MULTI-MODE RECONFIGURABLE SOFTWARE DEFINED RADIO ARCHITECTURE FOR AVIONIC RADIOS

Abdessamad Amrhar, Alireza Avakh Kisomi, Eric Zhang,
Joe Zambrano, Claude Thibeault, René Jr. Landry

Laboratory of Space Technologies, Embedded Systems, Navigation and Avionic (LASSENA)
Department of Electrical Engineering, École de Technologie Supérieure, (ETS), Montreal, CANADA

Abstract

This paper presents an architecture using Software Defined Radio (SDR) techniques to integrate multiple avionic radios in a single hardware module. This architecture allows avionic radio functions to share the same computational and hardware resources, run simultaneously numerous avionic radios, and dynamically reconfigure the resource allocation depending on the current active radio sets (or modes). This work aims to address two main challenges of avionics design: reducing the Size, Weight, Power and Cost (SWAP-C), and future proofing of the design to cope with the continuous changes in standards and communication needs. The presented prototype integrates five radio applications: four CNS/ATM commercially existing applications, namely Distance Measuring Equipment (DME), Automatic Dependent Surveillance – Broadcast (ADS-B) OUT, ADS-B IN and Transponder Mode-S (TMS), and a custom-made In-Flight Connectivity (IFC) radio service equipment referred as Wide-Band Radio (WBR). Additionally, the prototype is based on Low-Cost Commercial Off-The-Shelf (COTS) components. The experimental results show that, with current hardware, attaining the Minimum Operational Performance Standards (MOPS) is achievable as long as the hardware impairments are compensated via DSP techniques.

Introduction

Currently, in avionic systems, there is an ongoing migration from Federated architecture to Integrated Modular Avionics (IMA) architecture, due to the demand of reducing Size, Weight, Power and Cost (SWaP-C), which is a main concern in aeronautical equipment design.

The Federated design is based on the concept that avionic applications are handled by at least one Line Replaceable Unit (LRU). This self-contained hardware module (with its own processing unit and power supply) eases the certification procedure and maintenance [1]. However, the unnecessary redundancy that comes with these modules puts a burden on the aircraft with the addition of each new system. Also, Federated modules struggle to adapt to the continuing evolution of Communication, Navigation, and Surveillance (CNS) and Air Traffic Management (ATM) standards.

In contrast, the IMA architecture is based on resource sharing and reallocation. As illustrated in Figure 1, applications are sharing processing and network resources. This approach results in a significant reduction in SWaP-C [2], which is the main goal of the project.

The IMA architecture can be characterized by the following features [1]:

- Having a multi-layered scalable software architecture (for abstraction), which enables code reusability and portability;
- Allowing static reconfiguration of resource allocation (while aircraft are grounded) or even dynamic reconfiguration (while aircraft are airborne);
- Having an Operating System (OS) with a time deterministic (real-time) scheduling and robust protection against applications concurrency for resources;
- Physically integrating hardware while partitioning resources for best performance;
- Favoring COTS hardware and software;
- Favoring the usage of design patterns that eases incremental certification.

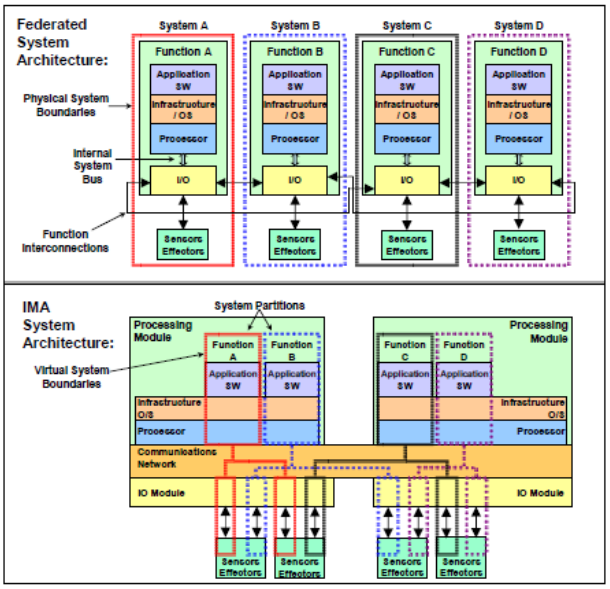

Figure 1. Federated vs. IMA Architectures [3]

A similar approach can be adopted for aeronautical communication equipment to harness the benefits of IMA by using Software Defined Radio (SDR) techniques, which means offloading hardware from waveform generation and/or decoding. As a result, it enables integration of multiple avionic radio modules in a single hardware unit. The proposed Multi-Mode Software Defined Avionic Radios (MM-SDAR) architecture allows the sharing of computational resources, radio frequency (RF) front-end and antennas (Figure 2). In addition, it is capable of reallocating resources depending on the modes, as each mode is a set of active avionic radios.

