FPGA-Oriented Real-Time EMD-Based Breath Signal Processing System on ARM11 MPCore Platform

Tsung-Che Lu, Chien-Liang Lin, Pei-Yu Chen, and Lan-Da Van
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.
Email: tclu@cs.nctu.edu.tw; besthacker99@gmail.com; pychenster@gmail.com; ldvan@cs.nctu.edu.tw

***Abstract*—In this paper, a complete FPGA-oriented real-time empirical mode decomposition (EMD)-based breath signal processing system on ARM11 MPCore platform is developed. The system consists of three subsystems including biomedical signal acquisition, FPGA-oriented EMD accelerator and EMD processing result display. In this work, the previous published multiple stopping criteria and high IMFs Application Specific Integrated Circuit (ASIC)-oriented EMD accelerator is deployed to FPGA of ARM 11 MPCore platform such that the computation time of the overall system can be reduced. An external SRAM architecture in ARM 11 MPCore platform is adopted to process a large amount of data. The three subsystems are connected by a high speed internet such that real time data transportation is achieved. From the evaluation results, for 10K data size, the FPGA-oriented EMD accelerator can speed up by 29.21 times, 20.20 times, and 15.47 times compared with ARM11 processor running single core, dual cores, and four cores, respectively.**

I. INTRODUCTION

Hilbert-Huang Transform (HHT) originally developed and devised by Huang *et al.* [1-5] is the analytic method for the non-linear and non-stationary signal analysis. HHT can be applied to many research fields and application domains. It is especially attractive that HHT is useful in the biomedical signal processing [2] for the breathing pressure analysis and EKG/EEG information analysis. Through HHT processing, different information compared to Fast Fourier Transform (FFT) or Discrete Wavelet Transform (DWT) can be observed such that HHT becomes a favorable tool.

HHT is composed of two parts: Empirical Mode Decomposition (EMD) and Hilbert Transform (HT). EMD is used to decompose the original signals into several individual signal frequency ranges named as Intrinsic Mode Function (IMF). Since EMD has large computation, many researches [6-12] have focused on ASIC and FPGA implementation to speed up the computation. However, the current EMD systems have following limitations. 1) Not a complete real-time system to support multiple stopping criteria and high IMFs to communicate with the real application. 2) Our previously published ASIC-oriented EMD [11] only shows the EMD hardware performance but not overall system application performance. In order to solve the above limitations, in this paper, we use ARM11 MPCore to build and activate one real breath signal processing system [12] using multiple stopping criteria and high IMFs EMD accelerator [11] in FPGA. ARM11 MPCore [13] is one of the most flexible design platforms since ARM11 MPCore has powerful peripheral interface, ARM SOC chipset and Xillinx FPGA daughter board, where the platform can provide the data transition and FPGA system status monitor. NI USB-6009 data acquisition device [14] is used to capture the real breath signal. The captured signals are send to remote ARM Versatile platform for EMD processing on FPGA, and then the platform transmits the result to a remote monitor for IMFs display by LabVIEW [15]. This real time breath signal processing system integrates breath signal instrument/device for signal acquisition, FPGA-oriented embedded board for EMD processing and virtual instrument for IMFs display. The rest is organized as follows. The EMD algorithm is briefly reviewed in Section II. Section III describes the whole developed EMD-based breath signal processing system infrastructure. In Section IV, the measurement and comparison results for breath signal processing are revealed. Last, the conclusion is remarked.

II. BRIEF EMD ALGORITHM REVIEW

The EMD proposed by Huang *et al.* [1] targets at analyzing and exploring the non-linear and non-stationary signals in the HHT. Through the EMD algorithm, the raw data $x(t)$ can be decomposed into the IMF components and the residue component. In order to make readers to catch up the EMD algorithm, herein, we adopt the same notations and repeat the same key content as [11, 1-4]. The steps of the EMD algorithm are summarized as follows.

