



Title	SYSCORE: A Coarse Grained Reconfigurable Array Architecture for Low Energy Biosignal Processing
Authors(s)	Patel, Kunjan, McGettrick, Séamas, Bleakley, Chris J.
Publication date	2011-05-03
Publication information	Patel, Kunjan, Séamas McGettrick, and Chris J. Bleakley. "SYSCORE: A Coarse Grained Reconfigurable Array Architecture for Low Energy Biosignal Processing." IEEE, May 3, 2011. https://doi.org/10.1109/FCCM.2011.38 .
Conference details	19th IEEE International Symposium on Field-Programmable Custom Computing Machines (FCCM), Salt Lake City, Utah, USA, 1 - 3 May, 2011
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/7033
Publisher's statement	© 2011 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/FCCM.2011.38

Downloaded 2026-05-01 23:43:25

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information

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

Removed for Blind Review

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. Herein, 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 savings in energy consumption of up to 99% and speed up of up to 64 times compared to a conventional DSP processor for typical biosignal processing functions.

Keywords—coarse grained reconfigurable architecture, systolic, low energy, biosignal

I. INTRODUCTION

Biosignals, for example the Electroencephalogram (EEG), Electromyogram (EMG), Electrocardiogram (ECG) and Magnetoencephalogram (MEG), are used by clinicians for diagnosis of medical conditions and diseases. Current wearable devices, for example the EEG halter monitor, allow recording of biosignals but do not provide real time analysis of the data. All processing and analysis occurs offline, after the device has been returned to the clinician. These devices typically have battery lives of 2-3 days.

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]. Real time analysis is important for many conditions, including, for example, seizure detection in subjects with epilepsy. 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 signal analysis task computationally expensive [2]. In addition, most wearable and implant applications require that devices are compact and have battery lifetimes of weeks or even months [1]. Previous work has shown that, for most biosignal processing applications, on-chip processing consumes significantly less energy

than wireless transmission for the purposes of server-based processing [1]. These application requirements have led to research interest in low power on-chip biosignal processing.

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. However, ASIC solutions lack flexibility. Designing a different ASIC for every medical condition is not cost effective given the low production volumes. Hence, a flexible low-power processor platform that can support a wide range of applications within the biosignal processing domain is required is required.

Herein, we propose a novel Coarse Grained Reconfigurable Array (CGRA) architecture, named SYSCORE, for low-power on-chip biosignal processing. A Coarse Grained Reconfigurable Array (CGRA) architecture consists of a grid of interconnected reconfigurable processing units which can perform logical or arithmetic operations. Unlike Field Programming Gate Arrays (FPGAs), the processing units are reconfigurable at the operation level rather than at the bit-level. This significantly reduces power consumption related to FPGAs while maintaining flexibility and increasing performance [3]. A host processor performs decision and control while computationally expensive tasks are offloaded to the CGRA.

The SYSCORE architecture is specifically designed for low-power biosignal processing and so differs from previous CGRA architectures in a number of ways. Firstly, the architecture is systolic in that, for regular biosignal processing algorithms, functional unit input and output data is pumped simultaneously and synchronously between nearest neighbor processing units arranged in an n-dimensional pipeline manner. Hard-wired systolic architectures are popular for DSP applications since they afford high throughput and data reuse [4]. In contrast to hard-wired arrays, the systolic CGRA approach allows flexibility, supporting a range of DSP algorithms. Secondly, the architecture supports mapping of irregular algorithms, such as the Fast Fourier Transform (FFT), by means of a novel

interconnection unit, Roundabout Interconnect (RAI), wherein non-nearest neighbor data transfer is supported without the area and power cost of a dense interconnect. Thirdly, in order to reduce power consumption, the SYSCORE architecture uses a minimal number of resources in each functional unit, 24-bit fixed-point with 2 operational units and 4 data registers. Overall, the architecture provides significant energy saving by eliminating the fetch-decode steps of traditional processors (by means of reconfiguration), by significantly reducing the number of intermediate data RAM accesses (by means of systolic data reuse), by reduced logic switching (by means of compact functional units) and by voltage scaling (by means of parallelism). To the authors' knowledge no previous paper has studied the used of CGRAs for biosignal processing applications.

