*flex*cell™ – DEPLOYMENT AND SOFTWARE VIEWS

Francois Luneau (Radical Horizon, Montréal, Québec, Canada;
fluneau@radicalhorizon.com)

ABSTRACT

This paper discusses the *flex***cell**™ system requirements. It provides the hardware architecture (deployment view) and the software architecture (software view) then summarizes *flex***cell**'s key features.

1. INTRODUCTION

The first *flex***cell**™ platform design was created in 1996 with production commencing in 1998. This platform included two RF front ends, one capable of digitizing the entire cellular band and the other, the entire PCS band. It also contained up to 6 processing cards. Each processing card had two ADCs, 8 DDCs, 4 FPGAs and 4 DSPs. The system contained a card with a FPGA that received the digitized signal from all ADCs, multiplexed them together, and sent the resulting signal to all processing boards. In early 2001, a decision was made to upgrade the platform. Following a market survey of processing cards, a decision was made to develop our own software radio platform because the interface with the RF analog section was found inadequate. However the hardware architecture of the 1998 platform was very flexible. The software architecture had to improve code reusability and provide cross-platform compatibility. Our analysis had shown that most of the software written for the 1998 platform could not be reused on the off-the-shelf platform.

2. *flex*cell™ OBJECTIVES

The *flex***cell**™ platform upgrade objectives included functional objectives, design constraints and quality factors.

Functional. The functional objectives have been categorized under wireless connectivity, wireless processing, wireline connectivity, and communication applications.

Wireless connectivity. The system shall be capable of receiving and transmitting multiple service bands such as: HF, VHF marine, cellular, ISM, PCS, simultaneously. Each service band transceiver shall be capable of digitising the whole band. The total number of channels (Tx/Rx) available on the platform must be customisable to meet different customer needs. The system also must be capable of dynamically changing the number of channels allocated to each service band to meet changing operational environments.

Wireless processing. Commercial Off-the-Shelf (COTS) processors (FPGA, DSP and GPP) shall be used for wireless processing. To reduce project risk, the frequency band channelizer shall be implemented in a COTS programmable ASICs called Digital Down Converters (DDC) and Digital Up Converters (DUC). The selected DDC and DUC shall be capable of digitally extracting a UMTS channel. Each wireless signal processing board shall provide at least 64 Tx and 64 Rx frequency band channelizer functions and the processing power required to process these channels.

Wireline connectivity. The initial wireline connectivity requirements were that a COTS Computer Telephony Integration (CTI) board that uses the cPCI 6U form factor and H110 bus shall be used. Later in the project this requirement was changed to allow the CTI board to have either a H110 interface or a PICMG 2.16 Packet Switching Backplane interface.

Communication applications. The requirements are that the software architecture shall be flexible allowing the communication application to be hosted on either a processor on the RF card, a single board computer, a remote host link by Ethernet, or a combination of the above.

Design Constraints. To allow for future technology insertion, the hardware module designed for *flex***cell**™ shall be hot swappable and meet the open industry cPCI standard for computer telephony to ensure compatibility between the wireless subsystem and wireline subsystem. Exchange of traffic data between the wireless processing unit and wireline processing unit shall be done using the H.110 interface and PICMG 2.16. The software architecture shall be based on the US DoD JTRS

SCA open standard. Finally, the performance of the RF subsystem shall meet regulatory requirements.

Quality Factors. The *flex***cell**™ system shall be modular allowing for different system configurations in terms of:

- the total number of cards included in the system;
- the total number of RF front ends that can be added;
- the total number of Tx and Rx channels that can be made available;
- the type and throughput capacity of wireline connectivity.

The software architecture had to ease the integration of third party wireless protocol software on the platform. It also had to ease the portability of existing wireless protocol software on the evolution of future platforms. To improve serviceability, all cards designed for *flex***cell**™ must be hot swappable.

bus has been designed to support the application needs of the software radio industry.

