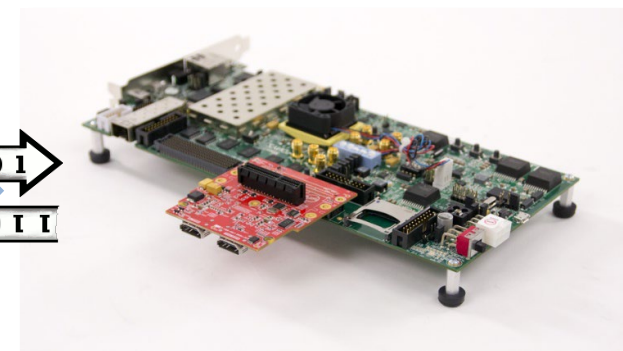
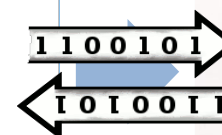
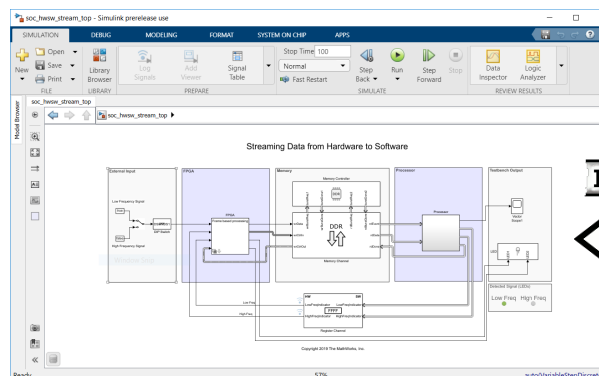
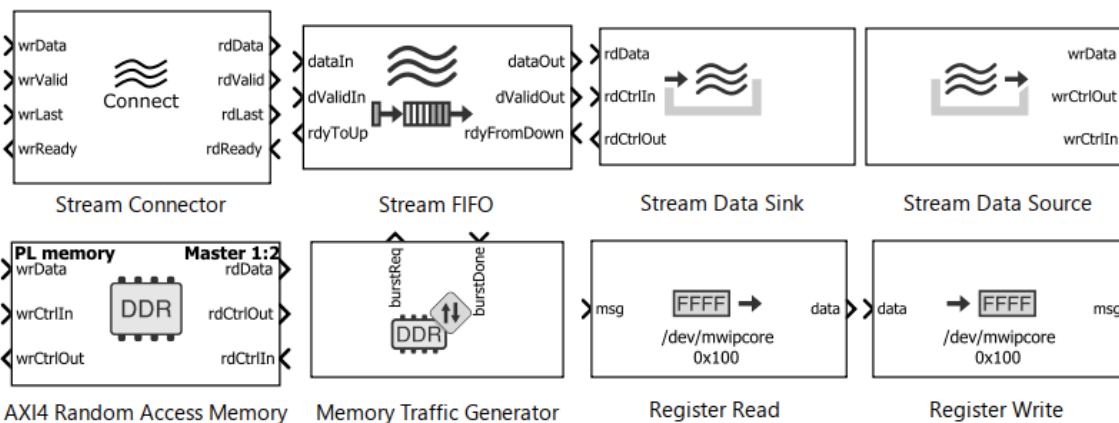
- 任务调度、中断
- 核间通信、数据缓冲
- 多核运行、可视化

快速原型

- 软硬件工程创建
- 集成设备驱动
- 适配定制开发板

实测运行

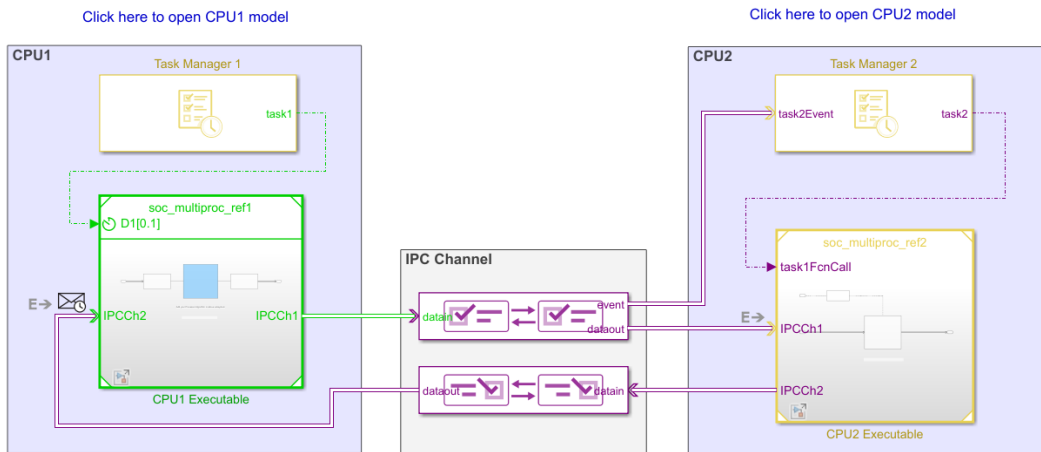
- 任务执行可视化
- CPU利用率
- 性能分析报告



从模板创建架构模型

1. Follow instructions below to add your algorithm for CPU1

2. Follow instructions below to add your algorithm for CPU2



多核架构模型

?

1. Double-click FPGA block to select Sample/Frame based simulation mode

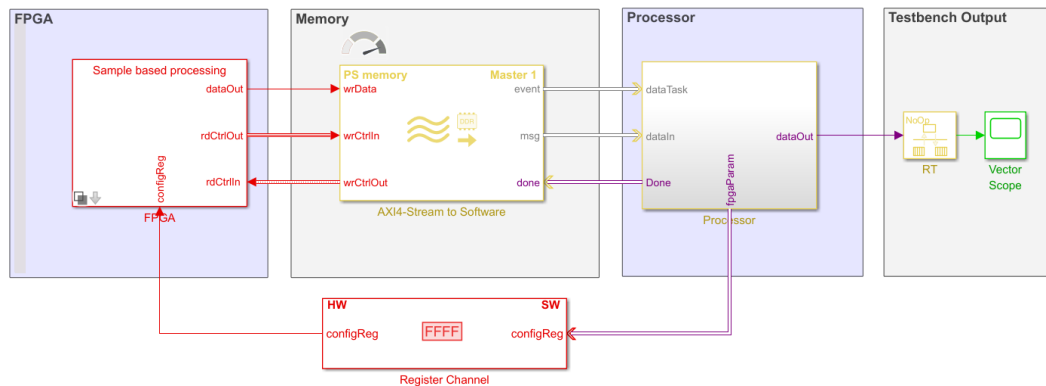
3. Follow instructions below to add your Processor algorithm

2. Follow instructions below to add your FPGA algorithm

Click here to open FPGA model - Sample based processing

Click here to open Processor model

Click here to open FPGA model - Frame based processing

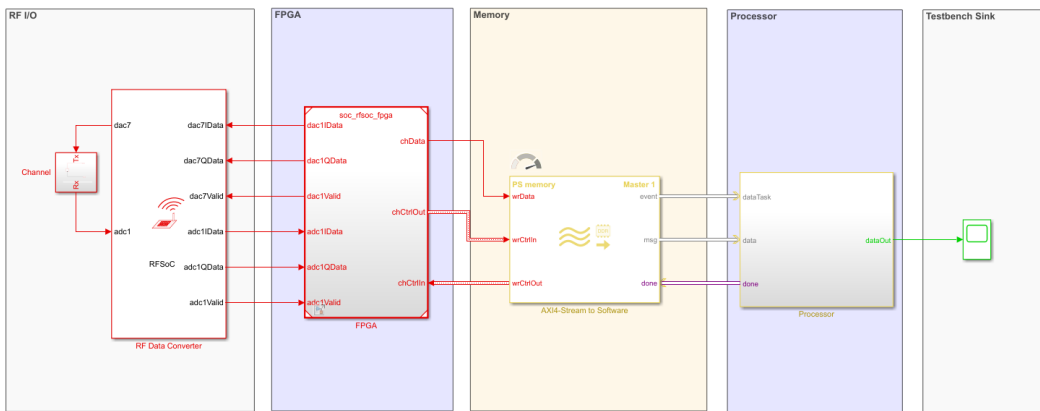


FPGA和处理器数据流模型

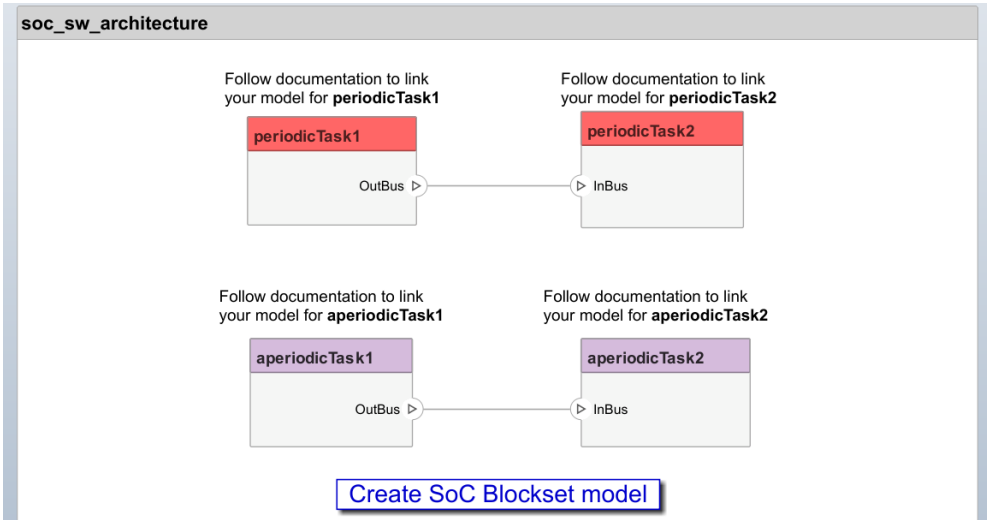
?

