EDI CON 2018
Electronic Design Innovation Conference & Exhibition
October 17-19 2018
Santa Clara Convention Center
Santa Clara, CA

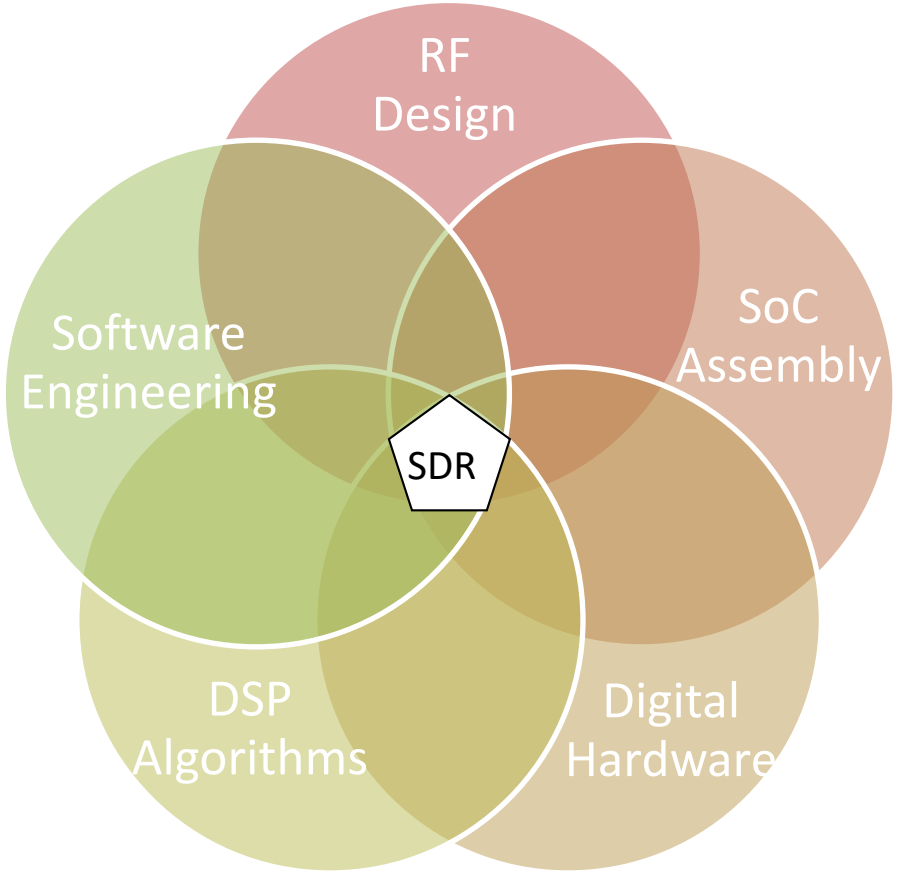
A Traceable Workflow for Software Defined Radio Development

Travis Collins, PhD
Andrei Cozma, PhD
Analog Devices, Inc.
<travis.collins@analog.com>

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Problems with SDR

- Software Defined Radio (SDR) is the unique combination
  - RF Design
  - SoC Assembly
  - Digital Hardware
  - DSP Algorithms
  - Software Engineering
- Few people are the experts on all aspects
- Academically – there is little overlap



EDI CON 2018
Electronic Design Innovation Conference & Exhibition
October 17-19 2018
Santa Clara Convention Center
Santa Clara, CA

Typical Toolsets and Workflow For Algorithms

Create MATLAB Floating Point Reference Design → Create Python Implementation For Data Streaming → Create C/C++ Fixed-Point Version → Create Verilog/VHDL Version

Create MATLAB Floating Point Reference Design → Stream Data into MATLAB design → Convert to Simulink and use Fixed-Point Tools → Generate HDL From Simulink Model

Create MATLAB Floating Point Reference Design → Create SysGen/DSPBuilder Models → Generate HDL From Simulink Model

Create MATLAB Floating Point Reference Design → Create Python Implementation For Data Streaming → Create C/C++ HLx Version → Generate HDL From HLx Blocks

Hardware

Start with Evaluation Board/Hardware

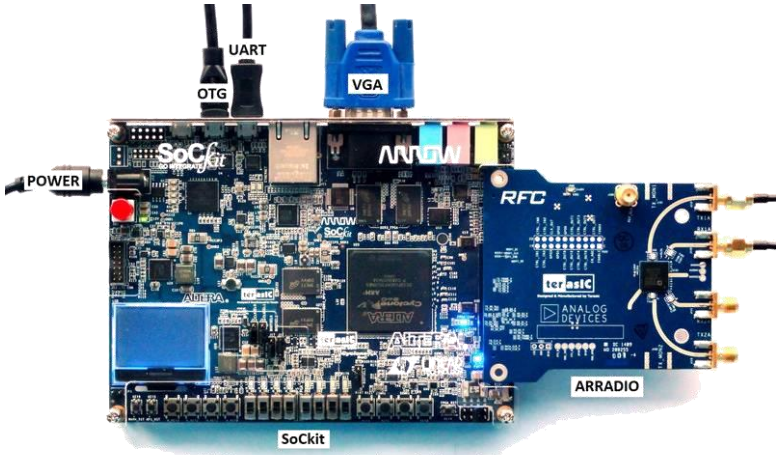
Build Custom Demo/Eval Board To Test Functionality

Build Final Production Board For Deployment

- Hardware
  - AD-FMCOMMS2-EBZ (AD9361)
    - Narrow RF Tuning Range
  - AD-FMCOMMS3-EBZ (AD9361)
    - Wide RF Tuning Range
  - AD-FMCOMMS4-EBZ (AD9364)
    - Narrow and Wide tuning range
  - AARADIO (AD9361)
    - Narrow RF Tuning Range
  - RF SOM (AD9361)
    - Wide RF Tuning Range
- Software
  - Device drivers
    - Linux and/or No-OS
  - FPGA HDL
  - IIO scope
    - Data visualization application
    - Graphical configuration application

- Not enough to make a data link

ZC706 + FMCOMMS2



Arrow SoCKit + ARRADIO

RF SOM + Breakout Board

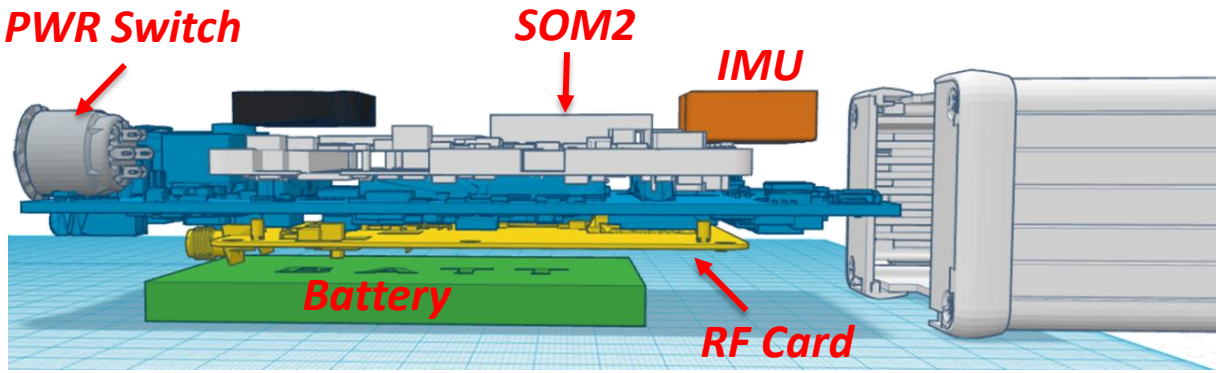
AD936X Transceiver Family

- **AD936X** is the SDR standard for high performance agile transceivers
- **RF-SOM** helps streamline system integration and development
- **PackRF** is a complete deployable system example

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

PackRF Details

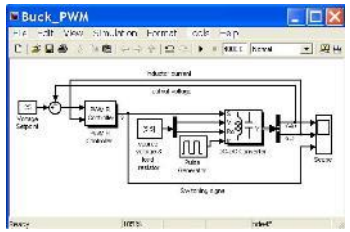
- Example design which shows how to design RF SOM into a custom carrier
- Custom Carrier includes:
  - OLED
  - Nav Switch
  - Power Button
    - Wake on RTC
  - Power over Ethernet (PoE+)
  - Automotive DC-DC converter
    - 8 – 48V DC input
  - Battery Management
  - Hot Power swap
  - Inertial Measurement Unit
    - Six Degrees of Freedom
  - GPS Chipset
    - 1 PPS in and out
  - Audio headset (stereo headphones, mic and button control)



ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Model Based Design Workflow With Hardware

Simulation and Code Generation Tools



Coder Tools

Transitional Hardware

Flexible Infrastructure

EDI CON 2018
Electronic Design Innovation Conference & Exhibition
October 17-19 2018
Santa Clara Convention Center
Santa Clara, CA

Research
- AD9361 Behavioral Simulation
- PlutoSDR Streaming to MATLAB

Algorithm Development
- MATLAB reference implementation
- Hardware streaming

Design Elaboration
- Simulink modeling
- Hardware streaming
- Data type conversion

Prototype
- Deployment to development board
- Design optimization
- HDL Integration
- Driver Integration

Production
- Deployment to custom hardware
- Validation with complete hardware solution

PlutoSDR

Streams over USB
Includes : Host Libraries (libiio, libad9361-iio), GUI Software, GNU Radio and MATLAB application interfaces

RFSoM+FMC Carrier or Eval FMC + FPGA Carrier

Streams over USB/Ethernet, allows access to FPGA and local CPU (standalone operation), blue wire to HW
Includes above plus : Device Drivers, HDL interfaces, HDL libraries, Schematics, Gerber

PackRF or RFSoM + Custom Carrier

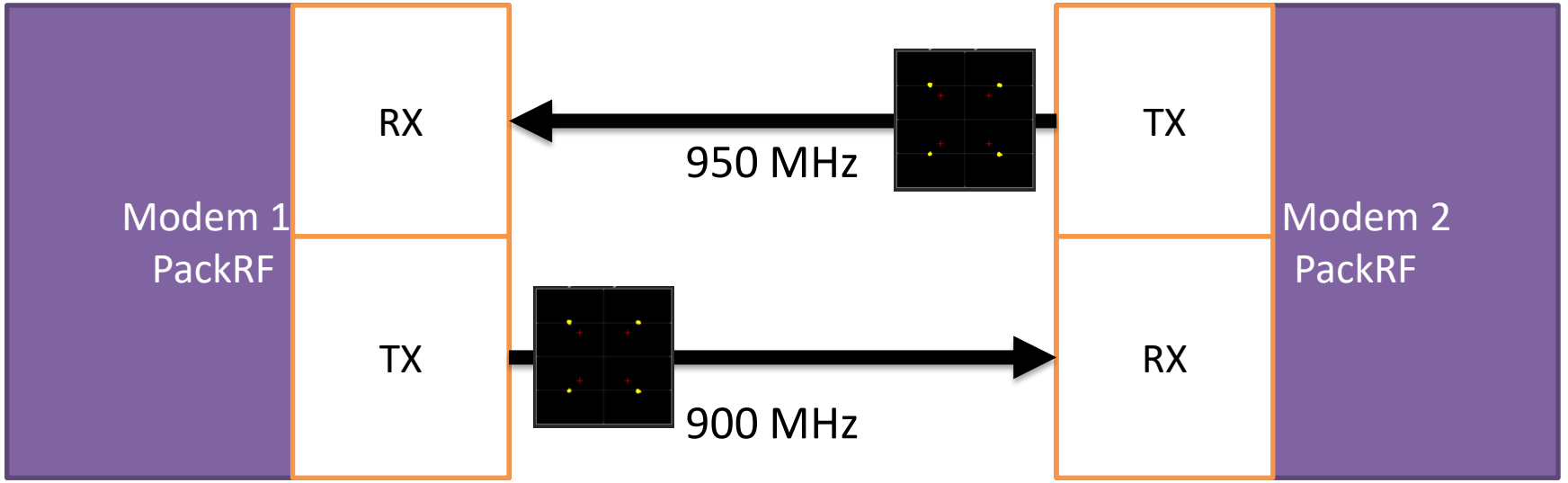
Prototype field testing, trials or bake off
Includes above plus standard peripheral access (screen, battery, GPS, PoE, Audio, etc)

Custom

Does whatever you want
Could include one or more or none of ADI: Host Libraries, GUI Software, Device Libraries, Device Drivers, HDL, Schematics, Gerber

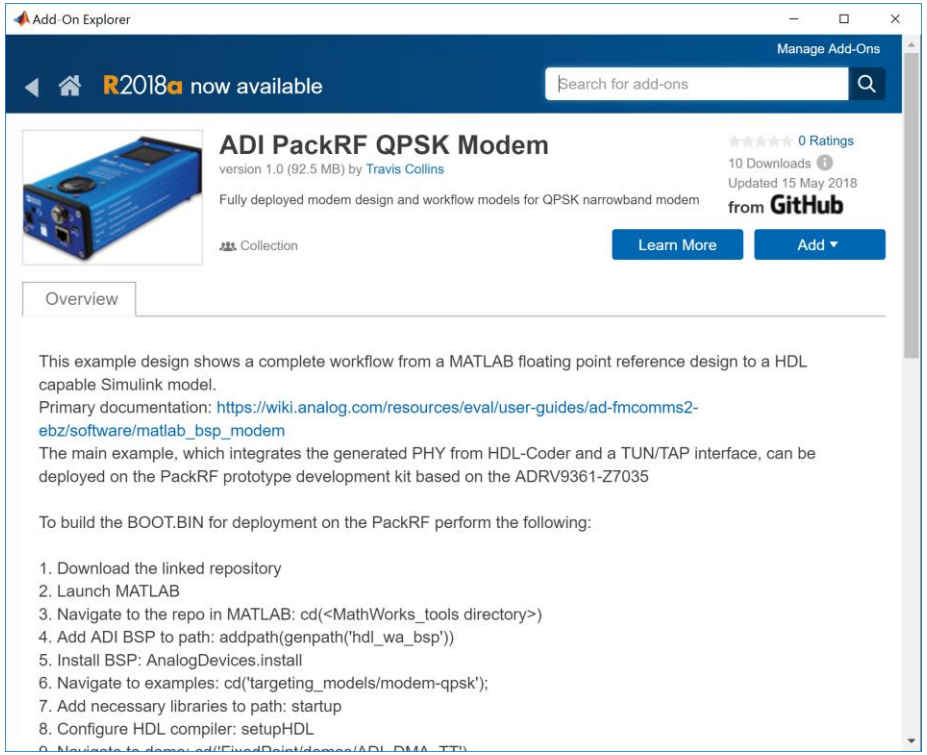
ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Example Reference Design Demonstrating Workflow



- Example design works through QPSK modem development
- Example details:
  - QPSK PHY with continuous link
  - Simple FDD system (MAC)
  - Built with common algorithms

Where can I get the code?



