



# RF Solutions with Zynq® UltraScale+™ RFSoc

Presented By

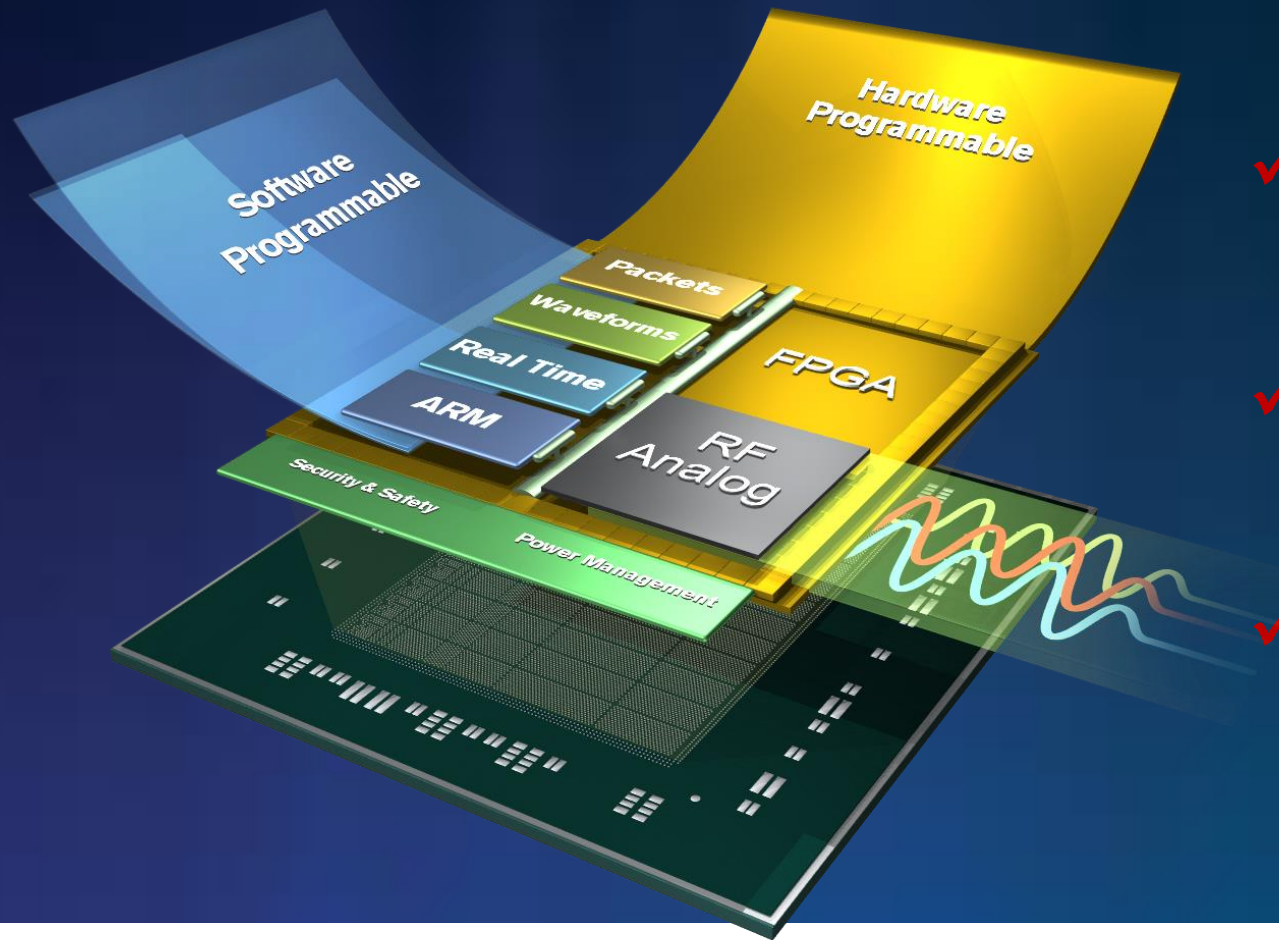
Glenn Steiner: System Software & SoC Solutions – Product and Technical Marketing  
David Brubaker: Product Line Manager – Zynq UltraScale+ RFSoc



# RFSoc Introduction



# The First Programmable RFSoc



- ✓ Integrated RF-Class Analog Converters and Error Correction Technology
- ✓ Delivering 50-75% Power & Footprint Reduction
- ✓ Full Programmability Across the RF Signal Chain

# Part of a Complete System Based on Production-Proven MPSoCs

*Monolithically Integrated*

**ARM Cortex** Processing System

- Quad-Core A53 (64-bit)
- Dual-Core R5 (32-bit)

**Hardened Engines**

- PCIe Gen3 & Gen4
- 100G Cores

**Programmable Logic**

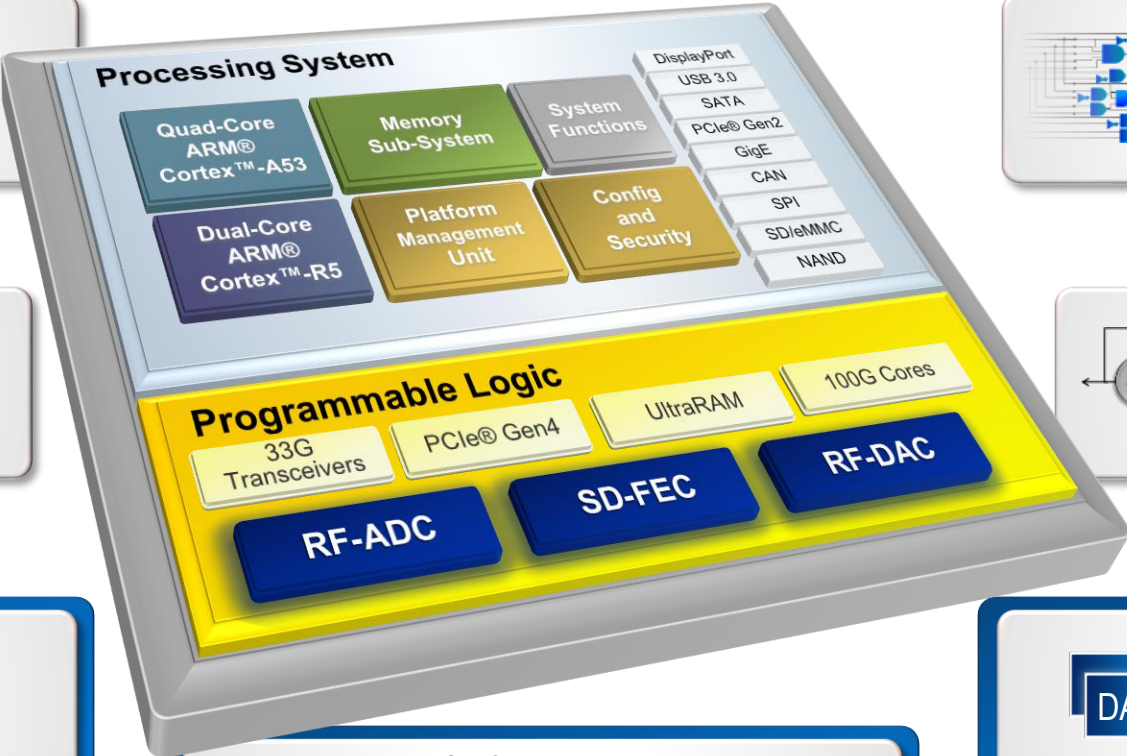
- 16nm FinFET
- UltraScale+ FPGA Fabric

**33G Transceivers**

- 33Gb/s
- 28G Backplane Capable

**DSP-Intensive**

- 4,272 DSP slices
- 7,612 GMACs



**ADC** Analog-to-Digital Converters

Up to 4.096 GSPS

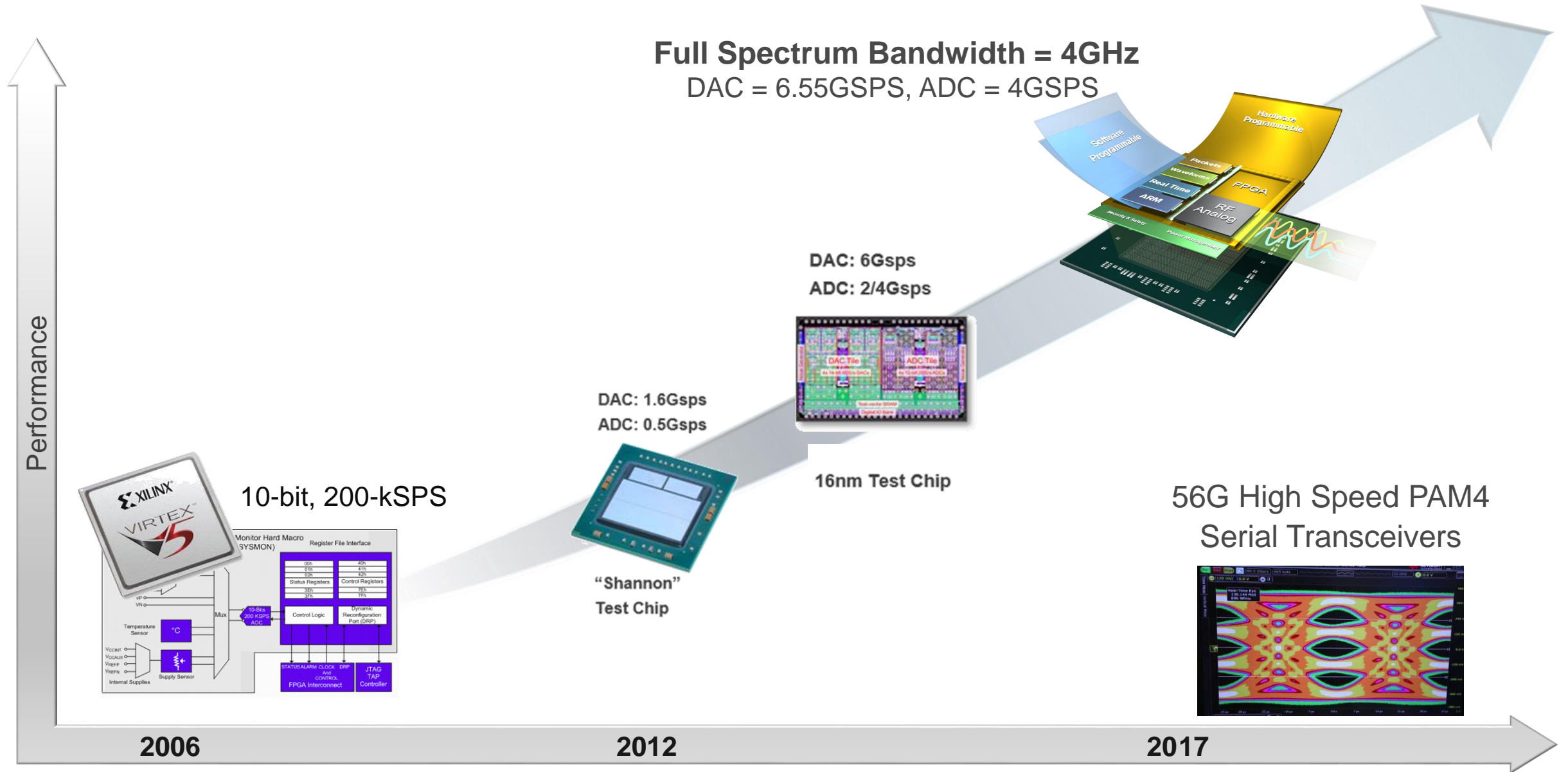
**DAC** Digital-to-Analog Converters

Up to 6.544 GSPS

**Soft Decision Forward Error Correction**

LDPC & Turbo Support

# Xilinx RF Converters – An Evolution and A Revolution

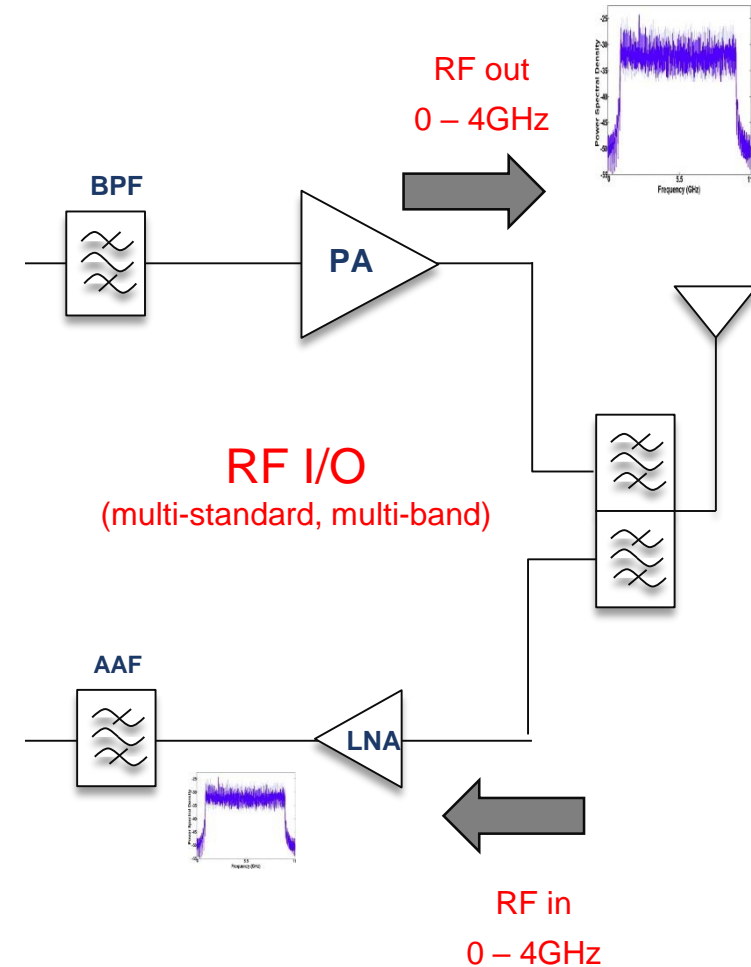
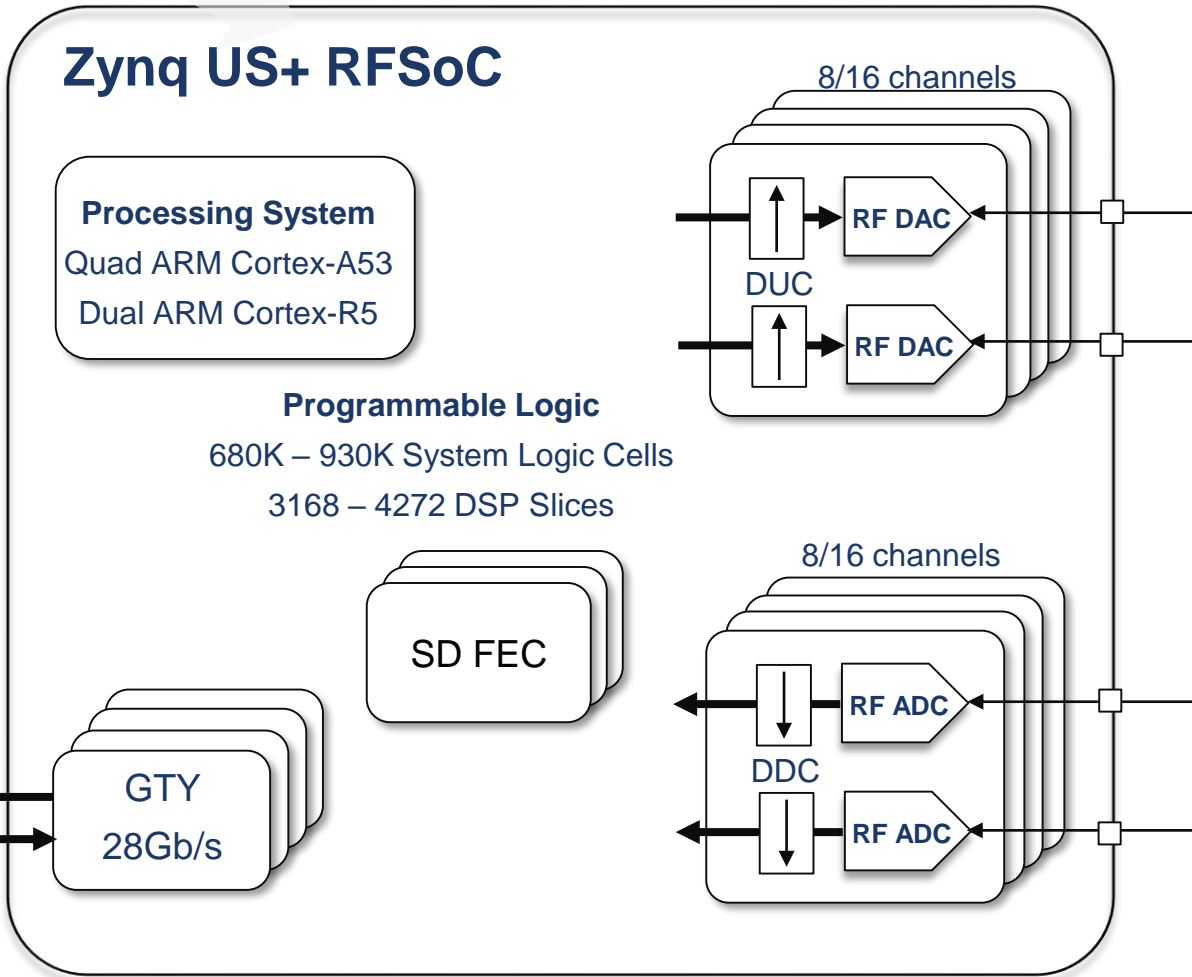
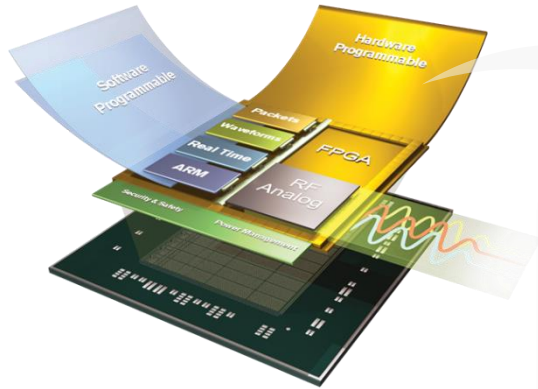


