

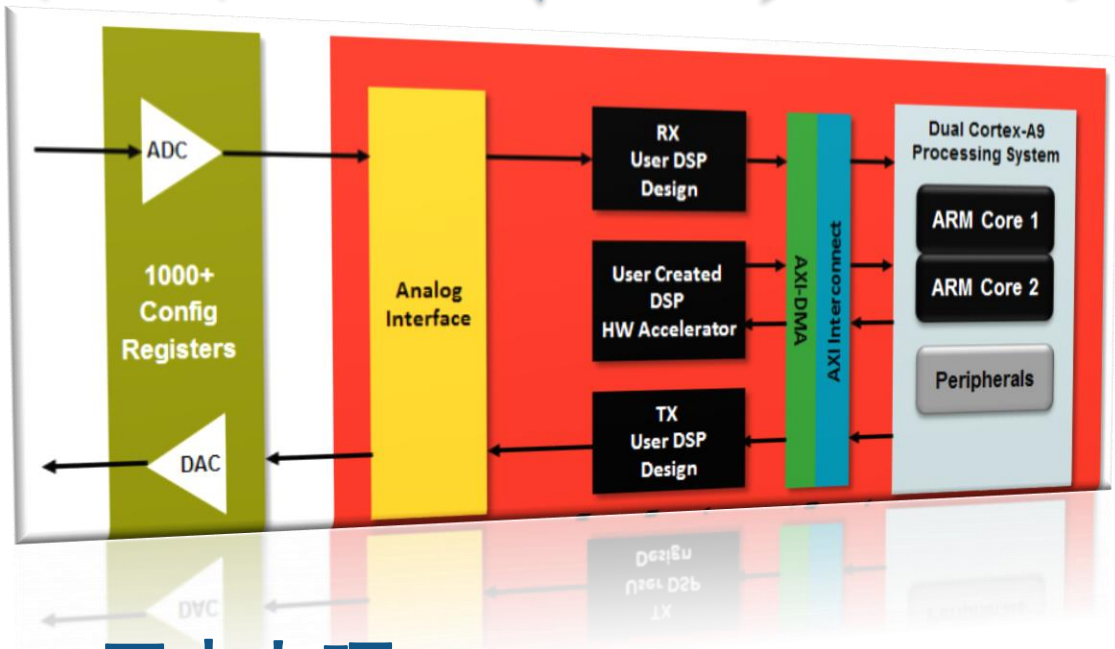


# MATLAB EXPO 2017

化繁为简：软件定义无线电设计  
的原型实现

阮卡佳 MathWorks 应用工程师

# 复杂的下一代无线系统



至少七项  
需要设计技能才能设计成功的产品!

# 系统设计常见挑战

- 数字信号处理复杂度不断增加
  - 无线，宽带，半导体
- 产品上市时间压力
  - 设计验证太晚
  - 设计错误的风险，影响上市时间
- 设计团队的合作
  - 模拟/混合信号，数字硬件，DSP，嵌入式系统开发，控制部分设计

# 工程师关心什么？

- 系统工程师和射频工程师
  - 模拟系统性能，然后才确认硬件
  - 在原型和生产阶段可以灵活的测试和验证系统性能
- 算法工程师
  - 设计并仿真新的无线电算法
  - 在 FPGA 和 SoC 硬件实现设计原型

# 今天的 MATLAB 和 Simulink 可以帮您做什么？

仿真



**端到端链路建模与仿真**  
设计并验证无线系统



测试



**Over-the-air 测试**  
利用 SDR 设备和射频设备验证模型



实现

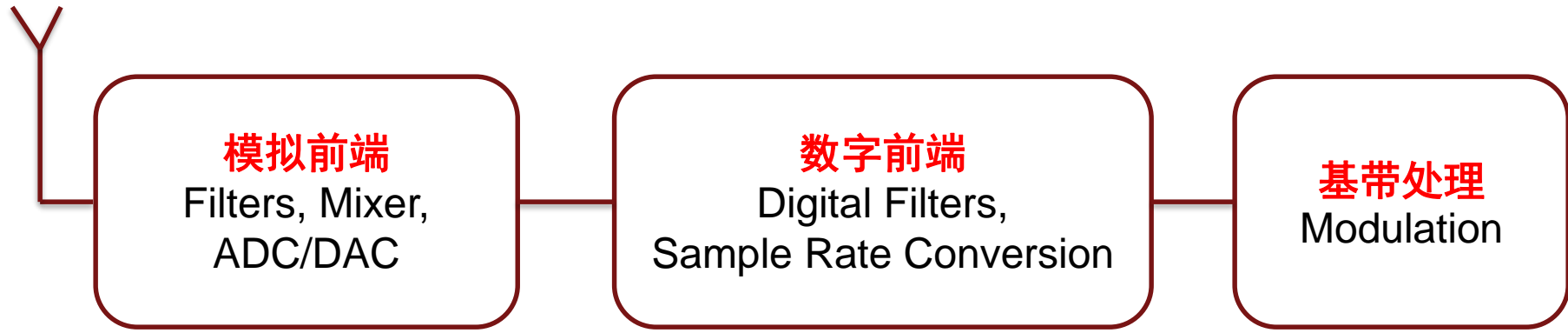


**原型与实现**  
将算法部署到 SDR 平台

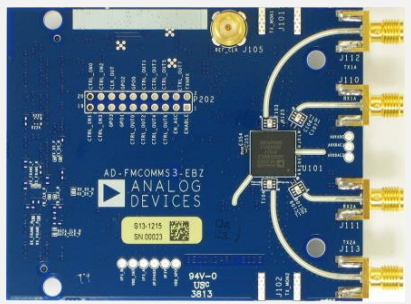


**设计一致性：从仿真、原型到实现**

# 典型的 SDR 系统



## 商用 SDR 平台

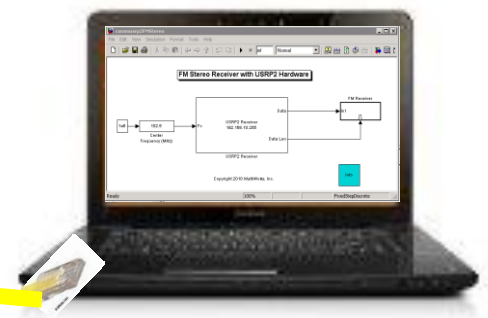


FMC

可调 RF 卡  
(比如: 70 MHz to 6 GHz)



FPGA 或 SoC  
开发板



GigE

主机

# SDR 硬件支持

## Xilinx Zynq-Based Radio

*ZC706, ZedBoard, PicoZed*

*ADI FMCOMMS1/2/3/4*



## Xilinx FPGA-Based Radio

*Virtex-6 ML605, Spartan-6 SP605*

*ADI FMCOMMS1, Epiq FMC-1Rx*



## USRP Radio

*USRP2, N200/210*

*B200/B210, X300/310*

*E310*



## Analog Devices ADALM Pluto Radio

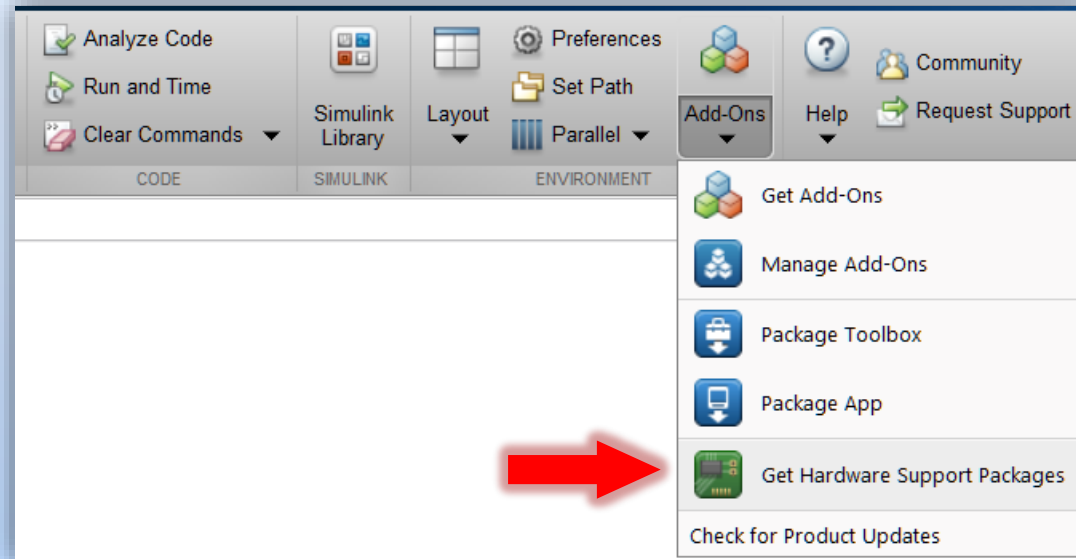
## RTL-SDR Radio





# 下载 SDR 支持包

从 MATLAB 菜单栏:  
**Add-Ons → Get Hardware Support Packages**



从 MATLAB 命令行窗口  
**>> supportPackageInstaller**



# 自动 SDR 硬件设置

**Support Package Installer**

**Configure network connection on the host**

The host computer communicates with the radio hardware automatically, select it from the list, provide the IP address.

If your network connection is not listed or the steps in the "Configure Network Card on Host" section are not followed, click the "Skip this step..." checkbox and click "Next" to continue.

Select a NIC to connect with the radio hardware.

Intel(R) Ethernet Connection I219-LM (In u)	
Name	: Ethernet
Description	: Intel(R) Ethernet
MAC Address	: 50-7B-9D-F1
IP Address	: 172.18.43.1

IP Address : 172.18.43.1

Subnet Mask : 255.255.255.0

Skip this step if your network card is already configured.

---

**Support Package Installer**

**Select a drive**

Insert a 4 GB or larger SD memory card into a memory card reader.

Select the drive letter that corresponds to the memory card.

Drive:  Refresh

**No memory card found**

If you do not find the memory card reader in the list of drives, click the "Refresh" button.

If the automatic SD Card configuration is not successful, follow the steps listed in the "Select a drive" section. Once configured, click the "Skip this step..." checkbox and click "Next" to continue.

Skip this step if your SD card is already configured.

---

**Support Package Installer**

**Connect the hardware**

---

**Support Package Installer**

**Verify setup**

Test results for the radio hardware are as follows:

- ✓ Ping network card on the host computer - SUCCESS
- ✓ Ping Zynq radio - SUCCESS
- ✓ Read hardware information from the Zynq radio - SUCCESS
- ✓ Test data path between host and RF card - SUCCESS

Setup of radio hardware is complete.

**Additional setup steps for targeting**

If you plan to target SDR algorithms to the Zynq FPGA fabric, you must install the HDL Coder Support Package for Xilinx Zynq-7000 Platform.

If you plan to generate code for targeting the ARM portion of the Zynq device, you must install the Embedded Coder Support Package for Xilinx Zynq-7000 Platform. You must follow additional steps for setup of ARM targeting.

Click the "Setup for ARM Targeting" button.

