ARESTOR: A Multi-role RF Sensor based on the Xilinx RFSoC

Nial Peters[1], Colin Horne[2] Matthew A. Ritchie[3]
Department of Electronic and Electrical Engineering, University College London, UK
{[1]nial.peters [2]colin.horne.14 [3]m.ritchie}@ucl.ac.uk

***Abstract*— In this paper we present ARESTOR, a multi-role RF sensor based on a Xilinx RFSoC device. The system is capable of operating as an active radar, passive radar and wideband Electronic Surveillance (ES) receiver. The system design, and development framework used to enable these modes of operation is summarised. Preliminary results from each of the sensing modes are presented demonstrating what is possible from each mode. Furthermore, a strategy for synchronising multiple RFSoC devices is explored and bistatic active radar measurements using two synchronised RFSoCs shown.**

***Keywords*— multistatic radar, passive radar, FMCW radar**

I. Introduction

The style of modern and future RF sensing has changed significantly from the traditional single role large, static, non-dynamic that radar systems originated from. Due to the increasing demand for RF spectrum resources, and new challenges presented by Electronic Warfare (EW), modern RF sensors can no longer afford to be single function, and should look to provide agile sensing outputs that were previously completed by multiple separate hardware devices.

A recently available commercial off the shelf (COTS) device that represents a step change in capabilities is the Xilinx Radio Frequency System on a Chip (RFSoC) [1]. This device combines a powerful processing system, Field Programmable Gate Array (FPGA) and high speed ADCs and DACs. The tight integration of these key components with high quality specifications has created a hardware tool that can enable a diverse array of applications.

We have developed the RFSoC device from a blank template into a multi-role sensor, named ARESTOR, which is capable of operating as (1) Active Radar (2) Passive Radar and (3) Electronic Surveillance (ES) device. Typically these three roles are performed by totally separate systems, which have been designed in isolation. Each of these devices either transmit, receive or do both and therefore by using a flexible digital system with highly capable specifications it has been possible to develop a device that can fulfil all three roles.

Little prior literature includes both active and passive radar sensing modes performed by the same device as shown in this paper. The RFSoC system has been proposed for use as a radar system [2]. However, this work focused only on active radar and not a multi-role configuration, but did show experimentally how the RFSoC is a potential effective digital back-end for a phased array configuration. The concept of joint active and passive sensing was reviewed within [3]. This paper focused mostly on passive radar with active modes which were designated as "fall back" modes for the sensor network. It does conclude that joint active and passive sensing provides a more resilient surveillance sensor network overall.

This publication presents the recent development of the ARESTOR system, a multi-function RF sensor based on RFSoC hardware and capable of operating in multiple different modes. Section II describes the system design methodology and implementation including a strategy for synchronous operation of multiple RFSoC devices. Section III shows the first results produced using the system experimentally with real targets in each of its sensing modes. Finally, Section IV describes the conclusions and planned future work for this research.

II. Methodology

The hardware used for this work was a ZCU111 evaluation board from Xilinx [4]. This board contains a "Gen 1" RFSoC device with 4 GHz of analog bandwidth, 8 ADCs operating at 4 GS/s and 8 DACs operating at 6.5 GS/s.

A. System Design

The need to implement many different configurations for the RFSoC to enable different RF sensing modes led us to create a development framework to expedite the process. This framework uses scripts written in TCL to automatically create FPGA configurations based on a small number of user-specified parameters.

The main parameters for any configuration are the specification of what transmitters and receivers are required in the Programmable Logic (PL). In this context a transmitter is a hardware block which delivers data to one or more DACs and a receiver is a hardware block which receives data from one or more ADCs and makes it available to the Processing System (PS) via one or more Direct Memory Access (DMA) engines. The design of the transmitters and receivers is as generic as possible, with customisation available through optional data processing hardware blocks. Different RFSoC configurations each contain different combinations of transmitters and receivers of different types, along with other generic design components (e.g. triggering infrastructure) which are reused between configurations.

Figure 1 shows a simplified block diagram of the transmitter and receiver design. Both transmitters and receivers use a "PRF Control" system to modulate data flow. This is a set of hardware counters capable of producing complex sequences of data-flow modulations, allowing precisely timed pulses of data transmission and reception. Block-RAM FIFOs are used as buffers into and out-of the DMA engines, which transfer data to and from DDR memory. Receivers make use of the DDR on the PS, since this can have up to 32 GB installed. The DMA engines on the receivers transfer data directly into user-space memory buffers using a zero-copy Linux kernel module that we developed. This allows efficient access to received data by user-space software. We have achieved transfers up to 29 GB in size at a rate of 10 GB/s using this system. Transmitters use pre-computed waveform data stored in the DDR attached to the PL, and once triggered do not require access to the PS. This prevents conflicts with receivers for PS to PL bus resources, which are limited. Should transmit waveform adaptivity based on the captured signals be required, waveform design might take place directly in the PL, or in the PS, with the new waveforms being downloaded to the transmit buffers in PL memory during idle time between transmit bursts.