# RFSoc Applications





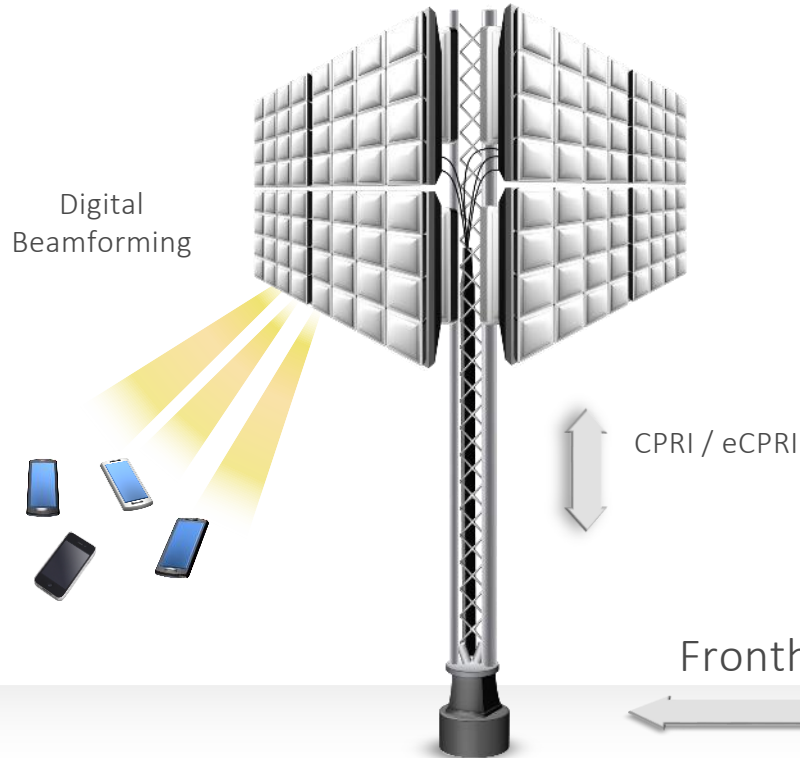
# Software Defined Radio on a Chip



# Enabling 5G Architectures

ZYNQ<sup>®</sup>  
RFSoC

Remote Radio for Massive-MIMO  
POWER • FORM FACTOR



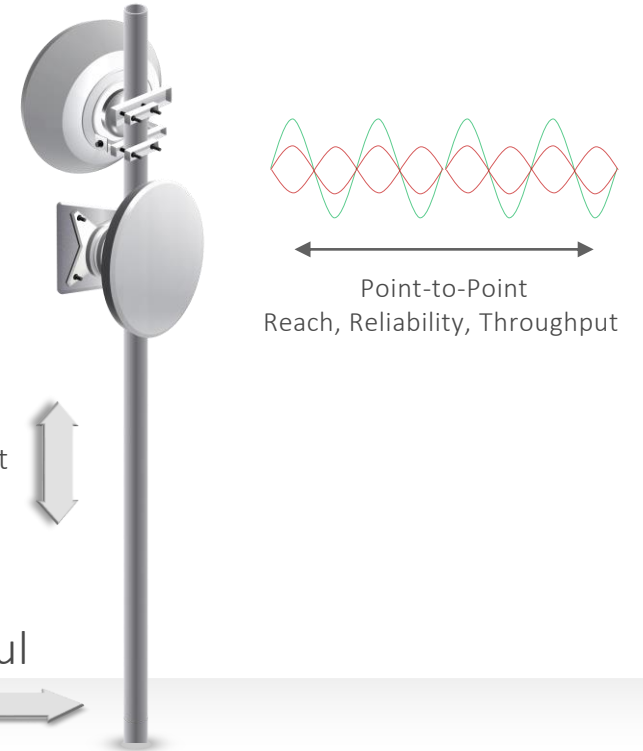
ZYNQ<sup>®</sup>  
RFSoC

Baseband  
THROUGHPUT • POWER EFFICIENCY



ZYNQ<sup>®</sup>  
RFSoC

Wireless Backhaul  
THROUGHPUT • POWER • FORM FACTOR



Fronthaul

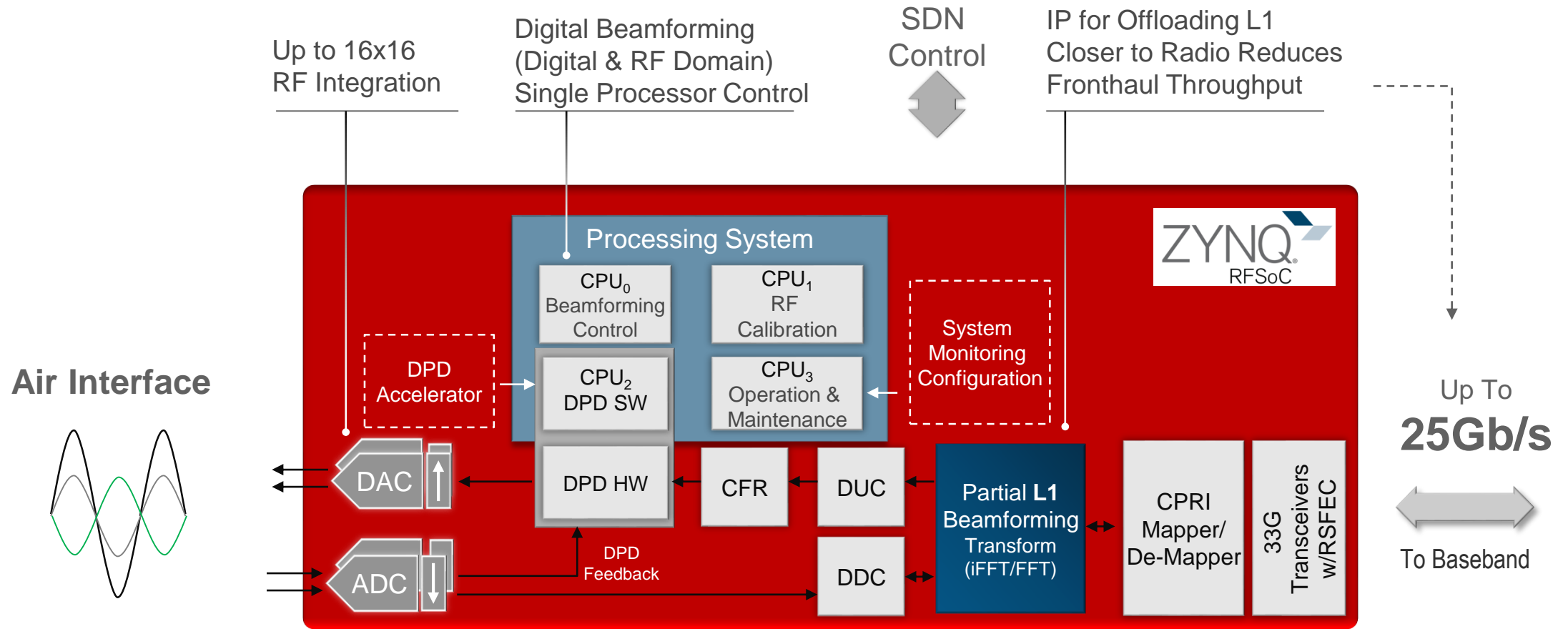
Backhaul

RFSoCs Advancing 5G Architectures

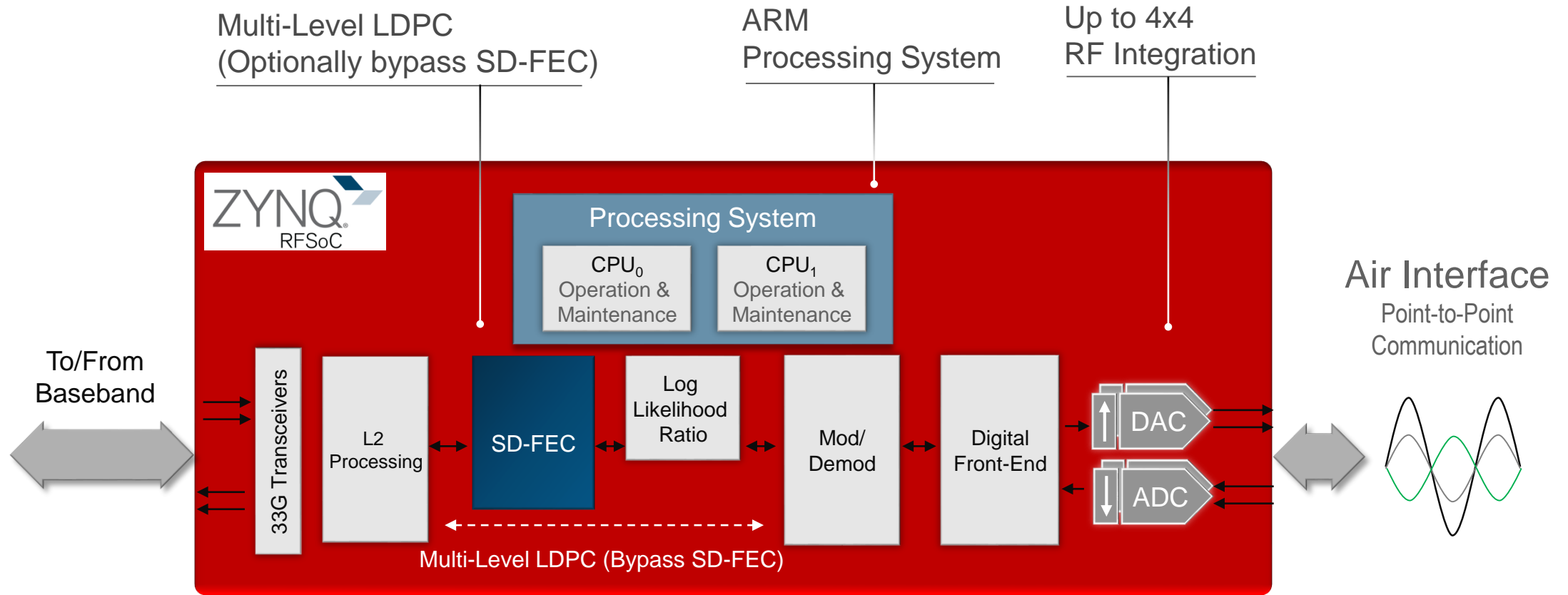
SPECTRAL EFFICIENCY • POWER EFFICIENCY • NETWORK DENSIFICATION



# Zynq UltraScale+ RFSoc in 5G New Radio

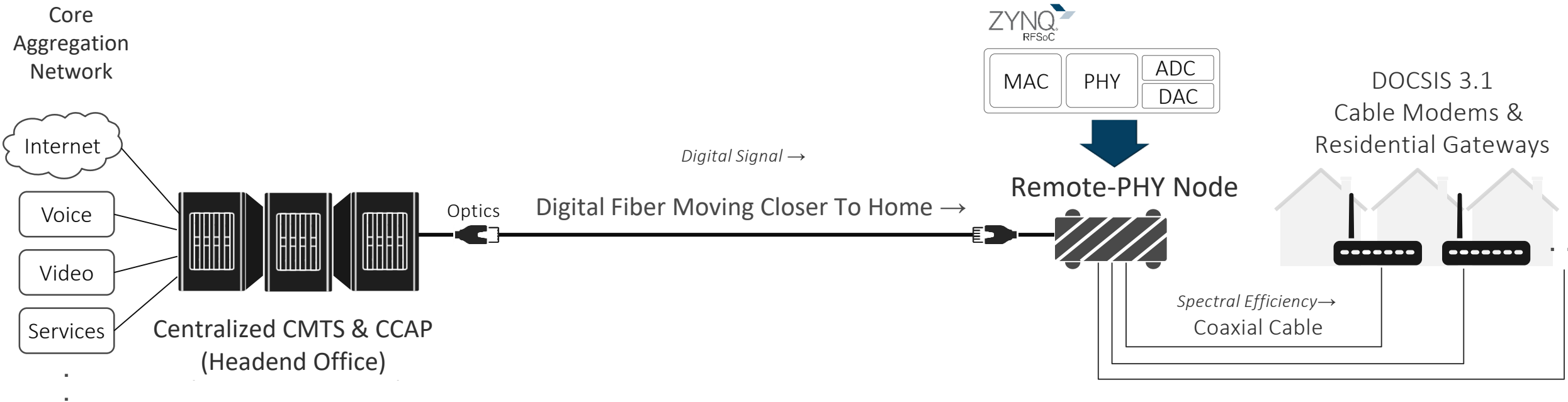


# Zynq UltraScale+ RFSoc in Wireless Backhaul



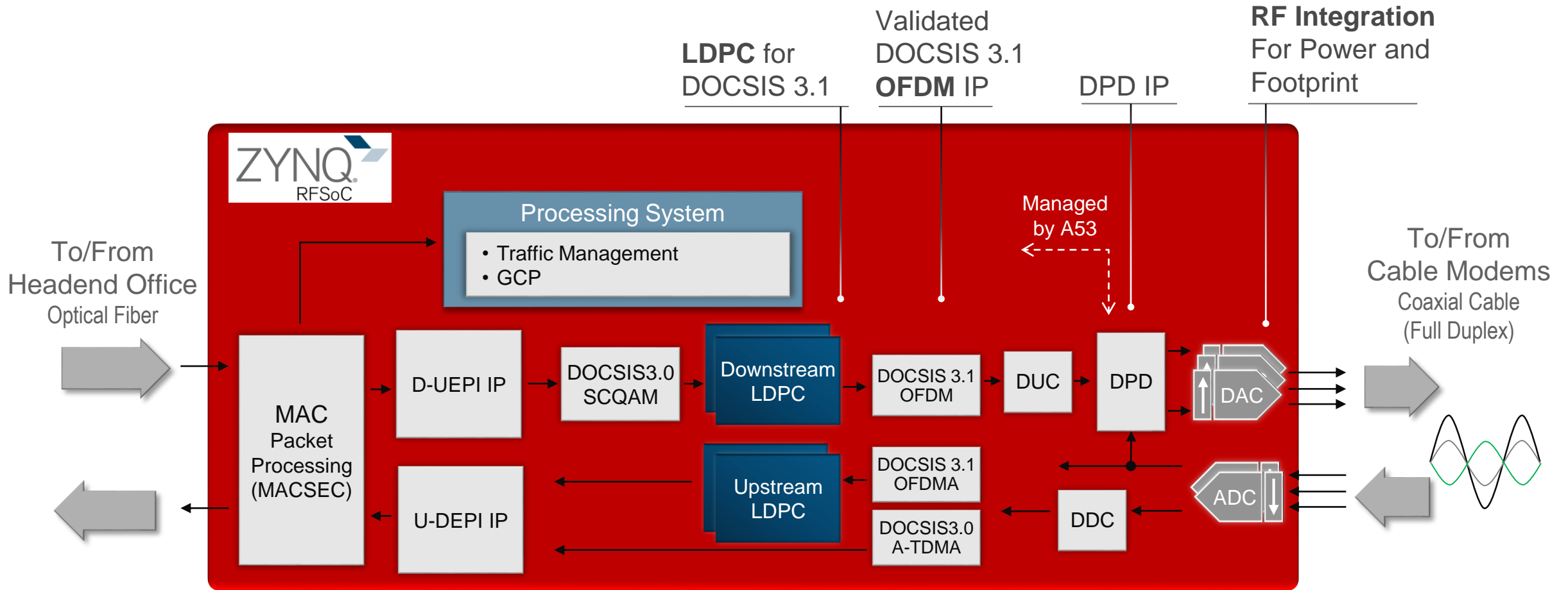
# DOCSIS 3.1 Remote PHY Node

## Distributed Access Architecture

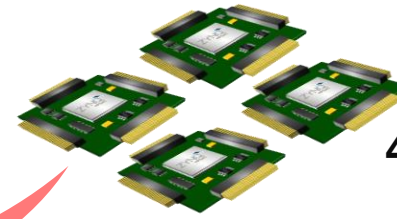


- “Fiber Deep” deployed closer to the home for greater bandwidth & power efficiency
- Remote PHY node moves PHY layer processing closer to the home, increasing network capacity

# DOCSIS 3.1 Remote PHY Node

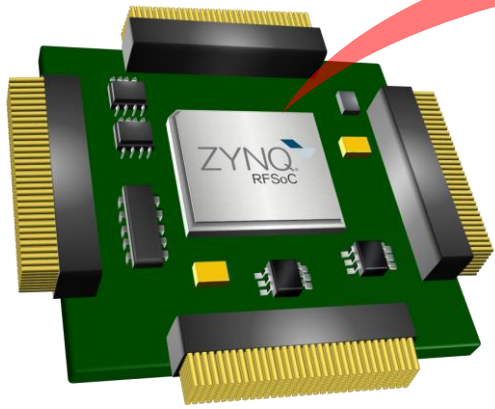


# Electronically Scanned Array



4 modules per Panel  
(64T/64R)

ZU29DR RFSoc Device  
To Implement one 16T/16R Module



10's to 100's  
Panels in an Array

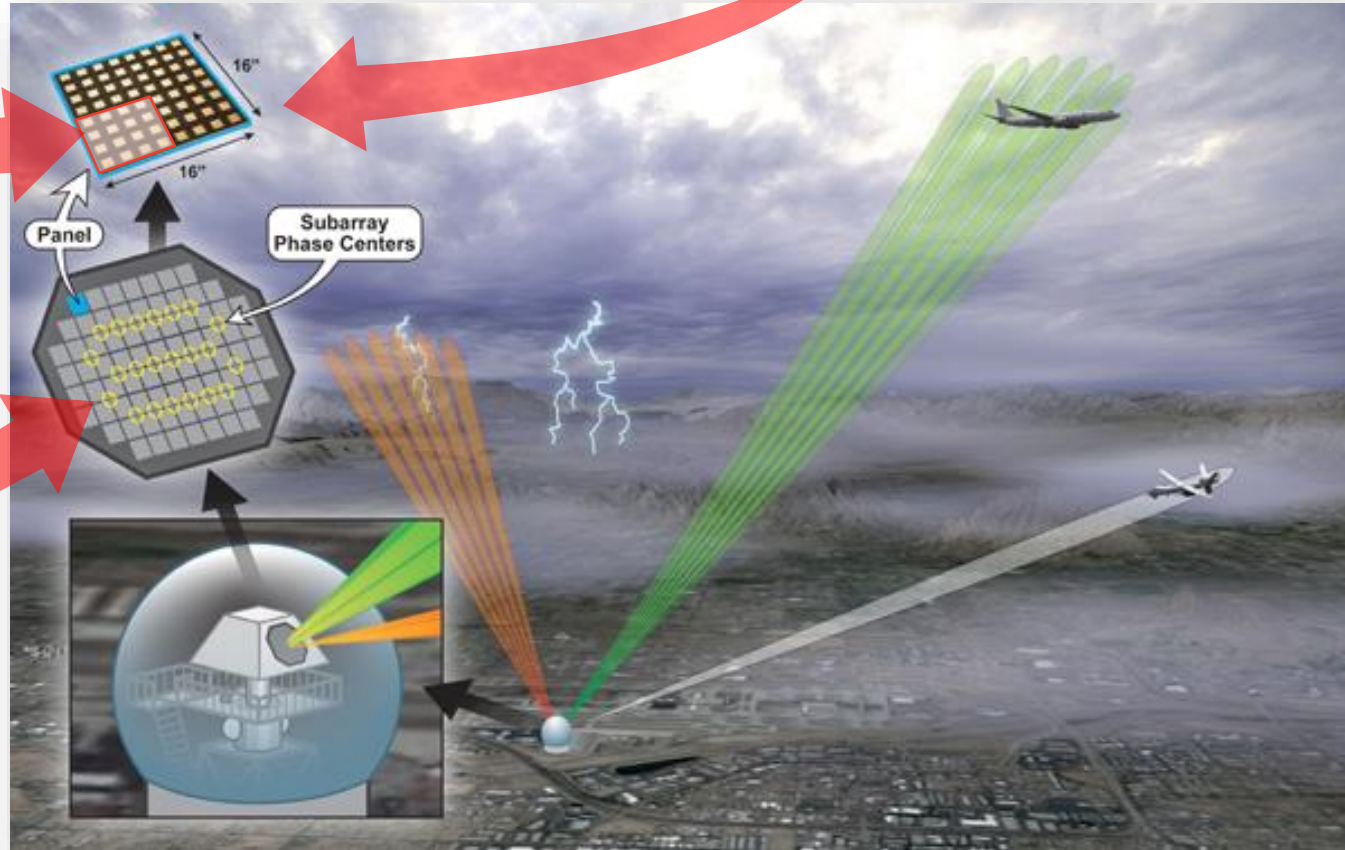


Image: MIT Lincoln Labs

# RFSoc Product Family and Benefits



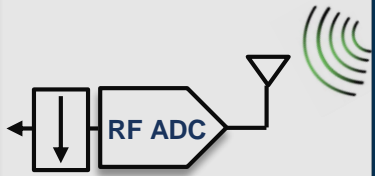


# RFSoc Family Overview

## Data Converter Enabled Devices

		Baseband	Wireless Radio		Backhaul, Remote-PHY	Phased Array Radar / Radio
		ZU21DR	ZU25DR	ZU27DR	ZU28DR	ZU29DR
RF Data Converters Soft Decision FEC	12-bit, 4GSPS ADC	-	8	8	8	-
	12-bit, 2GSPS ADC	-	-	-	-	16
	14-bit, 6.4GSPS DAC	-	8	8	8	16
