10.1109/ULTSYM.2014.0478

Dynamically Reconfigurable Analog Front-End for Ultrasonic Imaging Applications

Vidya Vasudevan, Pramod Govindan and Jafar Saniie
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, Illinois, U.S.A.

***Abstract*—With extensive applications in imaging and material evaluation, ultrasonic systems have evolved over decades to efficient and more sophisticated equipment. These systems are built aiming at a specific application or at specific target material. In order to make these systems portable and adaptable to the testing environment, we focus on building a more flexible and programmable hardware. We present a fully configurable Analog Front-End (AFE) which possesses the capability for dynamic re-configuration by using an ARM Core for real-time control, data acquisition and signal analysis. The flexibility built into the AFE facilitates various beamforming and signal conditioning requirements. This arrangement enables the back-end processor to support various signal processing algorithms.**

I. Introduction

Ultrasound imaging or sonography uses sound waves for non-invasive imaging of internal organs and soft tissues in the human body. Ultrasound data gathered in real-time is often displayed as a 2D image for examining the internal anatomy without exposure to radiation. Typically, during ultrasound examinations, a hand-held ultrasonic transducer is placed against the target area under study. This transducer operates as both a transmitter and receiver by converting the electrical pulses into mechanical vibrations incident on the target material and relaying the reflected vibrations or *echoes* to the processing system which displays it as a 2D image on a screen. Processing image frames in real-time requires a high processing capability in the system built for this application. Using devices such as microcontrollers, FPGAs, system-on-chips (SoCs) have, over time, revolutionized implementation of embedded systems and improvised over the designs of the past to provide more independent features and capabilities in a portable hand-held unit. With this motivation, we pursue the task of building a reconfigurable system for ultrasonic imaging.

A distinctive ultrasonic sensor system operates with a high voltage (HV) pulser driving high frequency pulses through an ultrasonic transducer. Typically, a controller and a processing element are present to control its actuation and beam steering. The electrical excitation pulses are converted to mechanical vibrations resulting in a series of pressure waves which are incident onto the target body. The beam incident on the target reflects back and this echo signal is channeled through the receiver consisting of signal conditioning circuit and data converter. The digitized information content is subsequently processed by the processing element. The design considerations involved in each of these blocks is purely dependent on the application and the target material to be evaluated [1]. The features including incident beam pattern, the center frequency of operation, signal conditioning and data conversion are to be corrected for optimal results and these are a drawback in a hardwired systems which does not provide support for future upgrades. The components: beam controller, noise attenuator and amplifier provide such enhancement features in our system. Ultrasonic measurement equipment operates in frequency range as dictated by the transducer which can vary from 20KHz to 20MHz. Thus, the data acquisition unit chosen must possess the capability to sample the received echo with minimum distortion within the chosen range of frequencies.

In this study, as we concentrate on developing a programmable system by enhancing its capabilities, a fully controllable front-end pulser/receiver sub-system is presented. The back-end consists of an ARM processing core providing the capability to process computationally intensive signal processing algorithms [2].

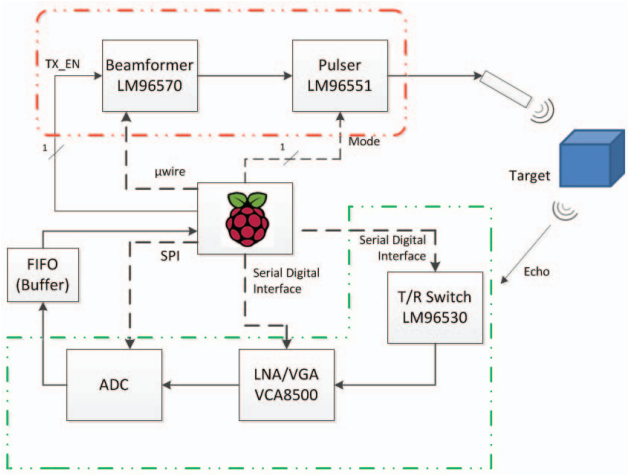

Figure 1 Proposed Ultrasonic System

Our ultrasonic testing system, shown in Figure 1, comprises of the front-end sensor interface with an associated conditioning block and a back-end processing/controller unit. The front-end hardware, shown in Figure 1, is built with Texas Instruments (TI) LM96551 HV pulser, LM96530 Transmit/Receive (T/R) Switch, and LM96570 Beamformer [3]. Noise filtering for the received echo is facilitated by the low noise pre-amplifier (LNP) followed by a variable gain

