



Free Subscription/Free Renewal | Online Reader Service Card | E-Newsletter

Web Exclusive Feature Articles

Software Defined Radio Solutions for DF Signal Processing Systems

Tim Owen, Spectrum Signal Processing

Introduction

Direction Finding (DF) is a term used for the electronic location (mainly), tracing and distinguishing of targets that are a source of electromagnetic signal transmissions. The technique can be employed in far ranging applications from radio hobbyists up to sophisticated military communications systems. DF is a long established technology but is gaining even more popularity and attention in the world of military (Signal Intelligence) and commercial communications.

DF systems employ methods of calculating the direction of arrival (DOA) of a radio transmission frequency using an array of spatially displaced antennae or a rotating antenna. Typically, an array of four to six (fixed) antennae is used although this number can vary. Many of the characteristics and the performance of a DF system are defined at the antenna array by their spacing, pattern and distance. This in turn usually dictates the type of DF algorithm to be implemented in the processor software. In high accuracy systems, this algorithm involves significant processing where a continuous data stream of digitized signals is stored, analyzed and tracked. In cases where the target or DF processing unit is mobile, accurate tracking of the target can only be achieved if these calculations are accomplished in real-time (i.e. all processing must occur without sacrificing loss of data of the incoming signal(s) of interest). This algorithm can be further complicated and processing intensified if targets are particularly agile or are attempting to 'camouflage' themselves to avoid detection.

As such systems vary in the number of antennae (dependent on the application processing and the type of antenna employed), scalability and the ability to perform efficient, cost effective system upgrade is key in the high-performance DF system for both the hardware and software. The requirement for these features is further strengthened by the necessity for upgrading systems both in the commercial communications world where new standards are constantly evolving, and in the military market where the required lifecycle of systems may outstrip the period that the hardware technology is available.

This paper examines the issues at the digitization and processing heart of the DF system and elaborates on the typical processing requirements of high-performance, multi-channel wideband DF systems using an array of fixed antennae. It highlights the main hurdles faced by the systems engineer in implementing the processing engine of such a system and then describes and suggests two COTS system solutions which are easily scalable and implemented using an SDR approach. One solution involves the implementation of a floating-point



Check out the contents of our current issue

- Editor's Choice
- New Products
- Buyer's Guide
- Web Exclusives
- Feature Articles
- Up Front
- Industry News
- Literature Review
- Website Reviews
- Corporate Profiles
- Technology Awards
- Links
- Advertising/Editorial Info
- Events Calendar

Home page

processor and/or FPGA system whilst the other solution details a FPGA-based processing system.

Finally, a software development methodology is briefly discussed and how, by using such a methodology, the DF system design engineer can evolve from system-level prototype through to development and deployment whilst preserving application software investment for future system upgrades.

Typical Requirements of the High Performance DF system

The requirement for a generic Digital Radio system for DF is that multiple receiver channels (i.e. antennae) are employed (where non-rotating antenna are not available or feasible). With the advent of Software Defined Radio (SDR), a more recent requirement is that simple scalability (at both hardware and software levels) is available. This facilitates software code and hardware portability for future system upgrades. Typical system requirements are as follows:

- Multiple ADC inputs (one per antenna) at 60 - 80 MHz sampling. This ensures that an IF of 30 - 40 MHz can be sampled whilst still meeting the Nyquist criterion.
- Detection of long distance targets or targets attempting to electronically 'camouflage' themselves requires a high dynamic range of the digitizing device. Typically 14-bits with SFDR > 80 dB. In addition, this dictates that floating-point processing is employed for the algorithms.
- Coherent sampling of all ADC channels so that phase information of the Signal of Interest is not 'lost' and accurate DF results are calculated. This requires synchronous clocking and triggering of the ADC devices.
- Digital down-conversion is usually required to provide a reduction in data suitable for real-time processing by a processor. This can be accomplished using a dedicated 'off-the-shelf' device or an FPGA.
- Raw input signal data sampled at up to 80 MHz stored into an optional large memory buffer (several seconds of each input signal is not unusual) for wideband analysis. At 80 MHz, 14-bit sampling, a 512 MB memory per channel will support up to 3.2 seconds of raw signal data storage. This facility can be used for wideband analysis of the spectrum several seconds after data has been captured.
- Option of the capability to store raw digitized data to hard disk (up to 200 GB of data) to enable analysis of large data sets and non-realtime offline access to the data.
- Facility for simple high bandwidth inter-processor communication to support the data transfer within the system. This is the key in any multi-processor DF system as it provides the mechanism of data transfer between processors involved in the front-end signal processing of an individual channel AND data transfer between processors of other channels. This multi-processing and data interconnect must enable the real-time system performance to be achieved.