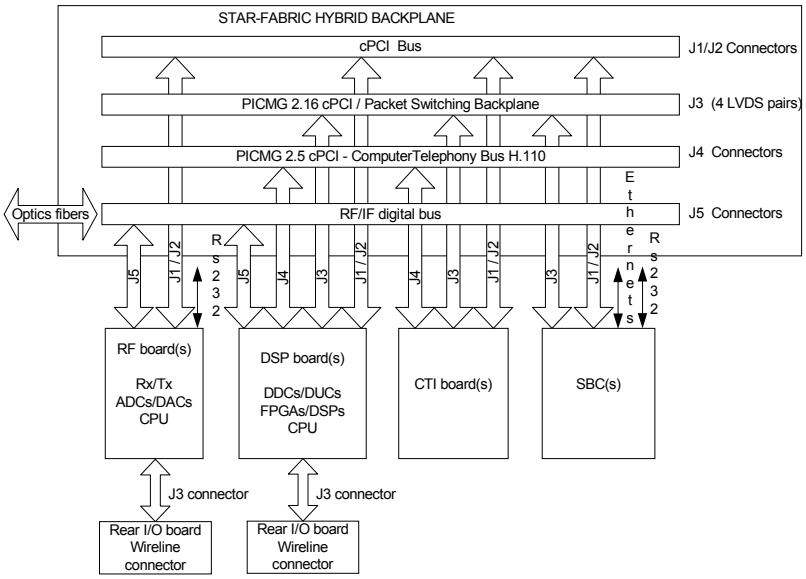

Figure 1. flexcell high capacity radio configuration.

3. HARDWARE ARCHITECTURE

3.1. Overview

The *flex***cell**™ platform can be easily scaled-down to meet requirements. It can consist of one RF board, one DSP mezzanine card, and, if wireline connectivity is required, a rear Input/Output (I/O) board connected to the RF board using pins from the J3 connector. This scaled-down example provides 16 Tx/Rx channels for one service band. This is a very cost effective solution but not as scalable as the *flex***cell**™ high capacity radio configuration.

Figure 1. is an illustration of the *flex***cell**™ high capacity radio. It consists of one or two cPCI chassis, multiple RF cards, DSP cards, CTI boards, one or two single board computer cards, and a RF/IF digital bus card.

The RF cards provide wireless access, incorporate ADCs and a DAC and transmit/receive the IF samples to/from the DSP board through the RF/IF digital bus card. The digital RF/IF bus card is a special back-end card that spreads across multiple consecutive slots. It attaches to the J5 connector of each of these slots to provide the digital RF/IF bus access to the DSP cards and RF cards. It consists of 16 pairs of LVDS for Rx and 16 pair of LVDS for Tx. It is capable of transmitting and receiving up to 30 Gbits of data. *flex***cell**™'s RF/IF digital

Access to wireline connectivity is provided by either rear I/O cards or Computer Telephony Integration (CTI) cards. The rear I/O card provides I/O interface standards such as 8 analog phone lines, Quad E1/T1, fast Ethernet, etc. The communication processor on the RF card and DSP card uses pins on the J3 connector that are not needed by the H110 bus standard to connect to the rear I/O card.

The CTI card is a front-end card that provides access to wireline connectivity for different wireline standards. It exchanges voice/data information with the DSP card through the H.110 telephony bus or the PCIMG 2.16 bus. This card is similar to the DSP card and the COTS software loaded on the card determines the wireline protocol supported.

All of the cards in the *flex***cell**™ platform have been designed to the 6U CT Front Boards specification to allow for interoperability with the computer telephony industry. All cards are hot swappable in accordance with the requirements of the cPCI hot swap specification. Details on the wireless access subsystem (RF boards) and wireless processing subsystem (DSP boards) architecture are discussed in the following paragraphs. Details on CTI board and SBC board architecture are not provided because these are COTS components and information can be provided by visiting the web pages of the manufacturers of these products.

