

# A Low Power Biomedical Signal Processor ASIC Based on Hardware Software Codesign

Z. D. Nie, L. Wang, *Member, IEEE*, W. G. Chen, T. Zhang, and Y. T. Zhang, *Fellow, IEEE*

**Abstract**— A low power biomedical digital signal processor ASIC based on hardware and software codesign methodology was presented in this paper. The codesign methodology was used to achieve higher system performance and design flexibility. The hardware implementation included a low power 32bit RISC CPU ARM7TDMI, a low power AHB-compatible bus, and a scalable digital co-processor that was optimized for low power Fast Fourier Transform (FFT) calculations. The co-processor could be scaled for 8-point, 16-point and 32-point FFTs, taking approximate 50, 100 and 150 clock circles, respectively. The complete design was intensively simulated using ARM DSM model and was emulated by ARM Versatile platform, before conducted to silicon. The multi-million-gate ASIC was fabricated using SMIC 0.18 $\mu$ m mixed-signal CMOS 1P6M technology. The die area measures 5,000 $\mu$ m x 2,350 $\mu$ m. The power consumption was approximately 3.6 mW at 1.8V power supply and 1MHz clock rate. The power consumption for FFT calculations was less than 1.5 % comparing with the conventional embedded software-based solution.

## I. INTRODUCTION

RECENT years have seen a surge of the development of wireless, low power wearable and implantable devices for physiological measurements and telemedicine applications [1, 2]. Subsequently, various signal processing algorithms were developed to process the measured signals such as ECG, EEG, EMG, EGG, respiration, PPG, and etc [3]. It is envisaged that most of the aforementioned signals are periodic and many vital signs such as heart rate, pulse rate, EEG rhythms and respiration rate have primary features in frequency spectrum. Therefore, spectrum analysis is quite often a fundamental building block being employed and a Fast Fourier Transform (FFT) unit is a first choice to be considered [4-6].

In another side of the spectrum, the wearable or implantable devices must be small and discreet. One of the design challenges is the processing-on-node capability.

Manuscript received April 24, 2009. This work was supported in part by Chinese Academy of Sciences "the 100 Talent People" Program and Key Lab for Biomedical Informatics and Health Engineering, Chinese Academy of Sciences.

Z. D. Nie, W.G. Chen, T.Zhang are with the Institute of Biomedical and Health Engineering (IBHE) Shenzhen Institute of Advanced Technology (SIAT) (e-mail: zd.nie@siat.ac.cn).

L.Wang is with the Institute of Biomedical and Health Engineering (IBHE) Shenzhen Institute of Advanced Technology (SIAT) (phone:86-755-86392277, e-mail: wang.lei@siat.ac.cn).

Y. T. Zhang is with the Institute of Biomedical and Health Engineering (IBHE) Shenzhen Institute of Advanced Technology (SIAT) (e-mail: yt.zhang@siat.ac.cn).

Because of the extreme size and power constraints, it is difficult to perform the spectrum analysis within the wearable or implantable devices. A possible alternative is to transmit all the raw data to a more powerful base-station, such as a PDA, for post-processing [7]. This approach is disadvantageous because a) the RF transmission consumes a large amount of battery power; b) it requires a high bandwidth RF channel.

Different off-the-shelf IC modules were used to tackle the on-node computational bottleneck. A conventional microprocessor is easy to use, but its architecture was not instruction sets. The use of FPGA has also been suggested, but the relatively high power consumption prohibits its use for practical wearable or implantable applications.

An Application-Specific Integrated Circuit (ASIC) can be fully customized, providing maximal design flexibility at the lowest-possible power consumption [8]. In an ASIC, all functional building blocks can be integrated into a single piece of silicon, which means potential size reduction for the sensor nodes. This also simplifies the subsequent packaging and assembly processes. An ASIC is cost-effective when volume production is applied.

In this paper, we represent a mixed-signal ASIC based on hardware and software codesign for scalable FFT calculations. Codesign is a methodology for solving design problems in processor-based embedded systems and allows the concurrent design of both hardware and software [9-10]. The design advantages of the codesign approach for biomedical signal spectrum analysis was also illustrated in this paper.

## II. SYSTEM ARCHITECTURE

The complete ASIC was designed based on the codesign methodology. It was partitioned into hardware portion and embedded software portion. Fig. 1 illustrated the system architecture of the ASIC.