1. Follow instructions below to add your FPGA algorithm
Click here to open FPGA model

2. Follow instructions below to add your Processor algorithm
Click here to open Processor model

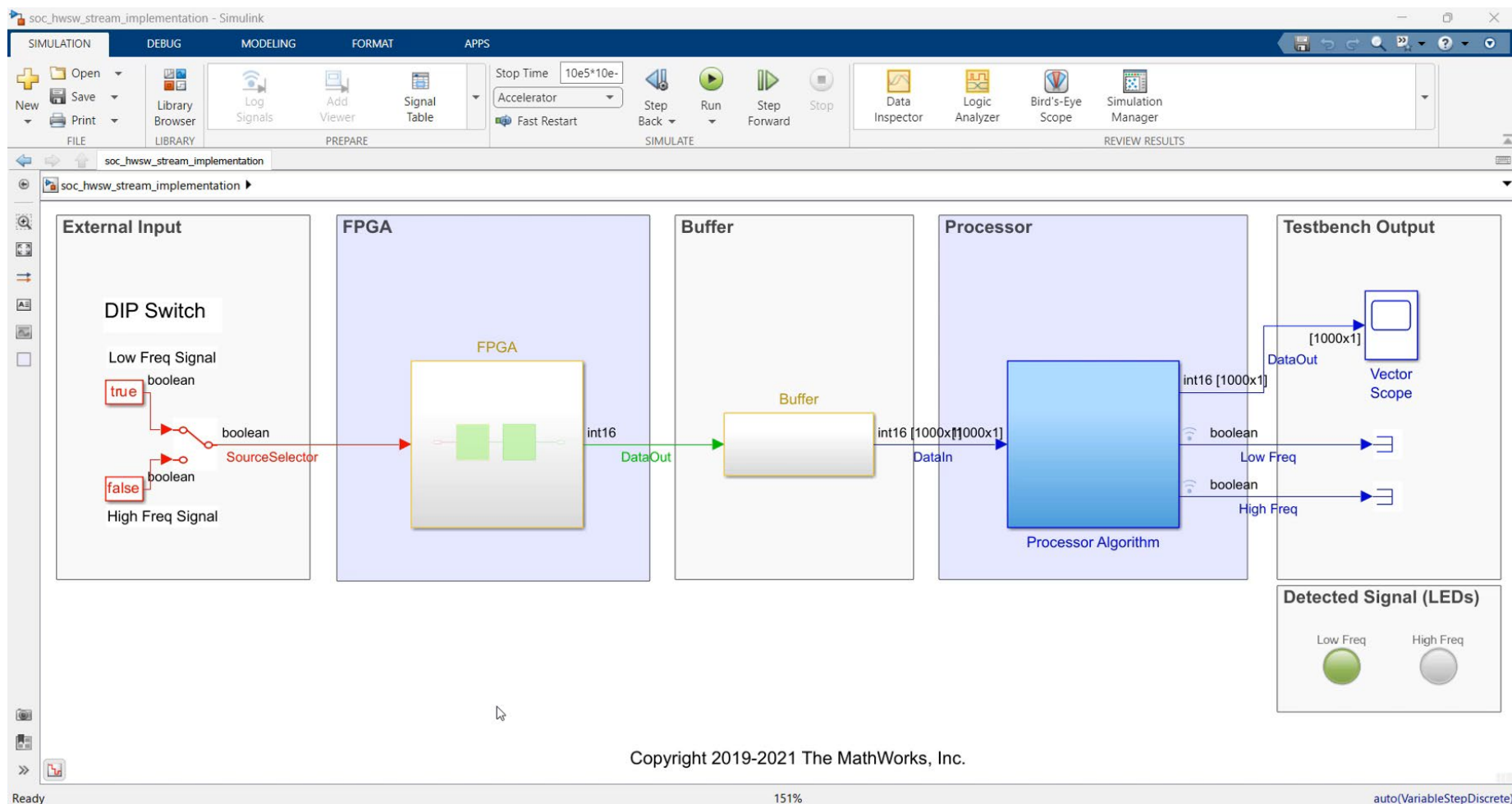


RFSoc应用模型

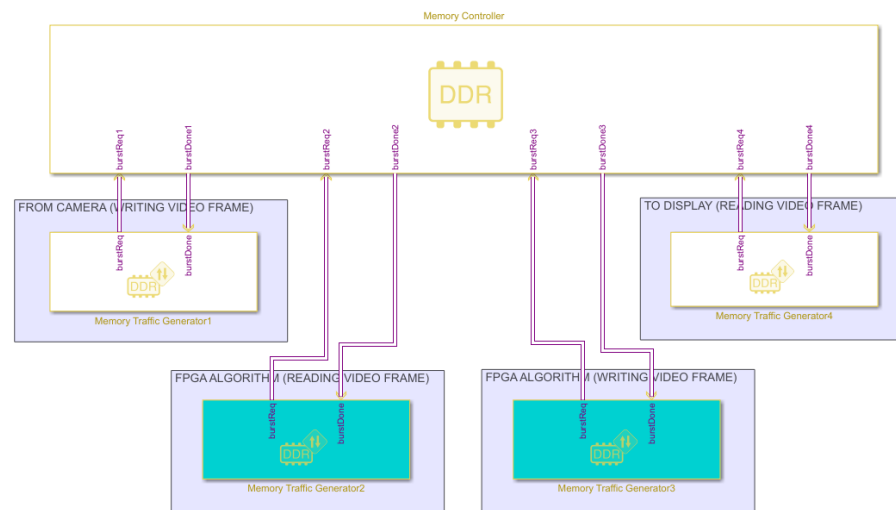
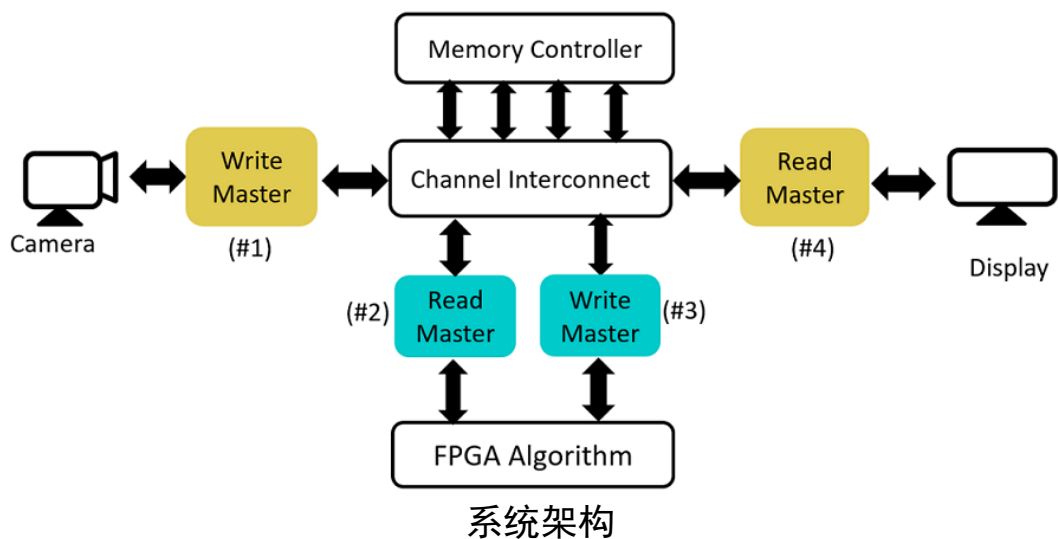


SoC软件架构模型

片内数据传输建模与性能分析



访存建模与性能分析



Show Memory Controller ports

Parameters

Request type: Reader

Total burst requests: 32400

Burst size (bytes): 512

Data width (bits): 32

Burst length (beats): 128

Allow simulation-only parameters

Simulation-only parameters

First burst time: 2.058e-7

Random time between bursts (s): [2.058e-7 2.058e-7] [2.058e-06, 2.058e-06]

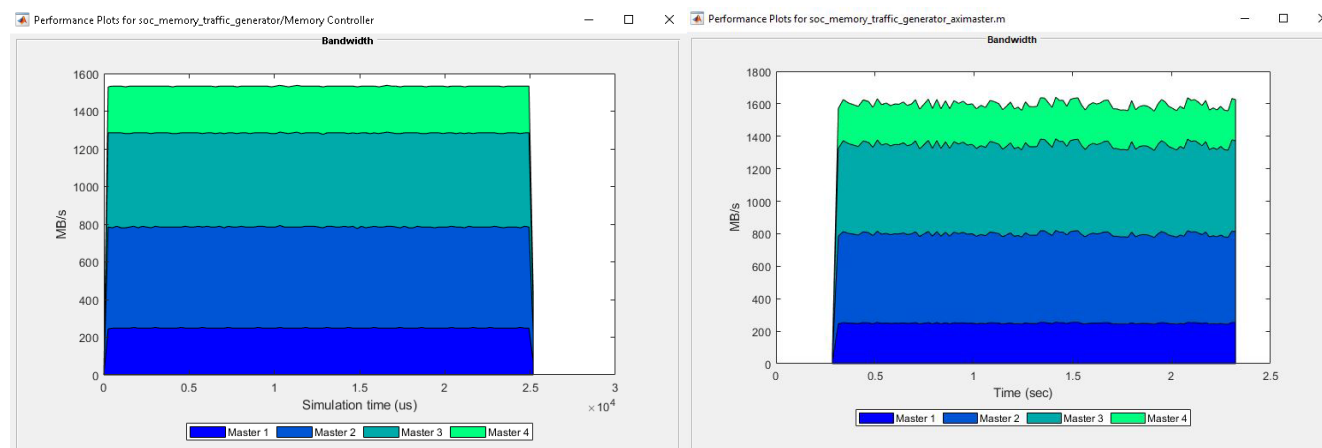
Wait for burst done

Hardware board settings

Target hardware resources

Groups	Controller clock frequency (MHz):	200
FPGA design (top-level)	Controller data width (bits):	64
FPGA design (PL mem controllers)	Bandwidth derating (%):	2.3
FPGA design (debug)	Bandwidth derating (%):	64
	First write transfer latency (clocks):	128
	Last write transfer latency (clocks):	256
	First read transfer latency (clocks):	5
	Last read transfer latency (clocks):	5