	SD-FEC	8	-	-	8	-
Processing System & Programmable Logic	Application Processor Core	Quad-core ARM Cortex-A53 MPCore up to 1.5GHz				
	Real-Time Processor Core	Dual-core ARM Cortex-R5 MPCore up to 533MHz				
	High Speed Connectivity	DDR4-2600, PCIe Gen3 x16, PCIe Gen4, 100G Ethernet				
	Logic Density (System Logic Cells)	930K	678K	930K	930K	930K
	DSP Slices	4,272	3,145	4,272	4,272	4,272
	33G Transceivers	16	8	16	16	16

# Key Benefits of Integrated RF Data Converters



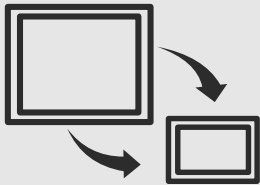
## Fully Programmable Direct RF Sampling Radio Platform

- RF-signal processing moved to the digital domain for a fully Programmable Solution
- Software Defined Solution for multi-mode and multi-band radios



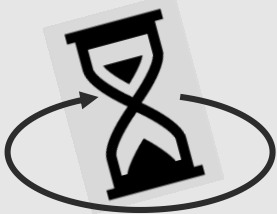
## Reduced System Power

- Reduces data converter power by using advanced technology and Digitally Assisted Analog
- Elimination of power hungry FPGA-to-Analog interfaces like JESD204



## Dramatic System Footprint Reduction

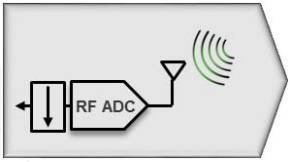
- Eliminates discrete converters and associated JESD PCB area
- Enables increasing channel counts across a range of new radio applications



## Shorter Design Cycle

- Simplified HW design with fewer RF components and the elimination of JESD Interfaces
- Simpler Data Converter Subsystem configuration from within Xilinx Vivado tools

# Programmable Direct RF Sampling For Radio



## > Moving RF Signal Processing into the Digital Domain

>> Flexible Platform based on Programmable HW and SW addresses a range of radio applications

## > Remove less flexible RF signal processing components

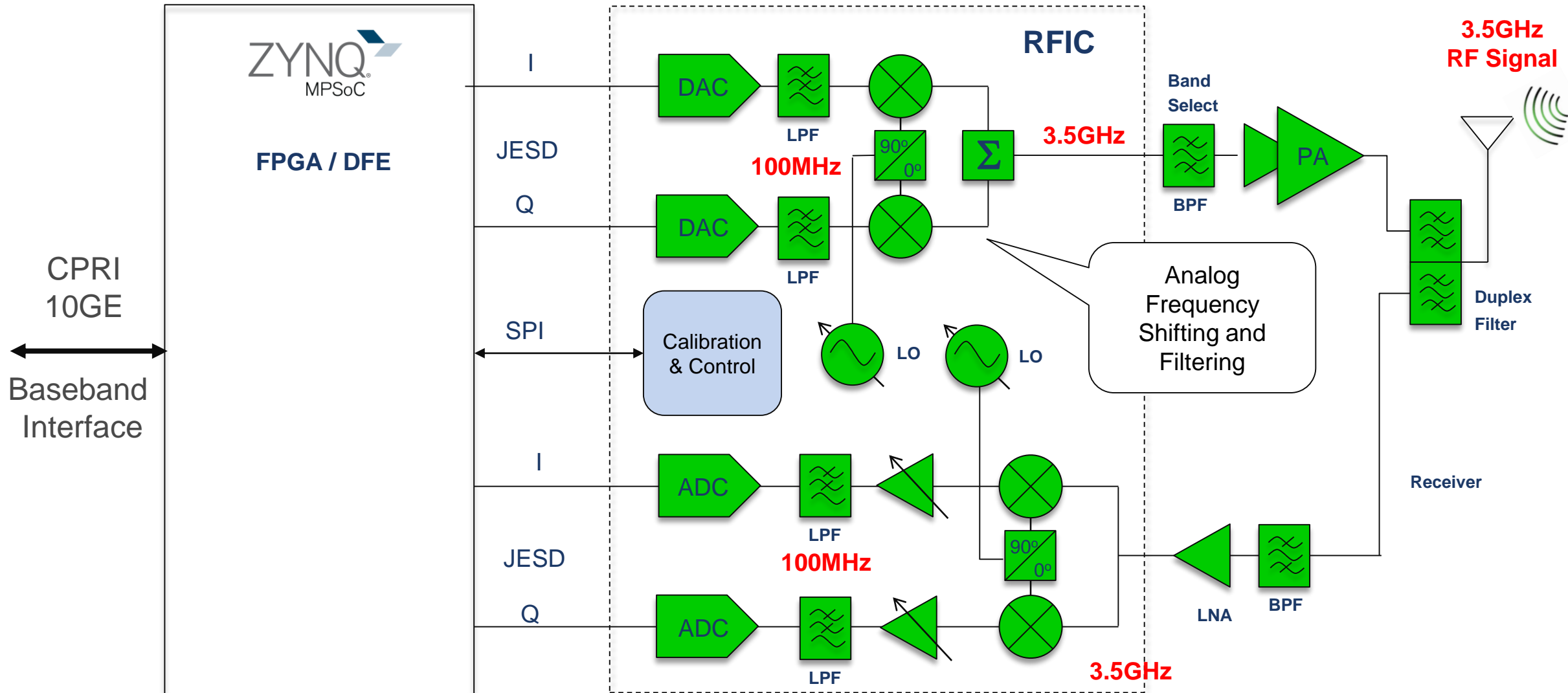
>> Analog/RF components have limited flexibility and performance