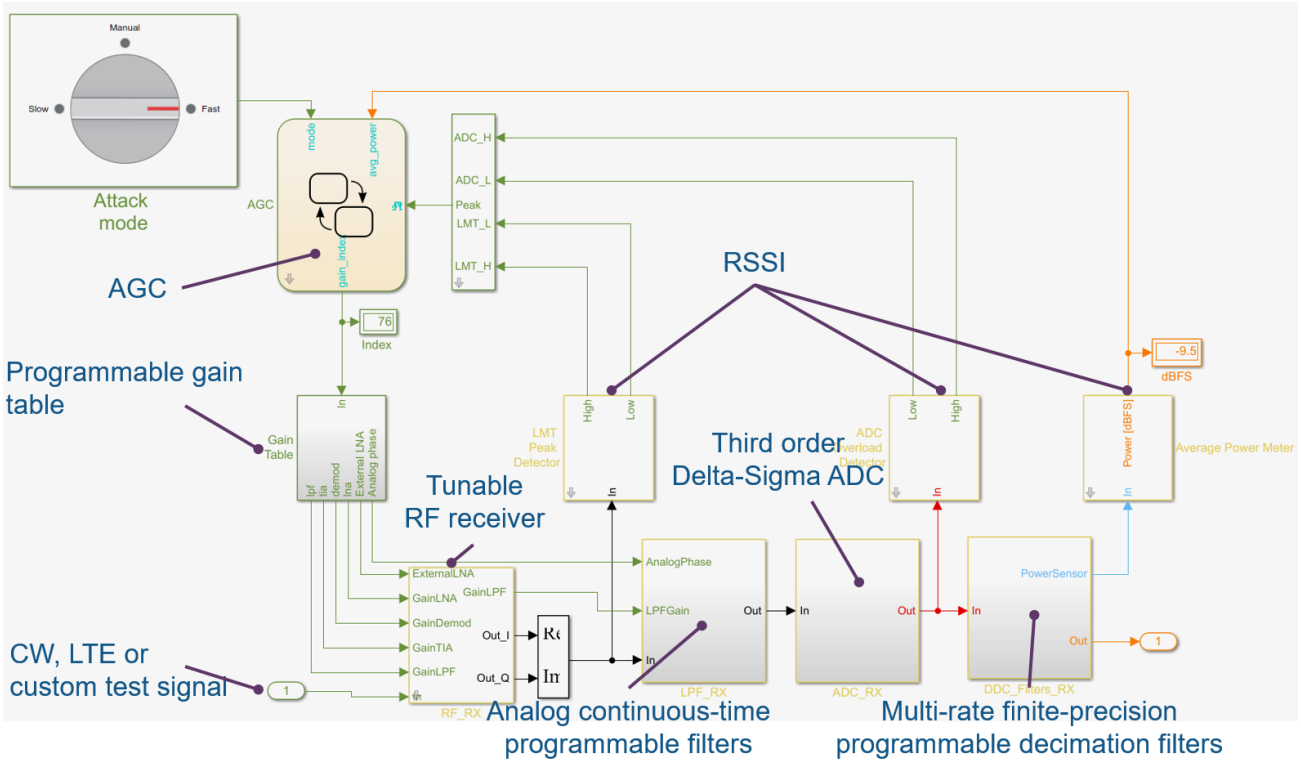
- Available in Add-On Explorer today
- 3 main reference designs
  - MATLAB Floating-Point
  - Simulink Floating-Point
  - Simulink Fixed-Point
- 5 deployable examples that show debugging techniques
  - Standard IQ
  - External Mode
  - AXI-MM
  - FPGA Capture
  - PackRF Custom BSP
- Testing harness
- Utility scripts

Workflow

A True Multi-Domain System-Level Model

- Standard and custom test signals
- Tunable RF receiver
  - Gain dependent IP2, IP3, LO leakage, I/Q imbalance
- Third order delta-sigma ADC
- Programmable analog and digital filters
- AGC described with a time-triggered state machine
- Simulates 1 LTE frame (10ms) in minutes

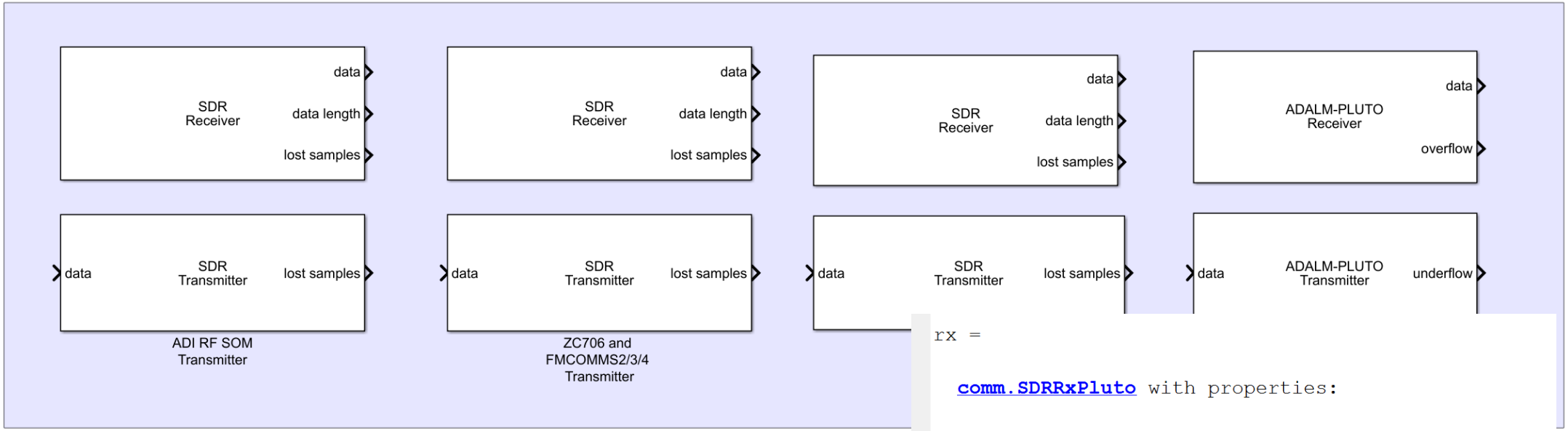
- The simulation behavior validated against actual silicon



ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

EDI CON 2018
Electronic Design Innovation Conference & Exhibition
October 17-19 2018
Santa Clara Convention Center
Santa Clara, CA

Blocks and System Objects



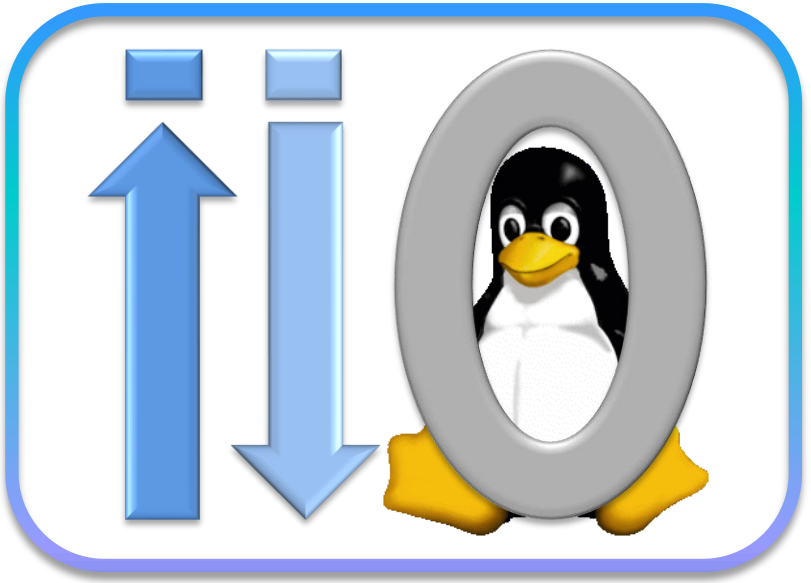

```
>> rx = sdrrx('ZC706 and FMCOMMS2/3/4');
>> rx = sdrrx('ZedBoard and FMCOMMS2/3/4');
>> rx = sdrrx('ADI RF SOM');
>> rx = sdrrx('Pluto')
```

```
rx =

  comm.SDRRxPluto with properties:

              DeviceName: 'Pluto'
                 RadioID: 'usb:0'
           CenterFrequency: 2.4000e+09
              GainSource: 'AGC Slow Attack'
          ChannelMapping: 1
        BasebandSampleRate: 1000000
            OutputDataType: 'int16'
          SamplesPerFrame: 3660
      ShowAdvancedProperties: false
```

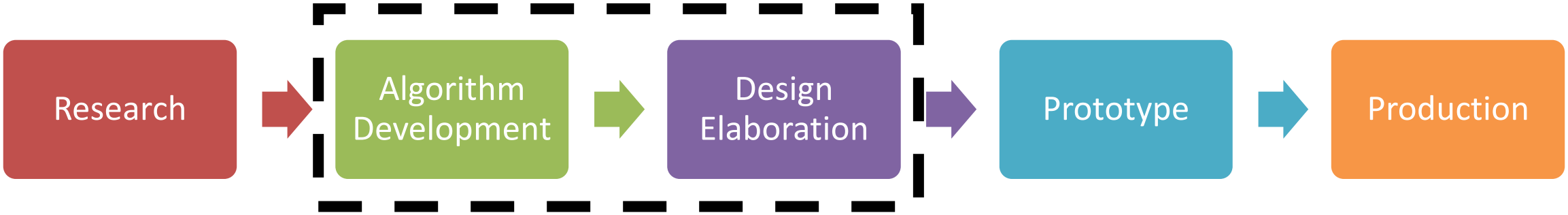
ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

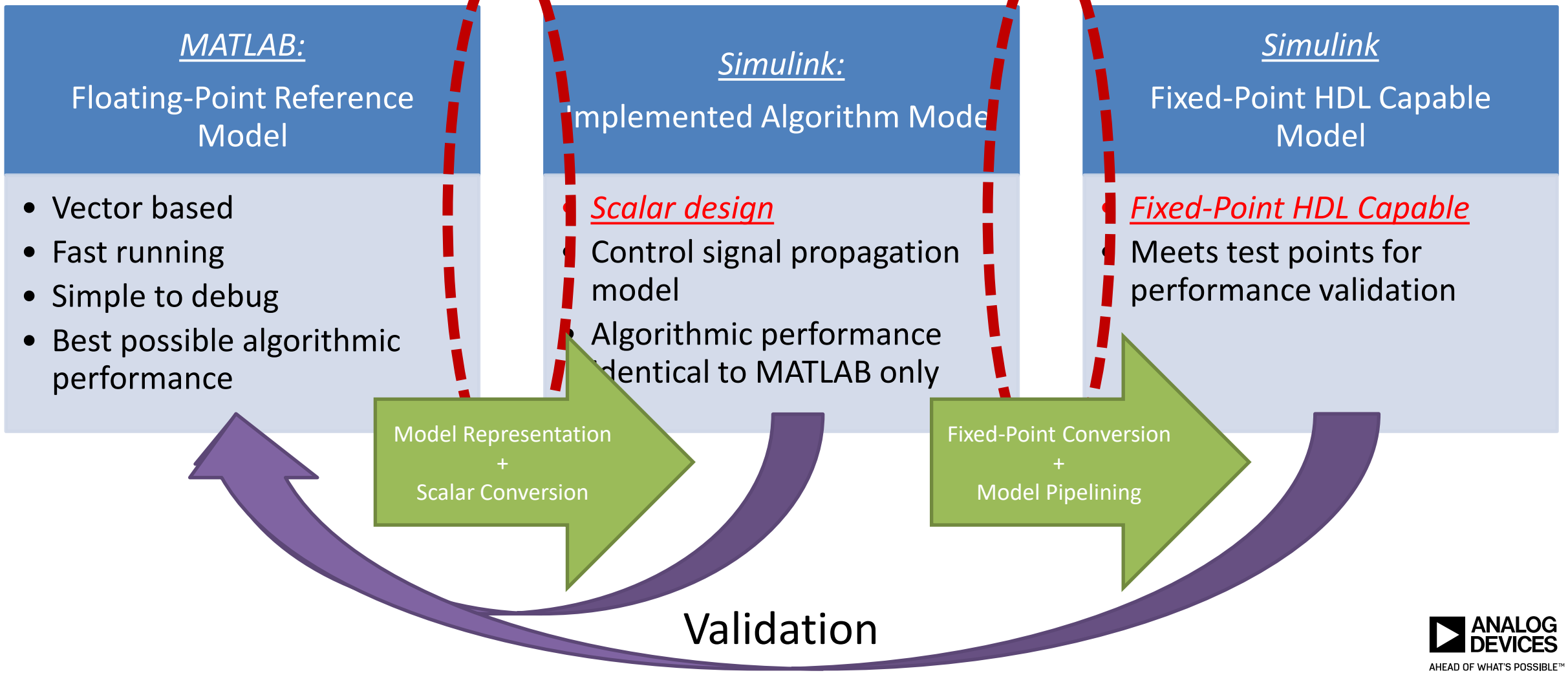
Integration with custom software – IIO, A Kernel Subsystem for Converters