3.2. Wireless Access Sub-System

The wireless access subsystem consists of at least one RF board connected to the following external modules: multiple antennas, and if necessary, power amplifiers and duplexers. The wireless access system has very limited programmability providing mainly embedded functionalities. The RF board shown in Figure 2. consists of two RF front end receivers (Rx) and one RF front end transmitter (Tx). Each receiver is connected to a Rx

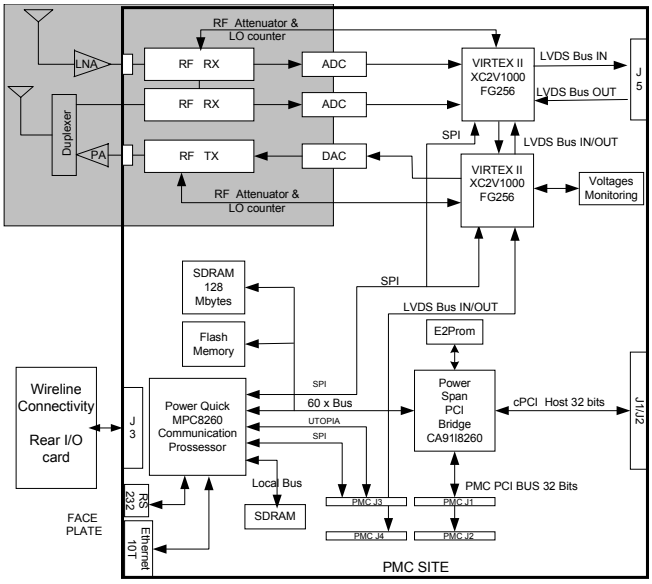

Figure 2. RF board

antenna or one element of an antenna array. The RF front end transmitter is connected to a Tx antenna. When the same antenna is shared for Tx and Rx, the RF board may be connected to a duplexer. The typical Tx power output for the whole service band is 0 dbm and, if this is not sufficient, it can be connected to an external power amplifier (PA). The Rx has a built in LNA but to increase sensitivity, the RF board may be connected to an external LNA and RF bandpass filter (BPF).

The RF front end receiver is a single conversion receiver. It uses a RF filter to select the desired service band (824 to 849 MHz for *flex***cell**^{TM}'s cellular band receiver). A mixer is then used to convert the signal to an intermediate frequency (IF) (*flex***cell**^{TM}'s cellular band receiver LO frequency is 777.6 MHz). Following the down conversion stage, the signal goes in IF filters to select the desired IF band, an attenuator and an amplifier. An automatic gain control (AGC) algorithm embedded in the FPGA controls the RF attenuator output to ensure the signal is going to be within the ADC's input parameter characteristics. The ADC digitises all channels for all waveforms being transmitted by external radios in the receiver service band. The sampling rate must be at least twice the signal bandwidth (the sampling rate for the *flex***cell**^{TM} cellular band receiver is 78.125 MHz). The samples are sent to a FPGA. The FPGA stamps the ADC identification (id) number on the samples. The samples with the ADC id number, are sent to *flex***cell**^{TM}'s RF/IF digital bus through the J5 connector.

For transmission, a FPGA receives, from the J5 connector, all digital samples stamped with the DAC id number. The FPGA removes the id number from the sample, and if required performs upsampling and filtering before it sends the signal to the DAC that converts the sample to analogue. The analogue signal then goes to an IF filter, an attenuator, a mixer, a RF filter, an amplifier and another RF filter. This signal is then sent to any of the following RF systems: a power amplifier, a duplexer or an antenna.

The RF board incorporates the following processors: two FPGAs (1 Million gates each), and one communication processor. The communication processor's operating system (OS) is real-time Linux. Each RF board is connected to the cPCI bus using J1 and to *flex***cell**^{TM}'s RF/IF digital bus using the J5 connector. A DSP card, compliant with the PMC standard, can be attached on the PMC connector. The board can also be connected to a rear I/O board using unused pins on the J3 connector to provide wireline connectivity such as Ethernet, E1/T1, JTAG, RS-232, or a combination of the above. The front plate of each board includes a RS-232 connector and an Ethernet connector. In addition, the RF board includes 3 antenna connectors, 2 for receiving and 1 for transmission.