The SDR technology is already adopted as a solution in other fields such as SatCom (Satellite Communication) [4]. For avionics, this technology is considered as a candidate for future digital communication as attested by a number of standardization efforts [5]. In the same sense, LASSENA Laboratory leads an important project, called AVIO-505, whose main objective is to establish new design methods and digital signal processing techniques for more robust and efficient equipment for communication, surveillance and navigation in the fields of aeronautics and aerospace.

The aim of this paper is to present a multi-mode reconfigurable SDR architecture for the application of avionic radios, using low-cost COTS hardware and open source software, developed within the AVIO-505 project.

This paper focuses on the low power laboratory tests of the MM-SDAR highlighting the achieved solutions for multiple hardware and software challenges. Firstly, the detailed explanation about the proposed architecture and its specification is presented. Then, different modules of the system are introduced and the required considerations are discussed. After that, the implementation challenges, which have been overcome, are presented. And the tests and results are illustrated to show the performance of the architecture. Finally, the conclusion is presented, also the future work and suggestions regarding this project are discussed.

MM-SDAR Architecture

As shown in Figure 2, the conceptual MM-SDAR architecture is composed of five modules: 1) The antenna switch matrix that selects antennas depending on running applications as well as handling duplexing the RX (reception) and TX (transmission) paths. 2) The RF front-end that amplifies and filters the RF signals. 3) The transceiver that converts the RF signals to digital signals and vice versa. 4) The Field Programmable Gate Array (FPGA) that filters the digital signals and converts the sampling rate of the signal from high to low and vice versa. 5) The General Purpose Processor (GPP) running the avionic applications, interfacing with cockpit and other avionics while controlling all the MM-SDAR chains.

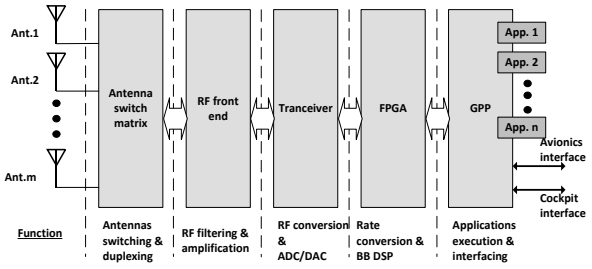

Figure 2. Conceptual MM-SDAR Architecture

To demonstrate the feasibility of the proposed architecture, a scaled down prototype is made using a COTS SDR platform (PicoSDR2x2E from Nutaq®). This platform comes with two radio cards (allowing to handle two avionic applications simultaneously), an FPGA and a GPP, in a single package.

Five avionic applications, four CNS/ATM commercially existing and one custom-made applications, were selected for the demonstration of this prototype: Distance Measuring Equipment (DME), Automatic Dependent Surveillance – Broadcast (ADS-B) OUT, ADS-B IN, Transponder Mode-S (TMS), and an In-Flight Connectivity (IFC) radio service equipment referred as Wide-Band Radio (WBR). The existing applications were designed according to their respective Radio Technical Commission for Aeronautics (RTCA) Minimum Operation Performance Standard (MOPS). WBR was selected to test a more complex communication since it is using Adaptive Coding and Modulation (ACM) algorithm to minimize the Bit Error Rate (BER). Each of the selected applications has the following characteristics:

DME [6]

- Measures slant distance between aircraft and a DME beacon
- Frequency band: 962 to 1231 MHz
- Channel spacing: 100 KHz
- Modulation scheme: Pulse Coded Modulation (PCM)
- Channel access: Broadcast on a fixed channel
- MOPS: RTCA DO-149

ADS-B [7]

- Broadcasts aircraft information (OUT) and gets air traffics information (IN)
- Frequency: 1090 MHz
- Modulation Scheme: Pulse Position Modulation (PPM)
- Channel access: Broadcast
- MOPS: RTCA DO-260

TMS [8]

- A transponder for ATC surveillance
- Frequency: 1030 MHz (RX), 1090 MHz(TX)
- Modulation scheme: PPM and DBPSK
- Channel access: Broadcast
- MOPS: RTCA DO-181

WBR

- Provides IFC radio service
- Frequency: Not fixed yet (tested with 2437/2417 MHz)
- Data rate: 0.5 to 2 Mbps
- Modulation scheme: DBPSK, DQPSK, D8PSK or D16QAM
- Coding: Reed Solomon 32 or 64 bits of parity
- MOPS: not applicable

The proposed architecture is capable of running any two of the listed applications simultaneously, and dynamically reconfigures resource allocation by swapping any of them.

