Take on CORBA for multiprocessor SDR

By Joseph Jacob
Senior Vice President
Sales and Marketing
Objective Interface Systems Inc.
E-mail: joe.jacob@ois.com

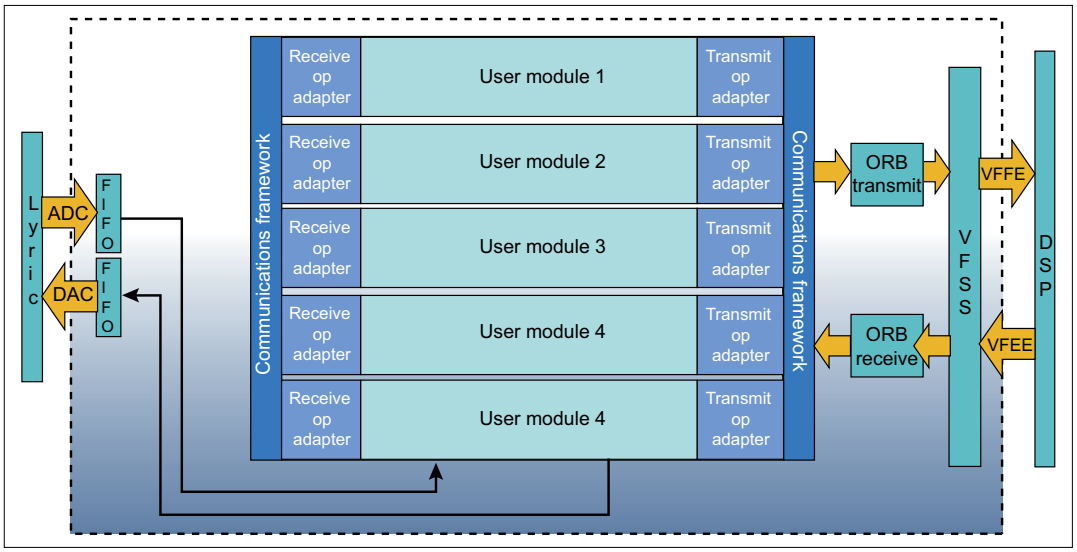

Figure 1: Radio builders can use a standard ORB for the GPP and an FPGA ORB to significantly improve the overall throughput of the radio.

The trend toward increasingly smaller and higher-performance SDRs means that radio designers must carefully select their component technology—both hardware and software. One technology that every designer should understand is the Common Object Request Broker Architecture (CORBA). CORBA is the required middleware layer for Software Communications Architecture (SCA)-based SDRs.

Commercial CORBA object request brokers (ORBs) are high-performance, small-footprint software products that provide the communications framework in single-processor and multiprocessor SDR systems. They promote the intelligent and optimal allocation of processing resources among GPPs, DSPs and FPGAs. Understanding the role of CORBA in these multiprocessor systems can save engineering resources, improve time-to-market, and increase performance and portability of small form factor SDRs.

A CORBA ORB is a software product that enables designers to simplify the development of distributed software applications; build scalable, efficient and robust applications; and reduce overall development time. An example is Objective Interface Systems' ORBexpress RT. A typical SCA SDR is built on ORB's distributed communications services and RTOS. The breadth of functionality provided by the SCA results in a sophisticated software platform that may occupy significant computing resources on the radio, placing a strain on size, weight and power (SWaP). This is one reason that small-footprint, low-power ORBs are important to small form factor SDR designs—they reduce SWaP on the system.

Many different efforts are underway to reduce the size and improve the performance of SCA implementations to enable SDRs to be implemented in ever smaller handheld devices with lower power requirements. Two ongoing efforts are SCA-Lite and CORBA ORBs in an FPGA.

'SCA-Lite'

SCA-compliance provides several benefits. Among them are software reuse, common hardware and software platforms to reduce production cost, and field upgradeability. At the same time, some users and developers have raised concerns about the cost of SCA-compliance in terms of size, cost and power. A number of companies interested in SDR have expressed concern that the current versions of the SCA may not fit into the very small form factors they plan on using for their SDRs.

Working with the SDR Forum, companies are reducing and modifying the required SCA components for systems that require a significantly smaller footprint than standard radios. The result is an "SCA-Lite" core framework for very small-form-factor SDR applications while preserving all core functionalities.

While form factor is important, radio builders are finding that 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. Software vendors have responded by providing technologies to increase performance.

One of them is to implement an ORB directly in a dedicated FPGA hardware. This can increase performance up to 100 times for selected functions. Radio builders can use a standard ORB for the GPP and an FPGA ORB to significantly improve the overall throughput of the radio. This will allow larger waveforms to be used in radios without overloading the capability of a standard GPP or DSP (**Figure 1**).

