February 13, 2019

Bob Muro, Pentek Inc.

Using a COTS SDR as a 5G Development Platform

This article is intended to familiarize radio engineers with the use of a multi-purpose commercial off-the-shelf (COTS) platform for software-defined radio (SDR) that can reduce development time for 5G.

COTS SDR has been traditionally used in military radar and communications applications for high performance and design flexibility. The latest COTS SDR products offer solutions with integrated I/O, ARM processors and large FPGAs that also include intellectual property (IP) for accessing, routing and processing digital data. These attributes, combined with superior signal integrity, phase-coherent sampling and multi-channel transceivers, make a COTS SDR system an ideal choice for a 5G development platform.

COTS SDR DEFINED

For clarity, each section of this article is divided into subsections discussing hardware, firmware and software. Hardware comprises the SDR printed circuit board (PCB) and supporting components; firmware includes the internal FPGA code for logic and digital signal processing (DSP) functions; and software is the C code that controls the FPGA with firmware and performs any additional DSP functions.

Hardware

An SDR replaces legacy analog systems consisting of RF filters, analog down-convertors (i.e., the local oscillator and mixer), bandpass filters and demodulators (see **Figure 1a**). These fixed analog systems are limited to a specific function, such as an AM or FM radio.

An SDR exploits programmable DSP techniques to flexibly handle the increasing complexity, precision and bandwidth of today's radio traffic. To use the SDR, data conversion is required between the antenna and DSP for both receive and transmit functions.[1] An SDR receiver converts an RF signal from an antenna into digital samples with an analog-to-digital converter (ADC) and uses subsequent DSP operations to extract the required information from the signal (see **Figure 1b**). An SDR transmitter accepts digital information to be transmitted and performs the necessary DSP operations to produce digital samples for a digital-to-analog converter (DAC), whose output drives a power amplifier for delivery to the antenna (see **Figure 1c**). Because these radios are software-defined, they can be programmed on-the-fly in microseconds with new parameters or re-configured for different purposes by simply loading new firmware from internal or external memory.
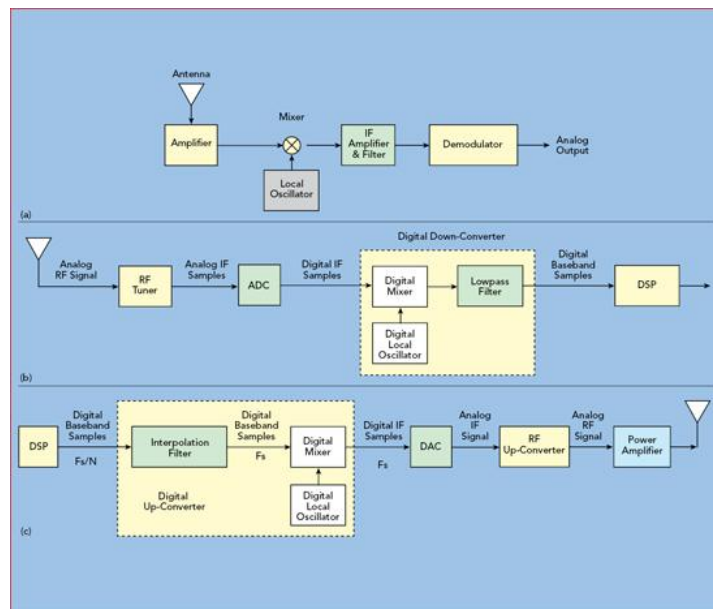

Figure 1 Legacy analog communications receiver (a) vs. SDR receiver (b) and SDR transmitter (c).