**Get started with an example**

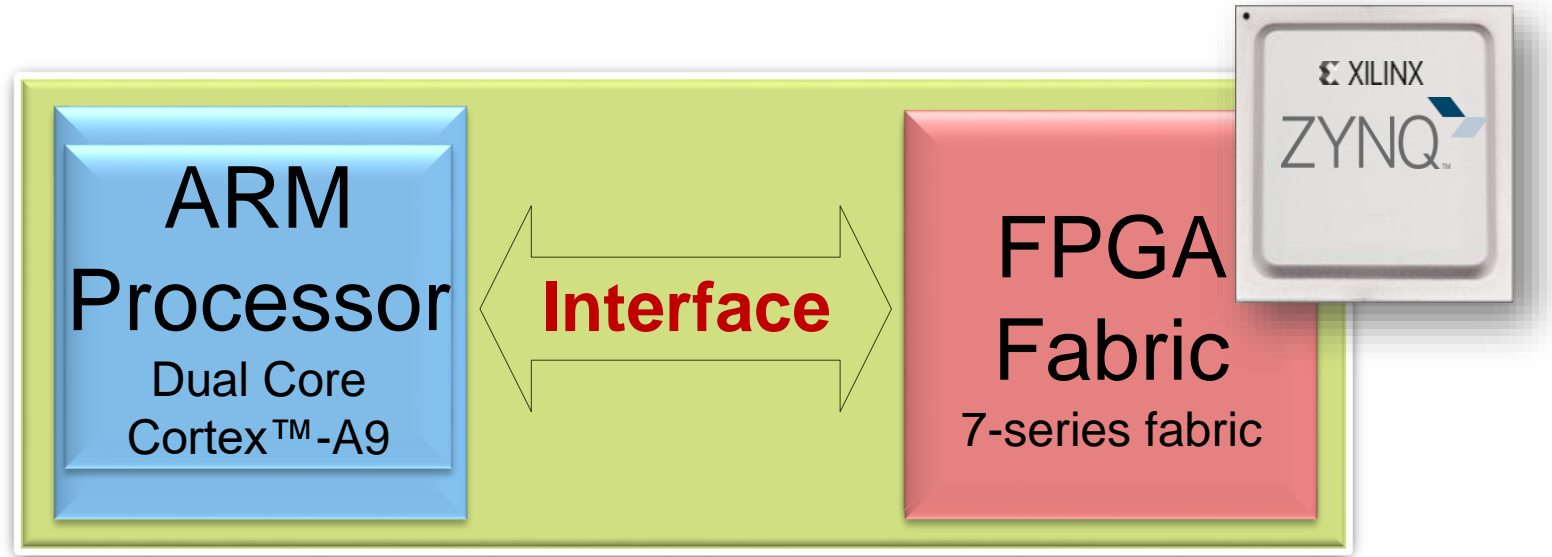
The receive tone signal example shows how to perform a simple loopback of a complex sinusoid signal at an RF carrier frequency.

Connect a loopback cable with attenuation or attach appropriate antenna between the 'RX' and 'TX' SMA connectors before you run the example.

View the documentation for this example to learn more.

< Back   Next >   Cancel   Help

# 什么是 Zynq?



- Zynq: 赛灵思公司推出的可扩展处理平台
  - 全可编程片上系统 (SoC)
  - ARM 处理器 + 可编程 FPGA
- 基于 Zynq 的 SDR 硬件平台
  - ZC706 / ZedBoard + FMCOMMS1/2/3/4
  - PicoZed SDR

测试

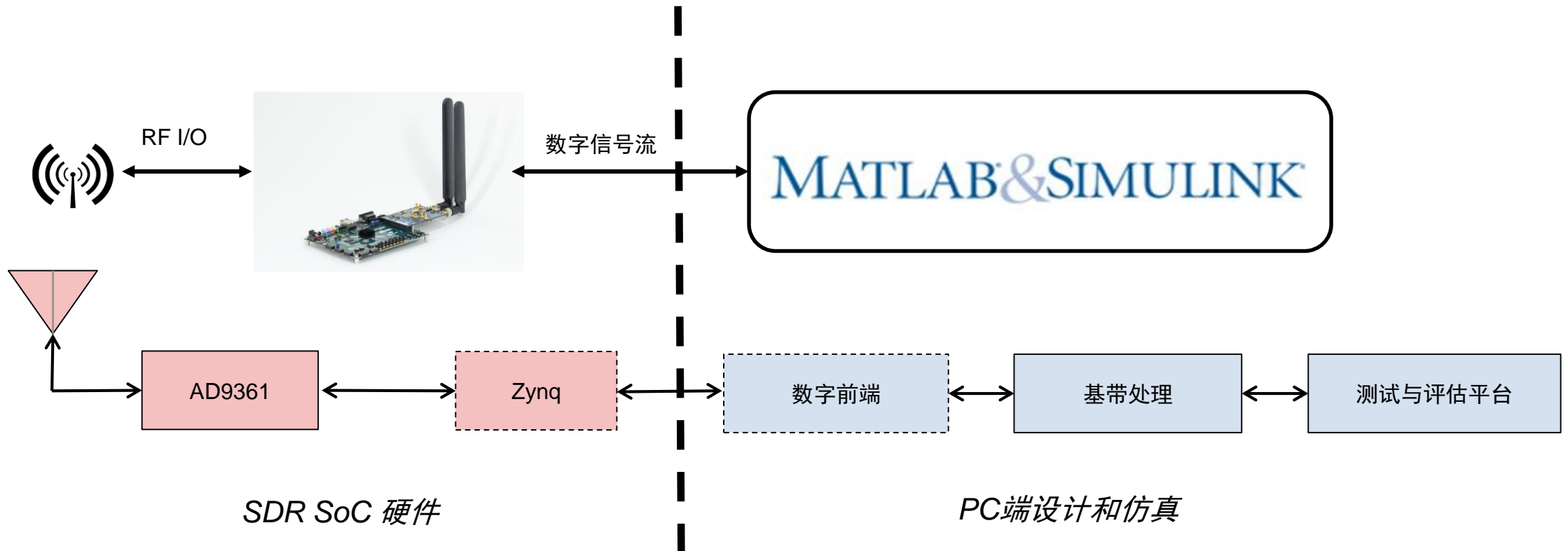


# SDR Radio I/O 设计流程

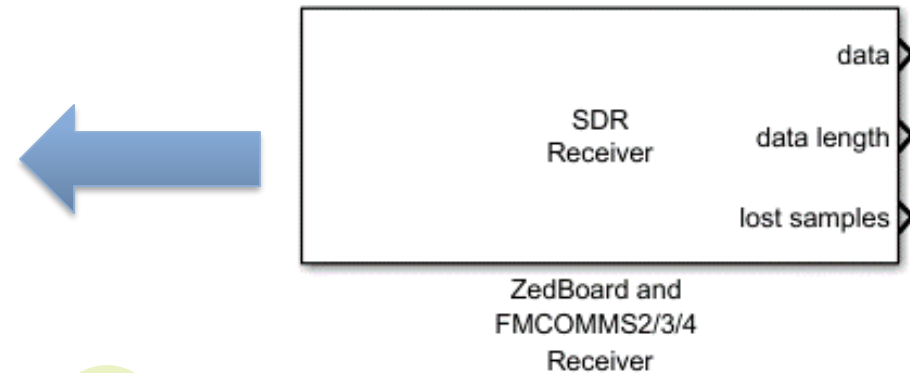
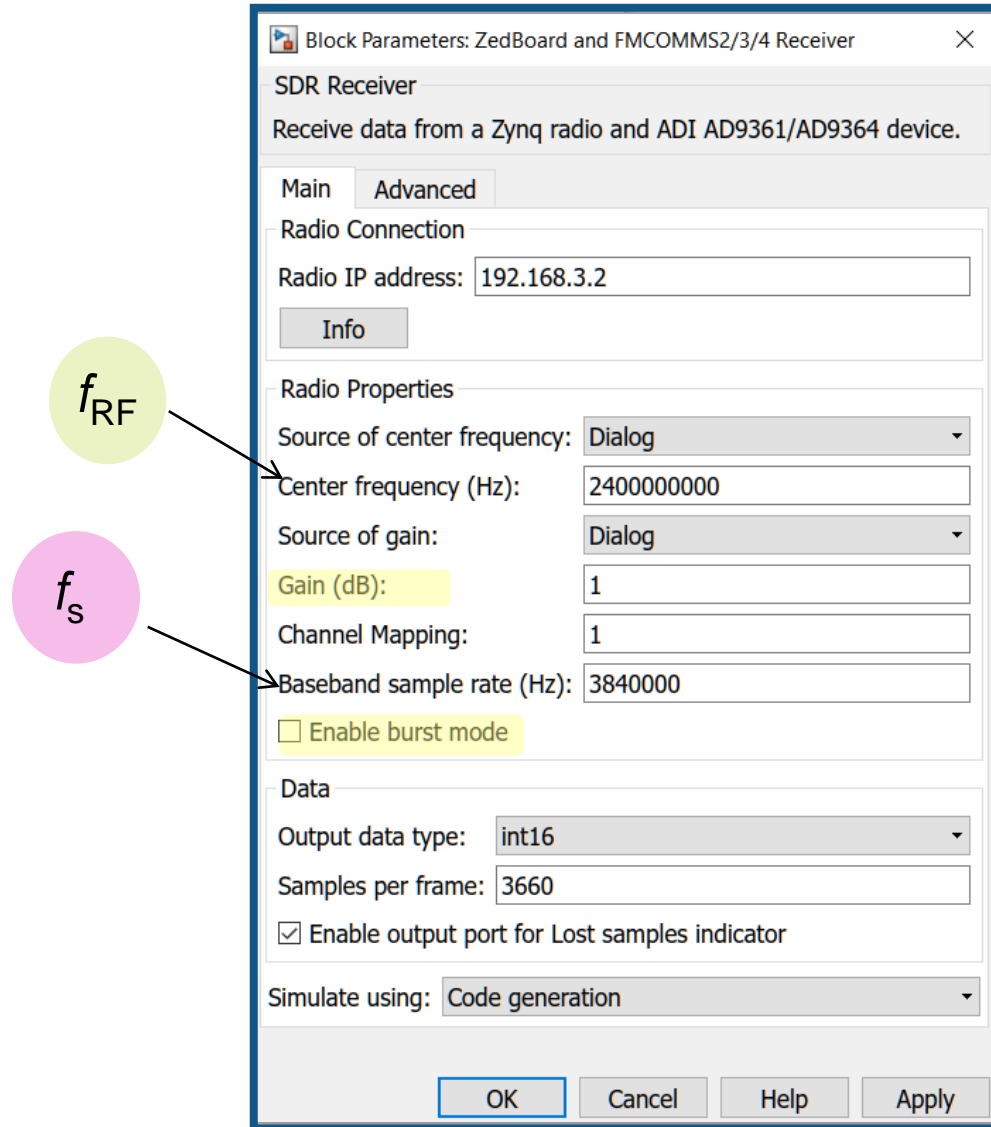
- *Over-the-air* 测试

# SDR Radio I/O

- 使用**真实**的无线电信号开发基带算法
- 射频收发器**参数可调**
- 轻松的开盒即用体验



# Zynq-SDR Radio I/O 接口模块 (Simulink)



$f_{RF}$  RF中心频率 (70MHz to 6GHz)

$f_s$  采样频率 (上至 61.44MHz)

可调增益, Burst 模式

# Zynq-SDR Radio I/O 接口系统对象 (MATLAB)

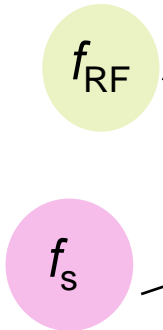
```

% Create receiver system object for radio hardware
rx = sdr_rx('ZedBoard and FMCOMMS2/3/4')

rx =
  comm.SDRRxZedBoardFMC234 with properties:

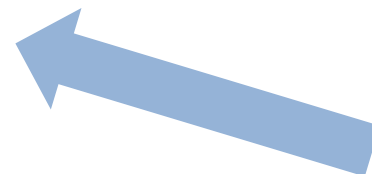
  Main
      DeviceName: 'ZedBoard and FMCOMMS2/3/4'
      IPAddress: '192.168.3.2'
      GainSource: 'AGC Slow Attack'
      CenterFrequency: 2.4000e+09
      ChannelMapping: 1
      BasebandSampleRate: 3840000
      EnableBurstMode: false
      OutputDataType: 'int16'
      SamplesPerFrame: 3660

  Show all properties
  
```



**Radio IO System Object**

- comm.SDRRxPicoZedSDR
- comm.SDRTxPicoZedSDR
- comm.SDRRxZC706FMC234
- comm.SDRTxZC706FMC234
- comm.SDRRxZedBoardFMC234
- comm.SDRTxZedBoardFMC234



