

K10. Embedded Real Time Ultrasound System

Mostafa A. El-Tager^{1,2,3}, Ehab A. El-Alamy^{1,2,4}, Amir S. Mahdy¹, Islam Youssef¹, Medhat N. El-Dien¹,
and Yasser M. Kadah^{1,2}

¹Biomedical Engineering Department, Cairo University, Giza, Egypt

²Computer and Electronic Systems Engineering Department, Cairo Higher Institute of Engineering, Cairo, Egypt

³m.eltager@chi.edu.eg, ⁴e.alamy@chi.edu.eg,

ABSTRACT

This paper investigates design, modelling, simulation, and control issues related to Doppler ultrasound involving an embedded system. The aim is to view the Doppler spectrogram commonly used in clinical ultrasound and compute the heart rate and blood velocity in the scanned part. Also, it is possible to detect the presence of turbulent or laminar flow. We introduce the embedded ultrasound imaging device based on the Beagleboard and allow the computations to be distributed within the integrated DM3730 processor between its ARM and DSP processors. The resulting system is low in cost, small in size and provide an interface that is comparable to PC-based systems indicating potential for clinical utility.

Keywords: *embedded systems, Doppler ultrasound, digital signal processing.*

I. INTRODUCTION

Medical ultrasound maging (Also known as sonography) is one of the most widely used imaging techniques presently available in medicine. Besides its versatility, performing ultrasound poses no known risks to the patient and hence can be used without reservations in all applications including during pregnancy when other techniques cannot be used. Ultrasound Imaging involves a number of modes that allow producing a variety of diagnostic information in the form of images or signals for the sonographer to make the correct diagnosis. Such modes include A-Mode and B-Mode and Doppler mode for example. All these modes have the advantage of being real-time imaging methods, which makes them very useful in many applications [1]. On the other hand, such methods bear a heavy load of data processing that must be done at high speed to maintain the real-time display of information. Usually, this is done using PC-based systems with even more processing added through special processing cards. Such systems are generally bulky and expensive. In this work, we attempt to take advantage of the advancements made in the area of embedded digital signal processing systems to allow such ultrasound systems to be implemented in a much more efficient manner. Our previous work has shown the design of an analog front-end (AFE) of the system, and an overview of the Digital back-end [2]. This paper focuses on the processing algorithm and the implementation strategy used in our new system.

II. METHODOLOGY

At first we started to apply a processing algorithm over the data received form the AFE, Our algorithm processes the Doppler signal (which is generally in the audio range) taken from Doppler ultrasound system. The data were processed using a periodogram window of size $N=256$. The fast Fourier transformation was applied follow by a shift by $N/2$ (commonly known as fftshift) and the square of the magnitude of the result is obtained. A look-up table is used to scale the result to reduce the dynamic range of the data to match that of the display used (a form of logarithmic compression to compress the dynamic range). Then, the obtained line is added to the spectrogram image which is refreshed on the screen in an independent manner from the processing part. The sweep rate of the spectrogram display is synchronized with the selection of the next data window to be processed to ensure correct time scaling of the spectrogram. A summary of the algorithm used is shown in Fig. 1.

Also after processing the collection of sufficient Doppler spectrogram data, more information can be estimated such as the heart rate. The main idea is to follow the peak velocity of the spectrogram and detect consecutive peaks to measure their time separation. The inverse of that time should be a good instantaneous estimate of the heart rate. Alternatively, detecting such peaks and counting them in a predetermined period of time can be a better estimate given the implicit averaging of peak to peak periods involved. A block diagram of the algorithm based on

the latter strategy that was implemented in the proposed system is shown in Fig. 2. Also, by computing a measure of data dispersion in each column (such as the standard deviation or the entropy), it is possible to detect the presence/absence or turbulence in the flow. Such parameters are of importance for diagnosis of several conditions in the blood vessel and the circulatory system in general.