Hardware and Software Environment

SDR platform

The PicoSDR is composed of three elements including an embedded computer with a 2nd generation Quad-core Intel Core i7 GPP CPU (SAM-C), an FPGA (Xilinx Virtex6 SX-315T) and a transceiver (Radio420X) with two TX-RX paths. The block diagram of the PicoSDR is shown in Figure 3.

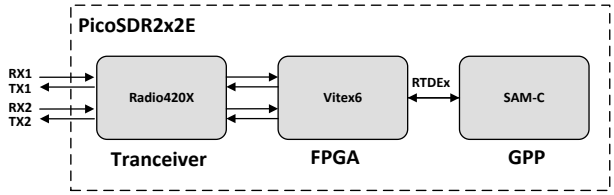

Figure 3. PicoSDR2x2E Block Diagram

Transceiver

Reception and transmission of Radio420X transceiver are based on direct-conversion architecture. It means that the frequency conversion is done directly from RF to baseband frequency (and vice versa) using a single Local Oscillator (LO) mixer stage, as shown in Figure 4. This part focuses on the reception since achieving its proper functionality is more challenging than the transmission.

Similar to other direct-conversion receivers, the Radio420X receiver shares the same benefits and flaws. This architecture is characterized by its high integrability, high out-of-band rejection, low cost and low power consumption. But, as it is shown in Figure 5, it comes with several impairments namely DC offset, IQ mismatch and flicker noise [9]. The DC offset is mainly because of LO leakage into the ADC. IQ mismatch is due to the mixing imperfections in I and Q branches, which creates an

image duplicate of the received signal. Flicker noise is a low frequency noise that mostly is a result of trapping and releasing of electrons in semiconductors [10]. Although these impairments degrade the performance of the receiver, they could be compensated with DSP techniques (as shown in the implementation section).

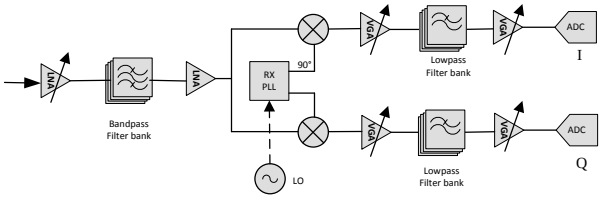

Figure 4. Radio420X Receiver's Architecture

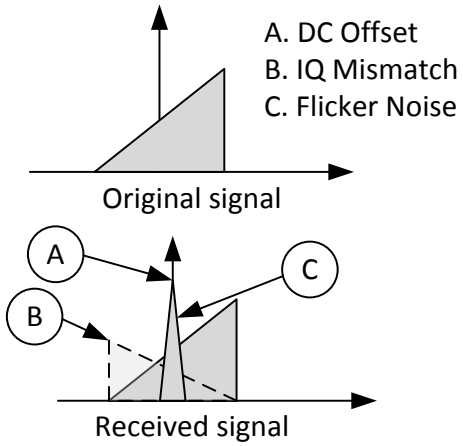

Figure 5. Impairments of Direct-Conversion.

GNU Radio

GNU Radio [11] is an open-source software development toolkit for designing real-time signal processing applications. It comes with a multi-threaded real-time scheduler, a C++/Python infrastructure and a plethora of libraries and code elements (or blocks) for common DSP functions.

The scheduler in GNU Radio uses a data-driven flow-graph, which starts by creating a thread for each block and a buffer for the output of that block. Then it shares these buffers with any connected block. At runtime, each block produces a number of samples, writes them into the output buffer and notifies the connected blocks. The blocks that are connected consume a number of samples presented at their inputs while producing the same amount of samples, write them in their output buffer then notify the next block. This procedure is shown in Figure 6.

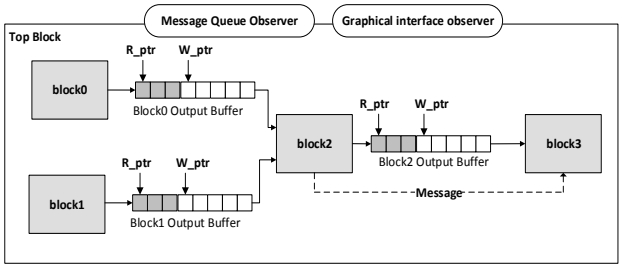

Figure 6. GNU Radio Scheduler

The advantage of processing samples in bulk is that it enforces a time-deterministic behavior for the designed system and it minimizes the overhead, due to switching between threads. However, because of data buffering, it introduces latency. Also, careless implementation might result in excessive memory consumption.

