OPEN-SOURCE HDL/IP AND SOFTWARE COMPONENTS FOR BUILDING LEADING-EDGE SDRS

MICHAEL HENNERICH
ADI FELLOW, EMBEDDED SYSTEM ARCHITECT
SOFTWARE AND SECURITY GROUP

GRCon24
GNU Radio Conference 2024
analog.com

Typical Engineering SDR Design Flow

RESEARCH

- Behavioral simulation
- Device Evaluation
  - Measure: SFDR, SNR, EVM, NF, NSD, etc.
- Streaming data to
  - MATLAB®
  - Python
  - GNU Radio
  - IIO-Scope
  - ACE, etc.

ALGORITHM DEVELOPMENT

- MATLAB/Python/GRC reference implementation
- Hardware streaming

DESIGN ELABORATION

- Modeling
  - MATLAB/Python/GRC
- Hardware streaming

PROTOTYPE

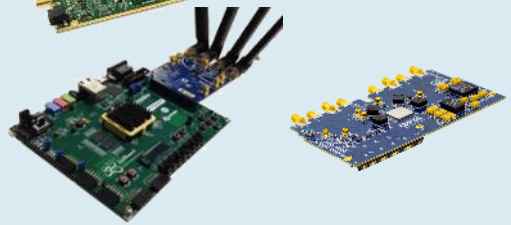
- Deployment to development board
- Design optimization
- HDL integration
- Driver integration

PRODUCTION

- Deployment to custom hardware
- Validation with complete hardware solution

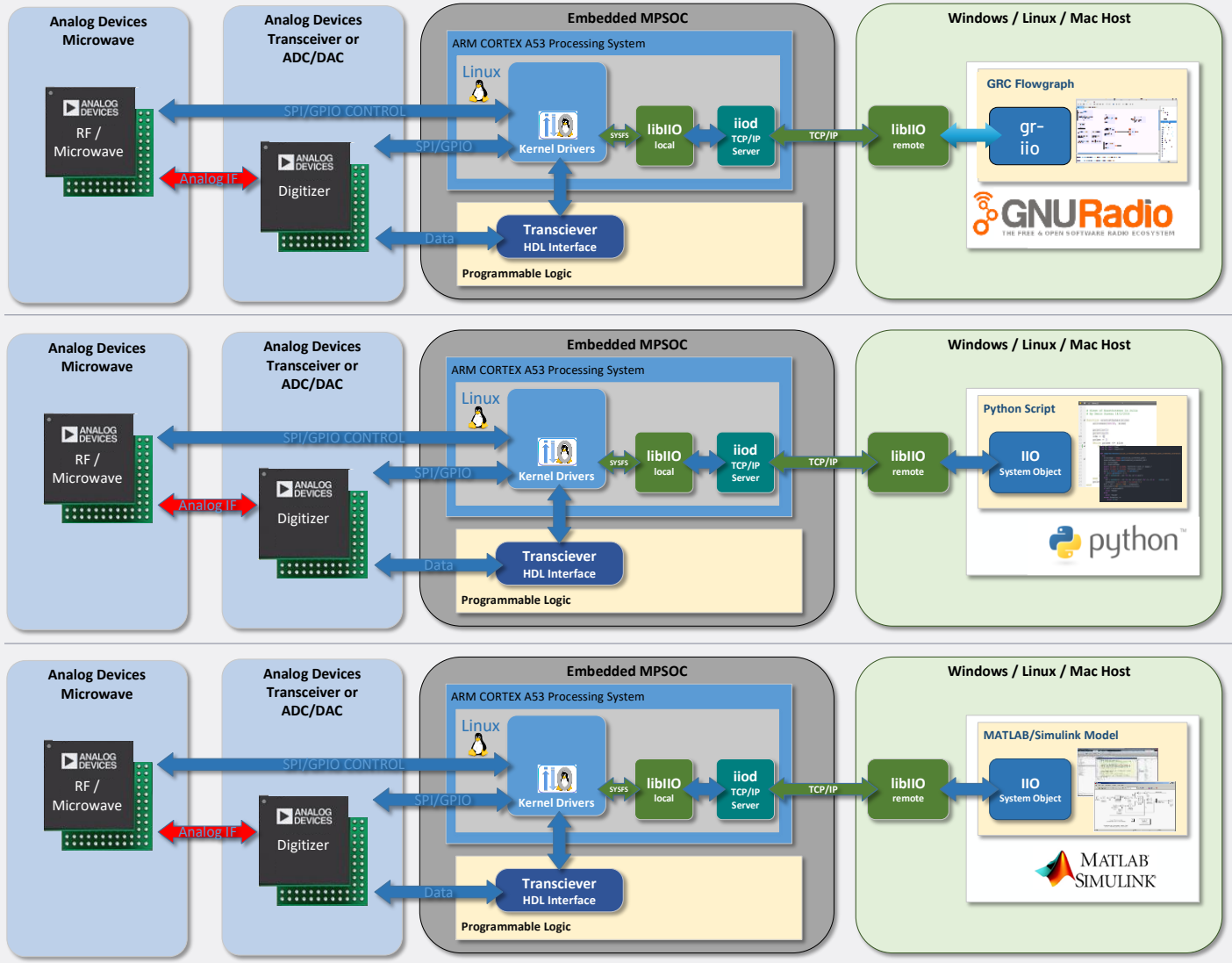
PLUTOSDR/JUPITERSDR/USRP/ETC.

EVALUATION BOARD FMC + FPGA CARRIER OF CHOICE



FULL CUSTOM DESIGN USING SAME HDL/SW/INFRASTRUCTURE

Algorithmic Development, Modeling, Prototyping



SINGLE COHESIVE SOFTWARE SOLUTION: MEETING ENGINEERS IN THEIR ECOSYSTEM OR AT THEIR TOOLS OF CHOICE

Common Architecture Makes It Easy to Transition Between Platforms

SHARES SAME SOFTWARE/HDL/HARDWARE STACK

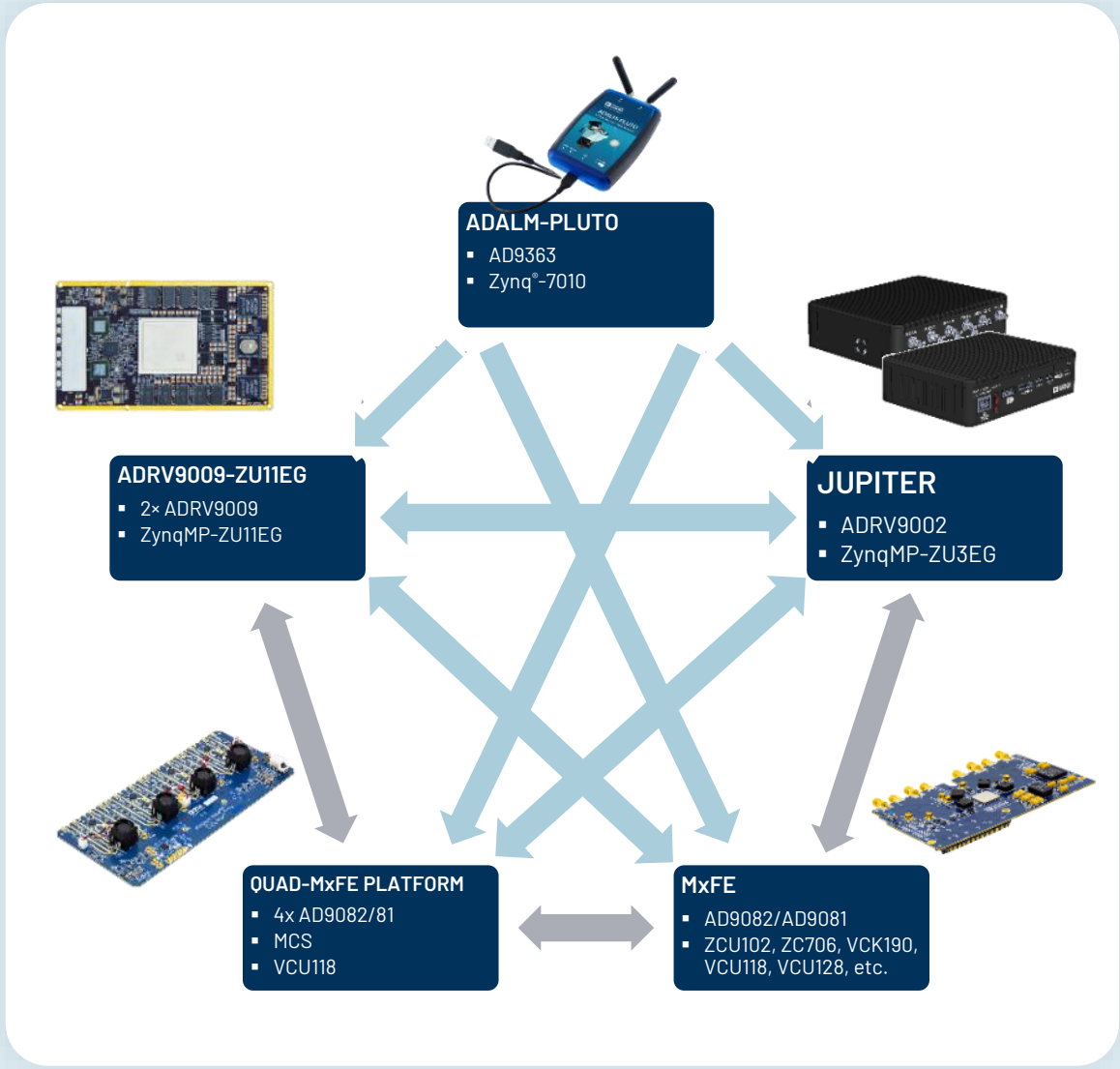
- Makes it easy to move from one to the other
- Differentiated on form factor, number of channels, connectivity, expandability, FPGA resources, CPU resources

START WITH ADALM-PLUTO

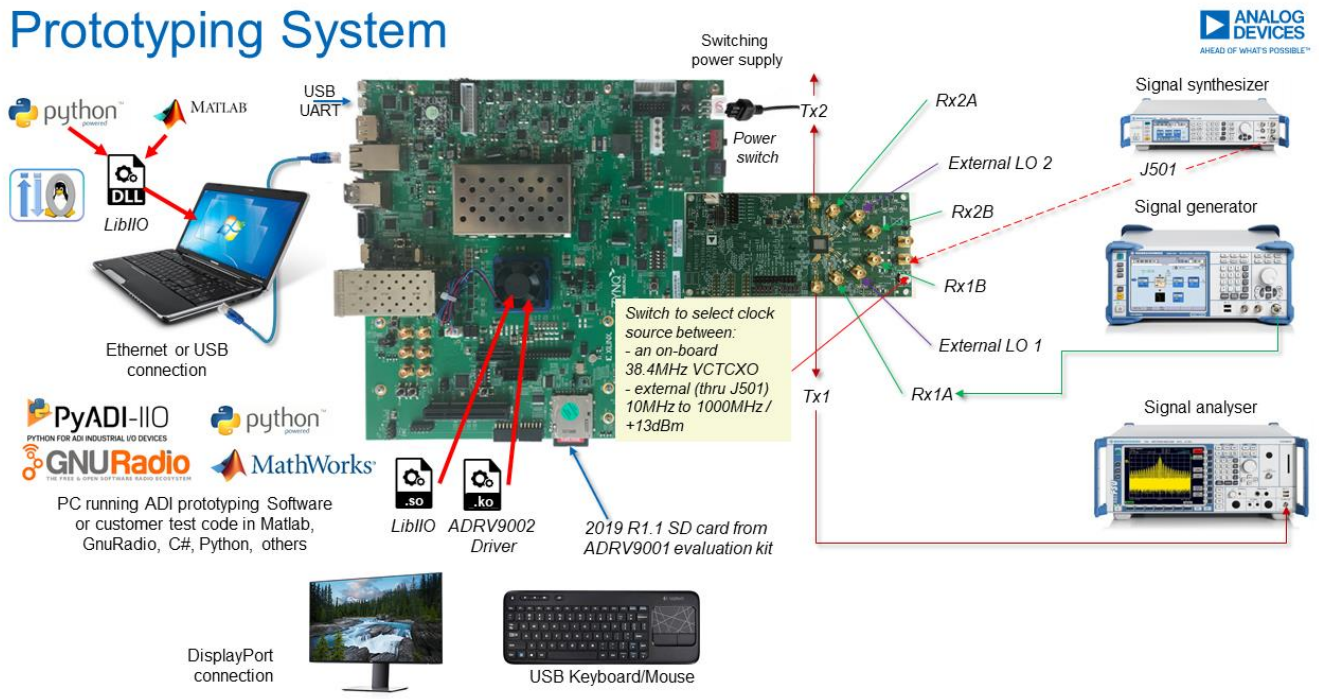
- Stream to MATLAB®, Simulink®, or GNU Radio via USB
- Take data in the field
- Validate your communication, radar, or SIGINT algorithms in MATLAB, Simulink, or GNU Radio
- Start moving to embedded signal processing
  - Transition to production-ready SOM
  - Use custom chip-down design