访问 MATLAB 帮助文档, 了解 Zynq System Object.

```

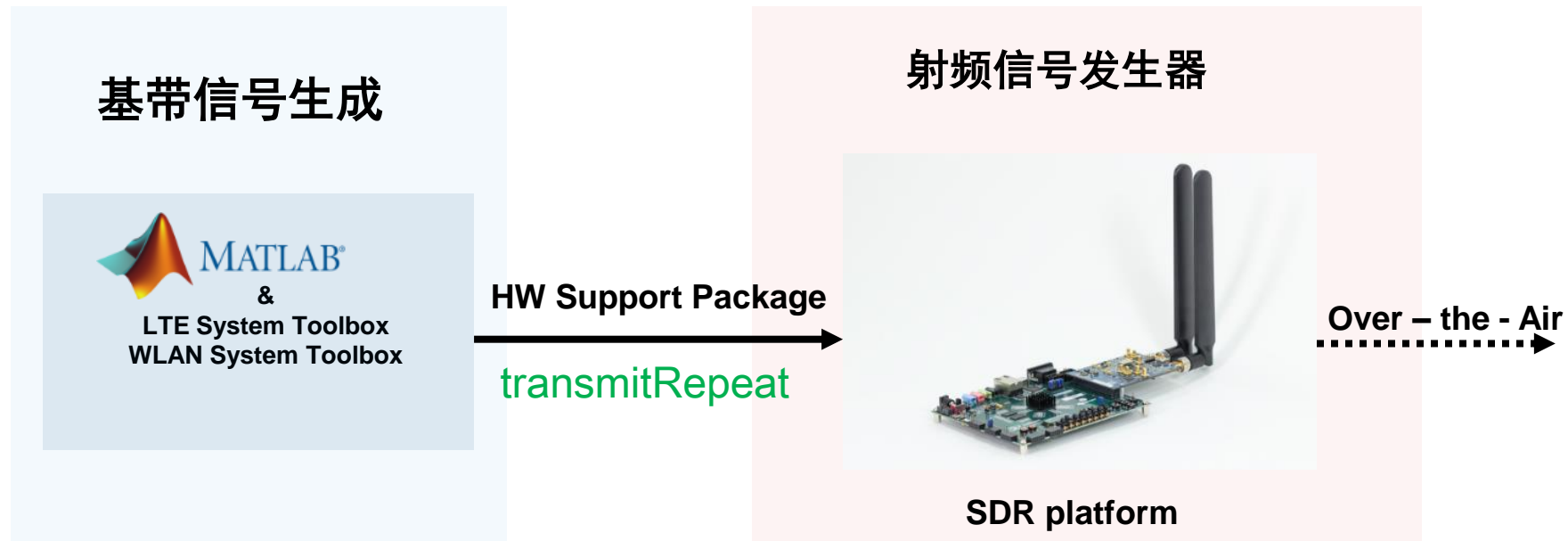
>>
>>
>> doc sdr_rx
>> doc sdr_tx
fx >>
  
```

- $f_{RF}$  RF 中心频率 (70MHz to 6GHz)
- $f_s$  采样频率 (上至 61.44MHz)
- 可调增益, Burst 模式

# SDR 波形重复发射功能

- Repeated Waveform Transmitter 模式
  - 参考信号由 MATLAB 预生成
  - 存储于 SDR 硬件，并由发送天线重复、连续发送
  - 符合标准的 LTE、WLAN 测试波形

```
tx = sdrtx('ZedBoard and FCOMMS2/3/4', ...  
          'BasebandSampleRate',    520.841e3, ...  
          'CenterFrequency',       2.4e9, ...  
          'ShowAdvancedProperties', true, ...  
          'BypassUserLogic',       true);  
  
transmitRepeat(tx, zynqRadioQPSKTransmitData);
```





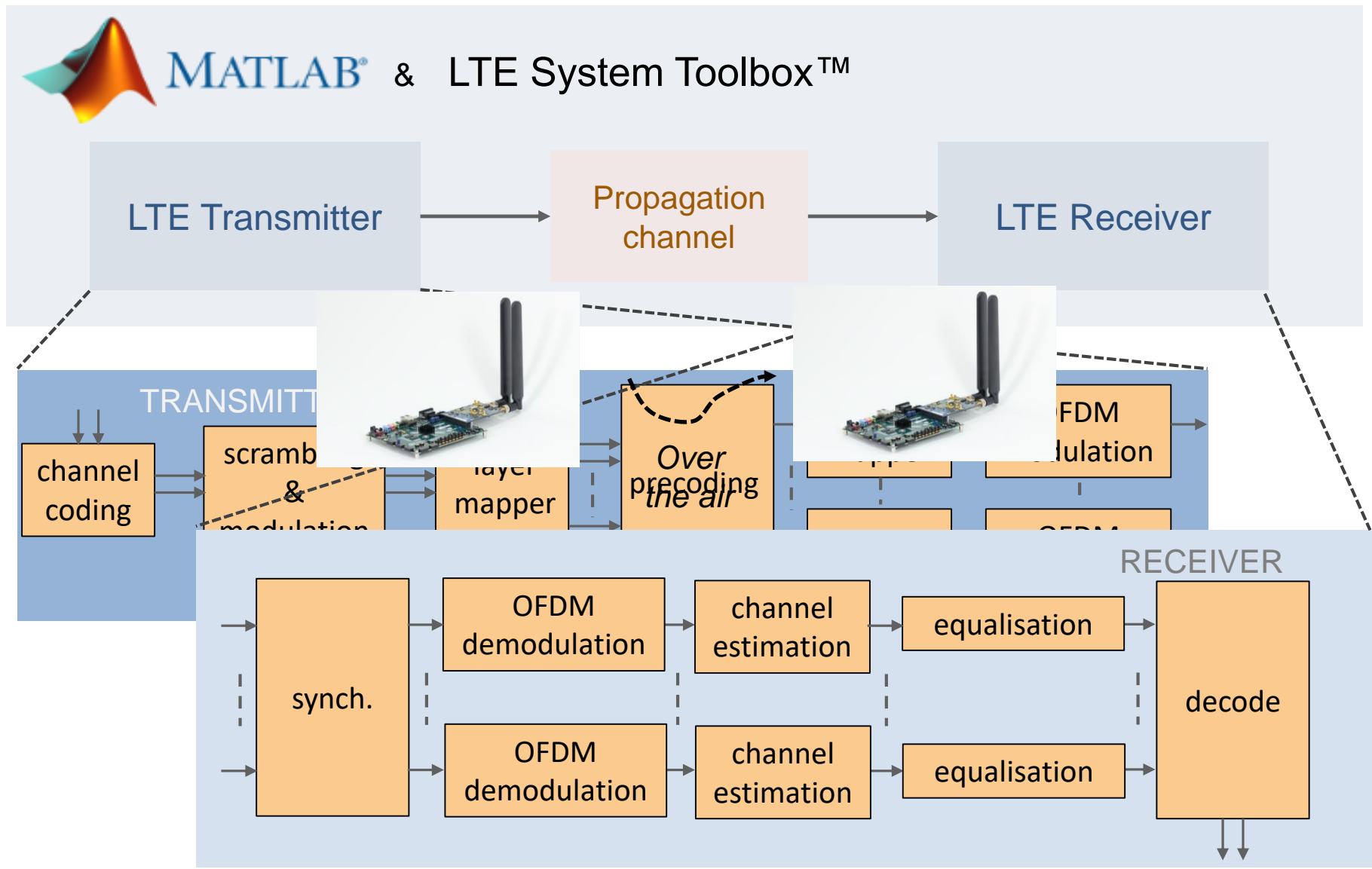
## Demo: LTE 发射分集 - Zynq (Zedboard) 和 AD9361 (2x2)

- 使用一个或两个天线，利用“Repeated Waveform Transmission”模式发送图片
- 使用一个或两个天线，解调信号、显示图像并且显示传输统计结果（BER 和 EVM）



[Available here](#)

# LTE 物理层处理链



## Demo: LTE 发射分集 - Zynq (Zedboard) 和 AD9361 (2x2)

```
% Initialize SDR device
txsim = struct; % Create empty structure for transmitter
txsim.SDRDeviceName = 'ZedBoard and FMCOMMS2/3/4'; % Set SDR Device
hdev = sdrdev(txsim.SDRDeviceName); % Create SDR device object

txsim.RC = 'R.7'; % Base RMC configuration, 10 MHz bandwidth
txsim.NCellID = 88; % Cell identity
txsim.NFrame = 700; % Initial frame number
txsim.TotFrames = 1; % Number of frames to generate
txsim.DesiredCenterFrequency = 2.45e9; % Center frequency in Hz
txsim.NTxAnts = 2; % Number of transmit antennas
txsim.Gain = -50;
```

[Available here](#)

## Demo

- 使用  
图片
- 使用  
EVM

```
% Create RMC
rmc = lteRMCDL(txsim.RC);

% Customize RMC parameters
rmc.NCellID = txsim.NCellID;
rmc.NFrame = txsim.NFrame;
rmc.TotSubframes = txsim.TotFrames*10; % 10 subframes per frame
rmc.CellRefP = txsim.NTxAnts; % Configure number of cell reference ports
rmc.PDSCH.RVSeq = 0;

% If transmitting over two channels enable transmit diversity
if rmc.CellRefP == 2
    rmc.PDSCH.TxScheme = 'TxDiversity';
    rmc.PDSCH.NLayers = 2;
    rmc.OCNGPDSCH.TxScheme = 'TxDiversity';
end

% Pack the image data into a single LTE frame
[eNodeBOutput, txGrid, rmc] = lteRMCDLTool(rmc, trData);
```

模式发送

(BER 和



## Demo: LTE 发射分集 - Zynq (Zedboard) 和 AD9361 (2x2)

```
% *Prepare for Transmission*
```

```
tx = sdrtx(txsim.SDRDeviceName);
```

```
tx.BasebandSampleRate = rmc.SamplingRate; % 15.36 Msps for default RMC (R. 7  
% with a bandwidth of 10 MHz
```

```
tx.CenterFrequency = txsim.DesiredCenterFrequency;
```

```
tx.ShowAdvancedProperties = true;
```

```
tx.BypassUserLogic = true;
```

```
tx.Gain = txsim.Gain;
```

```
tx.transmitRepeat(eNodeBOutput);
```