To support real-time data streaming of the raw digitized data (required in systems where data down-conversion prior to the processing stage is not required) an inter-processor mechanism supporting at least 160 MB/s is necessary. This is the highest required data rate in a system sampling at 80 MHz - see Figure 1.

A Single Channel Architecture

The RF input signal from the antenna must first be down-converted by analog circuitry to an Intermediate Frequency (IF) that is more conducive to the sampling rate offered by 12 or 14-bit ADCs. This IF is typically of the order of a few tens of MHz and is dictated by the ADC technology available. As this paper is concerned with the processing heart of the DF system, it covers the path from this IF input through signal digitization, digital down-conversion and processing.

The analog IF signal is digitized by an ADC resulting in a digitized data stream of about 160MB/s (for an 80 MHz sample rate). In DF systems, it is of paramount importance that the ADC can be synchronized and triggered synchronously (within picoseconds) to the other ADCs in the system. Without this synchronization, the DOA calculations become inaccurate. This clock source is usually driven by an external high-accuracy clock source such as a TCXO.

The high data rate output from the ADC dictates that each digitized sample is processed by the signal processor within four or five processor cycles (for a 400 MHz or 500 MHz processor). This is clearly not enough processing time for FFT processing and DF algorithm processing and it is for this reason that digital down-conversion is usually employed.

Digital Down Conversion is a method of reducing the data rate to the DSP by relocating the digitized data band to a baseband frequency and decimating the number of samples, thereby increasing the processing time available for each sample. This can be accomplished using a dedicated 'off-the-shelf' device e.g. GrayChip or with the advent of new generation FPGAs, even wideband down-conversion can now be implemented within a single FPGA device. Optionally, down-conversion algorithms can be accomplished in software on a processor but this requires a high-speed processor (and external bus to stream the raw digitized data into the processor) dedicated to this task. This is not a cost efficient approach to down-conversion given that FPGAs and COTS digital down-converters are typically cheaper than the processor.

Figure 1 shows how the data processing of a DF system is split between front and back end sections. Front-end processing typically executes FFT functions on the incoming data of each channel to analyse the spectral content. The results of these FFTs are scanned for Signals of Interest (SOI) that may be emanated from possible targets. Once a SOI is detected and selected by the front-end processing stage, intensive floating-point processing is then performed on the SOI frequency from all channels in the system by a back-end processing stage which calculates the DOA from the signal data from all antennae.

These requirements give rise to the architecture as shown in Figure 1, which illustrates typical data rates in such an antennae channel. To ensure that the system can be applied to varying DF type applications (i.e. varying sample frequencies, down-conversion rates and numbers of antennae) an overall requirement of the system is that the hardware and software solution should provide this architecture in a scalable and flexible manner to allow for extra channels and processing to be implemented.

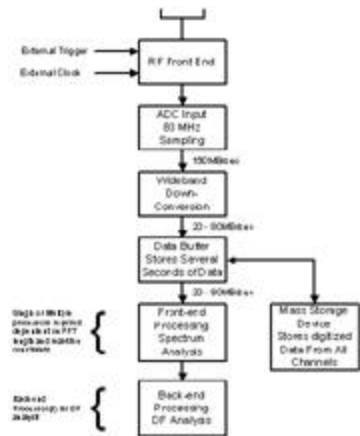


Figure 1: Single Channel Digitization and Processing Architecture

Complete System Architecture

The single channel model of Figure 1 scales proportionally for each antenna in the system. Note that an inherent requirement of the DF system data flow requires that the spectral results of all channels (i.e. after front-end processing) are transmitted to all back-end processing stages. This is the classic 'corner-turn' architecture also observed in radar systems. This architecture is required by the DF DOA calculation algorithm regardless of the DOA algorithm implemented, whether it is an adaptive beamforming or phase interferometer based algorithm. This system data flow can be difficult to construct, particularly if each of the front-end and back-end blocks consist of groups of processors, all requiring access to the data streams.