An SDR is often implemented on a specialized PCB called a "mezzanine card." The current generation is either a switched-fabric mezzanine card (XMC) or an FPGA mezzanine card (FMC). **Figure 2** shows XMC and FMC mezzanine cards with their corresponding functional block diagrams. Figure 2a is an XMC card with four 200 MHz ADC channels, and Figure 2b is an FMC card with two 3 GHz ADC channels and two 2.8 GHz DAC channels. Each board includes a precision timing system with a multi-bit, fractional synthesizer for variable sampling rates, which is locked to an on-board oven-controlled crystal oscillator (OCXO) or a reference input signal. These timing systems usually accept external synchronization signals from a network time protocol server or GPS receiver for the

precise timing requirements of a radar or cellular system. Precise time alignment is also required for phase-coherent sampling of the ADCs, FPGA DSP data synchronization and DAC signal transmission.
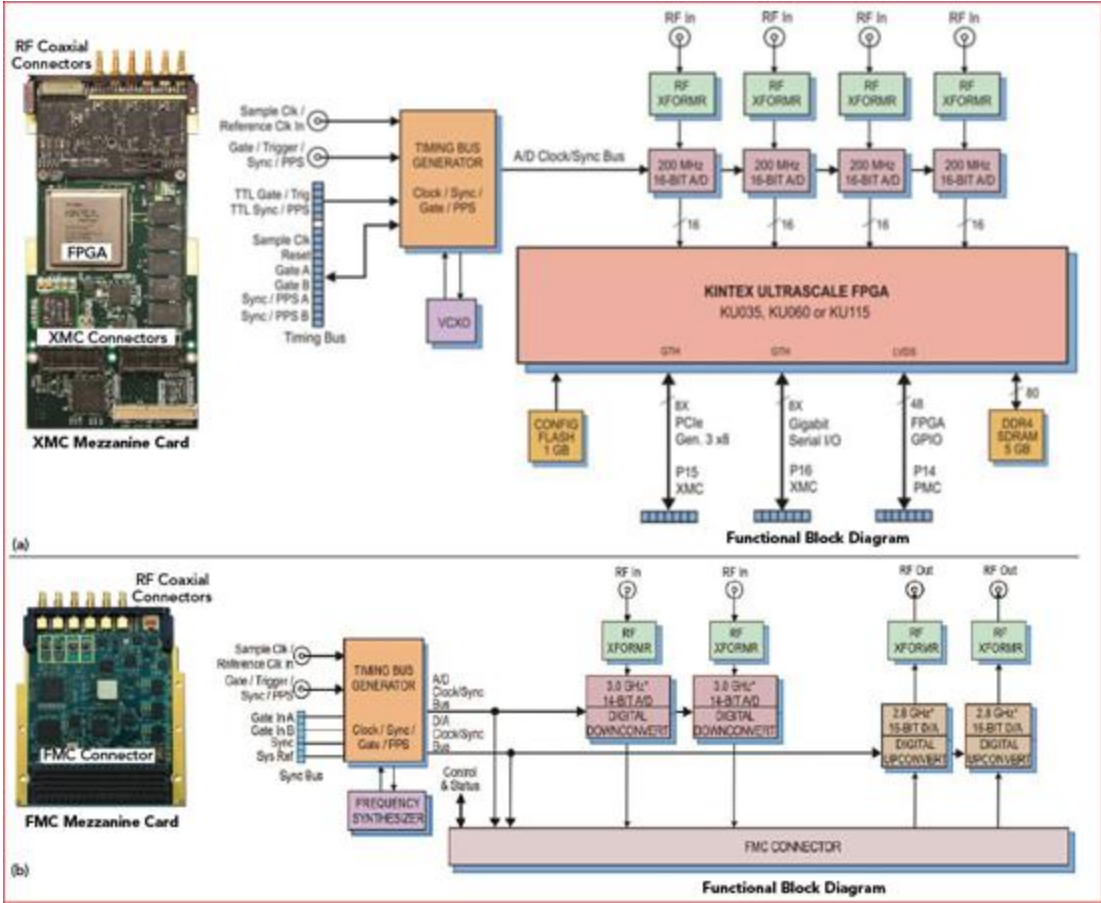

Figure 2 XMC (a) and FMC (b) mezzanine cards and functional block diagrams.

The XMC ADC has a 200 MSPS maximum sampling rate that can capture a 100 MHz Nyquist bandwidth, excluding filtering. A common technique with digital radio is to acquire channel information or intermediate frequency (IF) bandwidth by undersampling the signal (see Sidebar Nyquist Zones and Undersampling). Undersampling allows an ADC with a lower sample rate and higher dynamic range to capture a narrow bandwidth signal centered at a higher frequency without loss of information. For this to work correctly, the RF input path and the ADC must accommodate these higher frequency signals.