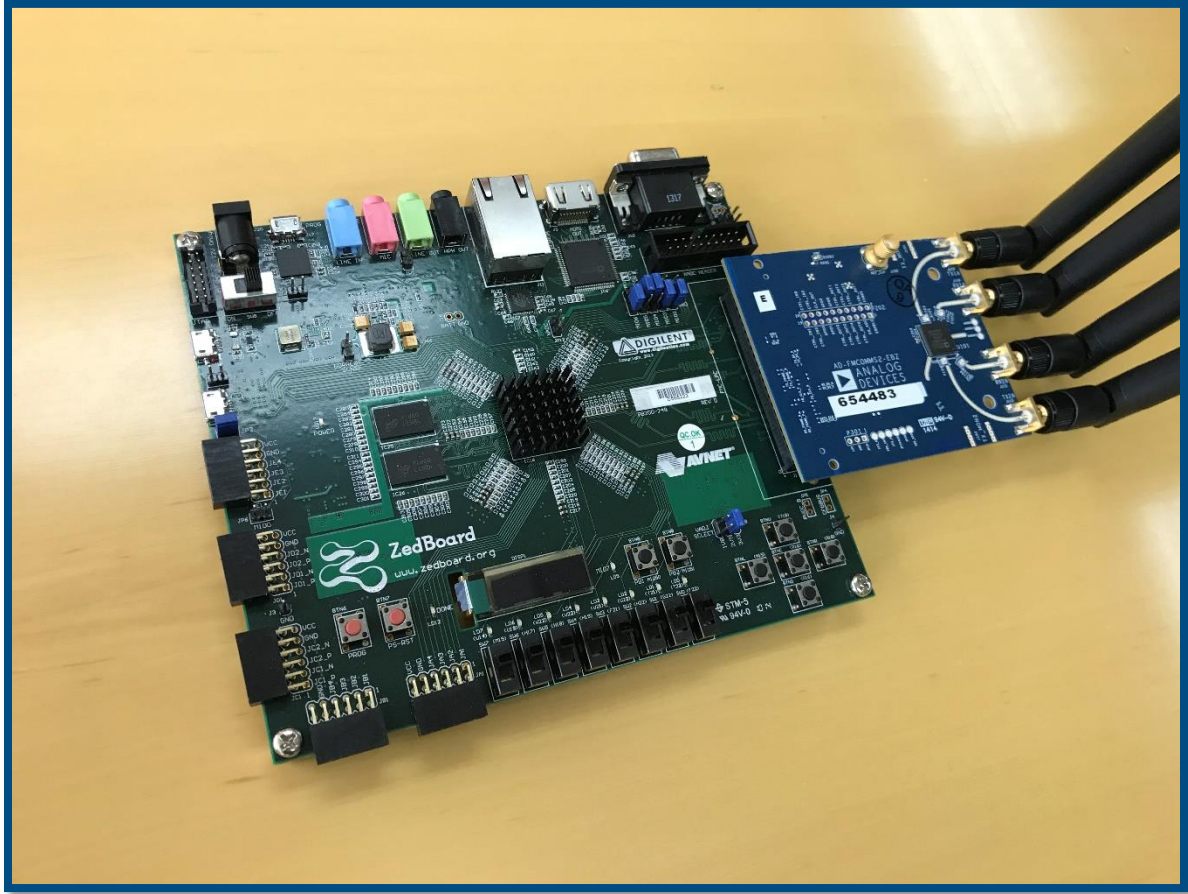
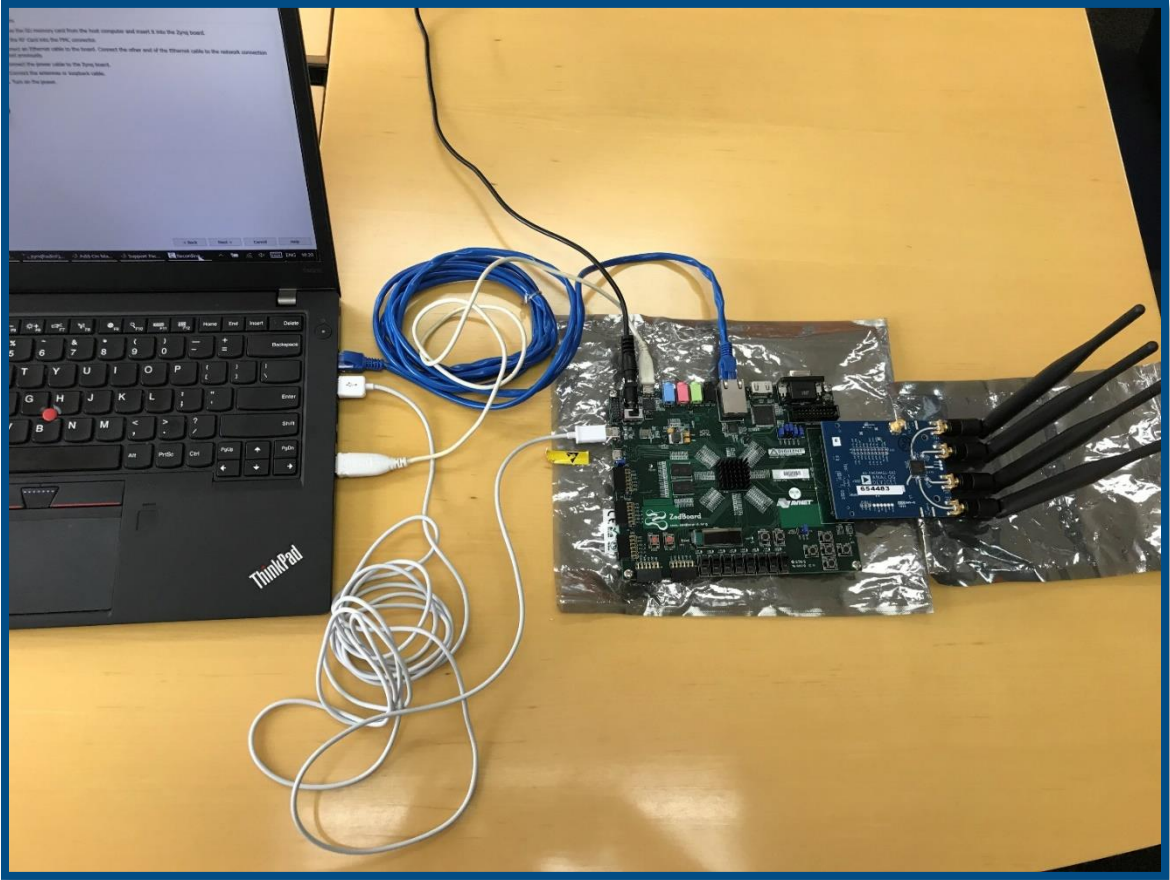


```
De rx = sdr_rx(rxsim.SDRDeviceName);  
rx.BasebandSampleRate = rxsim.RadioFrontEndSampleRate;  
rx.CenterFrequency = rxsim.RadioCenterFrequency;  
▪ rx.SamplesPerFrame = samplesPerFrame;  
图 rx.OutputDataType = 'double';  
▪ rx.EnableBurstMode = true;  
E rx.NumFramesInBurst = rxsim.FramesPerBurst;  
  
while rxsim.numBurstCaptures  
    for frame = 1:rxsim.FramesPerBurst  
        while len == 0  
            % Store one LTE frame worth of samples  
            [data, len, lostSamples] = step(rx);  
            burstCaptures(:, :, frame) = data; %#ok<SAGROW>  
        end  
    end
```

[Available here](#)



# Demo: Zynq (Zedboard) 和 AD9361 (2x2)





## SDR Radio I/O 设计流程小结

- 利用 [Zynq SDR Support from Communications System Toolbox](#), 你可以直接将 AD9361 的数据传输到 MATLAB 和 Simulink 中
  - 支持包还可以实现收发器的参数控制
- SDR 硬件作为 over-the-air 信号捕捉或者传输的射频前端
- 在 PC 端, 使用 MATLAB/Simulink 搭建快速验证平台
  - 真实的模型信号作为输入激励/生成输出
  - LTE、WLAN 标准无线电波形
- Zynq 上没有代码运行 – 基带算法在 MATLAB 和 Simulink 上实现
  - Zynq 的 ARM 处理器打包数据, 并通过以太网传输



# SDR HW/SW Co-design 设计流程

- 基于 SoC 硬件的原型实现

# 如果你的 SDR 设计原型需要 FPGA 和 ARM?

- 我们如何：
  - 将设计部署到 SoC?
  - 划分 FPGA 硬件设计和 ARM 软件设计?
  - 设计 FPGA 和 ARM 接口?
  - 设计 SoC 硬件与射频卡接口?
  - 开发 ARM/FPGA 算法?

