# Low Cost FPGA based Implementation of a DRFM System

M.B. Mesarcik University of Cape Town
South Africa
Email: msrmic004@myuct.ac.za

***Abstract*—Digital Radio Frequency Memory (DRFM) is a technique used to record an incoming Radio Frequency (RF) signal, in turn applying a series of time-delays, amplitude scalings and frequency shifts and retransmitting the signal. This technique is used widely in the electronic defense industry as a form of radar jamming, in that it allows for the synthesis of artificial targets. This paper discusses the design of implementation of a DRFM system on a low cost FPGA system and documents this system's performance. Describe results of the implementation sdf sadf sdf asdf asdf asdfsdfasdfadfsd fsdlfjlsdjflsj sdjfl sdjfl jsdlf jsdlf jsdlfj sdljsdlfjsdl jasdfsdjl asdjflsdjl sdjfsdlfj sldfjaslf jdlfjsdlfaljflasjflk jsdlf jasdlfjaslfjsdflsjadlfsjadlf jsdfl jasdlf jsdlkf jasdlf jasdl jasdlasdjl asdjlj lfjadlfjsfjasdflajsflajsfljas lkasdf lsadj lfjsdfl a asdf asdf asd fasf asdf asd asd fasd fasdf asdf asdf asdf asd asd afasd fasdf sdfasf asdfsfsdfsdfsd dfg dfg dfg dfg sr wer sdfl ;'**

## I. Introduction

### A. Digital Radio Frequency Memory

Digital Radio Frequency Memory (DRFM) systems digitize and store incoming RF input signals at specific frequencies and bandwidths. The captured signals undergo time delays, frequency shifts and amplitude scalings, after which they are retransmitted. DRFM is used in order to synthesize false targets for radar systems by exploiting the radar assumptions of target identification. This is achieved by retransmitting a time delayed, amplitude scaled and frequency shifted coherent replica of the inputted signal, thereby making it, from the radar's perspective, indistinguishable from other genuine signals [1].

The reason why a radar system is deceived is due to its fundamental operation. For example, a pulsed Doppler radar operates by transmitting pulses on a target and measuring the time delay and frequency shift of the return pulse from the target in order to infer information about a target's location and velocity. The implications of this are such that if we simulate echos with certain frequency shifts and time delays it is possible for a radar to identify a target incorrectly.

An illustration of a DRFM system may be seen in Fig. 1. It can be seen that the input signal is mixed down to intermediate frequency and then is sampled by an Analog to Digital Converter (ADC). The sample rate of the ADC is equal to the bandwidth of the incoming RF signal as the ADC is sampling complex I/Q data. The digitized signal is then stored in memory and may be manipulated by means of the control interface. The control interface is used to designate the various time delays, frequency shifts and amplitude scalings.

### B. FPGA based acceleration

Due to the high data rates and the Digital Signal Processing (DSP) requirements the implementation DRFM systems are highly implementable on Field Programmable Gate Arrays (FPGA). Therefore this paper describes the design and implementation of a DRFM system on a low cost FPGA, namely the Altera MAX 10 [2].

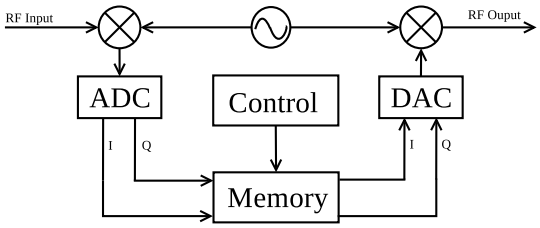


Fig. 1: Illustration of DRFM System

### C. Plan of Development

This paper begins by discussing the conceptual design of the proposed DRFM architecture by giving an overview of its implementation. After which, the details of each subsystem is described. Finally, results are shown and conclusions are drawn based on the prototype design.

## II. Overview

For means of explanation, the architecture of the FPGA based implemented DRFM system can be described by three subsystems. These being, the interfacing and control system, the external peripherals and the DSP system. Whereby each of the constituent systems are monitored and controlled by the top-level controller module. This simplified architecture may be seen in Fig. 2.

It must be noted that the architecture described in Fig. 2 has not yet been integrated into a RF front end and therefore this paper is a review of the work in progress. The implications of this are that although real time signal processing and data storage are realizable at the speeds dictated by the RF front-end, it has not been proven that this architecture is in fact capable of achieving such speeds.