Step 1: Initialize $input(t) = x(t)$.
Step 2: Generate the upper envelope and the lower envelope via searching local extrema of $input(t)$ and using the cubic spline method.
Step 3: Compute the mean value $m(t)$ of the upper envelope and lower envelope.
Step 4: Generate the IMF candidate $h_{i,k}(t)$ via $h_{i,k}(t) = h_{i,(k\text{-}1)}(t) - m(t)$, where $h_{1,1}(t) = x(t)$.
Step 5: Set $input(t) = h_{i,k}(t)$ if the stopping criterion is not satisfied and repeat Step 2~Step 5. Otherwise, go to Step 6.
Step 6: Set IMF $c_i(t) = h_{i,k}(t)$ and generate the residue $r_i(t)$ via $r_i(t) = r_{i-1}(t) - c_i(t)$, where $r_0(t) = x(t)$.
Step 7: Set $input(t) = r_i(t)$ and repeat Step 2~Step 7 to find the next IMF component.

The corresponding detailed hardware and ASIC implementation of the EMD accelerator is addressed in [11].

III. COMPLETE FPGA-ORIENTED REAL-TIME BREATH SIGNAL PROCESSING SYSTEM

In this section, we develop an EMD-based breath signal processing system as shown in Fig. 1 for breath signal application demonstration. The full system contains breath signal acquisition subsystem, ARM Versatile subsystem [13] with the hardware EMD accelerator on FPGA, and IMFs data display subsystem. The BIOPAC SS5LB transducer [16] is used to record and acquire the breath signal data. The breath signal data acquisition is transmitted from BIOPAC SS5LB to multifunction data acquisition (DAQ) instrument, NI USB-6009 [14], which converts the analog breath signal into digital domain at 48 KS/s sample rate in DAQmx driver and LabVIEW [15]. Breath signal acquisition is executed with user defined sample rate, parameter, and voltage on LabVIEW [15]. After receiving the sequential data from the DAQmx driver on LabVIEW, the raw data is packed according to the internal predefined format, and then sent to the remote ARM Versatile platform through the TCP/IP protocol for EMD processing. ARM Versatile platform contains CPU, FPGA, two SRAMs, and other interfaces components [13]. The CPU of ARM Versatile platform plays an important role on the major coordinator among aforementioned components. The CPU of ARM Versatile platform receives packed raw breathing data from Ethernet port through the TCP/IP protocol and stores raw breathing data into one of the two SRAM blocks. After that, the EMD accelerator on FPGA will be triggered to execute EMD process with the stored raw data and the output IMFs will be saved into the other SRAM block. After processing, the raw breathing data, IMFs and residue will be repacked and sent to remote display subsystem through the TCP/IP protocol. Then, the remote IMFs display subsystem will demonstrate information on LabVIEW.

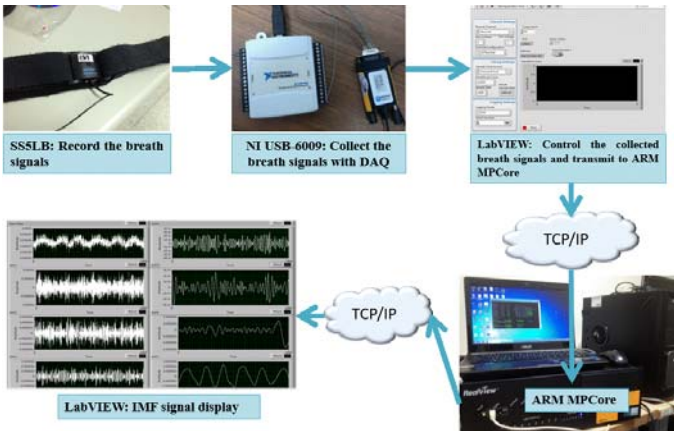
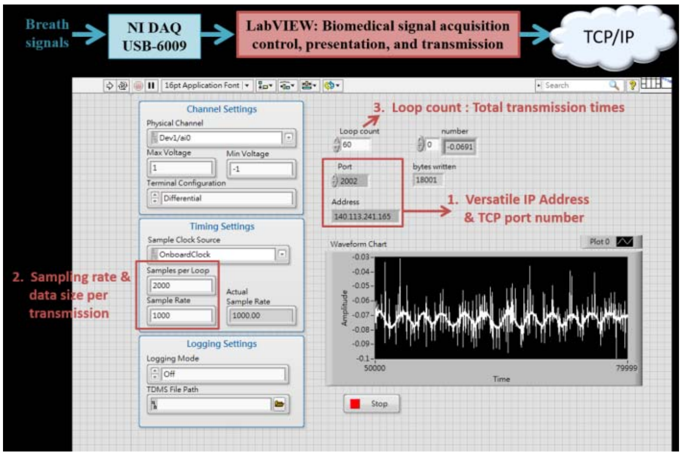