## > Enable a programmable platform that can be used across radio types

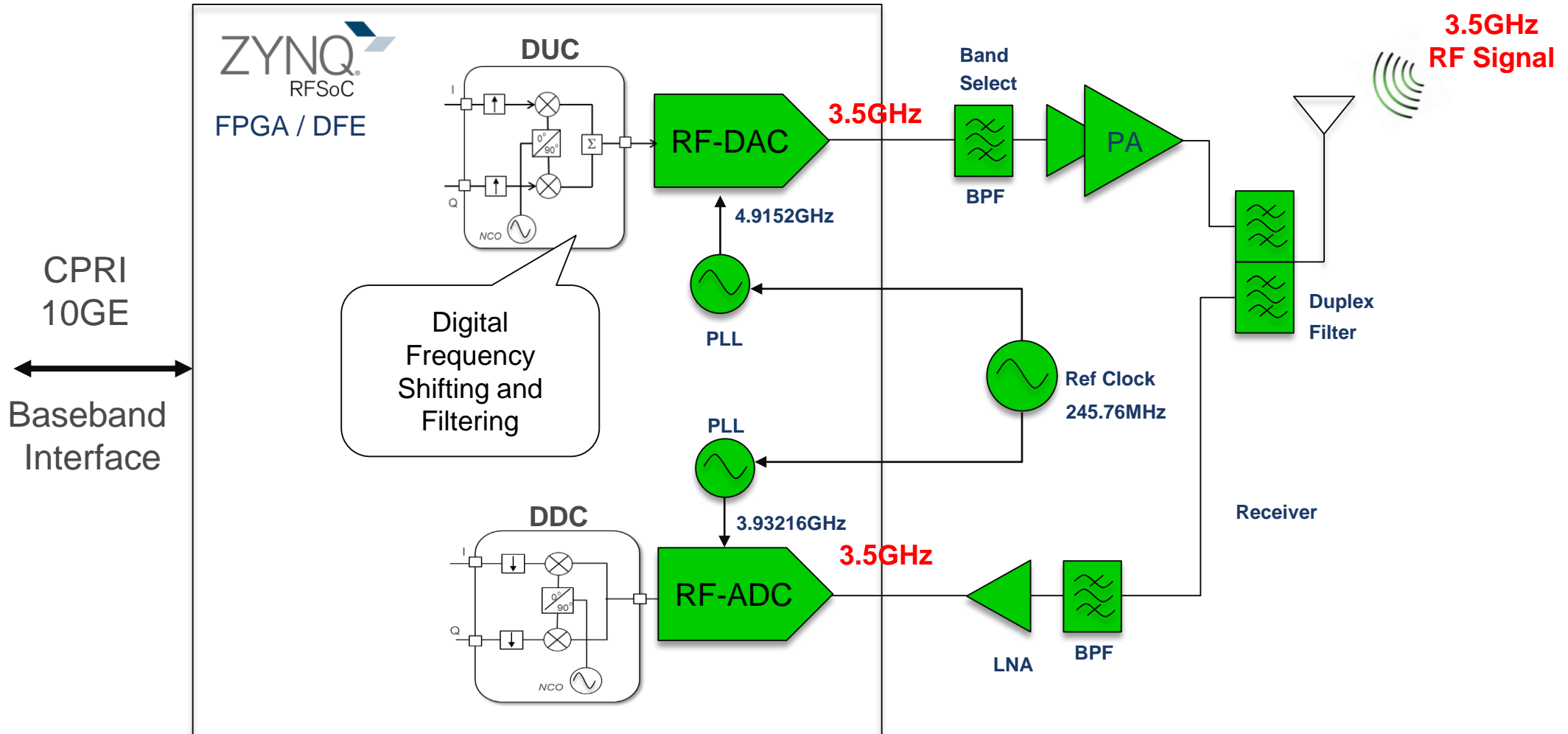
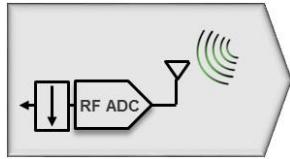
>> Multiple radio variants required to address global frequency allocations and different bandwidths

>> Ability to support new and emerging standards such as Carrier Aggregation

# Baseband/IF Sampling & RF Signal Processing

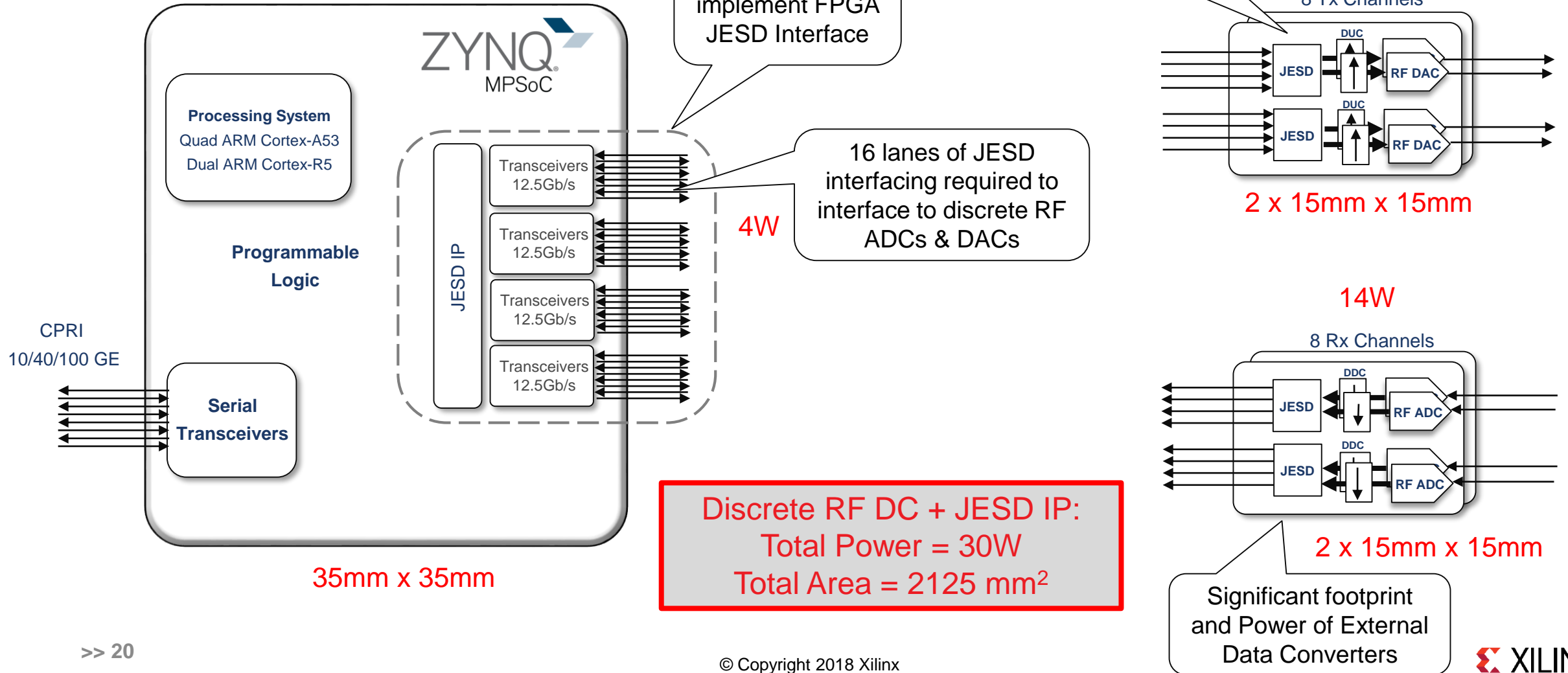


# Direct RF Sampling & Digital Signal Processing



# Discrete Direct RF Sampling Solution Case Study

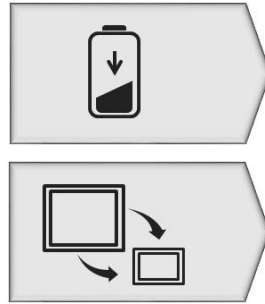
## 8T8R 200MHz Band 42 Radio





# RFSoc Integrated Direct RF Sampling Case Study

## 8T8R 200MHz Band 42 Radio



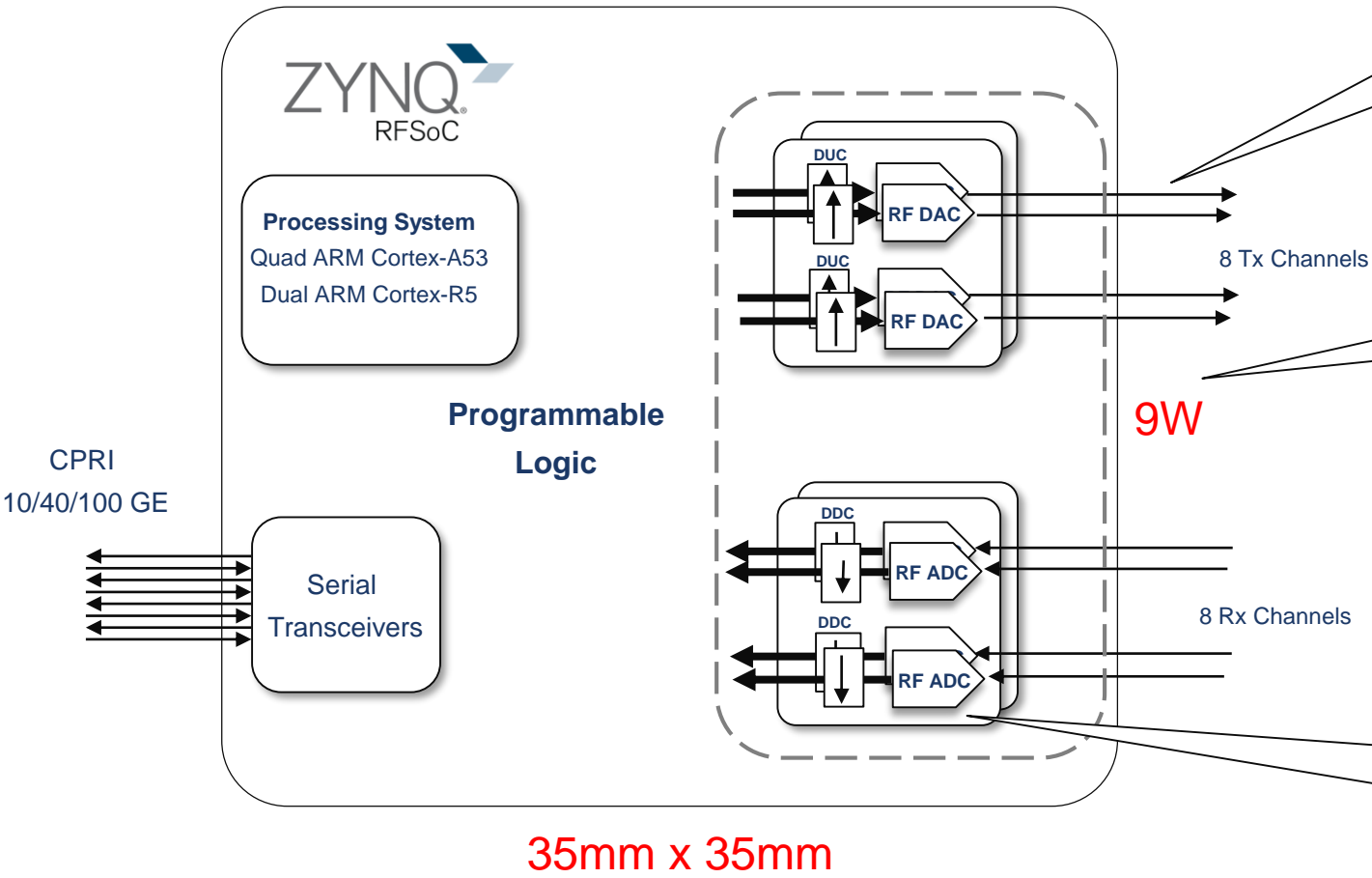
4W JESD interface is replaced with a 9W 8T8R RF Sampling Data Converter Subsystem

Eliminate the power and PCB area of 16 JESD lanes

Eliminate ~ 26W of discrete RF Data Converter Power and PCB area

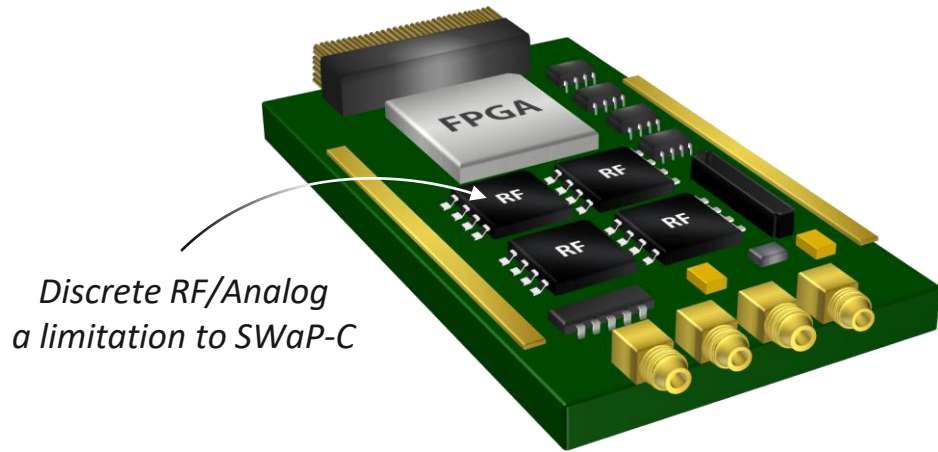
**Integrated RFSoc:**  
**Total Power = 9W**  
**Total Area = 1225 mm<sup>2</sup>**

Power consumption of Data Converters implemented on 16nm FinFET is greatly reduced by using the latest digitally assisted analog techniques



# RFSoc delivering Huge SWaP-C Advantages

4x4 Transmit / Receive Channels  
*Building Blocks for Phased Array Radar*

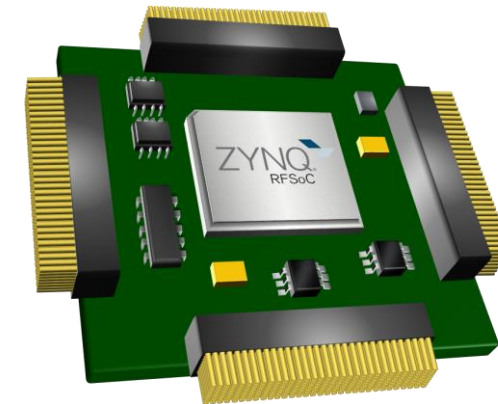


*Discrete RF/Analog  
a limitation to SWaP-C*

Most EW Modules Based on  
FPGAs or All Programmable SoCs

- Algorithms must be updated as threats change
- RF devices the “only ASIC left” on the board

16x16 Transmit / Receive Channels  
*Building Blocks for Phased Array Radar*



Government Programs  
Need to Scale SWaP-C

- **1000s** of modules in a system, **100s** of systems in a program
- Systems need **modularity** and **full** re-programmability

# Advantages of an Integrated SD-FEC



## High Throughput and Compute Bandwidth

- High performance core with robust LPDC and Turbo engines
- Configurable interface to control throughput per design requirements



## Flexible Customization and Design Integration

- Dynamically optimize parameters and codes for evolving standards
- Coupled with an HW & SW platform