After analog-to-digital sampling, the next stage is typically the digital down-convertor (DDC), which performs frequency translation and bandwidth reduction. The DDC is often implemented as IP firmware within the FPGA.

Nyquist Zones and Undersampling
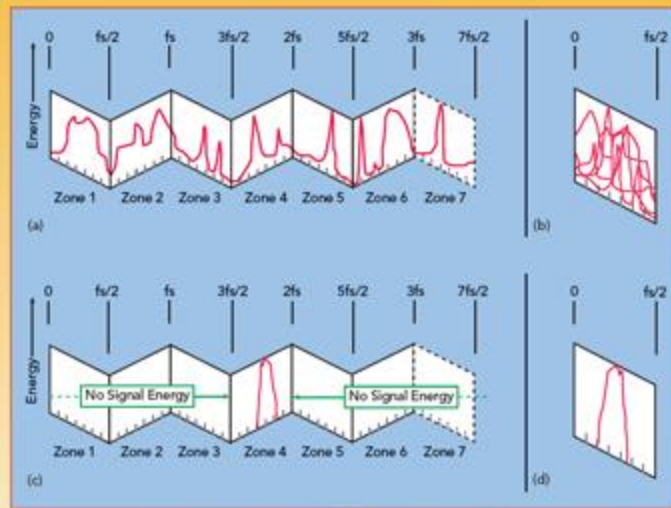
The Nyquist Theorem states: "Any signal can be represented by discrete samples if the sampling frequency is at least twice the bandwidth of the signal," where bandwidth is distinguished from frequency when explaining undersampling. Fan-fold printer paper can illustrate the concept of Nyquist zones (see **SB Figure 1a**), which are defined as multiples of half the sampling frequency, fs. If fs = 200 MHz, fs/2 = 100 MHz and successive Nyquist zones occur every 100 MHz.

All signal energy must fall within one Nyquist zone to satisfy both the bandwidth and frequency requirements of the Nyquist Theorem. The wideband signal shown in Figure S1a crosses multiple Nyquist zones and violates the "single zone" rule. Collapsing the fan-fold paper illustrates the result of sampling this wideband signal at fs: the signal energy above fs/2 will be "aliased" into the first zone, as shown in **SB Figure 1b**. Adding a lowpass filter to remove the frequencies above fs/2 before sampling will prevent aliasing.

However, aliasing can be beneficial when used on a narrowband signal that falls in an upper Nyquist zone, if the energy from the other zones is eliminated with a suitable bandpass filter (see **SB Figure 1c**). This is called undersampling.[1] In the Figure, the narrowband signal falls within Nyquist Zone 4, between 300 and 400 MHz if fs = 200 MHz. Although the signal frequency is above fs/2, the spectrum is contained within a single zone, satisfying the Nyquist Theorem. Undersampling this signal will alias it to Zone 1 (see **SB Figure 1d**) where signal processing is easier.

Firmware

An FPGA consists of unconnected logic, arithmetic and signal processing building blocks that are configured with IP firmware to perform specific functions. While ideal for extreme programming flexibility, firmware development is complex. To simplify the development process, some COTS SDR manufacturers provide FPGA IP for basic operation of their boards. This usually includes analog and digital I/O functions for acquiring and transmitting data, with DSP IP for specific radio functions like DDCs, filters, channelizers and engines to transfer data to the system.

The DDC function requires three IP building blocks: the numerically controlled oscillator (NCO) local oscillator, a complex mixer and digital filters to replace the functions of the legacy analog radio system (see Figure 1). The tuning stage of the DDC uses a complex digital mixer to translate the frequency of interest to baseband. A pair of multipliers driven by a direct digital synthesizer (DDS) NCO allows the user to "tune" the receiver to the desired frequency. The samples are then passed through a lowpass finite impulse response (FIR) filter to decimate the signal for a finite channel bandwidth. Two key benefits of the DDC are higher signal-to-noise ratio (SNR), as a result of decimation, and the ability to tune to the narrowband center frequency of the signal (see Sidebar Improving SNR with Digital Processing). Decimating the signal effectively lowers the sample rate and reduces uncorrelated white noise, and the NCO enables precise digital tuning to a specific carrier frequency within a single Nyquist zone.

