Published in IET Circuits, Devices & Systems
Received on 11th January 2012
Revised on 17th May 2012
doi: 10.1049/iet-cds.2012.0011

Special Issue on Thermal, Power and Timing Modeling, Design and Simulation

ISSN 1751-858X

Low-power processor architecture exploration for online biomedical signal analysis

A.Y. Dogan[1] *J. Constantin*[2] *D. Atienza*[1] *A. Burg*[2] *L. Benini*[3]

[1]*Embedded Systems Laboratory (ESL) – EPFL, Lausanne – 1015, Switzerland*
[2]*Telecommunications Circuits Laboratory (TCL) – EPFL, Lausanne – 1015, Switzerland*
[3]*UNIBO-Micrel Laboratory, Viale Risorgimento 2, Bologna 40136, Italy*
E-mail: ahmed.dogan@epfl.ch

Abstract: In this study, the authors explore sequential and parallel processing architectures, utilising a custom ultra-low-power (ULP) processing core, to extend the lifetime of health monitoring systems, where slow biosignal events and highly parallel computations exist. To this end, a single- and a multi-core architecture are proposed and compared. The single-core architecture is composed of one ULP processing core, an instruction memory (IM) and a data memory (DM), while the multi-core architecture consists of several ULP processing cores, individual IMs for each core, a shared DM and an interconnection crossbar between the cores and the DM. These architectures are compared with respect to power/performance trade-offs for different target workloads of online biomedical signal analysis, while exploiting near threshold computing. The results show that with respect to the single-core architecture, the multi-core solution consumes 62% less power for high computation requirements (167 MOps/s), while consuming 46% more power for extremely low computation needs when the power consumption is dominated by leakage. Additionally, the authors show that the proposed ULP processing core, using a simplified instruction set architecture (ISA), achieves energy savings of 54% compared to a reference microcontroller ISA (PIC24).

1 Introduction and related work

According to the World Health Organization, cardiovascular and modern human behaviour-related diseases are the major cause of mortality worldwide [1]. Close and potentially continuous medical supervision is strongly needed to control these types of diseases. They are thus expected to soon require healthcare costs and medical management needs that are unsustainable for traditional healthcare delivery systems. Personal health monitoring systems are poised to offer large-scale and cost-effective solutions to this problem. Wireless body sensor networks (WBSNs) are the enabling technology for such personal health systems [2, 3]. A WBSN for health monitoring consists of a number of light-weight sensor nodes attached to the human body, where each node is responsible for processing a specific low rate physiological signal. For instance, one of the most important physiological signals is the electrocardiogram (ECG), which is typically acquired at sampling rates between 125 Hz and 1 kHz to capture the often important details of the waveform. In order to monitor the heart rate for extended periods of time (up to multiple days or weeks), an ultra-low-power (ULP) design with embedded biomedical signal processing for feature extraction on the sensor node is necessary [4] to reduce the costly signal storage or transmission [5] to the essence.

An effective technique to reduce the computational power consumption is supply voltage scaling, potentially all the way to sub-threshold operation. In the literature, voltage scaling and its limitations and disadvantages such as performance loss, the risk of functional failure, performance variability etc., have been analysed extensively [6–9] and various low-power architectures have been presented. For example, Chen *et al.* [10] proposed a sensor platform capable of nearly-perpetual operation by using harvesting from solar cells. The proposed single processor architecture has an ARM Cortex M3 core with both retentive and non-retentive SRAM and a power management unit which controls the active and ultra low power sleep modes. In another work, Hanson *et al.* [11] presented a new ultra low energy processor with low voltage operations for wireless monitoring systems. They optimised the standby power consumption of the processor with the help of a new low leakage memory macros, memory size and instruction set adjustments and power gating. However, the main issue with low-voltage operation is the performance loss, which, for a given processing requirement, can limit the degree of use of voltage-scaling. Parallel computing using multiple cores can alleviate this issue, provided that the algorithms to be executed can be parallelised. To this end, Dreslinski *et al.* [12] proposed a near threshold computing (NTC), cluster-based multi-processor architecture with a shared cache that operates at a higher supply voltage to be able to serve multiple cores at the same time. Finally, Pu *et al.* [13] introduced a sub/near threshold co-processor for low energy mobile image

processing using architecture level parallelism to compensate for the performance loss.

