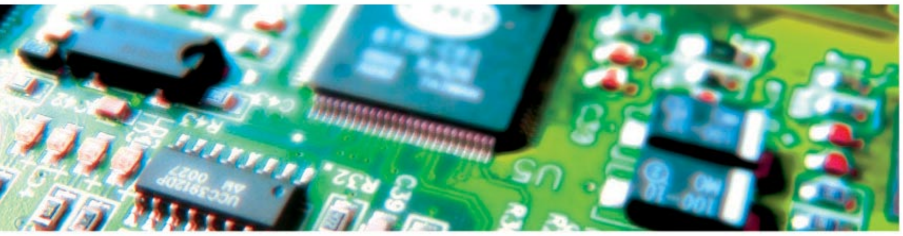

Infinite possibilities™.

CORBA for FPGA
The Missing link for SCA radios

Joe Jacob
Senior Vice President, Sales and Marketing
Objective Interface Systems
joe.jacob@ois.com

Maxime Dumas
Field Applications Engineer
Lyrtech
maxime.dumas@lyrtech.com

Introduction

As the first generation of software defined radios (SDR) built for compliance with the U.S. Military's Joint Tactical Radio System (JTRS) Software Communications Architecture (SCA) become available for deployment, radio manufacturers continue to innovate to create ever more powerful radios in smaller and smaller form factors. In order to drive down costs, they continue to increase their use of commercial off-the-shelf (COTS) tools. Their goal is to reduce development and deployment costs while enabling more flexible design options throughout the engineering process.

As radio builders research and develop new prototype radios, they discover they can never have too much processing power. As waveforms become larger and more complex, many radio builders are running into a performance wall as they try to manage multiple large waveforms at the same time. Modern SDRs often use a combination of general purpose processors (GPPs), digital signal processors (DSPs) and Field Programmable Gate Arrays (FPGAs) to provide the processing power necessary to build SDRs that meet their extensive engineering requirements.

Existing versions of the SCA (including the most recent edition, version 2.2.2) provide component flexibility for the GPP. One of the issues for the current publicly available standard is that it does not specify equivalent component flexibility for the DSPs and FPGAs. Recently, the JTRS Joint Program Executive Office has specified the Modem HW Abstraction Layer (M-HAL) as a mechanism for providing some of that component flexibility.

However, M-HAL is not a solution for the general market. It is restricted under ITARS (International Traffic in Arms Regulations) and thus the details of the technology are not available to the general public or to the international community of radio builders. It is also not a COTS solution, so that it cannot take advantage of the reduced costs and ubiquity and breadth of application that a COTS product provides.

The breakthrough

COTS software vendors have responded by providing high-performance off-the-shelf technologies to increase performance and provide broad, wide-ranging component flexibility on the radio across a variety of different hardware devices. An important part of these new technologies is providing a standards-based CORBA implementation for specialized processors. For the past few years, ORBs have been available for most DSPs, but FPGAs were left out up until very recently. Building an ORB for FPGA implies several challenges, mainly due to the fact that programming is still done at a very low level, and that designs are tightly linked to specific resources of the programmable fabric. (i.e. for FPGAs, there is no equivalent to the virtual addressing provided by operating systems). In spite of this, enough functionality can be attributed to the FPGA in order for it to be recognized as an SCA device. In fact, because of the parallel processing nature of configurable logic devices, the greatest advantage of an FPGA ORB is its performance, as it does not need to preempt the processor for a context change.

Objective Interface designed a CORBA solution for FPGAs that creates an abstraction layer between the developer's modules (components of a waveform) and the hardware communications interfaces. This solution, implemented in the Lyrtech Small Form Factor SDR Kit, allows the FPGA to be loaded with several components. What puts this solution at the state-of-the-art for component-based development is that it exploits the full potential of the partial reconfiguration feature of the Xilinx chips. By constraining the design of the various components to predefined areas of the FPGA at development time, it gives the possibility to instantiate partial bitstreams, and to individually load multiple resources onto the FPGA at run-time, acting more like an executable device (as opposed to a loadable device).