- The Linux industrial I/O subsystem is intended to provide support for devices that, in some sense, are analog-to-digital or digital-to-analog converters
  - Devices that fall into this category are:
    - Precision ADCs, high-speed ADCs
    - Precision DACs, high-speed DACs
    - Accelerometers, gyroscopes, IMUs
    - Capacitance-to-Digital converters (CDCs)
    - Pressure, proximity, temperature and light sensors
    - Health, chemical, magnetometer, amplifiers, etc.
  - Can be used on ADCs ranging from a SoC ADC to >1000 MSPS
  - Mostly focused on user-space abstraction, but also in-kernel API for other drivers exists
    - IIO to Linux input or HWMON subsystem bridges



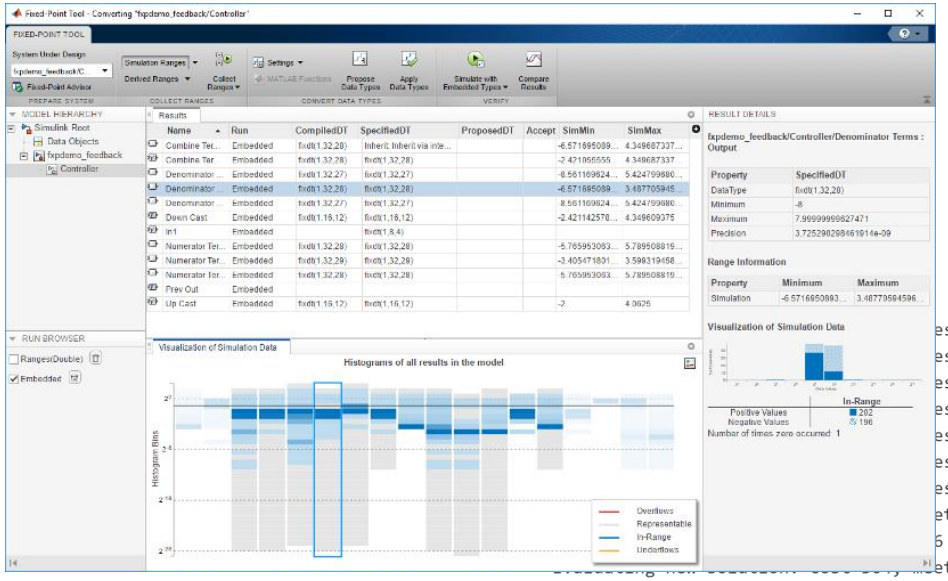
Workflow



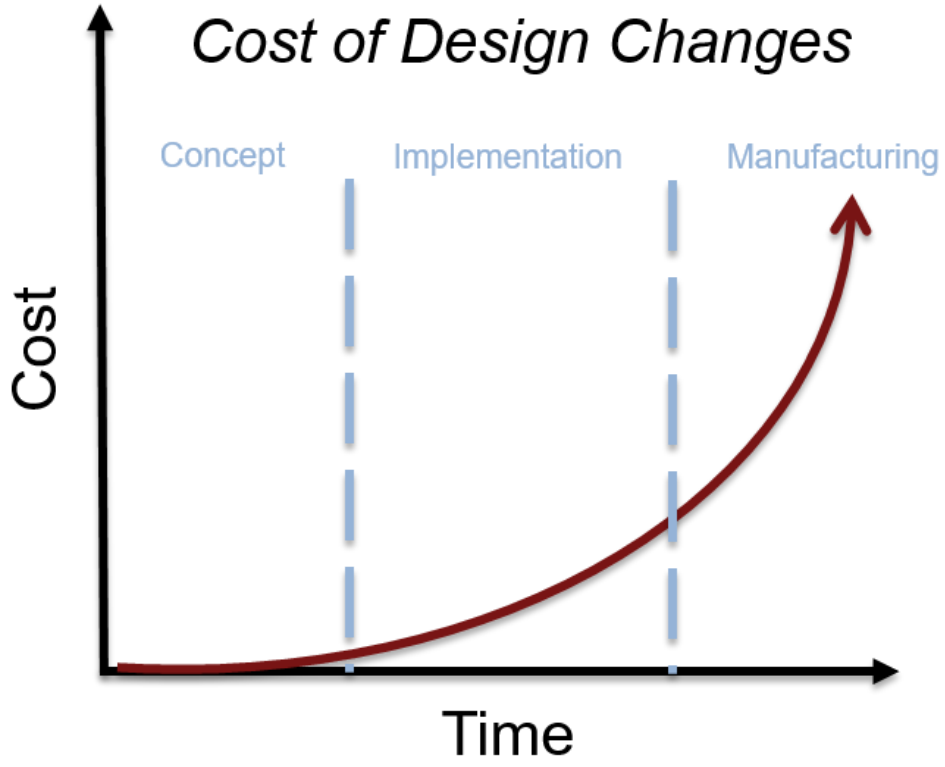

MATLAB:
Floating-Point Reference Model
Vector based
Fast running
Simple to debug
Best possible algorithmic performance
Simulink:
Implemented Algorithm Model
Scalar design
Control signal propagation model
Algorithmic performance identical to MATLAB only
Simulink
Fixed-Point HDL Capable Model
Fixed-Point HDL Capable
Meets test points for performance validation
Model Representation + Scalar Conversion
Fixed-Point Conversion + Model Pipelining
Validation

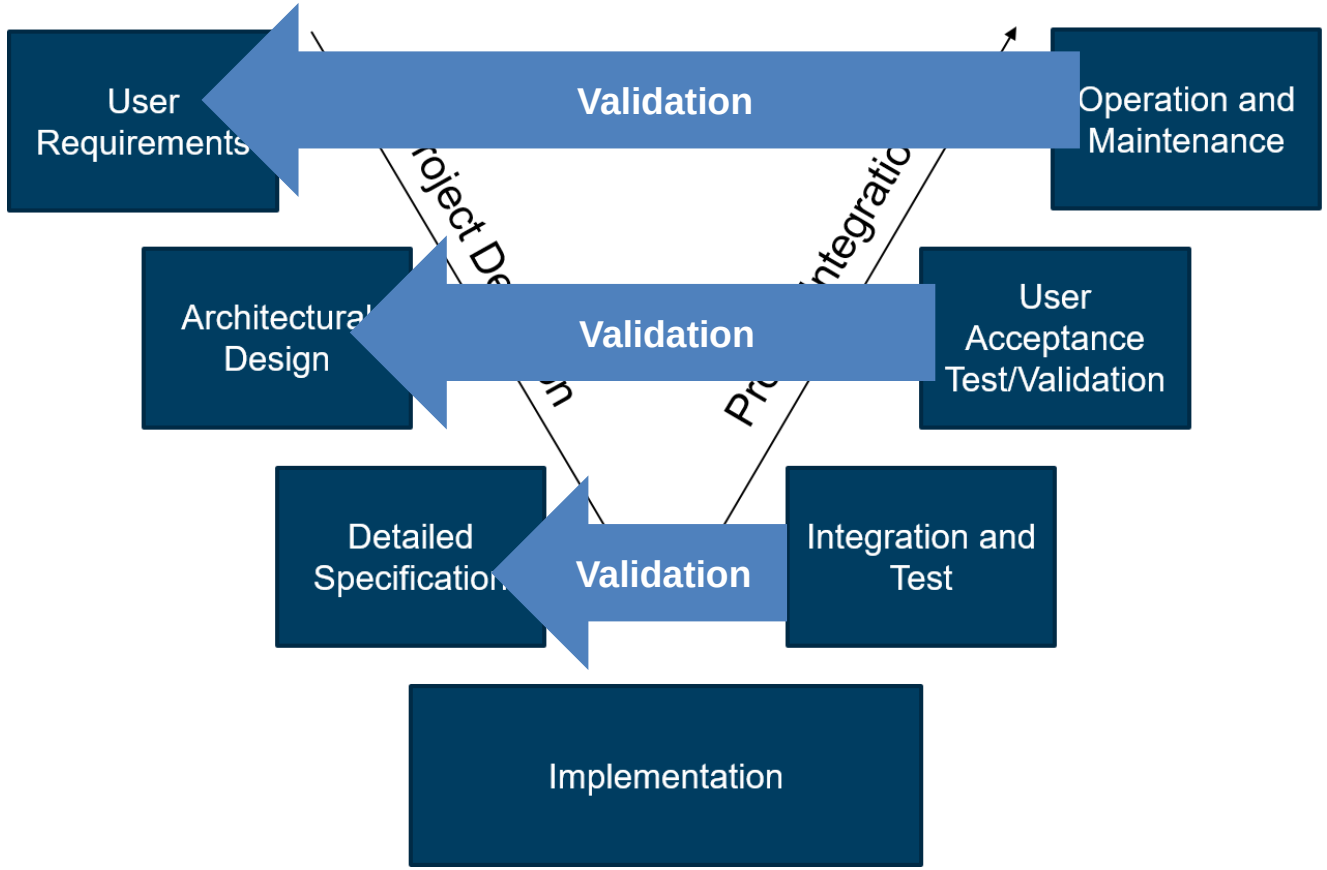
Fixed-Point Conversion

- Fixed-Point conversion tools
  - Fixed-Point Designer
  - Fixed-Point Tool
  - Toolbox mirrored functionality between MATLAB and Simulink
  - MATLAB Function blocks
  - Iterative optimization from high level constraints



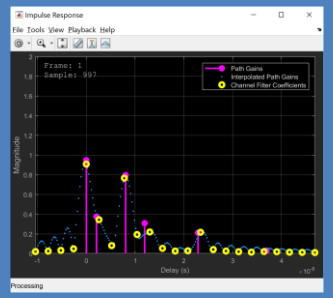

```
es not meet the tolerances.
es not meet the tolerances.
es not meet the tolerances.
es not meet the tolerances.
es not meet the tolerances.
es not meet the tolerances.
es not meet the tolerances.
ets the tolerances.
6
ets the tolerances.
- Updated best found solution, cost: 304
- Evaluating new solution: cost 300, meets the tolerances.
- Updated best found solution, cost: 300
- Evaluating new solution: cost 299, meets the tolerances.
- Updated best found solution, cost: 299
- Evaluating new solution: cost 298, meets the tolerances.
- Updated best found solution, cost: 298
```

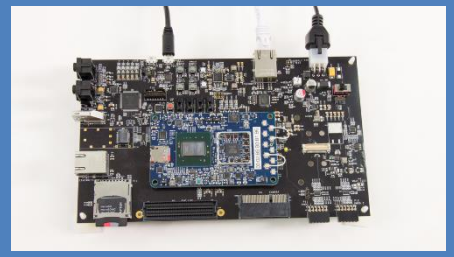

Cost of Design Changes
Concept
Implementation
Manufacturing
Cost
Time


User Requirements
Validation
Operation and Maintenance
Architectural Design
Validation
User Acceptance Test/Validation
Detailed Specification
Validation
Integration and Test
Implementation

Waveform Generation

Channel Models *Hardware Interfaces*





Receiver IP

MATLAB

```
    % Descramble
    rxDescram = descr(rxData);
    % Viterbi decode the demodulated data
    dataHard = vitdec(rxDescram,trellis,tbl,'cont','hard');
    % Removing coding delay
    rxDataWithTail = dataHard(tbl+1:end);
    % Remove tail bits
    rxDataWithCRC = rxDataWithTail(1:end-nTail);
    % Check CRC
    [rxData,e] = crcDec(rxDataWithCRC);
    if e
        log(testCase,2,'CRC Failed.');
        crcChecks = [crcChecks,1];
        failures = [failures;4];
    else
        log(testCase,2,'CRC Passed.');
        crcChecks = [crcChecks,0];
        failures = [failures;0]; %#ok<*AGROW>
        packetBits = {packetBits;rxData};
    end

end

log(testCase,4,'Receiver done processing data.');
% Pack results
results = struct('packetsFound',packetsFound,...
    'crcChecks',crcChecks,'failures',failures);
```