Software

While the vendor-provided FPGA IP might meet the specifications for a specific application, the system implementation may require controlling software to operate the radio. The FPGA IP needs operational parameters sent across the system interface from a software program, which is the function of a board support package (BSP) normally written as "C" callable routines for a Windows or Linux environment. The BSP contains library functions and pre-compiled example code that can be executed to test board functionality. One such function is commanding the ADC to capture and transfer data to the FPGA for further processing in the DDC. This processed data can be stored in memory or transferred to the DAC for conversion back to an analog signal and output for transmission. This is an example of a software program developed using the BSP software library functions and drivers. If any new FPGA IP is created by the user, additional control software must be written and included in the BSP package.

Data Transport Requirements

5G will require significantly more transport capacity than LTE.

With LTE, a RRH with two antennas and 5 MHz channel bandwidth will have the following data transfer requirements: The 5 MHz channel requires at least 10 MHz sampling, or 10 MSPS, to capture the information without aliasing. With two bytes per 16-bit sample and one sample each for the I and Q data, the maximum sampling rate will be:

SR_{Max} = 10 MSPS x 2 bytes/sample x 2 for I and Q = 40 MB/s x 8 bits/byte = 320 Mbps per antenna. With two antenna inputs, the data transport will require 320 Mbps x 2 = 640 Mbps data throughput, which poses no issue for a CPRI port, which has 10 to 25 Gbps capacity.

A 5G link with a 100 MHz channel and eight antenna inputs increases the data transfer requirement to approximately 52 Gbps, which will require multiple CPRI ports. These are approximations, as the calculations ignore encoding variations.

LATEST COTS SDR TECHNOLOGY

Hardware

Over the past 10 years, FPGA manufactures like Xilinx have been improving technology by reducing the size of silicon process nodes, which reduces device size, weight and power (SWaP). In late 2008, the Xilinx Virtex-6 family was constructed using a 40 nm process and averaged 2000 DSP slices per FPGA. By 2017, the Ultrascale family was on a 20 nm process, and the FPGA DSP slices increased to approximately 5,500. The latest system on a chip (SoC) from Xilinx, the RFSoC, consists of an FPGA fabric with ARM processors, ADCs and DACs, all on the same chip.

The 16 nm technology has over 4,200 DSP slices; four 1.5 GHz, A53 ARM processors; two 600 MHz, R5 ARM processors; eight 4 GHz, 12-bit ADCs; and eight 6.4 GHz, 14-bit DACs per device.
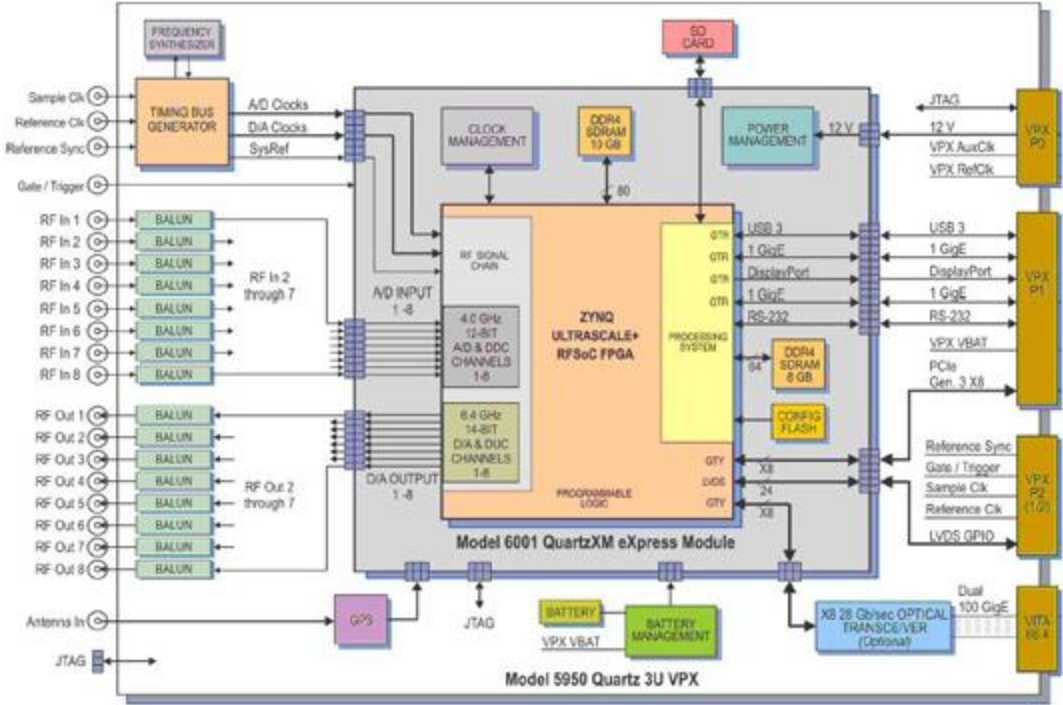