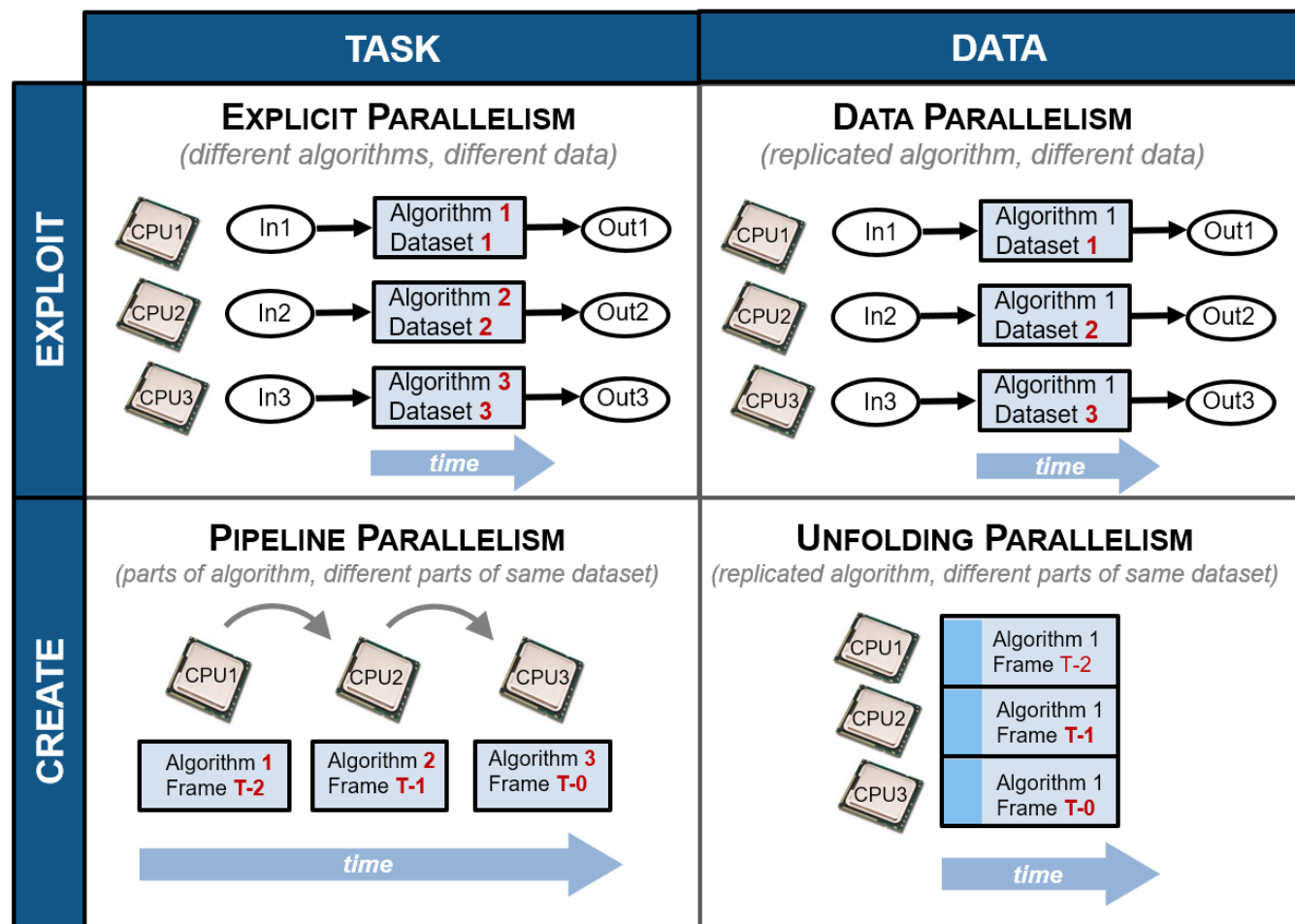
参数设置



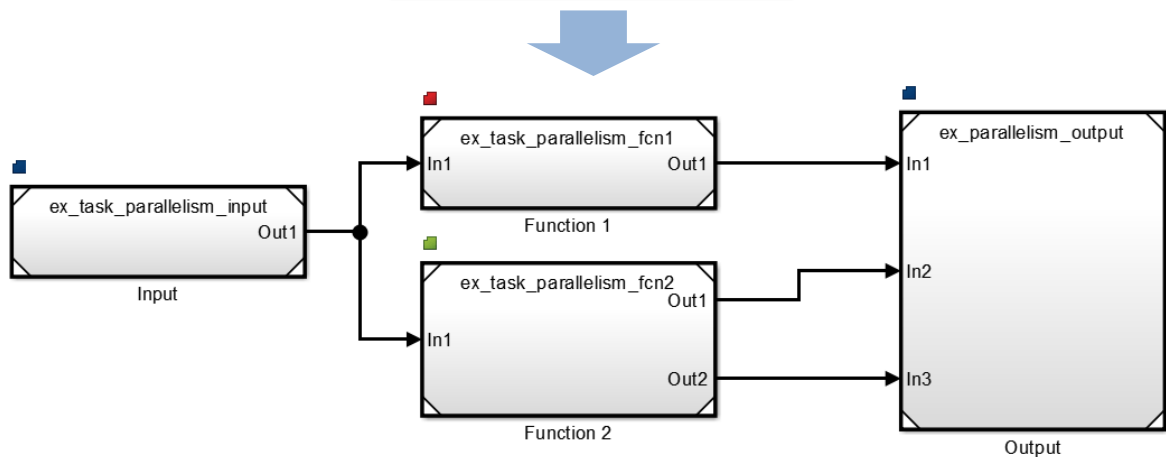
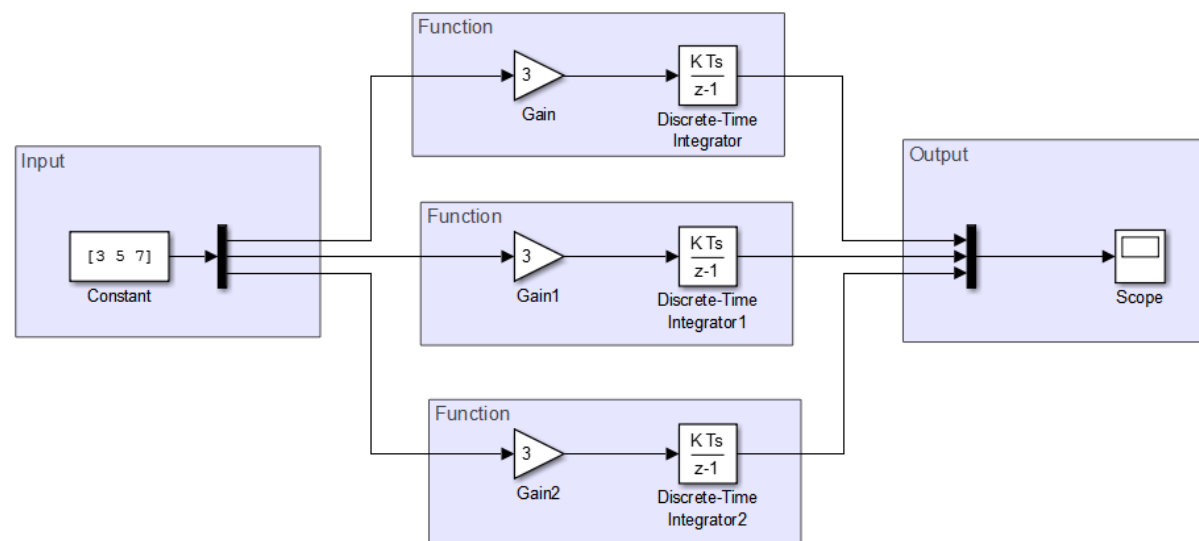
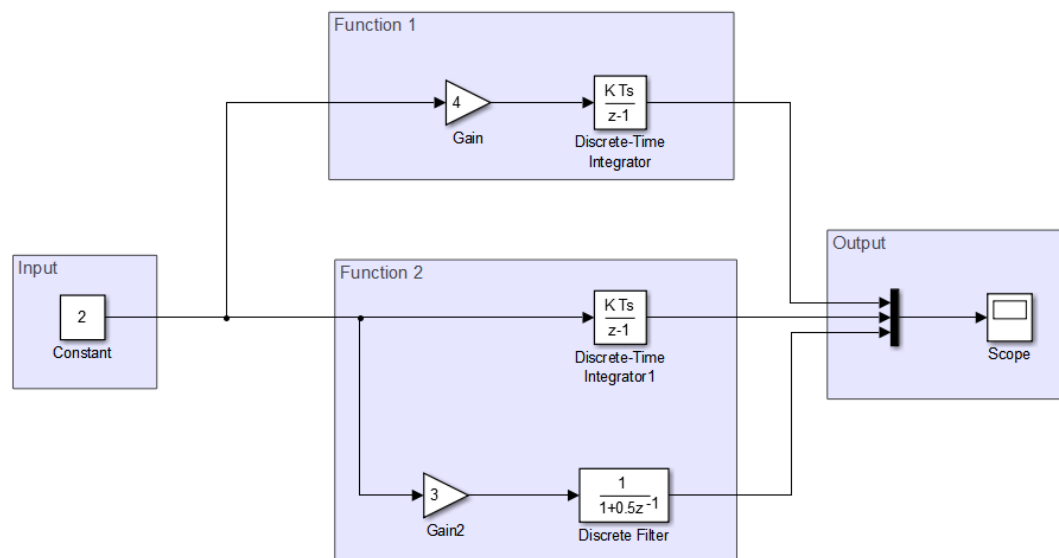
访存带宽

数据流 (Dataflow)

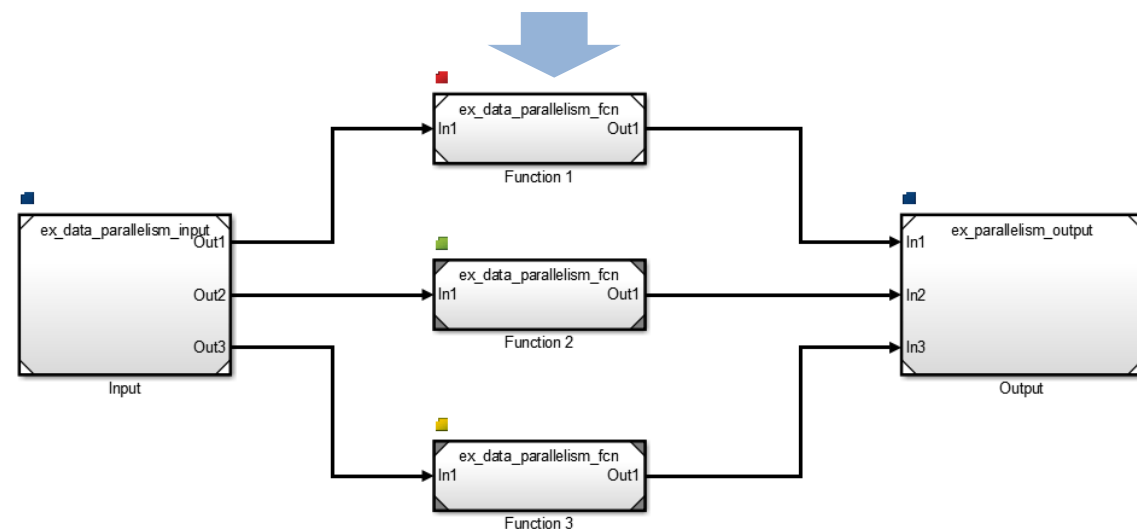
- 信号处理中的数据流
 - 模型计算密集部分
 - 数据驱动
 - 静态调度
- 使用数据流模型的收益
 - 开发算法内在并行性
 - 多线程提高仿真速度
 - 自动推断信号维度
- 数据流模型的仿真
 - 利用多核CPU
 - 多线程运行仿真
 - 代价分析