Latency and Jitters

Since data buffering ensures a deterministic behavior in the presence of jitters (caused by OS interruption and Ethernet connections), like any other GPP-based SDR system, latency will be an issue. In addition, GPPs achieve better performance when processing chunks of data rather than sample by sample.

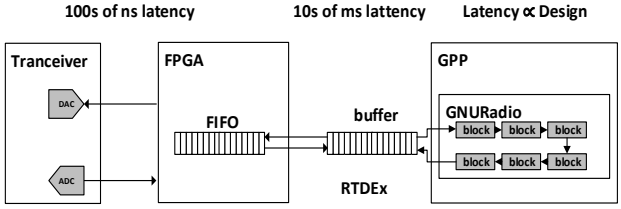

Figure 7. System Latencies

This means that applications that require a very fast response cannot be implemented on a GPP, which raises two concerns. First, time sensitive applications (such as DME) require a way to synchronize GPP and FPGA. Second, fast response applications (such as TMS, which requires a reply in less than 3μs) could not be functional if implemented on GPP.

In order to synchronize the FPGA and the GPP, a time aware mechanism was implemented (shown in Figure 8). Whenever DME transmits an interrogation, it tags that interrogation with a number. This tag is propagated to the FPGA and as

soon as it is received, it will be coupled with the current timestamps and sent back to the GPP. The GPP receives the tag and propagates it to the DME RX block to notify it when its transmission has actually occurred.

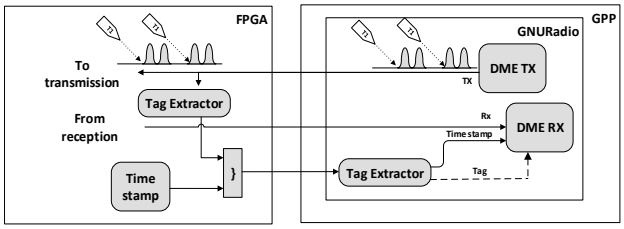

Figure 8. Time Aware Mechanism of DME

Multi-Rate Sampling

The variety of signal modulation scheme and bandwidths of the radio applications imposes a multi-rate architecture. Additionally, lowering the sampling rate reduces the computational burden on GPP. Also, the ADC/DAC sampling rate should be kept at maximum (in this case 20 MHz) because the quantization noise floor, N_c, is inversely proportional to sampling rate [12].

$$N_c = -1.8 - 6.02N - 10\log\left(\frac{f_s}{2}\right) dBc/Hz \quad \textbf{(1)}$$

where N is ADC resolution and f_s sampling rate. N_c decreases by 3 dB at each f_s doubling. Hence filtering an oversampled signal increases the Signal to Noise Ratio (SNR).

$$G_{SNR} = 10\log\left(\frac{B_F}{Fs}\right) dB \quad \textbf{(2)}$$

Special consideration should be taken into account for DME since the error is inversely proportional to the sampling rate.

$$\mathrm{d_{error}} = \pm \frac{1}{4 \times \mathrm{Fs} \times \mathrm{c}} \quad \textbf{(3)}$$

where d_{error} is the distance error and c is the speed of light.

Table 1 lists the operational sampling rates of each system as well as the conversion ratio. The sampling rates of the applications were selected as low as possible to minimize the computational load on the GPP side.

Table 1. Sampling Rate of the Application

Application	TMS	DME	ADS-B	WBR
Sampling Rate(MHz)	20	1	2	2
Conversion Ratio	1	20	10	10

Implementation

FPGA

Figure 9 and Figure 10 show the illustrations of the implemented MM-SDAR FPGA architecture, for the reception and transmission sides, respectively. As shown in these figures, the changes in sampling rates are done through a combination of a Cascaded Integrator Comb (CIC) filter and a compensation Finite Impulse Response (FIR) filter. This hardware implementation is an efficient method for filtering high sampling rate data. The RX and TX switches route the signals between the Radio420X and the active applications.

For each path of the reception model, as shown in Figure 9, an impairments compensation module corrects the DC offset, the frequency offset, and the IQ mismatch in this order.

The power measurement module measures the maximum full band power and sends it to the Analogic Automatic Gain Control (A-AGC) implemented in the GPP. The Digital Automatic Gain Control (D-AGC) auto-scales the baseband low rate signal to achieve the best possible dynamic range.

The Time Stamp block echoes back the tag recovered by the transmissions Time Domain Demultiplexer, as shown in Figure 10.

The avionic datalink module constructs the avionic data receiver from the GPP and stores it into TMS memory.