The remainder of the paper is structured as follows. Section II presents a discussions of previous biosignal processing platforms and CGRA architectures. A description of the proposed architecture is given in Section III. In Section V, the power management in proposed architecture is described. Section VI presents results. Finally, the paper is concluded in Section VII.

II. RELATED WORK

A. Architectures for Biosignal Processing Applications

Previous work on processor platforms for biosignal processing has focused on multi-core and ASIC architectures.

Multicore architectures allow parallel processing of multichannel data. The authors of [5] presented a multiprocessor system-on-chip for real-time human heart monitoring and analysis. An architecture with 12 DSP processors was proposed to process 12 channel ECG data. Since the DSP cores run concurrently, the architecture implements a semaphore and interrupt system for communication and resource sharing. The HiBRID-SoC [6] consists of three adaptable programmable cores integrated using an AMBA AHB bus. Each core is optimized for a particular application set. The architecture is not area efficient if only a single core is used for a particular applications. Multi-core architectures provide performances increases over single core architectures but do not typically provide energy consumption reductions, other than by voltage scaling. In fact, the resource sharing and communication overhead is often significant in terms of power consumption and area.

ASIC designs can achieve high performance with low power consumption. The authors of [7] presented an ASIC for heart rate variability parameter monitoring and assessment. The ASIC was used in conjunction with a microcontroller. Application specific tasks were offloaded on the ASIC in which separate a separate

hardware blocks were dedicated to specific functions. The ASIC reduced power consumption by the factor of 7 compared to a standalone microcontroller. An energy-efficient ASIC for ultra low power wireless sensor nodes was presented in [8]. The ASIC was designed to perform the main functions of a proposed wireless Body Area Network sensor node. The main disadvantage of ASICs is that they lack flexibility, so targeting other low volume biomedical applications is not cost effective since ASIC redesign is a lengthy and costly process.

B. CGRA architectures

To the authors' knowledge, only one previous publication has investigated the power consumption of a CGRA architecture. In [9], the authors reported the power consumption of a CGRA but didn't propose, discuss or prove the effectiveness of particular power saving techniques. Most previous work on CGRAs has focused on increasing performance. An array level comparison and an architecture level comparison of SYSCORE with 12 previous CGRA architectures are presented in Tables I and II, respectively.

CGRAs have previously been proposed for multimedia, embedded and DSP applications. Biosignal processing applications are typically based on DSP algorithms. CGRAs for embedded applications (AMBER) do not support this functionality. Multimedia CGRA architectures (Morphosys, REMARC, ADRES, CGRA Express, PACT XPP) primarily provide support for vectorize-able two-dimensional image and video processing algorithms. Biosignal processing algorithms are typically one-dimensional, but multi-channel. Hence a DSP specific CGRA architecture offers the energy-delay product for biosignal processing applications. Minimum energy is achieved by means a fixed-point architecture rather than a floating-point architecture (REMARC, PolySA). 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).

Single Instruction, Multiple Data (SIMD) architectures (Morphosys, REMARC, ADRES, CGRA Express, PACT XPP) require that all processing units execute the same operation. This is efficient for algorithms which can be vectorized, for example in multimedia algorithms, but is inefficient when processing steps can be concatenated, as in biosignal processing algorithms. VLIW architectures (Montium) include processing units which can perform a large number of operations in parallel. This is an unnecessary overhead in systolic architectures as address calculation is not needed since intermediate RAM accesses are eliminated. Uniquely, SmartCell can execute operations in