Fig. 1. The developed FPGA-oriented EMD-based breath signal processing system on ARM11 MPCore.

A) Biomedical Signal Acquisition Subsystem

From the information in BIOPAC website [16], the SS5LB [16] can not only measure slow signal without losing significant information but also have good linear characteristic. The SS5LB soft cable has a length of 2 meters with light-weighted feature [16]. In the biomedical signal acquisition subsystem, the 9-PIN female D-SUB connector is used to linked with BIOPAC SS5LB [16]. From the specification of NI USB-6009 in [14], the DAQ provides eight analog input (AI) channel connectors up to 48 KS/s sampling rate as well as two analog output (AO) channels, and 12 digital input/output (DIO) channels. In this work, the D-SUB male connector links PIN 2 to AI0+, PIN 4 to AI0-, PIN3 to AI GND, and PIN6 to +5v which are connected with the wires as shown in Fig. 2. Therefore, the NI USB-6009 can be connected with PC through full-speed USB interface. LabVIEW is installed on PC to communicate with NI USB-6009 through USB interface. When LabVIEW is running, it needs DAQmx driver to drive NI USB-6009 for data acquisition. In this paper, the data acquisition subsystem supports parameter configuration such as dynamic sampling rate range from 200~2K S/s, packed data size per transmission, loop count of transmission times, destination IP address and TCP port of Versatile Platform as shown in Fig. 3. Raw breathing data received from NI USB-6009 will be translated from single-floating-point data format to 30-bit fixed-point data format and packed with user defined data size per transmission. After collecting data at PC side, LabVIEW will send those data on the TCP session defined by user to the remote Versatile Platform. The operation will keep working until the loop count of transmission times is reached.

Fig. 2. Connection between SSL5LB and NI USB-6009.



Channel Settings
Physical Channel: Dev1/ai0
Max Voltage: 1
Min Voltage: -1
Terminal Configuration: Differential

Timing Settings
Sample Clock Source: OnboardClock
Samples per Loop: 2000
Sample Rate: 1000
Actual Sample Rate: 1000.00

Logging Settings
Logging Mode: Off
TDMS File Path:

Loop count: 60
number: 0 -0.0691
Port: 2002
bytes written: 18001
Address: 140.113.241.165

Fig. 3. Biomedical signal acquisition system.

B) ARM Versatile Subsystem with the FPGA-Oriented EMD Accelerator

The development platform is ARM11 MPCore platform [13] for embedded applications. Based on ARM SMP architecture, tightly hardware and software integration is implemented in ARM11 MPCore platform [13]. ARM