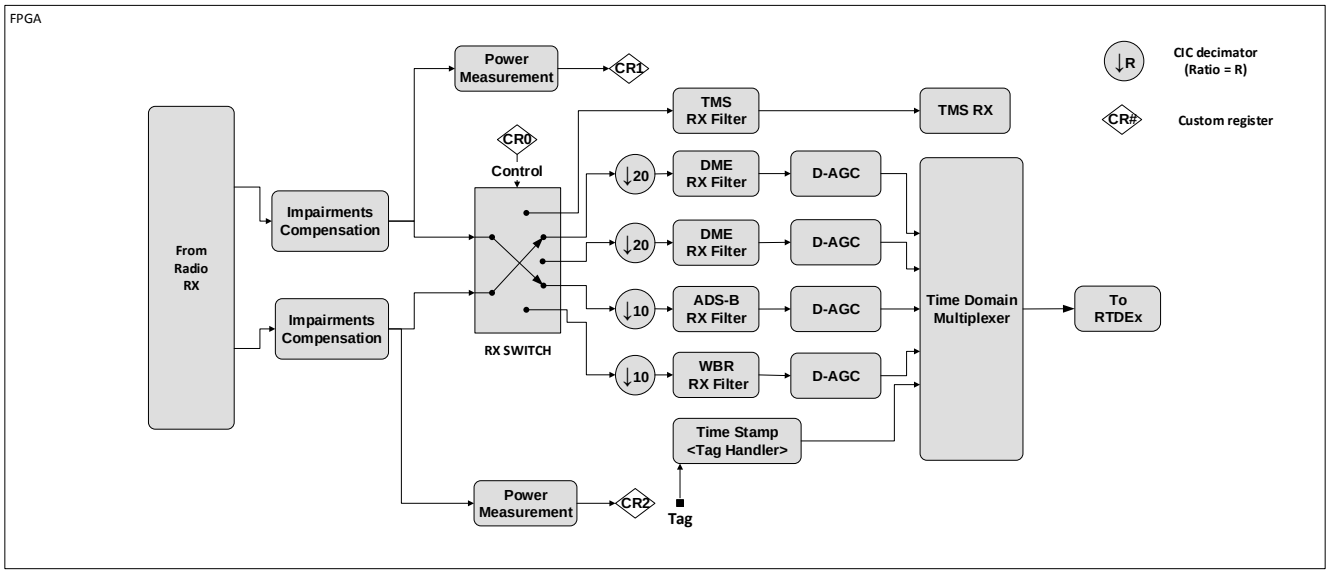

Figure 9. MM-SDAR FPGA Reception Architecture

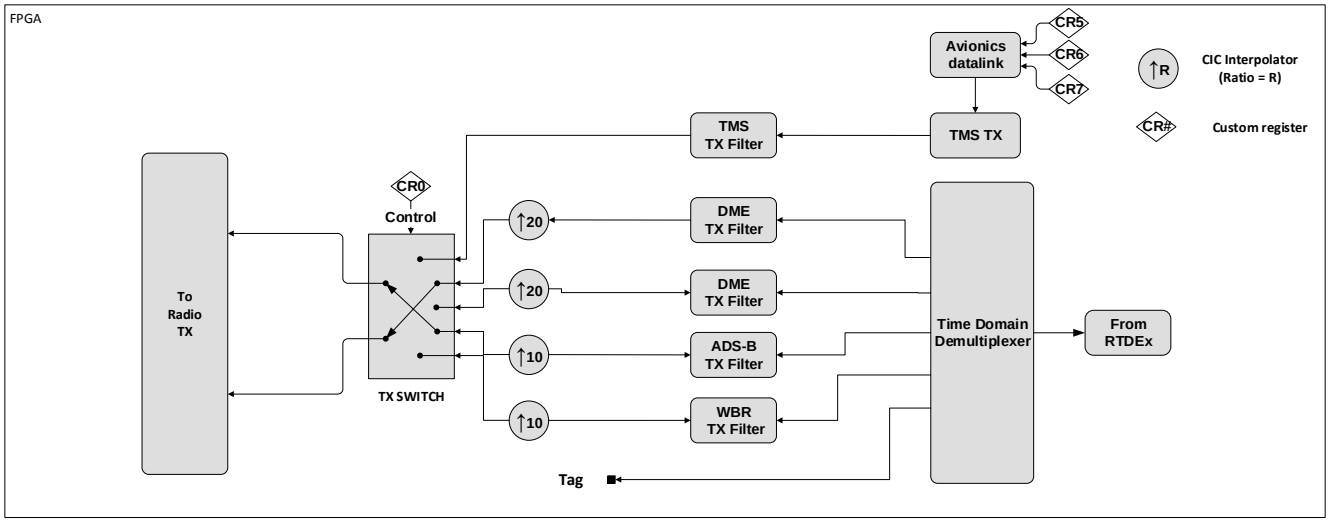

Figure 10. MM-SDAR FPGA Transmission Architecture

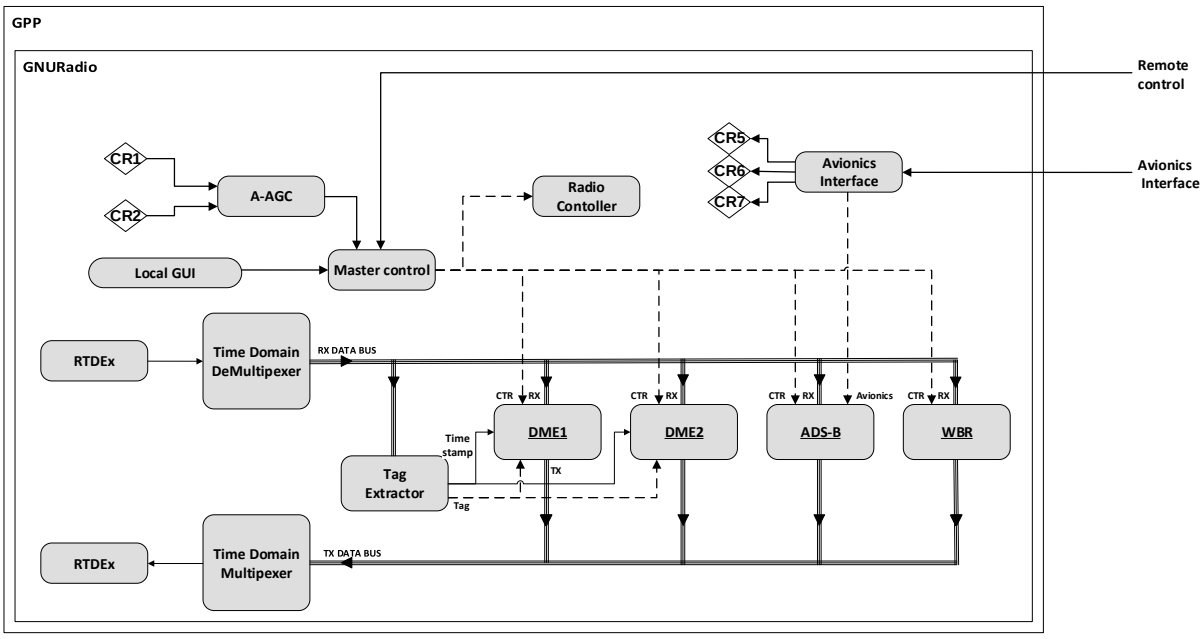

Figure 11. MM-SDAR GPP GNU Radio Model

GPP

Figure 11 illustrates the GNU Radio GPP model for MM-SDAR. The underlined blocks (DME1, DME2, ADS-B and WBR) are the implementations of the avionic applications. The received data from RTDEx[1], i.e. from FPGA, are demultiplexed then dispatched to the input of the corresponding application. The Time Domain Multiplexer collects the TX data of each active application and pads the rest of data with zeroes then sends it to RTDEx.

The whole system is controlled through the master controller block. It interprets the external commands issued by either the local GUI or a remote controller. The Radio Controller block controls the LO frequency and the amplifier gains. The avionic interface block parses avionic data from external sources (e.g. ARINC843) and sends them to TMS and ADS-B modules.

Tests and Results

Test setup

Figure 12 shows the equipment used to test the multiple modules of the MM-SDAR prototype. The PicoSDR2x2E (1) is running the proposed MM-SDAR architecture. Most of the radio application specifications (e.g. waveform shape, coding, timing, etc.) were validated using the certified test equipment IFR6000 (2), while the bandwidth and the spectrum mask were validated via a spectrum analyzer (not shown in the figure). A laptop running XPlane (6) simulates the necessary avionic instruments data and sends them as UDP packets to the prototype in order to generate ADS-B and TMS signals. WBR communication was tested using another WBR instance implemented in a laptop (4) and using PicoSDR (3) as a pass-through. The tablet (5) is used to simulate the pilot's cockpit as well as controlling and monitoring the prototype.

Sensitivity

One of the most important parameter in a receiver performance is its sensitivity. This

[1] RTDex is a real time data exchange protocol developed by Nutaq (see https://www.nutaq.com/blog/overview-rtdex-library)

parameter indicates the minimum required RF power at the receiver for an avionic radio system to perform properly. Table 2 lists the sensitivity of each MM-SDAR application in dBm and compares them with their respective MOPS requirements and their commercially available counterparts. The ADS-B and DME of MM-SDAR achieved the required sensitivity few decibels apart from commercial systems. The sensitivity of the current version of TMS is lower than the standards because the response time requirement for TMS significantly limits the order of the digital filter.