Table I
COMPARISON OF DIFFERENT ARCHITECTURES AT ARRAY LEVEL

Architecture	Number format	Configuration granularity	Reconfiguration	Configuration scheme	Processing model	On chip memory/buffer	Power Management	Application domain
Morphosys[10]	Fixed	Coarse grained	Dynamic	Row/column wise broadcast	SIMD	Yes	No	Multimedia
Montium[11]	Fixed	Coarse grained	Dynamic	Using configuration unit via NoC	VLIW	Yes	No	DSP
PipeRench[12]	-	Mixed	Dynamic	In pipeline fashion	Pipelined datapath	Yes	No	DSP
REMARC[13]	Floating	Array of nano processors	Static	Using nano instruction received by global control unit	SIMD	Yes	No	Multimedia
RaPiD[14]	Fixed	Coarse grained	Static, 27% dynamic	Using control signals which flow through control bus	Linear pipeline	Yes	No	DSP
ADRES[15]	-	Coarse grained	Dynamic	Row/column wise broadcast	SIMD	Yes	No	DSP and multimedia
CGRA Express[16]	-	Coarse grained	Dynamic	Row/column wise broadcast	SIMD	Yes	No	Multimedia
PPA[17]	Fixed	Coarse grained	Dynamic	Row/column wise broadcast	Pipelined	Yes	No	Multimedia
AMBER[18]	Fixed	Coarse grained	Dynamic	Cross bar switches based routing network	Custom ASIP	Yes	No	Embedded
PACT XPP[19]	Fixed	Coarse grained	Dynamic	Row broadcast	SIMD	Yes	No	Multimedia, DSP
PolySA[20]	Floating point	Fine grained	Most of Static, Partially Dynamic	Control signal lines	Linear systolic	Yes	No	Filtering
SmartCell[9]	Fixed	Coarse grained	Dynamic	Instruction memory and SPI structure	SIMD/MIMD/Systolic	Yes	No	Stream processing
SYSCORE	Fixed	Coarse grained	Dynamic	Column wise broadcast, coefficient storing in parallel	Systolic	No	Yes	Biomedical

Table II
COMPARISON OF DIFFERENT ARCHITECTURES AT FUNCTIONAL UNIT LEVEL

Architecture	Bitwidth	Configuration bitwidth	Supported operations	Operations per cycle	Data passing	Data bypassing
Morphosys	28	32	General ALU operations, MAC, absolute difference, conditional ADD/SUB	1	No	No
Montium	16	16	General ALU operations	5	No	No
PipeRench	8	80	General ALU operations	1	Yes	No
REMARC	16	32	ADD, SUB, shift, all logical operations (30)	1	Yes	No
RaPiD	16	100	General ALU operations	1	Yes	No
ADRES	32	53	General ALU (20) Operations including branch operations	1	-	No
CGRA Express	32	55	20 Operations including branch operations	1	Yes	Yes
PPA	32	-	ADD, SUB, MUL and other common operations	1	Yes	No
AMBER	32	32	Logical operations or ADD/ SUB/ COMPARE or SHIFT	1	No	No
PACT XPP	32	-	General ALU operations	1	No	No
PolySA	32	4	ADD, MUL, DIV	1	No	No
SmartCell	16	64	General ALU operations, MAC, abs sum	3	No	Yes
SYSCORE	24	22	MUL, ADD, SUB, MUL-ADD, MUL-SUB, MAC, NOP	3	Yes	Yes

SIMD, MIMD (Multiple Instruction, Multiple Data) or systolic fashion due to the inclusion of dense interconnections and instruction memory. However, the rich interconnect and associated instruction memory consume 53% of the total power, which is not appropriate for low power applications.

Two previous proposals claim to operate in a systolic fashion - PolySA and SmartCell. Neither are designed for low power. PolySA is floating-point and SmartCell is rich in interconnect logic. SYSCORE also 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). Previously proposed architectures (e.g. RaPiD, PPA) are capable of passing data to nearest neighbor processing units but cannot pass input data and output data in parallel with operation execution.

Due to its data access pattern, the Fast Fourier Transform (FFT) cannot be easily mapped to a systolic architecture. Hence more complex interconnect must be provided to allow for efficient computation of the FFT. This support is provided in some CGRAs, such as PolySA, Montium. However this requires dense interconnections which increases chip area and power. The SYSCORE architecture uses a reconfigurable RAI scheme to provide interconnect flexibility at low overhead.

