SYSCORE: A Coarse Grained Reconfigurable Array Architecture for Low Energy Biosignal Processing

Kunjan Patel, Séamas McGettrick and Chris J. Bleakley
UCD Complex and Adaptive Systems Laboratory,
UCD School of Computer Science and Informatics,
University College Dublin, Dublin, Ireland
Email: {kunjan.patel,seamas.mcgettrick,chris.bleakley}@ucd.ie

***Abstract*—The promise of 24/7 patient monitoring and online diagnosis using wearable and implantable biomedical devices has engendered significant research interest in the development of low power biosignal processing platforms. A novel Coarse Grained Reconfigurable Array (CGRA) architecture is presented for low power, real time processing of biomedical signals. The proposed architecture differs from previously proposed CGRAs in that it is designed for low power, rather than for high performance. The proposed architecture was implemented in a software modeler and simulator and in Verilog. The architecture was shown to provide average savings in energy consumption of 62% compared to conventional DSP and SIMD processors and average speed ups of 30x and 8x, respectively, for typical biosignal processing algorithms.**

***Keywords*-coarse grained reconfigurable architecture, systolic, low energy, biosignal**

I. Introduction

At present, there is significant research and commercial interest in developing wearable and implantable biomedical devices which can perform online, continuous biosignal processing allowing for automated real time symptom detection and diagnosis [1]. Although sampled at a comparatively low rate (hundreds of Hz), the multi-channel nature of biosignals (possibly tens of channels) and the difficulty of the detection and classification problem make the signal analysis task computationally expensive. In addition, most wearable and implantable applications require that devices are compact and have battery lifetimes of weeks or even months. The need for significant savings in the energy consumption of on-chip biosignal processing has led to research interest in low power biosignal processor platforms [1].

General purpose processors cannot meet the high performance and low power requirements of biosignal processing applications at the same time [1]. Custom hard-wired Application Specific Integrated Circuits (ASICs) can achieve the performance and power consumption requirements but do not provide enough flexibility and are not cost effective given the low production volumes.

Herein, we propose a novel Coarse Grained Reconfigurable Array (CGRA) architecture, named SYSCORE, for low-power on-chip biosignal processing. To the authors' knowledge no previous paper has studied the used of CGRAs for biosignal processing applications. The architecture supports mapping of irregular algorithms by means of a novel interconnection unit, **R**ound**a**bout **I**nterconnect (RAI), wherein non-nearest neighbor data transfer is supported without the area and power cost of a dense interconnect. The architecture provides significant energy savings by: eliminating the fetch-decode steps of traditional processors (via reconfiguration); significantly reducing the number of intermediate data RAM accesses (via systolic data reuse), reduced logic switching (via compact functional units); and by voltage scaling (via parallelism).

II. Related work

CGRAs have been proposed for multimedia, embedded and DSP applications [2], [3], [4], [5], [6], [7], [8], [9]. Most previous work on CGRAs has focused on increasing performance. The authors of [6] reported the power consumption of a CGRA but didn't propose, discuss or prove the effectiveness of particular power saving techniques.

Biosignal processing algorithms are typically one-dimensional and multi-channel. Single Instruction Multiple Data (SIMD) architectures (Morphosys, ADRES, CGRA Express) are efficient for algorithms which can be vectorized but are inefficient when processing steps can be concatenated, as in biosignal processing algorithms. Multimedia CGRA architectures (Morphosys, ADRES, CGRA Express) primarily provide support for vectorize-able two-dimensional image and video processing algorithms.

For accurate EEG analysis, a fixed bit-width of minimum 12-bits for IO and 24-bits for internal processing is required [1]. Certainly architectures with less than 16-bits are insufficient (PipeRench). Complex interconnections are required to support algorithms that

978-0-7695-4301-7/11 $26.00 © 2011 IEEE
DOI 10.1109/FCCM.2011.38

have irregular data access patterns, such as the FFT. This support is provided in some CGRAs, such as SmartCell. However, dense interconnections increase chip area and power. Two previous proposals claim to operate in a systolic fashion - PolySA and SmartCell. Neither are designed for low power. SYSCORE differs from these architectures in that its functional units have the capability of passing input data in parallel with outputting the results of the computation. This feature greatly facilitates systolic mapping of the applications (see later).

In SYSCORE, minimum energy is achieved by means of a fixed-point architecture rather than a floating-point architecture (PolySA). Based on publicly available information, we estimate that the functional units of all previous architectures are more than double that of SYSCORE except for PACT XPP and PolySA which are 20% and 50% larger, respectively.