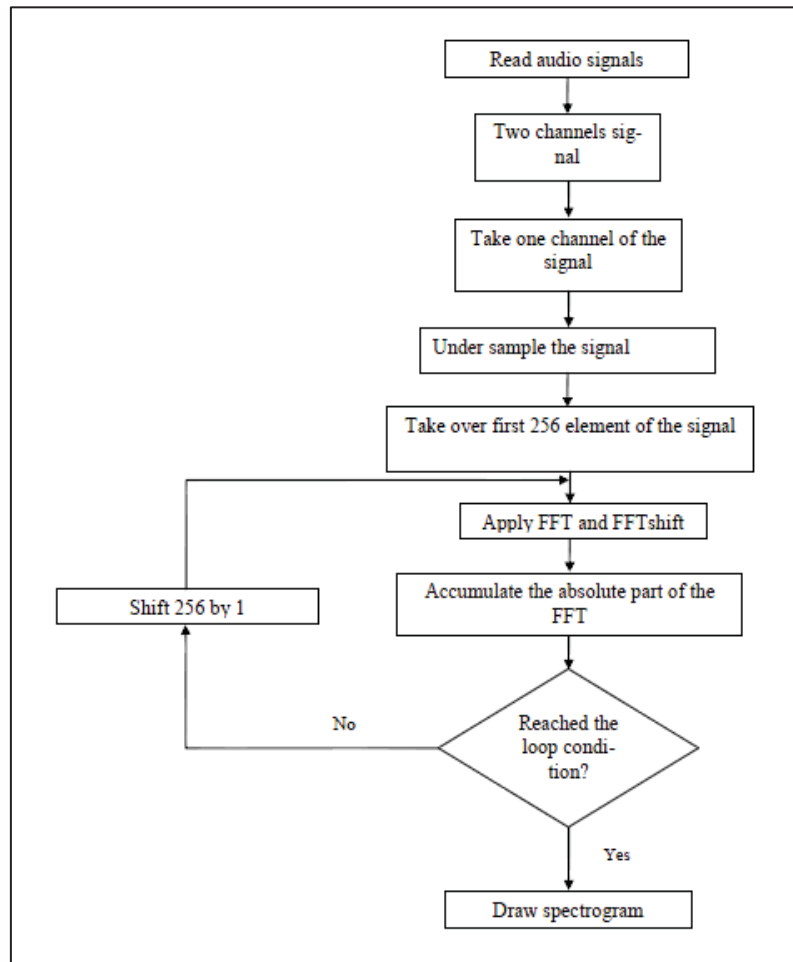


Fig. 1: Summary of the processing steps used in the proposed system

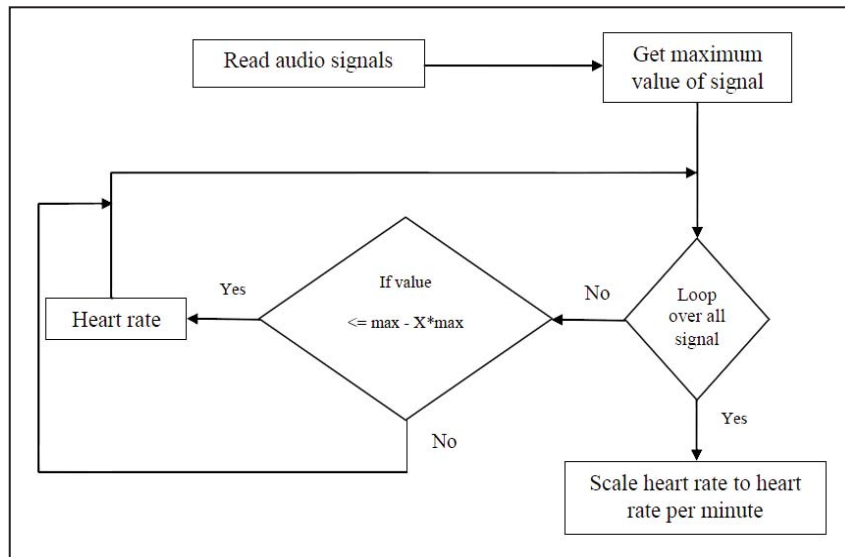


Figure 2: Algorithm used to calculate heart rate

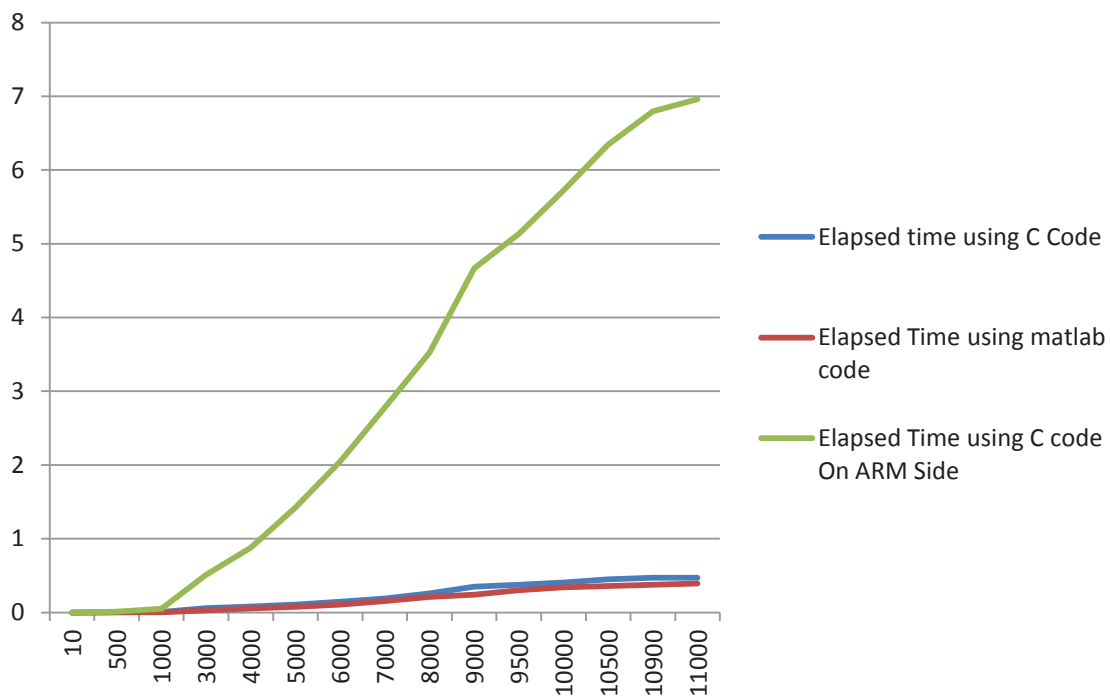


Fig. 3. Test result of complex number multiplication on PC vs. ARM based systems. Horizontal axis shows data size and vertical axis shows the time consumed to perform the computations in seconds.

As a preliminary investigation of the potential of the new system to perform the computational requirements, we compared the performance of a single core ARM processor versus a commonly used modern processor; namely, Intel core-i5®. We chose a simple task of multiplying large arrays of data since very much all steps in the methodology involve such multiplication (for example, windowing of periodogram window, discrete Fourier transformation, etc.). The task involved measuring the time elapsed due to the operation of multiplying two Arrays of different lengths with each other. This measurements are performed on a PC machine with an Intel Core-i5 ® running at 2.27 GHz clock and 4 GB of RAM and this was compared to processing on an ARM Cortex A8 running at a clock speed of 1 GHz and a memory of 512 MB. Fig. 3 shows the results comparing the computations on the PC under Matlab (Mathworks, MA) and C programming against that performed on the ARM processor using

C programming for different sizes of the arrays. The results indicate that the ARM processor alone is not powerful enough for the required real-time performance. Hence, a different platform had to be chosen.

The new implementation strategy selected was to work using a multi-core processor based on the Texas Instrument's OMAP/DaVince processors that offer a dual processor inside one chip. In particular, it includes a 1 GHz ARM cortex A8 microcontroller unit working as a general purpose processor (GPP) in addition to a dedicated digital signal processor (DSP) based on a fixed-point TMS320C64 family processor running at a clock speed of 800MHz. As a result, the system now incorporates two different processors that work together to perform all the required computational and display tasks in the Doppler system whereby the ARM core is used for the graphical user interface (GUI), control of data acquisition, and communication with the DSP core while the DSP core is dedicated to performing the computational tasks. The performance of the DSP was found to surpass that of the results shown in Fig. 3 for all processors and hence it was determined that this platform can indeed be suitable for the real-time requirements of this work.