Most previous CGRA functional units are large in terms of area. For example, REMARC using a floating point number format and AMBER contains 64 registers. 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

A 8x4 SYSCORE architecture is shown in Figure 1. 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. A column of RAI elements is inserted after every second column of CFUs to facilitate FFT computation. Array configuration and DMA operations are controlled by the host processor.

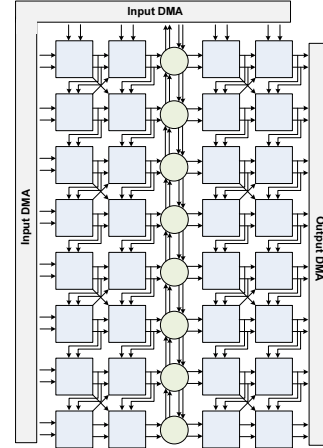


Figure 1. A SYSCORE block of 8x4

B. CFU Design

Figure 2 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 can perform computational operations. The CU differs from a conventional ALU/MAC in terms of the Set of Operations (SoOs) it can support. The CU can perform Addition (ADD), Subtraction (SUB), Multiplication (MUL), Multiply Accumulation (MAC), Multiply-Addition (MUL-ADD) and Multiply-Subtraction (MUL-SUB). These last operations, MUL-ADD and MUL-SUB, are more useful than the traditional MAC operation for systolic algorithms mapping [21]. 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. Two data can be passed in parallel with the result computed by CU through 3 output ports.

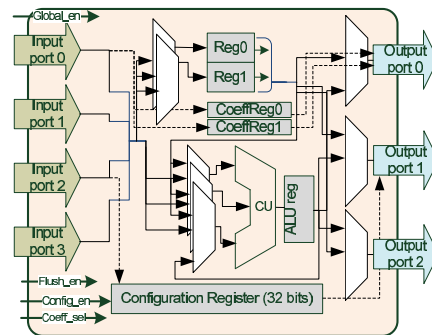


Figure 2. CFU architecture

C. Registers

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. All data registers are 24 bits.

D. Configuration Register

Each CFU has one 32 bits configuration register (Config_reg). It stores the configuration passed via port In2 when the Config_en signal is high. Table III shows the settings and purpose of different bit fields of the configuration register. The size of the configuration is 22 bits, the remaining bits are left for future use.

Table III
CONFIGURATION REGISTER BITFIELDS

Bitfield	Abbreviation	Purpose	Selection list
0 to 2	OP	Operation selection	ADD, SUB, MUL, MUL-ADD, MUL-SUB
3 to 5	ALU0	ALU selection line 0	In0-In3, 2 GPR
6 to 8	ALU1	ALU selection line 1	In0-In3, 2 CER, 2 GPR
9 to 11	ALU2	ALU selection line 2	In0-In3, 2 CER, 2 GPR, ALU_reg
12 to 13	REG0_sel	Input selection for GPR0	In0, In1
14 to 15	REG1_sel	Input selection for GPR1	In2, In3
16 to 17	OP0_sel	Input selection for Out0	ALU_reg, 2 GPR
18 to 19	OP1_sel	Input selection for Out1	ALU_reg, 2 GPR
20 to 21	OP2_sel	Input selection for Out2	ALU_reg, 2 GPR

E. Control Signals

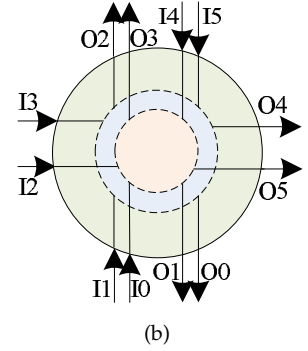
SYSCORE can operate in three different modes: configuration mode, execution mode and flush mode. The mode of operation is set using the following global control signals:

1) *Config_en*: When this signal is high, SYSCORE operates in configuration mode. The data from port In2 is stored in Config_reg and the data from port In0 is stored in the CERs.

2) *Flush_en*: When this signal is high, SYSCORE operates in flush mode. In this mode, each CFU passes data to the CFU in the east direction. This mode is used to transfer results which are stored in CFUs which is not directly accessible by DMA.