amplifier (VGA). LNP and VGA together provide a maximum gain of 50dB. A high performance ADC is used to convert the echo data into digital samples for processing. Once the data conversion is completed, a strobe End of Conversion (EOC) triggers the execution of the signal processing algorithm. This AFE is built around a Raspberry Pi (RPi) system in a Linux environment which provides active control for configuring the settings on the AFE components. Given the data storage limitations on the Raspberry Pi, an ADC with a sampling rate of 200MSPS requires an efficient data sample accumulation system such as a FIFO which can be used to access data by the ARM core. Further, this system is expected to be exported to a Xilinx Zynq based system which provides additional computational capability.

In this paper, Section II describes the proposed system architecture and its features. Section III details the control over excitation & signal conditioning sub-systems. Section IV discusses the results obtained during testing. Section V concludes this paper.

II. System Features

A programmable pulser and digital beamformer forms transmit and excitation path as shown in Figure 2 (a). The T/R Switch, Programmable Gain Amplifier (PGA) and a high-speed ADC forms receive and data acquisition path as shown in Figure 2 (b). On the transmit channel, the beamformer is a programmable device capable of configuring the delay pattern and pulse train required to set the desired transmit focal point up to 64 pulses with 12.5ns pulse duration making it suitable for a variety of ultrasonic imaging and evaluation applications. This 8-channel digital beamformer generates the pulse pattern in positive and negative levels for the high-voltage pulser control inputs serving as the sequence generator. Each channel drives a single transducer building the capability of this system to support up to 8 transducers. The pulse pattern on each of the 8 channels: CH0 to CH7 is programmed into the internal registers for positive and negative pulse trains. This is used to adjust the temporal shape of the ultrasound pulse. Each rectangular pulse at the pulse output results in a mechanical pulse output shown in Figure 3.

Various configurations for beamforming and beam steering are supported by the beamformer. This helps in realizing some of the ultrasonic signal processing algorithms such as discrete wavelet transform (DWT), chirplet signal decomposition (CSD), split-spectrum processing (SSP) and coherent averaging (CA). In this study, the T/R switch provides an 8-channel programmable receive side interface with each Rx switch channel driven by a HV pulser that is directly connected to a transducer. The T/R switch protects the receiver input from voltage spikes due to leakage currents flowing through the switches on Rx channel. Each T/R switch can be individually programmed ON or OFF allowing for low power operation by selectively configuring desired channels through SPI.

The back-end processing and control is coordinated by a Raspberry Pi model B based on ARM 11 microarchitecture [4]. The main components of this system are the ARM A11 CPU which supports 32-bit advanced ARM instructions. Raspberry Pi Model B has 512MB RAM, two USB ports and an Ethernet port. It has a Broadcom BCM2835 system-on-chip (SoC) which includes an ARM1176JZF-S 700 MHz processor, Video Core IV Graphics Processing Unit (GPU), and an SD card. The GPU is capable of Blu-ray quality playback, using H.264 at 40MBits/s. It has a fast 3D GPU accessed using the supplied OpenGL ES2.0 and Open VG libraries. The chip specifically provides HDMI but no VGA support.