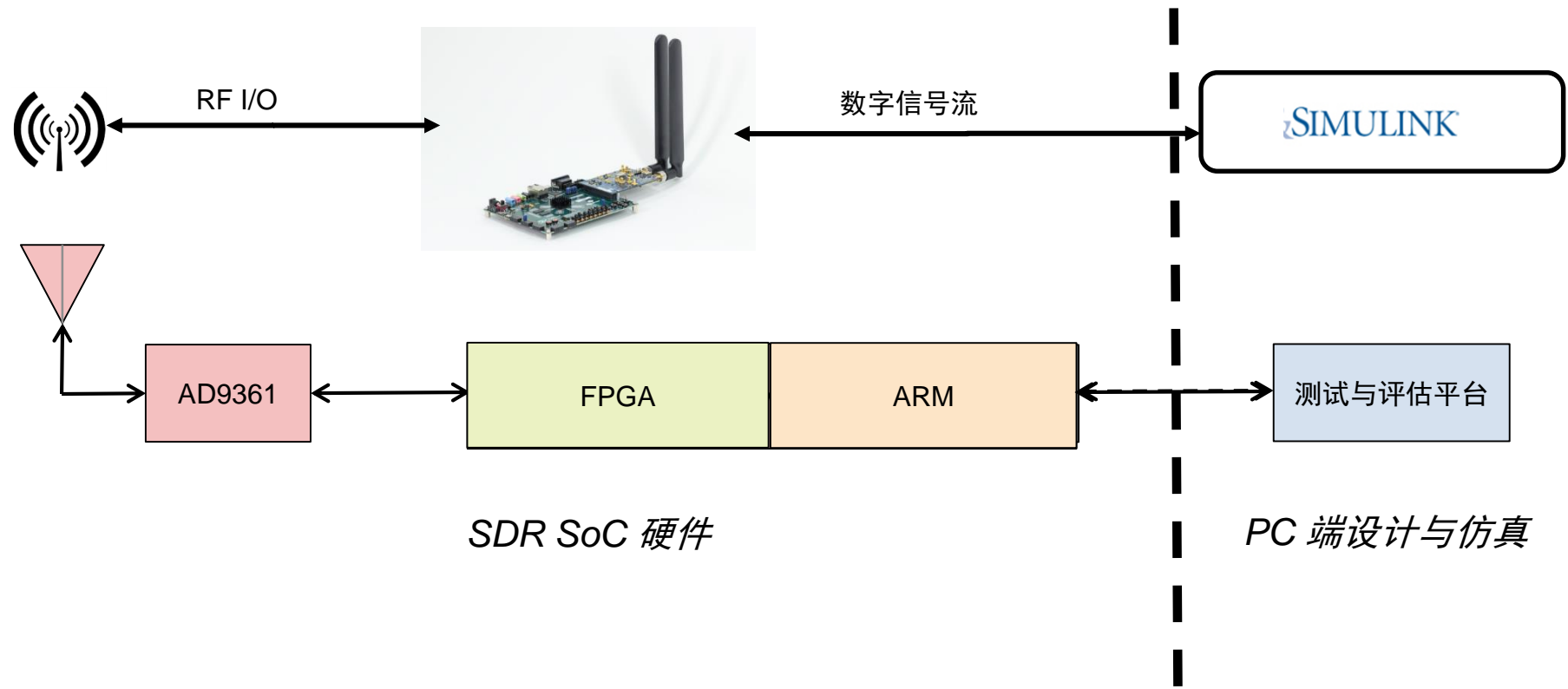
MATLAB/Simulink  
Algorithm Design



SoC FPGA's

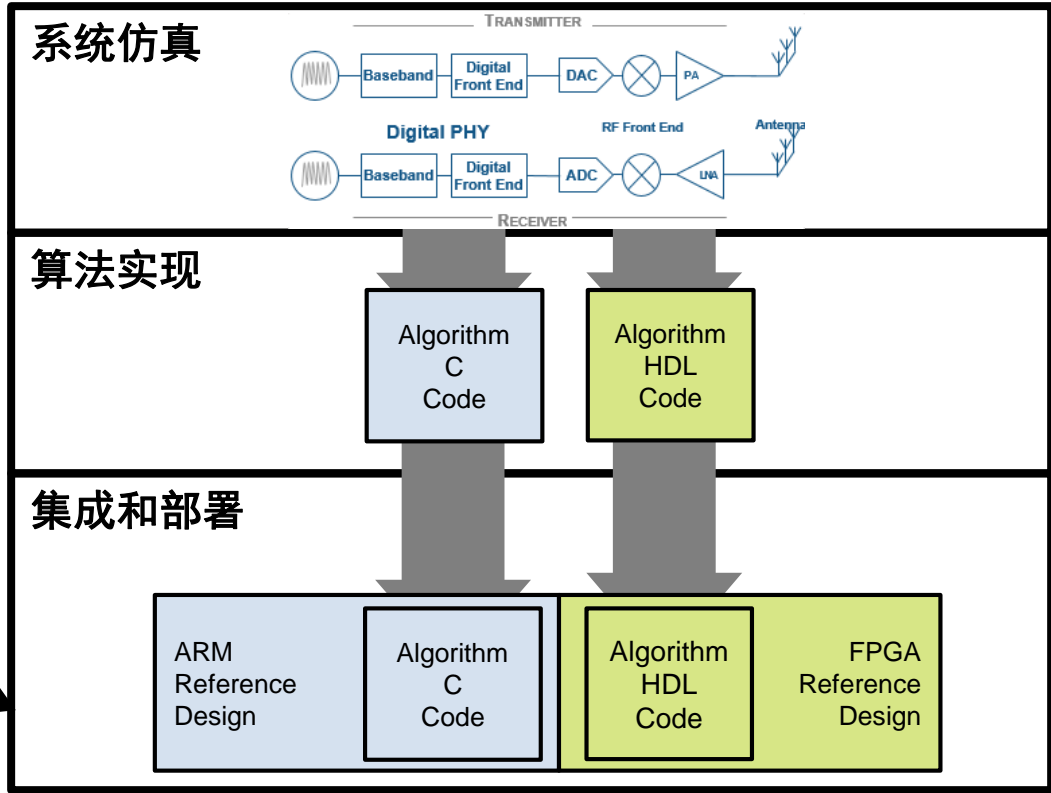
# SDR HW/SW Co-design

- SDR 设计系统级部署与实现
- 实时无线电信号验证与测试
- HDL 和 C 代码自动生成



# 基于模型设计： - 高级 SoC 设计流程

系统级设计与仿真，  
用户定义分区



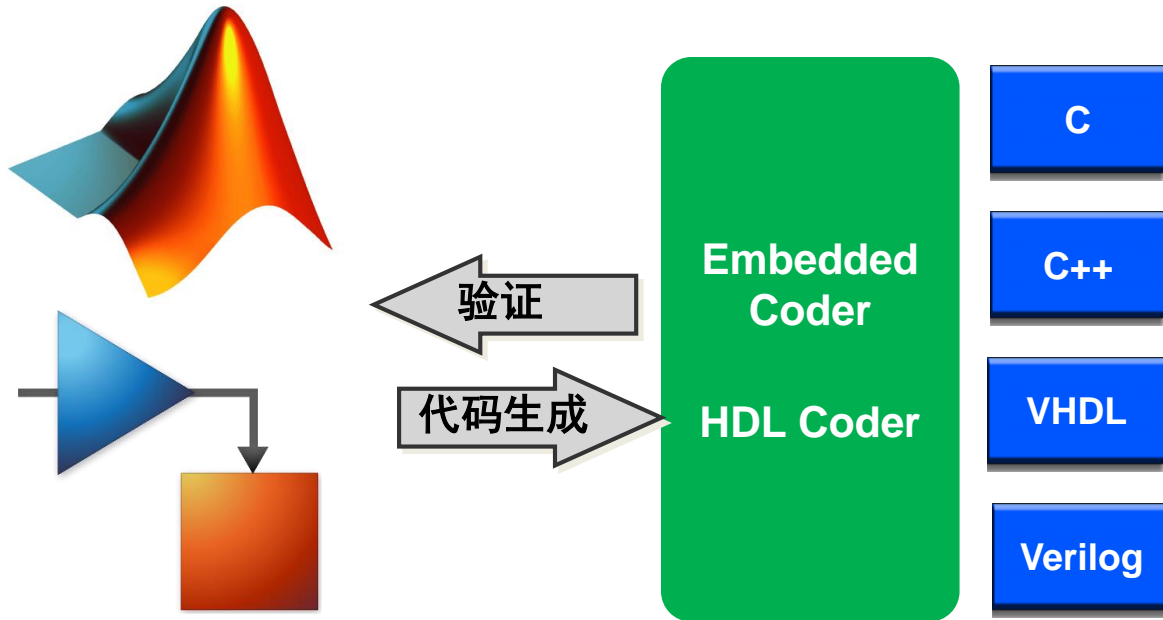
MathWorks 通过  
FPGA 工具实现编译  
和下载自动化

MathWorks  
自动生成代码和接口模型

Verification (PIL)

实时参数  
调优和验证

# 自动代码生成



## Embedded Coder:

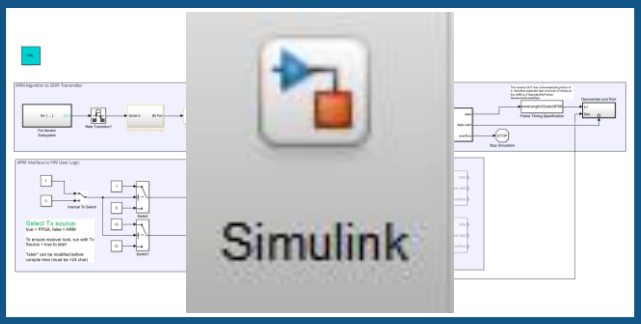
- 优化的 C/C++ 代码
- 适用于 MCU 和 DSP (定点、浮点支持)
- SIL、PIL 仿真

## HDL Coder:

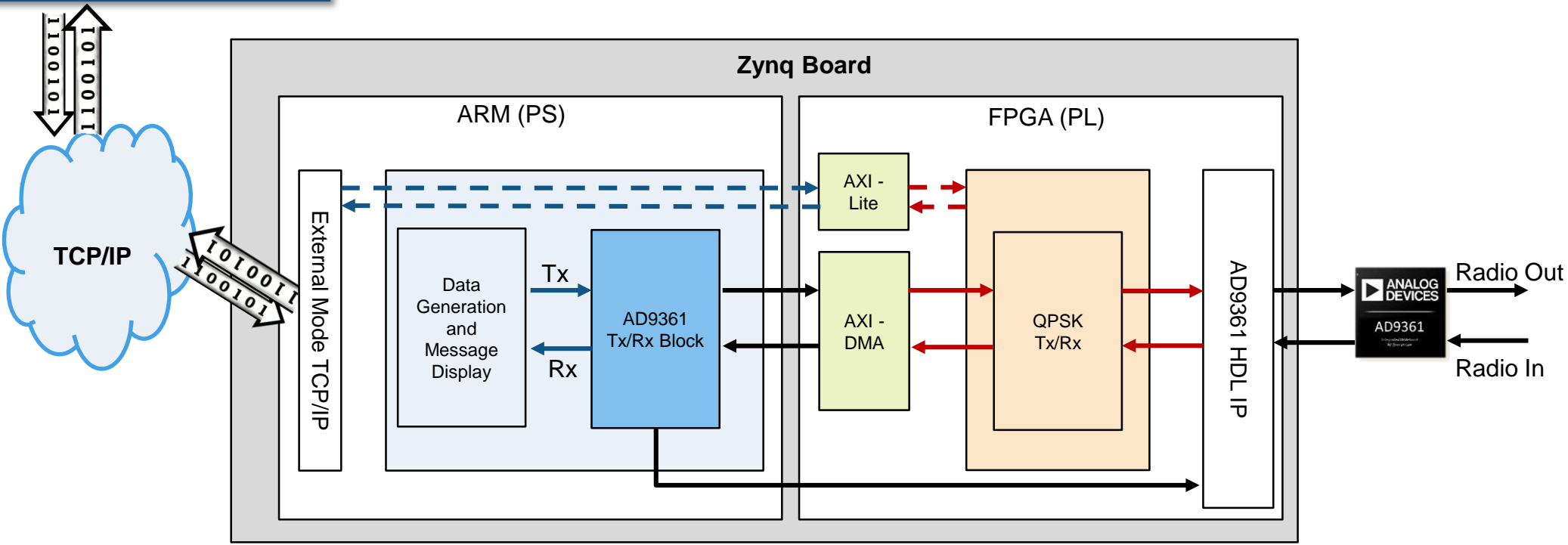
- 目标独立的可综合 HDL 代码
- 适用于 ASIC 和 FPGA (定点、浮点支持)
- 生成 HDL Testbench, FIL 仿真
- ModelSim and Incisive\* 联合仿真
- 集成的 Xilinx and Altera 综合工具接口