Versatile has independent ARM11 PB11MPCORE inside and acts as a high speed SMP-processor for software development platform [13]. With this architecture, concurrent execution with up to four CPU cores and high-speed memory bus system can be supported to fully utilize the CPU computing resources. To fully utilize the Versatile system, the fully software stack [17] including u-boot image, Linux kernel image and JFFS2 file system image is required to built up [17]. The boot monitor is used to write the images into the flash partition. That means u-boot image, Linux kernel image, and JFFS2 file system can be stored in NOR flash. The execution flow on the Versatile platform is depicted in Fig. 4. The EMD proxy application for the Linux operating system is implemented with the ARM processor. In the Versatile platform, following operations are executed by the EMD proxy: 1) create TCP/IP socket for packed raw data receiving from remote data acquisition LabVIEW subsystem, 2) send raw data to SRAM A, 3) trigger FPGA for EMD processing, 4) monitor EMD computation processing status, 5) receive the completed EMD result, 6) read IMFs from SRAM B, 7) transmit IMFs from SRAM B to the remote LabVIEW display. Due to the software modularity, in a standard Linux operating system data flow, user space and kernel space will need to allocate 4 buffer memories and perform 4 memory copies, accordingly, to finish all data transfers from TCP/IP to SRAM. In order to reduce the redundant buffer memory allocation and memory copy times, since the data will be needed to write to SRAM in our infrastructure, the Linux memory map mechanism [18] is used to map the SRAM memory space into the EMD proxy application memory space. The raw data received by the EMD proxy will be stored into SRAM memory without additional buffer memory of user space and kernel space. That means the number of used buffer memories will be reduced from 4 to 1 and the memory copies will be reduced from 4 to 1. Thus, CPU will not be busy on redundant memory copy operation.

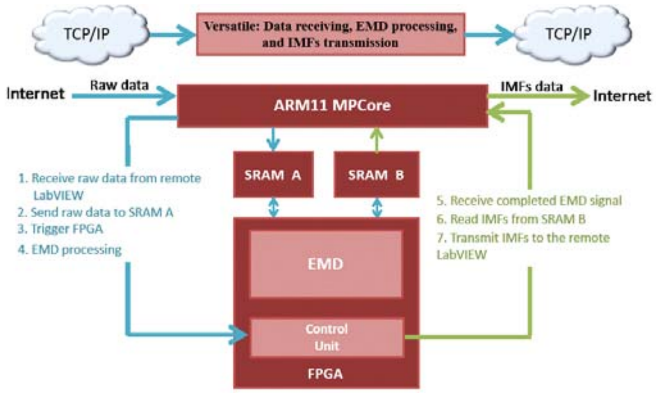


Fig. 4. Execution flow on the Versatile platform.

C) IMF Data Display Subsystem

In Fig. 5, the display subsystem is build up on LabVIEW. The TCP/IP protocol is provided to connect to the remote Versatile subsystem. Versatile will send those IMFs to display subsystem after the EMD process is completed. The receiving data will be depicted as wave signal on data amplitude panel which is set by LabVIEW subsystem. The panel can dynamically scale the receiving data and continuously show the real-time wave signal. Therefore, the real-time IMF can be observed immediately without waiting post process. Besides, the display subsystem and biomedical signal acquisition subsystem can execute in the same PC for easy demonstration.

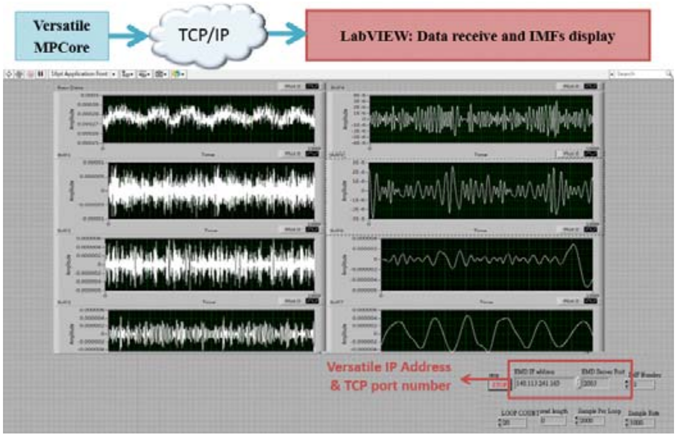
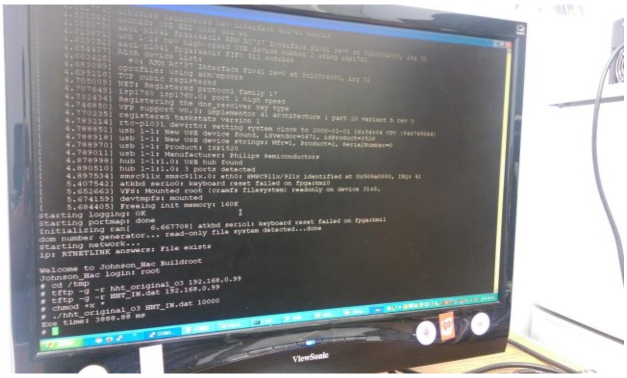


Fig. 5. LabVIEW display subsystem.