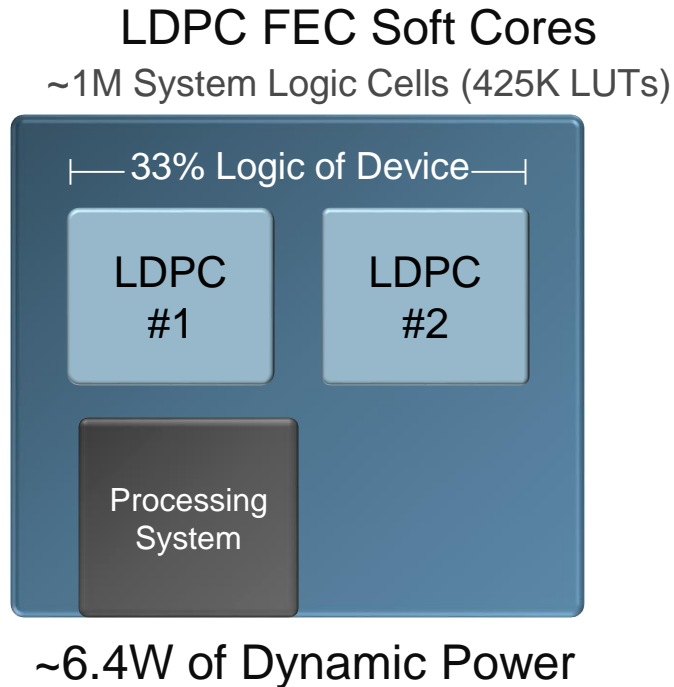


## Reduced System Power

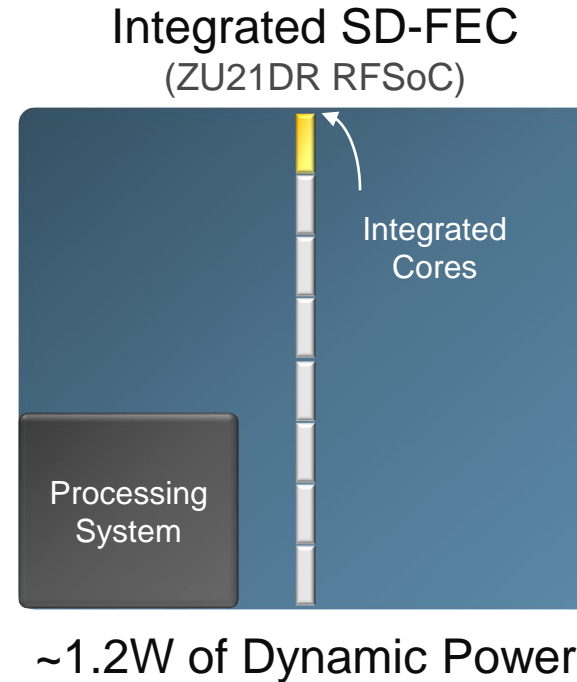
- Hardened 16nm FinFET silicon vs. soft implementation in FPGA fabric
- Meets thermal requirements for key applications

# Dramatic Power Reduction vs. Soft Core

## Example of 2x LDPC Cores at 2Gb/s Throughput



**80%**  
Power  
Reduction



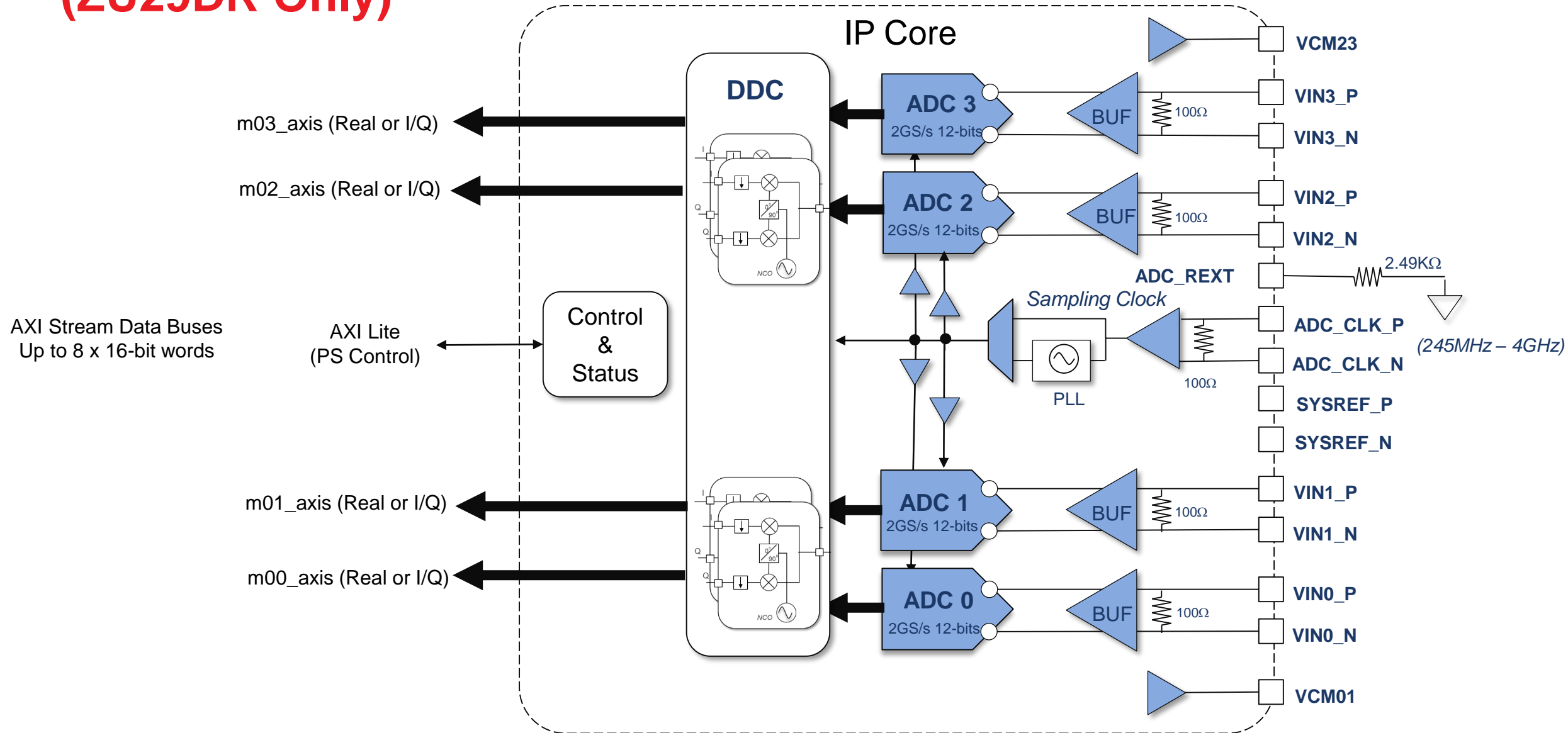
- 307MHz  $F_{MAX}$
- 150k LUTs
- 258 BRAM Kbits for storage & buffering

- 614MHz  $F_{MAX}$
- No additional resources required
- More flexibility & functionality available vs. soft core

# Zynq® UltraScale+™ RFSoc RF ADC & RF DAC Overview

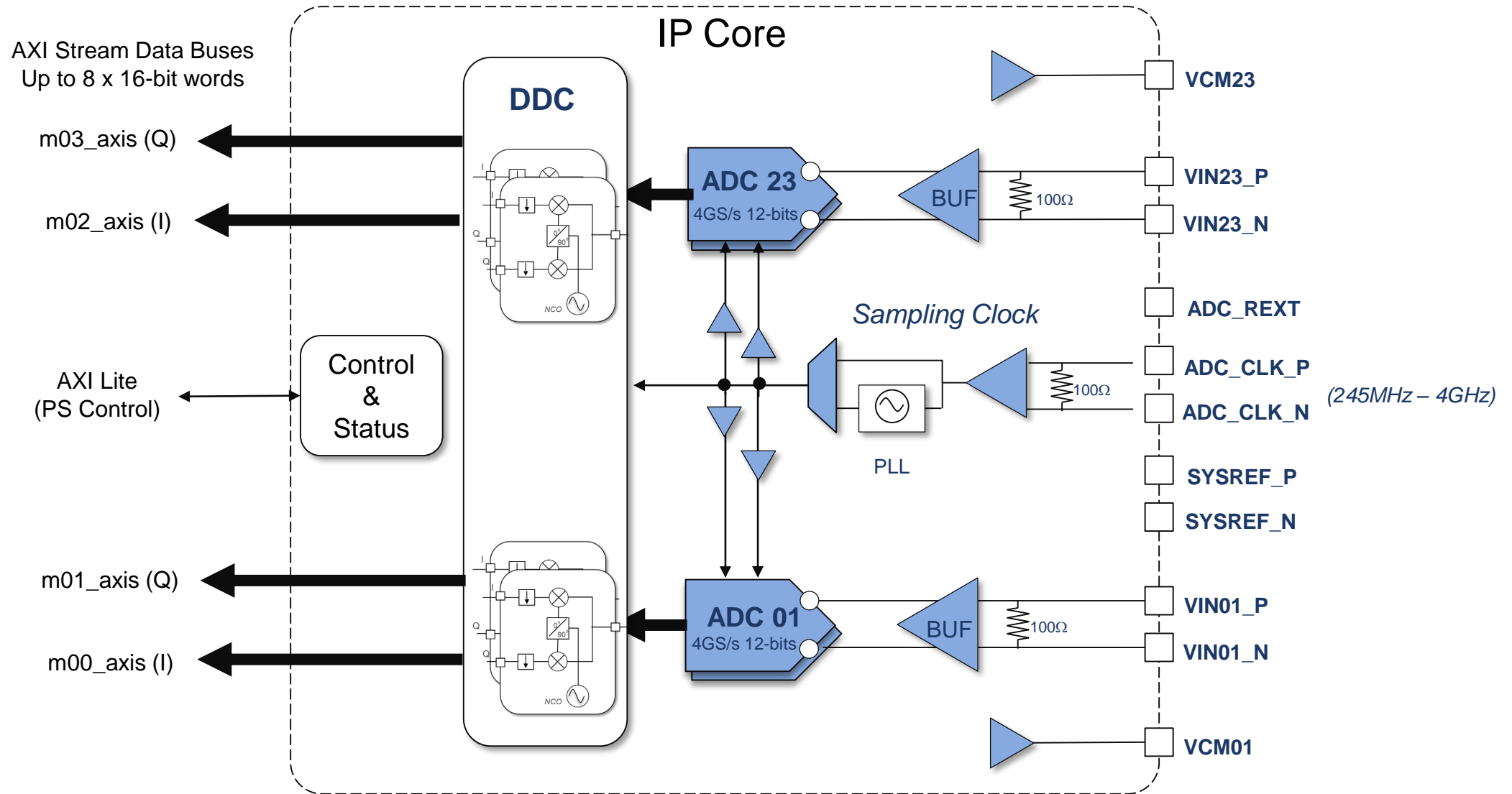


# RF ADC Block 2GS/s Configuration (ZU29DR Only)

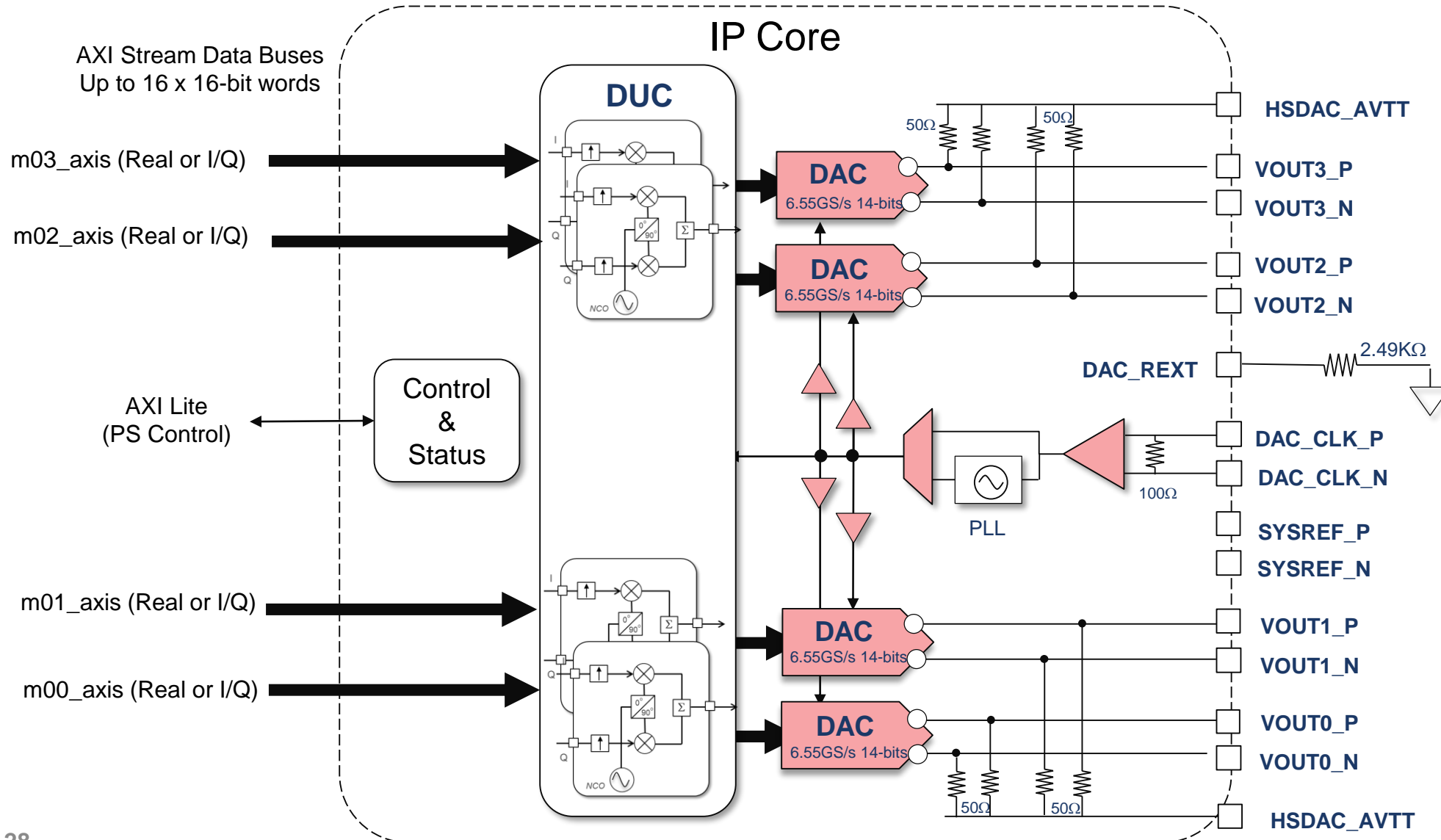




# RF ADC Block 4GS/s Configuration (ZU25DR, ZU27DR, & ZU28DR Only)



# RF DAC Block Diagram (ZU25DR, ZU27DR, ZU28DR, & ZU29DR)



# RFSoc Product Solutions



# Zynq UltraScale+ RFSoc Kits

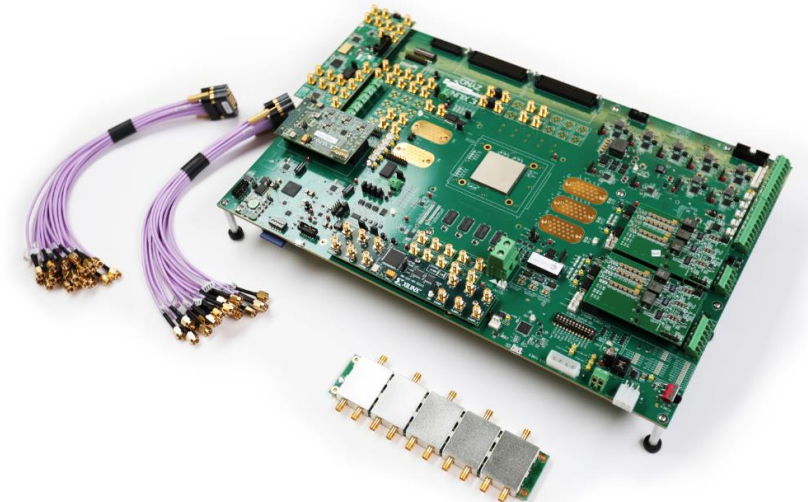
## > Zynq UltraScale+ RFSoc ZCU111 Evaluation Kit

- >> XCZU28DR-2FFVG1517E RFSoc
  - 8x 4GSPS 12-bit ADCs
  - 8x 6.5GSPS 14-bit DAC
  - 8 soft-decision forward error correction (SD-FECs)
- >> FMC+ :12 x 32.75 Gb/s GTY transceivers and 34 user defined differential I/O signals
- >> XM500 RFMC balun transformer card w 4 DACs/ 4 ADCs to baluns 4 DACs/ 4 ADCs to SMAs
- >> Price: \$8,995
- >> Part Number: EK-U1-ZCU111-G