DSP算法的并行性

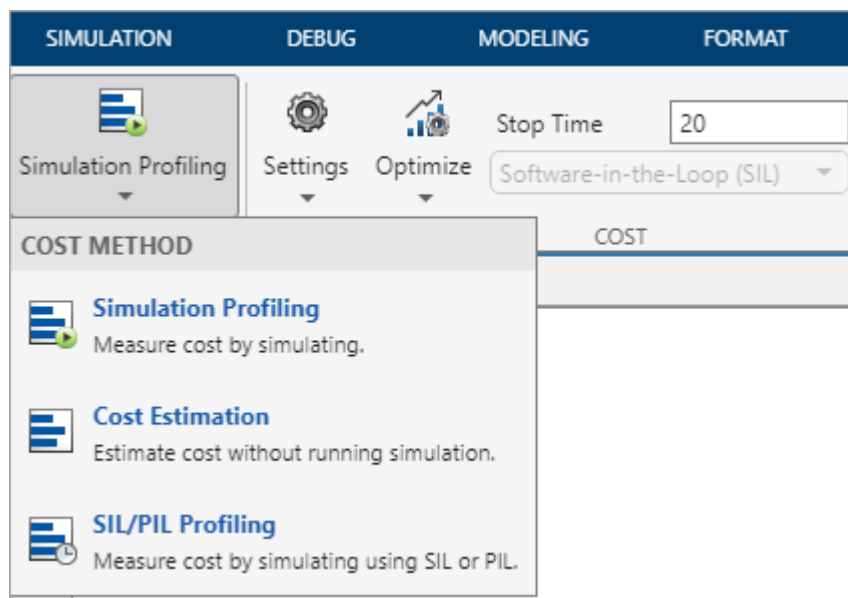
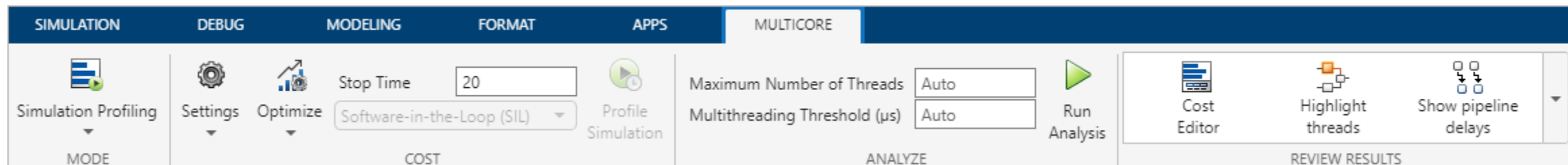


任务并行

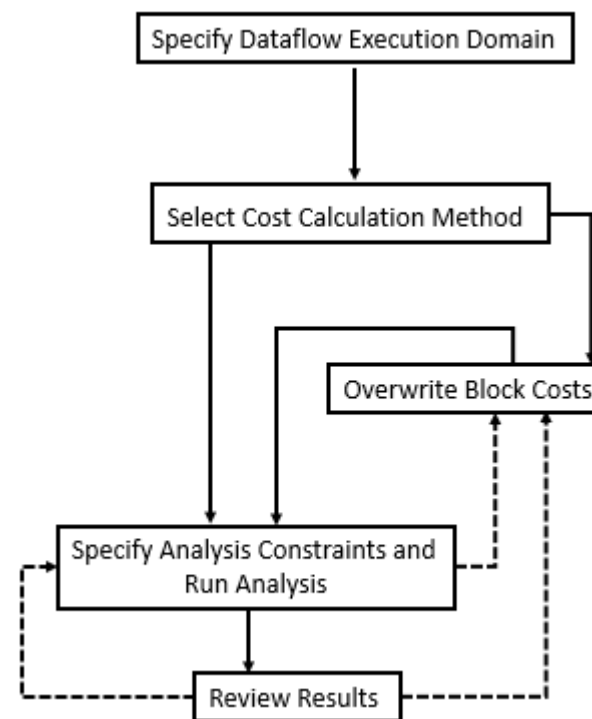


数据并行

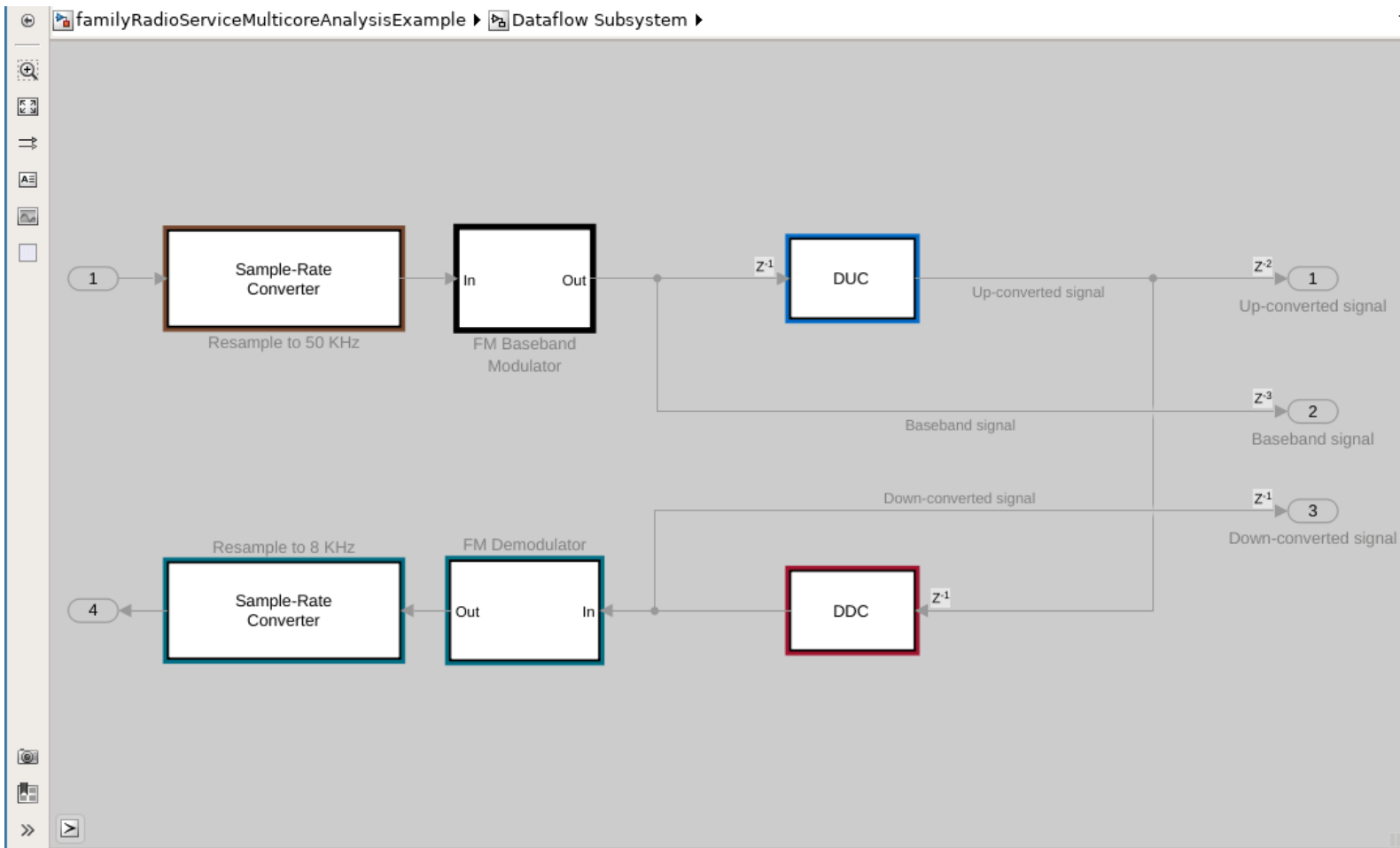
多核模型性能分析



评估方式



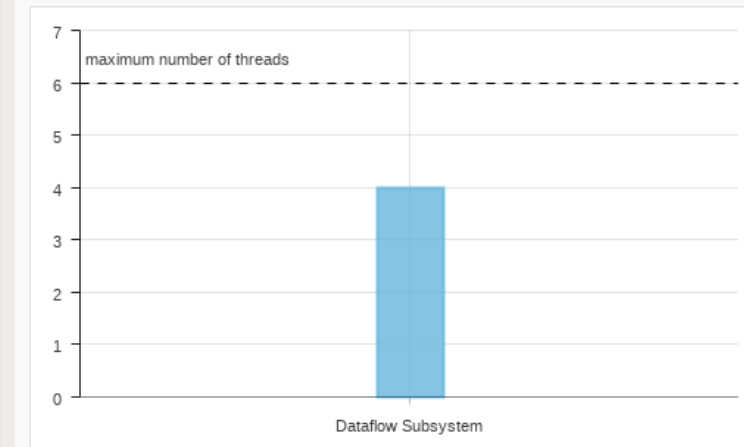
分析流程



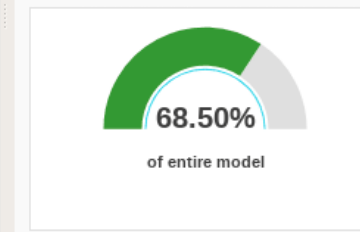
The dataflow subsystems will utilize 4 cores based on the analysis made with profiled cost values.