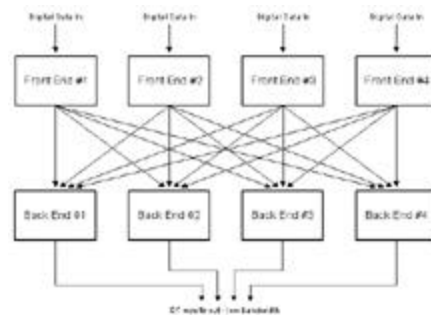


Figure 2: Example Processing and Dataflow Architecture of Four Channel DF System.

System Configuration and Inter-processor Communication Issues

Traditionally, it has been common to construct DF systems with custom or semi-custom hardware which is modular at the system channel level. This has the disadvantage that should any component in the channel become obsolete or require upgrade to meet a new specification, a new hardware (and possibly software) design cycle is required. However, implementing a lower level modular mezzanine standard (e.g. the PMC standard) for system construction of I/O and processing gives a 'building block' approach and provides high scalability of

the system as well as facilitating the simple implementation of future technologies (e.g. processors, FPGAs and I/O), simply by the replacement of the relevant modules. This provides preservation of some of the hardware investment when upgrading becomes necessary.

The use of a widely adopted industry standard module for the external I/O interfaces also provides the advantage of compatibility with the growing number of third-party, COTS modules that are available on the market for most varieties of digital and analog I/O. This becomes particularly useful when the system may be required to connect to external interfaces for legacy reasons.

Construction of the multi-node system architecture shown in Figure 1 and Figure 2 with the 'corner-turn' architecture and data rates of up to 160MB/s poses problems to the system designer, and is compounded further as the number of channels in the DF system increases. This data rate far exceeds the capability of industry standard backplane buses such as VME and PCI, so other solutions need to be considered. The need for such high data bandwidths and the flexibility required for system re-configuration and upgrade can be neatly addressed by the use of point-to-point data links. The deterministic nature of the point-to-point link permits quick and clean re-engineering of system data flow without the need for expensive crossbar type architectures and the latency introduced by the required supporting software. Traditionally, it is also often the case that various protocols (e.g. FPDP, PCI) need to be implemented within the system to meet the varying data bandwidths required. This adds further to headache of system software implementation and system upgrade, in turn heightening the desire to have a single uniform implementation of data interconnect within the system or at least a uniform software interface for all data paths.

However, the majority of current high-speed processors have no built-in facility for high bandwidth inter-processor communication. This is largely driven by the commercial communications market where applications and (voice) channels are higher density but narrower in band, enabling many channels to be processed on a single processor. Inter-processor communication therefore becomes a secondary requirement to the processor manufacturers. It is for this reason that COTS vendors are looking at developing ASIC-based solutions to solve the challenge of implementing system data communications.

The COTS SDR Solution and Technical Requirements. Software Defined Radio (SDR) Concept and Hardware

The concept of SDR was intended to provide radio transmitters and receivers that offer software control of modulation (and de-modulation) algorithms and bandwidth selection, while also controlling communications security (e.g. encoding and decoding). This control by the use of software permits the application of SDR to many radio-based applications, both commercial and military. The main incentive to engage a SDR methodology is that the software control enables the radio system to be re-programmed to provide flexibility for compliance to all current, evolving and future standards whereas historically a radio was constructed of fixed analog components and was tailored to the specific application.

As the software fundamentally defines the 'personality' and functionality of the radio, the use of a uniform hardware platform becomes possible across applications and coupled with the correct software and hardware strategy, also facilitates hardware upgrades as new technologies become available.