IV. SYSTEM PERFORMANCE EVALUATION AND COMPARISON RESULTS

In this section, we evaluate the proposed EMD hardware accelerator on FPGA and the software performance with constant stopping criteria. The breathing signals with 10K and 60K data size are fed into the FPGA-oriented EMD architecture and ARM11 MPCore to observe the execution time performance in Table 1. ARM11 MPCore CPU supports up to 4 cores running, where the multithread technology and OpenMP are utilized for the 4 cores on ARM11 MPCore. It is noted that the proposed FPGA-oriented EMD accelerator can speed up the execution time by 29.21 times, 20.20 times, and 15.47 times compared with software on single core, dual cores, and four cores, respectively, with 10K data size and 10 IMFs. The captured photos are shown in Fig. 6 and Fig. 7 using single core and FPGA, respectively, with 10K data size. In Table 2, the comparison among possible existing FPGA implementations is expressed. From Table 2, the presented system is a complete and real-time application system with high IMFs support and multiple stopping criteria. Compared with [11], the developed system can process larger size breath signals within the reasonable execution time.



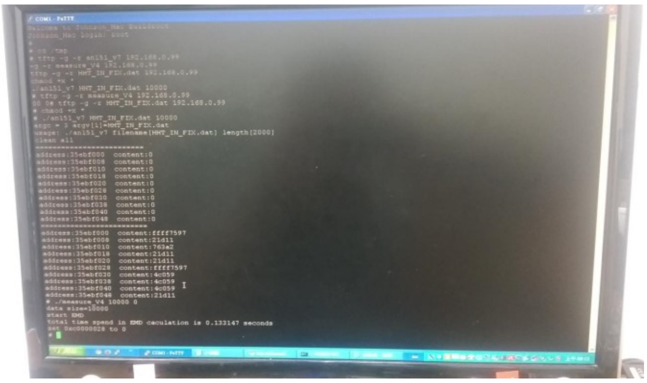
```
[    4.683089] oprofile: using arm/mpcore
[    4.695216] TCP cubic registered
[    4.705009] NET: Registered protocol family 17
[    4.707649] isp1760 isp1760.0: port 1 high speed
[    4.732434] Registering the dns_resolver key type
[    4.746850] VFP support v0.3: implementor 41 architecture 1 part 20 variant b rev 3
[    4.770235] registered taskstats version 1
[    4.783214] rtc-pl031 dev:rtc: setting system clock to 2000-01-01 16:54:04 UTC (946745644)
[    4.788851] usb 1-1: New USB device found, idVendor=0471, idProduct=3526
[    4.788918] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[    4.788970] usb 1-1: Product: ISP1520
[    4.789011] usb 1-1: Manufacturer: Philips Semiconductors
[    4.878998] hub 1-1:1.0: USB hub found
[    4.890510] hub 1-1:1.0: 3 ports detected
[    4.897534] smsc911x smsc911x.0: eth0: SMSC911x/921x identified at 0x508a0000, IRQ: 41
[    5.407542] atkbd serio0: keyboard reset failed on fpga:kmi0
[    5.652663] VFS: Mounted root (cramfs filesystem) readonly on device 31:0.
[    5.674159] devtmpfs: mounted
[    5.684405] Freeing init memory: 160K
Starting logging: OK
Starting portmap: done
Initializing ran[    6.667708] atkbd serio1: keyboard reset failed on fpga:kmi1
dom number generator... read-only file system detected...done
Starting network...
ip: RTNETLINK answers: File exists

Welcome to Johnson_Hac Buildroot
Johnson_Hac login: root
# cd /tmp
# tftp -g -r hht_original_o3 192.168.0.99
# tftp -g -r HHT_IN.dat 192.168.0.99
# chmod +x *
# ./hht_original_o3 HHT_IN.dat 10000
Exe time: 3888.88 ms
# 
```

Fig. 6. EMD execution time using one core with 10K data size.