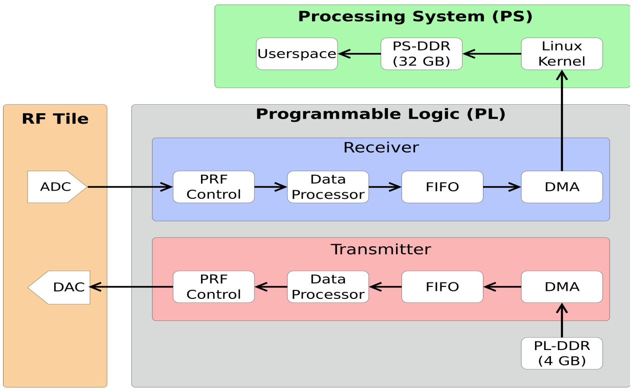

Fig. 1. Block diagram of transmitter and receiver design implemented by our framework within the RFSoC.

Software control of the system is through a Python API we developed. This is built on top of automatically generated wrappers for underlying C libraries, which in turn control the hardware components through a suite of Linux drivers. The software stack automatically configures itself based on the current RFSoC configuration, allowing the same API to be used for all RF sensing modes.

B. *Passive/FMCW Receiver Design*

The RF tile on the RFSoC provides functionality for down-mixing and decimating the data from each ADC. For the ZCU111 the maximum decimation factor is 8, resulting in a data rate of ~500 MS/s (I&Q data). For many applications >500 MHz of bandwidth is not necessary, and is wasteful in terms of data quantity. This is the case for both the FMCW active radar configuration and the passive radar configuration, making the addition of extra decimation stages within the PL desirable.

For FMCW the most benefit of extra decimation can only be achieved if the first stage of FMCW processing, the deramping to determine beat frequency, is also carried out within the FPGA fabric, prior to decimation. Before deramping, the bandwidth might be of the order of several hundreds of MHz, depending on the application and bandwidth restrictions. However, after deramping the useful bandwidth of the resulting signal is very small compared with the transmission bandwidth, typically of the order of hundreds of kHz to a few MHz. Our FMCW receiver design therefore performs the deramping as the first stage in its processing block.

Passive radar depends on illuminators of opportunity hence the bandwidth of the received signals is limited by the illuminators available. Two commonly used illuminators are DVB-T transmissions and Wifi routers. Typical bandwidths of DVB-T and Wifi are 8 MHz and 20 MHz respectively, although higher Wifi bandwidths (i.e. 40 MHz) are possible.

The comparison between FMCW radar and passive radar illustrates that a single decimation factor is inadequate to properly cover multiple applications. To solve this, we created a generic decimator structure which can be inserted into receivers as part of their data processing block. The decimator can be configured to operate on a single channel (e.g. for post-deramp FMCW), or two channels (e.g. reference and surveillance for passive). We use decimation factors of 8 or 16 for passive radar (depending on the required bandwidth) and 256 for FMCW. This significantly reduces the data throughput and storage requirements for both of these sensing modes.

Data from the ADCs arrives on a parallel data bus with a width of 8 samples. Our decimators are designed to be used only when 8 times decimation is applied on the RF tile, so the clock rate of this bus is $\sim$ 60 MHz. We found that serialising this data onto a bus with a width of 1 sample, but a clock rate of 500 MHz, allowed for a far more efficient (in terms of FPGA resources used) decimator design. We then pack the serialised data back onto a parallel bus before passing it to the DMA engine.

C. *Multi-device synchronisation*

RF sensing applications that utilise more than one RFSoC device (e.g. multi-static radar), require synchronisation between them. The RFSoC itself provides a mechanism for synchronising the sampling clocks for ADCs and DACs on separate devices, and for matching latencies through the data pipelines. However, this system requires two external “SYSREF” clock inputs, one of which is not available on the ZCU111 evaluation board. Our design overcomes this issue by generating both SYSREF clocks on-board, and synchronising them with an external clock. This allows multiple ZCU111 boards to be synchronised. Furthermore, we have implemented hardware within the PL to facilitate synchronous triggering, and also synchronisation of the phases of the built-in mixers of the RFSoC, across multiple devices. Thus, given a common clock source and trigger, our design can achieve phase-coherent measurements from two or more separate ZCU111 boards.

D. Dynamic Role Switching

The 8 ADCs and 8 DACs available on the ZCU111 allow for multiple RF sensor configurations to be implemented within the same FPGA configuration, allowing different sensing modes to be operated simultaneously. However, complex sensor designs may require significant quantities of the available FPGA resources and may not fit in the FPGA alongside other sensor configurations. In this case, multi-role capability can be achieved by switching between different FPGA configurations as required. Changing the FPGA configuration requires two components: a bitstream and a device-tree overlay. The bitstream contains the configuration information for the PL. The device-tree overlay describes the hardware components contained in the bitstream to the operating system (Linux running in the PS) allowing it to load and configure the required drivers to interface with them. Our system design allows many different bitstream and overlay files to be stored on the device and dynamically switched between at runtime (without requiring a reboot). The time taken to change configuration depends on the complexity of the design being loaded, but is typically a few seconds.