Accuracy (for DME)

According to DME MOPS, the accuracy of measured distance error should not exceed 1.7 NM or 0.25 % of the distance (whichever is maximum). Figure 13 illustrates the MM-SDAR DME error in Nautical Miles (NM). Since, in this curve, the maximum error does not exceed 0.21 NM, it shows that the error is within the accuracy range of the MOPS.

Throughput (for WBR)

Figure 14 shows the evolution of the prototype's implementation of WBR's throughput relative to input RF power. In an interference free environment, the measured dynamic range is 36 dB and the sensitivity is -61 dBm

Resources Usage

The target FPGA is a Virtex6 XC6VSX315T. Table 3 shows the amount of FPGA resources for Flip-Flops (FF), Look-Up Tables (LUT), Block Random Access Memory (BRAM) units and Digital signal processing (DSP) units used in this design. The label 'Used' signified the total resource usage; 'Nutaq' shows how much Nutaq firmware is using (Nutaq firmware handles communication between the radio card, the FPGA and the GPP); 'MM-SDAR' shows the proposed architecture usage; and 'TMS' shows the TMS application usage. The total amount of the used resources does not exceed 25 % of the FPGA and it is mostly used by Nutaq's firmware. Even though this implementation is still open to optimization, this could be ported to a smaller FPGA.

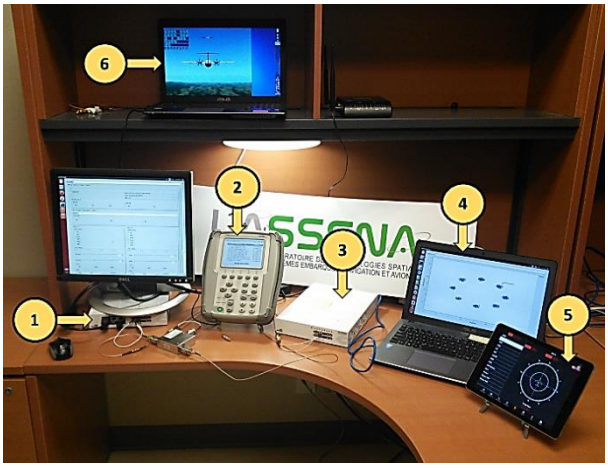

Figure 12. MM-SDAR Test Setup

Table 2. MM-SDAR Applications Sensitivity

	MM-SDAR	MOPS	Commercial
TMS	-62 dBm	-73 dBm	-74 dBm
DME	-87 dBm	-83 dBm	-90 dBm
ADS-B	-73 dBm	-72 dBm	-74 dBm

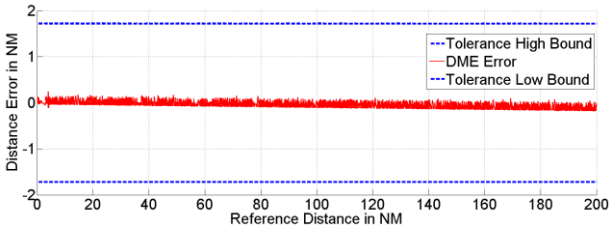

Figure 13. DME Error

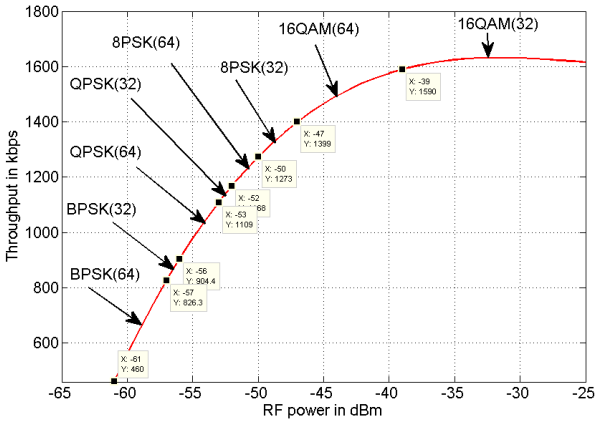

Figure 14. WBR Throughput vs. Input Power

Table 3. FPGA Resource Usage

	FF	LUT	BRAM	DSP
MM-SDAR	3.29%	6.23%	0.09%	11.01%
TMS	0.64%	1.18%	0.24%	2.75%
Nutaq	6.80%	16.30%	8.05%	2.38%
Used	10.09%	22.53%	8.14%	13.39%
Available	393600	196800	2112	1344

Conclusion

The aim of this work was to show that the integration of multiple avionic radios could be achieved using SDR techniques. This approach granted flexibility and adaptability regarding the hardware. In addition, multiple hardware limitations were overcome using DSP techniques. Switching between the applications did not alter the measured performance and it only has a downtime (when switching), which does not exceed 200 ms. The radio applications were selected to show different types of waveforms and communication mechanisms. The results showed the feasibility of implementation on a GPP as long as latency was not an issue (otherwise, it should be implemented on the FPGA).

