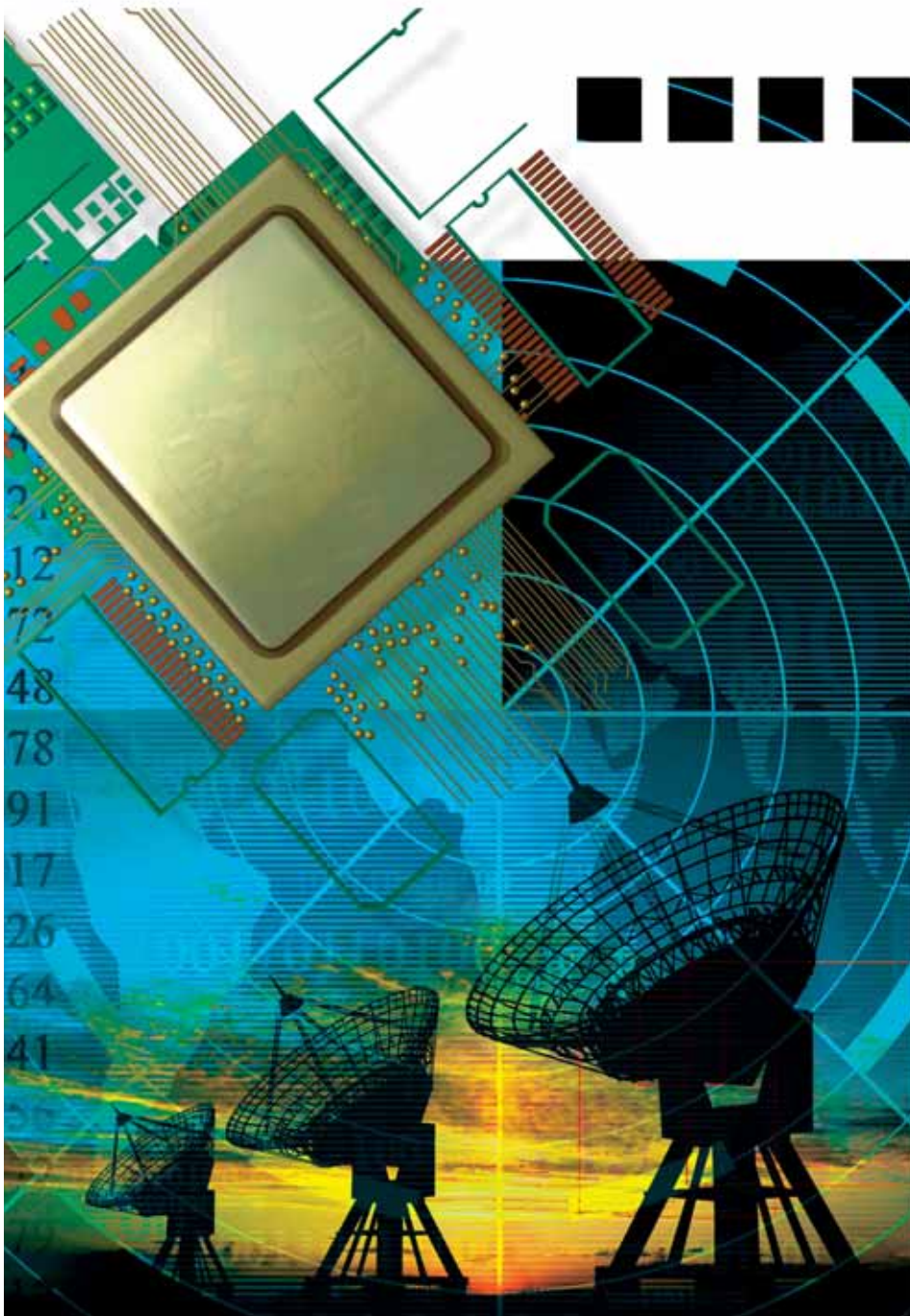


Accelerating System Development Cycles with the Radar Blockset Library

A radar blockset library offers components selectable from the generic Simulink environment for complex system designs using FPGAs.



by Meena Das
Scientist "E"
Electronics and Radar Development
Establishment (LRDE), DRDO
dmeenatish@yahoo.co.in

Karthik Kabbinahitlu Subrahmanya
Contract Engineer
Electronics and Radar Development
Establishment (LRDE), DRDO
karthik.urimajalu@gmail.com

Taniza Roy
Scientist "C"
Electronics and Radar Development
Establishment (LRDE), DRDO
tanzaz@yahoo.com

Gnana Michael Prakasam
Scientist "F"
Electronics and Radar Development
Establishment (LRDE), DRDO
lgmprakasam@yahoo.com

Designers often struggle with time-consuming development cycles during complex system realization using FPGAs. We have designed a radar blockset library in The MathWorks Simulink environment that addresses this challenge by providing pre-synthesized components.