## > Zynq UltraScale+ RFSoc ZCU1275 Characterization Kit

- >> XCZU29DR RFSoc
  - 16x 2GSPS 12-bit ADCs
  - 16x 6.5GSPS 14-bit DAC
- >> Balun Board, Bullseye Cables, Filters
- >> Price: \$14,995
- >> Part Number: CK-U1-ZCU1275-G



# RF DC Evaluation Tool Highlights (ZCU111)

## > LabVIEW based evaluation GUI running on PC

- Ethernet Interface to board

## > Loopback (DAC to ADC) for multiple channels evaluation

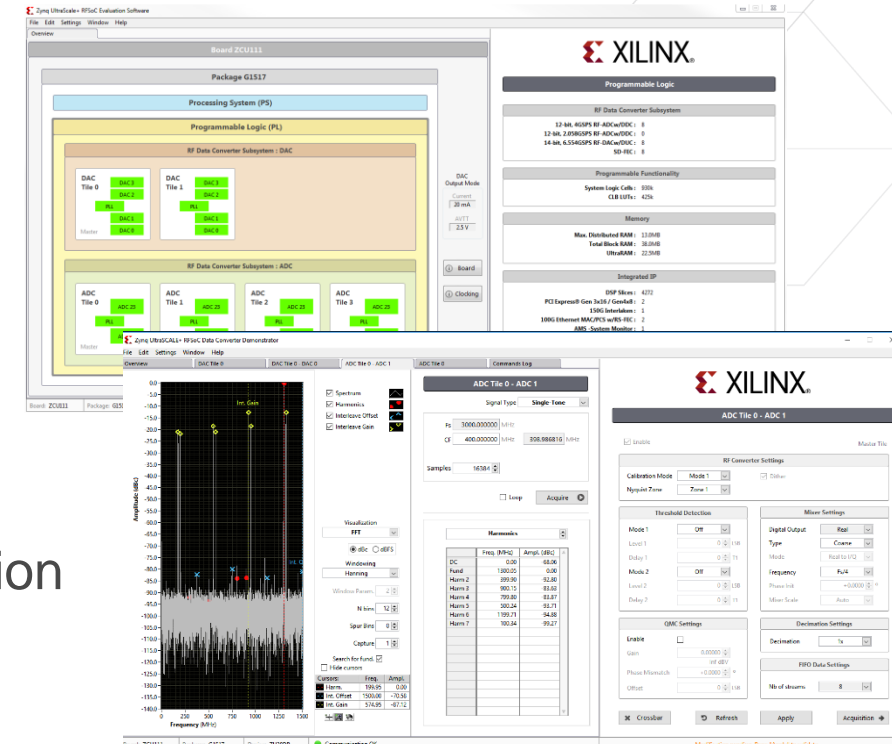
- Key parameters measurement (i.e. NSD, SFDR, THD, Harmonics, Spurious Performance)
- 2 tones test (i.e. IM3)

## > DAC / ADC standalone evaluation

- DAC analysis => generate test vectors
- ADC analysis => FFT spectrum analysis for various input test signals with signal generator

## > Advance Features

- Nyquist zone, DDC/DUC, Mixer, NCO, Looping feature
- File input / export for customized test vectors / modulation



# RF Analyzer Debug Tool Highlights

## > Act as a debug tool

- Support the RFSoc configuration
- Cross-check features and functionalities
- Ease of use – no FPGA experience required
- Not require any additional external resources (i.e. DDR)

## > Compatible with any platforms

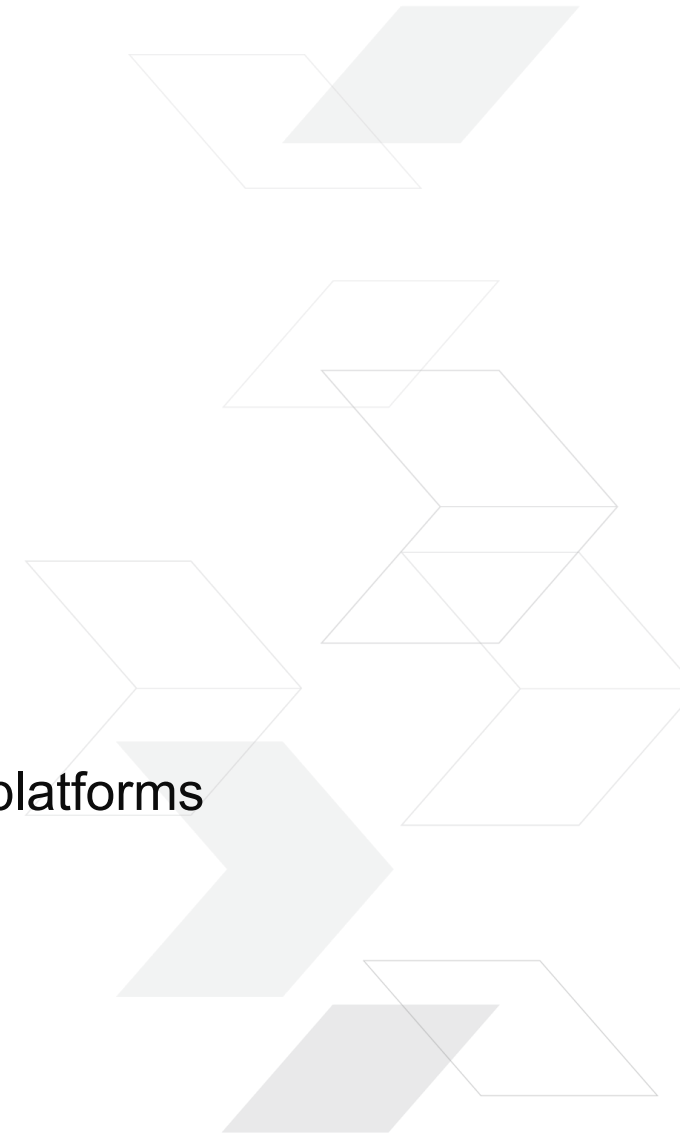
- RFSoc performance can be evaluated in any customers' boards

## > JTAG based communication interface

- JTAG USB cables connected between debug tool & customers' platforms
- All communications via JTAG:
  - CTRL: JTAG-to-UART
  - DATA: JTAG-to-AXI

## > Features

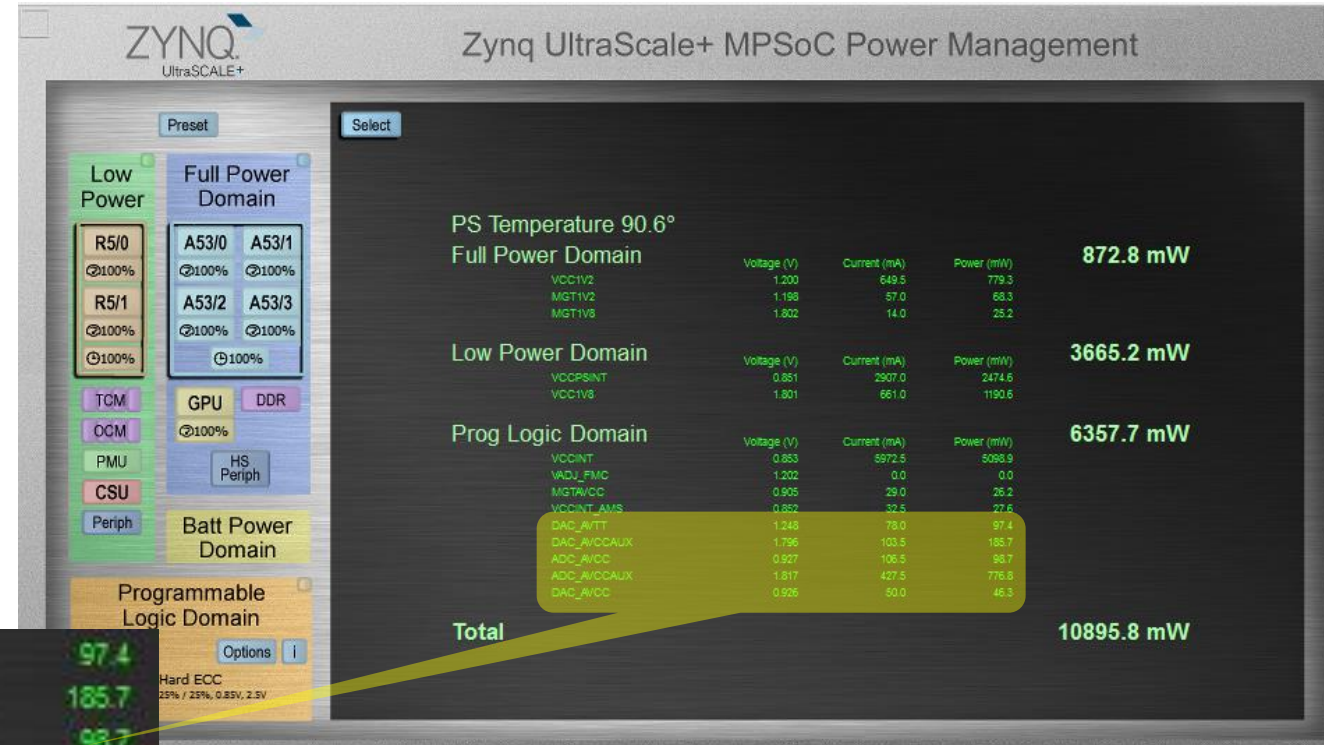
- Simplified version of RF DC Evaluation Tool





# ZCU111 Power Measurement & Power Advantage Tool

- > **Tool Measures & Displays All Rails, SysMon Voltages & Temperature**
  - >> Including RFSoc Converter Power
- > **Text, Plots, & Data Logging Included**
- > **Currently supported on ZCU102, ZCU106 and NOW ZCU111**
- > **Works with Customer Designs Without Impact**
  - >> Less temperature unless R5 code included
- > **Separate GUI Enables More to Be Seen**



Rail	Voltage (V)	Current (mA)	Power (mW)
DAC_AVTT	1.248	78.0	97.4
DAC_AVCCAUX	1.796	103.5	185.7
ADC_AVCC	0.927	106.5	98.7
ADC_AVCCAUX	1.817	427.5	776.8
DAC_AVCC	0.926	50.0	46.3

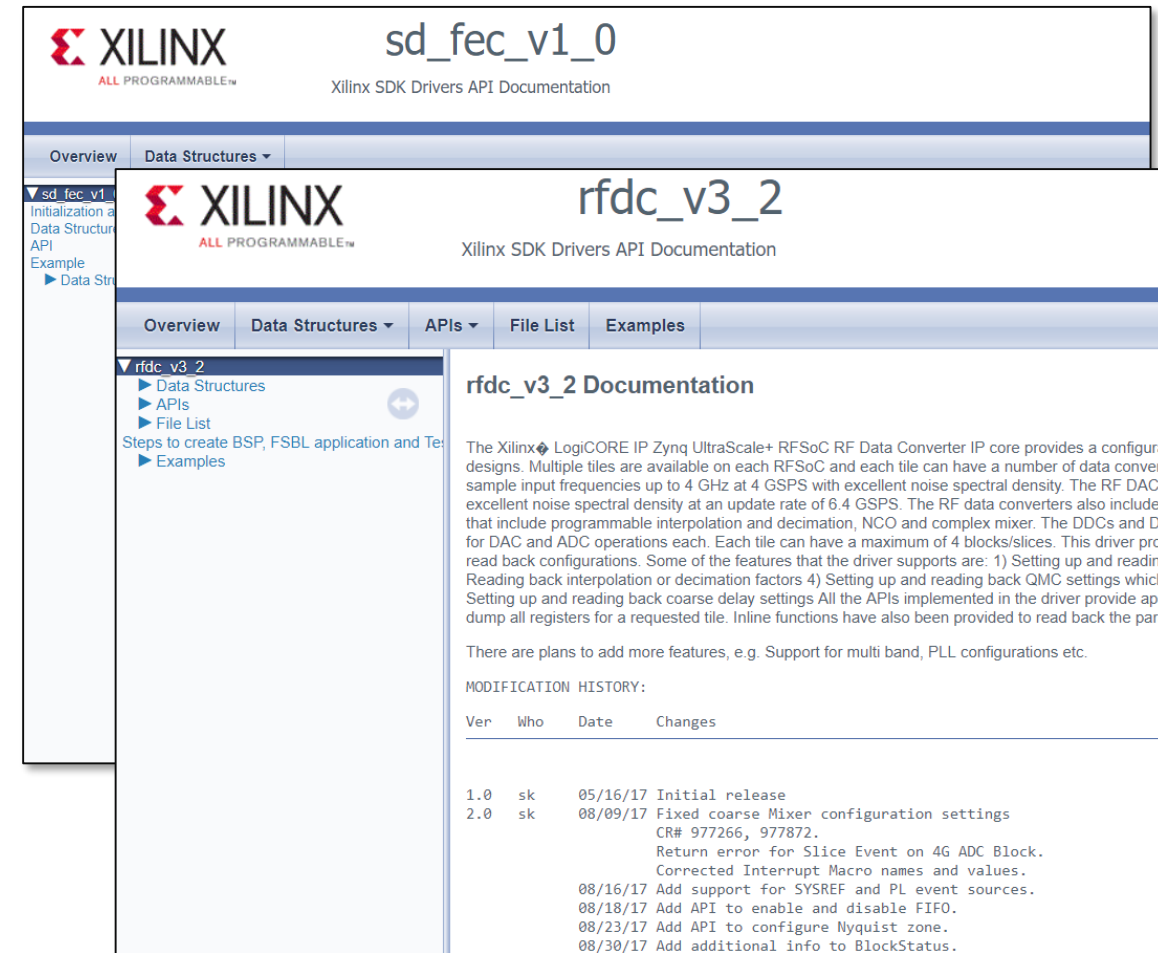
# Documentation

## > PG269 – RF-ADC/DAC Product Guide

- >> driver/API – Appendix C
- >> HTML driver docs in XSDK build (system.mss file Documentation link, GitHub)
- >> Xilinx linux/baremetal wikis

## > PG256 – SD-FEC Product Guide

- >> bare-metal driver/API – Appendix C
- >> Linux driver/API from source files via Doxygen
- >> HTML driver docs in XSDK build (system.mss file Documentation link, GitHub)
- >> Xilinx linux wiki



The screenshot displays the Xilinx SDK Drivers API Documentation for the `rfdc_v3_2` component. The page is titled "XILINX ALL PROGRAMMABLE™" and "Xilinx SDK Drivers API Documentation". The navigation menu includes "Overview", "Data Structures", "APIs", "File List", and "Examples". The main content area is titled "rfdc\_v3\_2 Documentation" and contains a detailed description of the Xilinx LogiCORE IP Zynq UltraScale+ RFSoc RF Data Converter IP core. It also includes a "MODIFICATION HISTORY" table with columns for Version, Who, Date, and Changes.

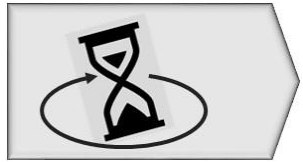
Ver	Who	Date	Changes
1.0	sk	05/16/17	Initial release
2.0	sk	08/09/17	Fixed coarse Mixer configuration settings CR# 977266, 977872. Return error for Slice Event on 4G ADC Block. Corrected Interrupt Macro names and values.
		08/16/17	Add support for SYSREF and PL event sources.
		08/18/17	Add API to enable and disable FIFO.
		08/23/17	Add API to configure Nyquist zone.
		08/30/17	Add additional info to BlockStatus.

