



Developing Advanced Wireless Devices with New Virtex FPGA Capabilities

New architectural features, tools, and development platforms facilitate the development of next-generation multi-mode/multi-protocol wireless devices.



by Louis Bélanger
Chief Technology Officer and Co-Founder
Lyrtech Inc.
louis.belanger@lyrtech.com

The SFF (small form factor) SDR (software-defined radio) development platform is a modular RF/IF/baseband platform (Figures 1 and 2) that showcases the latest silicon offerings from Xilinx and Texas Instruments (TI), advanced design flows, and software architectures. It also provides handset developers specific key features such as real-time power profiling and monitoring.

The platform was a joint development between Texas Instruments, Xilinx, and Lyrtech, as well as a host of leading software tool vendors. It features a Xilinx® Virtex™-4 device, which has advanced capabilities you can use to develop efficient and power-optimized designs.

In this article, I'll look at combined DSP/FPGA architectural and design trends and how they are embodied on the SFF SDR development platform. I'll also describe a simple Family Radio Service (FRS) FM modulation and a much more complex GSM modulation designed using a mixed design flow (using model-based development for the FPGA and C/assembly language for the system-on-chip DSP).



Figure 1 – The SFF SDR development platform modular platform

The SFF SDR Development Platform

The SFF SDR development platform provides a full signal chain from antenna to base-band processing. You can use the kit to create single- or multi-protocol handset radios for military, public safety, and commercial applications. The kit is also useful as a rapid prototyping and test platform. Additionally, because the platform is integrated to work with The MathWorks’s Simulink model-based design tool (MBD), you have the option to use C/HDL or MATLAB Simulink to quickly test proof-of-concept designs and optimize the architecture for cost and power.

Unlike other SDR development offerings on the market, the SDR development platform is a hardware/software co-development environment that supplies the complete components for a multi-protocol SDR device, including the RF front-end module, an analog-to-digital and digital-to-analog data conversion module, and a digital processing module. By separating out the base-band, IF, and RF as distinct modules rather than as a single fixed architecture, you can extend your radio development capabilities – as well as optimize for cost and power consumption – by substituting your own or third-party modules. This flexibility is vital, as it gives you the ability to adjust your products around the industry’s constantly varying requirements.

The baseband module features a Xilinx Virtex-4 SX35 FPGA and a TI TMS320DM6446 chip. The TI chip contains a TMS320CC64x+ DSP core and an ARM9 general-purpose processor core. The SDR development platform includes a