Ideally, the SDR can also be reconfigured "on-the-fly" to alter functionality and transform the characteristics of the radio in response to fluctuations in parameters of the incoming signals and/or volume of signals being analyzed. Realistically, the objective of the 'total SDR' (i.e. software accomplishes all of the system processing) is not yet achievable with present available technologies and a blend of processor software, FPGA firmware and possibly ASICs is usually required.

A COTS DF System Approach and Implementation

The advent of SDR specifications has meant that COTS systems established with standard backplanes, FPGAs and processors become a more cost-effective strategy for the system architect. The engineer is no longer required to discard system hardware as the system evolves technologically since the re-engineering and upgrade is much more of a software effort. In practice, however, it is likely that elements of the hardware need to be replaced and upgraded as technology advances (e.g. ADCs) and system requirements increase. To ensure that the minimum quantity of hardware is made redundant at each system upgrade stage, it is important to ensure that the system is constructed in a modular fashion. This also has the advantage of making the system easily scalable.

As previously noted, many COTS vendors are producing their own solutions to solve the complex nature of the DF system data flow. Spectrum Signal Processing has developed the Solano(tm) [3] Communications IC to address this issue. Solano essentially forms communications channels between a generic processor bus interface and other Solano ICs (via an LVDS interface) within the system. Also, the chip has an internal DMA engine enabling processors to spend minimal time involved in data movement and more time processing data. The Solano chip provides for four such dedicated data links, each capable of up to *200 MB/s full duplex communication*.

Arming each of the system processors with such a communications IC provides high-speed low-latency links to other processors. The bus interface is a fundamentally generic and simple interface whereby each of the *quicComm* links is 'seen' as a memory location (FIFO) by the master device. This expands the application of the data link mechanism beyond processors allowing ADCs and DACs to read and write the Solano with minimal (if any) glue logic. This enables the whole of the processing heart of the system to be constructed with a common interconnect fabric which in turn enables simplification of the overall system software because a uniform software interface is employed. As well as providing the facility to link I/O and processing nodes directly, the implementation of a standard data interconnect between modules also offers the potential to build heterogeneous processing architectures by using different processors on PMC modules.

A modular approach to system construction of wireless processing systems can further build upon the 'point-to-point' data link concept whereby links can be routed off a module (e.g. PMC) in the system. Using an extra low-profile connector ensures that the module complies with the limits of the standard specification but offers the enhanced capability of direct data links to the PMC carrier board, providing an alternative data path to the standard bus interface. This 'enhanced PMC' or ePMC modularity forms a flexible and scalable module concept around which systems can be designed.

A prime example of this modular construction model is Spectrum's flexComm ePMC-Carrier board and related ePMC modules. These products provide an ideal architecture for floating-point based DF processing applications. The ePMC-Carrier exists as either a single or dual slot VME64x board offering easily scalable

modular I/O and processing by supporting up to five module sites. Four of the module sites are interconnected with Solano quicComm links and links are also made available at the VME P0 connectors and the front panel for expansion of data connectivity between boards within the chassis and even between chassis. An embedded PowerQUICC processor capable of hosting VxWorks is also supported on the main board for 'stand-alone' operation and control of the modules via a local PCI bus as well as an optional RACE++ interface for connection to legacy equipment or use as an alternative data connection.

Analog - Digital Converter Solution

The typical requirement of 60 - 80 MHz sampling at 14 bits per sample requires a leading edge ADC such as the AD6645 (14-bits at 80MSPS). To meet the sampling coherency requirements of a DF algorithm, the ADC should have the option to be clocked and triggered by an external source.

COTS Modular 'Building Block' Solution: ePMC-2ADC: dual ADC channel ePMC module.

Data Decimation (Digital Down-conversion) Solution

The requirement for flexible, programmable digital down-conversion would suggest the need for a programmable DDC such as GrayChip GC1012A Wideband DDC [1] or GC4016 [2]. These currently provide the cheapest and least power hungry solution while providing a certain amount of flexibility. However, with the advent of FPGAs that are ever increasing in size and speed, implementing a DDC within a FPGA (particularly multiple narrowband DDCs) has become a cost effective solution and provides the system architect with the ultimate in flexibility and the ability to provide extremely high density in the future.