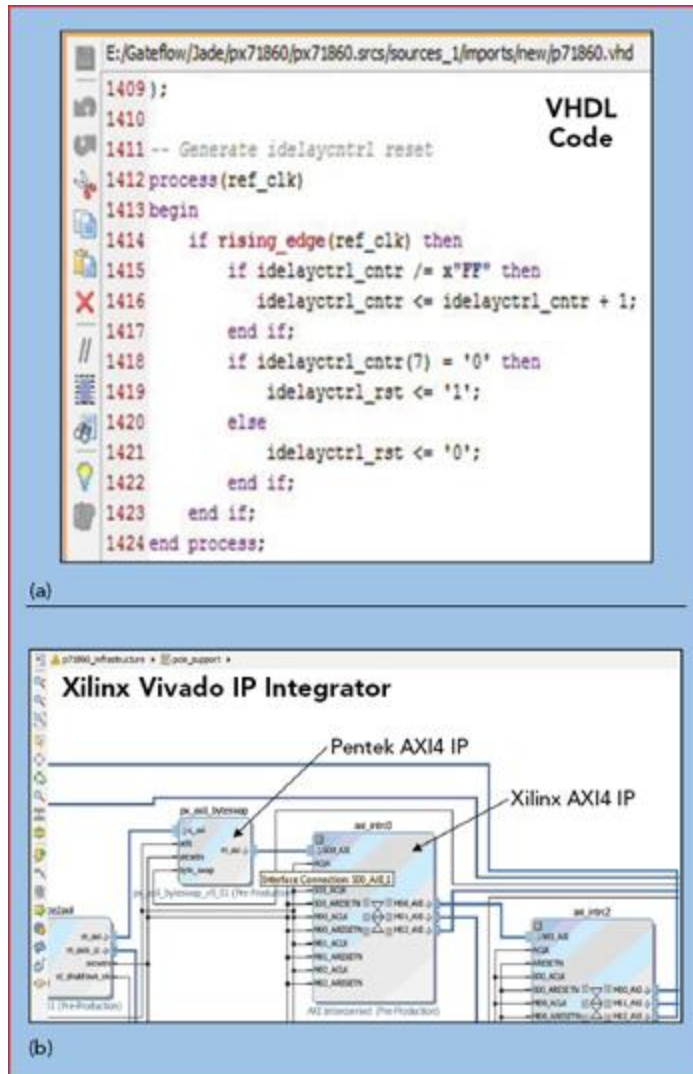**Figure 3** Pentek COTS SDR based on the Xilinx RFSoC.

```
E:/Gateflow/Jade/px71860/px71860.srcs/sources_1/imports/new/p71860.vhd
1409 );
1410
1411 -- Generate idelaycntrl reset
1412 process(ref_clk)
1413 begin
1414     if rising_edge(ref_clk) then
1415         if idelayctrl_cntr /= x"FF" then
1416             idelayctrl_cntr <= idelayctrl_cntr + 1;
1417         end if;
1418         if idelayctrl_cntr(7) = '0' then
1419             idelayctrl_rst <= '1';
1420         else
1421             idelayctrl_rst <= '0';
1422         end if;
1423     end if;
1424 end process;
```

VHDL Code

(a)

Figure 4 Xilinx IP Integrator tool, showing VHDL code (a) and intuitive "drag and drop" graphical blocks (b).