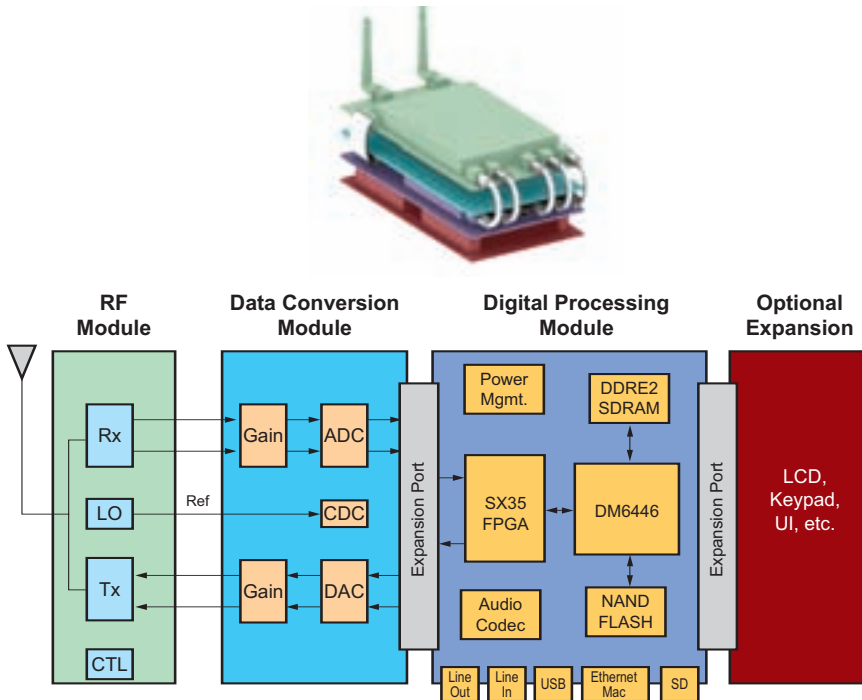


Figure 2 – SFF SDR development platform block diagram and technologies

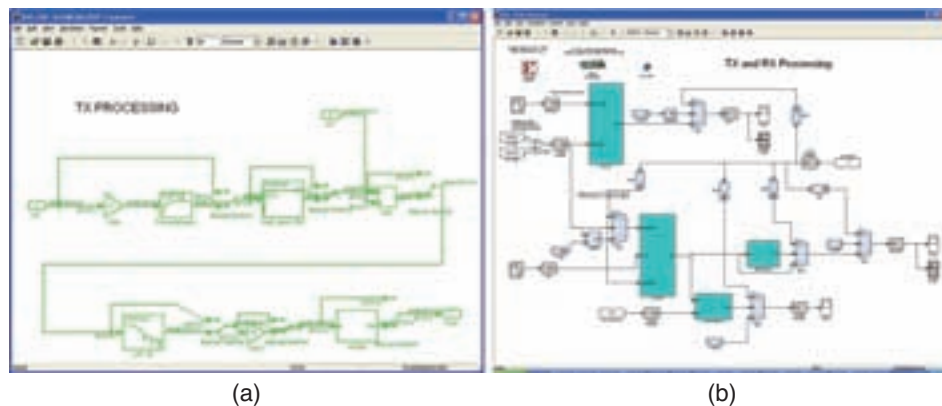


Figure 3 – Designing a simple FRS waveform in an “all-model-based design” approach. The DSP processing (Tx side) is shown in 3(a); the FPGA processing (Rx and Tx side) is shown in 3(b).

unique power measurement API. This API measures the loading of the FPGA, DSP, and ARM and reports real-time power data. This allows you to extract important information such as burst and peak power for a specific data rate, enabling you to accurately estimate battery life. You can also quickly assess the power impact of various system configurations. For example, you might experiment with different system partitioning between the FPGA and DSP to obtain an optimal power/performance balance.

FRS and GSM Application Examples

The SFF SDR development platform includes a basic application example: a simple FRS FM waveform. The example (see Figure 3) is designed entirely using a model-based approach in order to showcase the platform’s rapid prototyping capabilities. The application also shows how to partition an application between DSP and FPGAs, as well the effect of “shifting” different processing sections from the FPGA to the DSP (and vice-versa).

Figure 4 shows the FPGA part of a GSM physical layer implemented using a model-based approach in Simulink/Xilinx System Generator for DSP. This model is in fact an executable block diagram, where all signal processing functions can be simulated and verified using the sophistication of Simulink signal sources, channel simulation, output scopes, and data error-rate analysis.

The beauty of this approach lies in the fact that once simulations are completed, the model can be synthesized into an FPGA bitstream and executed with real signals.

Let's dig a bit into the development process, and more precisely to very specific DSP-related and timing implementation aspects.

The GSM project was originally targeted at the Virtex-II family – the same System Generator blocks are re-synthesized and re-verified using a Virtex-4 target. You can then see the first key benefit of having a model-based approach: easier device retargeting.

This retargeted implementation was good for a first pass, but it is in the interest of the design to use the more advanced capabilities of the Virtex-4 FPGA's DSP48 processing units, as more optimal designs will consume less power. Figure 5 shows a critical, high-speed part of the model: the transmit IF mixer, which runs at IF acquisition speed (104 MHz). Optimizing this section is key to power consumption. This was done by "tuning" the DSP48 micro-code (as shown on the GUI sub-window), which was programmed to execute a multiply add. FPGA resources are then much lower compared to Virtex-II devices. Power usage, verified using the power measurement capability of the platform, is also much lower thanks to this optimization, as well as the Virtex-4 device's overall better power consumption profile.