SAME TOOLS, SAME LIBRARIES, SAME HDL

- Vivado, MATLAB, IIO work the same on all platforms
- Common HDL at github.com/analogdevicesinc/hdl
- Common Linux® kernel at github.com/analogdevicesinc/linux



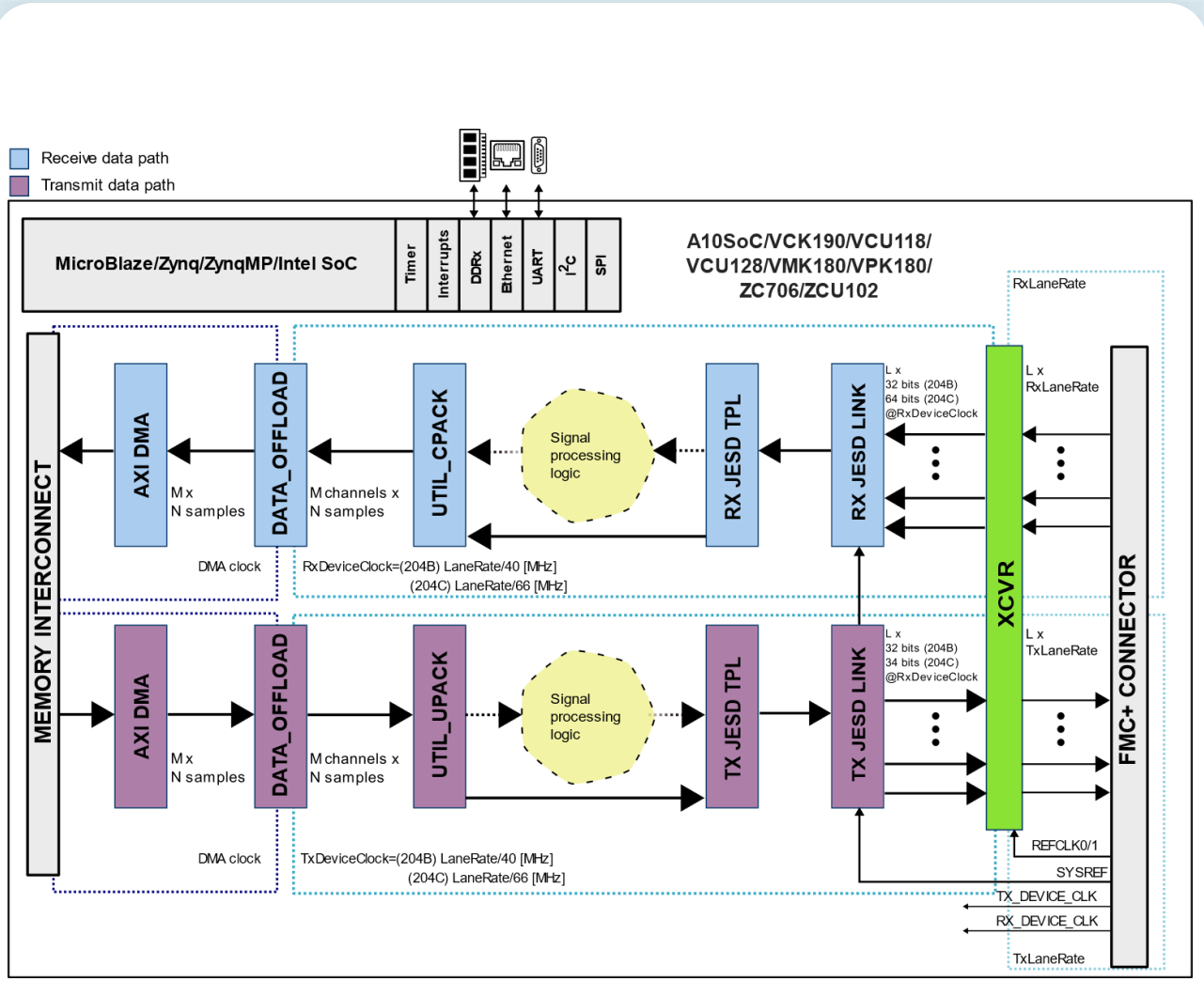
Typical Prototyping System



AMD®/Xilinx®	AC701	KC705	VC707	ZC702	ZC706	ZCU102	Zed	Cora Z7	MicroZed	ADRV9009-EG11-SOM	ADRV936x -SOM	VCK190	KCU105	VCU118	VCU128
Intel®/Altera®	A10SOC	C5SOC	DE10 Nano	A10GX	A5GT	A5SOC									

Full Stack HDL Designs

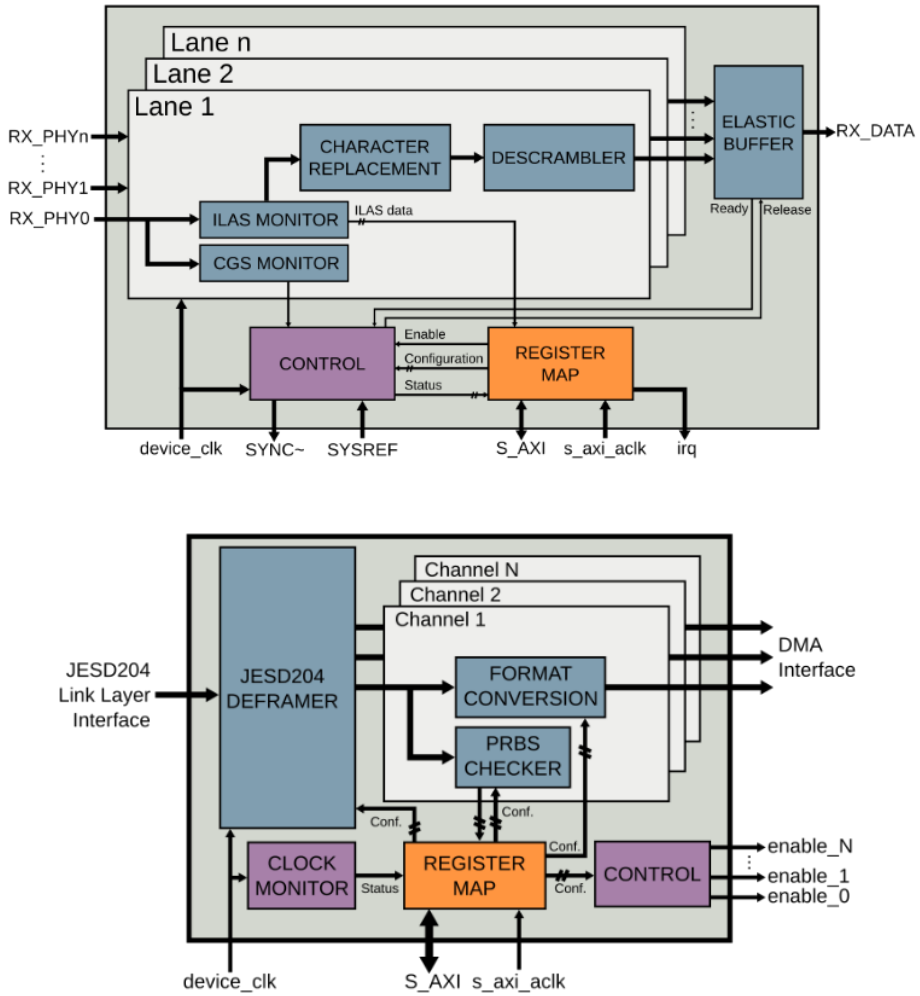
- Common HDL across all IIO reference designs
- Full stack reference designs include JESD204 physical layers (XCVR), link layers, and transport layers.
  - Termination to DMA via an AXI-STREAM or FIFO interface
  - Runs across different Intel® and AMD® Xilinx® carriers
- Are designed to be disconnected to “insert custom signal processing”
  - Your modem, your signals intelligence, etc.
- The example designs show how to use mux in different places in the design, to stream debug data (I/Q samples, or payload [data buffers]) as IIO streams.
  - Works with industry standard debug tools:
  - AMD Xilinx Integrated Logic Analyzer
  - Intel Signal Tap
  - MathWorks® HDL Verifier (in SoC Blockset)



FPGA HDL Cores

We design and support interface logic for various ADI IC devices, encapsulating any and all obscure designs, to provide a common, easy to understand, and consistent interface that is portable across commonly available FPGA devices and standard cell libraries.

- Full stack IP cores to manage physical, link and transport layers, using standard interfaces (AXI, FIFO)
- Portable IP cores for Intel®/Altera® Quartus and AMD®/Xilinx® Vivado
- Open source, documentation and support github.com/analogdevicesinc/hdl
- Verilog HDL source files, Vivado and Intel constraint files
- Updated and maintained to the latest versions of the tools
- Tested and proven on various hardware platforms
- Modular design, easy to use, scale, debug, and customize



IP Library

EXCERPT OF THE HDL IP LIBRARY

ADC/DAC/ Transceiver Specific Parallel LVDS/CMOS

- AD7668
- ADRV9001
- AD9265
- AD9361
- AD9265

ADC/DAC Specific SPI Engine

- AD7616-1
- AD40xx
- AD463x
- AD738x
- AD4134

JESD204 All ADI JESD204 Devices

- AD9081/MxFE®
- ADRV9009
- ADRV9026
- ADRV904x

Video

- AXI_HDMI_RX – ADV7611
- AXI_HDMI_TX – ADV7511

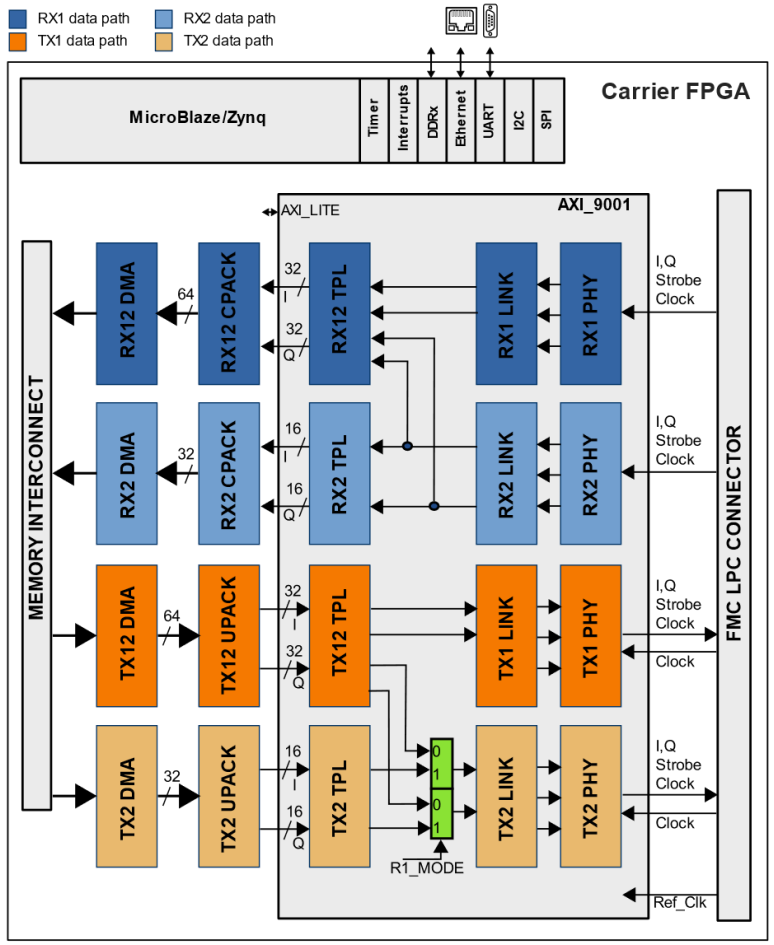
System Utilities

- AXI_FAN_CONTROL
- AXI_CLKGEN
- AXI_DMAC
- OFFLOAD FIFO
- UTIL_PACK/UPACK

IC Specific IP

- In many cases, there is an IC-specific IP
- These IPs are always (Linux and bare metal) software enabled
- For JESD204 interfaces, there is a framework consisting of multiple IPs and software drivers that can be configured either at runtime or at synthesis to interface a specific part
- Both specific IPs and JESD204 interfaces can be split into three logical parts:
  - Physical Layer
  - Data Link Layer
  - Transport Layer

