FIZZWARE – A HAL APPROACH APPLIED TO SCA

G. Mourikas [†]; Dr D. Lund [††]; Prof. B. Honary [†];
[†] Lancaster University, Communication Department;
[††] HW Communication Limited Ltd

ABSTRACT

Software Communication Architecture (SCA) is a vital part of the Software Defined Radio (SDR) with the right initiative to evolve the compatibility and upgrade-ability of a SDR radio system. Fizzware was developed primarily for a reconfigurable system which required fast and partial reconfiguration. This paper describes the integration of Fizzware functionality to the existing SCA architecture. The result from the integration will allow SCA to provide services to the System Integrator, Network Integrator and therefore to the Communicator in the area of fine-grain reconfiguration. With this integration, the SCA architecture can use this embedding for current and future reconfigurable devices. This paper describes SCA and Fizzware and the architectural benefits that this merger of mechanisms can potentially benefit real time reconfigured SDR systems.

1. INTRODUCTION

Software Communication Architecture (SCA) is considered a middleware in Software Defined Radio systems. SCA combines three functionalities (1) information gathering (using CORBA) from an XML information pool called Domain Profile; (2) utilize instructions from the developer for the application chain structure ; and (3) management of assessment and download process of the application XML files, included in the Domain Profile, to the available reconfigurable hardware. There are issues with SCA that have been identified when approaching:

1. The partial chain/block reconfiguration on Field Programmable Gate Arrays (FPGA) and FPGA hybrid processing devices are not supported in detail.
2. All the combinations and possible dynamic behavior are pre-synthesized and simulated so the reconfigurable application is not dynamically reconfigured and there is no active control/decision making explicitly defined in this context, within the SCA architecture.

Fizzware proposes an addition of enhanced extensible library definition and API to enable scalable libraries and improve hardware classification specially on the application chain blocks that are desired to be reconfigured but are at the same time vital in the real time exchange of information. Fizzware provides hooks for advanced intelligent/decision making to determine the structure of configuration code loaded on reconfigurable devices.

2. SDR BASED SYSTEM

SCA being middleware architecture that provides guidelines on how to implement a SDR platform is located on the lower levels of the Software Defined Radio system block diagram shown in Figure 1. Description of components in Figure 1 follows using actors from the value chain (Figure 2) that represent those involved from the technological construction to the beneficiary of the services provided from the resultant SDR system.

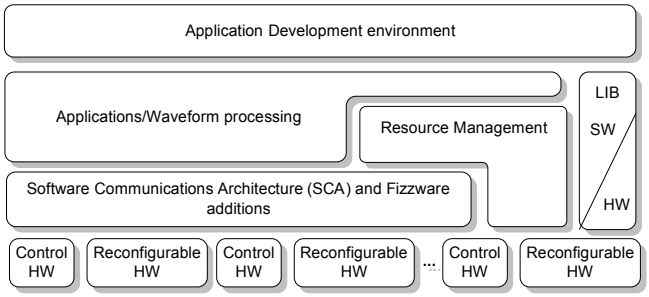

Figure 1: Software Defined Radio system block diagram

The *Application Development Environment* in Figure 1 is the part that a *System* or *Network Integrator* (Figure 2) operate so the use of the reconfigurable platforms are used with appropriate SDR development tools. These are the support tools that combine:

1. Primitive functions conceived by the Information Theorist. One example can be the current advances in LDPC codes. As these decoders develop, they could be integrated directly into the SDR system without a major system redesign.

Proceeding of the SDR 06 Technical Conference and Product Exposition. Copyright © 2006 SDR Forum. All Rights Reserved

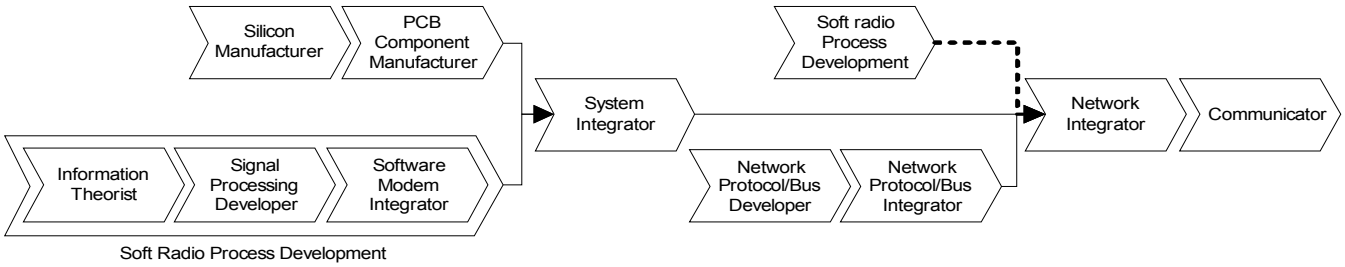

Figure 2: Actors within the value chain of a Software Defined Radio

2. *Soft Radio Process Development* primitives are then implemented into functions by the *Signal Processing Developer*. For example the specific implementation of the the LDPC decoder.
3. The functions are then combined into appropriate format for the Software Modem Integrator who can use the API formed by SCA and/or Fizzware additions as appropriate to the end system.