III. Proposed Architecture

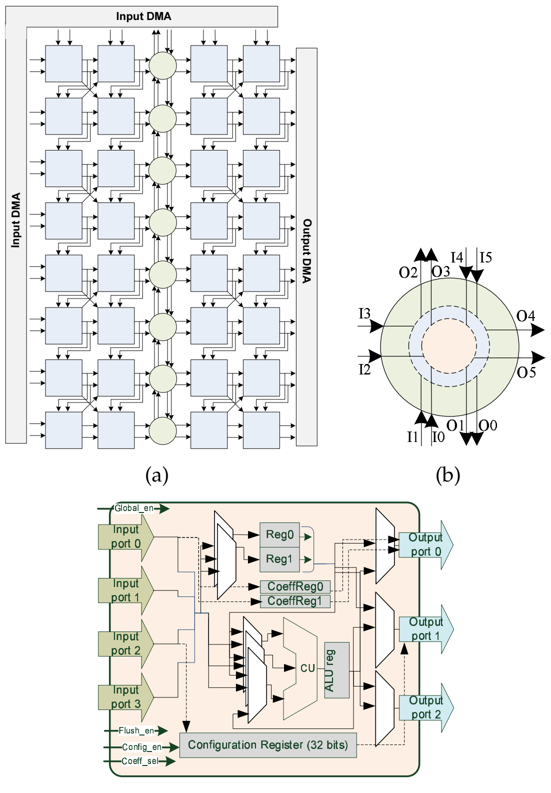
A. Overview

An 8x4 SYSCORE architecture is shown in Figure 1(a). There are two main elements: Configurable Function Units (CFUs) and RoundAbout Interconnect (RAI) units. The designer can use as many units as desired, according to the application performance targets and area constraints. Two Direct Memory Access (DMA) units inject data into the architecture from the West and North and one DMA units collects data from the architecture. Array configuration and DMA operations are controlled by the host processor.

B. CFU Design

Figure 1(c) shows the architecture of a CFU. The CFU has 4 input ports (In0-In3) and 3 output ports (Out0-Out2). It has a Computation Unit (CU) that differs from a conventional ALU/MAC in terms of the Set of Operations (SoOs) it can support. It supports MUL-ADD, MUL-SUB and compare (CMP) which are more useful than the traditional MAC operation for systolic algorithms mapping and feature extraction [10]. Because of the feedback from CU_reg to the CU, the CFU can be configured to perform a MAC operation without extra hardware cost. All the operations can be performed in a single cycle. Another unique feature of the CFU is that two data can be passed in parallel with the result computed by CU that greatly facilitates systolic mapping.

There are 2 General Purpose Registers (GPRs), 2 Coefficient Registers (CERs) and 1 CU register (CU_reg) in a CFU. GPRs are used to store input data from input ports, CERs are used to store coefficients for functions such as FIR and FFT. The CU_reg is used to store results from the CU unit. The bitwidth of all data registers and CU is 24 bits. Each CFU has one 32 bit configuration register (Config_reg). The bitfields of the configuration register control input and output multiplexers and CU.



(a) (b)

(c)

Figure 1. a) An 8x4 SYSCORE block; b) Conceptual diagram of RAI; c) CFU architecture

C. Control Signals

SYSCORE can operate in three different modes: configuration mode, execution mode and flush mode. The mode of operation is set using global control signals. SYSCORE operates in configuration mode when Config_en signal is high. Data from port In2 is stored in Config_reg and data from port In0 is stored in the CERs. SYSCORE operates in flush mode when the Flush_en signal is high. In this mode, each CFU passes data to the CFU to its east. This mode is used to transfer results stored in CFUs which are not directly accessible by DMA. The Coeff_sel signal selects between different CERs for input to the CU in both configuration and execution modes. The Global_en signal enables/disables the voltage supply to CFUs on row-by-row basis to save power when the CFUs are not in use.