COTS Modular Building block Solution: ePMC-FPGA.

Processing

Digital Signal Processors e.g. Texas Instruments' 'C6000 family are ideal for baseband processing in a radio system and provide software design using Assembly language or a high level language such as 'C' but are mainly available as fixed-point processing only. Despite future generations running at 1GHz+ and having built-in wireless processing related capabilities (e.g. Viterbi and Reed-Solomon en/decoding) they are less applicable to the high performance DF system where a large dynamic range may be required.

RISC processors such as the PowerPC MPC7410 are currently available at up to 500 MHz and provide this floating-point capability while not compromising the speed reduction usually associated with floating-point DSPs.

1) Front-end FFT Processing

Two MPC7410 processors can be implemented on a single ePMC module. This can provide sufficient FFT processing for two channels (dependent on frequency resolution and real-time processing constraints) with a memory buffer (up to 1GB) for shared data storage, as well as 2MB of cache memory per processor. Should greater processing capability be required, this can easily be facilitated by the addition of extra processors (modules) and using the Solano quicComm links as the data conduit between the modules.

2) Back-end DF Processing

The back-end processing section is responsible for the DF detection and extraction processing. In a four channel system, four of these processors are fed by the 'corner turn' mapping from all of the FFT results of each antenna channel and conventionally, more processors can be pipelined behind these four processors to perform post processing and data collaboration of the DF results.

The amount of processing required in the system at the front and back ends of the DF channel depends on the algorithm to be implemented and the real-time constraints placed on the developer by the system application. However, for example purposes, it is assumed that one PowerPC processor is required for the front-end processing of each channel and one processor for the back-end processing of each channel. Processing can be appropriately scaled as required by furnishing extra ePMC modules (and carrier cards, if necessary).

COTS Modular 'Building block' Solution: ePMC-PPC.

"Real-Time" Wideband Data Storage

Several seconds of storage of the raw input or down-converted I/Q data (at up to 160MB/s) requires a memory of up to 1GB (equivalent to 6.25 seconds of raw digitized data storage) per channel. This can be used to store the data for wideband analysis several seconds after signals have been captured.

COTS Modular 'Building block' Solution: ePMC-PPC (up to 1GB of memory).

Mass Data Storage (Hard Disk)

A mass data storage device (i.e. hard disk) for down-converted data from all channels can be used for offline analysis in non-real-time. Due to the varying standards available for disk access (e.g. SCSI, UltraSCSI etc.) and the modular PMC standard approach taken, this requirement can be met by a third party PMC product. Data from any point in the channel architecture after digitization can then be streamed to a hard disk via the local PCI bus and third party PMC.

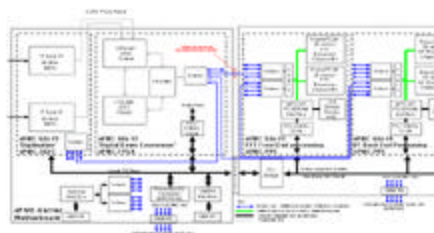


Figure 3: Example Architecture of COTS Dual DF Channel Implementation (Two VME Slots)

System Scaling for Increased Channels and/or Processing

From the basic analysis and requirements noted previously, a four channel DF system can be constructed within five VME slots. This solution builds on Figure 3 and adds an extra carrier card with 1 - 4 processors for post processing (scalable) after the back-end DF processing. The entire five slot solution contains four ADCs (and pre-processing FPGAs), FPGAs for digital downconversion, two PowerPCs for front-end and back-end processing of each DF channel and post processing processors. Note that the embedded controller on the ePMC-

Carrier removes the requirement for a separate embedded host controller board, which in turn reduces the total system cost.

All data interconnections between ADCs and DSPs can be accomplished using on-board or back panel cabling (Solano links), thus minimizing the need for front panel cables and reducing electromagnetic emissions.

