JTRS/SCA AND CUSTOM/SDR WAVEFORM COMPARISION

Daniel R. Oldham Ph.D.
Maximilian C. Scardelletti
NASA Glenn Research Center
Brookpark, OH 44135

ABSTRACT

This paper compares two waveform implementations generating the same RF signal using the same SDR development system. Both waveforms implement a satellite modem using QPSK modulation at 1M BPS data rates with one half rate convolutional encoding. Both waveforms are partitioned the same across the general purpose processor (GPP) and the field programmable gate array (FPGA). Both waveforms implement the same equivalent set of radio functions on the GPP and FPGA. The GPP implements the majority of the radio functions and the FPGA implements the final digital RF modulator stage. One waveform is implemented directly on the SDR development system and the second waveform is implemented using the JTRS/SCA model. This paper contrasts the amount of resources to implement both waveforms and demonstrates the importance of waveform partitioning across the SDR development system.

INTRODUCTION

NASA requires space radios to meet communication needs for future missions. NASA normally contracts with radio vendors to build and supply space radios based on mission requirements. While this method works for NASA, there is also a high cost with each radio development and the radio's architecture intellectual property is retained with the radio vendor. NASA needs to accumulate space radio technology to allow for reuse, lower cost and provide for reliable operations in the development of future space based radio systems.

NASA is considering Software Define Radios (SDR) as a possible solution for future communication systems. There are a number of SDR development systems currently available for waveform development and testing. The SDR development systems provide many features to increase portability of waveforms by using standard component interfaces, common operating systems (OS) and common board support packages (BSP). Waveforms can be developed directly on these SDR development systems and some support the Joint Tactical Radio System, Software Communication Architecture (JTRS/SCA) [1]. The Joint Tactical Radio System (JTRS) provides the Software Communication Architecture (SCA) as a software specification on an SDR development system.

This paper compares two waveform implementations generating the same RF signal using the same SDR development system. One waveform is implemented directly on the SDR development system and the second waveform is implemented using the JTRS/SCA model. The waveforms are partitioned the same across the GPPs and FPGAs devices. The majority of the waveform implementation is on the GPPs with only the final digital RF modulator stage implemented on the FPGA. Both waveforms implement an equivalent set of radio functions on the GPPs and use an equivalent FPGA bit file. This paper contrasts the amount of resources to implement both waveforms and demonstrates the importance of waveform partitioning across the SDR development system.

.

SDR DEVELOPMENT SYSTEM

The SDR development system for this waveform implementation comparison is from Spectrum Signal Processing and is the SDR-3000. [2] The SDR-3000 Transceiver is broken into two parts: the base band processing and the channelizer subsystem. The base band processing subsystem (PRO3500 module) has two PPC7410 and one PPC405 processors running at 250 MHZ and 200 MHZ respectively. The channelizer is broken into two parts: the input module (TM3150 module) and the IF processing engine (PRO3100 module). The input module has four ADCs and DACs running at 80 MHZ. The IF processing engine has one PPC405 processor running at 200 MHZ and four Xilinx Virtex II FPGAs running at 100 MHZ. The two PPC7410 processors have 128M bytes of RAM and the two PPC405 processors have 64M bytes of RAM. The SDR-3000 uses flexFabric interfaces for high speed data transfers between the hardware nodes. There are external digital interfaces on the PRO3100 module and Ethernet and RS232 ports on both boards. For the SDR-3000 the theoretical full duplex communication rate over each link is defined at 320M BPS. Spectrum Signal Processing provides a communication library (quicComm) as a standard interface to the hardware devices.

The SDR3000 architecture provides many possible waveform implementation opportunities. The flexFabric interfaces allow data routing between any of the hardware devices. Spectrum Signal Processing provides a board support package (BSP) and the processors use VxWorks for the real time operating system (RTOS). The two PPC7410s and four Xilinx FPGAs provide the normal digital RF signal processing paths for the SDR3000 development system. The PPC405 processors provide external Ethernet connections and control paths for the hardware devices. The SDR3000 development system includes a PC host running Windows 2000 OS with the Tornado cross compiler for VxWorks and Xilinx ISE software for programming the FPGA devices.