3.3. Wireless Processing Sub-System

The wireless processing sub-system consists of either one DSP daughter card that attaches on a RF card, multiple DSP boards or a combination of the above. The DSP daughter card attaches on the RF card using the PMC connector. The card can process digital RF/IF data to/from its own RF card or from another RF card in the system. Figure 3. shows a DSP board incorporating four DSP modules. The daughter card consists of one DSP module and therefore has approximatively four times less processing power.

*flex***cell**^{TM}'s DSP board is shown in Figure 3. It connects to the backplane using J1/J2, J3, J4 and J5 connectors. It also provides connection to Ethernet 10T and RS 232 interfaces using a front plate connector. It has four DSP modules, one communication processor,

on board memory, internal busses, one bridge PCI H.110 and one PCI bridge. The communication processor and PCI bridge are the same as the ones found on the RF card.

The most important components on the DSP board are the four DSP modules. Each DSP module includes: four Intersil ISL 5416 Quad Digital Down Converters (DDCs) and four ISL 5217 Quad Digital Up Converters (DUCs), one Xilinx Virtex II XC2V4000 or XC2 XilinxV6000 in a FF1517 package and one TI's C6416 DSP. On the DSP, the operating system used is TI DSP BIOS. In summary, each DSP board has tremendous processing power having 16 Quad DDCs, 16 Quad DUCs, 4 FPGAs and 4 DSPs.

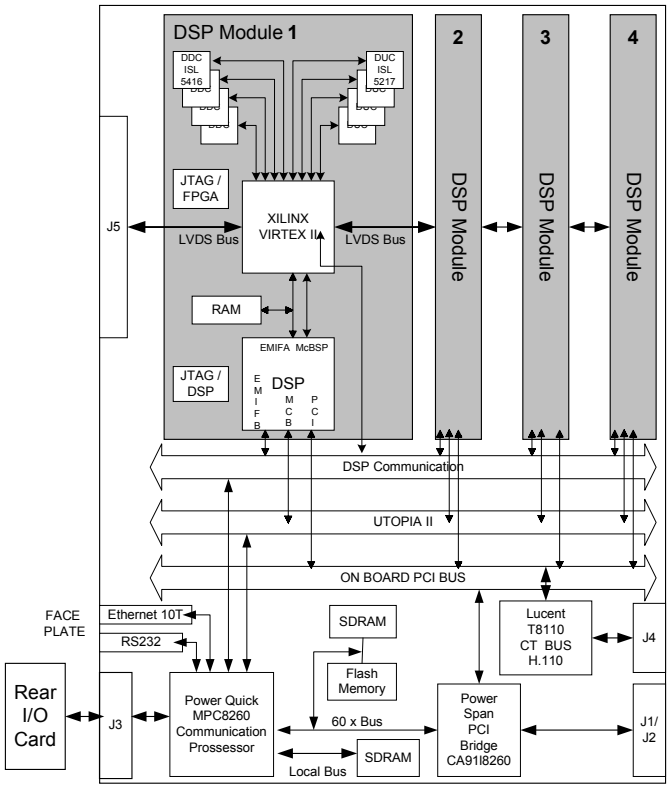

Figure 3. DSP board with DDCs, DUCs, FPGAs, DSPs and GPP

When a user opens a radio/waveform application using a computer connected to the *flex***cell**™ system, the devices (ADC, DAC, DDC, DUC, DSP, FPGA) are selected, logical connection between them established, and the system state changes to either standby or operation. When in the operation states the FPGA selects the samples containing the ADC's id number and forwards them to the DDC. The DDC extracts the samples that correspond to the carrier frequency and forwards them back to the FPGA. The FPGA demodulates the signal and forwards symbols to the DSP. The DSP performs de-interleaving, forward error correction, source decoding, layer 2 protocol, layer 3 protocol and if destination (sink) uses a different protocol, source coding to the format of the destination protocol and bridging function. As many channels can be received/transmitted simultaneously as there are DDCs/DUCs.