The following briefly describes the actions of the actors in Figure 2 with reference to the overall architecture shown in Figure 1.

The application chain is formed by considering the available Library Components (LIB) that are XML files containing descriptive files for SCA. LIB is separated into the Software Modules (LIB_SW) that can be processing blocks, storage for the function chain logical connections, ports and interface information, API and drivers that form an application chain. Hardware Modules (LIB_HW) XML files are descriptive with characteristics of the hardware devices. Additional files included within the LIB_HW are the bitstream and object code files. Bitstream and object code files can be used directly to configure an FPGA, DSP or other processing component.

The *Applications/Waveform processing* is the processing unit that compiles the information defined by the user placing the blocks in chain to form the application with the *Resource Manager* recording and controlling the chain manipulation and at the same time verifying the interfaces in the chain. The processed Waveform is supplied into to *SCA* and the *Resource Manager* decides the run time reconfiguration flow of information from the library through SCA.

The *Resource Manager* has increased importance in a real time reconfigurable SDR system because of overall supervision of information exchange between the *Applications/ Waveform processing, Library* and *SCA*. The application chain is formed and processed by the *Applications/ Waveform processing* layer. The *Resource Manager* manages information flow into a platform through *SCA*. Allocation of information is taking into account:

1. Reported information from SCA includes the available reconfigurable space, busy platforms, configured blocks location and status,
2. Functional block size is information from the XML tags stored in the *LIB* block
3. Rules associated with the space that this function block will take if installed into one silicon manufacturer product from another. *Silicon manufacturers* provide components like FPGA, FPAA, DSP, microprocessors and other raw hardware components.
4. Information from the *Applications/ Waveform processing* (Figure 1) include name ID, amount and order of blocks in the chain, platforms combination available for the chosen chain are useful for the resource manager.

Resource Manager block can have an Electronic Design Automation (EDA) usage by directly effecting the target hardware rather than passing the application descriptive information through the tools provided for verification. Example of tools that offer verification if application description information is written in any Hardware Descriptive Language (HDL) is the XILINX Integrated Simulation Environment (see the length of verification process until the bitstream is generated in Developer System documentation [1]). The verification process is time consuming and as the application becomes more complicated then verification becomes exponentially more expensive with more advanced tools required. The Resource Manager shall manage optimised methods of processing resource allocation which is a complex topic and is beyond the scope of this paper. For an FPGA example the resource manager could facilitate direct and fast processing of the FPGA bitstream and complete the reconfiguration using direct bitstream manipulation tools such as JBITs [2].

*Software Communications Architecture (SCA)*When the SDR platforms are tuned and the system is configured then

Proceeding of the SDR 06 Technical Conference and Product Exposition. Copyright © 2006 SDR Forum. All Rights Reserved

the Network Integrator needs a method to test the components provided by the Software Radio Process Development[3][4] is the point that all the application chain descriptive files and implementation of Application Programming Interface (API) are used to interface with the SDR target platforms converge before the application chain is implemented in the underlying configurable hardware. SCA provides:

1. Processing of the chain from CoreFramework objects
2. Control of gathering distributed information using Common Object Request Broker Architecture (CORBA),
3. Code portability between different programming languages using Interface Description Language (IDL) as a mapping before transfer of code from outside the SCA architecture and another IDL before the code transfered into SCA [5].
4. Operating environment (OE) of SCA is Portable Operating System Interface for uniX (POSIX) real time embedded Operating Environment functions to provide multi-threaded support for applications. POSIX profile is based upon the Real-time Controller System Profile (PSE52) as defined in IEEE 1003.13 standards [3][6].

The *Fizzware* additions to the SCA architecture are explained in Chapter 4.

Control Hardware symbolizes an intelligent control of the processing functions and definition of processing chains of a SDR platform. Every SDR platform is partitioned into *Control and Reconfigurable Hardware.* Controlling hardware can be a PROM memory that holds the configuration bitstream for the FPGA to be configured. Example of *Control Hardware* is a stand alone SDR platform needs some mechanism to initiate the arbitration of control triggering on the platform (CPLD interface component). *Control Hardware* is responsible for supervisory function such as facilitating practicality of configuration, monitoring status (watchdog functionality) and any appropriate localised management.