3) *Coeff_sel*: This signal controls switching between different CERs for input to the CU in both configuration and execution modes.

4) *Global_en*: This signal enables/disables voltage supply to a row of CFUs to save power when the CFUs are not in use. For simplicity of control, CFUs can be turned off row wise, not individually.



Output	Input from
0	2-5
1	2-5
2	0-3
3	0-3
4	0-5
5	0-5

Figure 3. a) Conceptual diagram of RA interconnect and b) Input port selection options for output ports

F. Interconnections

As shown in Figure 1, all CFUs are connected to their nearest neighbor 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. Cross interconnect functionality is provided by RAI elements that allow data to pass from any Westerly CFU to any Easterly CFU. The conceptual structure of a RAI element is shown in Figure 3. Each RAI element has 6 input ports (I0-I5), 6 output ports (O0-O6) and a 16 bit configuration register. As in a CFU, the RAI element can be reconfigured when SYSCORE is in the configuration mode. The output ports of the RAI element can be configured to take data from the input ports. Figure 3(a) shows the available output port options in RAI. There are no global interconnections, except control signals (as described in the previous section), which saves chip area and reduces power consumption and control overheads.

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 IV were manually mapped using a methodology that is based on Synchronous Data Flow (SDF) and Control and Data Flow (CDF) graphs. FIR filtering is a common function in biosignal processing. Figure 4 shows the SDF graph, the CDF graph and topology matrix for a 5 tap FIR filter. Once data is fetched, it is injected to the input ports of CFUs using DMA and the 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, which significantly reduces power consumption. More details on the algorithms mapping approach can be found in [21].

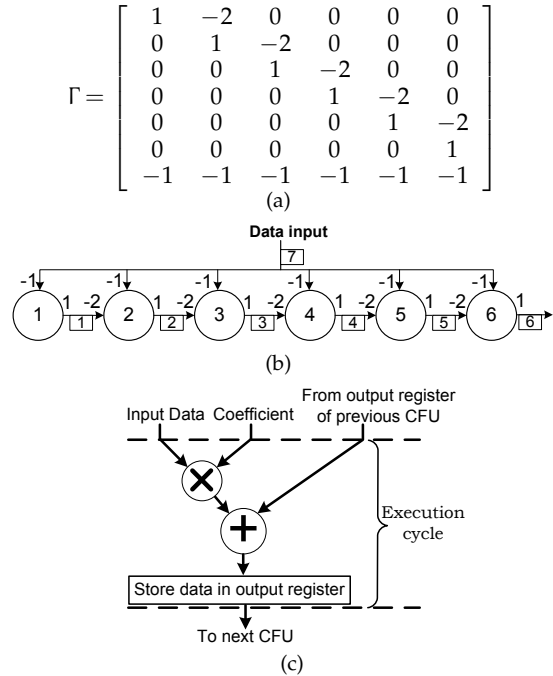


Figure 4. a) FIR filter topology matrix; b) 5 taps FIR filter SDF graph and c) FIR filter CDF graph for a single CFU

V. POWER MANAGEMENT

A. Dynamic Voltage and Frequency Scaling (DVFS)

DVFS can significantly reduce the power consumption of processing architectures [22]. The power consumption of an architecture is directly proportional to the clock frequency (f) and the square of the supply voltage (V_{dd}). Delay is proportional to $1/f$. Hence, the inherent parallelism in SYSCORE can be traded for reduced energy consumption by minimizing the clock frequency and supply voltage. Section VI-D, shows how energy varies with respect to V_{dd} for different applications.

B. Turning off Unused CFUs

The `Global_en` line allows unused CFUs to be disconnected from supply on a row-by-row basis. Energy savings arising are clearly dependent on CFU utilization.