D. Interconnections

As shown in Figure 1(a), all CFUs are connected to their nearest neighbour to the East and West. To avoid dense interconnections, cross interconnections are only introduced at odd numbered columns. Cross interconnections are useful for performing non-systolic functions, such as the FFT butterfly. A column of RAI elements is inserted after every second column of CFUs allowing data to pass from any Westerly CFU to any Easterly CFU in the next column. The conceptual structure of a RAI element is shown in Figure 1(b). Each RAI element has 6 input ports (I0-I5), 6 output ports (O0-O6) and a 16 bit configuration register. Output ports O0 to O6 can take data from input ports I2-I5, I2-I5, I0-I3, I0-I3, I0-I5 and I0-I5 respectively. As in a CFU, the RAI element can be reconfigured when SYSCORE is in configuration mode. There are no global interconnections, other than control signals (as described in the previous section), saving chip area and reducing power consumption.

IV. Mapping of Algorithms

Since no benchmark application suite is available for biosignal processing, we selected a set of algorithms from those generally used in biosignal processing applications. The algorithms listed in Table I were manually mapped using a methodology based on Synchronous Data Flow (SDF) and Control and Data Flow (CDF) graphs [10]. Once data is fetched, it is injected to the input ports of CFUs using DMA and partial results are passed directly to the next CFU in the following cycle, and so on, until the final result is obtained. No intermediate RAM access are needed, significantly reducing power consumption.

V. Results

A. Implementation

An 8x8 SYSCORE array was built using two array blocks, each as shown in Figure 1(a). The RaCAMS simulator was used to obtain performance results [11]. Simulation outputs were verified by comparison with Matlab. The hardware architecture was implemented in Verilog, the algorithms were mapped using SystemVerilog and 90nm CMOS technology library was used. The results shown were obtained by averaging over 10,000 iterations. The operating frequency was 100 MHz. The area of a CFU, a RAI element, SYSCORE were 38k, 6k and 2500k gates, respectively (NAND2 gate count equivalent). Because of the differences in technology libraries, it was not possible to directly compare SYSCORE's hardware metrics with those reported for previously proposed architectures. So, for comparison purposes, a hypothetical DSP processor was implemented which had 1 MAC unit, 24 registers, a fetch and decode unit, Program RAM, Data RAM and typical DSP ISA. Similarly, a hypothetical SIMD processor was implemented which has a DSP ISA with SIMD extensions. It is assumed that the SIMD processor can operate on 4 data words in one instruction [12]. Unused CFUs were disconnected from supply on a row-by-row basis to avoid unnecessary switching activity.

B. Performance and Energy Consumption

The performance and energy consumption comparison of SYSCORE with the DSP processor and the SIMD processor is shown in Table I. The clock frequency of SYSCORE, the DSP processor and the SIMD processor are assumed to be equal. Reconfiguration time and energy are included in the CGRA figures. DVFS was not taken into consideration in the analysis. For the CMOS technology used, we estimate that it can further reduce energy by up to 71%. As can be seen, the SYSCORE architecture gives energy savings in the range 30-99% and provides speed ups in the range 4-64x and 1-16x compared to the DSP and SIMD processor, respectively, depending on the algorithm. The energy savings arising are clearly dependent on CFU utilization. It can be seen that SIMD capability provides speed up. However, the energy consumption is almost same as for the DSP processor because SIMD reduces energy consumption in the fetch-decode stage of the processor cycle, but does not reduce memory access [10].

C. RAM Data Reuse (RDR)

When DVFS is not used, the majority of the power saving in the CGRA case is due to reduction in the number of RAM accesses. Figure 2 shows a comparison of RAM Data Reuse (RDR) between the DSP processor and SYSCORE architectures. RDR is given by:

$$RDR = \frac{Number\ of\ unique\ RAM\ addresses\ accessed}{Number\ of\ RAM\ accesses} \qquad (1)$$

It is clear from the results that data reuse in the SYSCORE architecture is considerably higher than in the DSP processor.

VI. Conclusion

A novel CGRA architecture, SYSCORE, is proposed herein for biosignal processing in wearable and implanted devices. The architecture allows systolic mapping of DSP algorithms to reduce memory accesses and so reduce power consumption. RAI interconnect elements were introduced to increase flexibility in supporting algorithms which cannot be easily mapped

Table I
DSP, SIMD AND SYSCORE PERFORMANCE AND ENERGY CONSUMPTION COMPARISON

Algorithm	Type	Cycles			Energy (pJ)			Speed up (x)		Energy Saving (%)	
		DSP	SIMD	SYSCORE	DSP	SIMD	SYSCORE	DSP	SIMD	DSP	SIMD
FIR	63	64	16	1	1852	1841	388	64	16	79	79
Matrix multiplication	4x4	64	16	15	4037	4015	2824	4	1	30	30
Wavelet transform	db2	8	2	1	254	2535	112	8	2	55	55
DFT	8 Point	61	16	1	99627	99617	791	61	16	99	99
FFT	8 point,radix-2	96	24	8	5345	5306	2886	12	3	46	46