WAVEFORM PARTITIONING

The JTRS/SCA model requires partitioning the majority of the waveform onto the GPP to allow for software implementations of waveforms. For this paper the QPSK waveform data source, framing and encoding stages are all implemented on the GPP (PPC7410) and only the final modulator stage is implemented on the Xilinx FPGA (Virtex II).

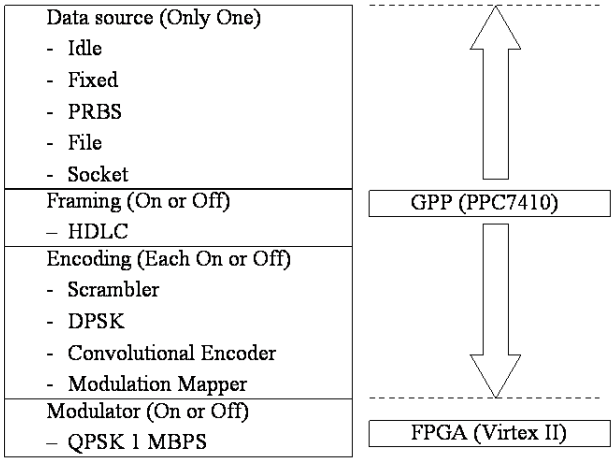

Figure 1 QPSK Waveform Partitioning

The QPSK waveform is broken into four parts: the data source, framing, encoding and the modulator stages. The data source stage switches input data streams between idle, fixed, bit error rate sequences (PRBS), file or socket based data. The framing stage is off or on using HDLC framing. The encoding stage includes turning on or off the scrambler, Differential Phase Shift Keying (DPSK), ½ rate Convolutional encoding and the modulator mapper functions. The modulator mapper function provides vectors to the modulator stage. The modulator stage is QPSK modulation using 2M BPS symbol rate or a 1M BPS data rate. The output of the modulator stage are digital RF vectors to feed the DAC to create a 70 MHZ IF. The 70 MHZ IF is then available to several fixed satellite modems to verify correct waveform operations.

OPERATING ENVIROMENT

The minimum operating environment (OE) for the SDR3000 development system is VxWorks RTOS and Spectrum Signal Processing's quicComm library. The Custom/SDR waveform is implemented directly on the minimum OE. The JTRS/SCA waveform OE requires the addition of the Harris Core Framework and SCA CORBA software. The Harris Core Framework and SCA software both sit on top of the RTOS and BSP for each processor. The footprint sizes for the files for Harris Core Framework and SCA software are shown in the tables below. The output file names are shown for only the PRO3100 PPC405 processor and the configuration files are the same for the other processors. All tables show file sizes in bytes.

Table 1 Harris Core Framework Footprint Sizes

Harris Core Framework	PRO3100 PPC405
dmtkCFBase.out	6,567,326
dmtkCosEventBase.out	935,688
dmtkCosNamingBase.out	510,390
dmtkDeviceManager.out	2,912,030
dmtkDevice.out	3,407,170
dmtkDomainProfile.out	3,211,660
dmtkEventServices.out	614,945
dmtkFileServices.out	901,411
dmtkLogServiceBase.out	1,832,400
dmtkUtility.out	1,401,437
Total bytes	22,294,457

Table 2 SCA Software Footprint Sizes

SCA software	PRO3100 PPC405
sca_devicemanager.out	2,912,030
sca_fabric_access_manager.out	149,976
sca_fabric_channel.out	1,540,400
sca_fpga_dev.out	1,975,751
sca_load_sdks.out	5,925
sca_partnum2target.out	5,655
sca_ppc_dev.out	1,805,549
sca_sdr3000_dev.out	744,663
sca_sync_manager_dev.out	2,341,779
Total bytes	11,481,728

The footprint size for the OE for each of the processors is defined in the tables below. The sizes of the files vary slightly and depend on the configuration options selected for each processor.