Allowing developers to do this gives flexibility on the granularity of the components to be created and is in line with one of the goals of the SCA; to develop and deploy waveform components distributed across multiple processors without any initial knowledge regarding where other components are located, whether they reside on the same processor, or on different ones. This does not mean that it has not been tailored for performance. In fact, one of the key features of the FPGA ORB is that communications between two components that reside within the same FPGA are simplified so as to minimize the latency as well as the overhead in logic. A sophisticated but lightweight connection infrastructure routes these internal messages directly to the appropriate FPGA component, while messages for CORBA objects that exist external to the FPGA are transparently routed to the ORB for full GIOP processing

Figures 1-a and 1-b illustrate respectively the architecture of current radios, based on SCA 2.2.2, and the architecture of the next generation of radios, implementing CORBA across all processors.

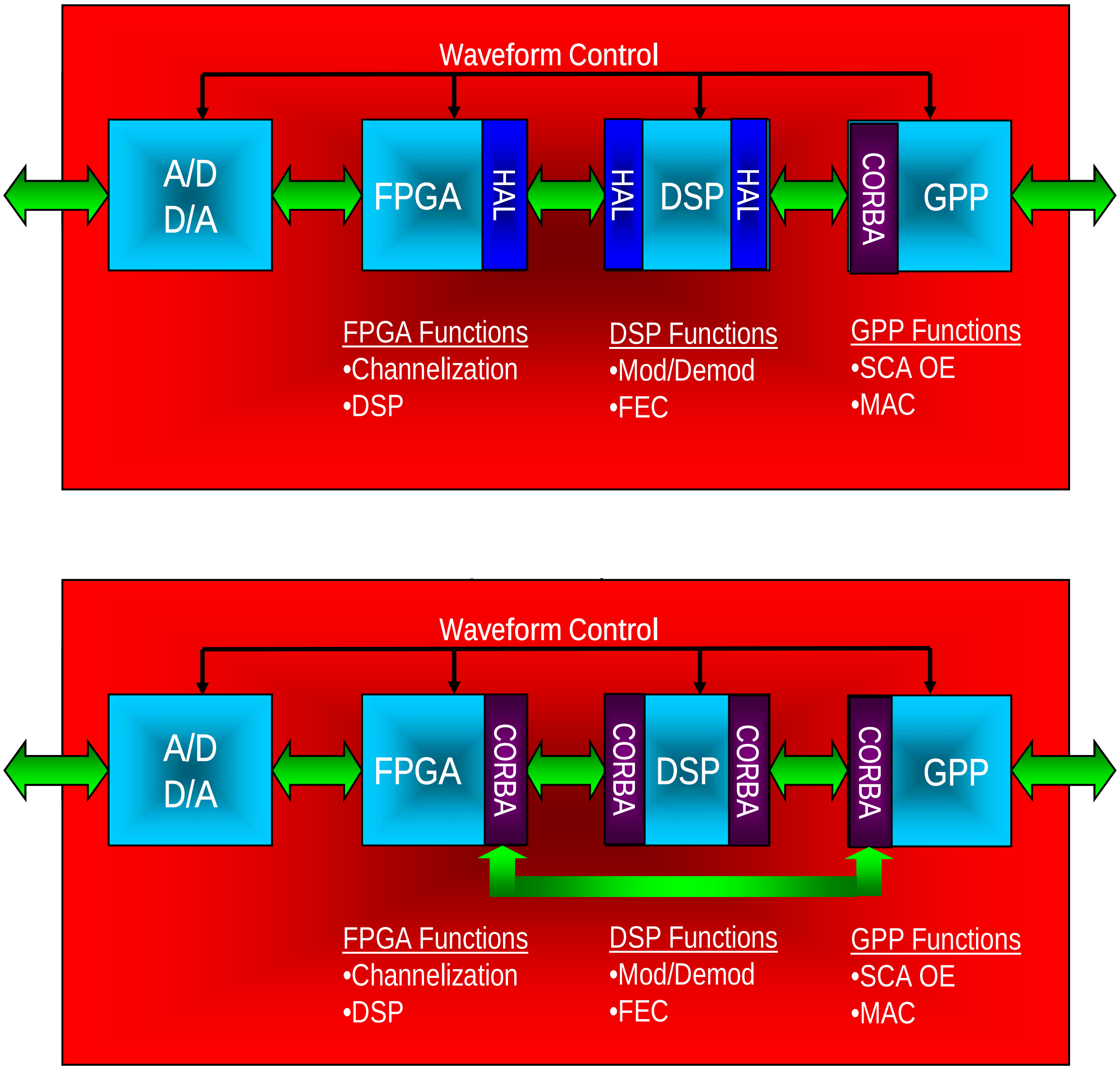

Figure 1: a) Architecture of current SCA-compliant radios. b) Architecture of next-generation, all-CORBA, SCA-compliant radios.

