

FPGAs for Software Radio

With Moore's law at work, these FPGA devices sport incredibly small silicon geometries capable of operating at very low core voltages and very high clock speeds.

Most of these new devices take advantage of ball grid array packaging to deliver thousands of I/O pins while keeping the footprint very small.

In the last decade, ASICs (application specific ICs) and DSPs have been deployed to handle nearly all the signal processing functions associated with radio communications. Even though field-programmable gate arrays have been around for decades, the latest generation of FPGAs is so powerful that it is now displacing both ASICs and DSPs in the latest software radio applications. Furthermore, a new class of design tools opens up FPGAs to both hardware and software engineers. Debugging these new devices has been greatly simplified by excellent modeling and analysis tools.

Perhaps the most exciting new twist in design tools is the growing libraries of IP (intellectual property) cores available from FPGA vendors, and a whole new industry of third party companies that offer IP cores for specific application areas.

Software Radio Functions

Figure 1 shows a typical wireless communication software radio system with various signal processing tasks identified. Usually, the digital downconverter or digital receiver section is handled in a dedicated ASIC device consisting of three major blocks: the mixer, the local oscillator and the filter.

The local oscillator, or NCO, consists of a phase accumulator, which is just a register and an adder available as standard library blocks for virtually all FPGAs. The phase value in the accumulator drives a sine/cosine lookup table which can be implemented as a simple ROM (read-only-memory). The mixer is nothing more than a pair of digital multipliers, now available as dedicated hardware resources in the latest FPGAs.

Programmable DSPs are often used to implement some common demodulation, decoding and analysis functions. However, FPGA vendors and third parties now offer a good selection of IP libraries to handle Viterbi, Reed-Solomon, convolutional and trellis decoders, data encryption standard engines, and various noisy channel models.

Both FFT and discrete cosine transform cores are also available with several different block lengths.

Several vendors are offering scalable FFT engines ranging in size from 16 to 1024 points and higher using complex radix 4 algorithms. Since the FFT oper-

ates on blocks of data, the block memory RAM available with the newer FPGA devices allows some design tradeoffs. You can save memory by doing an in-place FFT with a single block, or move up to a swinging buffer arrangement to provide continuous real-time calculations.

To help with control functions, a long list of IP cores is available for popular processors. The benefit here is that by using a well-supported core processor, one can take advantage of existing code and the software development tools already in place for these engines.

For speech and video signals, a wide range of IP cores including ADPCM, J-PEG and color space converters is available. The same is true for the decimating low pass filter which usually consists of several CIC filter stages followed by a FIR filter. If some or all of these signal processing functions are handled by the FPGA, the programmable DSP can then concentrate on the control and analysis functions, much more appropriate for its capabilities.

Digital Receiver ASICs vs. FPGAs

Although digital receiver functions can be implemented within the FPGA, it can draw more power and cost much more per channel than an ASIC, depending on many factors like sampling frequency, filter characteristics, and signal-to-noise requirements.

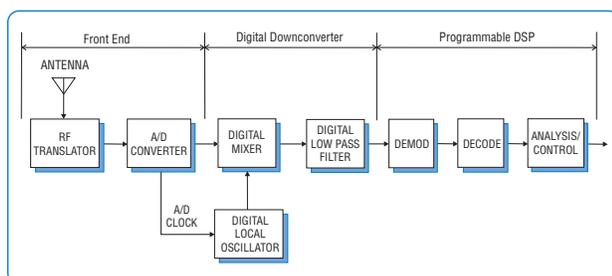
Usually, the ASIC digital receiver has been designed with a full set of standard operating modes and features and has been more thoroughly tested and characterized than a custom combination of IP core building blocks available for FPGAs. This gap will obviously shrink as the cores become more complete. Since the ASIC hardware is optimized for dedicated functions, the latest ASIC devices are usually the better choice for extremely high-performance receiver applications.

However, if you can't buy a standard digital receiver ASIC with just the right phase and frequency characteristics, the flexibility of the FPGA might get the job done. Also, for proof-of-concept systems or when time-to-market is critical, FPGAs are often the right choice.