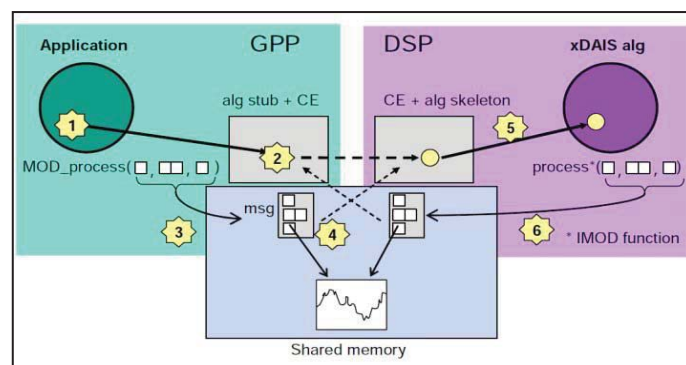


Fig. 4. Inter-Data communication between processors

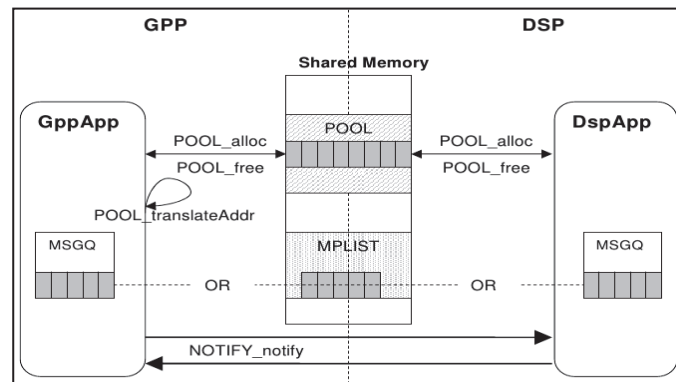


Fig. 5 :MSGQ schematic

In order to deal with the DSP core from the GPP side, first we started to load the GPP with an operating system based on Embedded Linux (Angstrom 2010 distribution) to be able to handle the requirements of the user interface and control tasks. Then, the DSP side is loaded with a special operating system called "DSP BIOS", which is an operating system provided by Texas Instruments to run on their DSP processors [3]. The DSP BIOS has API libraries that are fully compatible with C and Linux programming. The Angstrom distribution allows special functions provided by Texas Instruments to communicate between the GPP and the DSP cores as well. A diagram of the intercommunication between the processors is shown in Fig. 4. The communication between the GPP and DSP processors is possible through several different protocols. The most commonly used such protocol is the Message Queue (MSGQ) protocol shown in Fig. 5 where the two processors communicate through standard messages to a shared memory accessible to both processors.

III. RESULTS AND DISCUSSION

The proposed system was implemented using a development system based on a Beagleboard-xm [4] equipped with a DaVince Digital Media processor [5] (shown in Fig. 6). to be able to deal with both GPP and DSP processors to communicate with each other using MSGQ. Then, we implement our algorithm in order to be classified into two parts:

1. GPP part: where we implement our graphical user interface in order to be easily deal with user input and display results in real-time.
2. DSP part: where the calculations on the data provided by the GPP are done and communicated back to the GPP to display.

The system was interfaced to the analog front-end that acquires the continuous time Doppler signal (which is in the audio range) and the graphical user interface was displayed on a general purpose computer monitor connected to the HDMI output of the Beagleboard. A screenshot of the graphical user interface is shown in Fig. 7.

The cost of the electronic part of the new system is estimated to be below \$250 including the Beagleboard (\$150) and the analog front-end and a small LCD monitor (less than \$100). Given that the cost of a basic PC is significantly higher than that (at least double), the new design offers a much more efficient implementation while being much smaller in size (which will also translate to much lower mechanical package cost).

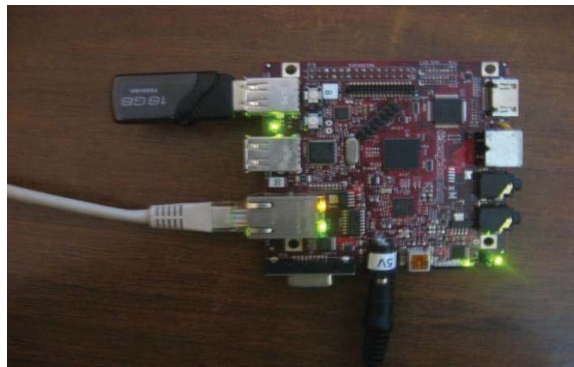


Figure 6: Digital processing part of the system based on the Beagleboard



Figure 7: GUI working on spectrogram mode

IV. CONCLUSIONS

An embedded small Doppler ultrasound device has been developed using a multicore processing platform. Applying the concept of multiprocessor parallel computing over a heterogenic multi-core processors allows more variability on processing the data and a faster performance. A general purpose processor (GPP) is used to handle the standard tasks of user interface, timing and control of data acquisition while a dedicated digital signal processor (DSP) handles computational requirements exclusively and a method of communication in between the two



processors allows data to be shared between the two processors. The performance of the system is comparable to that of present PC-based system with a much lower cost and a much smaller size. This suggests the potential of such embedded systems to be used as the new building blocks for new ultrasound imaging systems.

REFERENCES

- [1] P. Hoskins, K. Martin, A. Thrush, *Diagnostic Ultrasound Physics and Equipment*, Cambridge University Press, Cambridge, 2010.
- [2] A. Mahdy, E Alalamy , I. Youssef, M. Nor Eldien, and M. ElTager, *Embedded Real Time Ultrasound system*, Graduation Project Report, Cairo University, July 2011.
- [3] <http://www.ti.com/lit/ug/spru007i/spru007i.pdf> (DSP/BIOS 5.30 Textual Configuration (Tconf) User's Guide Literature Number: SPRU007H May 2006 , Texas Instruments).
- [4] <http://www.beagleboard.org> (BeagleBoard-xM Rev A2 System reference manual)
- [5] <http://www.ti.com/lit/ds/symlink/dm3725.pdf> (DM3730 Digital Media processor Datasheet, SPRS685D – August 2010 – revised April 2011, Texas Instruments).