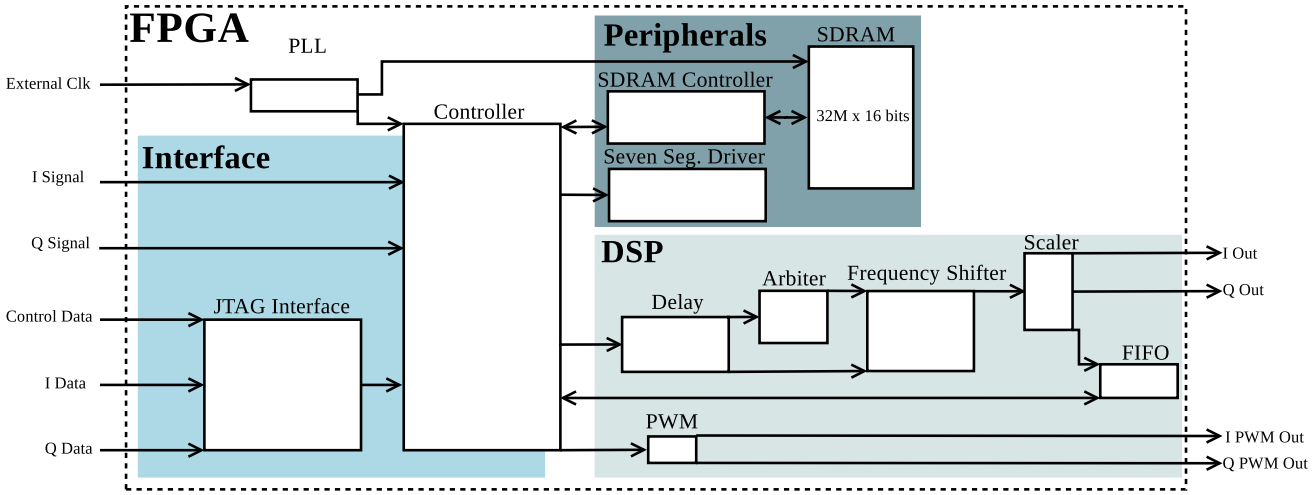


Fig. 2: Illustration of the implementation of the FPGA based DRFM System

### A. JTAG Interface

Joint Action Test Group (JTAG) was originally used as a simplified standard for interconnectivity testing for PCB manufacturing. It was created as a result of Integrated Circuits (IC) pins becoming more densely spaced and therefore classical interconnectivity test approaches were infeasible. The JTAG protocol mitigated the need for physical access to an IC's pins through the use of shift register chains.

In the JTAG standard (IEEE 1149.1) the collections of shift registers are referred to as Boundary Scan Cells (BSCs) which sample and hold data from and to the pin interface. As a result the BSCs can be configured into a serial shift chain thereby allowing to be driving by a serial interface.

The JTAG interface implemented on the DRFM system allows for software based monitoring, updating and controlling of the DRFM system through the JTAG port. Through the use of the Altera Virtual JTAG Intellectual Property (IP) core it is possible to access to the JTAG control signals that are routed to the FPGA core, which allow for a fine control over the JTAG resources thereby granting real-time general purpose serial communication [3].

Through the use of a Python maybe mention the pyqt model in a subsection based User Interface (UI) that runs a TCL JTAG server it is possible to transfer both I/Q data as well as control information to the FPGA as may be seen in Fig. 2. The control word consists of 64 bits with flags and data contained within its length. The control word composition may be seen in Table. I given that the most significant bit is at position 64.

TABLE I

CONTROL WORD COMPOSITION

| Control Information | Control Flag Position | Control Data Length |
|---|---|---|
| Doppler Shift | 33 | 32 |
| Time Delay | 44 | 10 |
| Amplitude Scale | 61 | 16 |

### B. External Peripherals

In order to account for the requirement of I/Q data injection, an external SDRAM memory was incorporated into the DRFM system. It was used as a means to store the injected I/Q data in order to recall it for the DSP operations of the DRFM system. The SDRAM device is a 512Mb high speed CMOS dynamic access memory with 32M of address space that address 16 bit words [4].

maybe discuss how I/Q data is dealt with in terms of inject. How only have 16bit words and each I/Q channel is 16 bits.