VI. RESULTS

A. Implementation

An 8x8 SYSCORE array was built use two array blocks, each as shown in Figure 1. The architecture was

implemented and application mapping was performed using a software model and simulator called RaCAMS [23]. RaCAMS was used to obtain performance results. Simulation outputs were verified by comparison with Matlab. The hardware architecture was implemented in Verilog and the algorithms were mapped using SystemVerilog. The Synopsys tool chain was used for synthesis, RTL simulation, gate-level simulation and power estimation. A 90nm CMOS technology library was used. The area of SYSCORE was 7750 cells and operating frequency was 100 MHz. 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. The processor architecture had 1 MAC unit, 24 registers, a fetch and decode unit, Program RAM and Data RAM. The ISA of this DSP can execute the instructions that a typical DSP processor can execute.

B. Performance and Energy Consumption

The performance and energy consumption comparison between SYSCORE and the DSP is shown in Table IV. The clock frequency of SYSCORE and the DSP are assumed to be equal. Reconfiguration time and energy are included in the CGRA figures. DVFS was not taken into consideration in the analysis. It can be seen that SYSCORE architecture gives up to 99% of energy savings and provides speed up factor of 64 depending on the algorithm.

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 5 shows a comparison of RAM Data Reuse (RDR) between DSP processor and SYSCORE architecture. RDR is given by:

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

A RDR value close to 1 indicates that RAM locations are only accessed once. A value close to 0 indicates that same RAM locations are accessed many times.

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

D. Dynamic Voltage and Frequency Scaling

Figure 6 shows energy saving by reducing the voltage for the different algorithms.

Table IV
DSP AND SYSCORE PERFORMANCE AND ENERGY CONSUMPTION COMPARISON

Algorithm	Type	Iterations	Cycles		Energy (μ Joules)		Speed up factor	Energy savings (%)
			DSP	SYSCORE	DSP	SYSCORE		
FIR	63	10000	640000	10000	18	4	64	79
Matrix multiplication	4x4	10000	640000	150000	41	28	4	30
Wavelet transform	db2	10000	80000	10000	3	1	8	55
DFT	8 Point	10000	610000	10000	996	8	61	99
FFT	8 point, radix-2	10000	96000	8000	29	54	12	46

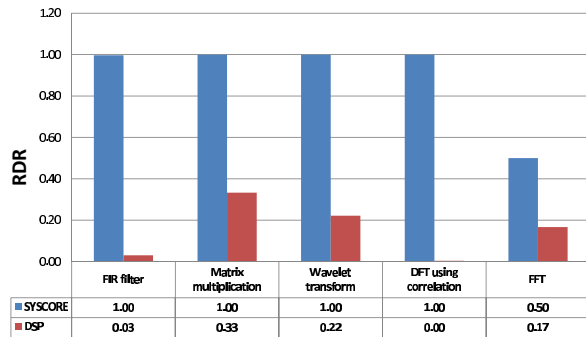


Figure 5. RDR comparison of some biosignal algorithms

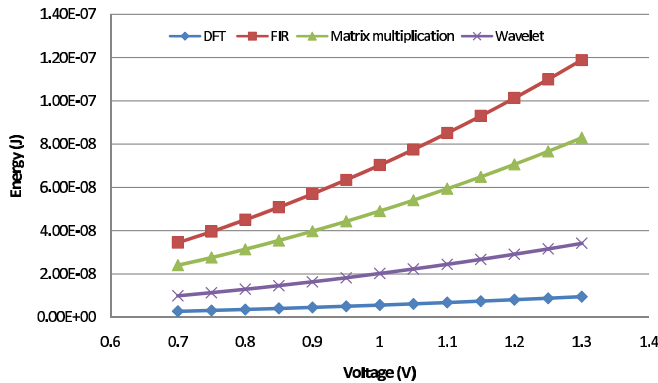


Figure 6. DVFS in SYSCORE

VII. CONCLUSION

Biosignal processing in wearable and implanted devices requires very low energy, medium performance DSP processing platforms. A novel CGRA architecture, SYSCORE, is proposed herein for these applications. The architecture allows systolic mapping of DSP algorithms to reduce memory accesses and so reduce power consumption. RAI interconnect elements were introduced to increase the flexibility in the architecture in supporting algorithms which cannot be easily mapped systolically. A number of power savings techniques were studied to further reduce the energy

consumption. The SYSCORE architecture gives up to 99% of energy savings and up to 64 times speed up compared to a conventional DSP processor architecture implemented using the same technology. Planned future work includes assessment of the architecture for a wearable biomedical monitoring application.