On each DSP card there are 64 DDC's and 64 DUC's. Therefore, each DSP card can be used to receive and transmit 64 carrier frequencies. The carrier frequencies can be dynamically configured, i.e. each DDC/DUC can be connected to any ADC/DAC. Because each connection is a logical connection, it can be changed anytime while the system is in operation.

4. SOFTWARE ARCHITECTURE

4.1. Overview

The software architecture is divided into three modules: wireless signal processing, wireline signal processing and communication applications. Figure 4. depicts this architecture.

The implementation of wireless signal processing is implemented in DDCs/DUCs, FPGAs, and DSP microprocessors. The FPGAs in the RF cards send broadband signals to the FPGAs on the DSP cards/mezzanine. Channel tuning is done in the programmable DDCs/DUCs on the DSP cards/mezzanine. The modulation/demodulation is done in the FPGAs on the DSP card/mezzanine. The DSP microprocessor on the DSP card/mezzanine performs channel coding/decoding, layer 2 and 3 functionalities, and source coding and decoding. It may also perform bridge or router functions as well as transcoder rate adapter unit (TRAU) functions. Also, the DSPs provide configuration and get status to/from FPGAs. The information transferred between DSPs and FPGAs is at the level of slots or frames, right before modulation (transmission) or after demodulation (reception).

When using rear I/O cards, the wireline signal processing is implemented in the communication processor (8260) on the RF card or the DSP card. When using a CTI card, the wireline signal processing is actually

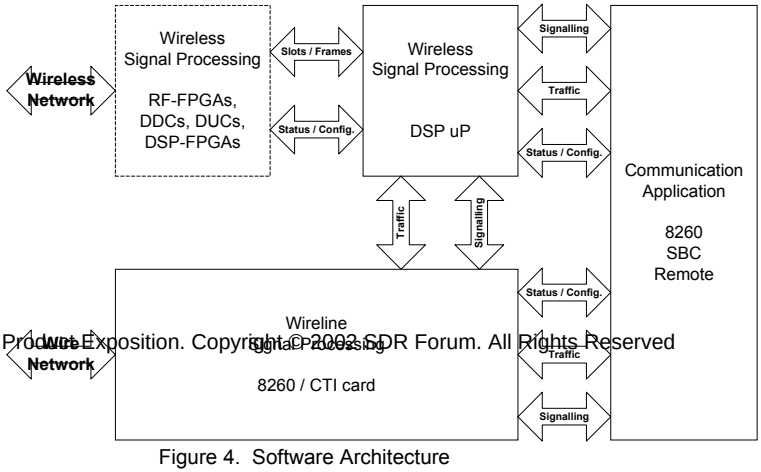

Figure 4. Software Architecture

done in that card using the software provided with the card. Data connectivity between the wireless signal processing and wireline signal processing is done through the H.110 bus or the PICMG 2.16 bus. When the latter is used, the communication processor (8260) on the DSP card executes the ethernet protocol.

Depending on strategic development partners, various communication applications at different scalability levels can be implemented in the *flex***cell**™ system. They may be implemented on SBC cards for medium/large-scale systems or directly in the communication processor (8260) of the RF cards and DSP cards for small-scale systems. For the development of these applications, an application development framework based on the JTRS-JPO SCA is provided that eases the development of scalable communication applications across multiple card cages and external systems by using a CORBA distributed system.

Details on the software architecture for the DSP microprocessor and communication application are provided in the following paragraphs. FPGA architecture is key to create a structure that enables flexible runtime configuration of waveforms. The description of the DAC-FPGA and the DSP-FPGA can be found in another paper.

4.2. ADC-FPGA

Figure 5. depicts the architecture of the ADC-FPGA. Only the main data paths are shown. The control path used by the Configuration Module is more like CHARIOT's[1] mesh architecture and not shown on the diagram. Every FPGA performs configuration of modules and clock management. The 8260 SPI I/F module receives configuration packets from the 8260. The configuration module extracts the different fields, latches data into configurable registers, and confirms the configuration. The Clk inputs module receives signals from an external timing reference clock, and the Clock Manager module controls and generates internal and external clocks. The Clock Manager also controls the internal reset and some external devices resets.