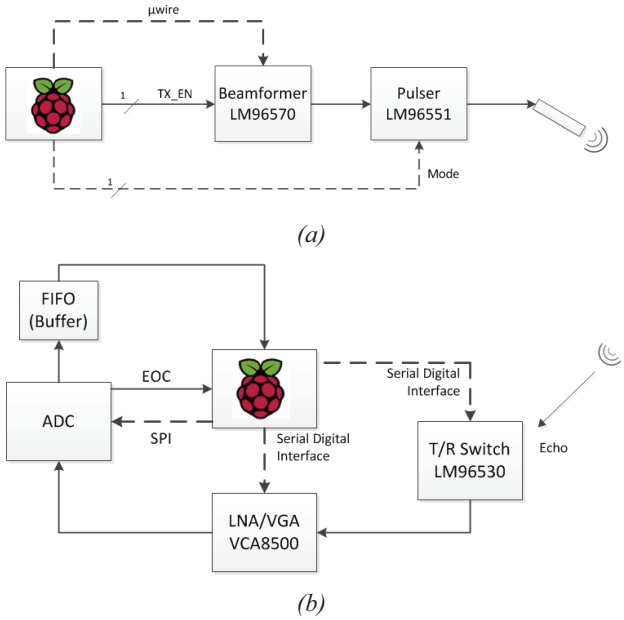

Figure 2 (a) Transmit/Excitation path block diagram (b) Receive/Data Acquisition path block diagram

Figure 3 Ultrasonic transducer operation

Each component in this ultrasonic system is configured by the ARM core through a Serial Peripheral Interface (SPI) or Serial Digital Interface (SDI). The choice of the ADC depends on the frequency of operation and the sampling rate. In our test setup, we are currently using the AD7829 8-bit 8-channel flash ADC which has a simple parallel interface for data transfer to the processing system [5]. This device supports up to a maximum throughput of 2MSPS. An upgrade to this is the AD9272 [6] which supports with an 8-channel 12-bit pipelined ADC architecture sampling at 10MSPS to 80MSPS. This device further provides a compact amplifier/filter network which provides an efficient support for Time Gain Compensation (TGC) for physiological signal attenuation. Another high performance prototype includes Analog Devices AD9467: a single channel 12-bit 250MSPS ADC, with a throughput of 500MB/s for a 250MHz clock.

III. Dynamic Control

The features offered by the front-end can be configured based on the target material and application. Table 1 provides the various characteristics that can be programmed and lists the devices which need to be programmed to achieve the desired configuration.

Table 1 AFE Programmable features

Feature	Range	Devices
Channel selection	1 to 8	BF, TR, DC
Pulse Pattern Length	4 to 64 pulses	BF
Frequency	625KHz to 80MHz	BF
Pulse Delay	0.78ns to 102.4µs	BF
Echo Amplification	40dB, 43db, 47dB, 50dB	VC
Sampling Frequency	*10 to 80 MSPS	DC
Transient selection	50V/600mA or 50V/2A	PR

BF: Beamformer (LM96570); TR: TR Switch (LM96530); PR: Pulser/Receiver (LM96551); VC: Amplifier (VCA8500); DC: ADC (*AD9272)

Based on the BCM2835 SoC, the Raspberry Pi is built around a high definition embedded multimedia applications processor with a ARM1176JZ-F core, Broadcom video core IV, 256MB LPDDR2-800 memory, composite HDMI interface, SD card slot, 2xUSB2.0 and 10/100 Ethernet.

The BCM2835 peripherals accessible by ARM are USB, PCM, DMA, I2C, SPI, PWM, UART, Timers and GPIO [7]. Specifically, this device has 3 SPI controllers out of which only SPI0 is available at the GPIO ports on the Raspberry Pi model B P1 connector. This controller supports SCLK, MISO, MOSI signals in a master mode among the slave devices via the chip-selects CE0 and CE1. The on chip hardware supports 2-/3-wire SPI with data transfer via interrupts, polling or DMA. As shown in Figure 4, this application consists of 3 slave devices and therefore, alternative GPIOs are being used for Slave select (SS_x). The hardware pins used by the SPI controllers for this application are listed in Table 2.

Table 2 Raspberry Pi (model B) GPIO connector table

SPI Signal	BCM GPIO pin #	RPi connector #
SCLK	GPIO11	P1-23
MOSI	GPIO10	P1-19
MISO	GPIO09	P1-21
SS_0	GPIO14	P1-08
SS_1	GPIO15	P1-10
SS_2	GPIO18	P1-12

The physical connection made on the hardware between Raspberry Pi and the slave devices are shown in Figure 4. Each of these devices shares the SPI signals except the Slave Select. With limited fan-out of the BCM2835 outputs: SCLK, MOSI; non-inverting buffers have been incorporated into the design for those signals to enable sufficient drive capability for high speed signals.