An SDRAM controller was implemented on the FPGA in order to facilitate communication with the external SDRAM peripheral. The controller made use of the Altera SDRAM Controller IP Core which uses the Avalon Interfacing standard to communicate with the SDRAM device. This was integrated into the system through the use of the Altera Qsys integration tool, that allowed for physical pin allocations to the external SDRAM device [5].

In addition to this it was required to clock the external SDRAM device at different clock rate. Therefore it was necessary to implement a Phase Locked Loop (PLL) in order to synchronously step down the clock speed.

Finally, for debugging and user interfacing purposes a group seven segment displays were used. The seven segment displays give visual feedback to the operator of the system by displaying the current state of the DRFM system.

### C. Signal Processing

As previously discussed, the digital signal processing aspect of a DRFM system is critical to its operation. As time delays, frequency shifts and amplitude scalings are needed in order to emulate a target from a radar system's perspective.

The major consideration in the implementation of this

signal processing scheme was the order in which the signal processing operations were performed. This was due to the real-time processing requirements of a DRFM system, such when performing variable time delays there would be no loss of coherence between the input and output signals. This meant that if a time delay was set by the user and a frequency shift was then applied, the system should not have to repopulate the delay buffer. For this reason the order of the signal processing operations were to first perform the time-delay, then frequency shifts and finally amplitude scaling, as can be seen in Fig. 2.

In order to describe the signal processing model the signal, $s(n)$, is considered to be the input to the signal processing chain. As the DRFM system receives I/Q data $s(n)$ may be expressed mathematically as THIS IS WRONG

$$s(n) = v_{rx}(n)cos(2\pi f_{lo}n) + v_{rx}(n)sin(2\pi f_{lo}n) \quad (1)$$

where $v_{rx}(n)$ is the received RF signal and $f_{lo}$ is the local oscillator frequency.

*1) Time Delay:* The time delay operation may be expressed as a method of altering the index of the input signal $s(t)$. This process may be summarized mathematically by

$$s_{delay}(k) = s(n-k), \quad n \geq k \quad (2)$$

This then may be implemented on a digital system by injecting a portion of the input signal $s(n)$ into a RAM. After which, it is possible to index previous values of $s(n)$ by means of the RAM's address pointers. Which then allows for the outputting of previous values of the inputted signal given that capacity criteria has been met. This being that the delay address index, $k$, is less than or equal to current RAM address index, $n$. In the implementation of the DRFM system, a dual port, 1024 address long, 16 bit wide RAM is implemented internally on the FPGA.

The limitations of this are that 1024 samples must be maintained in the RAM, thereby inhibiting the DRFM system from being able to output data for first 1024 clock cycles of operation. Naturally, this is an overhead of any DRFM system as real-time data must be sampled and stored. However, once the capacity criteria of the RAM have been met a single sample may be outputted every clock cycle.

For this reason, the signal processing chain was structured as it is. This was because had a user defined variable frequency shift occurred before the delay, the RAM would have had to be refilled as previous values of the frequency shifted signal would need to be outputted.

Finally, as the time delay control data word was specified to be 1024, it was necessary to maintain 1024 samples in the RAM at any point.

*2) Frequency shifting:* I OVER LOOKED THE FREQUENCY SHIFTED. NEED TO FIXED TOMORROW.
a

Principals of I/Q Data
Derive Mathematically how the shift works

*3) Fractional Amplitude Scaling:* a
input valies

## III. Methodology

## IV. Design

### A. System Peripherals

*1) PLL:*
*2) SDRAM Controller:*
*3) External SDRAM:*
*4) PWM:*

### B. Interface

*1) JTAG Interface:*
*2) Controller:*

### C. DRFM Subsystem

*1) Delay:*
*2) Arbiter:*
*3) Frequency Shifter:*
*4) Scaler:*

## V. Results

## VI. Conclusions and Future Work

### A. Future Work

– Streaming to file. – DAC for sensing.

## References

[1] S. J. Roome, "Digital radio frequency memory."
[2] *DE10-Lite User Manual.*
[3] *Virtual JTAG (altera_virtual_jtag) IP Core User Guide.*
[4] *ISSI SDRAM Data Sheet.*
[5] *SDRAM Controller Core User Manual.*