The system has the following primary features:

- 32-bit RISC ARM7TDMI processor.
- Low power AHB compatible bus (LPAHB).
- Scalable FFT module that could be scaled for 8-point, 16-point and 32-point FFT.
- 4K-word SRAM for data and program storage.
- Digital interfaces to various off-the-chip ADC and RF front end modules.
- On chip VCO oscillator to generate clock rates up to 30 MHz, eliminating the needs for off-chip oscillators.

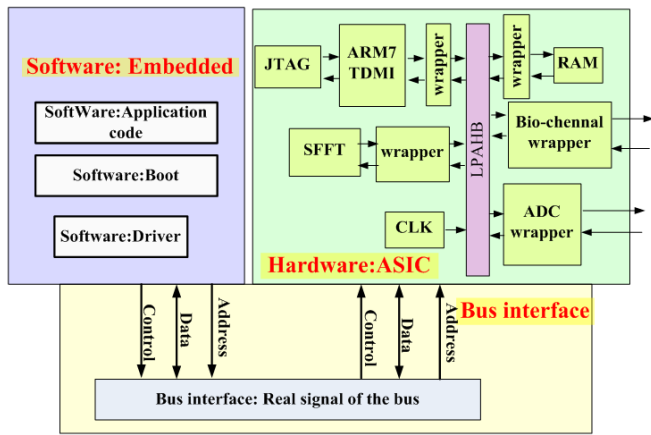


Fig. 1. ASIC system architecture

As for design flow, Verilog was used for RTL level descriptions. The RTL codes were simulated and synthesized using Synopsys tools, and then emulated in using ARM Versatile platform. The backend tools used are Cadence tools. The software part adopted C and assembler languages based on ARM RealView Development Suite (RVDS).

### III. HARDWARE IMPLEMENTATION

#### A. ARM7TDMI implemetation

The ARM7TDMI core is a 32-bit embedded RISC processor delivered as a hard macro cell optimized to provide the best combination of performance, power and area characteristics. The ARM7TDMI core enables system designers to build embedded devices requiring small size, low power and high performance [11].

#### B. Low power AHB compatible system bus design

In this design, an AHB compatible system bus was designed for inter-system connection. For simple purpose, the AHB Lite architecture is used for high efficiency communication.

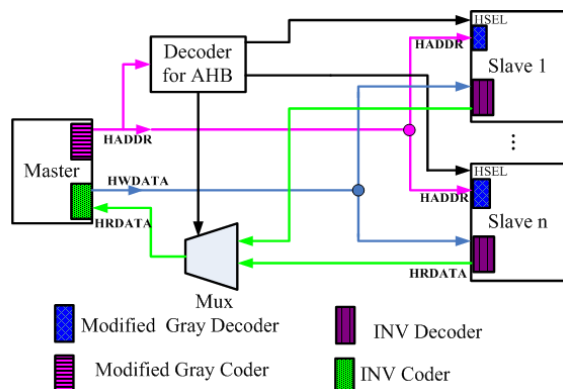


Fig. 2. The Structure of Low Power AHB Compatible Bus

Typically the power dissipated by system-level buses contributes the largest portion of the global power of a complex VLSI system. Therefore, the minimization of the

switching activity at the I/O interfaces provides significant savings on the overall power budget [12]. In our design a bus-invert code (INV) was used to minimize the switching in the data bus, for the address bus, a modified GRAY encoding was adopted to preserve the one-transition property for consecutive addresses of byte-addressable machines [13]. The structure of the low power AHB compatible bus is illustrated in Fig. 2.

#### C. Scalable FFT circuit description

A scalable FFT module that could be scaled for 8-point, 16-point and 32-point FFT was implemented into the digital co-processor. The FFT was designated in a scalable manner.

A 2-point FFT is the basic building unit for scalable design. For 2n-point FFT, the Decimation-In-Time algorithm and Decimation-In-Frequency algorithm were deduced from the Cooley-Turkey algorithm [14]. The Decimation-In-Time FFT Radix-2 was used and carried out by modified butterfly architecture.

The inputs of the scalable FFT were 8-bit complex number:  $X_{p-in}(n)$  and  $X_{q-in}(n)$ . The absolute value of a twiddle factor  $WN_k$  was set to be less than or equal to 1. Before  $X_{q-in}(n)$  multiply with  $WN_k$ ,  $WN_k$  was multiplied by 26. After the multiplying, the result was right\_shift six bits. The result of multiplication of complex number  $X_{q-in}(n) = X_{q-in-R}(n) + i X_{q-in-I}(n)$  and  $WN_k = WN_{k-R} + i WN_{k-I}$ , was transformed as:

$$X_{qn}(n) = [X_{q-I}(n) * (WN_{k-R} - WN_{k-I}) + WN_{k-R} * (X_{q-R}(n) - X_{q-I}(n))] + i [X_{q-R}(n) * (WN_{k-R} + WN_{k-I}) - WN_{k-R} * (X_{q-R}(n) - X_{q-I}(n))] \quad (1)$$