Reconfigurable Hardware is the PCB Component and a special case of silicon devices is the General Processor Processors (GPP) not always used for real time processing but are used for controlling purposes. Such devices include micro-controllers, microprocessors, etc. product like reconfigurable PCBs with silicon devices on board. Silicon devices that are reconfigurable are Field Programmable Gate Array (FPGA), Field Programmable Analogue Array (FPAA), Digital Signal Processor (DSP) and hybrids of reconfigurable silicon devices like Field Programmable Mixed-signal Array (FPMA) [7]. A special case of silicon devices is the General Processor Processors (GPP) not always used for real time processing but are used for controlling purposes. Such devices include micro-controllers, microprocessors, etc.

An SDR platform (in Figure 1) is a platform for different operators *System and Network Integrators* (Figure 2). *Network Integrators* combine more services and components on a reconfigurable platform. Combining reconfigurable systems to advance the services provided then the network protocols, information buses need to follow the standards in the field (provided by the *Network Protocol/Bus Developers*) but also to integrate the standards and finely tune the SDR platforms and protocol standards (this is the responsibility of the *Network Protocol/Bus Integrator*). When the SDR platforms are tuned and the system is configured then the Network Integrator needs a method to test the components provided by the Software Radio Process Development. All the actors Network Integrator, Information Theorist and Network Protocol/bus integrator create a product cycle. This product cycle can continually facilitate the dynamic addition of new processing to SDR systems which can be actively in service with active Communicators.

3. IN-SYSTEM BASEBAND RECONFIGURATION

Baseband Processing is refer to as a part of every radio system that comprises of processes that must perform in Real-Time. Best description for *Baseband Processing* is the Open Systems Interconnection Reference Model layers (OSI). First layer is the from the analogue plane to the digital plane (ADC) and second is the *Data Link Layer* that is frequently separated in two layer *Medium Access Control* (MAC) and *Logical Link Control* (LLC) Layer [8].

Baseband processing reconfiguration is a challenging area in SDR implementation because of the limitations of the reconfigurable hardware that can support real time reconfiguration. SDR platforms that are described by Joint Tactical Radio System group are Commercial Off-The-Shelf (COTS) that can largely only utilize APIs for Coarse-grain reconfiguration. Coarse-grain reconfiguration is the process in an SDR system with SCA as reconfigurable supervisory device over the physical hardware. But in the baseband processing when very fast reconfiguration is required then a finer grain of reconfiguration may be necessary. Coarse-grain function reconfiguration requires a long time frame (longer than fine-grain functions reconfiguration) to perform a reconfiguration. Because baseband part of the SDR system is operating in Real-Time the system needs to find a long time frame to perform the reconfiguration process so that data is not missed in the baseband level because the data

Proceeding of the SDR 06 Technical Conference and Product Exposition. Copyright © 2006 SDR Forum. All Rights Reserved