With each dual VME board set realizing two complete DF channels, the scaling of the system becomes very modular and upgrading becomes extremely simple. Processor scaling also becomes straightforward with the addition of extra processing modules and/or carrier cards. In this way the processing of the system is easily scaled to match the system processing requirements of the DF algorithms to be implemented.

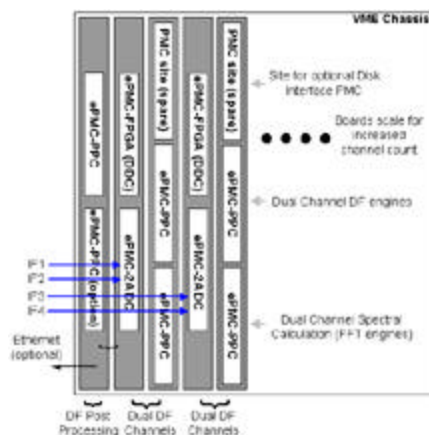


Figure 4: Example Chassis Front Panel of a Four Channel DF system

Software Development Tools

Development engineers face challenging decisions when selecting software development tools. Traditionally, high-level tools speed up development and alleviate some effort of upgrading and maintaining at the expense of more inefficient software implementation. Low-level software is more closely coupled to the hardware resources but requires more intense effort at the outset of software development and raises more headaches in the upgrade phases. Systems are increasing in complexity while market pressures demand faster project completion and flexibility in system configuration. For systems with longer periods of deployment, component obsolescence dictates that software portability and ease of maintenance are vital.

Ideally, a subtle balance of the two models would serve the developer best, allowing rapid system prototyping for 'proof of concept' and then refinement to optimize the system to the underlying hardware and maximize software efficiency. However, such an 'ideal' tool should not impose on the developer and not restrict the blend of the two extremes.

Such a software environment can ultimately produce substantial gains in productivity without compromising the runtime efficiency of the application.

Software Development Environments

Conventionally, the software developer has been required to develop host and target processor software separately with different environments, often dictating different programming models. quicComm is a multiprocessor communications 'C' environment from the flexComm product line and differs from traditional run-time solutions in that it is common to both the host and the target system. It is therefore independent of the OS of both the host and target processor. quicComm allows users to develop their applications in an environment that is abstracted from the hardware while allowing access to all low-level hardware features.

The quicComm software model uses software data links between host, processors and I/O for an easy system-programming model. The quicComm API calls are identical irrespective of the underlying data transfer hardware (independent of underlying hardware mechanism (e.g. Solano, PCI, VMEbus) ensuring that the developer is truly abstracted from the system hardware. This abstraction ensures that the software migration effort from current generation hardware to future generations is reduced. However, quicComm does not restrict the application being written in 'C' or assembly language or the use of an RTOS, if required.

A higher-level graphical data-flow programming environment or 'framework' for design of system waveforms is also desirable particularly during the prototyping phase where rapid 'proof of concept' is called for. Using a drag 'n' drop block diagram interface supporting the complete software development process from algorithm design and prototype through to production. The drag 'n' drop blocks are separate pieces of code that can be either 'C' code segments written by the developer, imported from third-party tools (e.g. Matlab) or optimized code segments from library routines (e.g. VSIPL libraries). This provides the code developer with control over the balance of optimized and less optimized code and thus control over the time to system software completion. Many engineers find this visual programming concept a natural method of constructing an algorithm or system design.

Ideally, this framework leverages from existing underlying software, (i.e. quicComm), with real-time interfaces and code import utilities. The block diagram implementation of the framework and the run-time environment handles all inter-processor communication automatically, and the provision of debug tools for the fine-tuning of system performance. Employing quicComm as the underlying inter-processor data transfer driver ensures that the system is independent of the target hardware and a highly efficient use of multi-processor system is developed. In this way migrating the application to future hardware platforms whether they are processor or FPGA-based, is not only made feasible but simpler.