The BCM2835 system uses the AMBA AXI-compatible interface structure. In order to keep the system complexity low and data throughput high, AXI on BCM2835 does not always read data in-order. The GPU has special logic to cope with data arriving out-of-order; however the ARM core does not have such logic. Therefore, it is prudent to ensure precautions while using ARM to access each of those peripherals in turn.

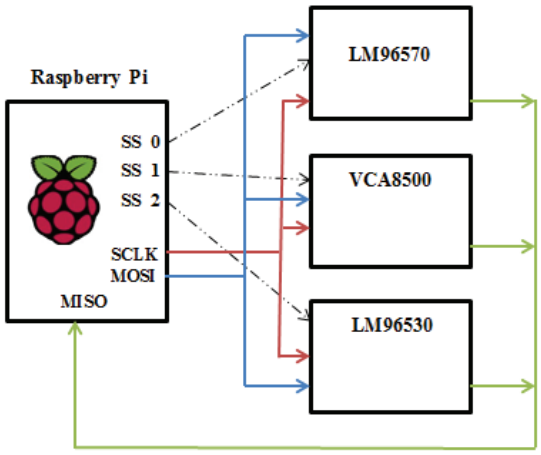

Figure 4 Raspberry Pi interface to the AFE components

The SPI masters once enabled can be used to transfer data with a length of 4 bytes in a single beat. It must be noted that the Raspberry Pi when loaded with the NOOBS OS [4] for the first time, SPI module is not enabled and it needs to be removed from the *raspi-blacklist.conf* file present as part of the kernel in our linux-based system. Following this procedure, the 2 SPI controllers are listed under the directory /dev as SPI0 and SPI1. Both of these SPI peripherals have a 32-bit wide FIFO which holds 4 bytes. The serial clock for SPI transaction can vary from 7.6kHz to 125MHz in discrete steps. Various features such as clock frequency, clock phase, clock polarity, least significant bit first (LSBF), clock inversion and data out hold time can be controlled based on the target slave to be programmed. With the 4 combinations of clock polarity (CPOL) and clock phase (CPHA), the SPI operates in 4 modes: mode 0 to 3. The clock phase and clock polarity are characteristics which define the combination of clock edge and active transition when the data on the SPI lines are latched-in/out on the MISO/MOSI links.

Based on SPI controller features, the specific parameters of the SPI protocol for the AFE components, LM96530, LM96570 and VCA8500 are presented in Table 3. The SPI protocol is implemented in a C-program using the existing *bcm2835.h* library [8] which provides direct access to the SPI control and status registers. By writing into the corresponding registers after the peripheral is enabled, data can be written into the slaves in the following sequence: LM96530 (Beamformer), VCA8500 (VGA/LNA) and LM96570 (TR Switch). VCA8500 and LM96530 support up to 10MHz for serial data transfer and therefore, the frequency for those devices is set at 7.8MHz and that for LM96570 is set at 62.5MHz. After programming the AFE components, TX_EN signal is fired by the ARM processing core to trigger transducer excitation and subsequent data acquisition through the ADC. The digitized data is then processed through the ARM linux environment with the signal processing algorithm or accessed via a GUI interface for displaying the ultrasonic scan of the target.

Table 3 SPI settings for AFE communication

SPI Feature	LM96570 Beamformer	LM96530 TR Switch	VCA8500 PGA/LNA
Clock Frequency	62.5MHz	7.8MHz	7.8MHz
Clock Phase	Low	Low	High
Clock Polarity	Low	Low	Low
Mode	0	0	1
LSB First	No	No	Yes
Clock Inversion	No	No	No
Data Hold time	100ns	50ns	80ns
Max. Bytes in a burst	8	1	5
Time to program	1.5µs	1µs	5.2 µs

IV. Results

The BCM2835 based Raspberry Pi (model B) controls the components on the AFE through a serial interface and provides excitation for the immersion-type 3.5MHz piezoelectric transducer. Each of the AFE devices is sequentially programmed with the desired data. An instance of the chosen pulse pattern in our study is shown in Figure 5 (a). This pulse pattern is configured with the settings listed in Table 4. Our system test setup is shown in Figure 5 (b). The incident pulse pattern results in an echo response as shown in Figure 5 (c). Four bursts of echo can be seen each of which represents the surface boundary between metal and the liquid medium. Depending upon the quality of echo received, certain parameters on the AFE is re-programmed by a trigger from the ultrasonic algorithm on the ARM core to ensure sufficient signal quality for further processing.