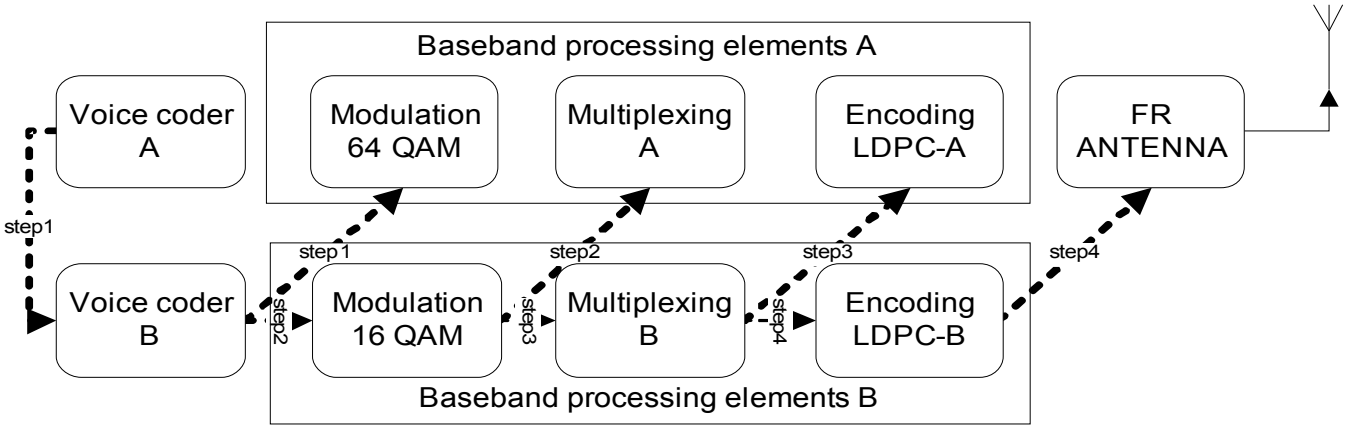

Figure 3: Example of seamless reconfiguration of an SDR Tx platform

recovery process in these layers (physical and MAC) is to issue a re-transmission. The solution for that is to add fine-grain reconfiguration functionality to SCA. With fine-grain reconfiguration the example in Figure 3 can be implemented to change the baseband element from "baseband A" to "B" with time constraints which disallow any disconnection or loss of data within the upper layers of the communication protocol.

Physical Layer (Phy) defines all the electrical and physical characteristics of devices that can process the information bits before (transmitter side) or after (receiver side) the RF antenna. This layer is the conversion layer from the analogue plane to the digital plane (ADC) and visa versa (DAC). Example of services that the Physical Layer can provide are the filtering stages, communication control of the communication medium, character coding/decoding and if the medium supports contention resolution and flow control. All services provided by the *Physical Layer* are characterized as Real-Time therefore always performed in every Tx/Rx through the antenna. The Physical layer medium can not be reconfigurable in the sense of having one medium and change (reconfigure) without loss of information.

Flexibility on the Baseband as described (Figure 3) discusses fine-grain processing required to be used in the following example. A Binary Phase Shift Keying modulation in a harsh (fast or slow fading or Additive White Gaussian Noise) channel might not be effective modulation technique compared with a 16-Quarter Amplitude Modulation (16-QAM) scheme through the same harsh channel. Therefore the *System Integrator* should be aware of the channel that the information is exchanged and if not aware of the channel an intelligent channel recognition system needs to evaluate the channel (this is the concept of Cognitive Radio) and after the evaluation of the channel the necessary changes to the baseband part of the SDR system can be applied.

One of the combinations that an SDR platform is fine-grain reconfigured from a Baseband element A to B with an application is presented in Figure 3. Many assumptions need clarification based on fine-grain reconfiguration:

1. Any reconfiguration needs an appropriate time-window that the block targeted for reconfiguration is idle. Example is the MAC layer in the transmitting side (Tx-MAC) operating and the receiving MAC (Rx-MAC) layer is idle so the MAC (Tx and Rx) layer can not be reconfigured. This kind of decision making needs to be enforced in the Resource Management (figure 1), SCA and Fizzware or both functional blocks for a more effective (avoiding loss of information data) reconfigurable system.
2. The reconfiguration in figure 3 is not the only series of reconfiguration events. Reconfiguration is decided according to the demands of the communication channel.
3. Fine grain reconfiguration is possible only with the integration of Fizzware with SCA architecture.

4. FIZZWARE

Fizzware includes a guide to create a more hardware orientated library with added feature that can take advantage of a more detailed library to provide enhancements over current SCA definitions. Essentially the files provided and used by SCA (XML files in Domain Profile) are modified with information that can be provided by Silicon and PCB Component Manufacturers. These additions are concentrated in areas which partition the distributed SDR system into configurable partitions. Each partition has XML and classes associated with it that provide more information and control such as

Proceeding of the SDR 06 Technical Conference and Product Exposition. Copyright © 2006 SDR Forum. All Rights Reserved

1. Silicon component properties down to pin by pin electrical and logical signal properties,
2. interface standard, functionality, location on the component,
3. size of reconfigurable space,
4. a map of specific resources on the reconfigurable space and how to address them
5. level of reconfigurability of partition

A whole database structure, XML detail and appropriate API's abstract the partially reconfigurable real time processing platform. System Integrators and/or Network Integrators can focus on the performance and work with the SDR platform based waveform implementation and control. Fizzware modifications to the SCA library allow access to previously inaccessible parts of the SDR platform unlocking functionality or revolutionary implementation with the platform

Fizzware's improved Library can offer improvement in the SDR system instead of been deadlocked with hidden and inaccessible, but valuable, functionality that is provided by the Silicon Manufacturers and PCB Components Manufacturers. If the overall SDR platform has limitations with real time reconfiguration then the System and/or Network Integrator/s needs to repeat the full cycle of negotiating with the Manufacturer (Silicon or PCB components) for a near alternative or brand new platform but in any case go through the waveform/SCA/XML files adaptation to reproduce the application.

Fizzware takes advantage of the extended XML files with the underlying SDR platform hardware partition (according to [9]) to control the placement of the chain blocks in the SDR system of platforms by providing at the same time freedom from the shear complexity of the current and developing technologies.