The results of the modified butterfly were:

$$X_{p-out}(n) = [X_{p-in}(n) - R + X_{qn-R}(n)] + i [X_{p-in}(n) - I + X_{qn-I}(n)]$$

$$X_{q-out}(n) = [X_{p-in}(n) - R + X_{qn-R}(n)] - i [X_{p-in}(n) - I + X_{qn-I}(n)] \quad (2)$$

Equations (1) and (2) indicated that one complex number multiplier needs 3 multipliers, 3 subtractors and 2 adders [15]. The basic structure was illustrated in Fig. 3. Two complex numbers were fed into Stage2, the result of Stage2 was extended with sign and then fed into DFF 2A\_0 after the demultiplexer. In the next step, Stage2 received another two complex numbers, the result of Stage2 was again extended with sign and then input to DFF 2A\_1. If the inputs of Stage4 were ready, Stage4 was immediately enabled. Via reusing the basic modules, 4-point FFT was achieved [16].

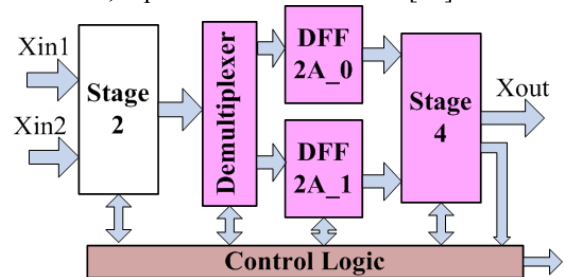


Fig. 3. The Structure of The Basic Module

Fig. 4 illustrated that the by scaling up, the 8-point, 16-point and 32-point FFTs were achieved hierarchically.

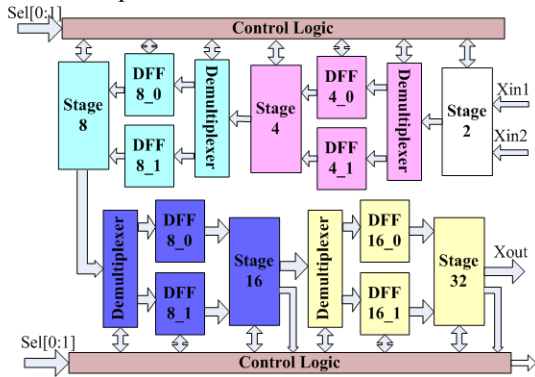


Fig. 4. The Structure of the complete Scalable\_FFT

The signal SEL was used to set different operation models: “00” for idle status, “01” for 8-point FFT, “10” for 16-point FFT, and “11” for 32-point FFT.

#### IV. SOFTWARE IMPLEMENTATION

The codesign methodology was used to bridge the software and hardware design. Fig.5 demonstrates the software flow. The source codes were hybrid of ARM assemble code and C code, which were compiled and linked in the RVDS. The codes were debugged in the Instruction Set Simulator that was hardware independent. The Realview ICE platform downloaded codes into the hardware for emulations.

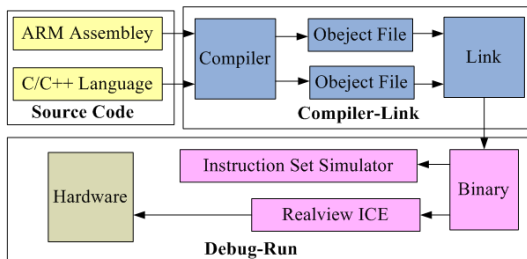


Fig. 5. Software Implementation Flow

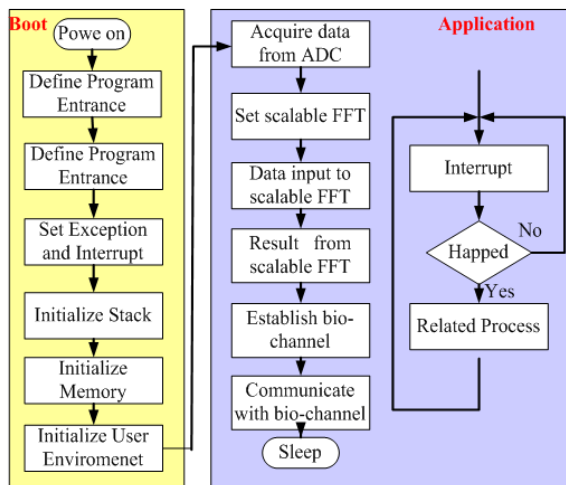


Fig. 6. Source Code Flow

Fig. 6 illustrates the source code design flow. The source

code mainly contained two parts: boot code and application code. In this design, all the modules except ARM7TDMI were slaves, and the boot code was responsible for booting the complete digital system. For the application code, Tables I gives an example of reading and writing the FFT coprocessor.