The main purpose of the ADC-FPGA is to perform the AGC functions and send the ADC data and AGC values to a DSP-FPGA. While in the initialisation state the Config module sets up parameters to indicate if diversity are used and the ADC id. The modules: ADC Interface, Rx AGC and Attenuator receives the ADC data, computes the gain and adjusts the attenuator power level. The modules: ADC Interface, Rx Framer, cPCI J5 I/F and PMC I/F, receive the ADC samples, multiplexes the ADC samples with the AGC value, adds a header containing the ADC identification number and the type of data (ADC or AGC) and sends the result to a DSP-FPGA.

4.3. Wireless Signal Processing in DSP

In the DSP microprocessor, the architecture of the wireless signal processing is divided into three main parts: wireless signal framework, protocol definitions and channel processes. It must support simultaneous processing of multiple bi-directional frequency channels using different protocols. Figure 6. provides an overview of the wireless signal processing architecture.

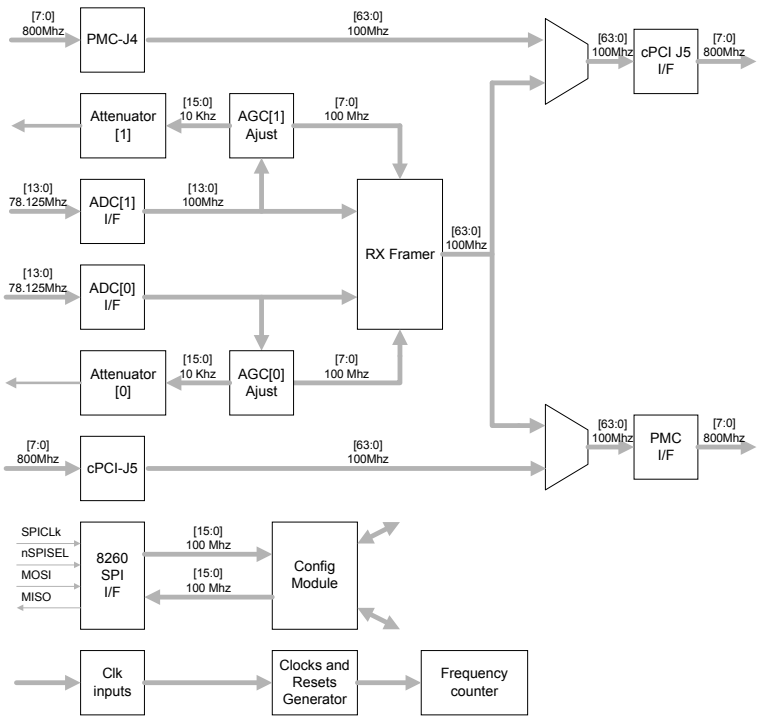

Figure 5. Architecture of the ADC-FPGA

The wireless signal framework is composed of the operating system and interfaces to all external systems. DSP/BIOS provides basic software interfaces to manage interrupts, processes, and internal peripherals, including the Viterbi decoding and turbo code coprocessors. The frequency channel interface allows channel processes to configure frequency channel tuners (DDC/DUC), modulators and demodulators as well as allowing base band data transfers in slot or frame format. The flash interface provides flash memory access to channel processes. The wireline interface provides

wireline signal processing access to channel processes. This interface is part of the junction between wireless/wireline worlds and allows transfer of traffic and signalling data. The application interface provides communication application access to channel processes. It allows the channel processes to get configuration from the application as well as transferring traffic and signalling data.