III. Results

In this section we present brief results from sensing configurations built using the framework and tested either in over-the-air trials or, in the case of the ES mode, in wired loop-back trials of wide-band signal captures.

A. FMCW Radar Mode

All active sensing modes for the trials reported here use the FMCW radar design, section II-B. A single channel example design was tested with the target being a person standing still for a number of seconds, followed by walking towards and away from the sensor. The captured data was processed to produce the plots in Fig. 2, the upper plot shows the RTI with moving target indication processing (MTI) applied, while the lower shows the Doppler-time plot, also with MTI applied. The RTI with MTI shows the target range variation over several seconds of capture. The Doppler-time plot clearly reveals the micro-Doppler signature of the swinging arms and the bulk Doppler caused by the walking target.

More complex designs employing multiple transmit and receive channels have also been constructed, allowing half-pol and full-pol data captures.

B. Passive Radar Mode

Demonstration of a passive radar mode makes use of a Wifi router as the illuminator, configured to transmit continuously with a 40 MHz bandwidth using the MGEN Tool [5]. The signals are decimated by 64x within the RFSoC, providing net sampling rates in each channel of 60 MHz.

A pseudo-monostatic radar configuration was used due to limited trials space. The target consisted of a person walking a straight line path towards and away from the sensor, over a range of approximately 10 m.

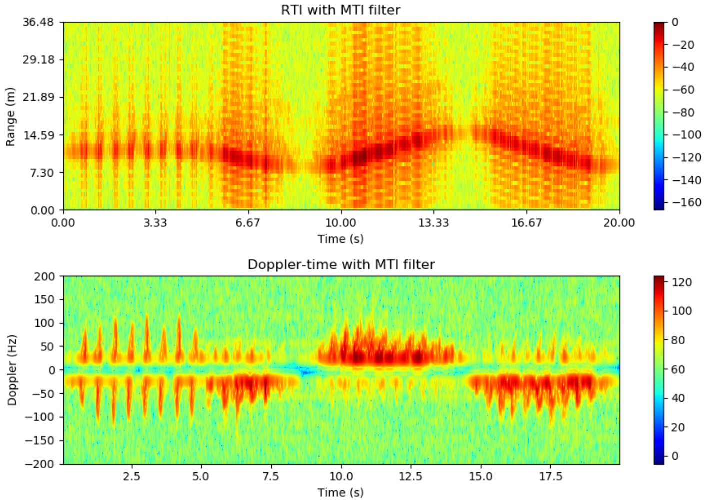

Fig. 2. RTI with MTI (top) and Doppler-Time (bottom) plots of walking person target from FMCW radar configuration

Passive radar processing [6][7][8] was implemented offline on a PC to filter the direct signal interference (DSI) from the surveillance channel, and to create the ambiguity surface by cross-correlating the two receiver channels. The result is further processed through a constant false alarm rate (CFAR) algorithm. Figure 3 shows a 10 s sequence of plots from the CFAR processing at 2 s intervals, which illustrates the range and Doppler variations seen in the walking person data over time.

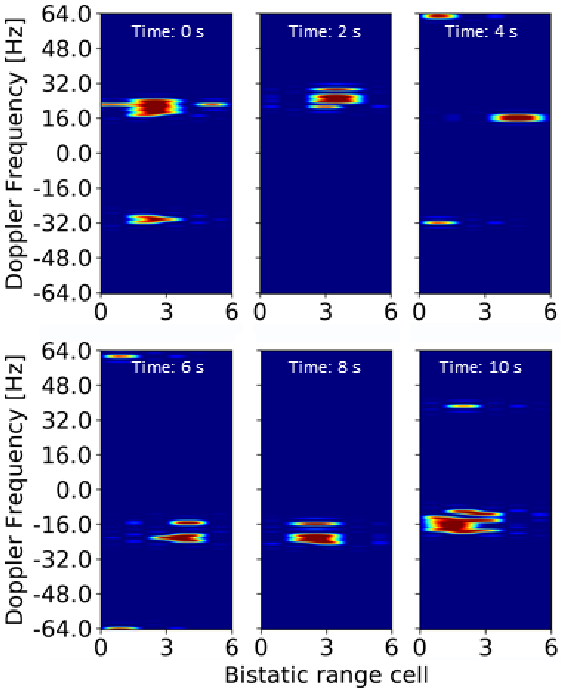

Fig. 3. 10 s sequence of passive radar CFAR output plots (with DSI suppression) of person walking. 2 s intervals