## Also very helpful to new ZU+ users:

- >> UG1209 – ZU+ MPSoC Embedded Design Tutorial
- >> UG1228 – ZU+ Embedded Design Methodology Guide
- >> UG1087 – ZU+ MPSoC Register Reference Guide

# Zynq® UltraScale+™ RFSoc Hardware & Software Design Flow

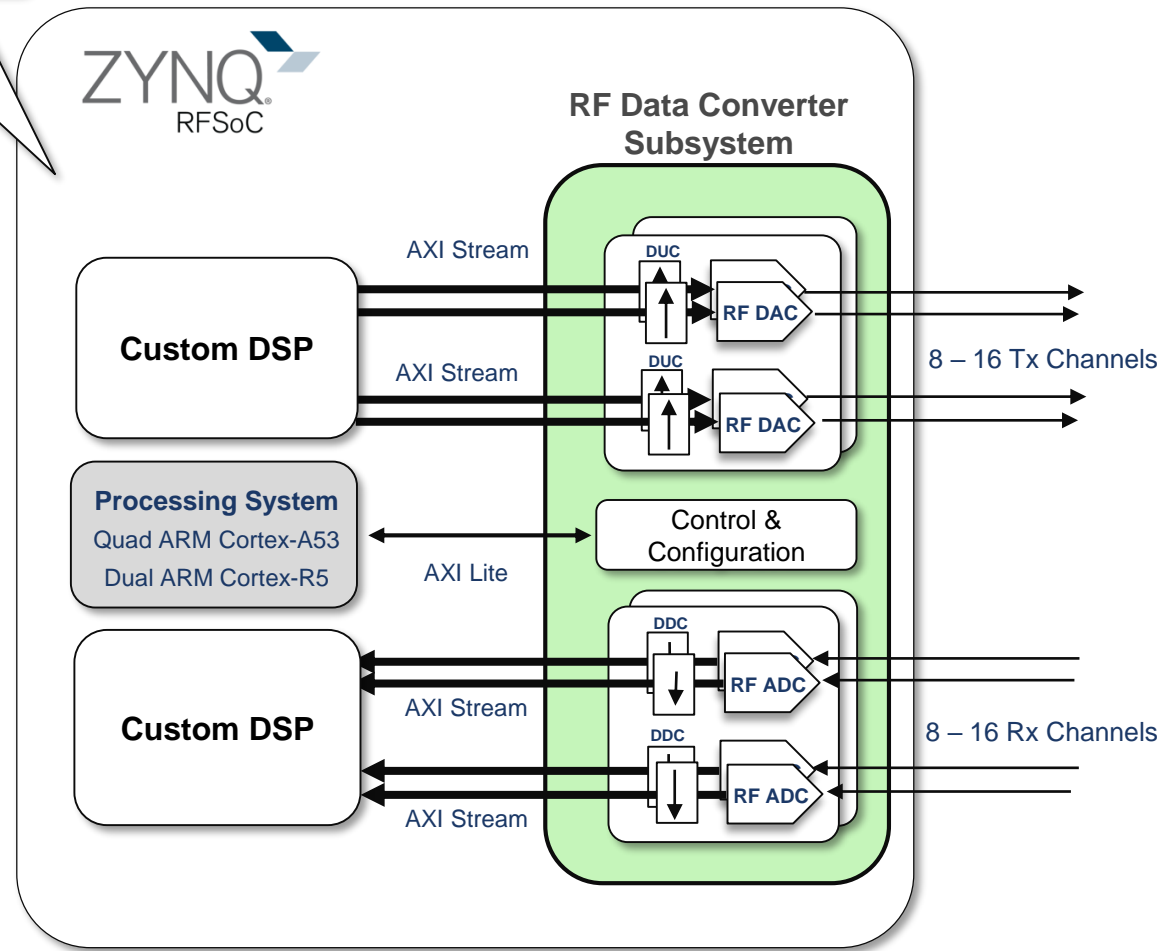
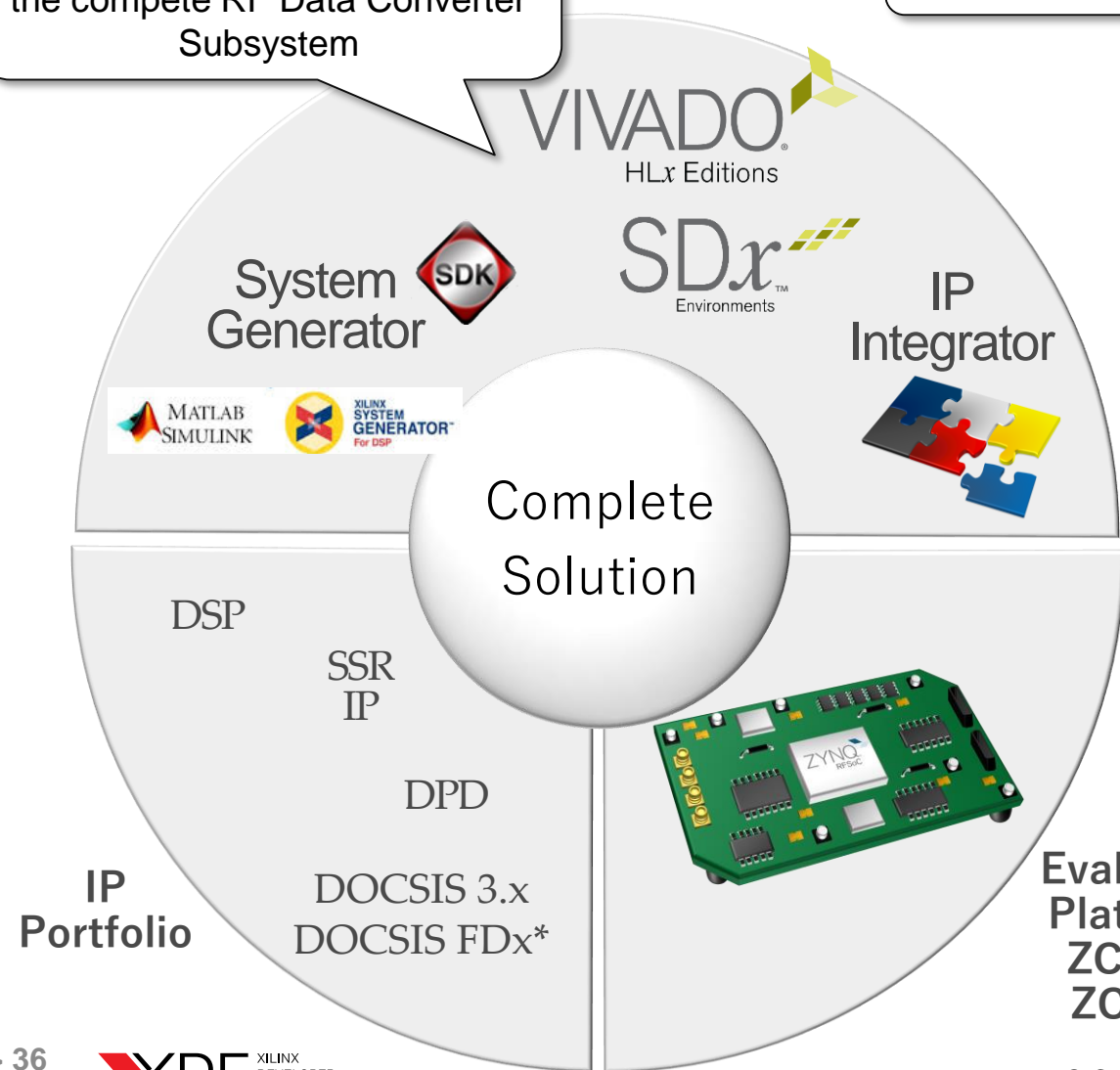
# RFSoc Design Flow Overview



Xilinx tools support the configuration and integration of the complete RF Data Converter Subsystem

Tool Suite

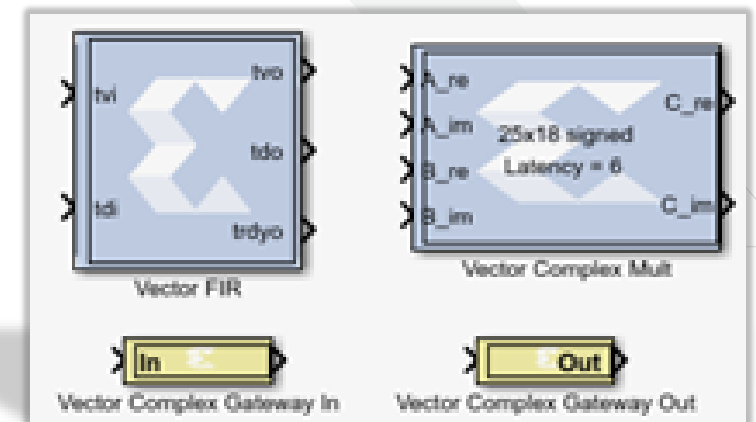
Up to 6 TMACs of customizable DSP



Evaluation Platforms  
ZCU111  
ZC1275

# Super Sample Rate Support & IP

- > **Super Sample Rate – Processing multiple samples per clock**
  - >> Data into FPGA @ much higher sample rate than the FPGA clock
    - Sample rate into FPGA greater than PL clock rate
  - >> Need to parallelize the input and process multiple samples per FPGA clock cycle
  - >> Requested by A&D customers where RF-ADC/DACs do not meet there DUC/DDCs needs
  
- > **SysGen has developed an SSR programmatical library of 26 SSR IP blocks**
  - >> Including FIR, Complex Mult, Mult, DDS and others (2018.3)
  - >> SysGen provides additional Super Rate Support



# RF-ADC/DAC Implementation Steps

VIVADO

IDE

- 1. Add an RF-ADC/DAC instance using IPI**
  - Single instance
- 2. Use GUI to configure and customize the IP**
  - Right click IP to generate example design and testbench, plus DAC HW stimulus generator and ADC HW sink
  - Use BSPs for HW examples per board
- 3. Connect the RF-ADC/DAC instance to the PS, additional logic, RTL, outside world...**
- 4. Implement (Synthesis, PnR...)**
- 5. Generate the bitstream, export the HDF**
- 6. Implement your Software Project**
  - XSDK, Petalinux, 3rd party...

The image displays two screenshots from the Vivado IDE. The top screenshot shows the 'Re-customize IP' window for the 'Zynq UltraScale+ RF Data Converter (2.0)'. The 'Basic' tab is active, showing the component name 'usp\_rf\_data\_converter\_0' and various configuration options like 'System Configuration' and 'Converter Setup'. A search dropdown menu is open, listing various IP blocks, with 'Zynq UltraScale+ RF Data Converter' selected. The bottom screenshot shows a block diagram of the design. It features a 'ZYNQ UltraSCALE+' block connected to a 'ps8\_0\_axi\_periph' block. The RF-ADC/DAC instance is connected to the system clock and data buses. The diagram also shows the 'Processor System Reset' and 'AXI Interconnect' blocks.





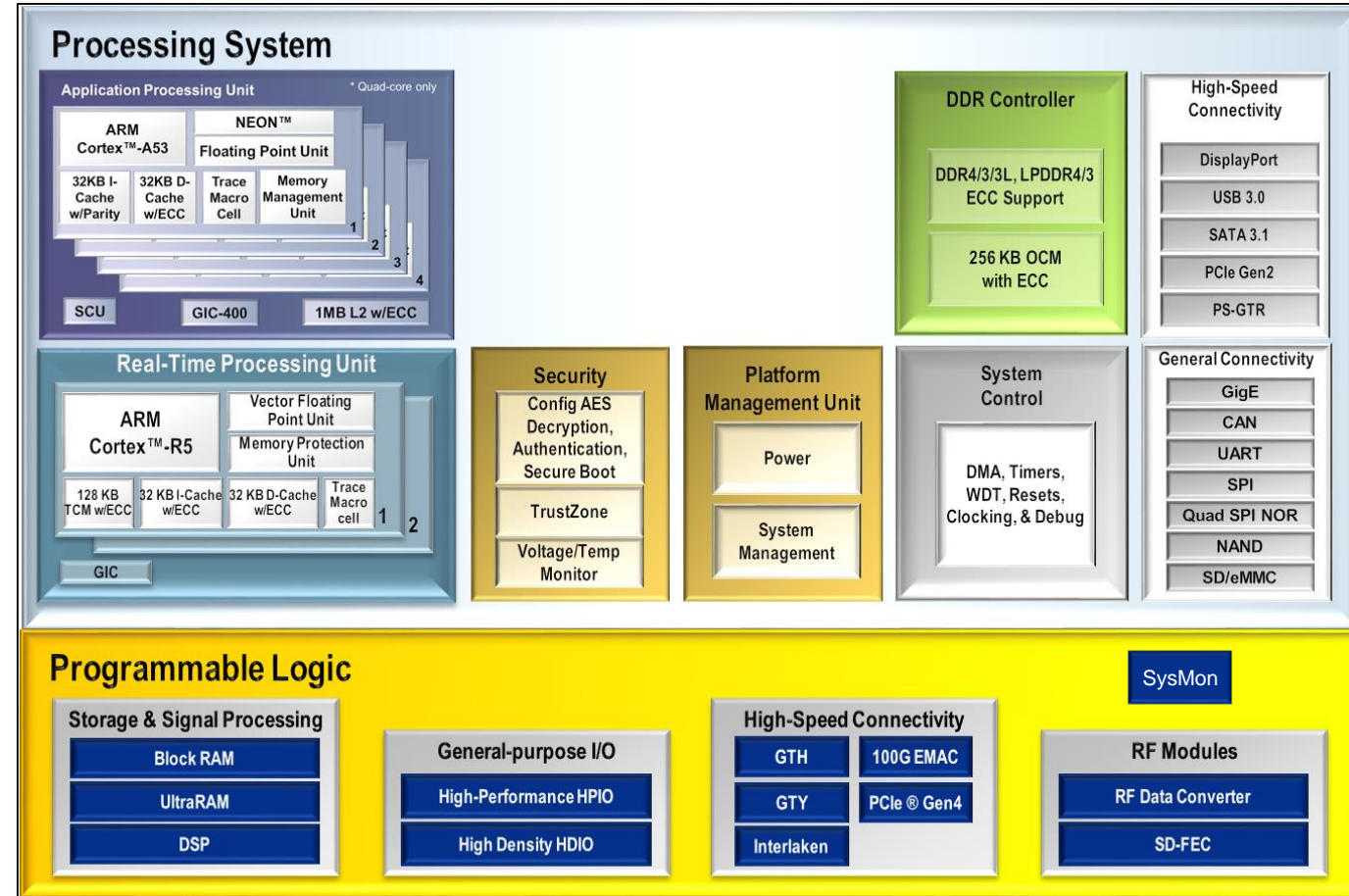
# Drivers & Software





# Zynq UltraScale+ RFSoc Device

- > The Processing System is identical to a ZU+ MPSoC, except:
  - >> No GPU,
  - >> Quad Cortex-A53 APU only (no dual)
  - >> All other PS blocks remain the same
- > A portion of the PL of a ZU+ MPSoC device has been replaced with the SD-FEC, RF-ADC/DAC blocks
- > No change to peripheral interfaces or drivers (I2C, QSPI...)