Similar features apply to Simulink models

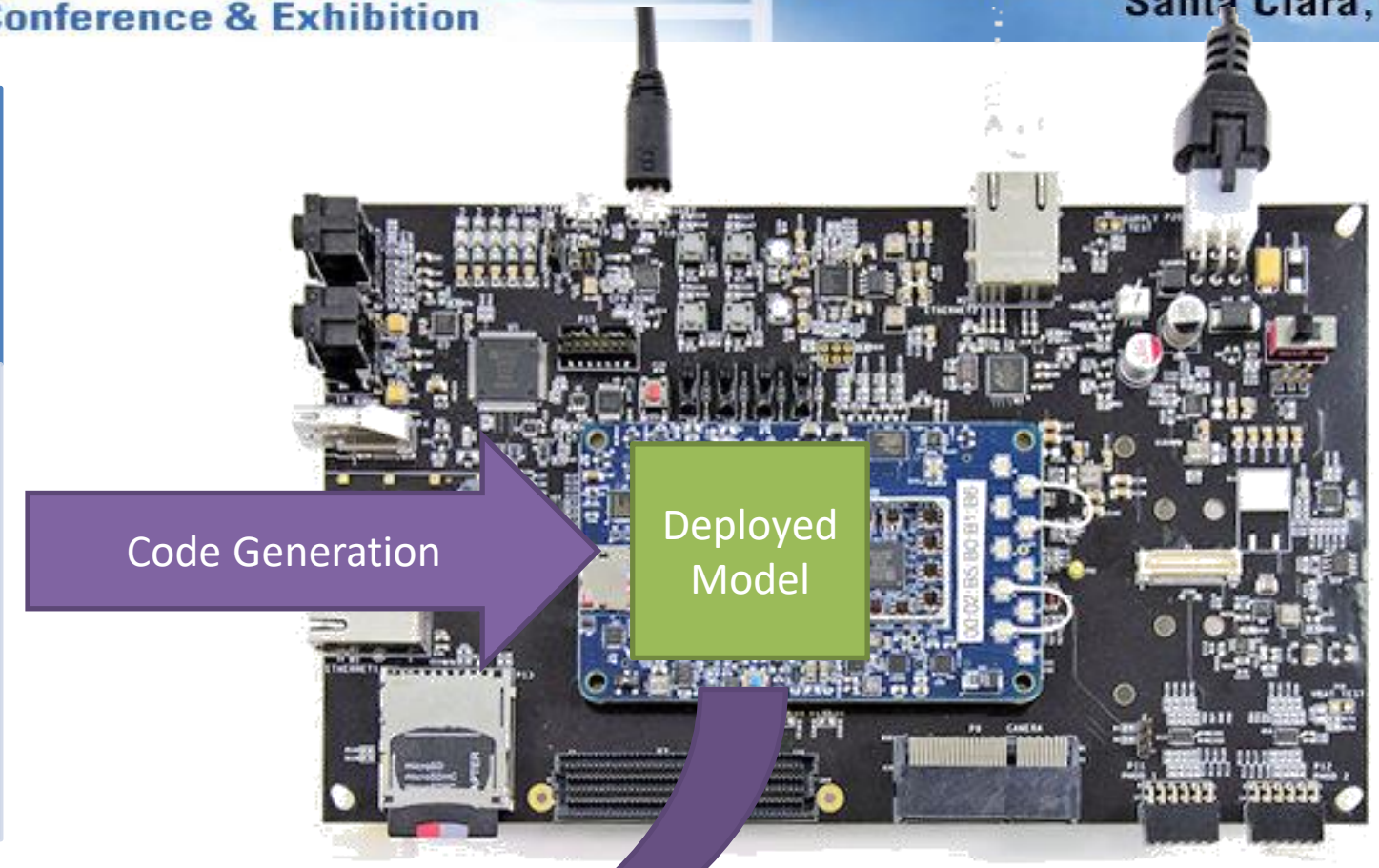
ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Workflow

Simulink

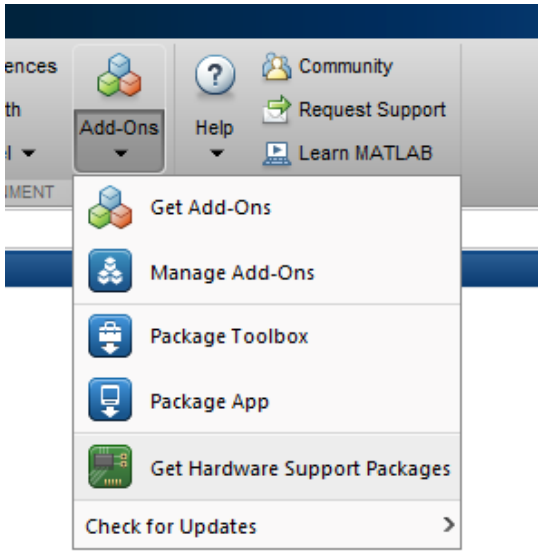
Fixed-Point HDL Capable Model

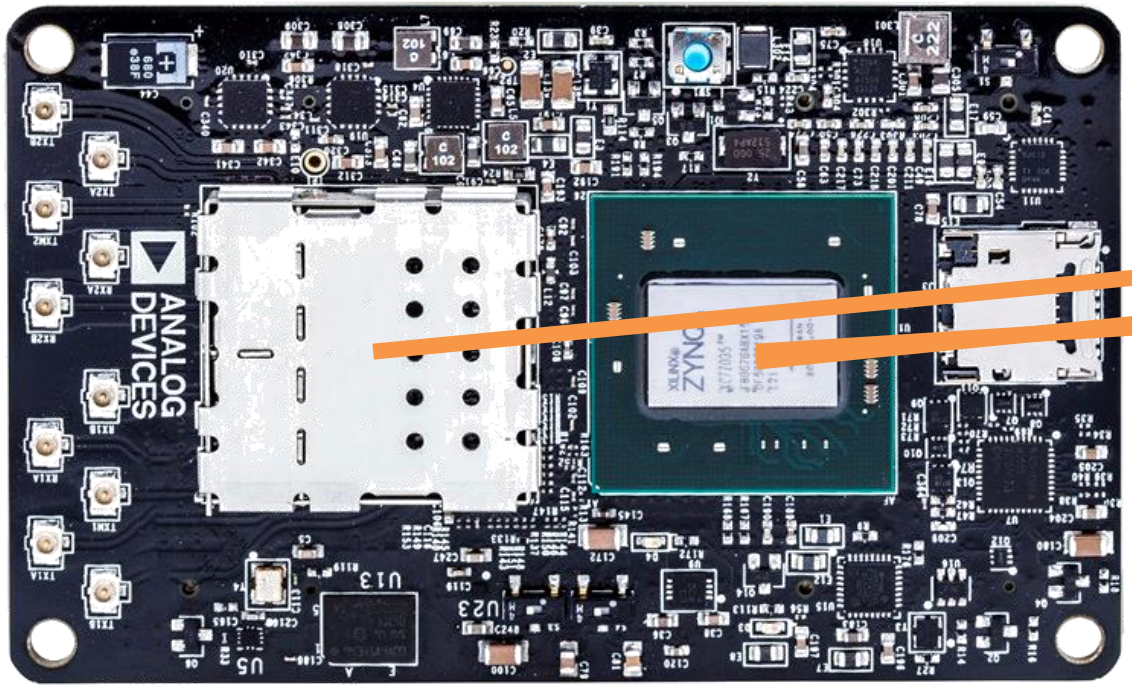
- *Fixed-Point HDL Capable*
- Meets test points for performance validation



Support Packages

- The bridge between MATLAB and Simulink and Hardware
- Enable radio I/O, prototyping, and production deployment
- **Hardware Support Packages** are available via the add-on explorer in MATLAB
  - Provide board-specific **reference designs** for C and HDL code generation
  - Provide portable Linux drivers for data I/O
- Third-party-authored **reference designs** enable custom hardware targeting
  - Leverages published APIs





Hardware:

Transceiver

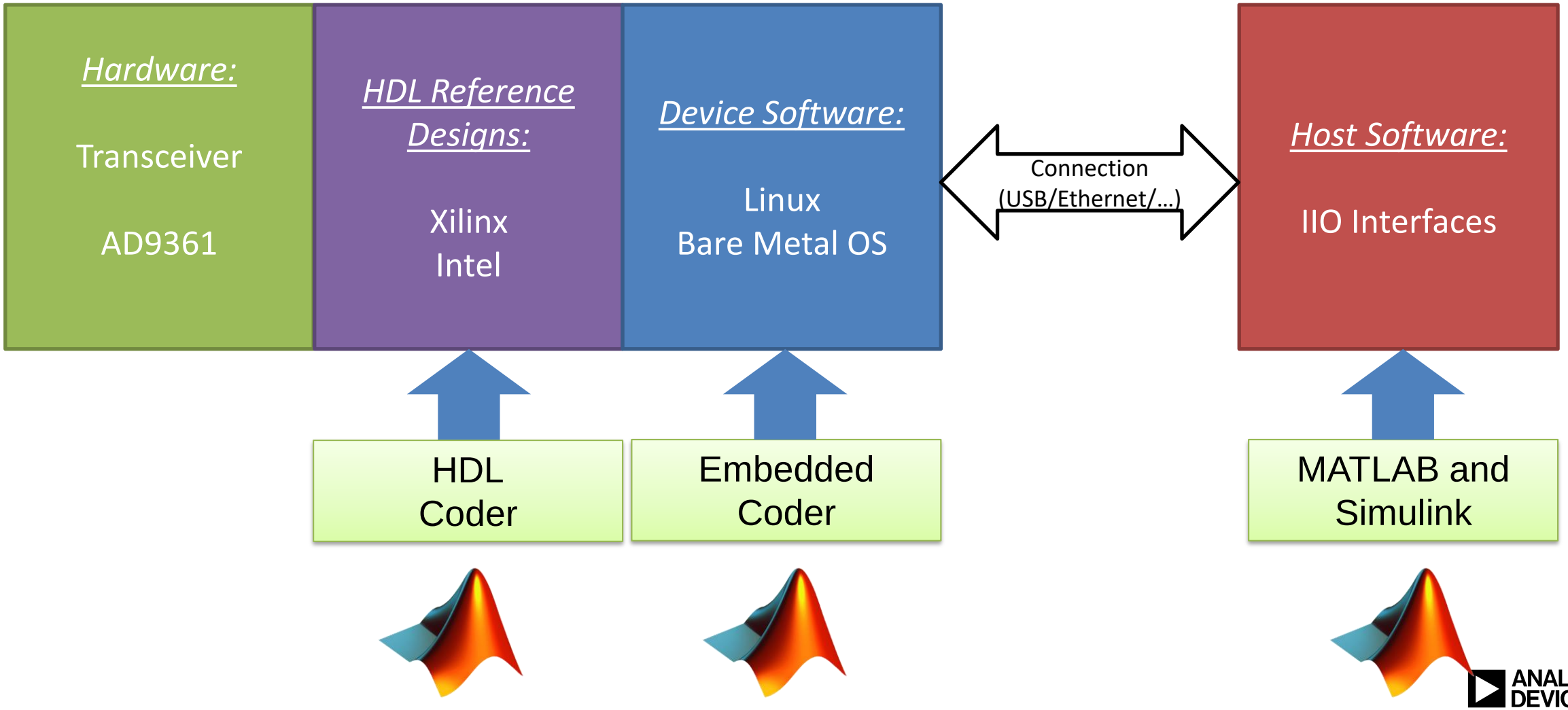
AD9361

HDL Reference Designs:

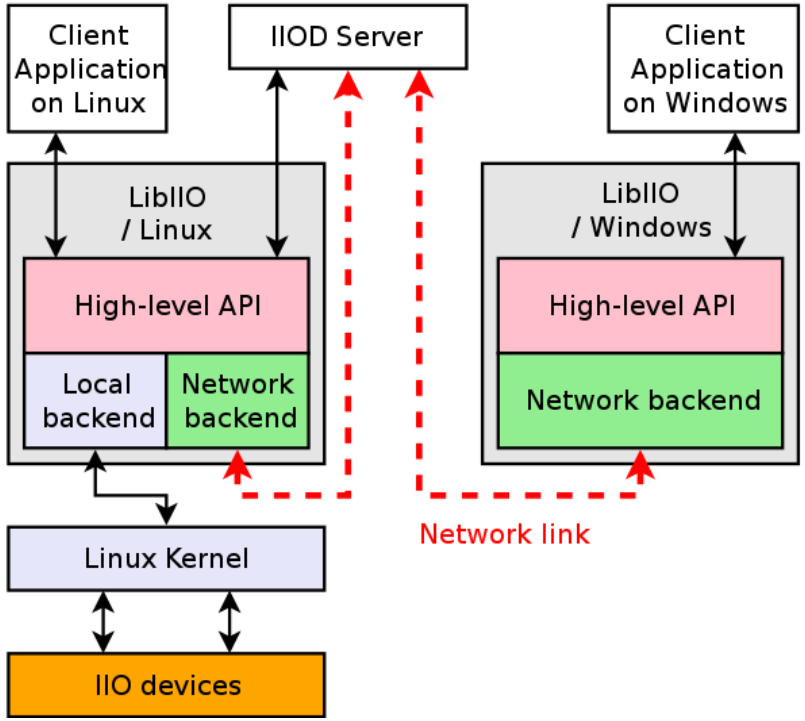
Xilinx
Intel

Device Software:

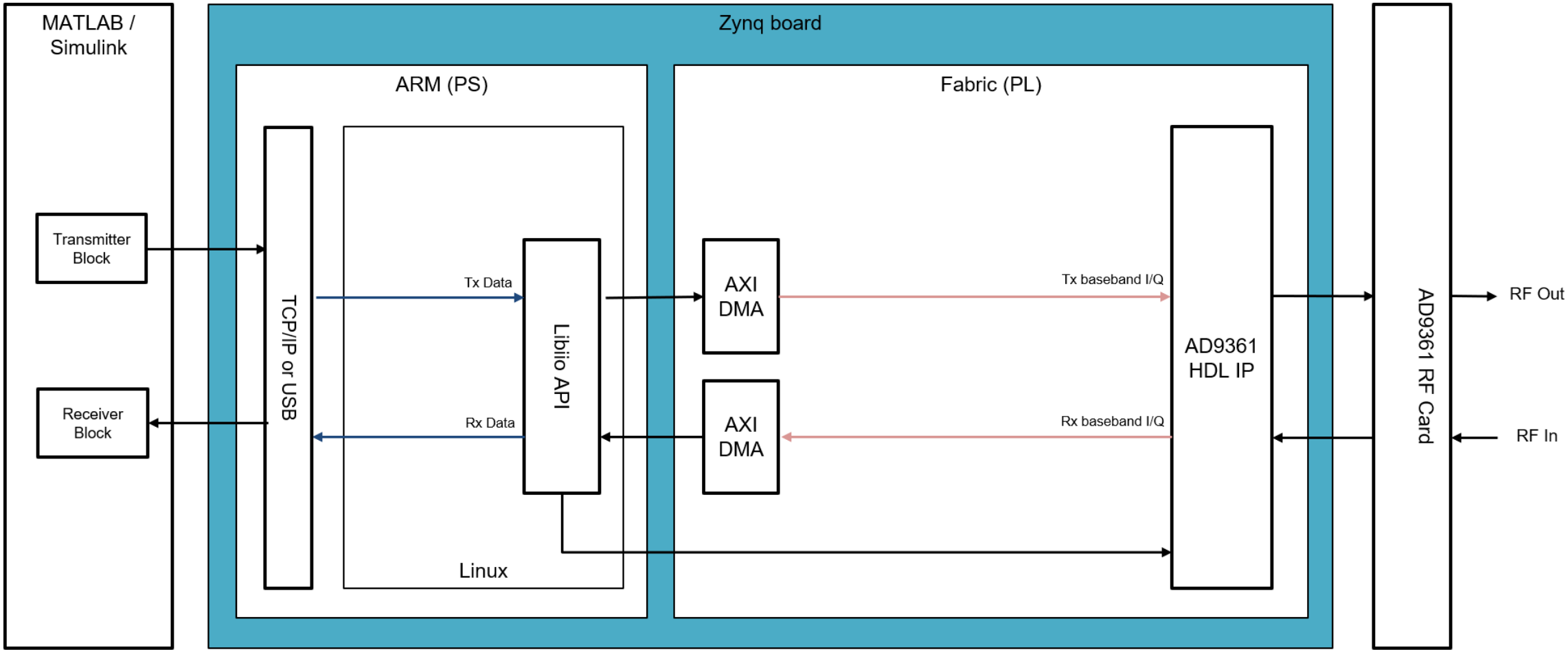
Linux
Bare Metal OS

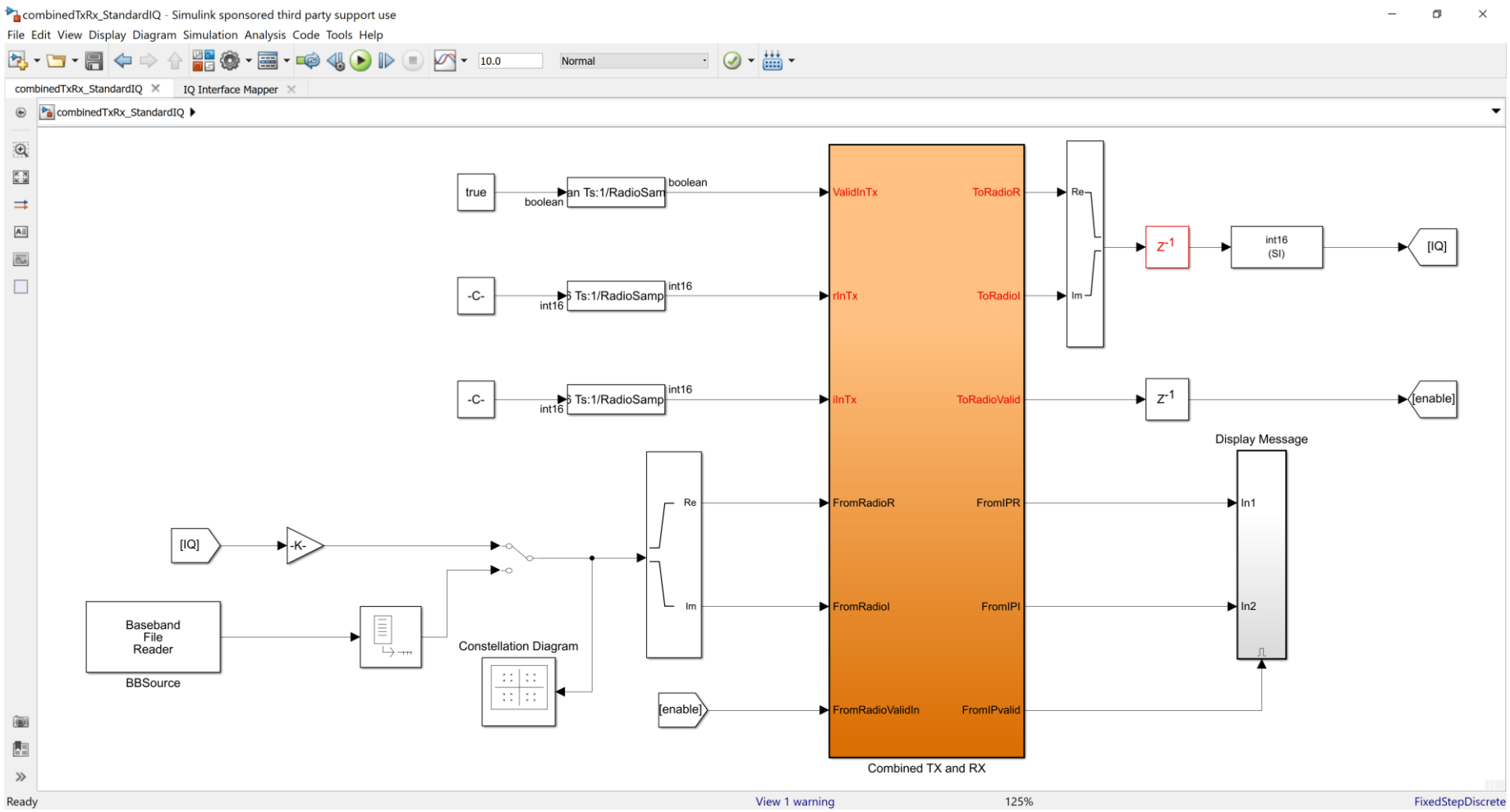

Hardware:
Transceiver
AD9361
HDL Reference Designs:
Xilinx
Intel
Device Software:
Linux
Bare Metal OS
Connection
(USB/Ethernet/...)
Host Software:
IIO Interfaces
HDL Coder
Embedded Coder
MATLAB and Simulink

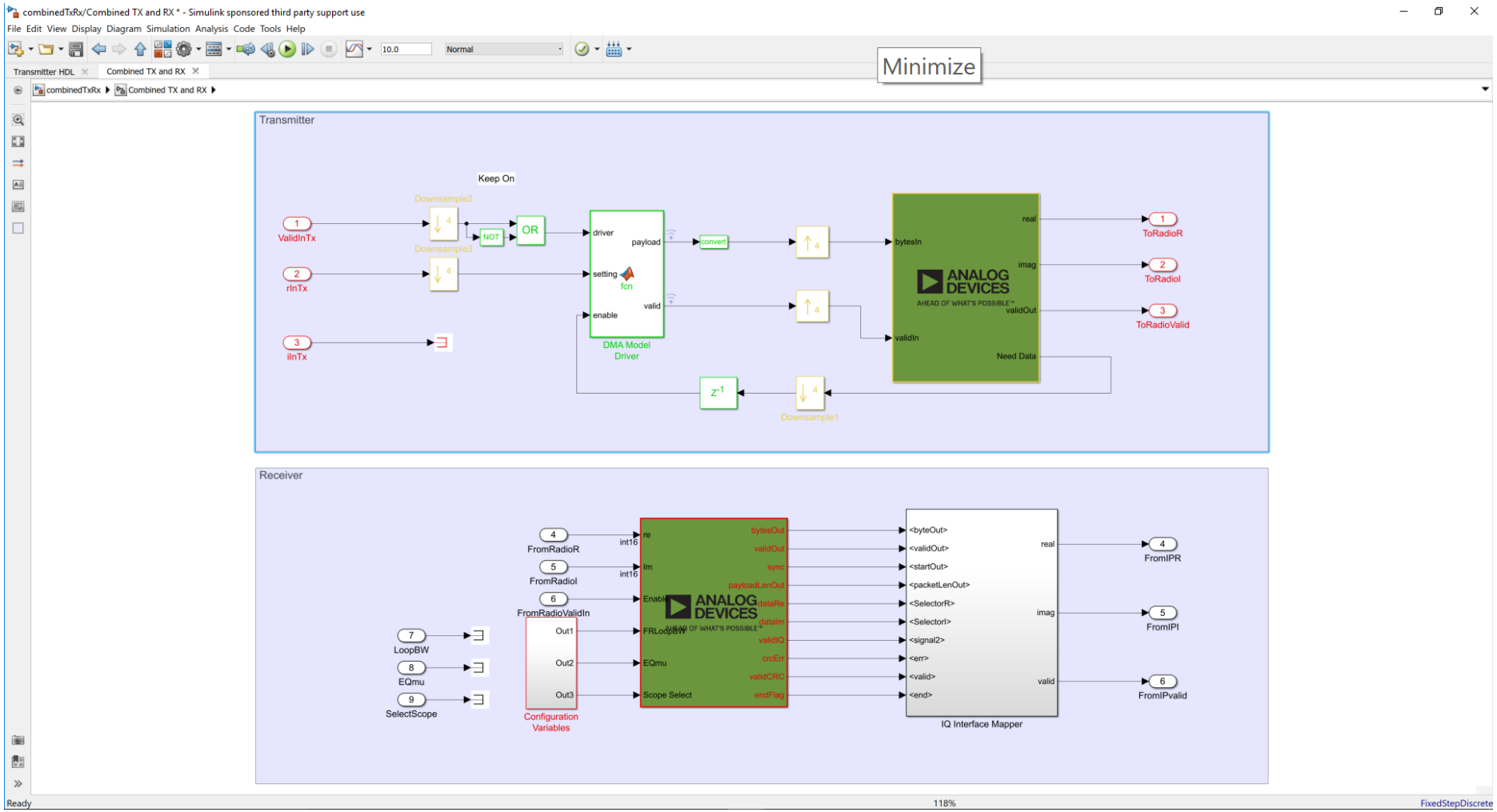
libIIO



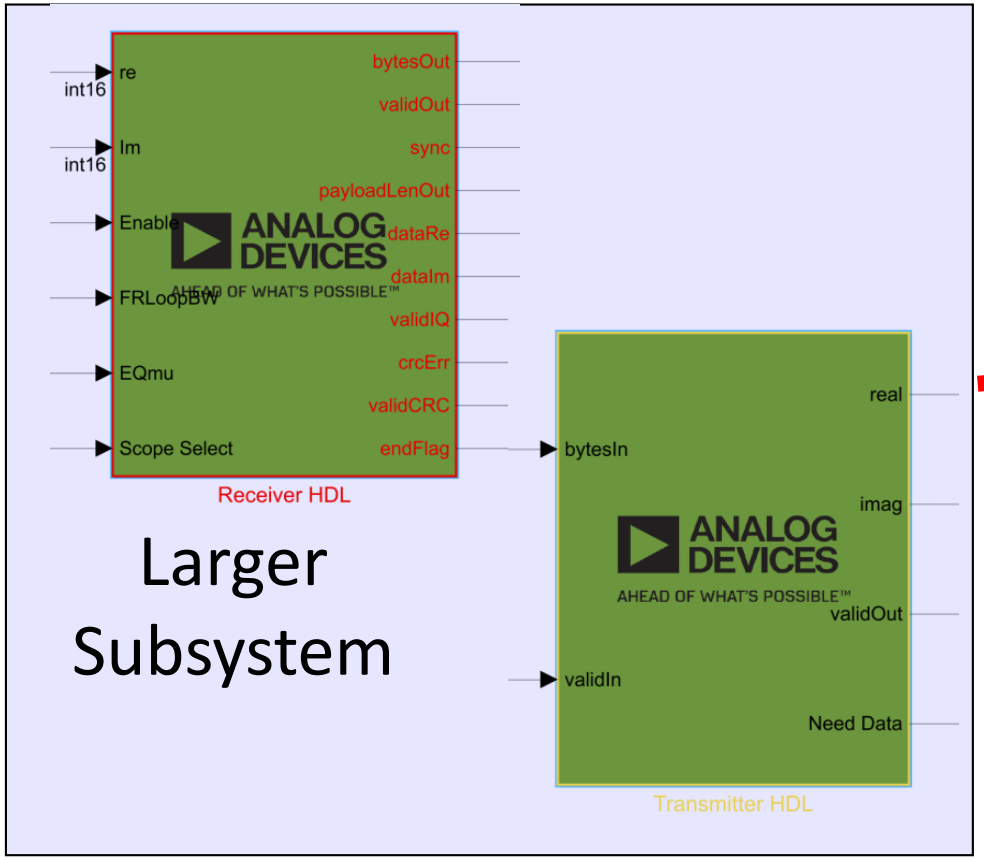

```
struct iio_context *ctx;
struct iio_device *dev;
struct iio_channel *ch;

/* Error handling is missing */
ctx = iio_create_default_context();
dev = iio_context_get_device(ctx, 0);
ch = iio_device_get_channel(dev, 0);

iio_device_attr_write_longlong(dev, “sample_rate”, 1000);
iio_channel_attr_write_double(ch, “scale”, 0.525);
```