EXAMPLE: IC-specific IP (ADRV9002)



JESD204 Layers

LAYERS COMMUNICATE VIA WELL DEFINED INTERFACES

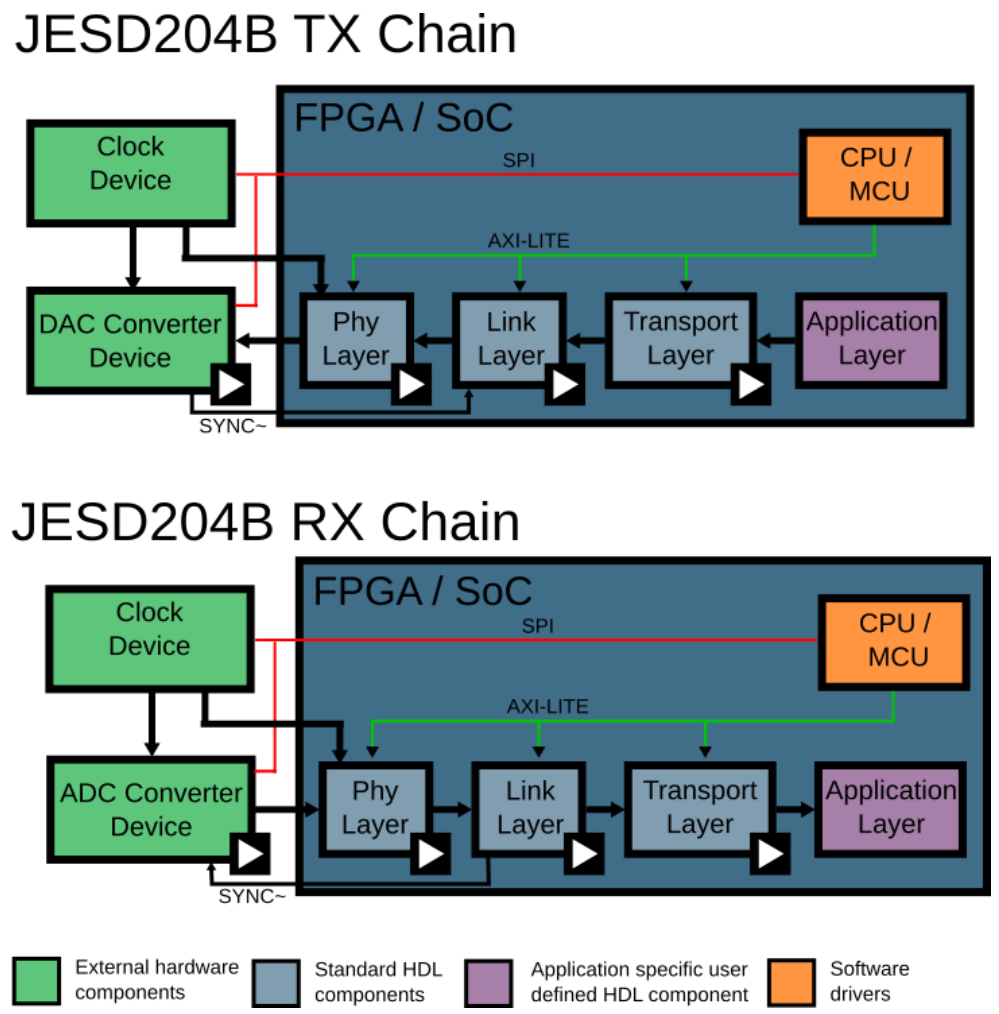
PHY layer
- FPGA vendor and family specific high speed transceiver

Link layer
- Generic JESD204 link layer processing core

Transport layer
- Converter specific

Application layer
- Reference design by ADI
- Replaceable with customer custom logic

Software drivers
- Control various layers and devices
- Diagnostics and heuristics to ensure proper operation



JESD204 Interface Framework

System-level integrated HDL and software framework covering the whole stack

- Hardware: Reference and rapid prototyping systems
- HDL: Components for JESD204 protocol handling
- Software: Drivers to manage clock-chips, converters, and HDL IP Cores

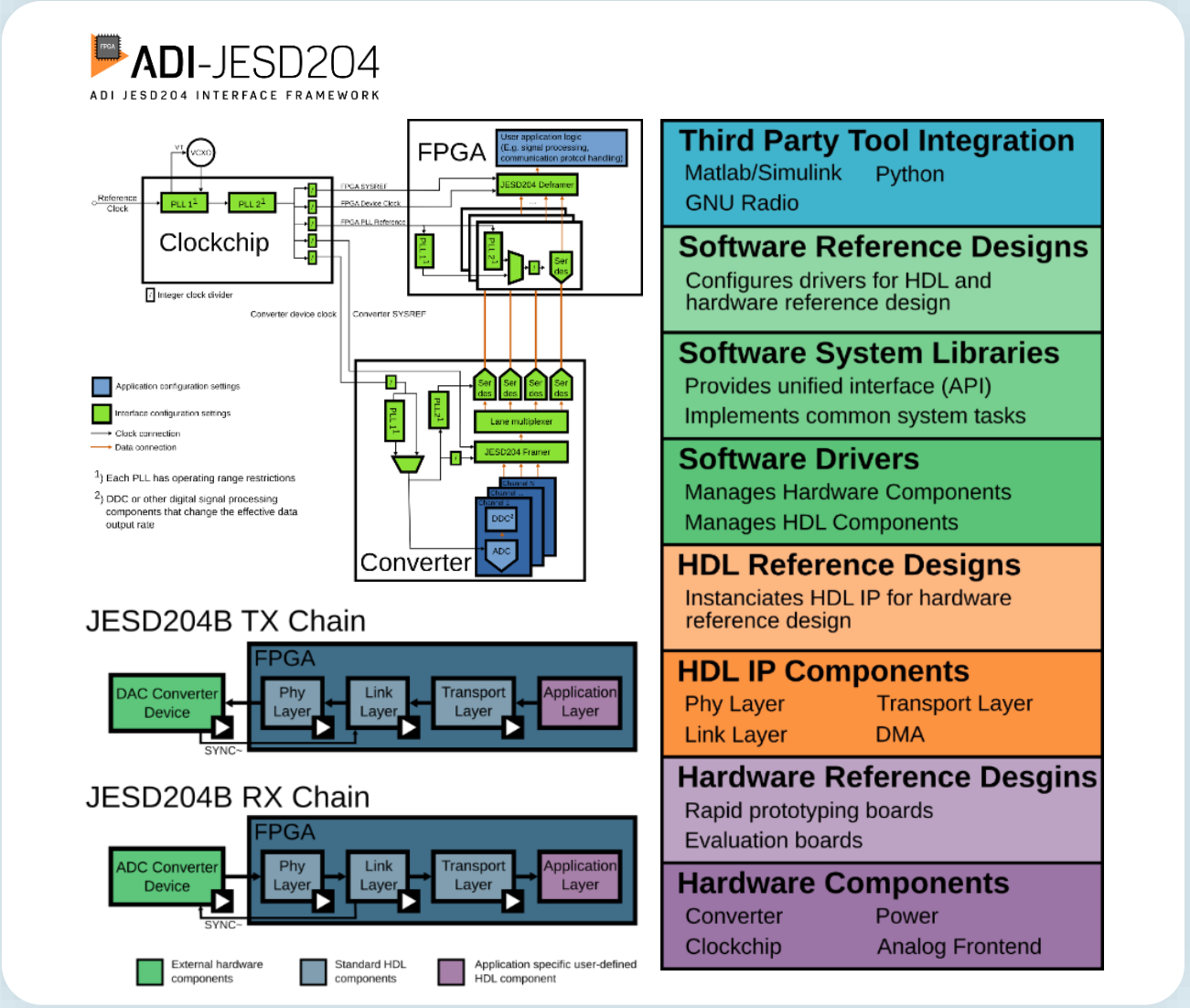
Components have been co-designed for improved interoperability

Key features

- Automatic interface configuration based on application settings
  - High-level API
- Dynamic re-configuration
- Improved diagnostics

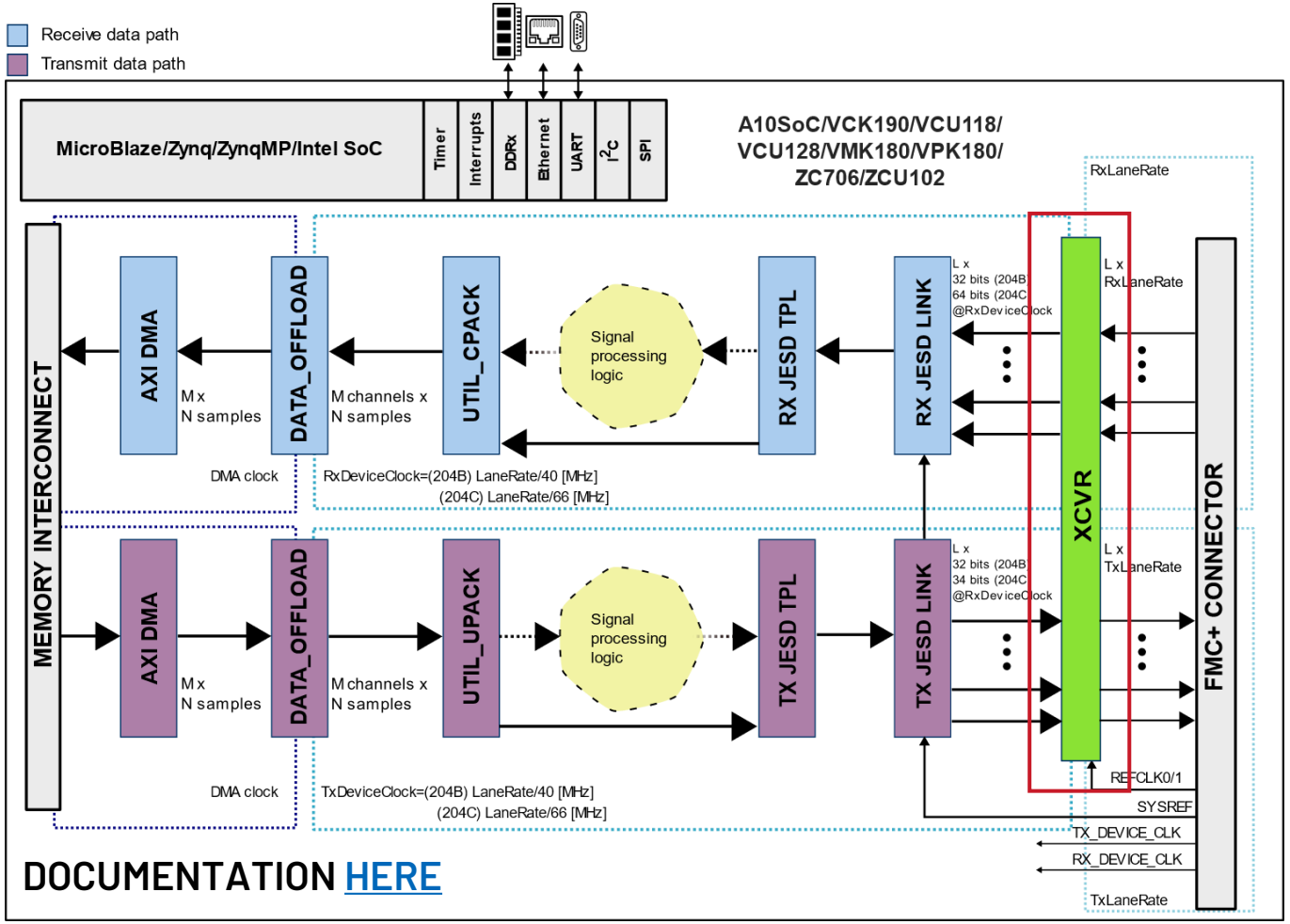
ADI provides full stack reference designs

- Works out of the box
- Starting point for development of custom designs



HDL/IP CORES BY LAYERS

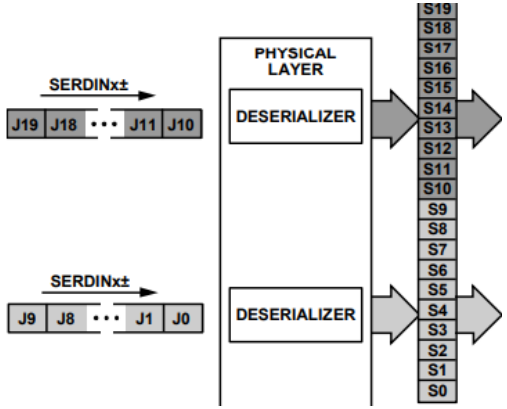
Physical Layer



Physical Layer

OVERVIEW

- Its primary purpose is to take care of transfers of bits to and from the data converter to the FPGA
- It is used to control and configure low level FPGA primitives
- Handles the physical medium and serialization/deserialization
  - High speed transceivers and high speed PLLs
  - Clock and data recovery
- Very FPGA specific



ADI_ADXCVR

- Supports Intel® and Xilinx® devices.
- Software can access the core's registers through an AXI4 Lite Memory Mapped interface.
- Link reset and monitor for Intel and Xilinx.
- Reconfiguration interface control with broadcast capability for Xilinx.
- Access to the statistical eye scan interface of the PHY (Xilinx).

