

COTS digital radio receiver system is good to go

By Daniel McLane, Amit Mane, and Billy Kao

This software-defined radio reference design implements a 40-channel digital receiver system that integrates reprogrammable FPGAs with high-speed DSPs to deliver 1 GHz data rates in 10 kHz frequency increments. The hardware-in-the loop design methodology allows the developer to design and download a whole system to hardware in hours. Each design iteration can be implemented and tested in less than a day.

In the competitive business world of the quick and the dead, a commercial-off-the-shelf (COTS) design solution can give a software-defined radio (SDR) designer a significant time-to-market advantage. Designing a digital radio receiver (DRR) system from the board up would not only be unacceptably time-consuming, it would also be prohibitively expensive in terms of non-recurring engineering (NRE) costs.

This paper demonstrates a HW/SW COTS reference design package for a high-performance SDR DRR system using new design tools and hardware that can dramatically reduce the complexity and development time to bring a product to market on a mission-critical timeline.

System description

The DRR reference design implements a 40-channel digital receiver on a single 6U CompactPCI card set. The DRR system integrates the reprogrammable flexibility and computing power of Xilinx Virtex-II Pro XC2VP40 FPGAs with high-speed Texas Instruments (TI) TMS320C6416 DSPs to provide a fully programmable SDR system.

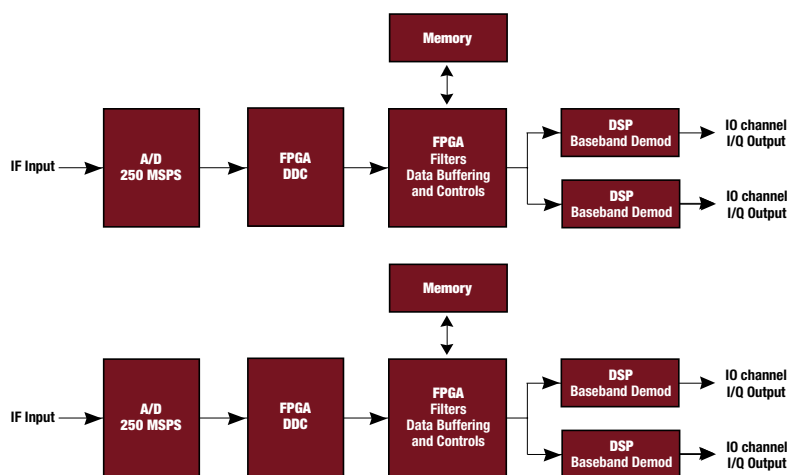


Figure 1

The V-II Pro FPGAs each have four IBM PowerPC embedded processors that, combined with the 1 GHz TI DSPs, provide a powerful and flexible computing core, allowing all of the signal processing to be implemented in software.

The system block diagram in Figure 1 shows the major hardware features from the signal digitizers through the signal processing FPGAs and into the DSPs.

Signal path

The DRR signal processing starts with a wideband, intermediate frequency (IF) Linear Technology LTC2245 signal digitizer capable of up to 250 MSps. The 12-bit analog-to-digital device has a 350 MHz analog input bandwidth suitable for a wide range of IF signal sampling and under-sampling applications. The IF signal digitizing is followed by a digital down converter (DDC) and then channel filters. The down converted channel signal is subsequently filtered into baseband channel data, ready for demodulation by the DSPs.

The Mathworks' MATLAB Simulink design tool allows the developer to implement a block diagram directly into the logic hardware. MATLAB uses Xilinx System Generator software to create an FPGA signal processing design, and then the DRR reference design hardware board support package (BSP) enables the implementation straight from the MATLAB program to hardware.

Using the DRR reference design, the DRR signal processing can be customized for multiple applications including high-end coprocessing, wireless communications, video, biometrics, and electronic warfare.

Real-time simulation saves time

In the MATLAB Simulink environment using the Xilinx System Generator blockset, the simulation data is bit-true, cycle-true, and accurately reflects the performance of the real system. Characteristics of

a system can be easily modified by changing the parameters in the blocksets and then immediately verified in real-time simulation in hardware.