Table 3 OE Footprint Sizes PRO3100 PPC405

Spectrum Signal SDR3000	PRO3100 PPC405
Dynamic RAM bytes	67,107,840
VxWorks OS	1,840,105
QuicComm Library	1,447,241
Sub Total RTOS	3,287,346
Percent RTOS to RAM	5%
SCA Software	11,481,728
Harris Core Framework	22,294,457
Sub Total SCA	33,776,185
Percent SCA OE to RAM	50%
Total RTOS+SCA	37,063,531

Table 4 OE Footprint Sizes PRO3500 PPC405

Spectrum Signal SDR3000	PRO3500 PPC405
Dynamic RAM bytes	67,107,840
VxWorks OS	1,877,699
QuicComm Library	1,292,759
Sub Total RTOS	3,170,458
Percent RTOS to RAM	5%
SCA Software	10,738,249
Harris Core Framework	22,294,457
Sub Total SCA	33,032,706
Percent SCA OE to RAM	49%
Total RTOS+SCA	36,203,164

Table 5 OE Footprint Sizes PRO3500 PPC7410-A

Spectrum Signal SDR3000	PRO3500 PPC7410-A
Dynamic RAM bytes	134,215,680
VxWorks OS	1,747,205
QuicComm Library	1,384,515
Sub Total RTOS	3,131,720
Percent RTOS to RAM	2%
SCA Software	7,363,541
Harris Core Framework	22,703,137
Sub Total SCA	30,066,678
Percent SCA OE to RAM	22%
Total RTOS+SCA	33,198,398

Table 6 OE Footprint Sizes PRO3500 PPC7410-B

Spectrum Signal SDR3000	PRO3500 PCC7410-B
Dynamic RAM bytes	134,215,680
VxWorks OS	1,747,205
QuicComm Library	1,384,515
Sub Total RTOS	3,131,720
Percent RTOS to RAM	2%
SCA Software	7,363,541
Harris Core Framework	22,703,137
Sub Total SCA	30,066,678
Percent SCA OE to RAM	22%
Total RTOS+SCA	33,198,398

The total OE footprint resources for all processors are shown in the table below.

Table 7 Total OE Footprint Sizes for SDR3000

Spectrum Signal SDR3000	Total Bytes
Dynamic RAM	402,647,040
VxWorks OS	7,212,214
QuicComm Library	5,509,030
Sub Total RTOS	12,721,244
Percent RTOS to RAM	3%
SCA Software	36,947,059
Harris Core Framework	89,995,188
Sub Total SCA	126,942,247
Percent SCA OE to RAM	32%
Total RTOS+SCA	139,663,491

WAVEFORM RESOURCES

The JTRS/SCA waveform is broken into seven executable images defined in the table below.

Table 8 JTRS/SCA QPSK Waveform Footprint Sizes

QPSK Waveform	JTRS/SCA
strsAppController.out	1,675,772
strsCommon.out	794,404
strsDataGenerator.out	701,965
strsFilterUpConv.out	1,080,384
strsEncoder.out	772,947
strsFramer.out	701,233
strsModMapper.out	929,569
Total bytes	6,656,274

A number of XML files are used to configure the software parameters for the JTRS/SCA waveform. The XML files are defined in the table below.

Table 9 JTRS/SCA XML Waveform Footprint Sizes

QPSK Waveform	JTRS/SCA
strs405QC.spd.xml	477
strs405TM13300.spd.xml	491
strs7410QC.spd.xml	483
strsAppController.prf.xml	11,902
strsAppController.scd.xml	1,178
strsAppController.spd.xml	1,672
strsCommon_405.spd.xml	484
strsCommon_604.spd.xml	484
strsDataGenerator.scd.xml	1,072
strsDataGenerator.spd.xml	1,445
strsEncoder.scd.xml	1,185
strsEncoder.spd.xml	1,423
strsFilterUpConv.scd.xml	1,211
strsFilterUpConv.spd.xml	1,415
strsFramer.scd.xml	1,185
strsFramer.spd.xml	1,418
strsModMapper.scd.xml	1,185
strsModMapper.spd.xml	1,433

strsTelemetryWFT.sad.xml	13,113
strsWFTXFPGA.spd.xml	2,014
Total bytes	45,270

The Custom/SDR waveform uses one executable image and is shown in the comparison table below. The waveforms use the same FPGA bit file with one small change to the JTRS/SCA version. The difference is to handle a change in the data packet header when the JTRS/SCA OE is loaded onto the SDR development system. In this case the digital logic and the interfaces are the same for both FPGA bit files.