Figure 3 shows a functional block diagram of one COTS implementation of the Xilinx RFSoC, the central component of the 5950 3U VPX board from Pentek. The center area including the RFSoC is a fully connectorized system on module (SoM) that plugs into a 3U VPX carrier. While this device can be controlled via a Gigabit Ethernet port, similar to the previous generation FPGA, the on-board ARM processors allow autonomous operation and the ability to communicate with, or control, devices locally or on an external network.[3]

Firmware

Previous generation FPGAs were programmed using a textual hardware description language (HDL) like VeriLog or very high speed integrated

circuit description language (VHDL). The latest AXI4 compliant IP blocks are included in Vivado from Xilinx. The IP Integrator tool from Xilinx has virtual graphical blocks that represent HDL code, which can be connected to one another via drag-and-drop wiring. **Figure 4** shows an example of VHDL code (see Figure 4a) and the corresponding drag-and-drop graphical blocks (see Figure 4b). This more intuitive way to program allows someone new to FPGAs to wire together logical blocks representing hardware like FIR filters and DDCs to create an SDR. This programming method supports fast integration of vendor-supplied, hardware-specific IP blocks with Xilinx IP blocks to create a working SDR. Both IP block types can be combined to create a common library.

Improving SNR with Digital Processing

The calculation of the theoretical SNR of an ADC is $SNR = 6.02N + 1.76$ dB, where N is the number of bits in the ADC.[2] For a 200 MHz, 16-bit ADC, $SNR = 6.02 \cdot 16 + 1.76$ dB $= 98$ dB. This calculation assumes the input signal is full-scale, so the actual SNR is likely lower because the gain of the front-end LNA will be reduced, "backing off" to compensate for high crest factor communication signals. The theoretical SNR will be further reduced because of ADC nonlinearity, additional noise from the LNA input and sample clock jitter. The actual measured SNR for a good quality ADC is limited to approximately 75 dB.

However, the dynamic range can be improved by reducing the bandwidth using a DDC, from the full Nyquist bandwidth to the channel bandwidth of the system. For example, the calculated SNR for a 200 MHz ADC with $f_{Nyquist} = fs/2 = 100$ MHz and a 5 MHz LTE channel bandwidth (BW) will be $SNR_{PG} = SNR_{Nyquist} + 10\log_{10}(f_{Nyquist}/BW) = 75 + 10\log10 (100/5) \approx 88$ dB, a 13 dB improvement.

Software

These IP programming advances have provided an opportunity for COTS vendors to create a single BSP module that corresponds to one IP module with all the necessary FPGA program parameters in one location. One example is a "clock control BSP module" that corresponds directly to a "clock control IP module."

5G APPLICATION

This latest generation of SDR technology is game changing and can be used by COTS manufacturers to provide multi-channel SDR transceivers for engineers developing 5G radio products.
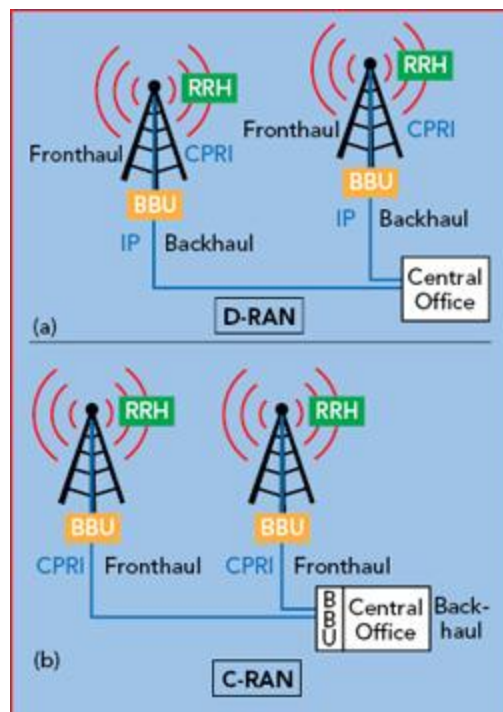

Figure 5 Distributed (a) and centralized (b) RAN.

Figure 5 illustrates the difference between distributed and centralized radio access networks, D-RAN and C-RAN. With LTE, the traditional D-RAN cell sites were being replaced by newer C-RANs to improve data transfer efficiency and reduce radio cost. However, the mmWave massive MIMO architecture for 5G requires the separation to move the remote radio head (RRH) closer to the end user because of the increased RF path loss.