Table 1: Performance Comparison between FPGA and ARM11

Data Size	Computation Time (Sec)			
	1 ARM Core	2 ARM Cores	4 ARM Cores	FPGA
10K	3.888888	2.68958	2.05945	0.133147
60K	33.6411	23.478	18.2358	0.848895



```
Welcome to Johnson_Mac Buildroot
Johnson_Mac login: root
#
# cd /tmp
# tftp -g -r an151_v7 192.168.0.99
-g -r measure_V4 192.168.0.99
tftp -g -r HHT_IN_FIX.dat 192.168.0.99
chmod +x *
./an151_v7 HHT_IN_FIX.dat 10000
# tftp -g -r measure_V4 192.168.0.99
00 0# tftp -g -r HHT_IN_FIX.dat 192.168.0.99
# chmod +x *
# ./an151_v7 HHT_IN_FIX.dat 10000
argc = 3 argv[1]=HHT_IN_FIX.dat
usage: ./an151_v7 filename[HHT_IN_FIX.dat] length[2000]
clean all
========================
address:35ebf000  content:0
address:35ebf008  content:0
address:35ebf010  content:0
address:35ebf018  content:0
address:35ebf020  content:0
address:35ebf028  content:0
address:35ebf030  content:0
address:35ebf038  content:0
address:35ebf040  content:0
address:35ebf048  content:0
========================
address:35ebf000  content:ffff7597
address:35ebf008  content:21d11
address:35ebf010  content:763a2
address:35ebf018  content:21d11
address:35ebf020  content:21d11
address:35ebf028  content:ffff7597
address:35ebf030  content:4c059
address:35ebf038  content:4c059
address:35ebf040  content:4c059
address:35ebf048  content:21d11
# ./measure_V4 10000 0
data size=10000
start EMD
total time spend in EMD caculation is 0.133147 seconds
set 0xc0000028 to 0
#
```

Fig. 7. EMD execution time using FPGA with 10K data size.

V. CONCLUSION

In this paper, an FPGA-oriented real-time EMD-based breath signal processing system including data acquisition, remote EMD processing, and remote IMFs display is developed and tested. From the evaluation results, for 10K breath signal data size, the FPGA-oriented EMD accelerator is able to speed up by 29.21 times, 20.20 times, and 15.47 times compared with ARM11 processor running 1 core, 2 cores, and 4 cores, respectively. From the evaluation and comparison results, the implementation demonstrates that the FPGA-oriented EMD-based breath signal processing system can real time decompose the component. The prototype can be expected to provide a cloud service for health-care environment.

ACKNOWLEDGEMENT

The authors would like to thanks Prof. T. C. Hsiao to borrow the related equipment and device. This work is supported in part by the MOST 106-2221-E-009-028-MY3 and MOST 106-2218-E-009-029.

REFERENCES

[1] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N. C. Yen, C. C. Tung, and H. H. Liu, “The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis,” *Proceedings of The Royal Society of London Series A*, vol. 454, pp. 903-995, Mar. 1998.

[2] M. C. Wu and N. E. Huang, “Biomedical data processing using HHT: A review,” *Advanced Biosignal Processing*, Springer-Verlag Berlin Heidelberg, pp. 335-352, 2009.

[3] G. Wang, X. Y. Chen, F. L. Qiao, Z. H. Wu, and N. E. Huang, “On Intrinsic Mode Function,” *Advances in Adaptive Data Analysis*, vol. 2, no. 3, pp. 277-293, 2010.

[4] N. E. Huang and S. S. P. Shen, *The Hilbert-Huang Transform and Its Applications*, World Scientific, 2014.

[5] “The EMD program for MATLAB, Research Center of Adaptive Data Analysis, NCU,” http://rcada.ncu.edu.tw/research1_clip_program.htm

[6] L. Wang, M. I. Vai, P. U. Mak, and C. I. Ieong, “Hardware-accelerated implementation of EMD,” in *Proc. IEEE BMEI*, 2010, pp. 912–915.