The Prospects for SDR

Processors offer the most straightforward software programming with excellent debug tools but algorithm processes that can be partitioned to operate on several signals in parallel can be better implemented in an FPGA device. This algorithm partitioning enables the parallel calculation of many more MAC cycles in a single FPGA device than on a processor. The FPGA has matured significantly in speed, I/O capability and size in recent years to the point where it is now possible to implement processor cores and DSP-based algorithms within a single FPGA. Certain FPGAs such as the Xilinx Virtex II family incorporate internal logic specifically aimed at implementing DSP algorithms efficiently.

These technology developments mean that it is now realistic to accomplish all required processing of an SDR system in FPGA devices. This offers the ultimate in flexibility to the developer implementing a combination of off-the-shelf and custom processing blocks. A total FPGA processing solution offers 'on-the-fly' reconfigurability of the system processing as well sustaining hardware investment through the system upgrade phases.

Many new system specifications are defining high-availability ('five nines' or 99.999% uptime) and high reliability as pre-requisites. This dictates that hot-swap capability is implemented to minimize system 'down-time' which in turn means that many SDR systems are going to be cPCI based (only cPCI supports hot-swap capability). The Compact PCI specification provides for a transition module for connection to the back of the cPCI backplane. Connections to the main cPCI board or 'blade' can be made via the the cPCI J5 user defined connector. This transition module PCB can then be used to mount I/O devices such as ADCs and/or DACs. This still provides modularity of the I/O and processing and with the cPCI specification also allowing 'hot-swap' capability, system down time is minimized should repair be required. This is becoming an increasingly popular requirement in commercial markets but also has legitimate advantages in military systems.

Current industry standard buses such as VME and PCI are already reaching their bandwidth capacity for the newest system architectures. Specifications for a switched packet backplane (e.g. PICMG 2.16, Rapid IO, Serial RapidIO) are being introduced for data transfer, leaving the 'standard' backplane assigned for functions requiring less bandwidth e.g. control.

Discussion groups such as the SDR Forum [4] consisting of many of the world's leading companies with a vested interest in SDR are working together to determine the software standards and interfaces of SDR. Emerging high-level software such as CORBA is being considered to increase the rate of software development by creating a software blend of optimized processing algorithm libraries and high-level system development while maintaining software portability and guaranteeing interfacing to other hardware. This insulates the system designer from changes in the underlying hardware.

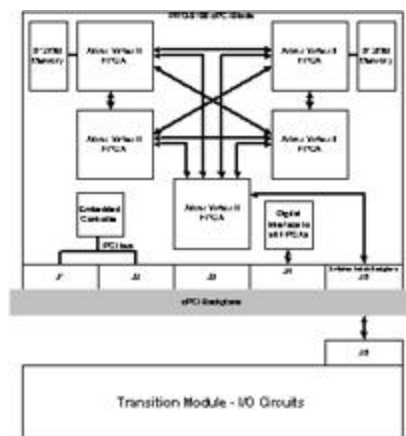


Figure 5: Example Next Generation COTS FPGA based Processing Platform

Summary

SDR provides an ideal platform for the development, maintenance and evolution of DF systems. COTS platforms based on a modular concept address the hardware and key software building blocks (i.e. tackling I/O and inter-processor data communication) to allow the developer to focus efforts on the algorithms specific to the DF application. The use of a uniform hardware interconnect and the correct combination of high-level and lower-level software can provide genuine benefits in reducing time to market and upgrading the system.

Reference Documents

The following are the primary references for this Whitepaper.

REF [1] GrayChip 1012A Wideband Downconverter Chip datasheet.

<http://www.graychip.com/pdf/GC1012A.pdf>

REF [2] GrayChip 4016 Multi-Standard Downconverter Chip datasheet.

<http://www.graychip.com/pdf/GC4016.pdf>

REF [3] Solano IC Datasheet.

<http://www.spectrumsignal.com/Products/solano.pdf>

REF [4] SDR Forum Website. <http://www.sdrforum.org/>

Spectrum Signal Processing Inc.

Suite 200, 2700 Production Way, Burnaby, BC V5A 4V7.

- [Email for more information](#)
- [Company's other products](#)
- [Email to a colleague](#)
- [Printer friendly format](#)

Advertising/Editorial Info | **Contact Us** | **Feedback**