Multiprocessor platforms

Signal processing systems like SDRs often include multiple types of processors such as DSPs, GPPs and FPGAs. These disparate processors must be interoperable with one another, thus presenting several challenges. Most notably, communicating among the various hardware

What is CORBA?

CORBA is an open, vendor-neutral standard created by the Object Management Group consortium. It enables pieces of programs—called objects—to communicate with one another over networks, regardless of what programming language they are written in, what OS they are running on or where they are located in the system.

CORBA is often described as a "software bus," because it is a software-based communications interface through which objects are located and accessed.

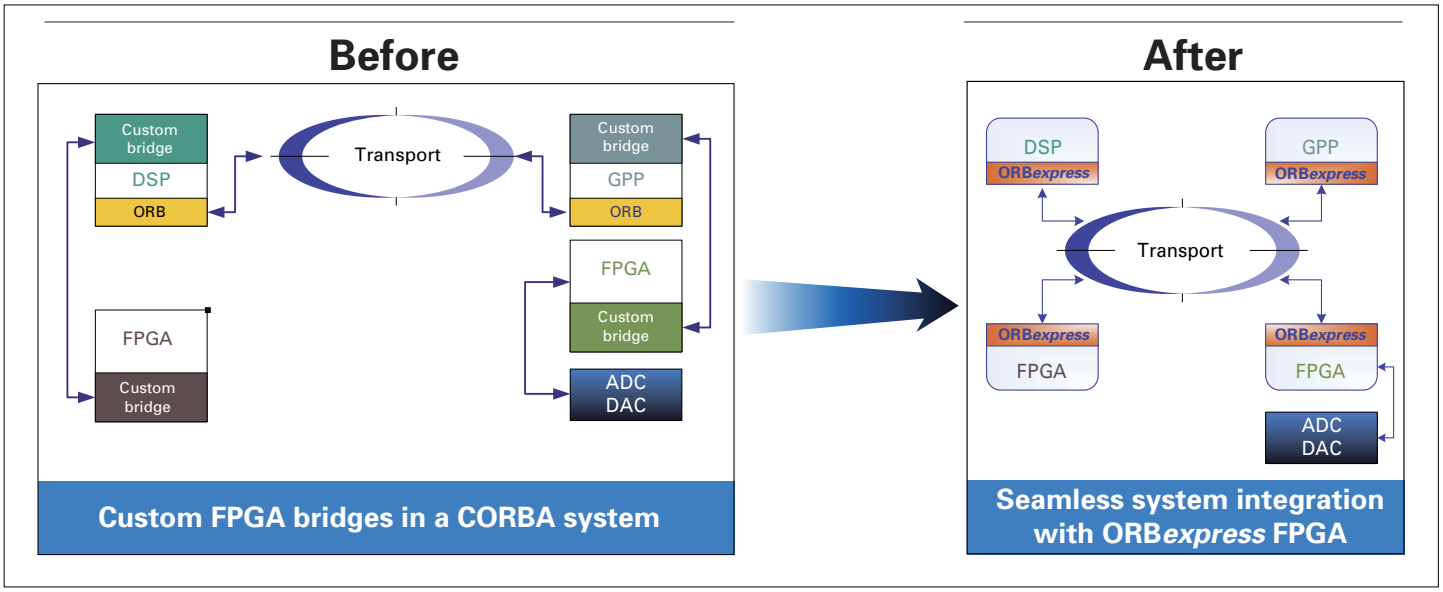

Figure 2: Using CORBA on the GPP, DSP and FPGA provides a COTS solution and common communications architecture for all three processor types.

architectures, OS and implementation languages presents a significant engineering effort replete with subtle behaviors impacting performance, security and data integrity.

Using CORBA on the GPP, DSP and FPGA, however, provides a commercial off-the-shelf (COTS) solution and common communications architecture for all three processor types. This architecture offers several benefits over the custom GPP-to-FPGA bridging solutions that are currently used to connect FPGAs with GPPs and DSPs:

- It promotes object mobility and location transparency.
- It improves throughput by limiting system size, weight and power.
- It allows designers to use high-performance COTS solutions instead of custom solutions that require a substantial investment in custom hardware and software (**Figure 2**).

Flexibility

CORBA ORBs provide location transparency and processing mobility, two features promoting the flexibility of SDRs. Location transparency makes it easier to move functionality or a piece of logic among processor technologies. It makes no difference whether an object is called within the same processor or on a remote processor. Engineers can re-allocate logic without having to redesign systems from scratch, which is an enormous savings in time, resulting in much faster time-to-market and better product adaptability in subsequent revisions. Processing mobility leverages location transparency, allowing designers to move logic to another processor with ease.

This means designers can build an SDR system on their workstation. They can then easily move component parts to the GPP, DSP and FPGA piece by piece, optimizing their system's overall performance.