View: Threads

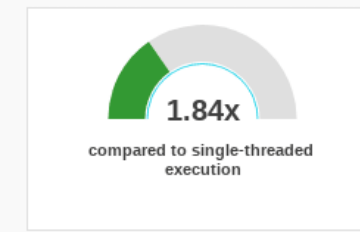
Number of threads per dataflow subsystem



Relative weight of dataflow subsystems



Maximum theoretical speedup



Property Inspector | Analysis Report and Suggestions

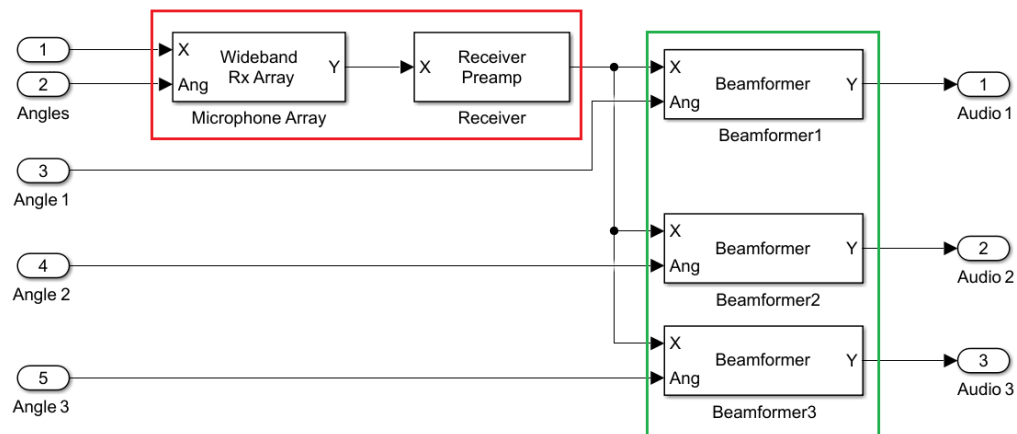
Cost Editor

Block	Auto	Cost (μs)	Relative Load
Dataflow Subsystem			
Dataflow Subsystem/Digital Down-Converter	<input type="checkbox"/>	70000	
Dataflow Subsystem/Digital Up-Converter	<input type="checkbox"/>	70000	
Dataflow Subsystem/Resample to 50 KHz	<input type="checkbox"/>	70000	
Dataflow Subsystem/FM Baseband Modulator/Magnitude-Angle to Complex	<input checked="" type="checkbox"/>	1921	
Dataflow Subsystem/FM Demodulator/Complex to Magnitude-Angle	<input checked="" type="checkbox"/>	1504	

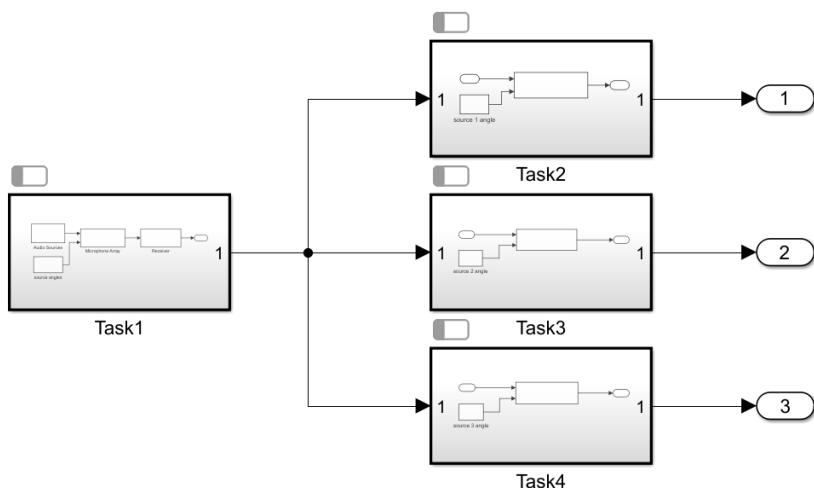
Thread Highlighting Legend

Enabled	Thread	Color
<input checked="" type="checkbox"/>	Thread 1	
<input checked="" type="checkbox"/>	Thread 2	
<input checked="" type="checkbox"/>	Thread 3	
<input checked="" type="checkbox"/>	Thread 4	
<input type="checkbox"/>	Multiple	

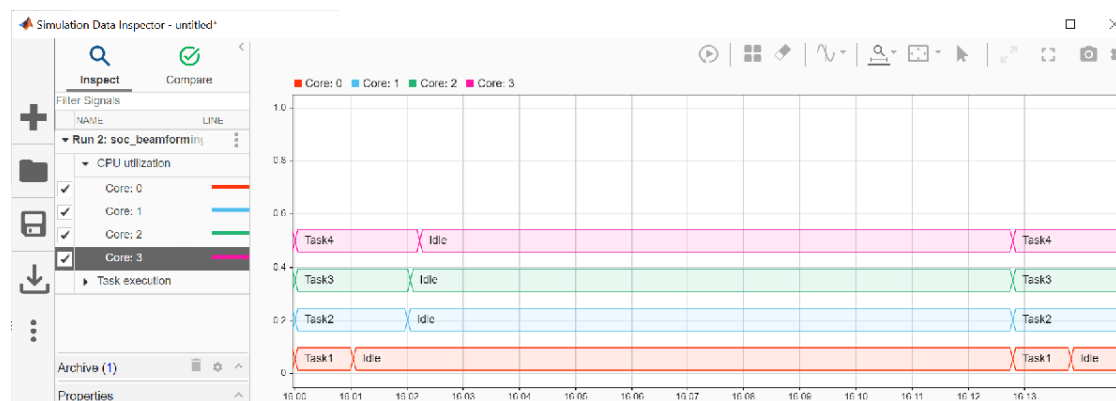
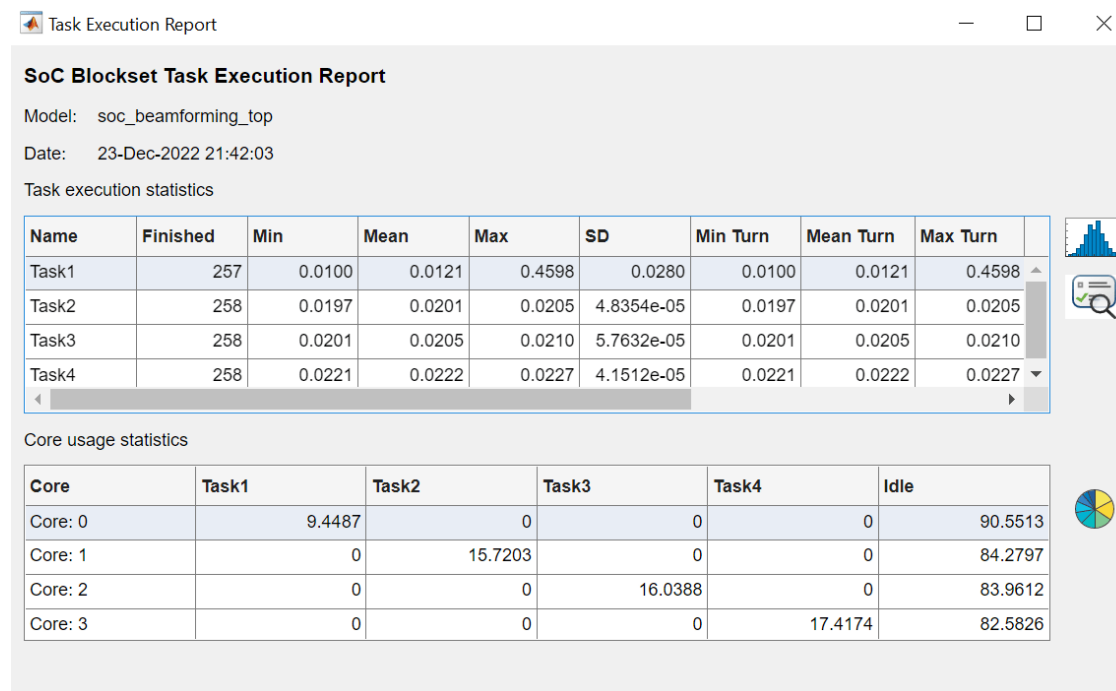
算法多核映射与仿真



算法架构

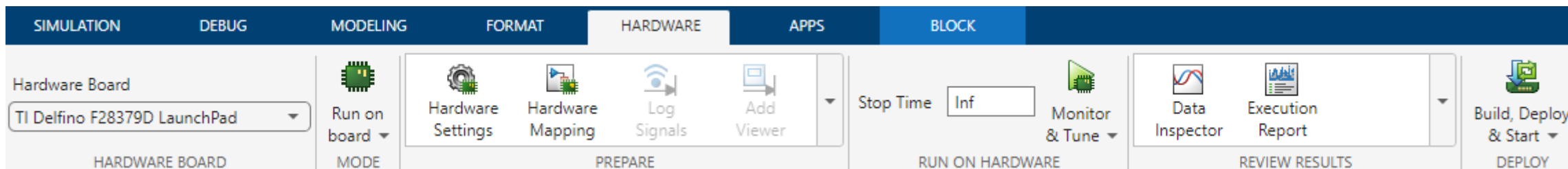


任务划分



仿真分析

多核DSP部署 – C2000

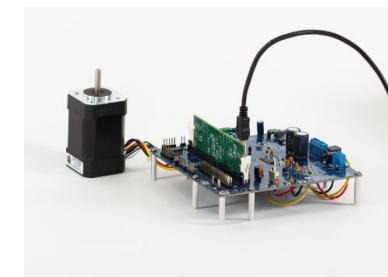
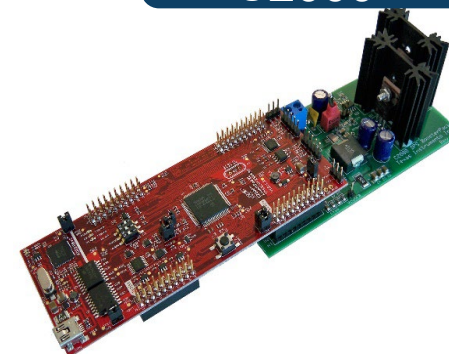
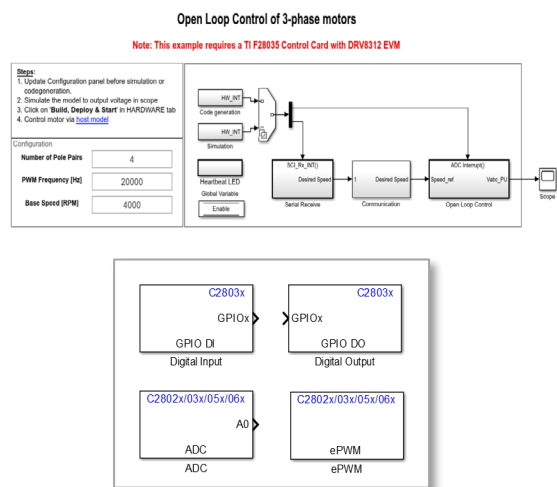


Simulink Model

Embedded Coder

CCS Project

Run on C2000



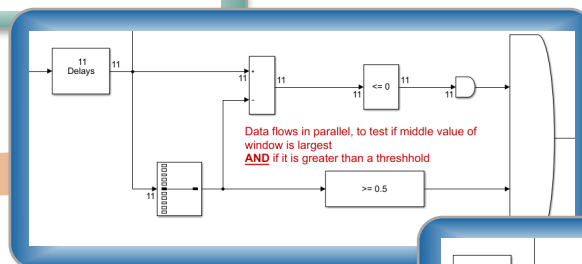
FPGA设计流程

MATLAB
参考算法

```
%% MATLAB reference detector
% this uses high level MATLAB functions
% computing a global maximum requires holding the entire signal at once
% this is impractical in a hardware implementation but serves as a golden
% reference

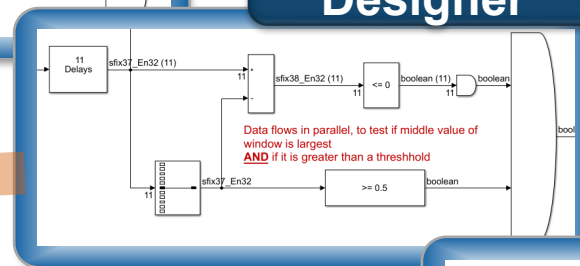
y=filter(CorrelationFilter,1,RxSignal); % correlate against the pulse
[peak, location]=max(abs(y).^2);
fprintf('Found Global Maximum at location %d Value %3.3f \n',location, peak)
```