[7] M. H. Lee, K. K. Shyu, P. L. Lee, C. M. Huang, and Y. J. Chiu, “Hardware implementation of EMD using DSP and FPGA for online signal processing,” *IEEE Trans. Ind. Electronics*, vol. 58, no. 6, pp. 2473-2481, Jun. 2011.

[8] S. C. Chen, C. C. Chen, W. C. Guo, T. J. Lin, and C. W. Yeh, “Complexity-effective Hilbert-Huang transform (HHT) IP for embedded real-time applications,” in *Proc. Asia and South Pacific (ASP)-DAC*, Feb. 2012.

[9] W. C. Fang, C. C. Chou, T. H. Hung, K. C. Lin, A. H. T. Li, Y. C. Chang, B. K. Hwang, and Y. W. Shau, “An efficient and accurate empirical mode decomposition of the technical design and methods for biological sound,” in *Proc. IEEE Biomedical Circuits Syst. (BioCAS)*, 2012, pp. 320–323.

[10] Y. Y. Hong and Y. Q. Bao, “FPGA implementation for real-time empirical mode decomposition,” *IEEE Trans. Instrum. Meas.*, vol. 61, no.12, pp. 3175–3184, Dec. 2012.

[11] T. C. Lu, P. Y. Chen, S. W. Yeh, and L. D. Van, “Multiple stopping criteria and high-precision EMD architecture implementation for Hilbert-Huang transform,” in *Proc. IEEE Biomedical Circuits Syst. (BioCAS)*, Oct. 2014, pp. 200-203, Lausanne, Switzerland.

[12] Chien-Liang Lin, *Design and Implementation of EMD-based Breath Signal Processing System*, Master’s thesis, National Chiao Tung University, CS Dept., Taiwan, 2014. (Advisor: Lan-Da Van)

[13] “RealView Platform Baseboard for ARM11 MPCore User Guide,” http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.dui0351e/I1007079.html

[14] http://www.ni.com/pdf/manuals/371303n.pdf

[15] http://www.ni.com/labview/zht/

[16] http://www.biopac.com/respiratory-effort-transducer-bsl

[17] https://buildroot.org/

[18] D. Stancevic, “Zero copy I: User-mode perspective,” *Linux Journal*, Jan. 2003. (https://www.linuxjournal.com/article/6345)

Table 2: Comparison Results among the FPGA-Oriented EMD Implementations

Reference	BMEI’10 [6]	TIE’11 [7]	ASP-DAC’12 [8]	BioCAS’12 [9]	TIM’12 [10]	BioCAS’14 [11]	This work [12]
Application	EKG	EKG	EKG/breathing signal	Biological sound	Ultrasonic signal	Breathing signal	Breathing signal
Complete Application Demo Picture	N/A	Yes	Yes	N/A	N/A	No	Yes
Sampling Rate	N/A	360 Hz	250 Hz/100 Hz	11.025 KHz	12.5 MHz	1000 Hz	1000 Hz
Operation Frequency	100 MHz	50 MHz	N/A	62.5 MHz	12.5 MHz	40 MHz	N/A
Latency/Computation Time	63.833 ms (512 data)	2780 ms (1K data)	N/A	N/A	0.1 ms (1K data)	11.494 ms (2K data)	133.147 ms (10K data)
IMF components	N/A	N/A	N/A	8	2	10	10
Support of Multiple Stopping Criteria	No	No	No	No	No	Yes (3 Stopping Criteria)	Yes (3 Stopping Criteria)
Implementation Approach	FPGA	DSP+FPGA	ARM+FPGA	FPGA	FPGA	ASIC	FPGA
FPGA Device	Altera Nios II	TI C6713+ Altera Cyclone II 2C20	Xilinx Virtex-5 XC5VLX110	Xilinx VEX330	Altera Stratix III EP3SL150F1152C2	N/A	ARM11 MPCore