The channel processes are the actual workers in the DSP. Each process is surrounded by the wireless signal framework and uses a protocol definition, as specified, by the communication application. This definition is then the active definition of the channel. Two processes are used for each simultaneous channel to support full-duplex operations: one for reception and one for transmission (only one process per channel is required for half-duplex operations). The two processes of a

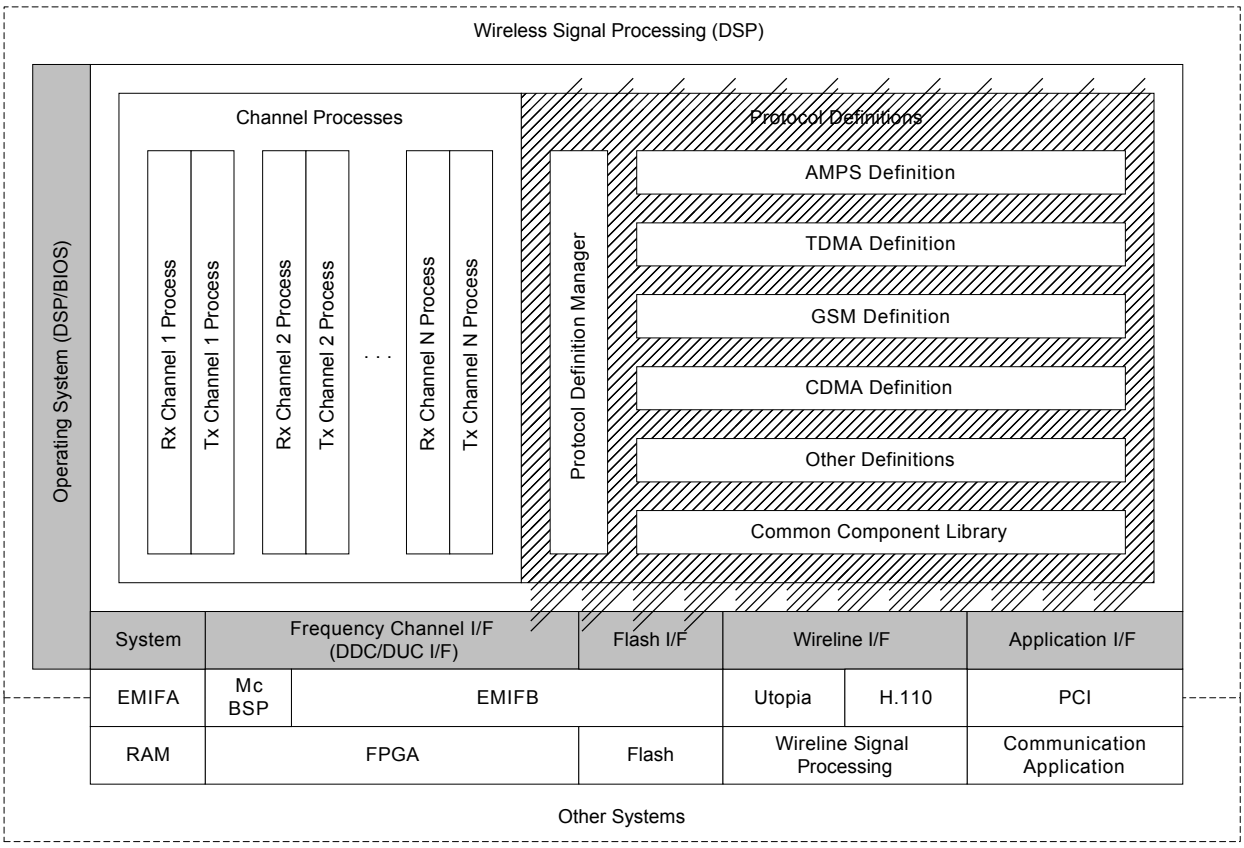

Figure 6. Wireless Signal Processing - Architecture Overview

To support different standards, the channel processes have access to protocol definitions. These definitions provide the procedure to follow according to different protocols, like AMPS and TDMA. As an example, a channel process may be assigned to receive AMPS data while another may be assigned to receive TDMA data, according to respective protocol definitions. Protocol definitions are re-entrant; many channel processes may use the same protocol definition at the same time. The design takes into consideration that protocol definitions need to be easily upgraded; either by updating current definitions or by adding new ones and that dynamic loading of protocol definitions is an essential feature that needs to be implemented in the near future.

channel talk to each other to keep reception and transmission in-sync, as required.