Table 4 Test Settings for AFE

Device	Settings
Beamformer/TR Switch	16-pulses, 2.5MHz, 1-channel custom pattern: 1111111111111111
VCA8500	Gain set to 43dB, 10-MHz filter

V. Conclusion

Advancement in technology is derived through the need for transformation in the existing capability. Through the use of advancement in the field of programmable hardware, a simple ultrasound system evolves into a very flexible and efficient yet complex system which provides means for adaptability. Dynamic reconfiguration enables run-time decision making to improvise on the system productivity. We have built this system around a powerful ARM core which supports various ultrasonic signal processing algorithms including beamforming, beam-steering and pseudo-random signal analysis.

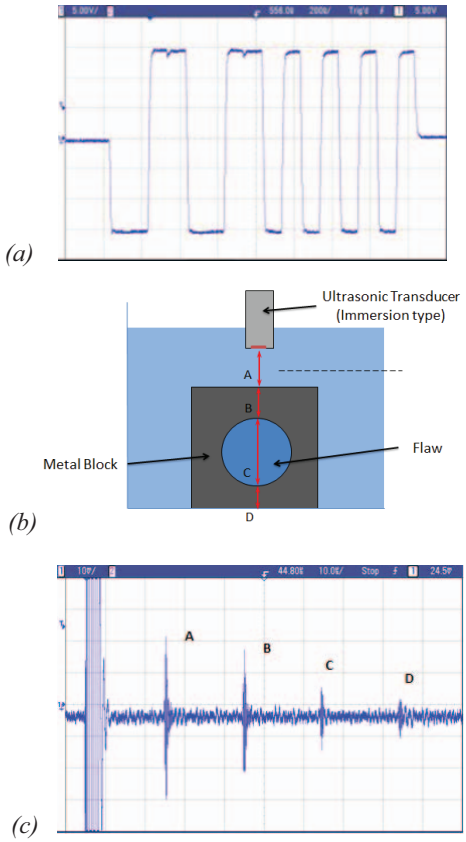

Figure 5 (a) Test pulse pattern (b) Test setup (c) Echo received for the test pulse sequence

References

[1] Analog Devices, 2002, "Ultrasound System Considerations and their Impact on Front-End Components" [Online]. Available: http://www.analog.com/library/analogdialogue/archives/36-03/ultrasound/ultrasoundfrontend.pdf

[2] S. Gilliland, P. Govindan, T. Gonnot and J. Saniie "Performance Evaluation of FPGA based Embedded ARM Processor for Ultrasonic Imaging", IEEE Ultrasonics Symposium, pp. 519 – 522, 2013

[3] Texas Instruments, 2013, TX-SDK-V2 Evaluation Board for the Ultrasound Transmit Solution. [Online]. Available: http://www.ti.com/tool/TX-SDK-V2

[4] Raspberry pi [Online]. Available: http://www.raspberrypi.org

[5] Analog Devices, 2006, AD7829 2MSPS 8-bit 8-channel Sampling ADC [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD7822_7825_7829.pdf

[6] Analog Devices, 2009, AD9272 Octal LNA/VGA/AAF/ADC and Crosspoint Switch Datasheet. [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD9272.pdf

[7] Broadcom BCM2835 ARM Peripherals [Online]. Available: http://www.raspberrypi.org/wp-content/uploads/2012/02/BCM2835-ARM-Peripherals.pdf

[8] BCM2835 library functions [Online]. Available: http://www.airspayce.com/mikem/bcm2835/bcm2835_8h_source.html