TABLE I. SIMPLE INSTRUCTIONS FOR WRITE AND REDA FFT

```
MOV r1, #0xc8000000 //FFT control register address to r1
STR r0, [r1,#0xdc] // move data to FFT control register
LDR r2, [r0,#0x44] // read the result from FFT result register
```

#### V. RESULTS AND DISCUSSIONS

The completed system was intensively simulated and emulated using ARM DSM model and the ARM Versatile platform, the latter implements a Logic Tile (Xilinx Virtex-5 FPGA XC5VLX330) to host all the digital hardware. Results indicated that the ARM could be booted successfully with the current setup and all the digital hardware worked as intended.

The digital design was synthesized in Synopsys DC compiler and full-chip layout was conducted in Cadence SoC-Encounter and Virtuoso using SMIC 0.18- $\mu$ m mixed-signal CMOS 1P6M library. Fig.7 gives the layout view of the ASIC. The die area measures 5000 $\mu$ m by 2350 $\mu$ m. Table II illustrates the ASIC specification.

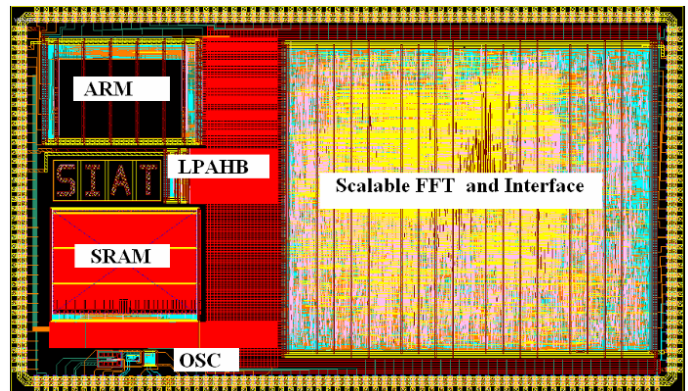


Fig. 7. The Layout View of the ASIC

TABLE II. ASIC SPECIFICATION

	Size (mm <sup>2</sup> )	Power (mW) *
ARM	0.96	0.26
Scalable FFT	6.58	2.95
Interface		0.06
SRAM	0.95	0.21
LPAHB	0.08	0.02
OSC	0.02	0.9
Whole chip	11.75	3.6

\*: the power was estimated at 1.8V power supply and at 1 MHz clock rate.

TABLE III. CLOCK CIRCLES FOR THE EXECUTION OF DIFFERENT FFTS

	FFT_8	FFT_16	FFT_32
Embedded approach	59500	114800	208000
Codesign approach	50	100	150

Table III represents the computational performance comparisons between an embedded (pure software) approach and the codesign approach that was implemented in our design. It indicated that, to complete a 32-point FFT, it took approximate 200K clock circles using a 16-bit ARM-compatible microprocessor, the number of clock circles for our approach is merely 150. As illustrated in Table II the FFT hardware consumed 11 times more power than the ARM core. Therefore, the power consumption per one 32-point FFT using our approach was less than 1.5 % of the power consumption from a pure embedded software solution.

In order to evaluate the accuracies of the FFT computation, the error rates were compared between a floating point solution (using a PC) and the fixed point solution which was adopted by our design. The following formulas (3) – (8) were used:

$$Sout\_mat\_r = \sum_{k=0}^{k=N} |Xout\_mat\_k\_r| \quad (3)$$

$$Sout\_mat\_i = \sum_{k=0}^{k=N} |Xout\_mat\_k\_i| \quad (4)$$

$$Sout\_vcs\_r = \sum_{k=0}^{k=N} |Xout\_vcs\_k\_r| \quad (5)$$

$$Sout\_vcs\_i = \sum_{k=0}^{k=N} |Xout\_vcs\_k\_i| \quad (6)$$

$$Error\_r = \frac{\|Sout\_mat\_r - |Sout\_vcs\_r|\|}{|Sout\_mat\_r|} \quad (7)$$

$$Error\_i = \frac{\|Sout\_mat\_i - |Sout\_vcs\_i|\|}{|Sout\_mat\_i|} \quad (8)$$

$Xout\_mat\_k\_r$  and  $Xout\_mat\_k\_i$  were the real and imaging portions calculated in floating point; and the  $Xout\_vcs\_k\_r$  and  $Xout\_vcs\_k\_i$  were the real and imaging portions calculated in fixed point based on our approach. The results were illustrated in table IV. It indicated that the averaging relative errors, which were mainly caused by quantization, were approximately 3 %.