4.4. Communication Application

Radios, base transceiver stations (BTS), jammers, scanners, etc. are examples of communication applications. The communication application uses waveform software. The software architecture meets the JTRS's SCA architecture standard. It is composed of project-specific classes that use the waveform API services. It inherits attributes from the SCA Core Framework classes, and is surrounded by the services provided by the SCA Core Framework and the CORBA ORB. Communication applications are created using the Application Factory class and may use Waveforms'

Application Factory to create the waveforms needed. Figure 7. provides an overview of the communication application architecture.

The commercial multi-mode BTS under development is an example of a communication application. This application is created using the BTS Application Factory. It uses the AMPS Application Factory to create many instances of the AMPS waveform the UMTS application factory to create instances of the UMTS waveform.

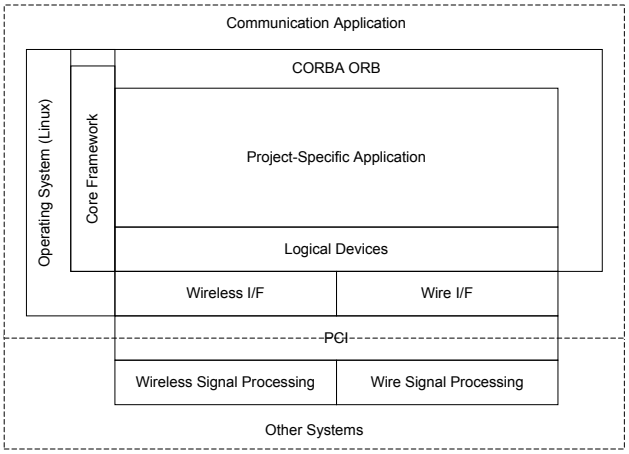

Figure 7. Communication Application

5. SUMMARY

*flex***cell**™ is a modular architecture that can be used to develop a family of radio applications. Many different system configurations are possible: 1 card to 21 cards; 1 RF board to 19 RF boards; and 16 to 1216 carrier frequencies (64 DDCs/DUCs x 19 DSP cards). The *flex***cell**™ RF/IF bus throughput (30 GHz) allows reception and transmission of 800 MHz discontinued RF bandwidth. The RF/IF bus also allows any DDCs/DUCs to be connected to any ADCs/DACs providing the ability to dynamically change the number of channels allocated to a band. The *flex***cell**™ RF/IF bus has been designed specifically to exchange RF/IF data between cards like the H.110 bus has been designed to exchange voice/data between CTI cards. The *flex***cell**™ RF/IF bus aims to become, to the Software Radio industry, what the H.110 bus is to the CTI industry.

Wireless signal processing is implemented in a multi-processor FPGA and DSP platform. Each FPGA and DSP can process up to 16 channels simultaneously. Each channel can be configured with a different waveform (protocol). Multiple waveforms can be supported (HF, VHF marine, UHF Satcom, WCDMA, GSM, TDMA, Inmarsat, WLAN, MMDS, LMDS, etc.) The software loaded on the platform defines the platform capabilities. Services can be updated by replacing current software modules with updated software modules. Services can also be added by loading new software modules. Because services are provided by software, it is possible to adjust services 'on the fly' to adapt to current operational needs. The FPGA architecture, not often discussed in literature, is very important to allow 'on the fly' reconfiguration.

*flex***cell**™ provides seamless operation between wireless and wireline networks. Wireline signal processing is performed by inserting COTS CTI boards or by the communication processor executing COTS wireline protocol software. More than one wireline network can operate on a *flex***cell**™ system.

The application software uses the layered software architecture defined by JTRS's SCA. This architecture makes the application software independent from the hardware. It also enhances the system's scalability. This design simplifies the insertion of future enhancements to the hardware and software modules. When JTRS's SCA architecture becomes the industry standards it will also simplify the insertion of technology from third-party vendors.

6. REFERENCES

[1] J.H. Reed, *Software Radio – A Modern Approach to Radio Engineering*, Prentice Hall PTR, Upper Saddle River, 2002