204B/C Transceiver Support

AMD®/XILINX®

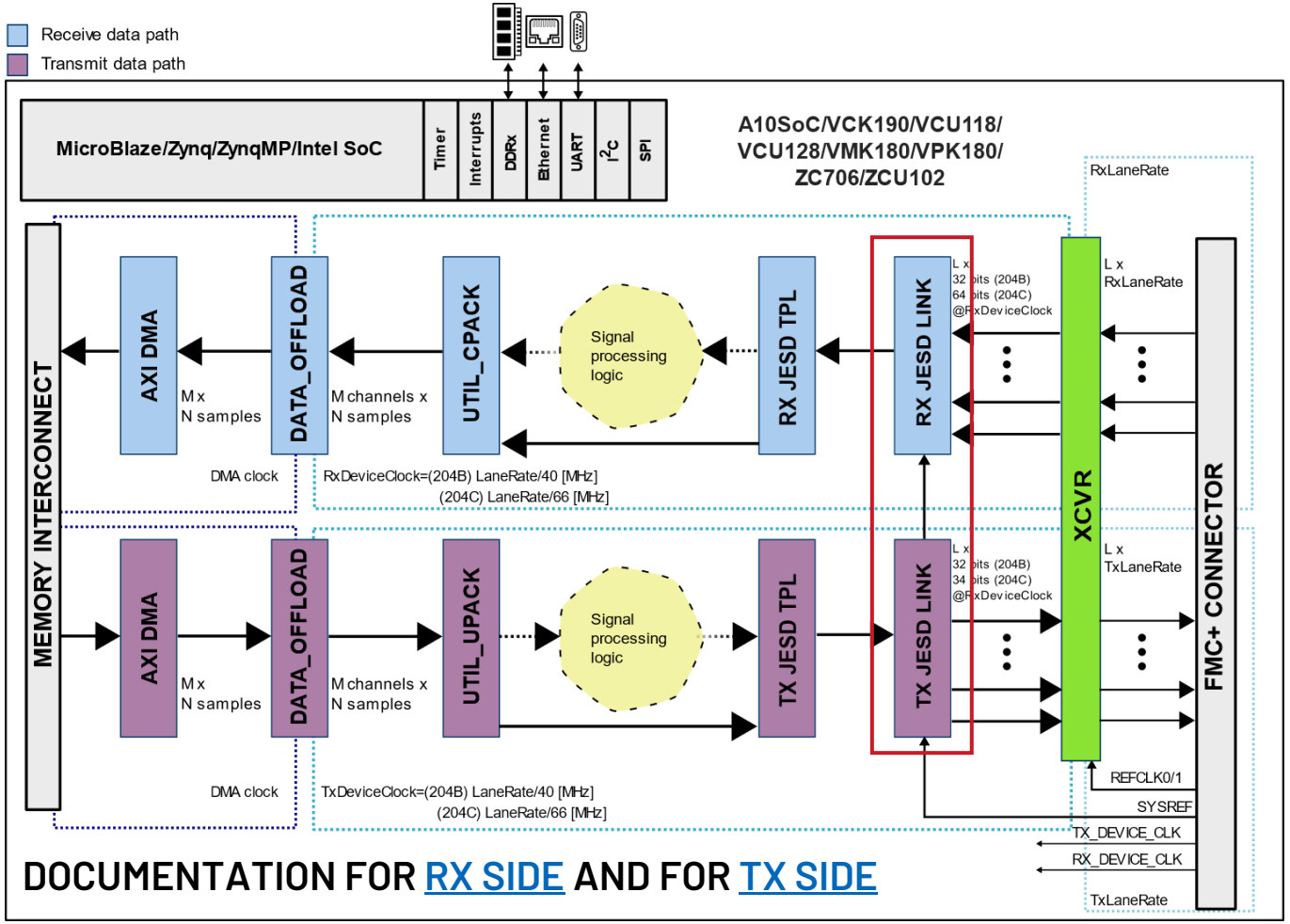
- Supported transceivers
  - GTX2: Kintex 7, Virtex 7, Zynq (204B only)
  - GTH3: Kintex Ultrascale, Virtex Ultrascale
  - GTH4: Kintex Ultrascale, Virtex UltraScale, Zynq MP
  - GTY4: Virtex UltraScale+
  - Versal using Xilinx PHY
- Dynamic reconfiguration through DRP interface
- Digital probabilistic eye-scan support
- PRBS generation/checking

INTEL®/ALTERA® (204B MODE)

- Supported transceivers
  - Arria 10 (SoC/GX)
  - Stratix 10
    - L/H tile
  - Agilex
    - F-tile
- Dynamic reconfiguration support

WIKI:
wiki.analog.com/resources/fpga/docs/axi_adxcvr
wiki.analog.com/resources/tools-software/linux-drivers/jesd204/axi_adxcvr

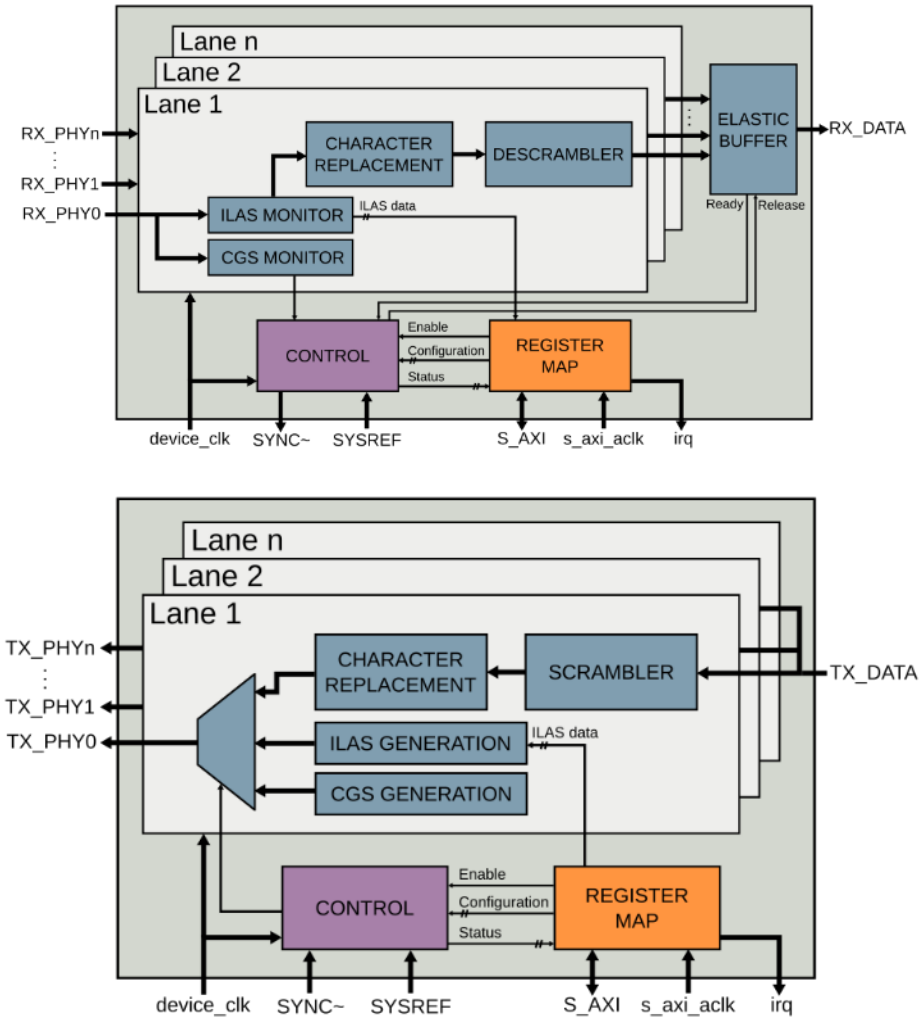
Data Link Layer



DOCUMENTATION FOR RX SIDE AND FOR TX SIDE

Data Link Layer

- The data link layer is specific to JESD204 communication and implements 8B/10B or 64B/66B modes
- It implements support for synchronization of multiple lanes and multiple devices
- Has some diagnostics capabilities
- Has run-time reconfiguration support

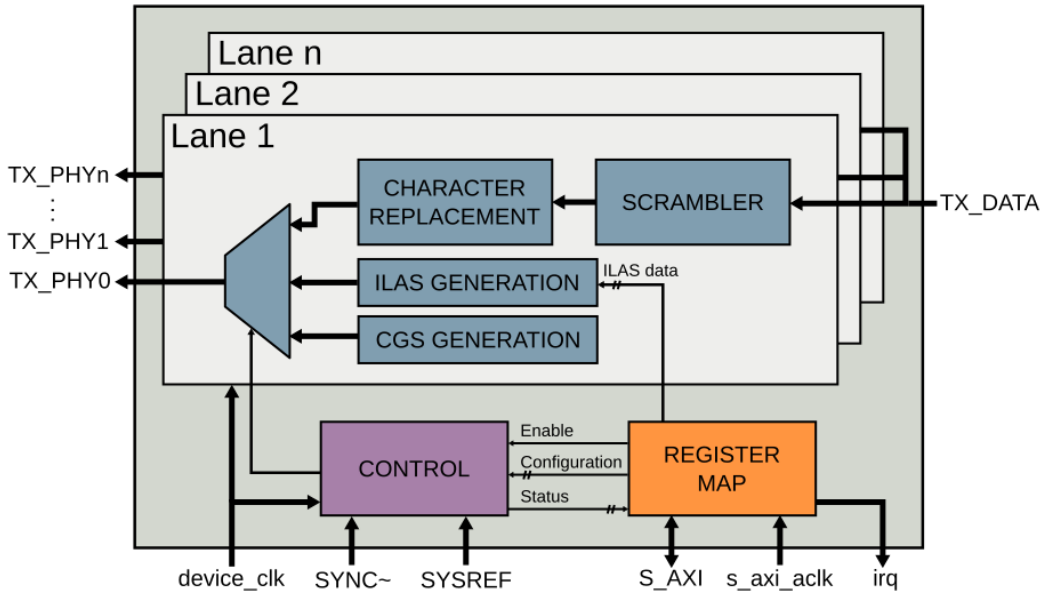


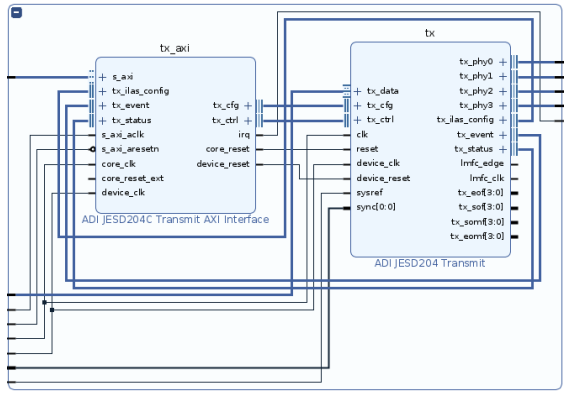
204B/C Transmit Link Layer IP

FEATURES

- Subclass 0 and Subclass 1 support
- Deterministic latency
- Diagnostics
- Fully run-time reconfigurable through memory-mapped register interface (AXI4-Lite)
- Multilink support
- Interrupts for event notification
- Low latency
- Up to 32 lanes
- Gearbox (N' = 12 or F > 4)