MATLAB / Simulink
Transmitter Block
Receiver Block
Zynq board
ARM (PS)
TCP/IP or USB
Tx Data
Rx Data
Libiio API
Linux
Fabric (PL)
AXI DMA
AXI DMA
Tx baseband I/Q
Rx baseband I/Q
AD9361 HDL IP
AD9361 RF Card
RF Out
RF In

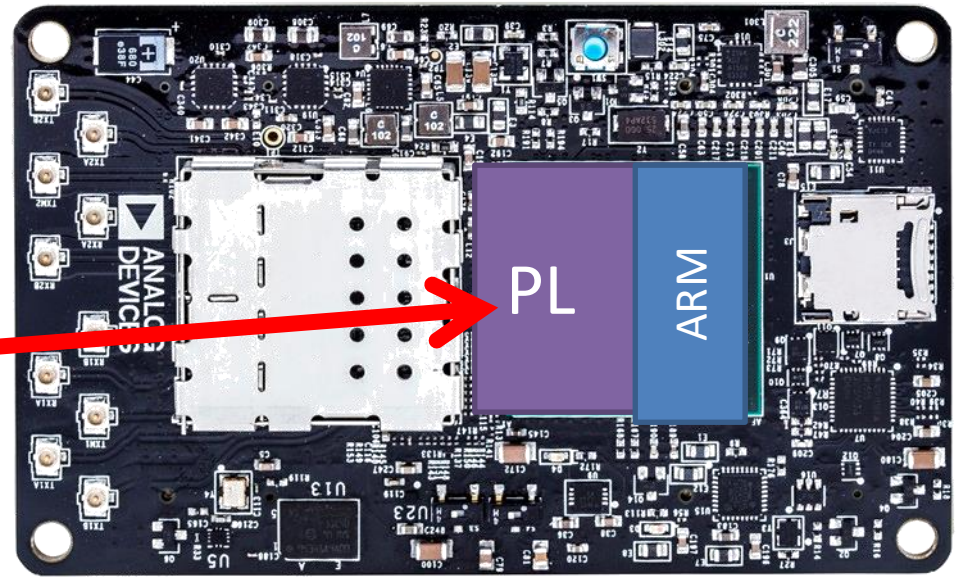

combinedTxRx_StandardIQ - Simulink sponsored third party support use
File Edit View Display Diagram Simulation Analysis Code Tools Help
combinedTxRx_StandardIQ
IQ Interface Mapper
true
boolean
ValidInTx
ToRadioR
Re
Z-1
int16 (SI)
[IQ]
-C-
int16
rInTx
ToRadioI
Im
iInTx
ToRadioValid
[enable]
Display Message
FromRadioR
FromIPR
In1
FromRadioI
FromIPI
In2
-K-
Baseband File Reader
BBSource
Constellation Diagram
FromRadioValidIn
FromIPvalid
Combined TX and RX
Ready
View 1 warning
125%
FixedStepDiscrete



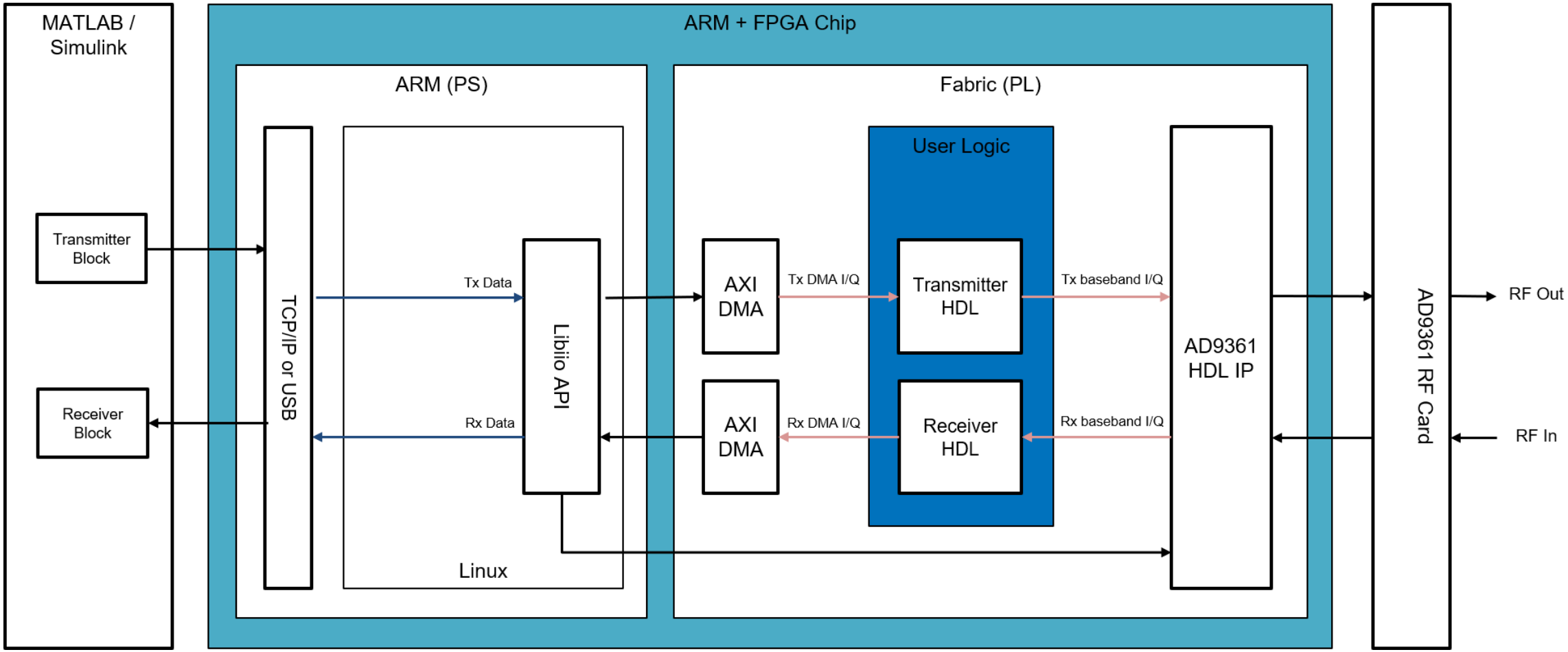
ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™



Larger Subsystem



Inserted into ADI HDL reference design


MATLAB / Simulink
Transmitter Block
Receiver Block
ARM + FPGA Chip
ARM (PS)
TCP/IP or USB
Tx Data
Rx Data
Libiio API
Linux
Fabric (PL)
User Logic
AXI DMA
Tx DMA I/Q
Transmitter HDL
Tx baseband I/Q
AXI DMA
Rx DMA I/Q
Receiver HDL
Rx baseband I/Q
AD9361 HDL IP
AD9361 RF Card
RF Out
RF In

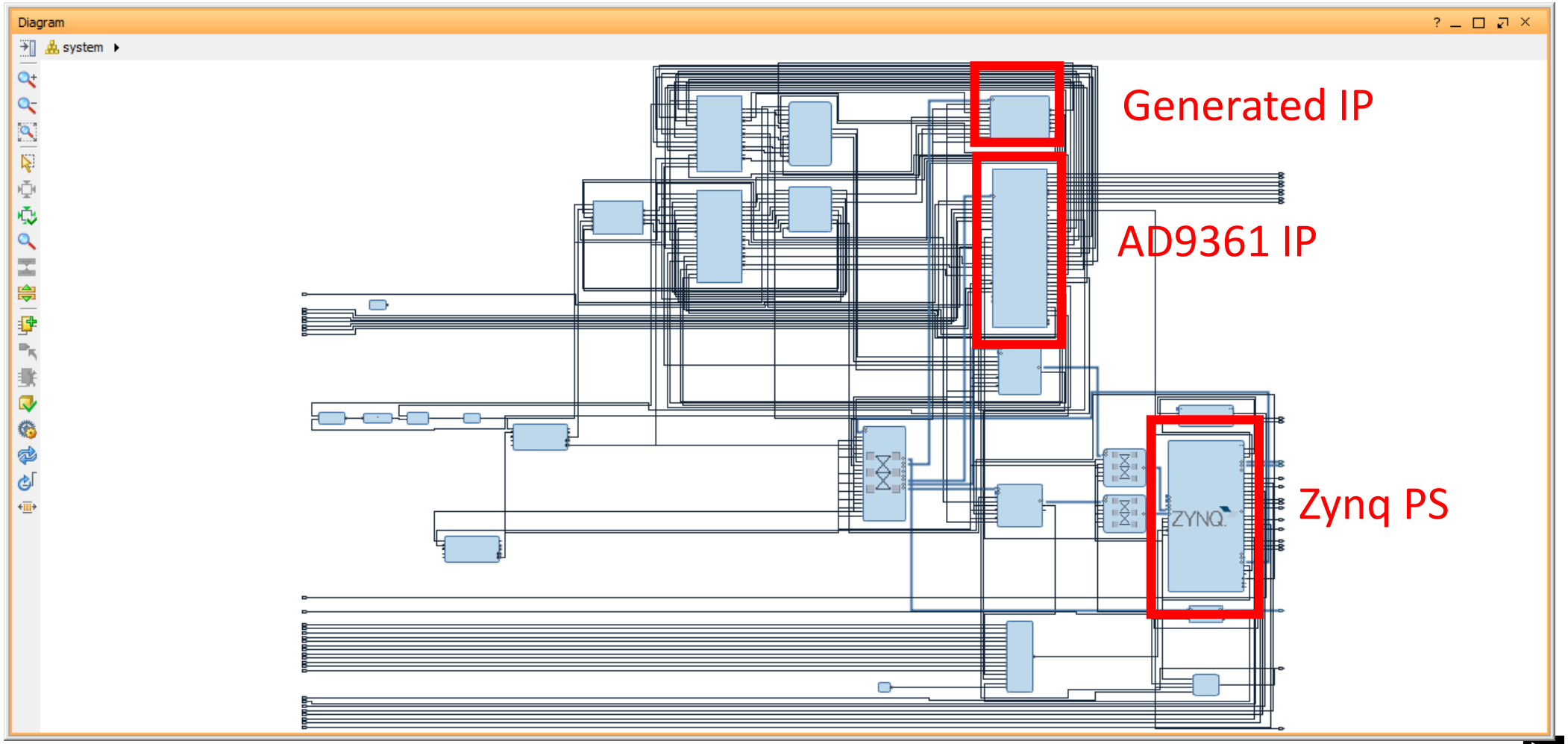
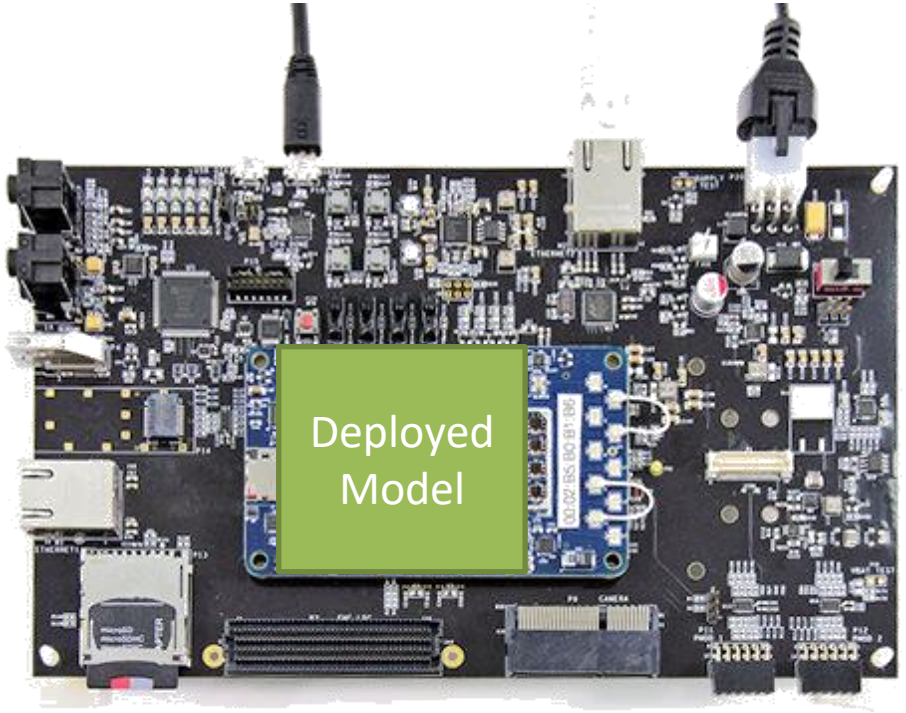
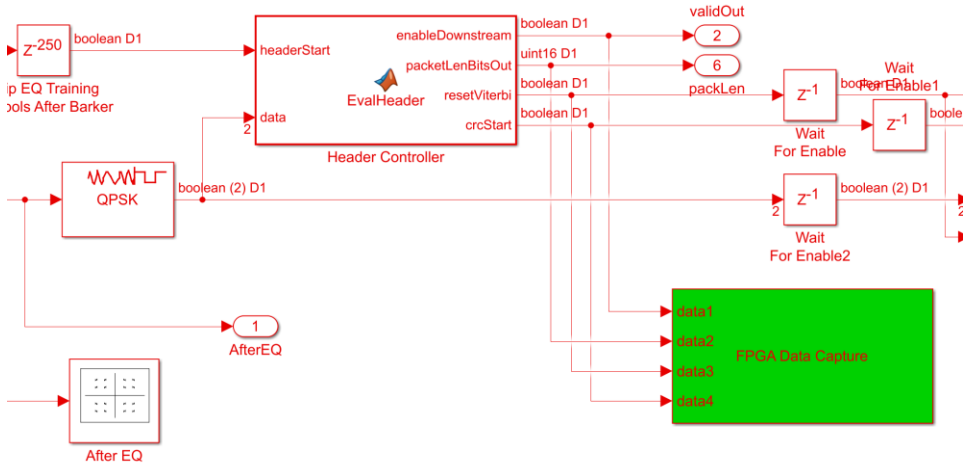