The positive outcome of the performed tests validated this prototype for the next phase, which will be designing and manufacturing the RF front-end and the antenna switching module. After testing this ensemble, the prototype will be tested in a real flight scenario.

Among the multitude of challenges addressed in this paper, achieving the minimum required sensitivity of TMS is the most difficult task to overcome. Because filtering options are not satisfactory, and relaxing the interrogation detection criteria will trigger unwanted replies (MOPS allows a maximum of five unwanted replies per second averaged over 30 seconds). The safest solution to improve TMS sensitivity is to use an ADC with a higher resolution. According to Eq.1, if a 14-bits ADC is used instead of a 12-bits ADC (the ADC of Radio420X) the sensitivity should decrease by 12.04 dB, which results in a sensitivity of -74.04 dBm.

Future work will include increasing the number of radio applications running simultaneously by using a custom transceiver. A potential solution is to use sub-sampling techniques in a low-IF architecture as described in [13]. A similar approach was adopted in [14] for Very High Frequency (VHF) radio applications yielding promising results. Another perspective is to implement a real-time OS with robust resource partitioning to comply with IMA standards. The developed avionic system is performing real test flight with adapted Power RF.

References

[1] T. Gaska, C. Watkin, and Y. Chen, 2015, "Integrated modular avionics - Past, present, and future," *IEEE Aerospace and Electronic Systems Magazine*, vol. 30, pp. 12-23.

[2] J. W. R. Ramsey. (2007). *Integrated Modular Avionics: Less is More*. Available: http://www.aviationtoday.com/av/issue/feature/8420.htm

[3] C. B. Watkins, 2006, "Integrated modular avionics: Managing the allocation of shared intersystem resources," *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, pp. 1-12.

[4] J. Zambrano, E. Zhang, O. Yeste-Ojeda, R. Landry, and V. Gheorghian, *2016*, "Development and implementation of new architecture for robust satellite data unit with software defined radio for airborne network,", *IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, pp. 1-10.

[5] W. Schütz and M. Schmidt, 2007, "Action Plan 17-Future Communication Study: Final Conclusions and Recommendations Report", *EUROCONTROL/FAA Memorandum of Cooperation, Tech. Rep,*.

[6] R. S. C. 149, 1985, "Minimum Operational Performance Standards (MOPS) for Airborne Distance Measuring Equipment (DME) Operating Within the Radio Frequency Range of 960-1215 MHz," in *RTCA/DO-189*, ed.

[7] R. S. C. 186, 2009, "Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)," in *Rtca/Do-260B*, ed.

[8] R. S. C. 209, 2011, "Minimum Operational Performance Standards for Air Traffic Control Radar Beacon System/Mode Select (ATCRBS/MODE S) Airborne Equipment," in *Rtca/Do-181E*, ed.

[9] B. Razavi, 1997, "Design considerations for direct-conversion receivers," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 44, pp. 428-435.

[10] T. J. Rouphael, 2014, "Wireless Receiver Architectures and Design," *Wireless Receiver Architectures and Design*,

[11] G. R. team. *GNU Radio overview*. Available: http://gnuradio.org/redmine/projects/gnuradio

[12] T. J. Rouphael, 2009, *RF and Digital Signal Processing for SDR*.

[13] O. A. Yeste-Ojeda and R. Landry, *2015*, "Integrated direct RF sampling front-end for VHF avionics systems," *Integrated Communication, Navigation and Surveillance Conference (ICNS)*, pp.L1-1-L1-11.

[14] A. Q. Nguyen, A. Avakh Kisomi, and R. J. Landry, 2017, "New Architecture of Direct RF Sampling for Avionic Systems Applied to VOR and ILS," *IEEE Radar Conference (RadarConf) (Accepted)*.

Acknowledgements

The works presented in this paper is a part of the AVIO-505 project at LASSENA, ÉTS. It is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Consortium for Research and Innovation in Aerospace in Quebec (CRIAQ) as well as three main strategic partners, namely Bombardier Aerospace, MDA, and Marinvent Corporation.

Email Address

abdessamad.amrhar@lassena.etsmtl.ca

alireza.avakhkisomi@lassena.etsmtl.ca

eric.zhang@lassena.etsmtl.ca

joe.zambrano@lassena.etsmtl.ca

claude.thibeault@etsmtl.ca

renejr.landry@etsmtl.ca

2017 Integrated Communications Navigation and Surveillance (ICNS) Conference

April 18-20, 2017