TABLE IV. ERROR RATE BETWEEN FIXED AND FLOATING POINT CALCULATION

	FFT_8	FFT_16	FFT_32
Error_r	2.9 %	3.1 %	3.6 %
Error_i	0 %	2.0 %	2.8 %

## VI. CONCLUSIONS

A low power biomedical digital signal processor ASIC based on hardware and software codesign methodology was

presented. The codesign methodology was employed. The ASIC was fabricated in SMIC 0.18- $\mu$ m mixed-signal CMOS 1P6M technology. The die area measures 5000  $\mu$ m by 2350  $\mu$ m. Simulation results indicated the dedicated architecture consumes only 1.5 % of the power for FFT calculations, as comparing with a pure embedded software approach. In the future we will characterize the error propagations of the scalable FFT, and also develop more complicated codesign strategies to host a greater variety of biomedical spectrum analysis applications.

## REFERENCES

- [1] E. N. Bruce, Biomedical Signal Processing and Signal Modeling. New York: IEEE Wiley, 2001.
- [2] [Rangayyan, Biomedical Signal Analysis: A Case-Study Approach. IEEE Press, 2002.
- [3] Cerutti, S., "In the Spotlight: Biomedical Signal Processing," Biomedical Engineering, IEEE Reviews in , vol.1, pp.8-11, 2008
- [4] Ajit P. Yoganathan<sup>1</sup>, Ramesh Gupta<sup>1</sup> and William H. Corcoran<sup>1</sup>. Fast fourier transform in the analysis of biomedical data. Medical and Biological Engineering and Computing. Vol 14, Number 2. pp 239-245, 1976
- [5] CESARELLI M. ; CLEMENTE F. ; BRACALE M. ; A flexible FFT algorithm for processing biomedical signals using a personal computer. Journal of biomedical engineering, 1990, vol. 12, no6, pp. 527-530
- [6] Basano, L.; Ottonello, P., "Real-Time FFT to Monitor Muscle Fatigue," Biomedical Engineering, IEEE Transactions on , vol. BME-33, no. 11, pp. 1049-1051, Nov. 1986
- [7] Benny Lo, Surapa Thiemjarus, Rachel King and Guang Zhong Yang, "Body Sensor Network ?A Wireless Sensor Platform for Pervasive Healthcare Monitoring" Adjunct Proceedings of the 3rd International Conference on Pervasive Computing (PERVASIVE 2005), pp.77-80, May 2005
- [8] Lei Wang, Surapa Thiemjarus, Benny Lo, and Guang-Zhong Yang, Toward A Mixed-Signal Reconfigurable ASIC for Real-Time Activity Recognition, 5th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2008) and 5th International Summer School and Symposium on Medical Devices and Biosensors (ISSS-MDBS 2008), Hong Kong, Jun. 1-3, 2008
- [9] T. Sumanaweera and D. Liu, Medical image reconstruction with the FFT, in GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation, M. Pharr, Ed. Addison-Wesley, March 2005, pp. 765-784
- [10] De Michell, G.; Gupta, R.K., "Hardware/software co-design," Proceedings of the IEEE , vol.85, no.3, pp.349-365, Mar 1997
- [11] www.arm.com
- [12] Luca Benini, Giovanni De Micheli, Enrico Macii, et al. Address Bus Encoding Techniques for System-Level Power Optimization. Design, Automation, and Test in Europe. Part IV. 2008.1, pp. 275-289
- [13] Mehta, H.; Owens, R.M.; Irwin, M.J., "Some issues in gray code addressing," VLSI, 1996. Proceedings., Sixth Great Lakes Symposium on , vol., no., pp.178-181, 22-23 Mar 1996
- [14] C.H.Sung, K.B.Lee, and C.W.Jen, "Design and implementation core", ASIA-Pacific Conference on ASIC ,pp.295-298, 2002
- [15] C. S. Wallace suggestion for fast multiplier [j] .IEEE transactions on electronic compute, 1964, 13 (2): 14-17.
- [16] A. D. booth. A signed binary multiplicand technique [J] .quarterly journal of mechanics and Applied Mathematics, 1951, 4(2): 236-240.