Figure 6 shows a functional block diagram of a C-RAN consisting of a baseband unit (BBU), RRH, GPS time/frequency reference and an interconnect module. Several of the blocks are highlighted to note possible use of COTS SDRs. The BBU is located at a central office or a virtual network "in the cloud," with access to multiple optical data lines for backhaul. The RRH is at an external location closer to the end user. The BBU and RRH in this fronthaul connection example can use a common public radio interface (CPRI), open base station architecture initiative (OBSAI) or standard Ethernet connection, depending upon system requirements. New fronthaul concepts like extensible radio access

networks (xRAN) and open radio access networks (ORAN) will replace these legacy interfaces in the future.

These various transfer mode options combined with legacy cellular, Verizon 5G Technical Forum (5GTF) or the 3GPP 5G New Radio (NR) specification are configured to form a complex heterogeneous network, requiring a flexible development platform.[4-6]
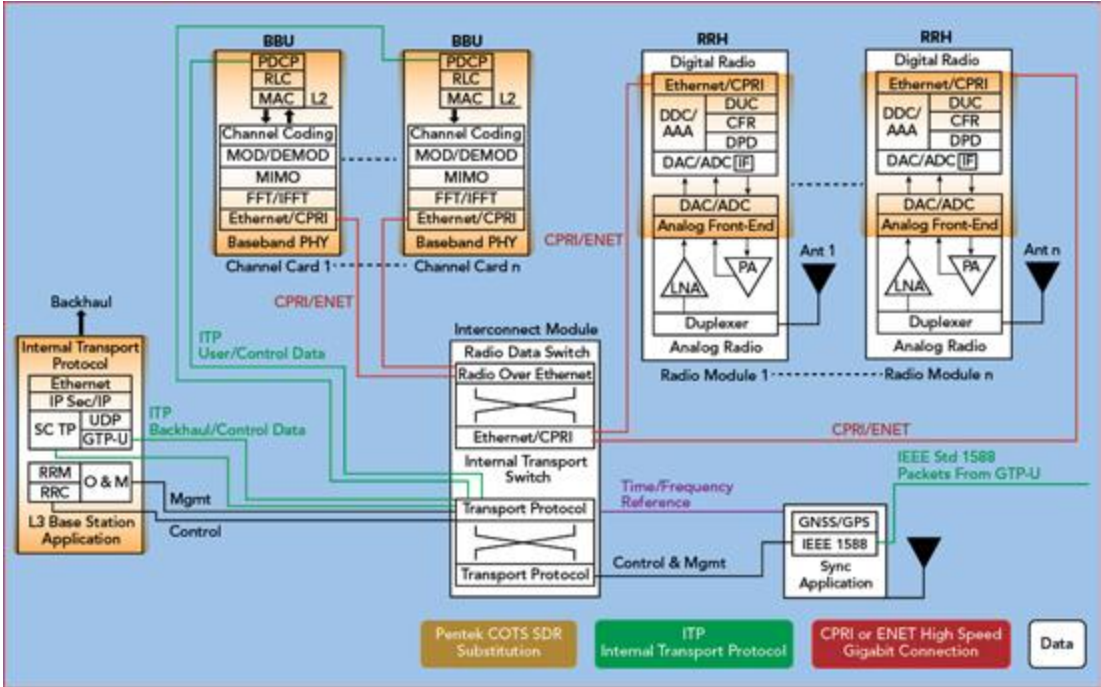

Figure 6 C-RAN functional block diagram, showing where COTS SDR can be used.

Hardware

Figure 7 RRH functional block diagram (a), showing the functions that can be implemented with a COTS SDR (b).

Figure 7 shows an example of using a COTS SDR board to emulate a RRH in a C-RAN architecture. A subsection of the original C-RAN with the RRH is shown in Figure 7a, with the COTS SDR RRH in Figure 7b. The encircled area in Figure 7a can be realized with the carrier card shown in Figure 7b. The custom modular carrier card contains the receive and transmit amplifiers, a GPS receiver and an optoelectronic transceiver module. The inner SoM contains the RFSoC and all connections for power management, data storage and analog/digital I/O. The incoming RF signal from the antenna is connected to the receive low noise amplifier via a duplexer, isolating it from the high power amplifier transmit levels and connecting it to one ADC channel. With the necessary IP, this SoM and custom carrier combination can emulate the original RRH.