Figure 2 presents a detailed floorplan of how the CORBA solution for FPGA is laid out. In this particular case, the FPGA resides between a DSP and the A/D and D/A converters. For illustration purposes, parts of a spread spectrum waveform have been added in the figure.

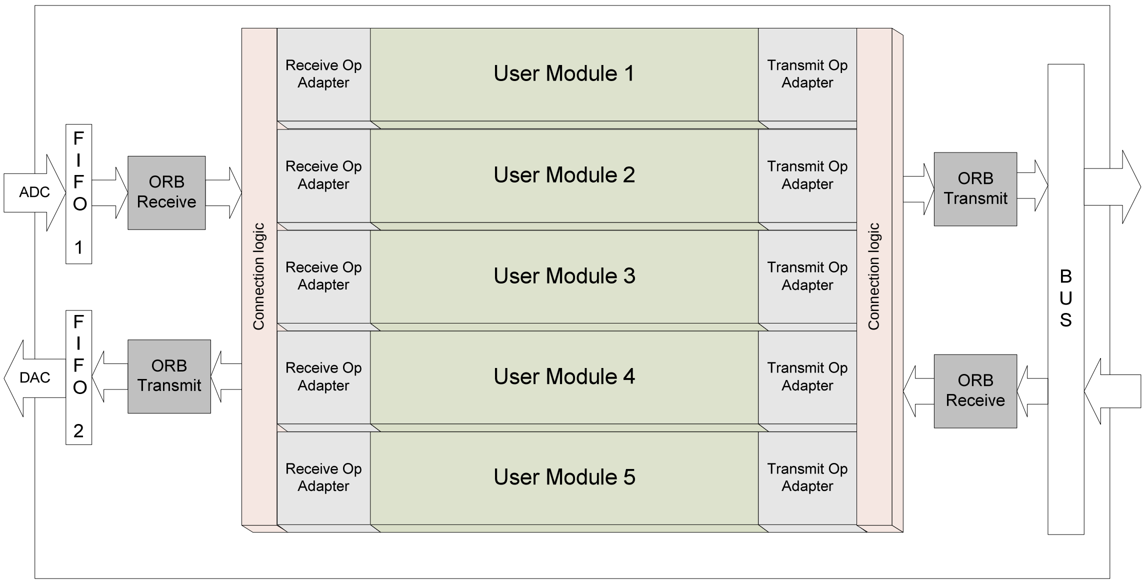

Figure 2: Floorplan of the CORBA solution for FPGA

Design flow

Although the design of the FPGA ORB is very different from the design of traditional ORBs, the design flow a developer needs to follow is somewhat similar. Starting with the source code, a developer first needs to define the IDL for a given component. Using the IDL-to-VHDL translator, some adapter code is generated, which has to be linked to the waveform component. Specifying pre-defined location constraints to the module and following the partial reconfiguration design flow will result in a partial bitstream that contains everything it needs to communicate with the ORB and/or with other components, and allows it to be dynamically loaded at any time. If a component requires more FPGA resources than a single placeholder offers, the developer can constrain its design to a larger area. Figure 3 shows the floorplan of a Virtex-4 SX35 containing the ORB (in pink), the transport adapters (in yellow and green), the usual I/O interfaces and glue logic (in dark red and blue), and a single SCA resource (in dark green).