5. CONCLUSION

Reconfigurable Baseband processing in an SDR platform gives the possibility to dynamically choose the baseband architecture which could be cognitively decided. Current SCA architecture can limit the capability to reconfigure real time baseband processing. This paper presents the benefits of the addition of the Fizzware architecture which give fine grained access to reconfigurable processing resources in order to facilitate effective run time reconfiguration of real time processing. The benefits of this study are indicated in Table 1 with reference to the actors within the value chain of a software defined radio system.

6. REFERENCES

[1] Xilinx “Development System Reference Guide” part of the Xilinx Software documentation for ISE 8.2i, posted 2006

[2] S. A. Guccione and D. Levi, "Xilinx Bitstream Interface: A Java-based interface to FPGA hardware", in Configurable Computing Technology and its uses in High Performance Computing, DSP and Systems Engineering, Proc. SPIE Photonics East, J. Schewel, ed., SPIE - THE International Society for Optical Engineering, (Bellingham, WA), November 1998

[3] Burns P, “Software Defined Radio for 3G” Chapter 7 “Software Architecture and components” ISBN 1-58053-347-7, edited by Artech House, 2002

[4] Raytheon Company, DoD JTRS, “SCA Training For developers and testers” 5 day training course, presentation 2002.

[5] By DoD JTRS “Software Communications Architecture Specification”, document version “SCA V3.0”, posted August 2004.

[6] IEEE Std 1003.1-2001, published on 2001 and ISO/IEC 9945-1,2,3,4:2002, published on 2002.

[7] P. Chow, P. Chow and P.G. Gulak, “A field-Programmable Mixed-Analog-Digital Array”, Proceedings of the 1995 ACM third international symposium on Field-programmable gate arrays, Pages: 104 – 109.

[8] D. Lund and B. Honary. Baseband Processing for SDR. Chapter Contribution to ‘Software Defined Radio: Enabling Technologies’, Wiley, 2002, ISBN: 0-470-84318-7

[9] Lund D, Martin I, Honary B, “Fizzware a generic control architecture for the Management of present and future processing hardware”, SDRF Edinburgh meeting.

Proceeding of the SDR 06 Technical Conference and Product Exposition. Copyright © 2006 SDR Forum. All Rights Reserved

	Current SCA	*SCA + Fizzware*
Silicon Manufacturer	SCA provides a template for API to interface different reconfigurable silicon devices	Silicon manufacturers can provide more detailed information about reconfigurable features of their device. An 'electronic datasheet' can be provided with which reconfigurable features become easier accessible.
PCB Component Manufacturer	Coarse-grained abstraction of silicon manufacturer products	Abstraction is available which can either abstract or describe the physical and logical construction in detail
Signal Processing Developer	Soft components can be defined and deployed to SCA enabled hardware systems	Soft components can be faster deployed and can take advantage of hardware features which support real time reconfigurability. A choice of basic or detailed abstraction is available
Software Modem Integrator	Full processing chains can be defined and deployed over SCA compliant systems	A wider scope for in-system upgrading of the individual components of the soft modem. Cognition can be easier controlled rather than being largely defined within the real time processing domain..
System Integrator	Coarse-grain configuration of the SDR system that limit the base-band configuration	Fine-grain reconfiguration of an SDR board so the base-band configuration is available
Network Protocol/Bus Developer and Network Protocol/Bus Integrator	Tuning in SDR systems to produce more efficient reconfiguration buses and protocols can reduce some important factors to a base-band reconfiguration.	Fizzware provides this fine-grain configuration and any reconfigurable part can be changed using the finest detail to service an adaptive system so even network protocols between processing hardware can be manipulated to improve performance
Network Integrator	CORBA offer interfaces to reconfigurable components and different Operating Environments	Run time design automation and reconfigurable resource allocations processes can take advantage of the distributed nature of SCA.
Information Theorist	Information theorist is a long way from the end system	The Information Theorist can use the fine-grain reconfiguration to make algorithmic optimisations for reconfiguration. Cognitive Radio can benefit from the direct input of the information theorist and the signal processing developer.
Communicator	The communicator benefits from a flexible communication system	The communicator benefits from a more flexible communication system which has potential for more advance reconfigurable functionality. Cost benefits to the user arise from an even more software oriented system due to the unlocked capability of current and future theoretical techniques.

Table 1: Benefits for the value chain actors role in SCA+Fizzware

Proceeding of the SDR 06 Technical Conference and Product Exposition. Copyright © 2006 SDR Forum. All Rights Reserved