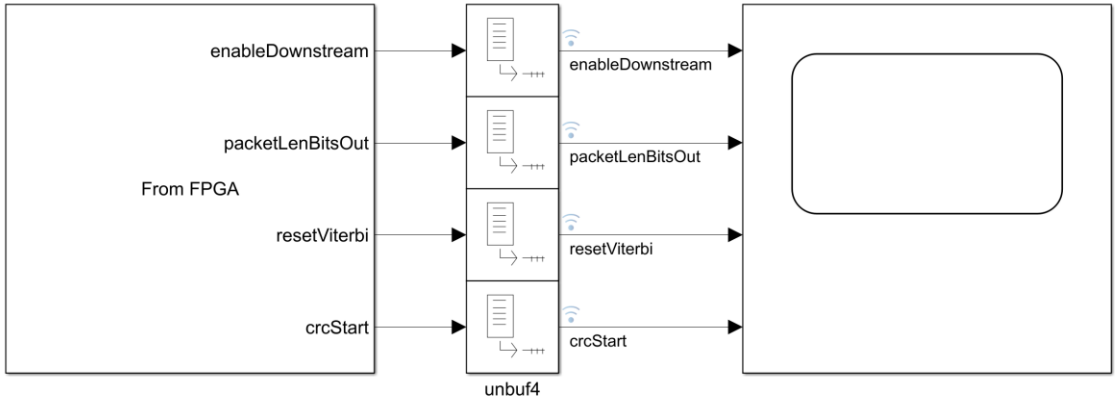
Diagram
system
Generated IP
AD9361 IP
Zynq PS
ZYNQ

Deployed Debugging Options

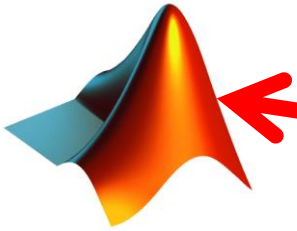
- Standard streaming from base reference design:
  - These will always exists and always must be mapped
  - Can handle high speed data
- External Mode:
  - Utilize for low-speed analysis and tuning
- IIO AXI-MM:
  - Built on top of IIO infrastructure
  - Works at high speed without Embedded Coder
- FPGA Capture:
  - Timing diagram debugging
  - Very useful for debugging IP integrations

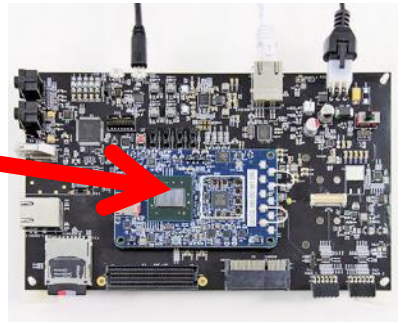







```
%% Writers
% Frequency Recovery Loop Bandwidth
w1 = matlabshared.libiio.aximm.write('uri',radioIP);
w1.AddressOffset = hex2dec('100');
w1.HardwareDataType='int16';
% Equalizer Step Size
w2 = matlabshared.libiio.aximm.write('uri',radioIP);
w2.AddressOffset = hex2dec('104');
w2.HardwareDataType='int16';
% IQ Scope Selection
w3 = matlabshared.libiio.aximm.write('uri',radioIP);
w3.AddressOffset = hex2dec('108');
w3.HardwareDataType='int8';
% Debug Status Signal Selection
w4 = matlabshared.libiio.aximm.write('uri',radioIP);
w4.AddressOffset = hex2dec('10C');
w4.HardwareDataType='int8';
```

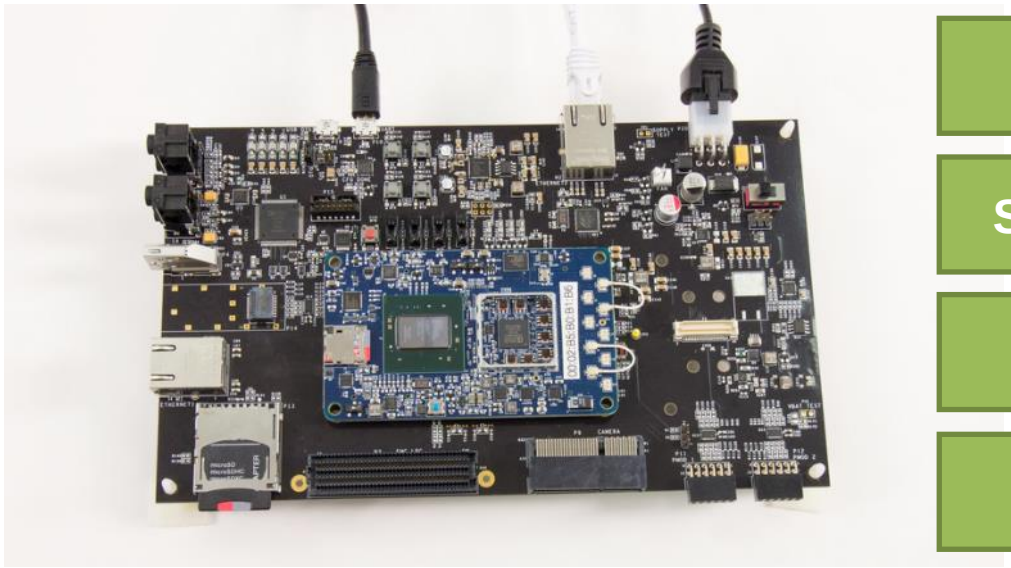




ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

Targeting Custom Hardware

Workflow



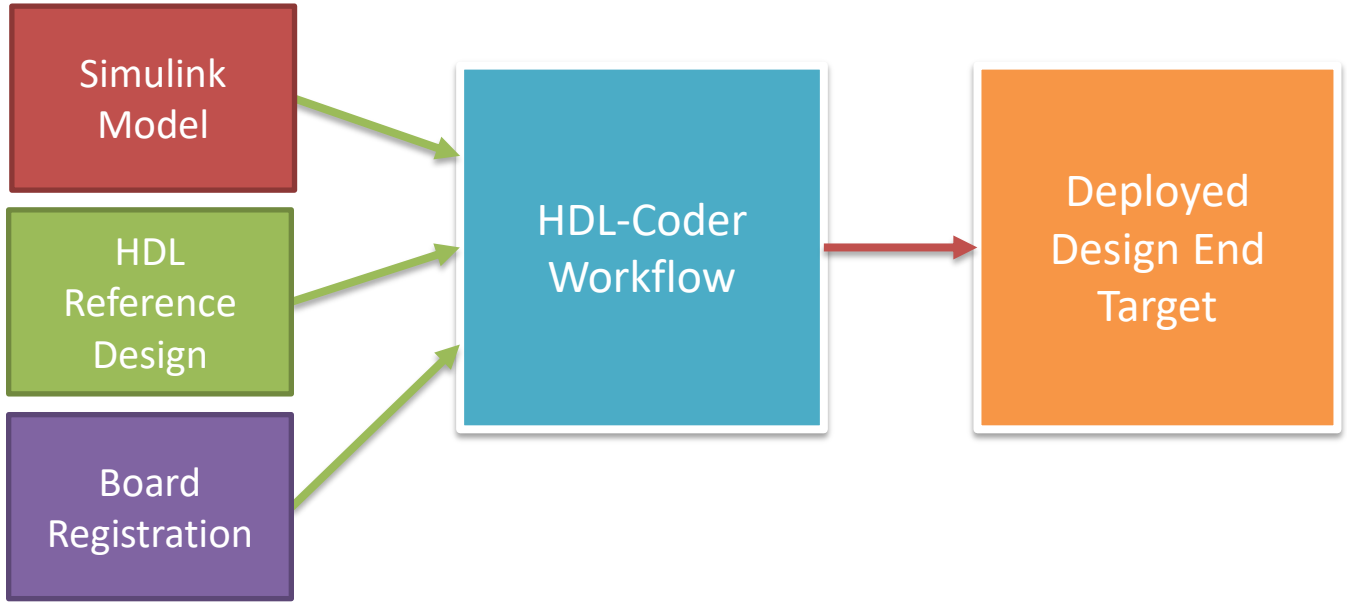
Algorithm Unchanged

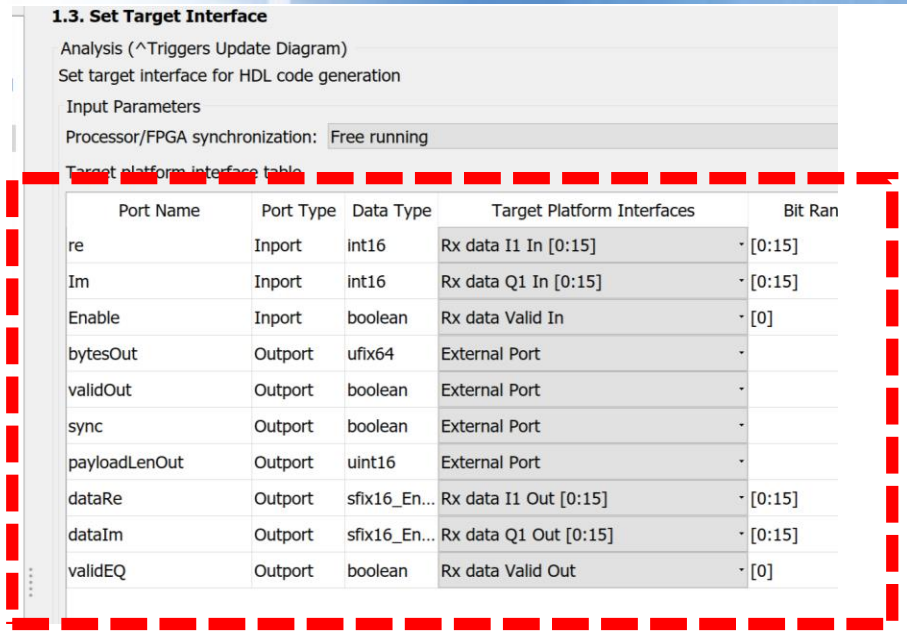
Same RF-SOM

New Carrier Board

Same Connectivity

- Path from supported development hardware to production hardware
- Documented process from MathWorks called ***Board Support Packages***
  - ADI maintains examples for different board variants
- Provides same connectivity in final production hardware back to MATLAB




1.3. Set Target Interface

Analysis (^Triggers Update Diagram)