Hardware interfaces and supporting features in the FPGAs – such as system controls, DSP interface, and data buffering – are provided as HDL code into which the signal processing is integrated. The TI DSP real-time operating system (RTOS) provides device drivers for the DSP that allow the SDR developer to concentrate efforts on application-specific signal processing.

Hardware-in-the-loop

The entire SDR signal-processing model for the DDC and channel filtering is built using Xilinx System Generator under the MATLAB Simulink environment. This is a key feature for rapid DRR development, because the designer can use MATLAB as part of the system verification. This hardware-in-the-loop methodology means the MATLAB model can run in real-time in hardware and immediately be evaluated against the theoretical performance.

This is an extremely powerful method of code development, because it bridges the gap between the whiteboard concept and the hardware. The whole DRR system can be designed and downloaded to the hardware in hours. Each design iteration only takes hours, not days, to implement and test.

Digital down converter and filters

The initial signal processing step in the DRR system is digital down conversion. The DDC first mixes the digitized IF signal with a tuning signal resulting in a frequency shift to the baseband represented as an in-phase and quadrature (I/Q) data pair for the channel. The I/Q signals are then filtered and decimated using a cascaded integrator-comb (CIC) filter.

The CIC filter allows the FPGA to process the data at rates of up to 208 MSps from the digitizer. The CIC filter is followed by compensation (CFIR) and programmable (PFIR) filters built in another FPGA. These filters reject out-of-band noise and provide channel isolation after the digital down conversion.

The end-to-end system data is decimated by a factor of 120 in the DRR reference design. This decimation results in channel data at 1.083 MSps for each channel.

Filter design

During the filter design process, the designer trades off the competing demands of achieving better channel separation by using a sharper filter with higher out-of-band rejection against the size and complexity of the filters. Filters with high data rates and large numbers of poles require more logic resources to implement. Therefore, the filter designs are a critical factor in meeting the system channel separation requirements, while still keeping the design practical.

This design process is made much easier because trade-offs can be determined during the design stage with the assurance that the system behavior will match the theoretical model created in the Xilinx System Generator software.

Once the DDC and system filters are designed using the MATLAB system model, the designer can simulate the entire DRR process and verify functionality using bit-true and cycle-true modeling of the hardware. The reliability of the modeling allows the designer to fine-tune the system performance to meet the most exacting requirements.

Hardware

The IF signal is digitized using Linear Technology A/D devices mounted on two ultra-wide (UWB) PMC modules. After the signal undergoes digital down conversion and initial filtering in a Virtex-II Pro FPGA on the UWB module, the signal is transmitted to a Quadia CompactPCI board for baseband processing by two more Virtex-II Pro FPGAs and four TI DSPs. Each DSP processes 10 discrete channels. The Quadia board is shown in Figure 2.

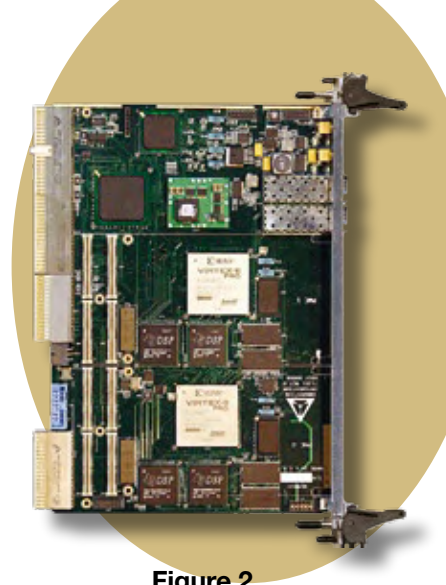


Figure 2

The host computer controls the DRR system initialization, tuning frequency, CIC filter gain, and spectral inversion. Configuring the DRR channel features directly from the host PCI bus saves time, because these functions do not have to be recompiled in the logic.

The system allows all channels to trigger simultaneously and to operate synchronously. Data buffering in the DRR for each DSP helps to meet the real-time demands on the system for processing. The data buffering implements a 16 MB queue for each DSP in external DDR DRAM.