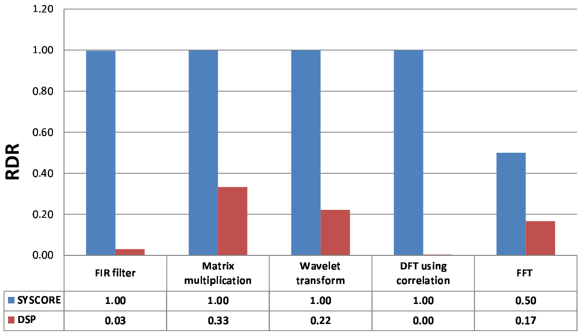


Figure 2. RDR comparison of some biosignal processing algorithms

systollicaly. The SYSCORE architecture gives 62% average energy savings compared to a conventional DSP and SIMD processor and average speed ups of 30x and 8x compared to conventional DSP and SIMD processors respectively. Planned future work includes assessment of the architecture for a wearable biomedical monitoring application and investigation of energy saving by making CFU bitwidth scalable.

Acknowledgment

This research was funded as a part of the Efficient Embedded Digital Signal Processing for Mobile Digital Health (EEDSP) cluster (grant ref:07/SRC/I1169) by Science Foundation Ireland (SFI).

References

[1] K. Patel, C.-P. Chua, S. Faul, and C. J. Bleakley, “Low power real-time seizure detection for ambulatory EEG,” in *Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on*, April 2009, pp. 1–7.

[2] H. Singh, M.-H. Lee, G. Lu, F. Kurdahi, N. Bagherzadeh, and E. Chaves Filho, “MorphoSys: an integrated reconfigurable system for data-parallel and computation-intensive applications,” *Computers, IEEE Transactions on*, vol. 49, no. 5, pp. 465 –481, May 2000.

[3] S. Goldstein, H. Schmit, M. Budiu, S. Cadambi, M. Moe, and R. Taylor, “Piperench: A reconfigurable architecture and compiler,” *Computer*, vol. 33, no. 4, pp. 70–77, 2002.

[4] C. Ebeling, D. Cronquist, P. Franklin, and C. Fisher, “RaPiD-a configurable computing architecture for compute-intensive applications,” *University of Washington Department of Computer Science & Engineering Tech Report TR-96-11-03*, 1996.

[5] B. Mei, S. Vernalde, D. Verkest, H. De Man, and R. Lauwereins, “ADRES: An architecture with tightly coupled VLIW processor and coarse-grained reconfigurable matrix,” *Lecture notes in computer science*, vol. 2778, pp. 61–70, 2003.

[6] Y. Park, H. Park, and S. Mahlke, “CGRA express: accelerating execution using dynamic operation fusion,” in *CASES ’09: Proceedings of the 2009 international conference on Compilers, architecture, and synthesis for embedded systems*, 2009, pp. 271–280.

[7] H. Park, Y. Park, and S. Mahlke, “Polymorphic pipeline array: a flexible multicore accelerator with virtualized execution for mobile multimedia applications,” in *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2009, pp. 370–380.

[8] A. Sudarsanam, R. Barnes, J. Carver, R. Kallam, and A. Dasu, “Dynamically reconfigurable systolic array accelerators: A case study with extended Kalman filter and discrete wavelet transform algorithms,” *Computers Digital Techniques, IET*, vol. 4, no. 2, pp. 126 –142, 2010.

[9] C. Liang and X. Huang, “SmartCell: A power-efficient reconfigurable architecture for data streaming applications,” in *Signal Processing Systems, 2008. SiPS 2008. IEEE Workshop on*. IEEE, 2008, pp. 257–262.

[10] K. Patel and C. Bleakley, “Systolic Algorithm Mapping for Coarse Grained Reconfigurable Array Architectures,” *Reconfigurable Computing: Architectures, Tools and Applications*, pp. 351–357, 2010.

[11] K. Patel, S. McGettrick, and C. Bleakley, “Rapid functional modelling and simulation of coarse grained reconfigurable array architectures,” *Journal of Systems Architecture*, 2011, in Press, Accepted Manuscript.

[12] ARM Cortex™-A series Processors with NEON™ Technology. [Online]. Available: http://www.arm.com/products/processors/cortex-a/cortex-a15.php