ACKNOWLEDGMENT

Removed for blind review

REFERENCES

- [1] "Removed for blind review."
- [2] H. Sackman, *Biomedical information technology: global social responsibilities for the democratic information age*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1997.
- [3] R. Hartenstein, "Coarse grain reconfigurable architecture (embedded tutorial)," in *Proceedings of the 2001 conference on Asia South Pacific design automation*. ACM New York, NY, USA, 2001, pp. 564–570.
- [4] K. Johnson, A. Hurson, and B. Shirazi, "General-purpose systolic arrays," *Computer*, vol. 26, no. 11, pp. 20–31, 1993.
- [5] I. Al Khatib, D. Bertozzi, F. Poletti, L. Benini, A. Jantsch, M. Bechara, H. Khalifeh, M. Hajjar, R. Nabiev, and S. Jonsson, "Mpsoc ecg biochip: a multiprocessor system-on-chip for real-time human heart monitoring and analysis," in *CF '06: Proceedings of the 3rd conference on Computing frontiers*. New York, NY, USA: ACM, 2006, pp. 21–28.
- [6] H. Stolberg, M. Berekovic, L. Friebe, S. Moch, S. Fl "ugel, X. Mao, M. Kulaczewski, H. Klußmann, and P. Pirsch, "HiBRID-SoC: a multi-core system-on-chip architecture for multimedia signal processing applications," 2003.
- [7] W. Massagram, N. Hafner, M. Chen, L. Macchiarulo, V. M. Lubecke, and O. Boric-Lubecke, "Digital heart-rate variability parameter monitoring and assessment asic," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 4, no. 1, pp. 19–26, Feb. 2010.

- [8] X. Zhang, H. Jiang, L. Zhang, C. Zhang, Z. Wang, and X. Chen, "An energy-efficient ASIC for wireless body sensor networks in medical applications," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 4, no. 1, pp. 11–18, Feb. 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] H. Parizi, A. Niktash, N. Bagherzadeh, and F. Kurdahi, "MorphoSys: A Coarse Grain Reconfigurable Architecture for Multimedia Applications," *Memory*, vol. 1024, p. 16.
- [11] P. Heysters, G. Smit, and E. Molenkamp, "Montium-balancing between energy-efficiency, flexibility and performance," 2003.
- [12] 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.
- [13] T. Miyamori and K. Olukotun, "REMARC: Reconfigurable Multimedia Array Coprocessor," in *IEICE Transactions on Information and Systems E82-D*, 1998, pp. 389–397.
- [14] C. Ebeling, C. Ebeling, D. C. Cronquist, D. C. Cronquist, P. Franklin, P. Franklin, C. Fisher, and C. Fisher, "Rapid - a configurable computing architecture for compute-intensive applications," 1996.
- [15] 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*, pp. 61–70, 2003.
- [16] 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*. New York, NY, USA: ACM, 2009, pp. 271–280.
- [17] 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*. ACM, 2009, pp. 370–380.
- [18] F. Mehdipour, H. Honda, H. Kataoka, K. Inoue, I. Kataeva, K. Murakami, H. Akaike, and A. Fujimaki, "Mapping scientific applications on a large-scale datapath accelerator implemented by single-flux quantum (SFQ) circuits," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2010*, 2010, pp. 993–996.
- [19] V. Baumgarte, G. Ehlers, F. May, A. Nüchel, M. Vorbach, and M. Weinhardt, "PACT XPP: A self-reconfigurable data processing architecture," *the Journal of Supercomputing*, vol. 26, no. 2, pp. 167–184, 2003.
- [20] 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.
- [21] "Removed for blind review."
- [22] B. Zhai, D. Blaauw, D. Sylvester, and K. Flautner, "Theoretical and practical limits of dynamic voltage scaling," in *Proceedings of the 41st Annual Design Automation Conference*. ACM, 2004, p. 873.
- [23] "Removed for blind review."