Figure 5 also shows another very interesting optimization capability of System Generator – retiming. This powerful feature lets System Generator introduce latching at appropriate places throughout the pipeline. This automatic retiming capability really makes things easier, especially for high-speed sections such as the direct digital synthesis (DDS) function.

The implementation of the GSM physical layer benefited greatly from the use of the Virtex-4 architecture and tools. We realized a much more optimized implementation in terms of FPGA resources and power consumption using the SFF's power consumption measurement capabilities. Further capabilities such as continuous power monitoring will allow you to characterize the power usage during operations, which might lead to additional processing optimization.

Conclusion

The SFF SDR development platform provides a very flexible platform for handset developers. Based on advanced processors from silicon vendors such as TI and Xilinx and software tools from leading vendors such as The MathWorks, it provides handset developers a true "Lego box" with which to build advanced products in the ever-accelerating and competitive – yet extremely promising – wireless devices marketplace. ●●●

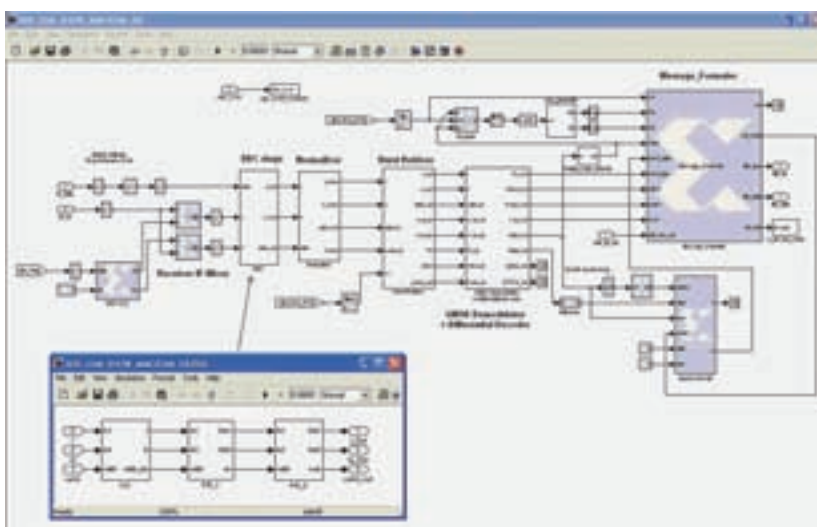


Figure 4 – FPGA model of the GSM physical layer

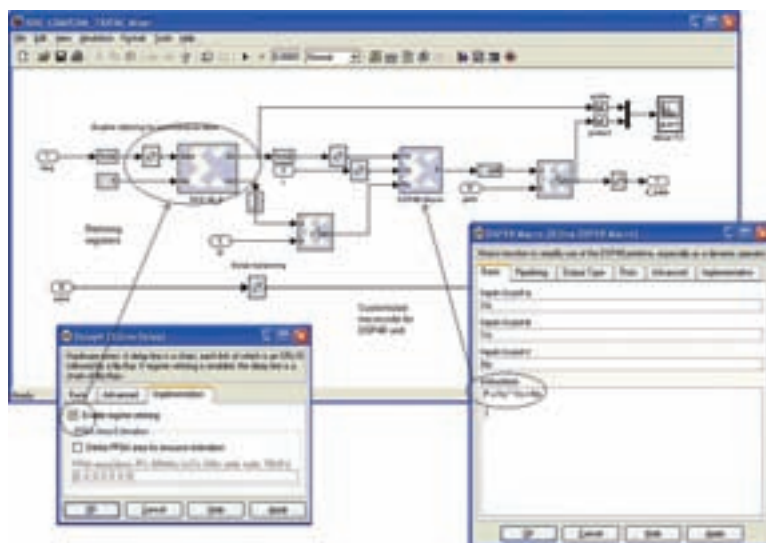


Figure 5 – Customizing the DSP48 processing units using micro-code