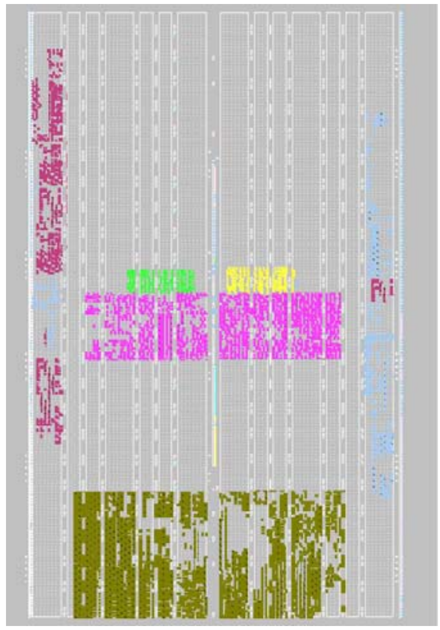

Figure 3: Place and route result of a waveform component and the FPGA ORB.

****
In order to make the FPGA ORB an off-the-shelf solution, its design needs to be platform-agnostic as much as possible. In fact, its only dependency is on the number of ORB instantiations, which depends on the number of different physical transports the FPGA will use to communicate with other CORBA-enabled devices.
****

The Platform

The FPGA ORB has been integrated into Lyrtech's Small Form Factor SCA Development Platform. The SFF SCA DP is the first COTS development platform to include a CORBA-enabled FPGA. The complete software tools integration allows developers to readily create SCA components using standard, non-restricted, hardware-agnostic communication interfaces for GPP, DSP, and FPGA processors. In this way, the partitioning of the waveform on the processors can be revisited at any time during development or after deployment without modifying the structure of the application; only the platform-specific implementation of the migrated CORBA operations need be considered.

Conclusion

Enabling CORBA on DSPs and FPGAs for SDR provides a variety of advantages for radio builders:

- In many development shops, the GPP, DSP and FPGA development groups work in isolation. ORB-enabled GPPs, DSPs and FPGAs provide a common communication description between disparate engineering disciplines.
- As the SCA is already based on CORBA, it extends the robust technology throughout the radio preserving SCA-compliance while increasing portability of algorithms and logic. Systems engineers can now move functionality to FPGAs *without* modifying the structure of their applications
- The system architect can migrate functionality from GPP to FPGA as needed
  - Assignment of functionality does **not** need to be decided early in the radio development process
  - Functionality can be tested on GPP, then migrated to FPGA as needed, without changing the main application. For the first time, entire unified radio designs can be done on a workstation and then piece by piece moved to the relevant processing device.
- Specialized interfaces for DSPs and FPGAs are no longer needed. System architects can maintain their focus on improving functionality in the radio while the ORBs provide the necessary interface details for the GPP, DSP and FPGA code developers.
- For the first time, full technology transparency is feasible in a heterogeneous system architecture for SDRs
- The technology is available today and has no ITARS restrictions, so can be used by worldwide development teams.

A number of different SCA-based SDR programs around the world are prototyping systems with ORBs for all devices in the radio, including DSPs and FPGAs. In doing so, they are reaping the benefits of high performance, low power and small footprint requirements through the use of this standards-based architecture.

FOR MORE INFORMATION, CONTACT US AT:

Lyrtech Headquarter - International
4495 Wilfrid-Hamel Blvd., Suite 100
Quebec City, Quebec G1P 2J7 CANADA
Phone: 1 (418) 877-4644 (Worldwide)
1 (888) 922-4644 (Toll-free USA and Canada)
Fax: 1 (418) 877-7710