\* HDL Verifier 支持 co-simulation 和 FPGA-in-the-loop verification

# Demo: QPSK 收发器设计的HW/SW Co-design 流程

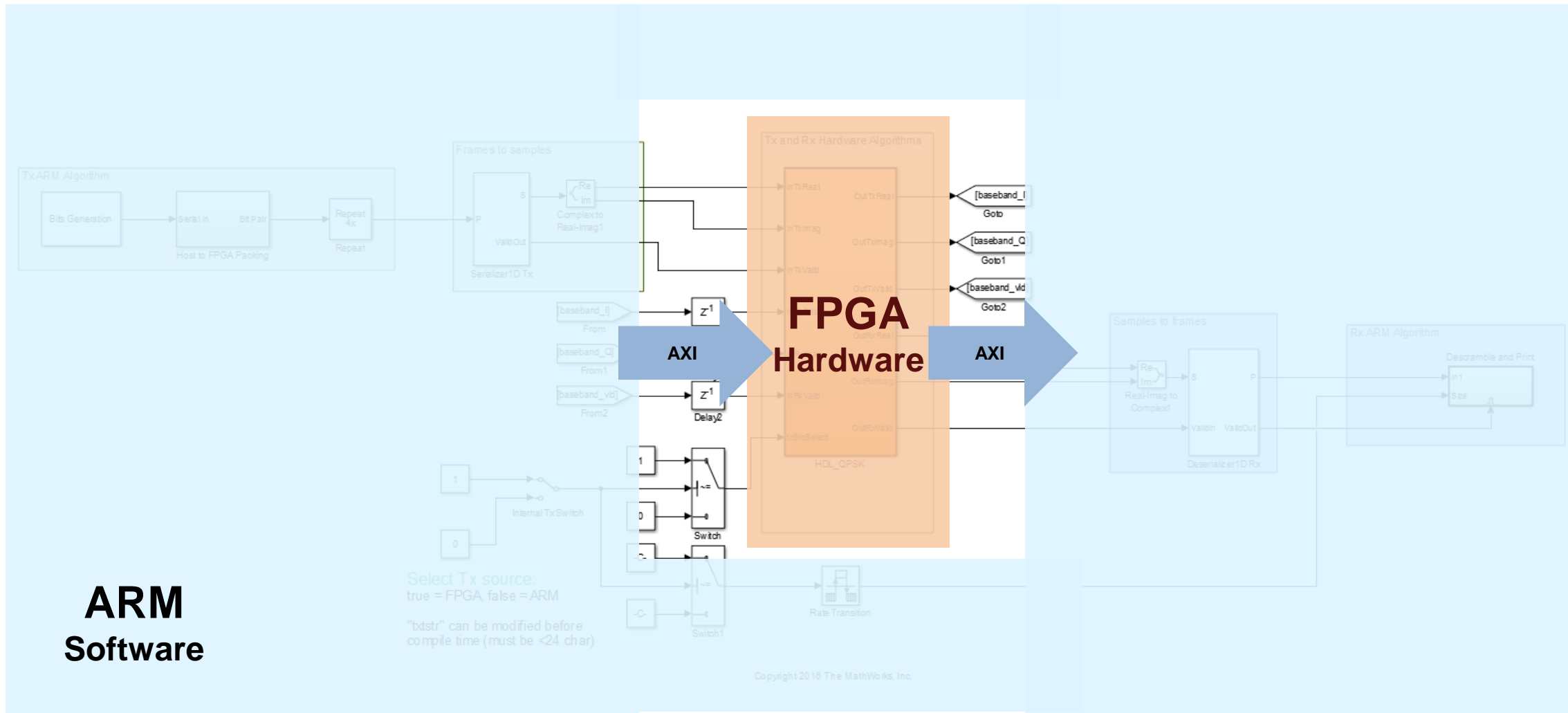


- QPSK 调制/解调部分在 FPGA 实现
- 编解码算法在 ARM 实现
- ARM 部分参数实时可调
- 真实无线电信号收发

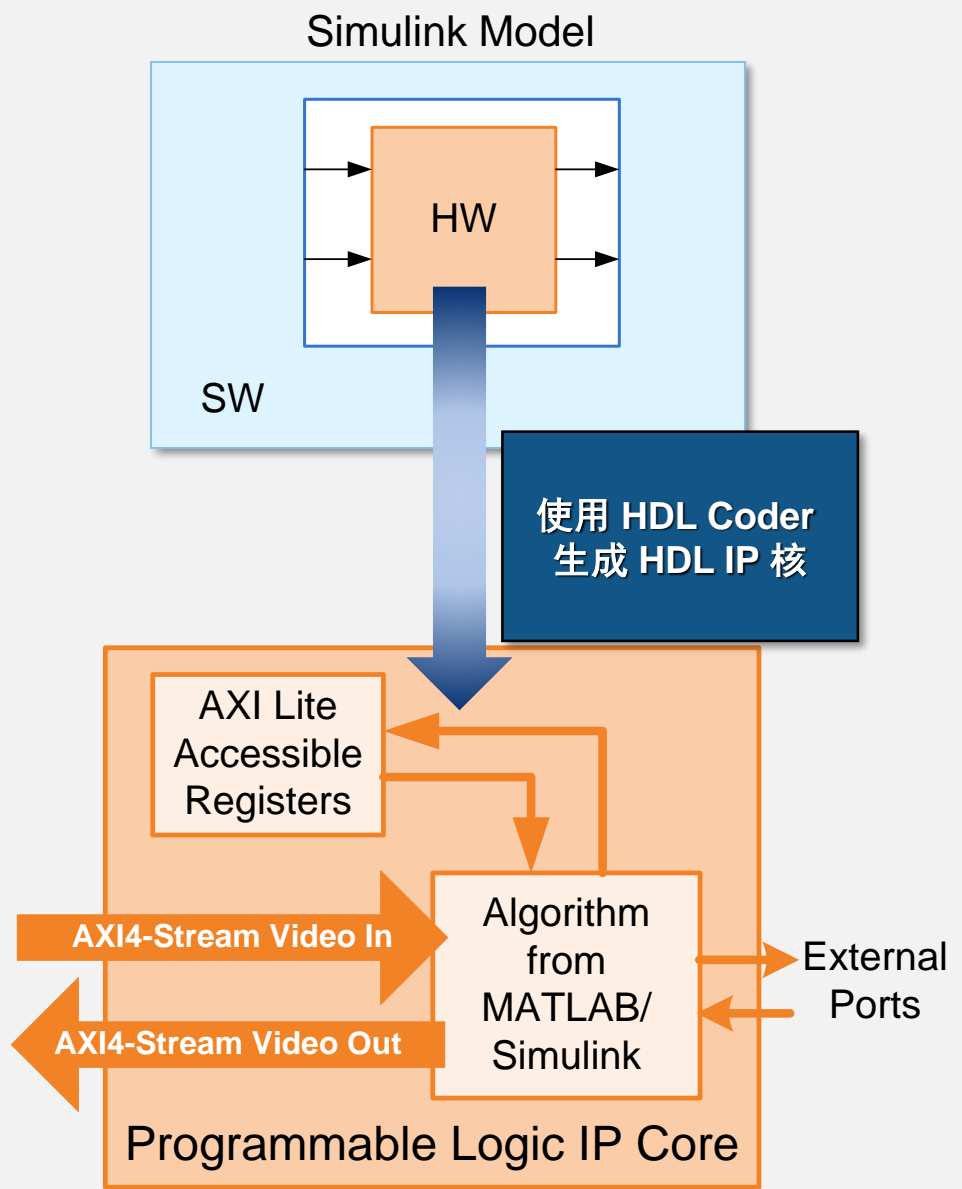




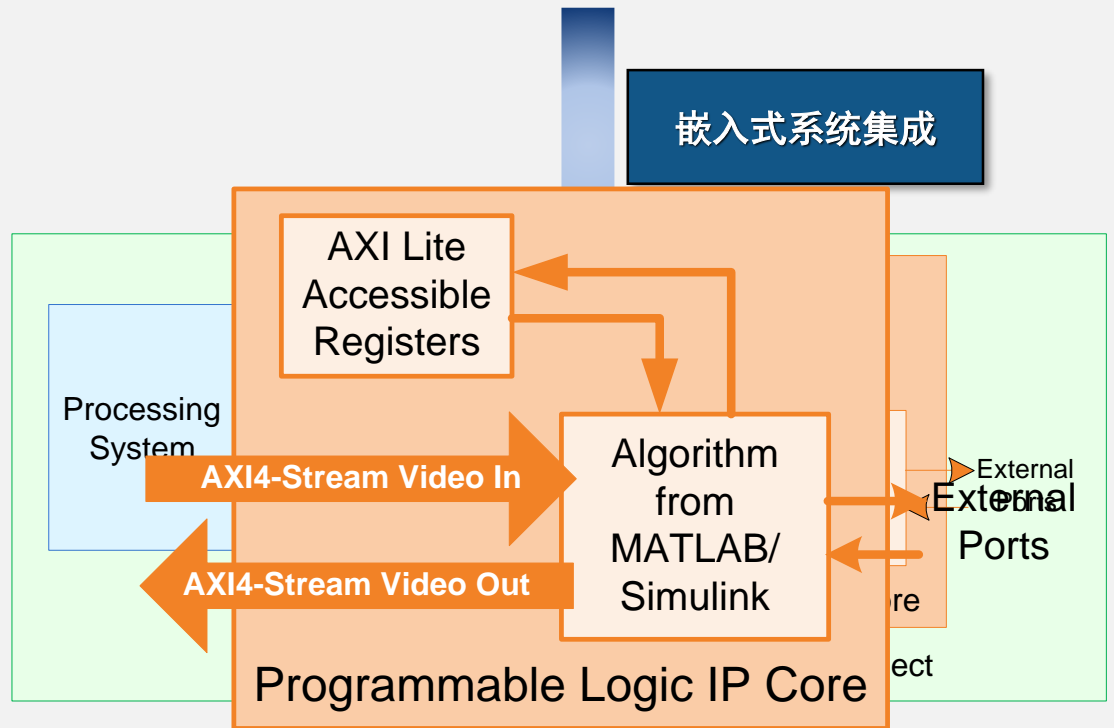
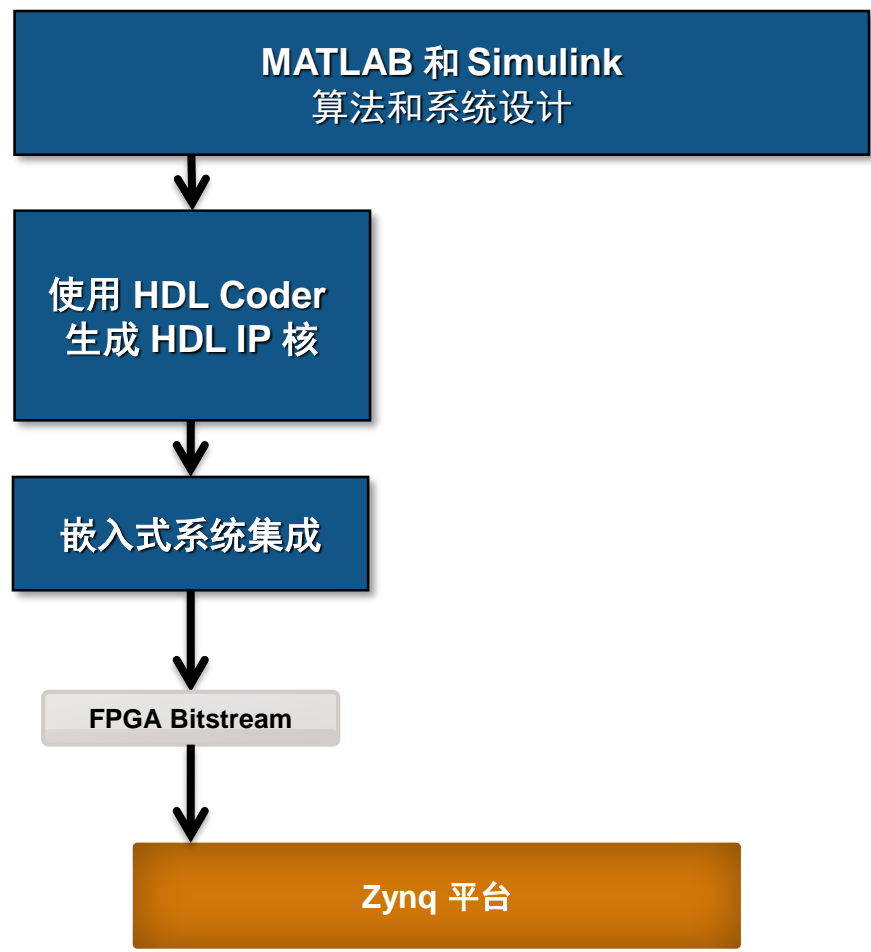
# Demo: QPSK 收发器设计的HW/SW Co-design 流程



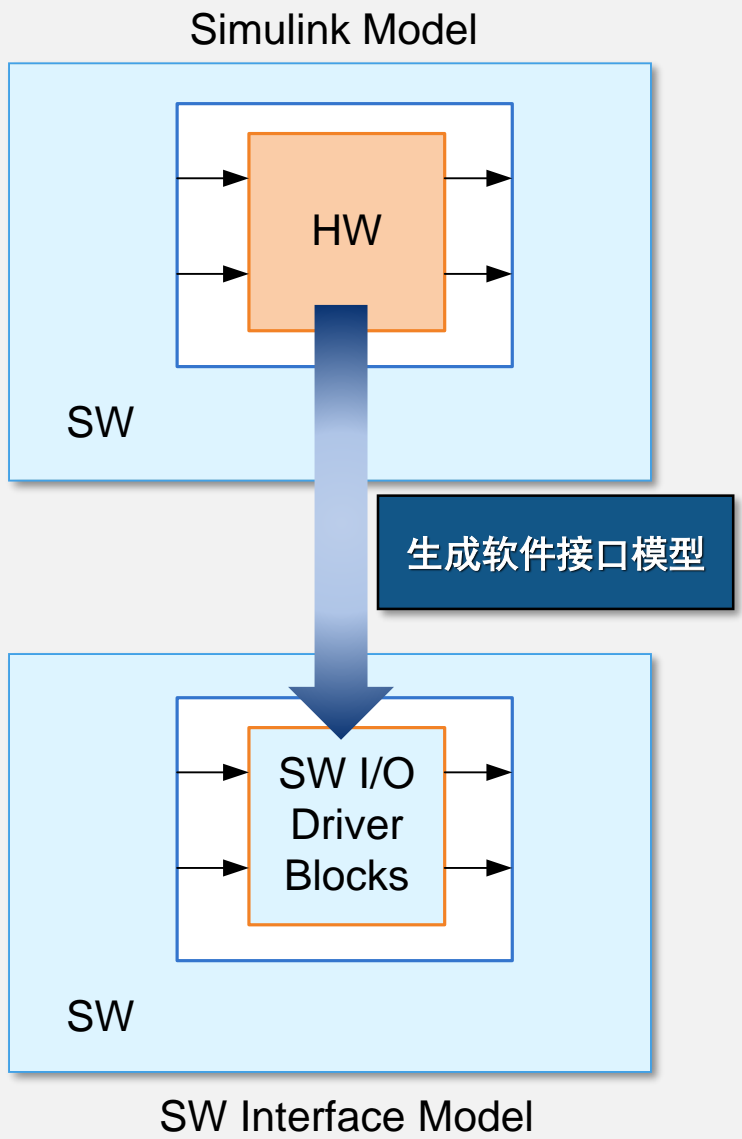
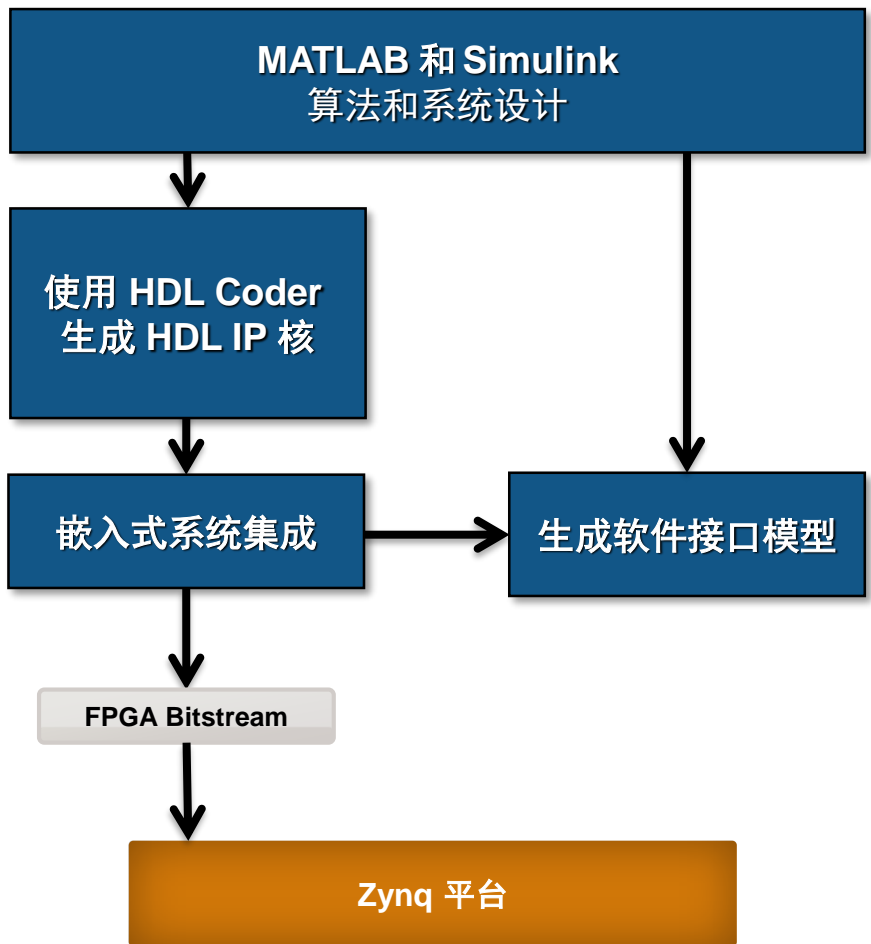
# 1. 生成 HDL IP 核