WIKI/DOCS:
wiki.analog.com/resources/fpga/peripherals/jesd204/axi_jesd204_tx
analogdevicesinc.github.io/hdl/library/jesd204/axi_jesd204_tx/index.html



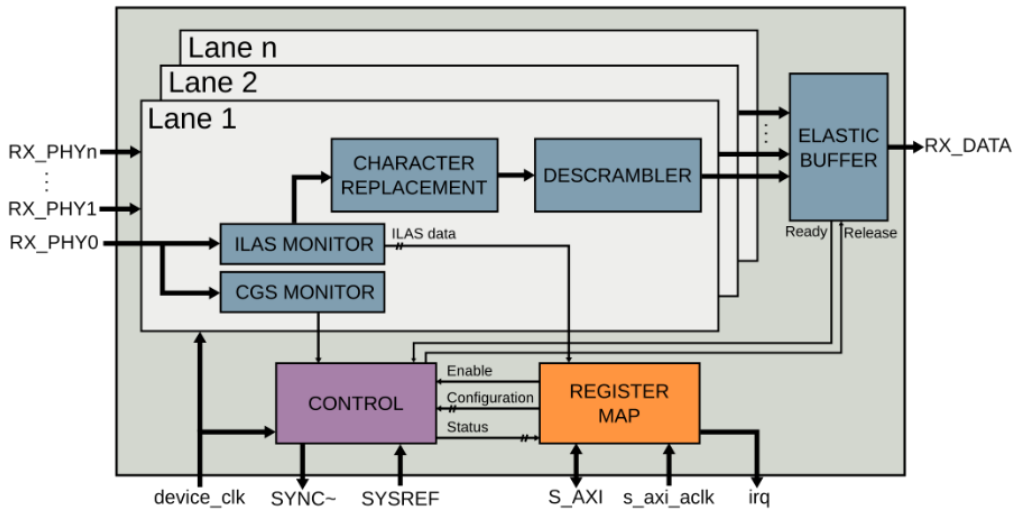


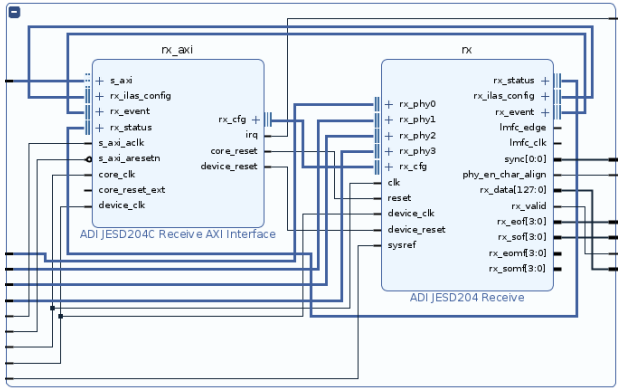
204B/C Transmit Link Layer IP

FEATURES

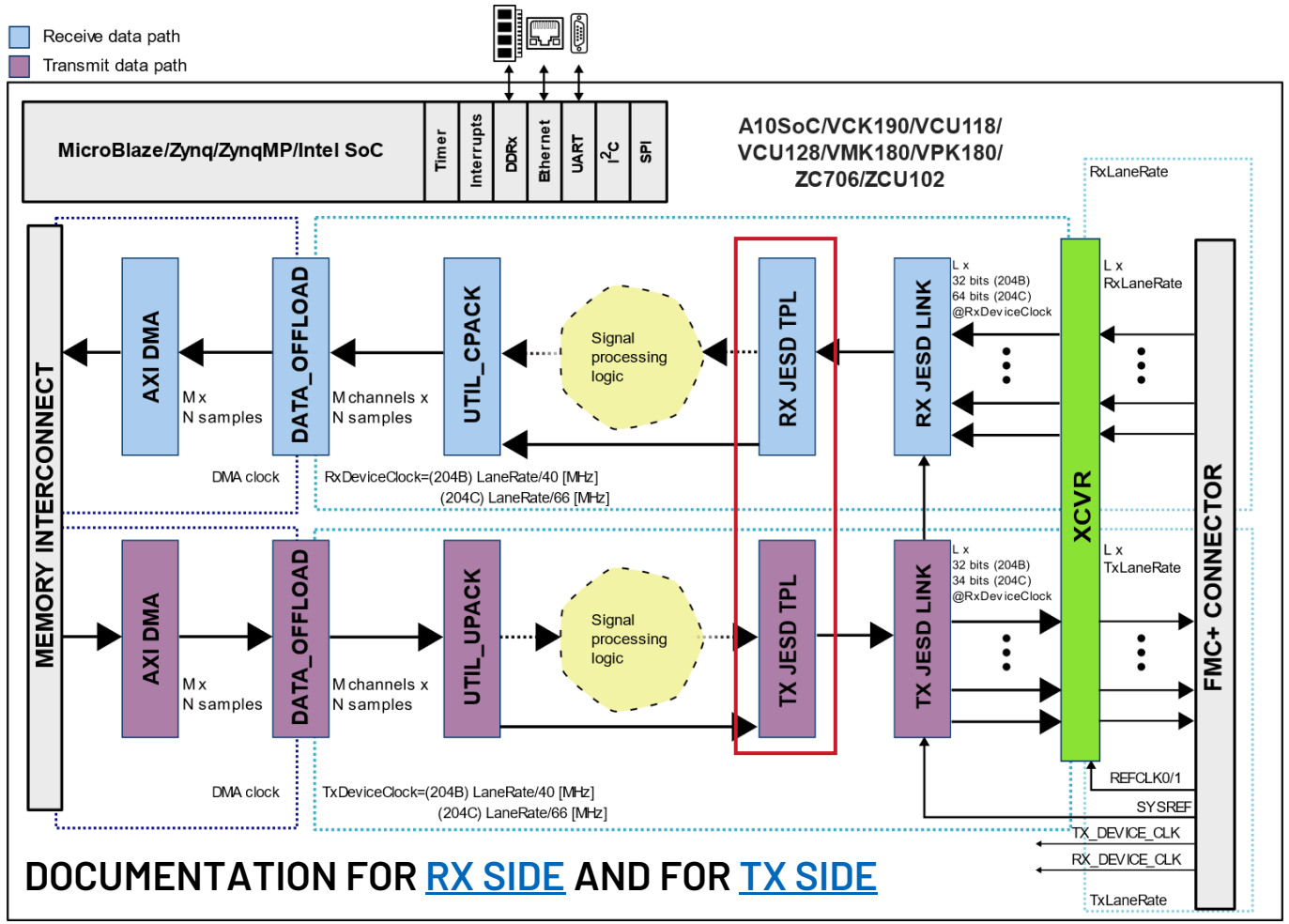
- Subclass 0 and Subclass 1 support
- Deterministic latency
- Diagnostics
- Fully run-time reconfigurable through memory-mapped register interface (AXI4-Lite)
- Multilink support
- Interrupts for event notification
- Low latency
- Up to 32 lanes
- Frame alignment monitor and auto link-restart option

WIKI/DOCS:
wiki.analog.com/resources/fpga/peripherals/jesd204/axi_jesd204_rx
analogdevicesinc.github.io/hdl/library/jesd204/axi_jesd204_rx/index.html



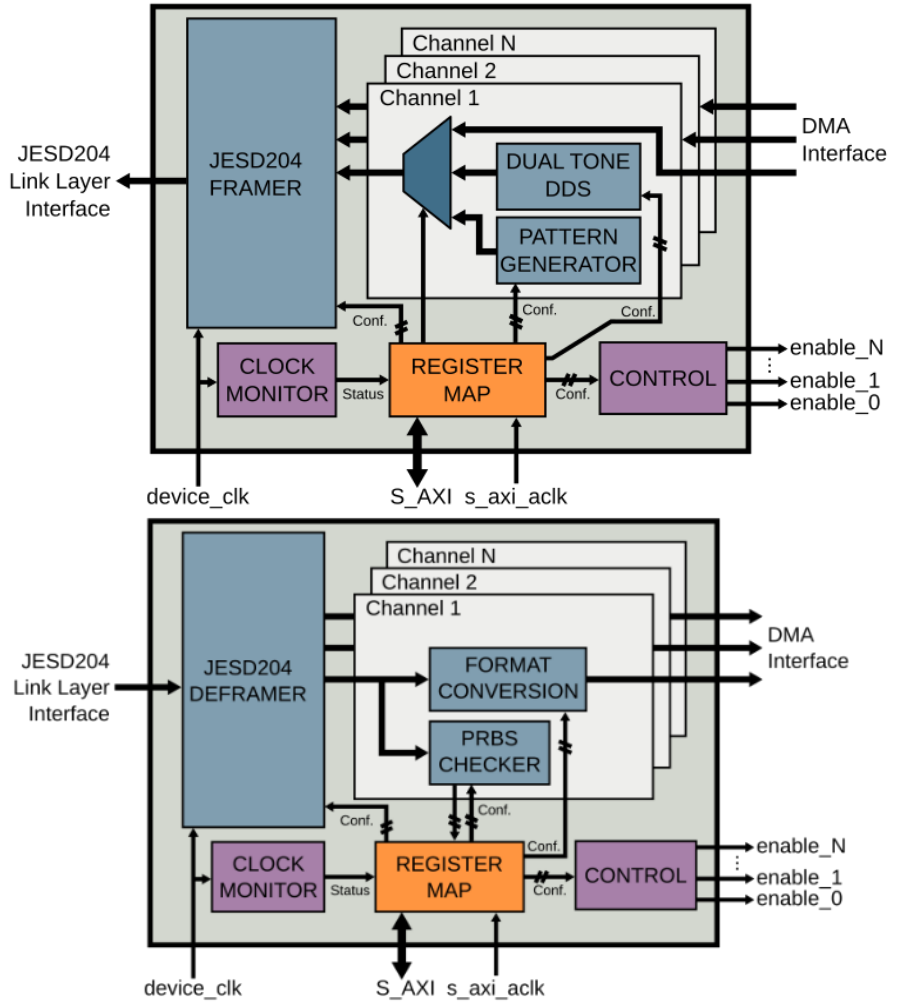


Transport Layer



Transport Layer

- The transport layer takes the data from the physical layer or data link layer and splits it in channel specific streams
- The streams may include one or more samples on the same clock
- Depending on the specific part, it may implement PRBS monitoring or generation, per-channel data formatting, DDS, and pattern generation



DAC Transport Layer IP

AD-IP-JESD204-TPL-DAC

- Generic transport layer core to support all standalone DACs

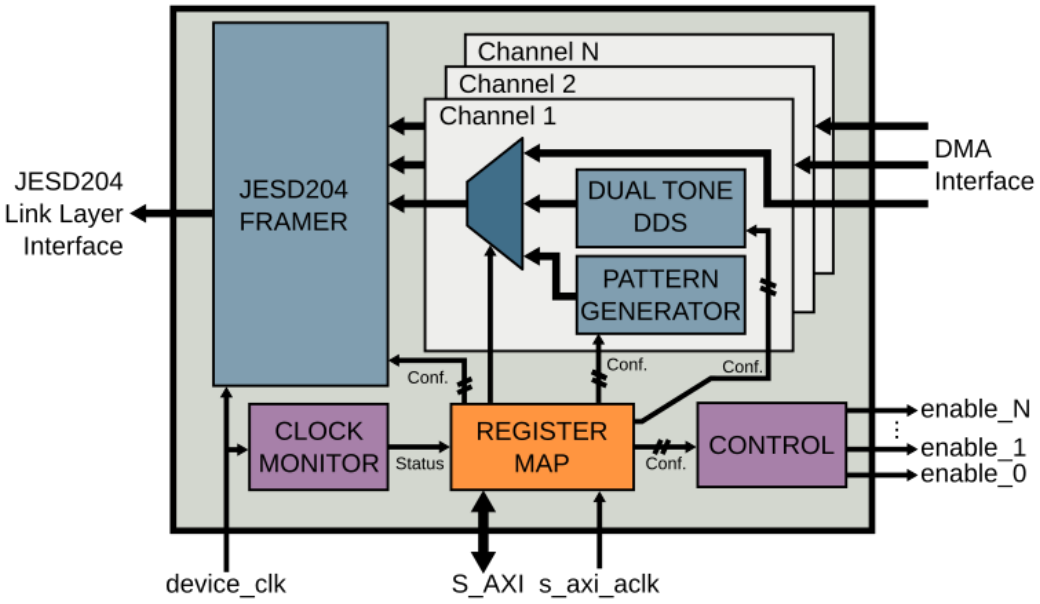
Converts sample data to raw data for the link layer