Our radar blockset library components directly generate programming code for Xilinx® FPGAs on a target board. Thus, the blocks lead to a single design solution rather than having to use multiple blocks for dedicated functions.

For applications such as complex radar systems, a single design solution speeds up development activity and keeps designers from having to design from scratch. To highlight this feature, we'll describe how a typical radar blockset library component works in a radar receiver design.

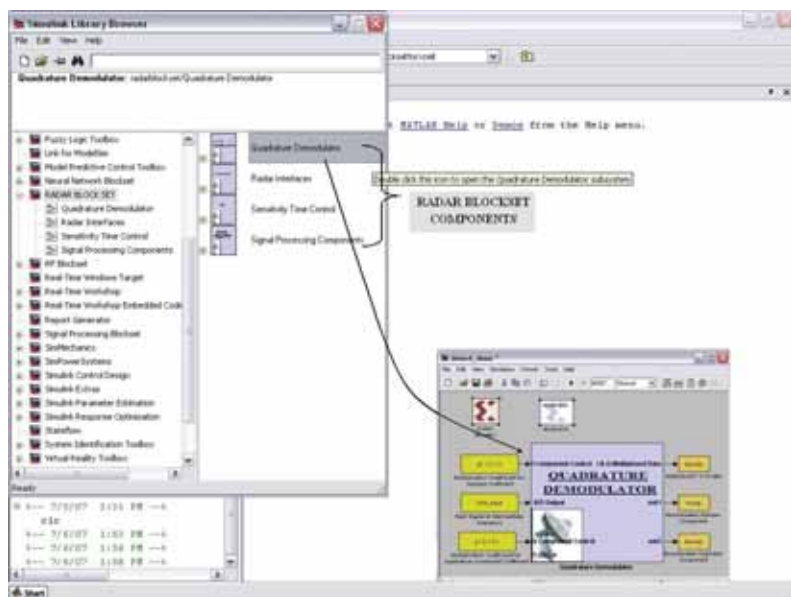


Figure 1 – Radar blockset library

You can use these library components just as you would use Xilinx System Generator components. The blocks are optimized for FPGA resource utilization and are pre-synthesized and compiled in the radar blockset library. Each component in the library is equipped with the required parameterization block and help menus for configuration and proper usage, just like System Generator block sets.

The first version of the radar blockset library comprises a quadrature demodulator for generating complex video from real samples, various radar interface blocks, a sensitivity time control (STC) block, and a signal processing block (Figure 1). Designers widely need these components when designing complex systems.

Our team designed the radar blockset library and its components using VHDL and the System Generator tool. It is exclusively intended for implementation on Xilinx FPGAs.

Integrated Library Components

Let's explain the principles on which we designed each of the components of the radar blockset library.

Quadrature Demodulator

The quadrature demodulator disintegrates real input sampled signals into in-phase and quadrature (I and Q) components. It then

demodulates the intermediate frequency-to-baseband signal.

You can use this component as a part of your receiver design and generate the bit file directly from the design model to program the FPGA.

STC Block

Sensitivity time control is a gain control method for radar receivers. Gain control adjusts the sensitivity of the receiver gain as a function of range, thereby regulating the intensity of the returns. STC causes receiver gain to vary with range in such a way that receiver output is less dependent on range, thus reducing the effect of clutter on receiver saturation. STC helps in increasing the dynamic range of the receiver and prevents receiver saturation by selecting suitable STC laws in real time. Even though complex logarithmic calculations are required to realize this function, this library component aids designers because they can use the library components in a new design model through simple drag-and-drop operation.

Radar Interface Block

The interface block comprises various components such as a global positioning system (GPS) receiver interface block, Manchester code inversion protocol interface block, controlled receiver interface block, and RS232 standard interface block.

GPS Receiver Interface Block

This module interfaces a system with GPS receivers. It expects the receiver to work according to the Trimble Standard Interface Protocol (TSIP). The component was initially designed to interface a radar system with a GPS receiver, but you can use it wherever a GPS receiver interface is required with the same protocol.

The initialization, request, and report packets are stored in FPGA memory as per TSIP. Initialization packets set the GPS parameters when the GPS is initialized. These packets decide the GPS's mode of operation, the filters to be used, and the I/O port status. After initialization is complete, the request packets are used for inquiring about the health and status of the GPS. Report packets handle all of the inquiries made by the system to the GPS receiver.

CMI Protocol Interface Block

This module serves as an interface between the beam steering unit (BSU) and another radar subsystem. The BSU message comprises a header word followed by descriptive words. The number of words that will follow is indicated in the header word. Standard CMI protocol is used for coding these messages. This module provides an interface solution for any system based on the CMI protocol.

Controlled Receiver Interface Block

This module interfaces systems to other systems by providing a control signal to initiate the reception process, according to the RS232 standard for radar applications. The block expects a reply based on a control signal within a fixed duration, which is decided by the next occurrence of a control signal.

RS232 Standard Interface Block

This component is designed to support communication controller implementation according to the RS232 standard for serial communication. It supports both transmission and reception processes. You can configure the component with a suitable baud rate for communication from the component GUI.