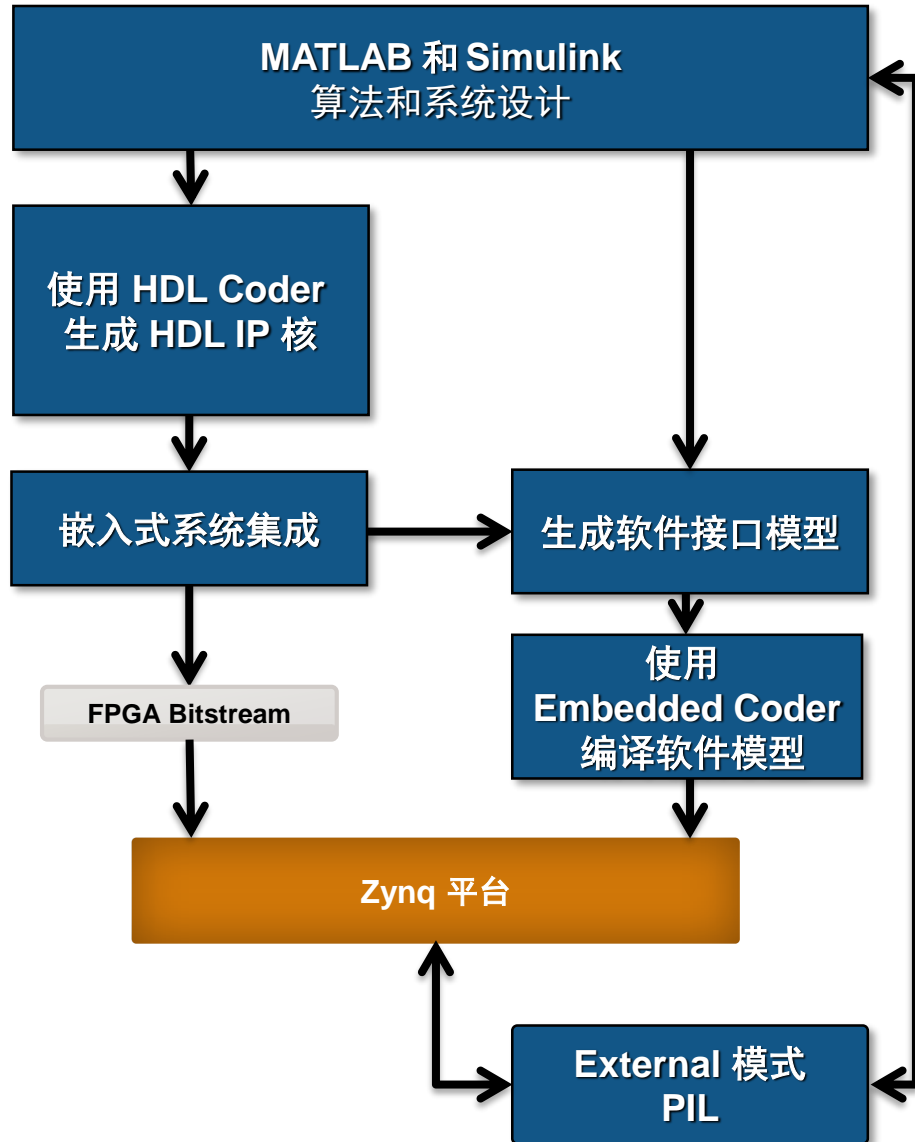
## 2. 嵌入式系统集成



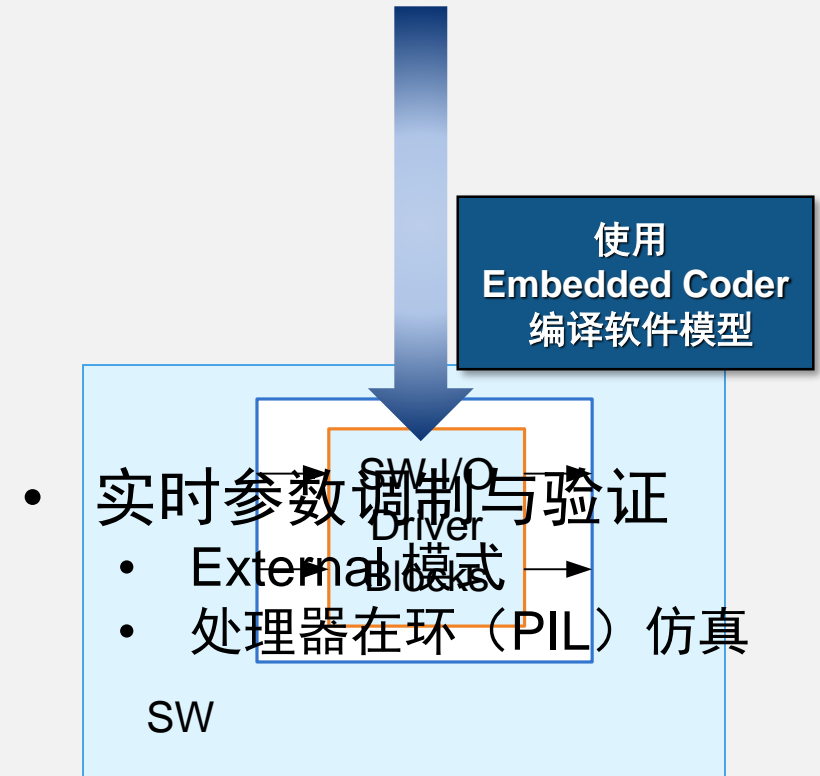
### 3. 生成软件接口模型



# 4. Zynq 平台上实时运行



## SW Interface Model



# 设计自动化：HDL Workflow Advisor 工具

HDL Workflow Advisor - zynqRadioHWSWQPSKAD9361AD9364SL/HDL\_QPSK

File Edit Run Help

Find:

- ▼ HDL Workflow Advisor
  - ▼ 1. Set Target
    - ^1.1. Set Target Device and Synthesis Tool
    - ^1.2. Set Target Reference Design
    - ^1.3. Set Target Interface
  - ▼ 2. Prepare Model For HDL Code Generation
    - 2.1. Check Global Settings
    - ^2.2. Check Algebraic Loops
    - ^2.3. Check Block Compatibility
    - ^2.4. Check Sample Times
  - ▼ 3. HDL Code Generation
    - > 3.1. Set Code Generation Options
    - ^3.2. Generate RTL Code and IP Core
  - ▼ 4. Embedded System Integration
    - 4.1. Create Project
    - 4.2. Generate Software Interface Model
    - 4.3. Build FPGA Bitstream
    - 4.4. Program Target Device

### 1.1. Set Target Device and Synthesis Tool

Analysis (^Triggers Update Diagram)