硬件架构



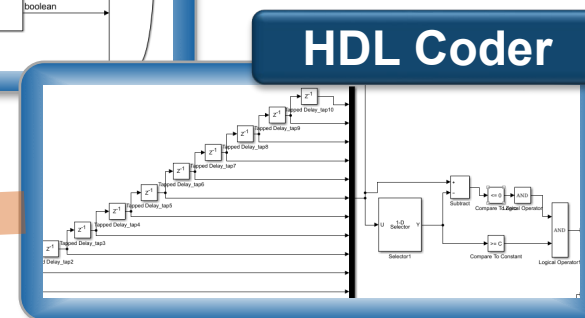
Fixed Point
Designer

定点实现



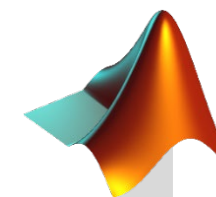
HDL代码
生成和优化

逐级验证



HDL Coder

HDL验证和部署



MATLAB



Simulink

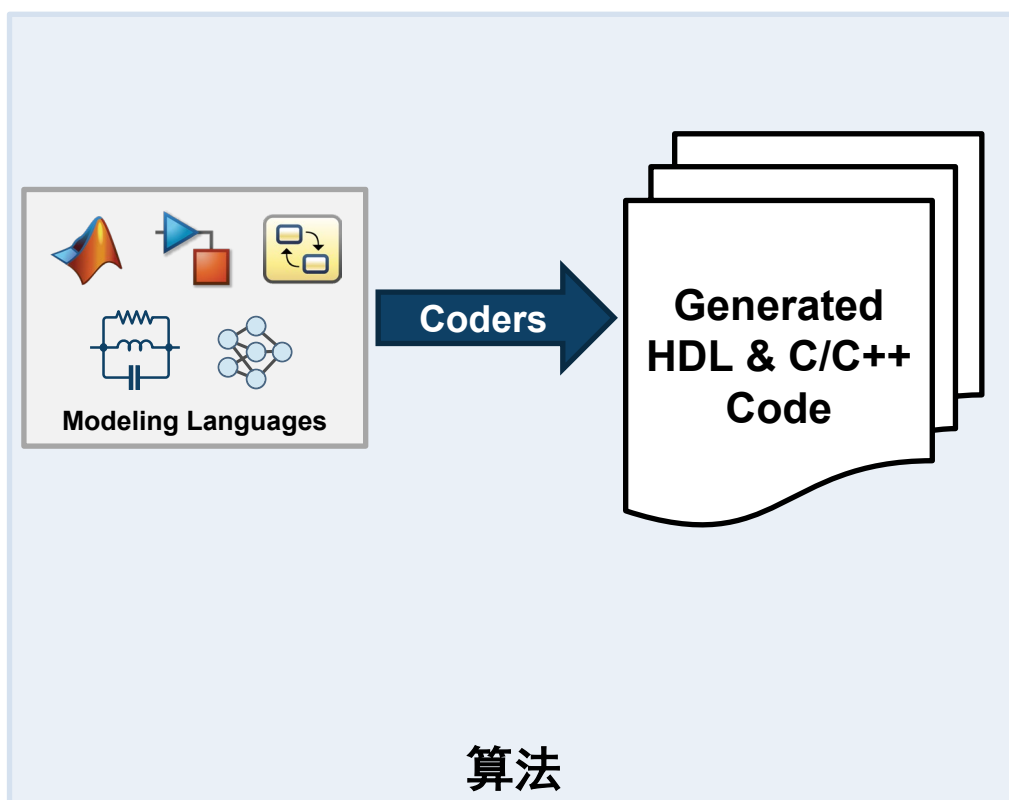


软硬件联合设计

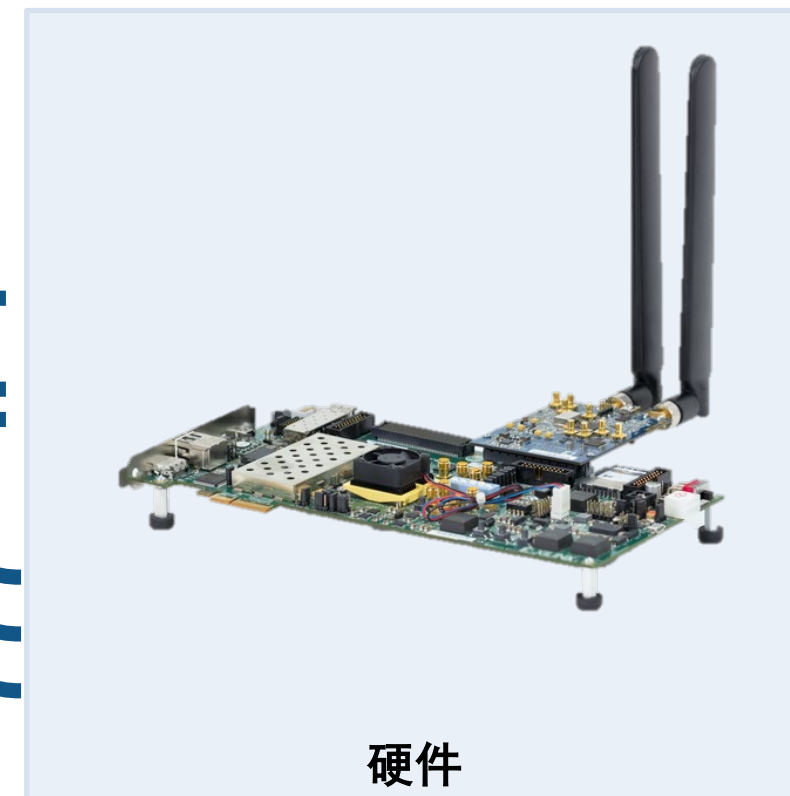
将生成的HDL代码集成到FPGA工程

将生成的C代码集成到嵌入式工程

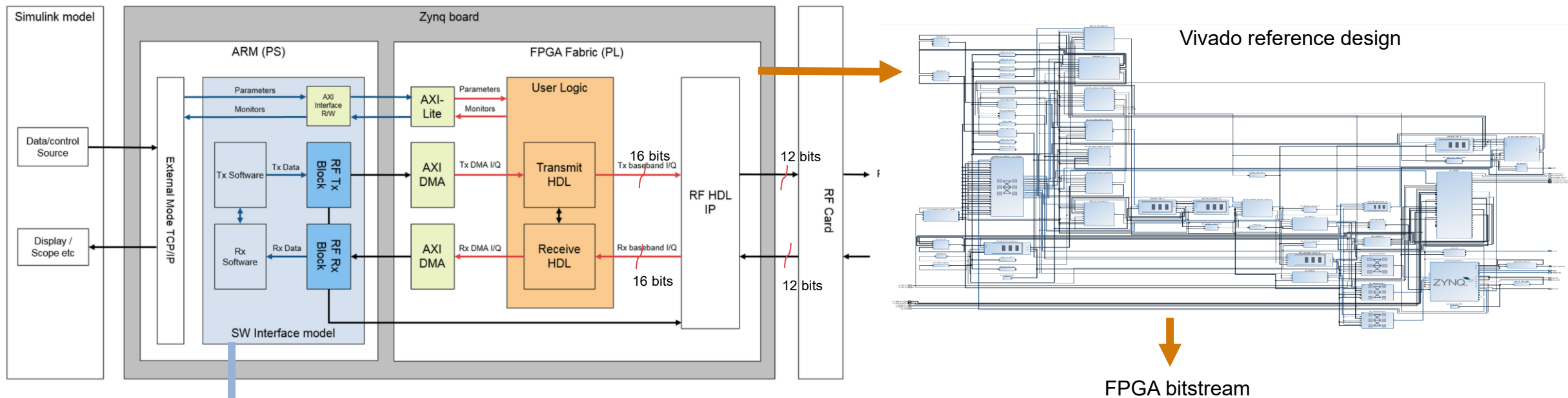
让算法在实际硬件板卡上运行起来



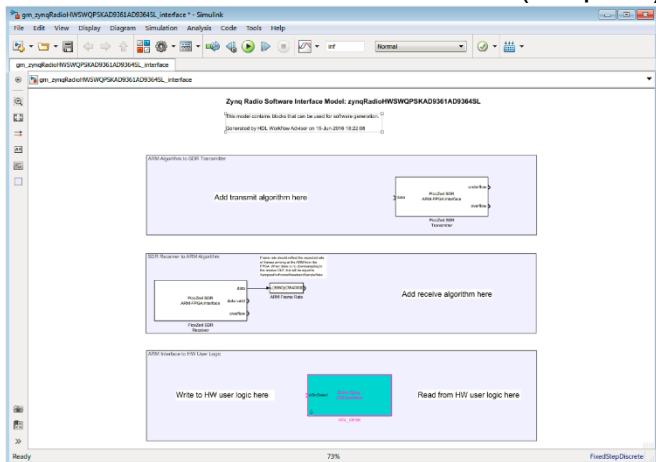
软硬件联合设计流程



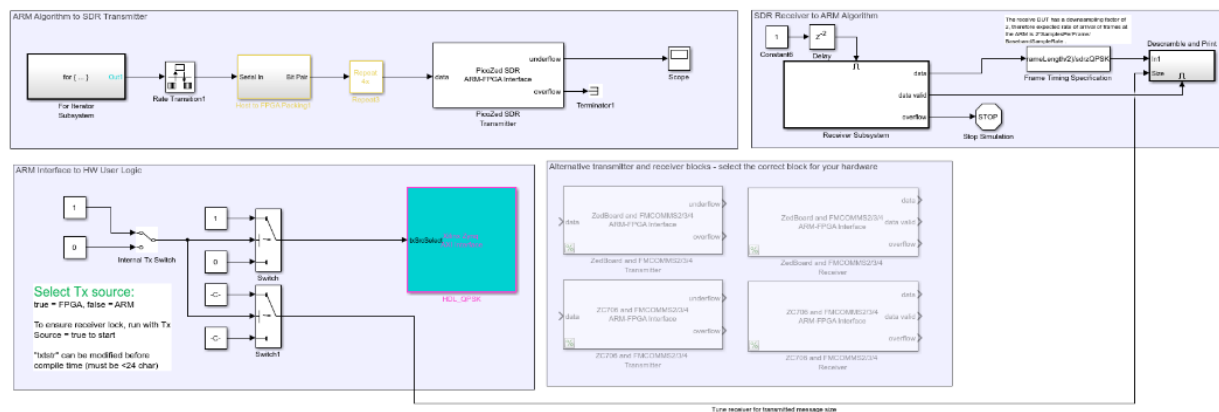
软硬件联合设计



SW interface model (template)