The DRR logic consumes about 90 percent of each UWB FPGA and 50 percent of each Quadia FPGA. The TI DSP chips are virtually 100 percent available for baseband processing and analysis.

DRR in operation

Figure 3 charts captured I/Q data for the 10 channels delivered to one DSP for a 97.5 MHz IF signal under-sampled at 130 MSps. The frequency of the input sine wave is 97.5 MHz, and the tuning frequency for each of the 10 channels ranges from 32.51 MHz to 32.60 MHz in 10 kHz increments.

The Figure 3 results match the MATLAB theoretical model for frequency response and performance. Ultimately, the DRR system is only limited in its performance by the design of the digitizing hardware.

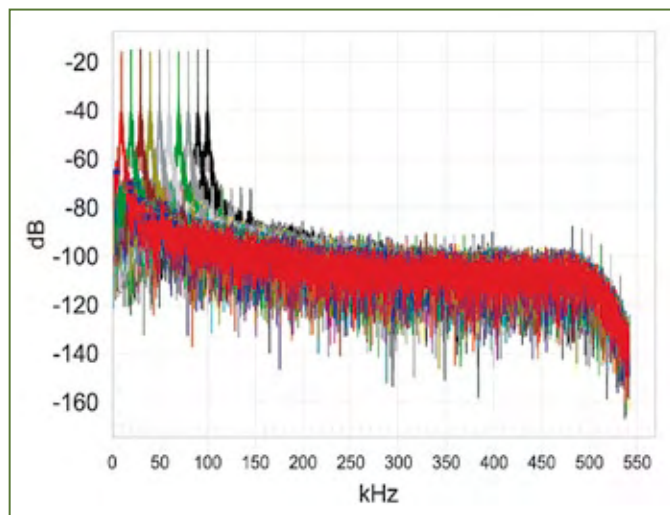


Figure 3

DRR host and DSP software

The DSP software, running under TI's DSP RTOS is responsible for the baseband signal processing and data flow to the host. The host software, running under Windows, is used for DRR configuration, data logging, and system monitoring.

The DRR reference design provides C++ development libraries on both the host and the target DSPs to perform the management of the hardware and data flow. The DRR system can capture and deliver snapshots of data to disk on command from the host. The application then logs the data to disk for analysis.

Conclusion

By taking advantage of a high-performance COTS reference design such as the DRR described in this paper, the SDR designer can be a winner in the race to be first-to-market.

The DRR system was developed by Innovative Integration Inc. of Simi Valley, Calif., USA. The company specializes in DSP-based signal processing and hardware-assisted (FPGA) signal processing, as well as real-time systems and PC-based real-time data acquisition.

Offered as a total COTS package, the DRR reference design package has all the hardware and software – including the MATLAB board support packet and proprietary libraries – a designer needs to implement a customized, high-performance SDR receiver system. For more information, please visit www.innovative-dsp.com/DRR. 

Daniel McLane co-founded Innovative Integration in 1990, and leads the hardware and signal processing engineering teams. He is a key designer for Innovative Integration and has designed electronics and embedded systems for signal processing, controls, and data acquisition for more than 20 years. Dan holds a BSME degree from Purdue University and an MSME from the University of California at Los Angeles (UCLA). Daniel can be reached at dmclane@innovative-dsp.com.



Amit Mane joined Innovative Integration as a system engineer in March 2003. His areas of expertise are embedded system design using DSP and FPGA, PCI, PCIe, and VME. He holds BSEE and MSEE degrees from Texas Tech University (Lubbock, TX), as well as a postgraduate diploma in VLSI design. Amit can be reached at amane@innovative-dsp.com.



Billy Kao is a senior system engineer specializing in signal processing. He has developed many signal-processing applications for digital radio and data analysis in MATLAB for embedded applications in FPGAs. He holds BSEE and MSEE degrees from Texas Tech University (Lubbock, TX). Billy can be reached at bkao@innovative-dsp.com.



Innovative Integration

2390-A Ward Avenue • Simi Valley, CA 93065
Phone: 805-578-4260 • Fax: 805-578-4225
Email: sales@innovative-dsp.com
Website: www.innovative-dsp.com