Set target interface for HDL code generation

Input Parameters

Processor/FPGA synchronization: Free running

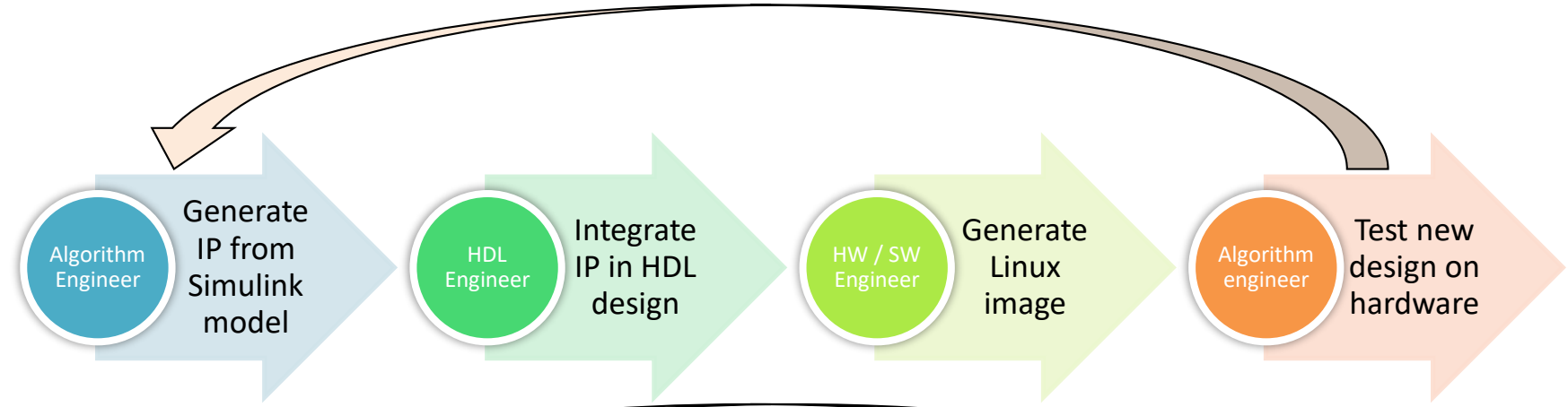
Target platform interface table

Port Name	Port Type	Data Type	Target Platform Interfaces	Bit Ran
re	Inport	int16	Rx data I1 In [0:15]	[0:15]
Im	Inport	int16	Rx data Q1 In [0:15]	[0:15]
Enable	Inport	boolean	Rx data Valid In	[0]
bytesOut	Outport	ufix64	External Port	
validOut	Outport	boolean	External Port	
sync	Outport	boolean	External Port	
payloadLenOut	Outport	uint16	External Port	
dataRe	Outport	sfix16_En...	Rx data I1 Out [0:15]	[0:15]
dataIm	Outport	sfix16_En...	Rx data Q1 Out [0:15]	[0:15]
validEQ	Outport	boolean	Rx data Valid Out	[0]

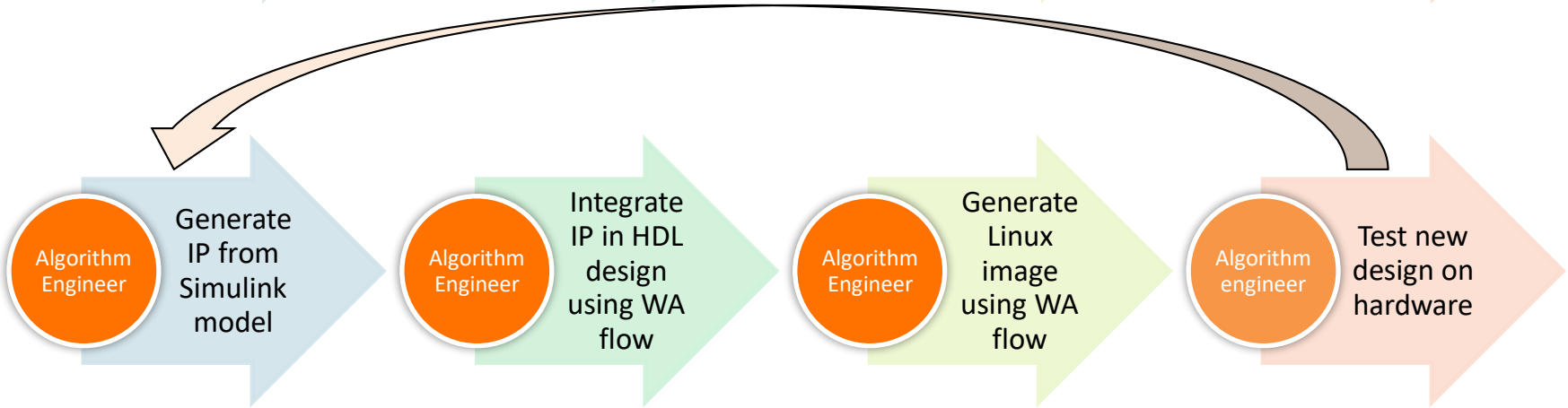
- Hardware Support Package (HSP)
  - Standard development kits
  - Fixed reference designs
  - Fixed board registration
  - End targets:
    - MATLAB/Simulink
    - ARM codegen application

- Board Support Package (BSP)
  - Custom boards (ex: PackRF board)
  - Custom reference designs
  - Custom registration API
  - End target:
    - Up to user
    - TUN/TAP in our design

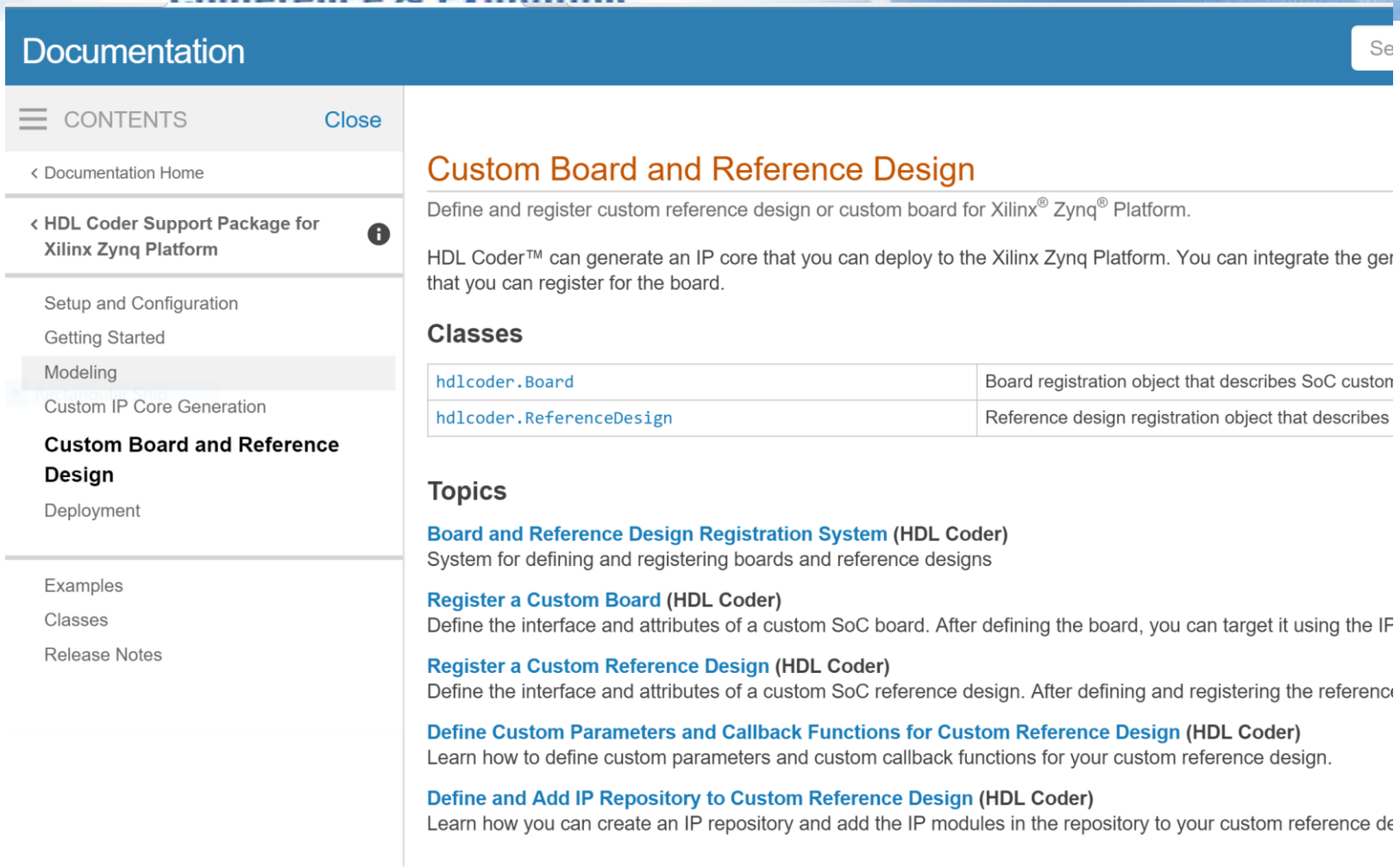
- **Without custom BSP**
  - Back & forth between teams
  - Interface definition mismatch probability
  - Prone to errors between steps



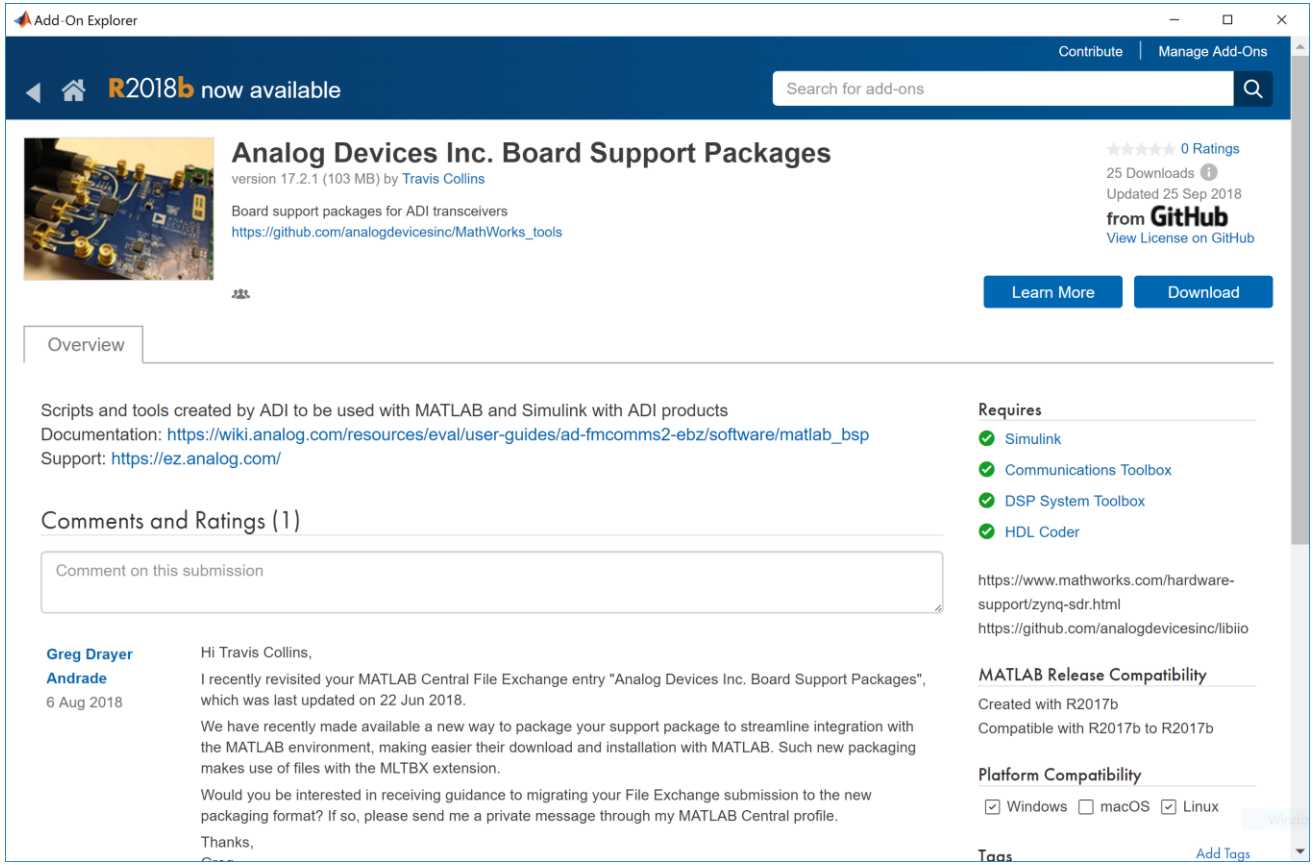
- **With custom BSP**
  - Same person does all the steps
  - Stay in the same flow as for the previous designs
  - Reduced integration time



- Process is well documented with MATLAB
- ADI BSP is an example of using this workflow
- HSPs from MathWorks are also examples of this workflow



Where can I get the code?



EDICON 2018
Electronic Design Innovation Conference & Exhibition
October 17-19 2018
Santa Clara Convention Center
Santa Clara, CA

Research
- AD9361 Behavioral Simulation
- PlutoSDR Streaming to MATLAB

Algorithm Development
- MATLAB reference implementation
- Hardware streaming

Design Elaboration
- Simulink modeling
- Hardware streaming
- Data type conversion

Prototype
- Deployment to development board
- Design optimization
- HDL Integration
- Driver Integration

Production
- Deployment to custom hardware
- Validation with complete hardware solution

PlutoSDR

RFSoM+FMC Carrier or Eval FMC + FPGA Carrier

PackRF or RFSoM + Custom Carrier

Custom

ANALOG DEVICES
AHEAD OF WHAT'S POSSIBLE™

- Thank You!

- Questions?