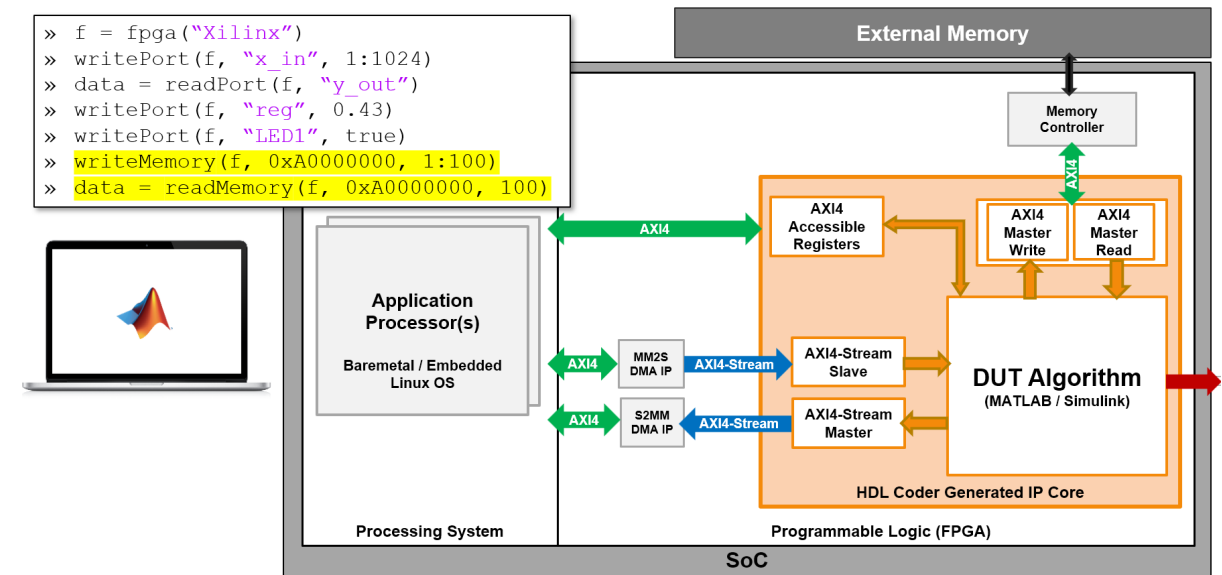


SW interface model (+user logic)



软硬件联合设计功能更新

- IP核流程
 - 新增支持生成通用IP核
- 帧模型及部署
 - 将帧信号映射到AXI4-Stream视频接口
 - 将帧延迟映射到外部存储器
- IP核接口
 - AXI4从接口寄存器支持>32位数据类型
 - AXI4主接口支持访问>32位地址位宽
- 硬件平台
 - Xilinx Versal IP核流程参考设计
 - Microchip硬件支持
 - NI硬件支持



IP 集成

■ IP 集成

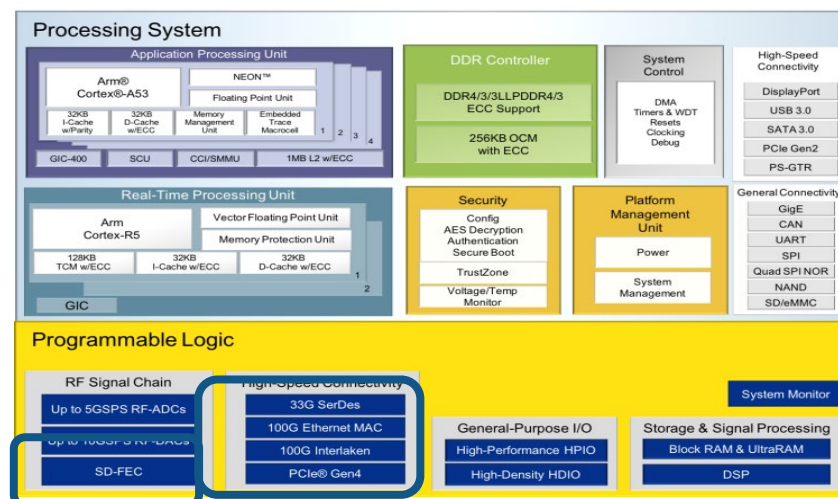
- 在 SoC 模型中集成第三方 IP 核
- 从用户输入生成 SoC 模块
- 向导式 IP 集成界面

■ 使用场景

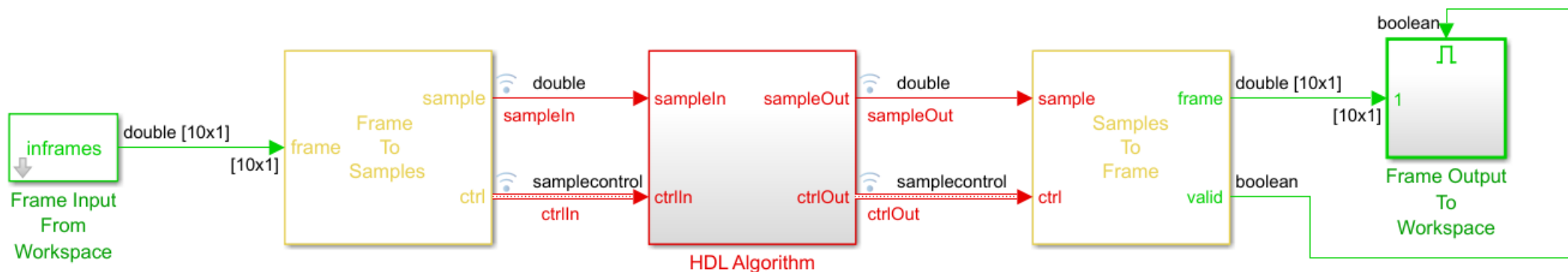
- 集成高速外设 – PCIe, 100G Ethernet, 150G Interlaken and GTH
- 集成第三方 IP – DL IP, RF 前端控制 IP
- 集成 Xilinx IP – SPI, SDFEC