Signal Processing Block

This block comprises various signal processing functions such as pulse compression and Doppler filter processing. The pulse compression block involves the transmission of a long coded pulse and the processing of the received echo to obtain a relatively narrow pulse. The Doppler filter block extracts Doppler information from radar echo signals.

Interactive Component Help

Selecting the component provides the first level of information about it, similar to how other blocks are described in Simulink. Additional detailed information is provided as a help file in Simulink (Figure 2). These help files support designers in component configuration.

Radar Receiver Example

The quadrature demodulator component forms an important building block of modern receivers for the demodulation of intermediate frequency (IF) signals and generation of video signals. This component expects a sampling frequency four times the intermediate frequency. The sampled input signal at baseband frequency forms the real-time input to the demodulator component. In addition to this, the component expects the corresponding I and Q multiplication coefficients for configuring the library component.

The design of the quadrature demodulator component comprises block designs for functions such as filtering, quadrature component extraction, sign extension, decimation, and I and Q signal multiplexing (Figure 3). Selecting the appropriate filter in the digital domain for the extraction of the I and Q components is crucial in the design.

The radar blockset component model from Simulink can be directly used as a System Generator model to realize the quadrature demodulator. Configuring the component for the new design is as easy as dragging and dropping the component from the radar blockset library. The functional design model expects the required inputs in the proper format for hardware realization. Like any other System

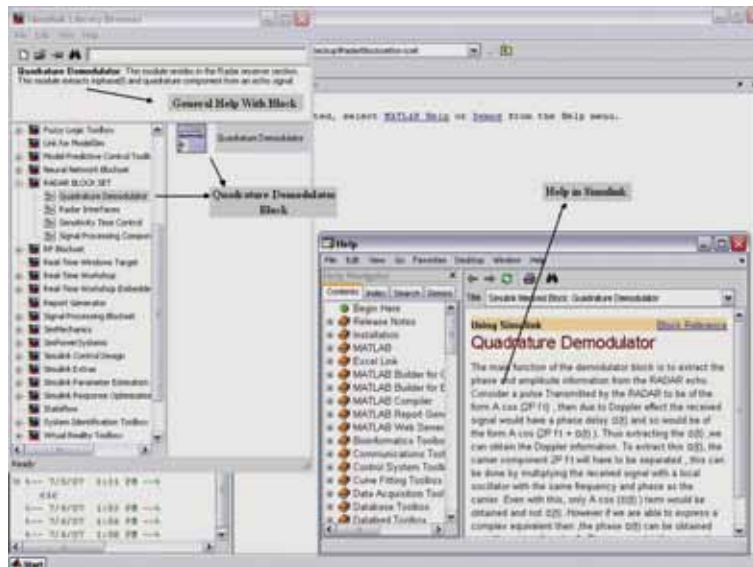


Figure 2 – Interactive component help

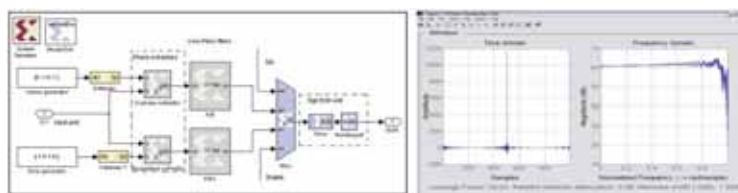


Figure 3 – Design model for the quadrature demodulator component, with its pulse compression result

Generator model, the component will generate code for Xilinx FPGAs, which you can then download to your target systems. This process is much faster than designing with discrete models.

Let's explain how you could use the component for a radar receiver design. Figure 1 shows a design model for a quadrature demodulator component in a typical radar receiver test case. The component inputs are taken as echo samples of linear frequency modulated (LFM) code at 20 MHz, an IF of 5 MHz, and a bandwidth of 2.5 MHz. The I and Q multiplication coefficients are provided as $[0 \ -1 \ 0 \ 1]$ and $[-1 \ 0 \ 1 \ 0]$, respectively.

We evaluated the model for logic and timing in the simulation environment by using ModelSim software, resulting in the generation of the required I and Q components. To check and validate component output, it is pulse-compressed with complex conjugate LFM code at baseband frequency (Figure 3). As the output

of the component is I and Q signal-demodulated at baseband frequency, the output of the pulse compression shows a perfect peak; thus the function of the component is validated.

Conclusion

You can configure generic library components in The MathWorks Simulink environment for any real-time application without altering other programming-level aspects. These blocks lead to a single design solution rather than having to use multiple blocks for dedicated functions.

Most of the design modules use IP core elements during implementation to reduce propagation delays. These customized cores deliver high levels of performance and area efficiency, resulting in the efficient implementation of logic by using fewer FPGA resources in less time. 🌈

We wish to acknowledge Soumitra Das and Sandeep Kulkarni from Avnet Pvt. Ltd. for their support in writing this article.