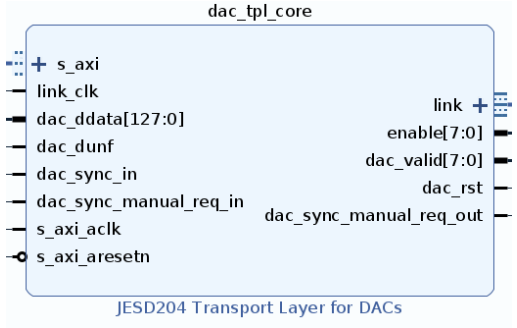
- Implements JESD204 framer
- Contiguous stream for all modes

Opt-in features

- PRBS generator
- Per-channel dual tone DDS
- Per-channel data formatting (sign-extension, two's complement to offset binary conversion)

WIKI/DOCS:
wiki.analog.com/resources/fpga/peripherals/jesd204/jesd204_tpl_dac
analogdevicesinc.github.io/hdl/library/jesd204/ad_ip_jesd204_tpl_dac/index.html





ADC Transport Layer IP

AD-IP-JESD204-TPL-ADC

- Generic transport layer core to support all standalone ADCs

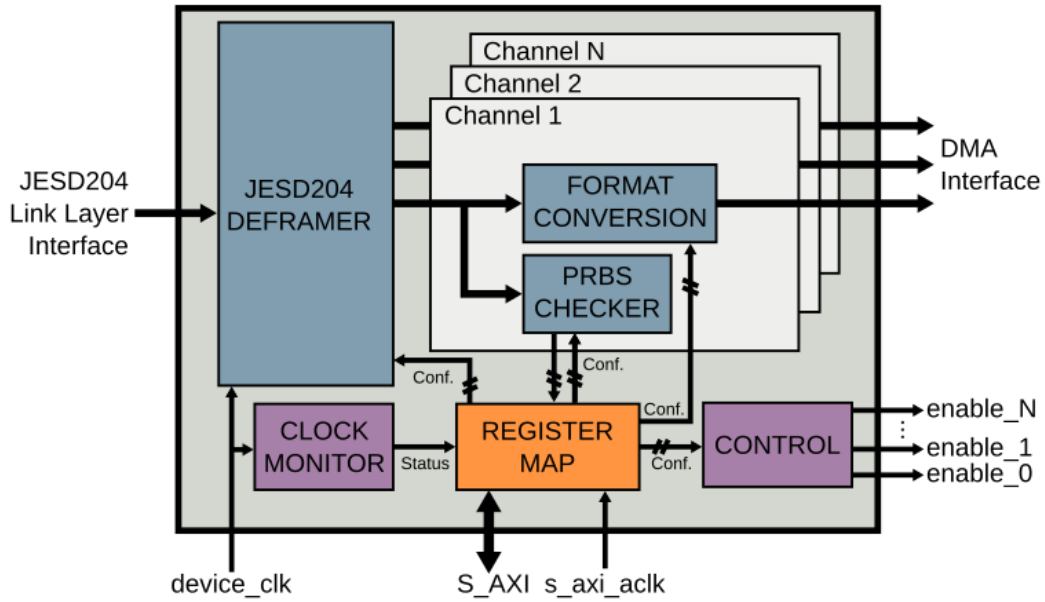
Converts raw data of the link layer to sample data

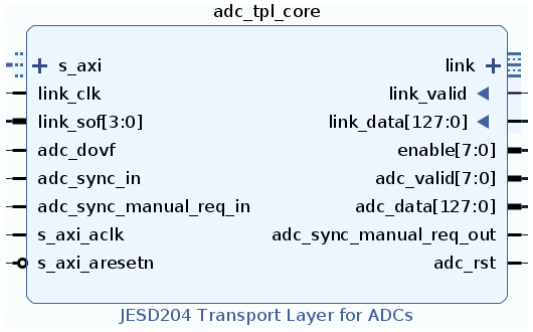
- Implements JESD204 de-framer
- Contiguous stream for all modes

Opt-in features

- PRBS checker
- Per-channel data formatting (sign-extension, two's complement to offset binary conversion)

WIKI/DOCS:
wiki.analog.com/resources/fpga/peripherals/jesd204/jesd204_tpl_dac
analogdevicesinc.github.io/hdl/library/jesd204/ad_ip_jesd204_tpl_dac/index.html

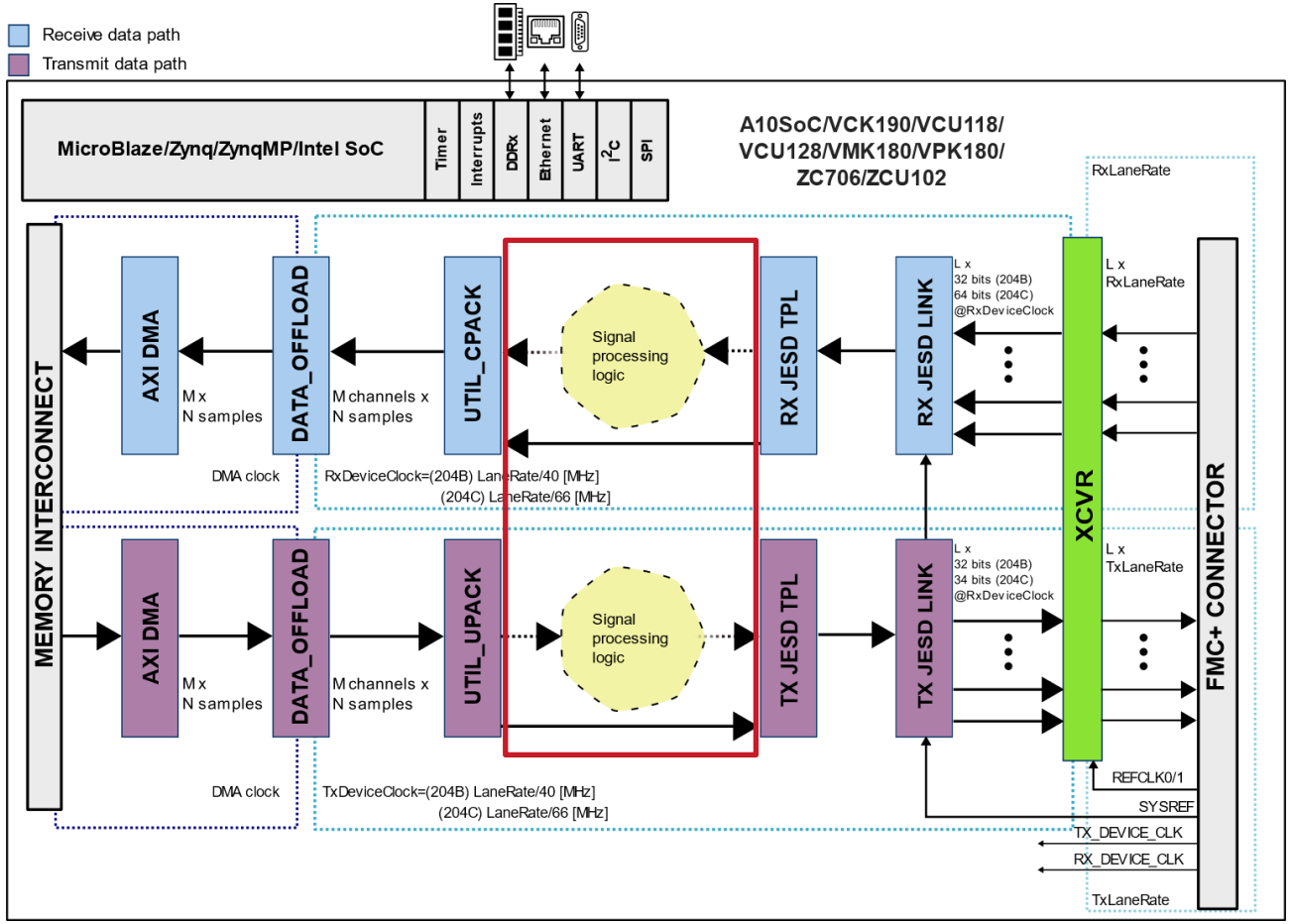




Signal Processing Logic

PLACEHOLDER FOR INSERTING CUSTOM SIGNAL PROCESSING LOGIC

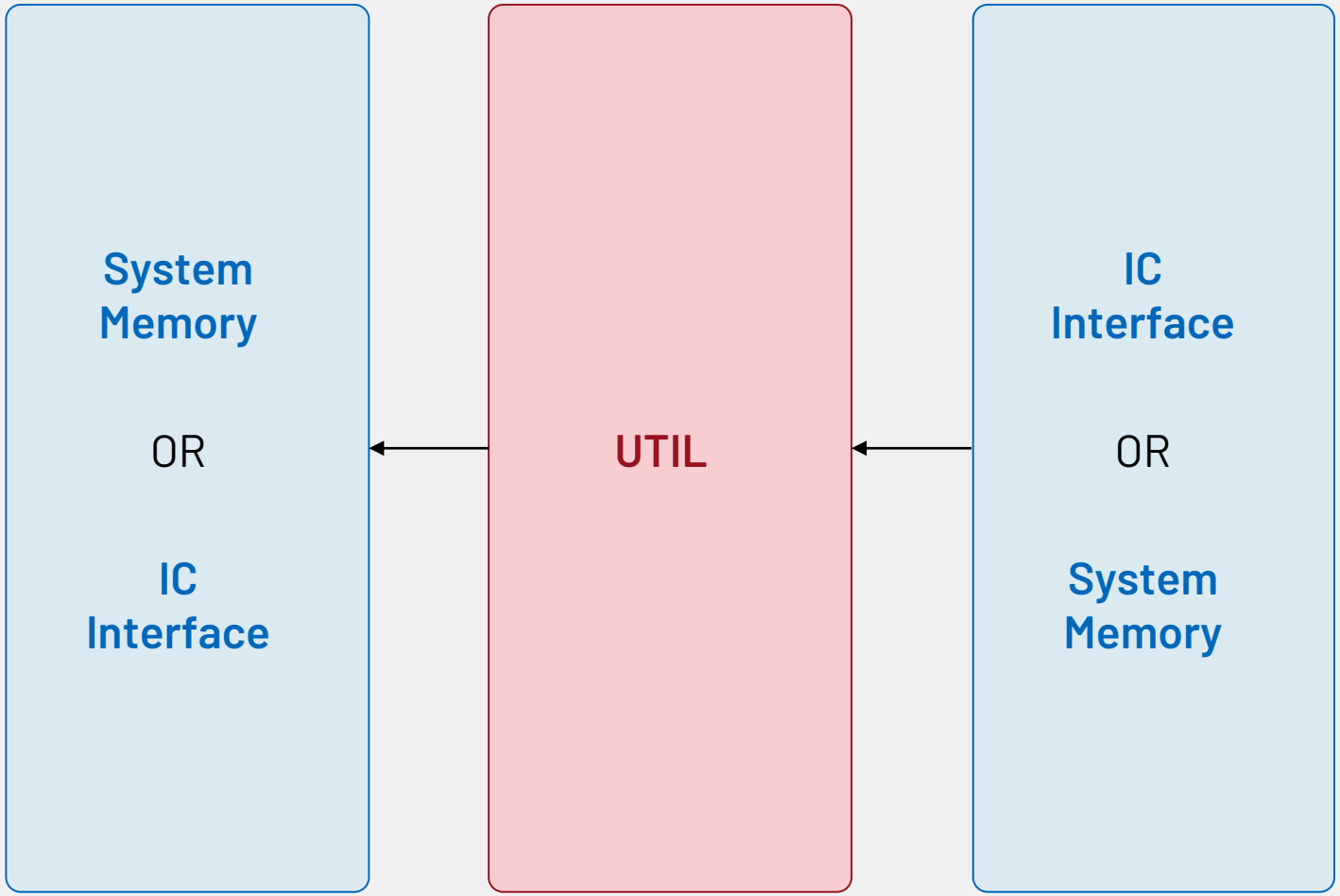
- FIFO like interface (Data and Valid)
- If per-channel operation is required, it can be moved after the PACK IPs
- In some systems, communication with system memory is not required, as the data is sourced or sinked in HDL