■ 支持定制

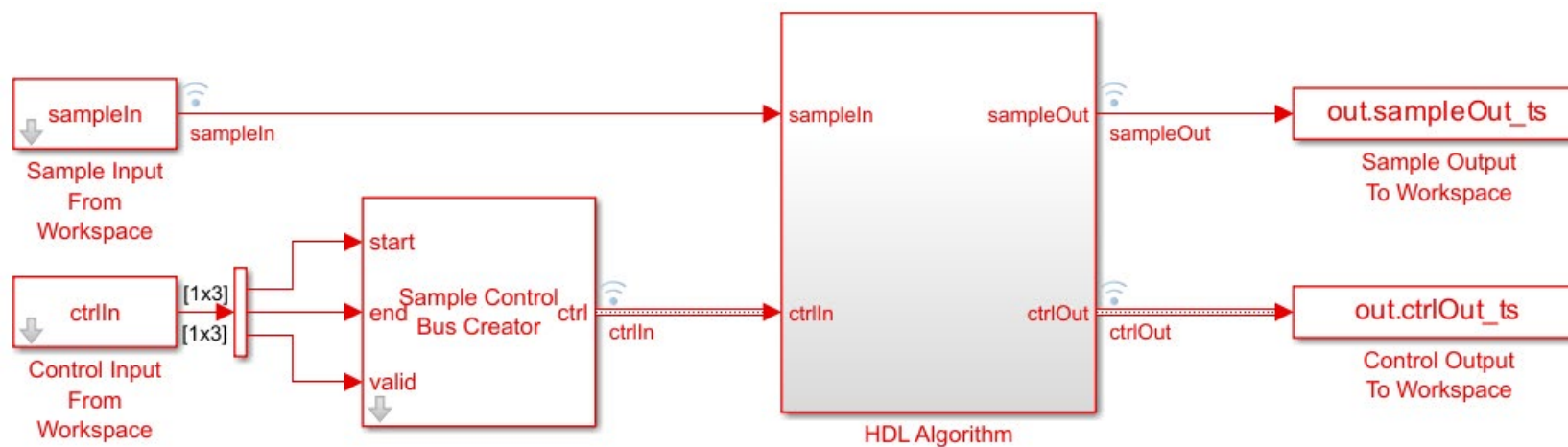
- 支持 Crebrus RFSoc
- 支持 Aurora IP 集成



从模板创建无线通信HDL模型

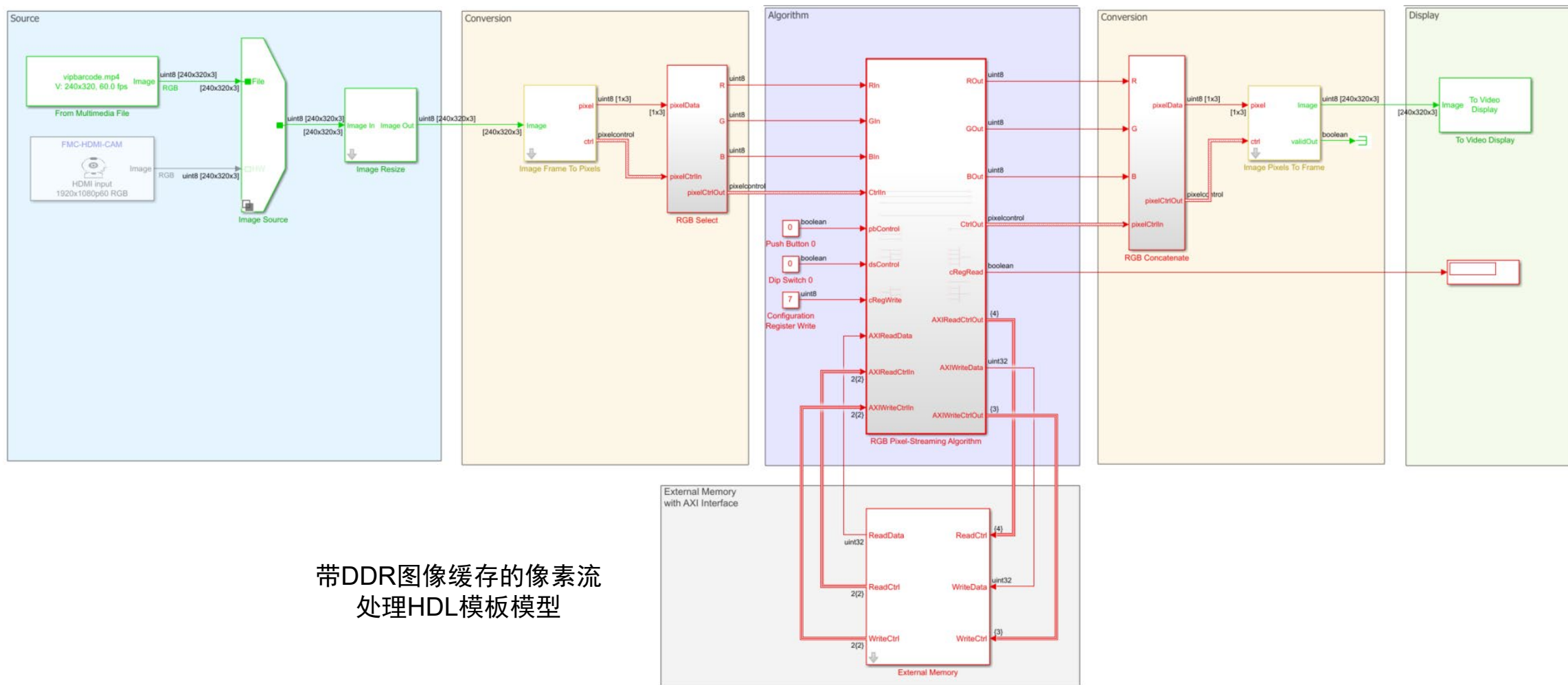


信号帧输入/输出模板模型



信号流输入/输出模板模型

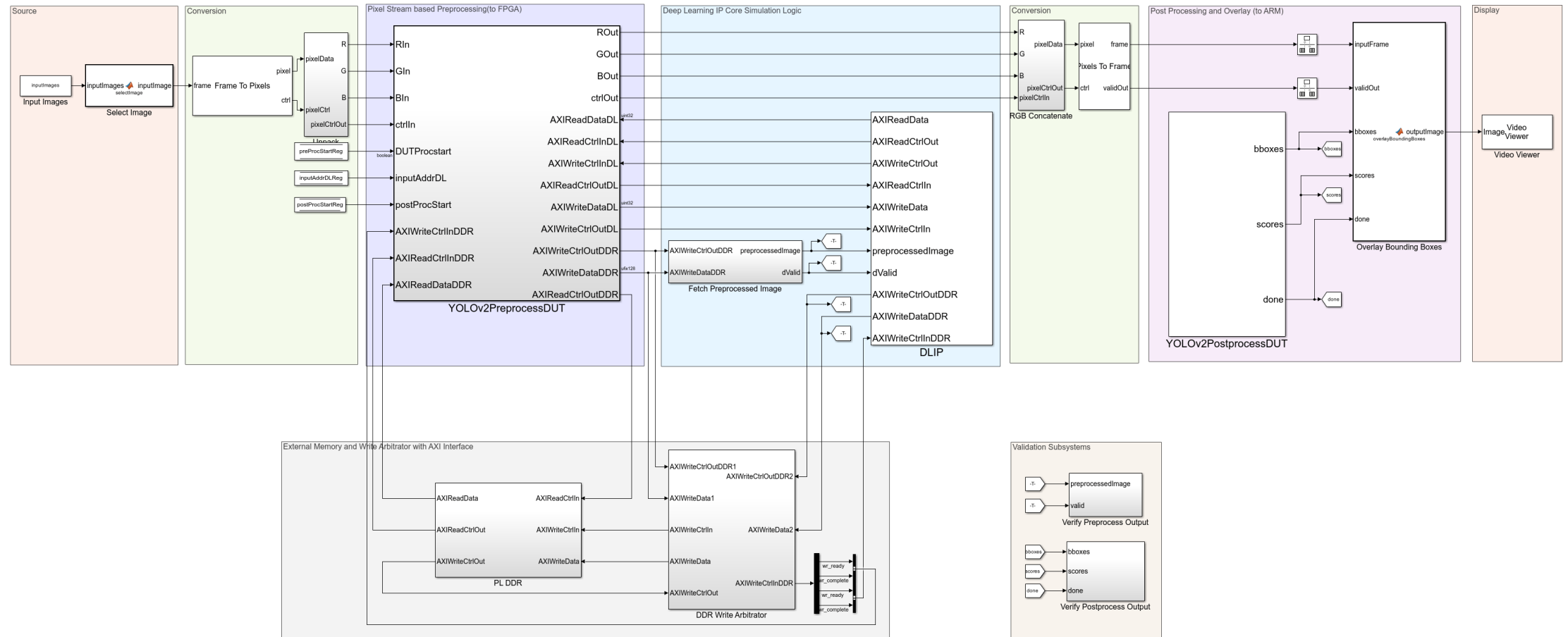
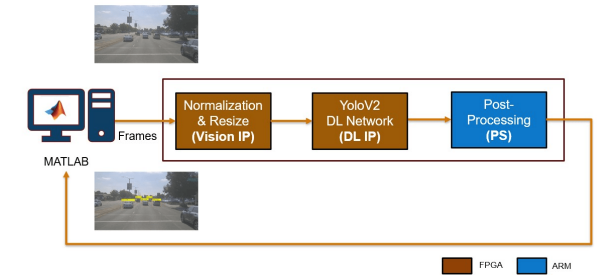
从模板创建图像处理HDL模型



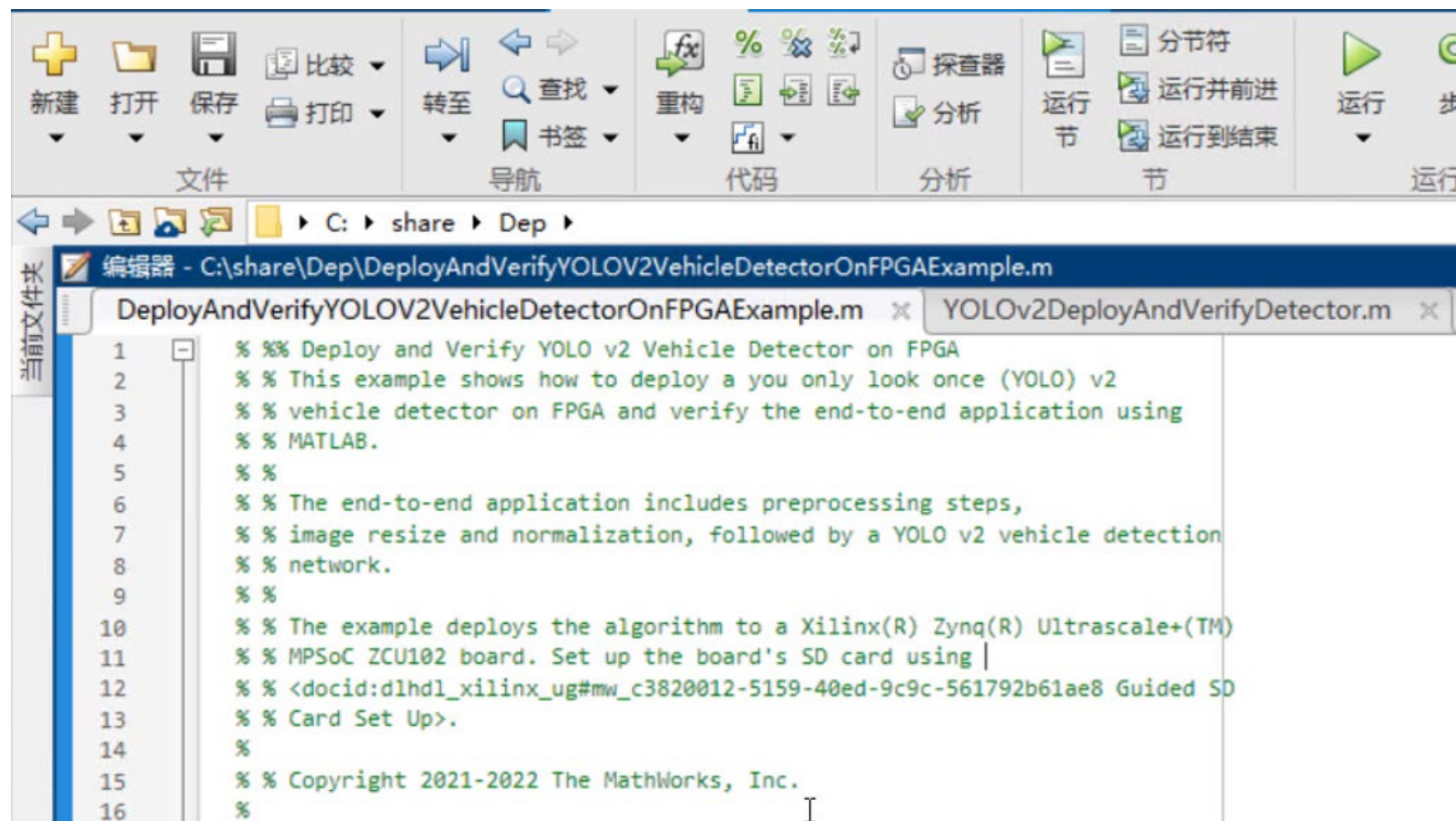
带DDR图像缓存的像素流
处理HDL模板模型

深度学习FPGA部署

YOLO v2 vehicle detector on SoC - Preprocess and Postprocess with deep learning handshake logic



深度学习FPGA部署示例



The screenshot displays the MATLAB IDE interface. The top toolbar includes icons for file operations (New, Open, Save, Compare, Print), navigation (Go to, Find, Bookmarks), code management (Rebuild, Refresh, Run), and analysis (Debugger, Analyze). The current directory is C:\share\Dep. The active script is 'DeployAndVerifyYOLOV2VehicleDetectorOnFPGAExample.m'. The script content is as follows:

```
1 %% Deploy and Verify YOLO v2 Vehicle Detector on FPGA
2 %% This example shows how to deploy a you only look once (YOLO) v2
3 %% vehicle detector on FPGA and verify the end-to-end application using
4 %% MATLAB.
5 %%
6 %% The end-to-end application includes preprocessing steps,
7 %% image resize and normalization, followed by a YOLO v2 vehicle detection
8 %% network.
9 %%
10 %% The example deploys the algorithm to a Xilinx(R) Zynq(R) Ultrascale+(TM)
11 %% MPSoC ZCU102 board. Set up the board's SD card using |
12 %% <docid:d1hd1_xilinx_ug#mw_c3820012-5159-40ed-9c9c-561792b61ae8 Guided SD
13 %% Card Set Up>.
14 %%
15 %% Copyright 2021-2022 The MathWorks, Inc.
16 %%
```


MATLAB EXPO

谢谢



© 2023 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [mathworks.com/trademarks](https://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.