- Software users coming from a ZU+ design already know how to use the RFSoc PS

# RFSoc Drivers

## > RF-ADC/DAC – rfdc\_v\* (3.2) (PG269 – Appendix C)

- >> Bare-Metal – XSDK build, GitHub, Linux – GitHub (embeddedsd)
- >> Linux and bare-metal APIs are identical
- >> Control plane manipulation, avoiding registers
- >> 77 APIs total (as of 2018.1)

## > SD-FEC – sd\_fec\_v\* (1.0) (PG256 – Appendix C)

- >> Bare-Metal – In the XSDK build, GitHub  
Linux – GitHub (linux-xlnx) – linked from Xilinx linux drivers wiki
- >> Linux and bare-metal APIs differ
- >> Control plane manipulation, data table updates, register manipulation option via API
- >> 7 main bare-metal APIs, plus 84 specialized register/table API calls (as of 2018.1)

## > Three of the four driver combinations use libmetal library

[.../Xilinx/embeddedsw/XilinxProcessorIPLib/drivers](https://github.com/Xilinx/embeddedsw/XilinxProcessorIPLib/drivers)

Driver	Type	Uses Libmetal?
rfdc	Bare-metal	Yes
rfdc	Linux	Yes
sd_fec	Bare-metal	No
sd_fec	Linux	Yes

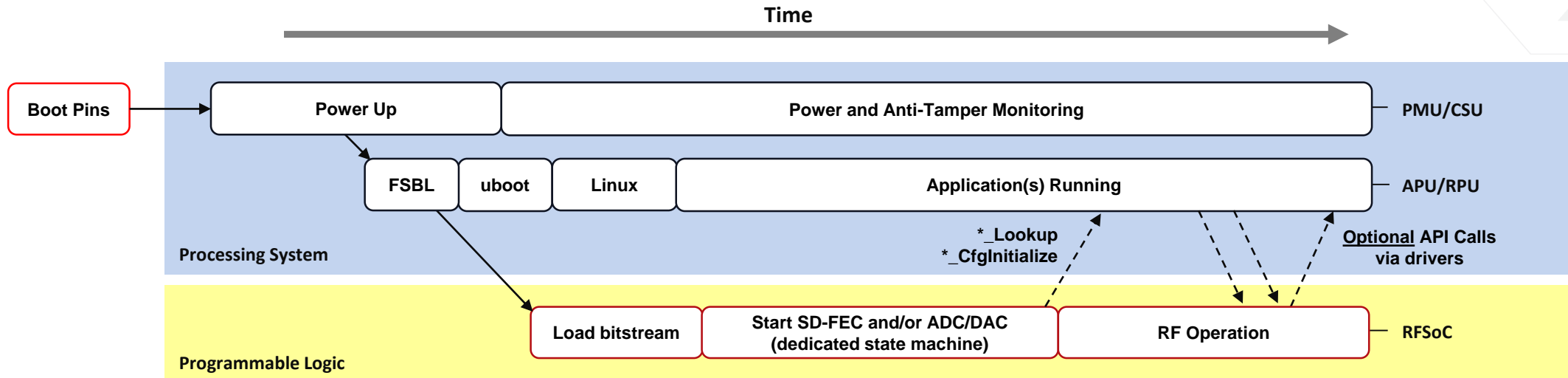
# A Simple RF-ADC/DAC Example Explained

- > Set the RF-ADC/DAC instance
- > Populate the data structures per the initial Vivado settings
- > Two nested loops checking which blocks are enabled
  - >> The first runs through each Tile
  - >> The second runs through each Block within each Tile
- > Modify Mixer Settings from initial configuration
- > Write new Mixer Settings
- > Modify QMC Settings from initial configuration
- > Write new QMC Settings

```
213
214 /* Initialize the RFdc driver. */
215 ConfigPtr = XRFdc_LookupConfig(RFdcDeviceId);
216 /* Initializes the controller */
217 Status = XRFdc_CfgInitialize(RFdcInstPtr, ConfigPtr);
218
219 for (Tile = 0; Tile <4; Tile++) {
220     /* Check for Tile Enabled */
221     for (Block = 0; Block <4; Block++) {
222         /* Check for Block Enabled */
223         if (XRFdc_IsDACBlockEnabled(RFdcInstPtr, Tile, Block)) {
224
225             Status = XRFdc_GetMixerSettings(RFdcInstPtr, XRFDC_DAC_TILE, Tile, Block, &SetMixerSettings);
226             Status = XRFdc_GetQMCSettings(RFdcInstPtr, XRFDC_DAC_TILE, Tile, Block, &SetQMCSettings);
227
228             /* Modify mixer configurations */
229             SetMixerSettings.CoarseMixFreq = 0x01;           // 1 (0 dbV)
230             SetMixerSettings.Freq = 2000;                   // MHz
231             SetMixerSettings.PhaseOffset = 22.56789;        // in Degrees, valid range -180 to 180
232             SetMixerSettings.FineMixerScale = 0x2;         // 0.997 (-0 dbV)
233
234             /* Write Mixer settings */
235             Status = XRFdc_SetMixerSettings(RFdcInstPtr, XRFDC_DAC_TILE, Tile, Block, &SetMixerSettings);
236
237             /* Modify QMC configurations */
238             SetQMCSettings.EnableGain = 1;                 // QMC gain correction enabled
239             SetQMCSettings.EnablePhase = 1;               // QMC phase correction enabled
240             SetQMCSettings.GainCorrectionFactor = 0.95;    // Gain Correction Factor
241             SetQMCSettings.PhaseCorrectionFactor = -5.0;   // Phase Correction Factor
242
243             /* Write QMC settings */
244             Status = XRFdc_SetQMCSettings(RFdcInstPtr, XRFDC_DAC_TILE, Tile, Block, &SetQMCSettings);
245         }
246     }
}
```

*Software changes can have drastic effects on the hardware (example: setting the wrong data rate will generate a FIFO overflow)*

# A Simplified Linux RFSoc Boot Example



- >> The PMU/CSU initialize as in a ZU+
- >> The FSBL (First Stage Boot Loader) loads the bitstream including the SD-FEC and/or RF-ADC/DAC blocks
- >> In parallel the PMU/CSU/APU/RPU finish initialization and the SD-FEC and/or RF-ADC/DAC blocks initialize via on-board state machines (*no user interaction*)
- >> Software access to the IP is **optionally** started through `*_Lookup` then `*_CfgInitialize` API commands
- >> Application code can then **optionally** interact with the SD-FECs or RF-ADC/DAC as needed through APIs
  - *The SD-FECs and/or RF-ADC/DAC initialize and can operate **without** software interaction*

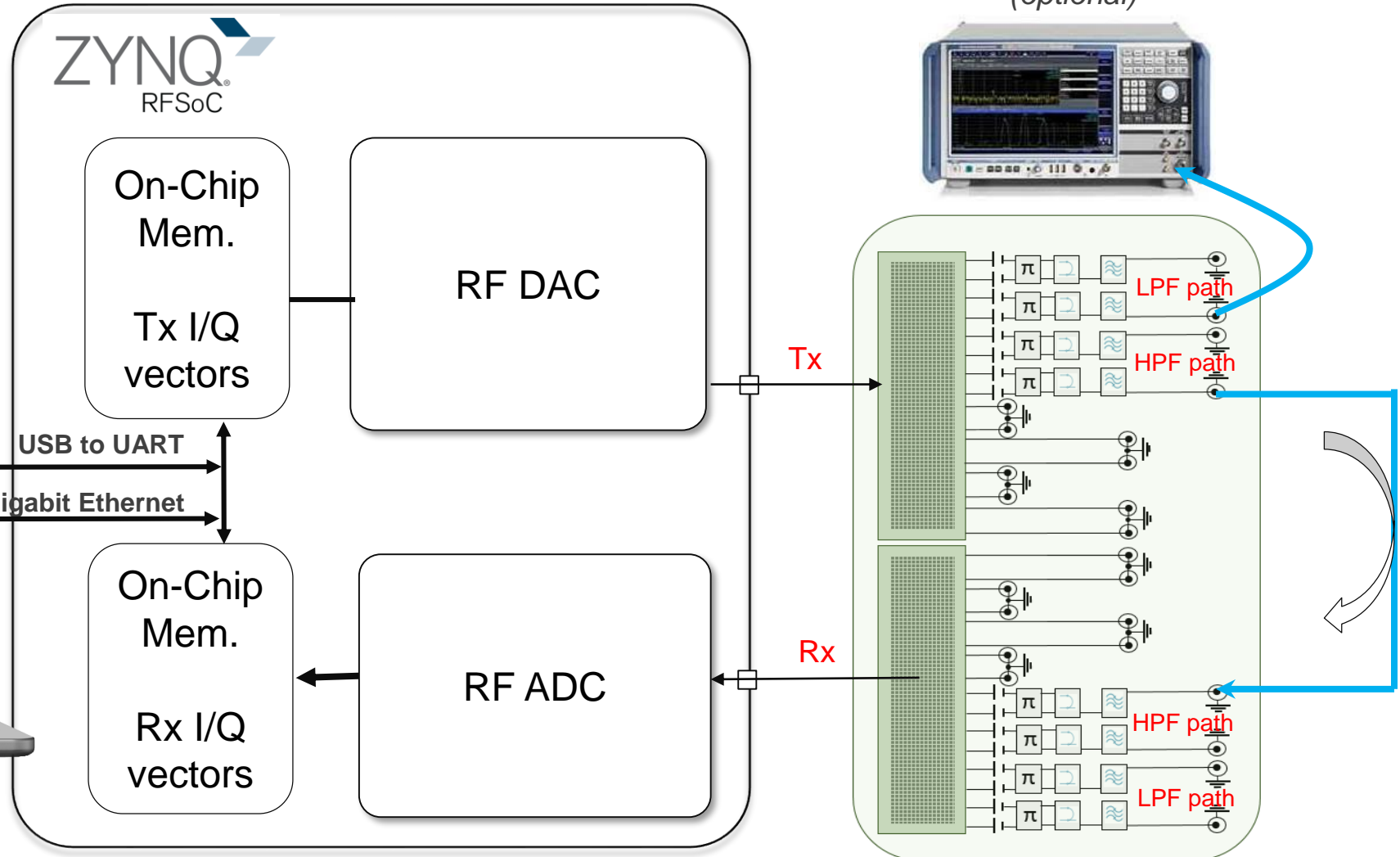
**See The RF Evaluation Tool Demonstration  
During The Break**

# RF Data Converter Evaluation Tool - Overview

DAC Results & Analysis  
(optional)

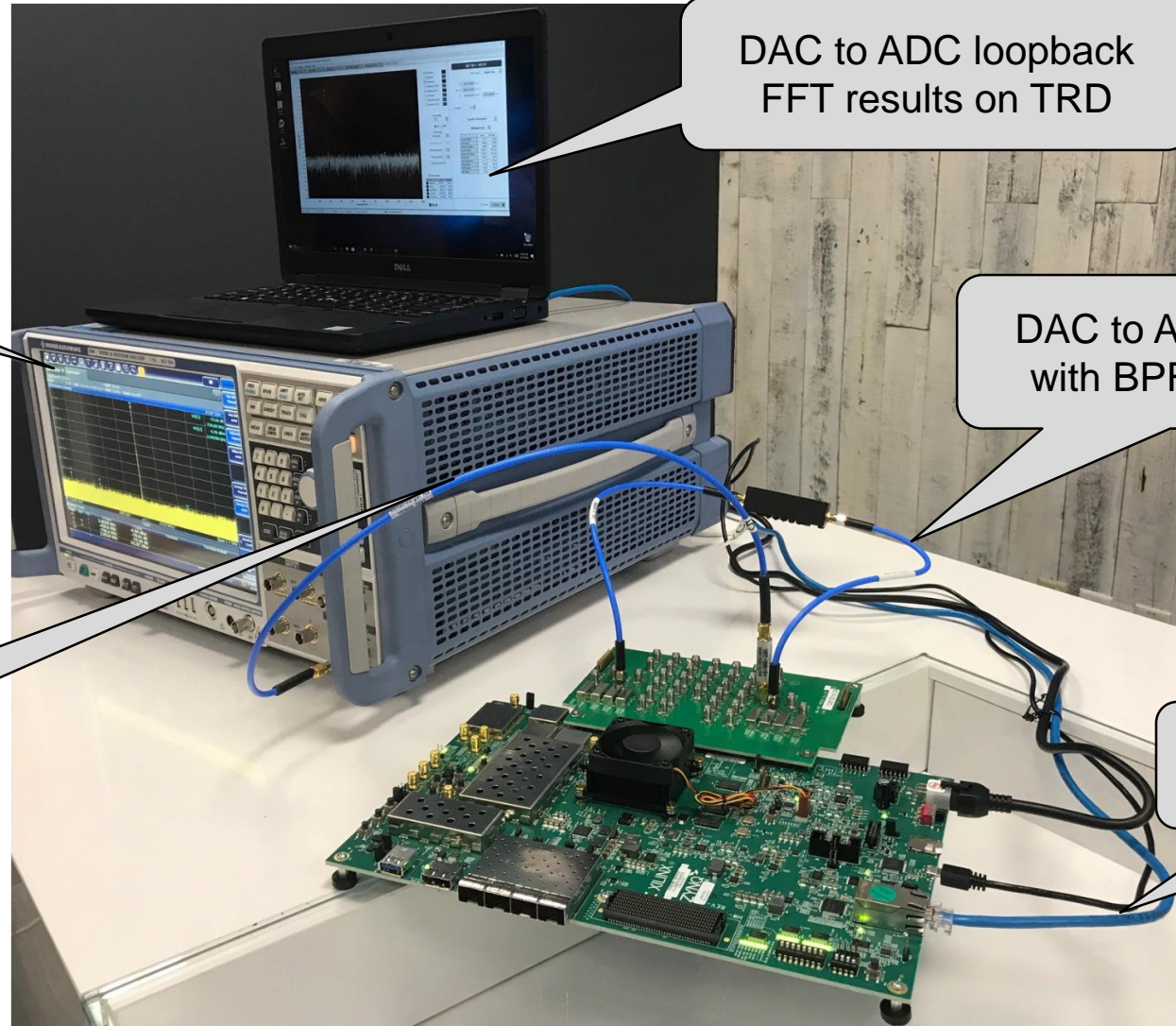


NI Labview GUI  
ADC Results & Analysis  
and  
DAC stimuli building





# Beta RF DC Evaluation Tool Measurement simple set-up



DAC outputs to Spectrum Analyzer (optional)

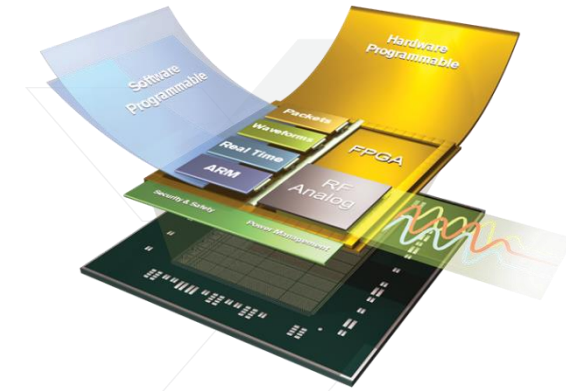
DAC to ADC loopback FFT results on TRD

DAC to ADC loopback with BPF in between

DAC channel output to Spectrum Analyzer

USB / Ethernet cable connected to PC

# Summary



- > **Integrated RF Data Converter Subsystem addresses a wide range of applications**
- > **Significantly reduces the Power and Footprint of high channel count systems**
- > **Enables adaptable Radio HW platforms**
- > **Full support in Vivado accelerates development time versus discrete solutions**
- > **Data Converter Evaluation Board, Design, and Evaluation Tools**



**Adaptable.**  
**Intelligent.**





# RF Solutions with Zynq® UltraScale+™ RFSoc

Presented By

Glenn Steiner  
David Brubaker  
Xilinx