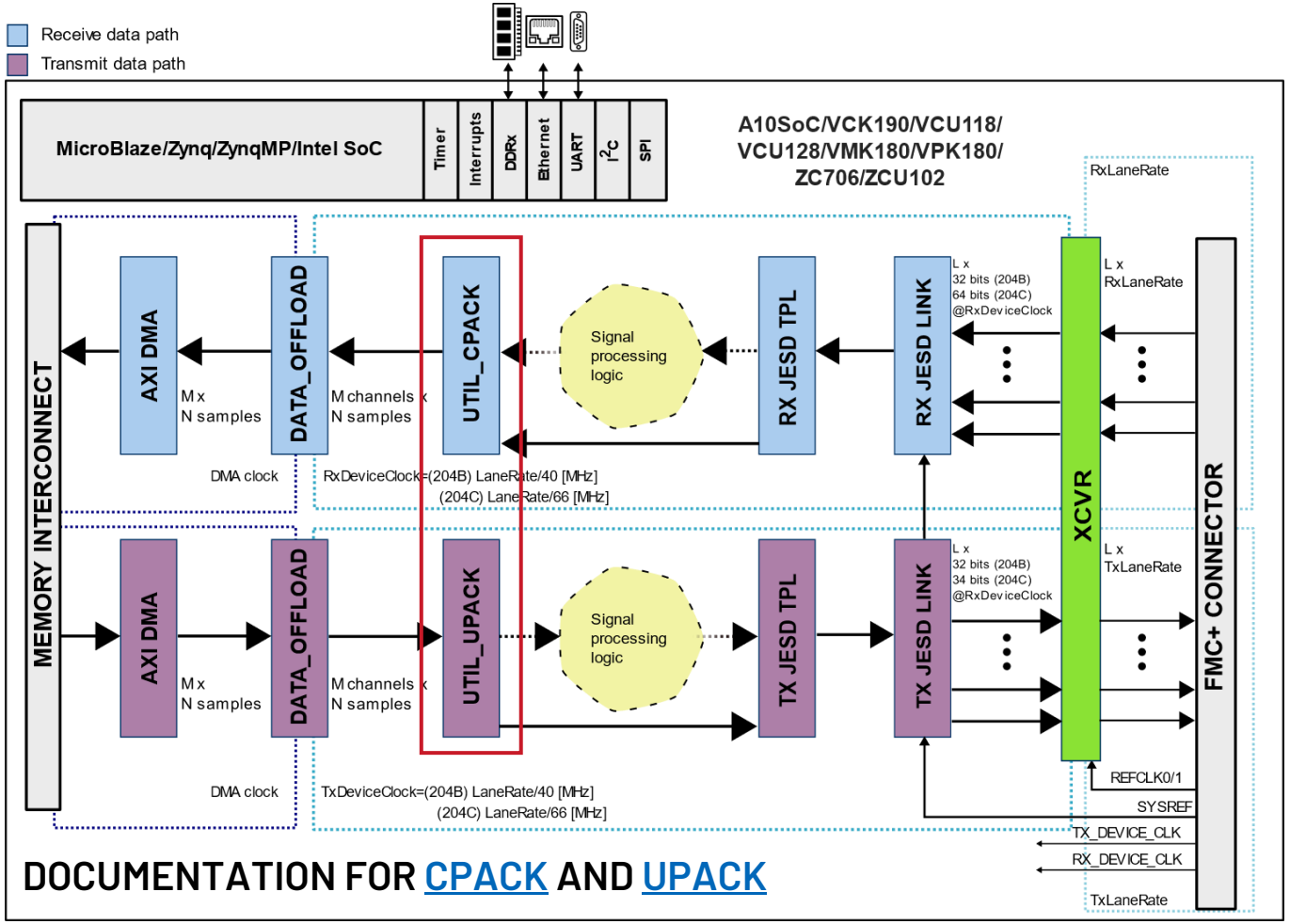
System Utilities

To have a full embedded system that can handle the requirements for SDR, we need more than the IC specific IPs

These IPs allow system-level data transfers that keep up with the bandwidth requirements

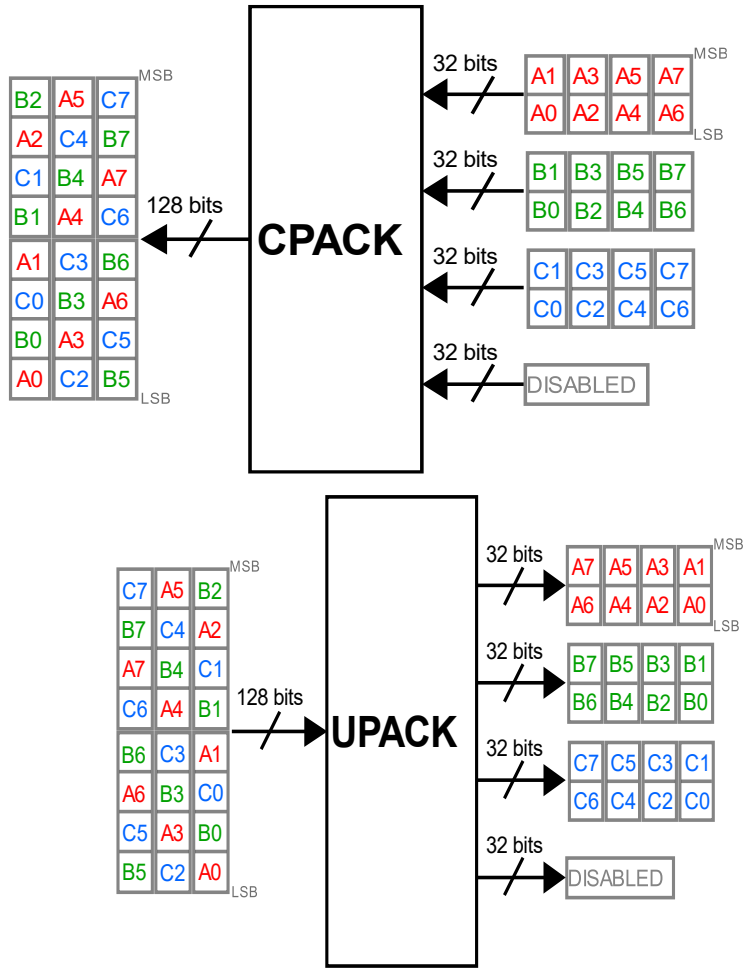


CPACK/UPACK



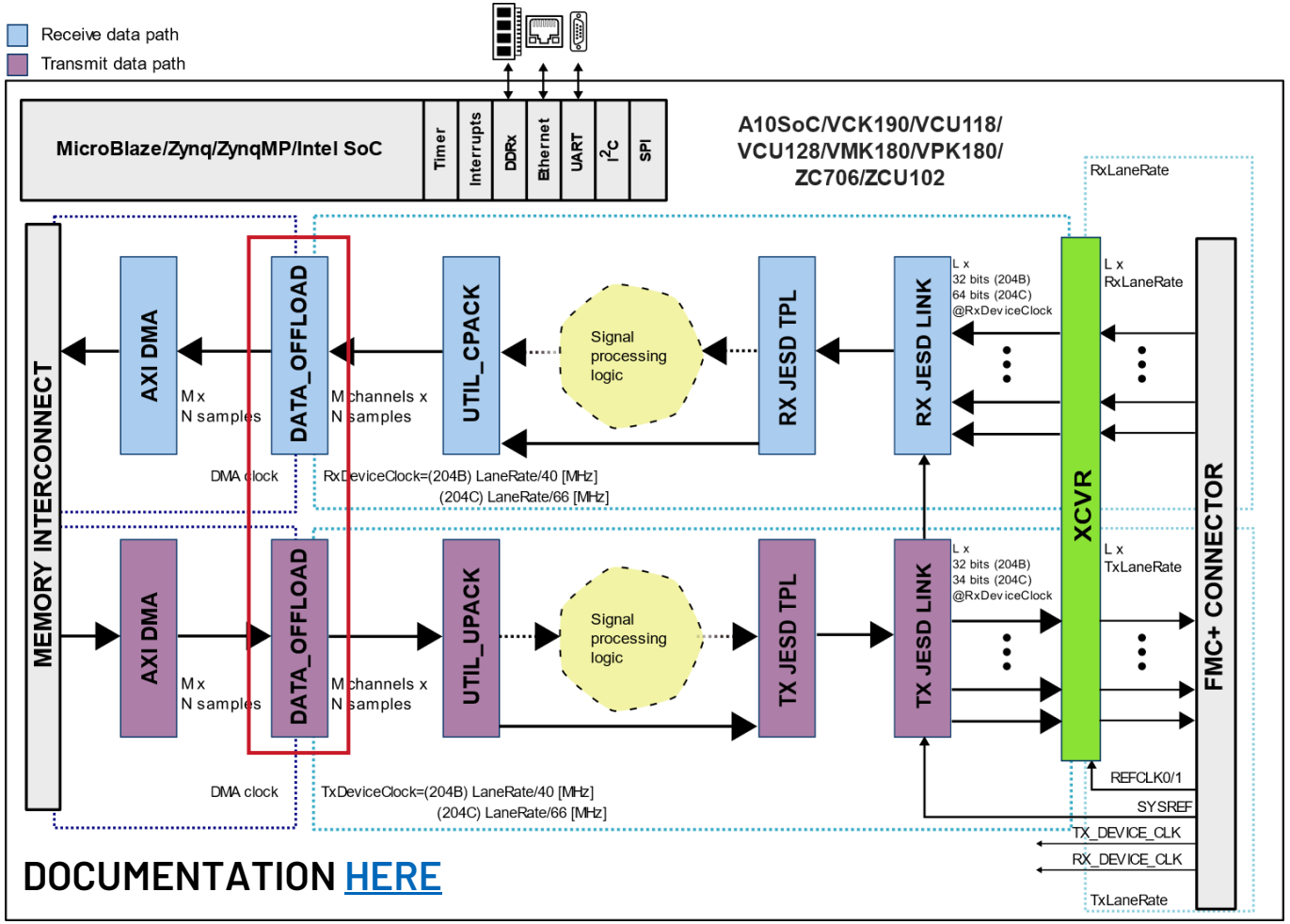
CPACK/UPACK

- Typically, the physical interface captures data continuously for the maximum number of channels the system is configured at boot
- The IPs are meant to allow one or more channels to be enabled by software without any padding, allowing full usage of the DMA bandwidth without any overhead
- Configurable number of channels, samples per channel, and sample data width



THREE CHANNELS ENABLED (4'b1110)

DATA OFFLOAD



DOCUMENTATION [HERE]

DATA OFFLOAD/ADC FIFO/DAC FIFO

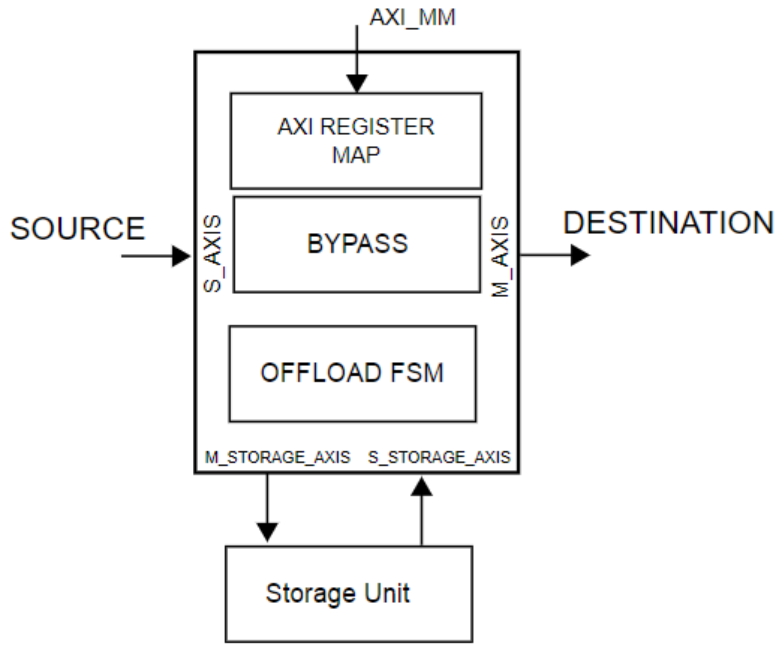
If the bandwidth of the data coming into the FPGA or the stream toward the device is higher than the capability of the memory interface, we need an additional FIFO. Sometimes this FIFO is implemented in the FPGA memory, sometimes we use an additional DDR memory, used exclusively for this purpose.