Set Target Device and Synthesis Tool for HDL code generation

Input Parameters

Target workflow: IP Core Generation

Target platform: ZedBoard and FMCOMMS2/3/4

Synthesis tool: Xilinx Vivado

Family: Zynq

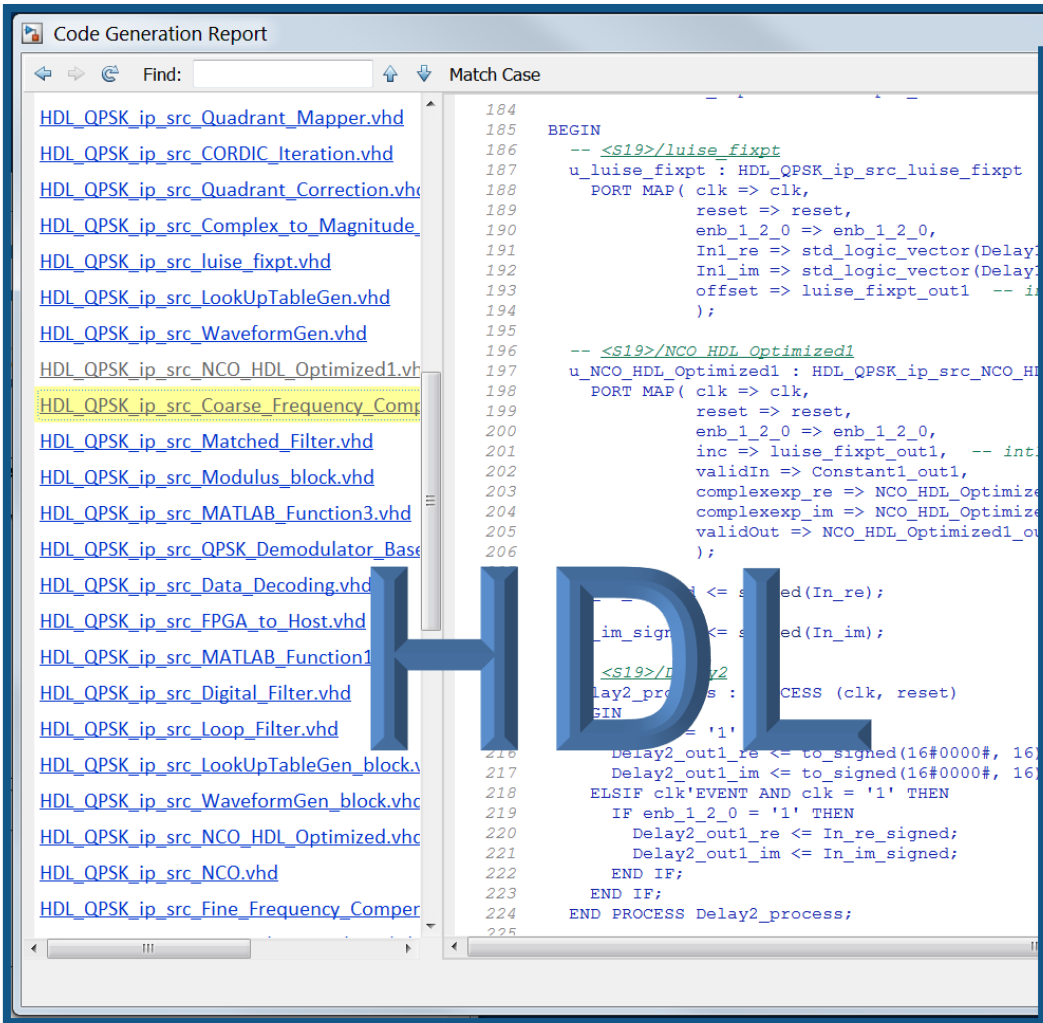
Package: clg484

Project folder: hdl\_prj

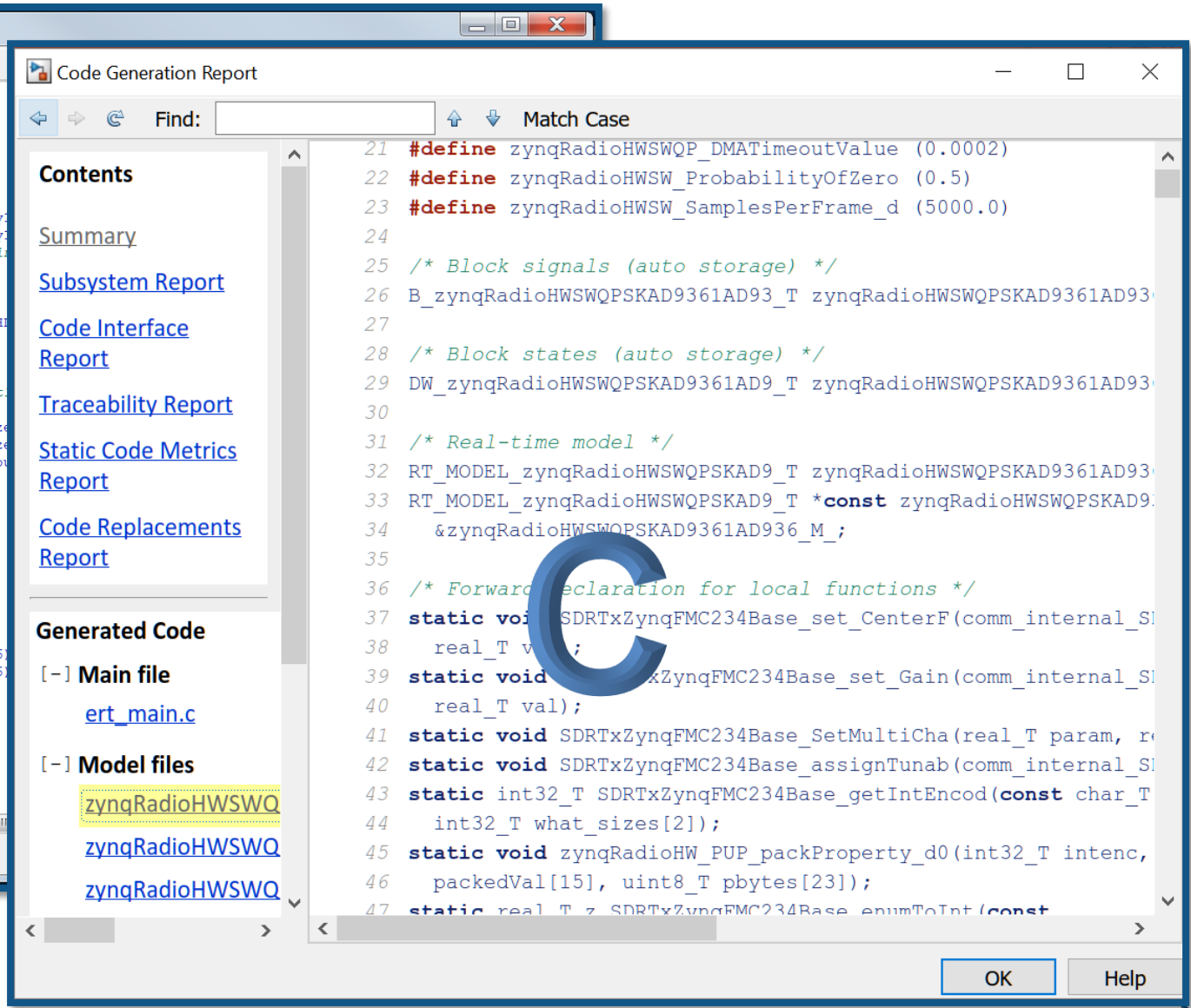
Result:

Click **Run This Task**.

# 模型、代码双向可追踪

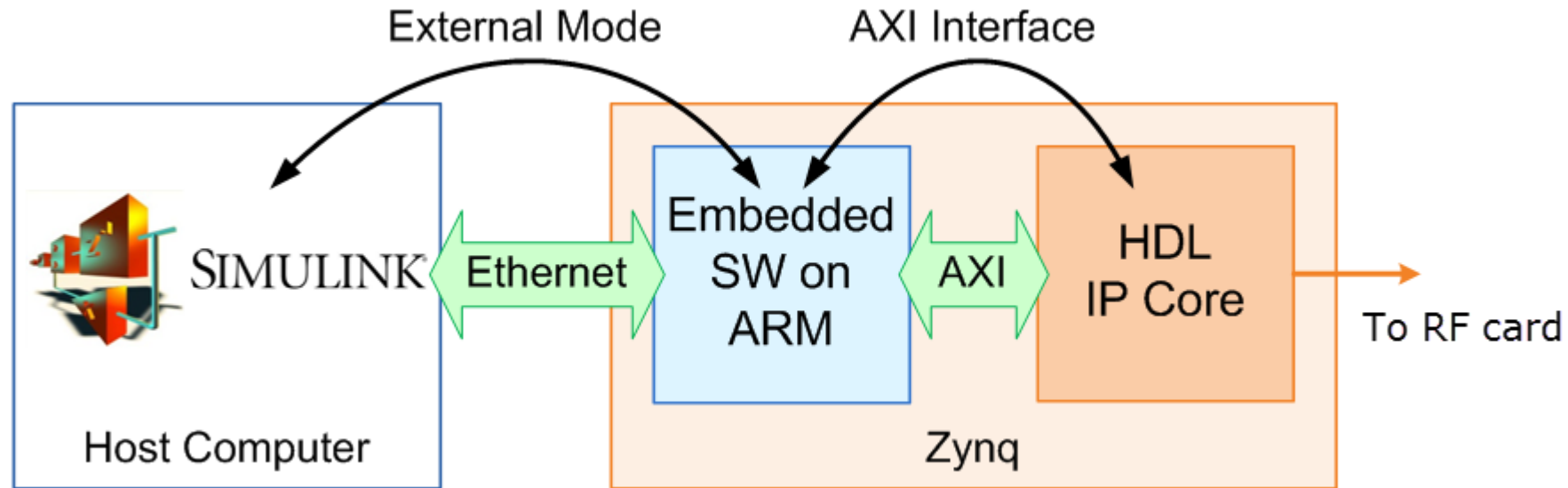


# HDL



# C

# Zynq HW/SW Co-Design 设计流程小结



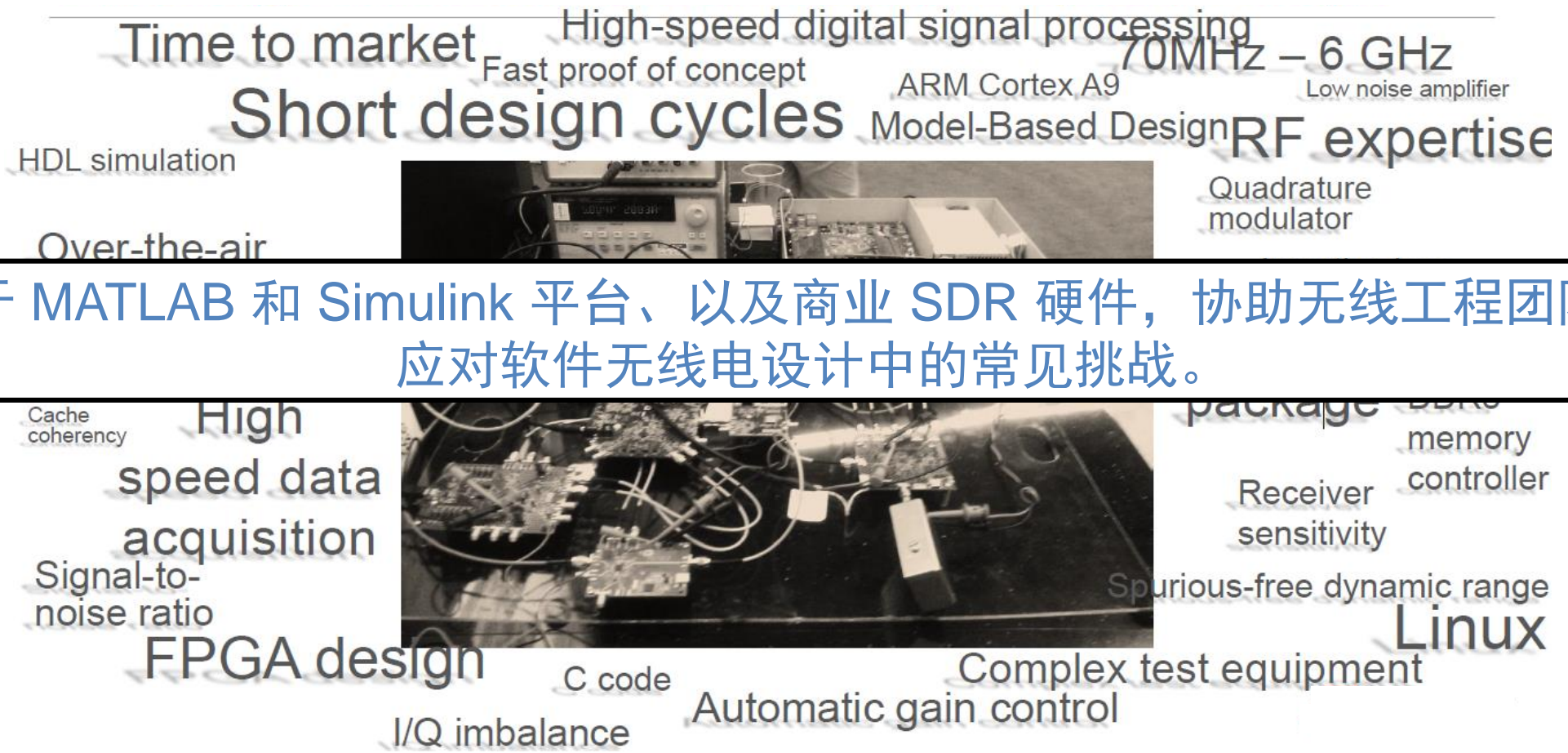
- 基于标准的 Zynq HW/SW Co-design 构架
  - ARM : 与 Simulink 平台数据传输 + 软硬件接口 + 软件算法
  - FPGA : 硬件算法 + 实时无线电信号收发
  - 利用 Simulink 的 External 模式实现参数调制和数据可视化



# 化繁为简：MATLAB/Simulink 的软件定义无线电设计

- 从系统建模仿真到硬件部署
  - SDR Radio I/O
  - SDR HW/SW Co-design
- 多个团队共同的统一开发环境、精简开发流程
- 针对现成硬件开发原型 – 具有生产的路径
  - 无需丰富的 HDL 开发经验，即可实现快速原型开发
  - 通过目标原型进行早期验证
  - 快速代码迭代允许更好的设计权衡
  - 模型与测试平台灵活复用
- 在统一的设计环境中包含所有系统元素
  - 射频
  - 基带算法
  - FPGA 编程
  - 软件接口开发
  - ...

# 我们的目标



基于 MATLAB 和 Simulink 平台、以及商业 SDR 硬件，协助无线工程团队从容应对软件无线电设计中的常见挑战。

**谢谢！**