When the required signal-to-noise ratios, filter skirts, or frequency templates are beyond the complexity of the standard filter inside a commercial ASIC, the flexibility of IP core filter designs for FPGAs can provide just about any characteristic.

[Continued on next page]

Figure 1. Digital signal processing functions for software radio.





FPGAs are successfully taking up many of the roles formerly played by DSPs and ASICs. One of the major benefits of the IP core concept is that, like high-level languages, these cores can migrate to the next generation devices. This means that the wealth of core functions can accumulate and diversify, eliminating the need to start over again for each new family.

Pentek offers the GateFlow collection of FPGA Resources, compatible with the Xilinx Virtex products.

GateFlow includes the FPGA Design Kit which allows the engineer to implement custom algorithms and signal processing functions.

The GateFlow IP Core Library is a collection of highly optimized Pentek IP Cores for implementing functions such as FFTs and Digital Receivers on any Xilinx FPGA platform.

The GateFlow Installed Cores are high-performance signal processing functions installed at the factory in the FPGAs of Pentek boards.

DSPs vs. FPGAs

While FPGAs can handle many of the tasks traditionally performed on a programmable DSP chip, there are several factors worth considering. For example, even though today's FPGAs have quite a bit of on-board RAM, it's still a far cry from the large external SDRAMs normally surrounding a DSP chip. In spite of all the best design tools and simulators, DSP code (like any software) always seems to need more memory sooner or later. Newer FPGAs are now capable of embedding SDRAM controller cores to help alleviate this shortcoming.

While FPGA code can be reconfigured for new modes of operation and feature enhancements, it's usually much easier to make the more significant changes on a programmable DSP instead. Sometimes, a very small change can have a profound impact on the gate and logic cell topology.

System Example

Figure 2 shows a 32-channel digital receiver system suitable for a wide range of applications such as signal intelligence, direction finding, and signal tracking receivers. It consists of a Pentek Model 6230 VIM-4 mezzanine module attached to a Model 4294 Quad G4 PowerPC VIM processor board at the bottom.

The receiver module has four 14-bit 80 MHz A/D converters for digitizing IF or HF analog inputs entering through front panel SMA connectors. All four A/Ds feed a bank of eight quad digital downconverter ASICs with four channels of local oscillator, mixer and filter in each chip.

On board are two Xilinx Virtex-E FPGAs, each receiving the sixteen baseband signals from the four quad receiver chips. The

FPGAs are used to handle data formatting and channel selection for delivery through the VIM interface to the 32-bit mezzanine FIFOs on the processor board. Since the receiver signals flow through the FPGAs, they can also be used to perform demodulation and decoding functions to offload these tasks from the DSP.

All four A/Ds are connected to all eight quad receiver chips. Inside the front end of each receiver chip is a programmable crossbar switch that allows each of the four narrowband channels inside to independently select any one of the four A/D inputs.

This means all 32 receiver channels on board can independently select which A/D or antenna they are looking at. This provides a very dynamic antenna-to-channel assignment scheme for systems that need to adapt to changing traffic patterns.

The digital outputs of two A/D converters are delivered directly into each FPGA. This allows wideband A/D data to stream directly through the FPGA to the DSP since the high bandwidth of the VIM interface supports clock rates up to 100 MHz.

Even more important, it also allows the FPGA to handle signal processing algorithms on the raw A/D data before it goes to the processor, again, to offload some of its tasks. After handling the factory standard functions of data formatting and control, a high percentage of these resources is still available for custom applications.

The factory default code takes care of all the basic functions for most applications including selection of channels for delivery to the DSP board, selection of real or complex data modes, selection of receiver data or raw A/D data, and selection of data packing modes. For custom signal processing tasks, the optional design kit for each product includes the VHDL source code for all these standard factory default modes.

Users can extend the factory code by adding their own algorithms at appropriate points in the signal flow path. FPGA algorithms can be developed and simulated by drawing on the Pentek GateFlow Design Kit and on the wealth of IP cores and design tools available for these devices. Once compiled, the custom code can be downloaded through utility loaders into a non-volatile user memory on the board. □

Figure 2. 32-Channel software radio system.