The Data Offload Engine is, in essence, a clock-domain crossing store-and-forward buffer (or FIFO) with some extra features useful in bursty RF applications.

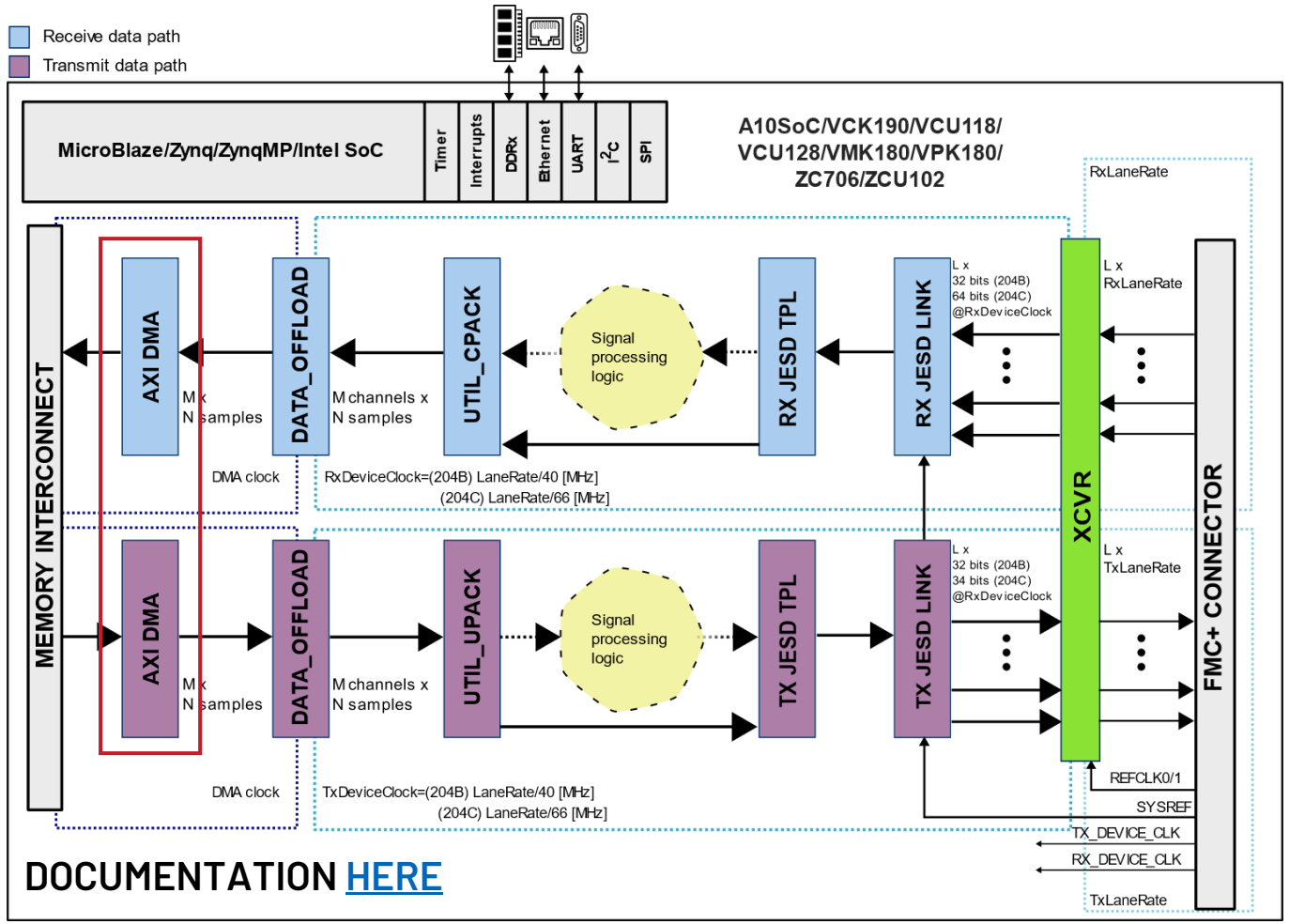
Specifically, it was designed to sit between the DMA and the DAC for the TX and between the ADC and the DMA for the RX path of a digital RF chain.

FEATURES

- Configurable storage unit with support for block-RAM and external DRAM (up to 16 GiB) or external high bandwidth memory (HBM)
- Configurable interface width and rates
- External timing synchronization for precisely timed buffers (For example, in combination with the timing-division duplexing controller)
- Cyclic and oneshot store-and-forward operation
- Bypass mode to completely bypass all features and act as a pure CDC FIFO
- Many settings configurable at runtime via an MM AXI4-Lite bus



AXI DMAC



AXI DMAC

The DMA is a crucial piece of the system, allowing transfer of data to and from memory at very high bandwidths

Cyclic functionality for the TX path is available

Sync functionality for both TX and RX path, which triggers data capture or data streaming starting from a precise time

FEATURES

Supports multiple interface types

- AXI3/AXI4 memory mapped
- AXI4 Streaming
- ADI FIFO interface

Zero-latency transfer switch-over architecture

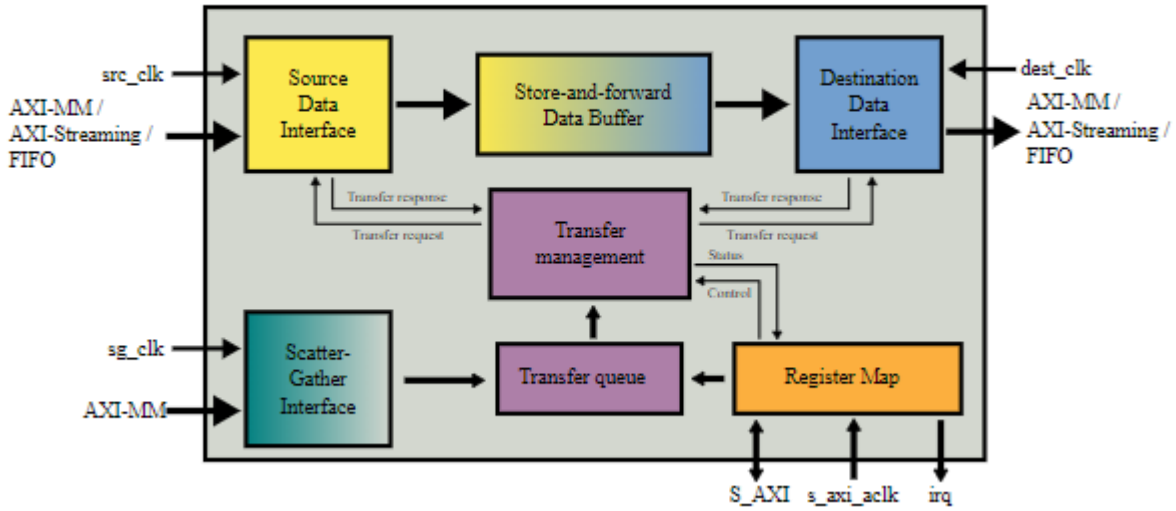
- Allows continuous high speed streaming

Cyclic transfers

2D transfers

Scatter-gather transfers

DOCS:
analogdevicesinc.github.io/hdl/library/axi_dmac/index.html



Software News and Updates

SOFTWARE

LINUX® NEW INFRASTRUCTURE MERGED MAINLINE

- DMABUF import interface for USB gadget framework
- DMABUF interface infrastructure for the IIO core
- IIO back-end support for complex aggregate devices

LIBIIO V1.X TO RELEASE VERY SOON

- DMABUF support
- Multi-buffer hardware support
- A lower-level samples buffer handling mechanism
- Modular back ends
- New asynchronous libiio/IIOD protocol

GNURADIO GR-IIO LIBIIO V1.X DRAFT PR

- github.com/gnuradio/gnuradio/pull/7442

FPGA/HDL

VERSAL AND ZYNQ ULTRASCALE+ MPSOC CACHE COHERACHE

- Coherent transactions from different masters connected to the cache coherent interconnect (CCI-400)
- Significant (3×) throughput increase over GbE and USB3

References

SOFTWARE DOCUMENTATION

FRAMEWORK
wiki.analog.com/resources/fpga/peripherals/jesd204
wiki.analog.com/resources/tools-software/linux-drivers/jesd204/jesd204-fsm-framework

LINK LAYER DRIVERS
wiki.analog.com/resources/tools-software/linux-drivers/jesd204/axi_jesd204_tx
wiki.analog.com/resources/tools-software/linux-drivers/jesd204/axi_jesd204_rx

PHY LAYER DRIVER
wiki.analog.com/resources/tools-software/linux-drivers/jesd204/axi_adxcvr

TRANSPORT LAYER DRIVERS
wiki.analog.com/resources/tools-software/linux-drivers/iio-dds/axi-dac-dds-hdl
wiki.analog.com/resources/tools-software/linux-drivers/iio-adc/axi-adc-hdl

TOOLS
wiki.analog.com/resources/tools-software/linux-software/jesd_eye_scan
wiki.analog.com/resources/tools-software/linux-software/jesd_status
github.com/analogdevicesinc/pyadi-jif

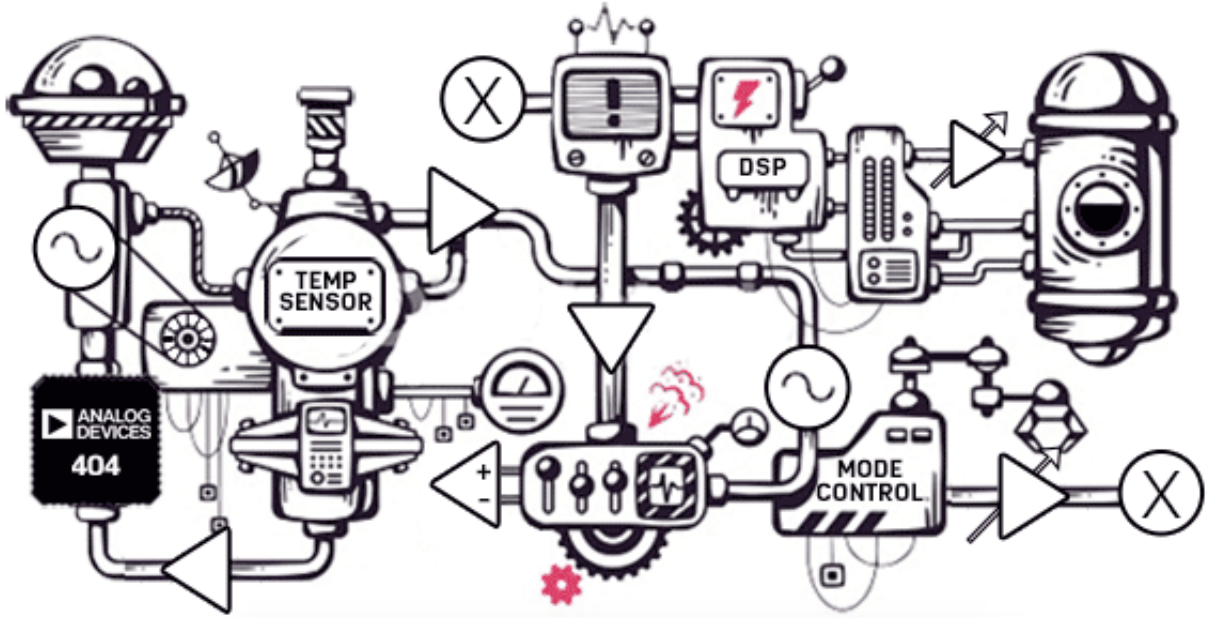
FPGA/HDL DOCUMENTATION

We are in the process of moving the HDL documentation from wiki.analog.com directly into the HDL GitHub repository.

During this transition period, please watch out the github.io and wiki docs. We apologize for any inconvenience that may occur due to this migration.

- analogdevicesinc.github.io/hdl/index.html
- wiki.analog.com/resources/fpga/docs/hdl

AHHH, TECHNOLOGY. WE CAN'T FIND THAT PAGE.



THANKS

Q & A

AHEAD OF WHAT'S POSSIBLE

analog.com