Table 10 Comparison of Waveform Footprint Sizes

Waveform resources bytes	JTRS/SCA	Custom/SDR
Total Xilinx FPGA bit file	2,733,252	2,733,252
Total XML configuration	45,270	0
Total GPP executable size	6,656,274	509,380

OPERATIONS

Both QPSK waveforms are loaded onto the SDR development system; started; stopped and unloaded. The JTRS/SCA model requires loading the Harris Core Framework and SCA software before the SCA waveform is loaded. The table below compares the JTRS/SCA to Custom/SDR implementations.

Table 11 Comparison of SCA to SDR Operations

Operation	JTRS/SCA	Custom/SDR
Start Name Service	Uses unique name IDs for each object in the system and XML files for configuration.	None, uses static data tables for RF equipment, fixed IP addresses and hardware resources.
Start Device Managers	Start SCA device manager on each processor.	Start SDR device manager on each processor.
Install Waveform	Open Harris Domain Manager Monitor software and use Application Factory to install waveform.	Waveform is built into SDR software and included with SDR device manager.
Start/Stop Waveform	Use menu from Application Factory to start and stop the waveform.	Use external command interface to start and stop the waveform.
Unload Waveform	Use Application Factory to unload waveform.	None, waveforms are part of SDR device manager software.

MAXIMUM PERFORMANCE

The performance of the PPC7410 processor at 250 MHZ is measured to determine the maximum possible data rate. The maximum data rate depends on the size of the transfer buffer between the PPC7410 and the FPGA. The smallest transfer size is 8 bytes and is constrained by the flexFabric interface. The largest transfer size is 16K bytes and is constrained by the DMA on the SDR3000. The metric is to measure the control loop data rate on the processor for different size buffers with no output data to the flexFabric interface. The results are shown in the table below.

Table 12 Maximum PPC7410 Data Rates BPS

Buffer Size in Bytes	Average Data Rate in BPS
8	24,031,568
16	15,782,160
32	21,733,088
64	27,407,744
128	31,108,096
256	33,878,528
512	35,005,440
1,024	36,057,088
2,048	36,147,200
4,096	36,638,720
8,192	36,298,752
16,384	36,470,784

On average, the performance of the PPC7410 at 250 MHZ for all buffer sizes is within the real time requirements for the 1M BPS QPSK waveform data rate. The maximum data rate for the FPGA is determined by the maximum clock rate for the device. The FPGA is clocked at 100 MHZ and therefore has about three times the maximum performance of the PPC7410 processors.

CONCLUSION

Two equivalent waveform implementations have been characterized using the Custom/SDR and the JTRS/SCA models. The implementations use the same SDR development system with the same hardware partitioning, and generated the same RF signal. Both waveform configurations use the same RTOS and BSP. The average RTOS and BSP footprint is 3% of available memory on the SDR development system. The JTRS/SCA model uses CORBA and Harris Core Framework on top of the RTOS and BSP. The average JTRS/SCA operating environment footprint is 32% of available memory. The JTRS/SCA OE footprint is about ten times larger than the Custom/SDR RTOS development footprint. The waveform application

footprint for the custom/SDR model is 495K bytes and the JTRS/SCA model is 6,500K bytes. JTRS/SCA waveforms must conform to the specifications so the majority of the waveform partitioning is onto the GPP. The waveform application for the JTRS/SCA model is about 12 times larger than the optimized custom/SDR version. As a result, the waveform performance limit is reached at 1M BPS on the GPP for the SDR development system and achieving higher levels of performance requires partitioning the waveform onto the FPGA devices.

REFERENCES

[1] Military & Aerospace Electronics by John McHale, December 2004.

[2] Spectrum Signal Processing Product Specifications, SDR-3000 Series Software Defined Radio Transceiver Platform, 2006.

BIOGRAPHY

Dr. Daniel R. Oldham is a computer engineer at NASA Glenn Research Center working in the Satellite Networks and Architectures Branch. Dr. Oldham is also an adjunct professor for Digital Logic Design at Case Western Reserve University. Dr. Oldham received his M.S. and Ph.D. degrees in Computer Engineering and Science from Case Western Reserve University, Cleveland, Ohio, in 1993 and 2001, respectively. Dr. Oldham received his B.S. degree in Computer Science and Engineering from the University of Toledo, Ohio, in 1984. From 1984 to 1998 he was a research and design engineer for Rockwell Automation and developed a number of real time control products (PLC-5, Control Logix). From 1973 to 1979 he served as an electronic technician, in the U.S. Navy Fleet Ballistic Missile Submarine service.