Firmware

Once inside the FPGA fabric, the digital samples are decimated, frequency selected or tuned and filtered in the DDC. The DDC output samples can be streamed to the power meter module for measurement and sorted in the threshold detector IP module. These processed samples can be streamed to the ARM processors for crest factor reduction and digital predistortion routines before being up-converted in the digital up-converter (DUC) for re-transmission. The DUC is the reverse of the DDC, using frequency translation and interpolation instead of decimation. The digitized I/Q sample data is packetized in the digital radio for transport to the BBU via a radio data switch. Because of the variety of channels and data transfer protocols, it is necessary to understand the maximum data throughput of the signal and ensure sufficient network capacity (see Sidebar Data Transport Requirements).

Software

Depending on the desired level of control, BSP routines would be created for the new IP and ARM processors, or the ARM processors, in conjunction with the FPGA, can be programmed to operate autonomously.

CONCLUSION

The purpose of this article is to familiarize a traditional radio engineer with the latest hardware, firmware, software and design tools available from COTS SDR suppliers, showing that an SDR can be used as a 5G development platform. These SDR platforms provide superior signal integrity, high test repeatability and modular assemblies that adjust to constantly changing 5G design requirements. 5G evolution will require many development cycles for experimentation and optimization, and the use of a COTS system as a starting point will accelerate time to market.

Improving SNR with Digital Processing

The calculation of the theoretical SNR of an ADC is SNR = 6.02N + 1.76 dB, where N is the number of bits in the ADC.[2] For a 200 MHz, 16-bit ADC, SNR = 6.02•16 + 1.76 dB = 98 dB. This calculation assumes the input signal is full-scale, so the actual SNR is likely lower because the gain of the front-end LNA will be reduced, "backing off" to compensate for high crest factor communication signals. The theoretical SNR will be further reduced because of ADC nonlinearity, additional noise from the LNA input and sample clock jitter. The actual measured SNR for a good quality ADC is limited to approximately 75 dB.

However, the dynamic range can be improved by reducing the bandwidth using a DDC, from the full Nyquist bandwidth to the channel bandwidth of the system. For example, the calculated SNR for a 200 MHz ADC with $f_{Nyquist}$ = fs/2 = 100 MHz and a 5 MHz LTE channel bandwidth (BW) will be SNR_{PG} = $SNR_{Nyquist}$ + $10\log_{10}(f_{Nyquist}/BW)$ = 75 + 10log10 (100/5) ≈ 88 dB, a 13 dB improvement.

References

1. R. Hosking, "Putting FPGA's to Work in Software Radio Systems," 11th Edition, Pentek, March 2018, www.mwee.com/Learning-center/pentek-putting-fpgas-work-software-radio-systems-11th-edition.
2. W. Kester, "Taking the Mystery Out of the Infamous Formula 'SNR = 6.02+1.76dB,' and Why You Should Care," Analog Devices, October 2008, www.analog.com/media/en/training-seminars/tutorials/MT-001.pdf.
3. S. Sgandurra, "Strategies for Deploying Xilinx's Zynq Ultrascale+ RFSoC," Pentek, May 2018, www.pentek.com/pipeline/27_2/rfsoc.cfm.
4. S. Ahmadi, "Towards 5G; Xilinx Solutions and Enablers for Next-Generation Wireless Systems," *XILINX*, June 2016, www.xilinx.com/support/documentation/white_papers/wp476-toward-5g.pdf.
5. A. Oliva, J. Hernandez, D. Larrabeti and A. Azcorra, University of Madrid (UC3M), An Overview of the CPRI Specification and Its Application to C-RAN-Based LTE Scenarios IEEE Communications Magazine, February 2016.
6. P. Moakes, "5G New Radio Architecture and Challenges," CommAgility White Paper.

http://www.microwavejournal.com/articles/31756-using-a-cots-sdr-as-a-5g-development-platform?page=1