Unfortunately, even though researchers focused on low energy solutions in both multi-core and single-core approaches individually, the two approaches have not been compared in terms of energy efficiency for the moderate workloads that are typical for biomedical applications. Thus, in this paper we propose as a main contribution a single-core and a multi-core architecture for embedded biomedical signal processing on WBSNs, where algorithms have a limited, yet, at near-threshold voltage, non-negligible complexity and where a significant part of the processing can be done in parallel. We also propose an ULP custom processing core with minimal instruction set, which achieves up to 54% energy saving with respect to a well established instruction set architecture (ISA), namely the PIC24 ISA [14]. This custom core is used in the single- and multi-core architectures as the processing element. We explore the power/performance trade-offs between the single- and multi-core architectures for different online biomedical signal processing applications while exploiting NTC. Our results show that the multi-core approach achieves 62% power saving with respect to the single-core approach for high biosignal computation workloads (i.e. 167 MOps/s), however it consumes up to 46% more power than the single-core approach for relatively lighter workloads when the power consumption is dominated by leakage.

The rest of this paper is organised as follows: first Section 2 analyses the biomedical processing features and introduces several reference benchmarks. Next, Section 3 describes our ULP processing core as well as the single-core and multi-core processor architectures. Then, Section 4 gives the experimental setup and results. Finally, we summarise the main conclusions of this work in Section 5.

2 Biopotentials processing features

Signal processing on wearable personal health monitoring systems consists mostly of arithmetic computations with relative complexity on single- or multi-input biological signals. Hence, it has been recently shown that they can be optimised to run in real-time on typical embedded low-power microcontrollers [15, 16]. For instance, Rincon *et al.* [16] showed how delineation of multi-lead ECG signals, using a complex multi-scale wavelet transform algorithm, can be realised on a commercially available personal health monitoring system node with limited computation capability. In fact, multi-lead biological signals analysis is often needed to obtain an accurate view of biological events. However, the analysis of these multi-lead signals entails considerably parallel computation opportunities, which can be exploited on multi-core processing platforms.

In this work, we consider three different reference benchmarks: two different ECG signal compression applications and one ECG signal conditioning application. The first data compression application is based on compressed sensing (CS) theory [3], while the second one is a discrete wavelet transform (DWT)-based data compression algorithm. Our reference benchmarks have fundamental applications in WBSN systems for automated ECG analysis as well as data compression [3, 16]. All of our reference benchmarks are real-time multi-lead ECG processing applications that operate on eight leads (a typical configuration) to make the system more accurate and resilient to noise artefacts. Moreover, all the benchmarks perform computations on a block of 512 samples of ECG data (sampled at 250 Hz) per lead. However, to investigate different processing requirements related to the application, we consider ECG sampling rates between 125 Hz and 1 kHz for capturing signals with quality levels from barely acceptable to excellent.

The first reference benchmark in this work is an ECG processing application which comprises two components: CS and Huffmann coding. CS [3] performs a 50% compression on a block of ECG data per lead whereas the Huffmann coding part encodes the compressed data further for wireless transmission. In CS-based data compression, only few linear random measurements are acquired from the ECG signal. The algorithm implicitly relies on the sparse characteristics of ECG signal to guarantee accurate reconstruction.

The second reference benchmark, DWT-based data compression [3], performs a 50% compression on a block of ECG data per lead similar to the CS-based data compression. As opposed to CS, DWT-based data compression explicitly exploits the sparsity of the ECG signal by computing its sparse expansion and adaptively encoding it with Huffmann coding. The DWT-based data compression requires more complex computations than the CS-based data compression because of discrete wavelet transformation.

The last reference benchmark in this work, ECG signal conditioning, is referred herein as ECG2, and is based on the morphological filtering given in [17]. Raw ECG signals, even when recorded in a controlled environment, contain various types of noise and baseline drifts. ECG2 performs baseline correction and noise suppression on a block of ECG data per lead. The corresponding kernel has a broad application in WBSN systems, especially in automated ECG analysis.

3 Processing platform architectures

3.1 Processing cores

We consider for the processing platforms two different processing core architectures: Firat and TamaRISC. Firat has a well-established and extensive ISA, which is a subset of the PIC24 ISA from Microchip [14]. TamaRISC is a custom designed processor with a similar core architecture as Firat, especially regarding memory interfaces. The main differences are a minimal ISA and overall reduction of processor complexity [true reduced instruction set computer (RISC)]. The following subsections explain both processing architectures in detail.

3.1.1 Firat: Firat, shown in Fig. 1*a*, is a RISC-like architecture with a Harvard memory model. The simple three-stage pipeline (fetch, decode and execute stages) matches the low to moderate performance requirements of biomedical applications and reduces the number of registers that need to be clocked. The core operates on a data word length of 16-bit, comprises 16 working registers and three external memory ports (one for instruction read, one for data read and one for data write), all accessible in the same cycle. The register file has four read ports and four write ports. The core addresses an instruction memory (IM) with a 24-bit wide program counter. The instruction word size is 24-bit and almost all instructions occupy only a single instruction-word. Each single-word instruction is divided

doi: 10.1049/iet-cds.2012.0011

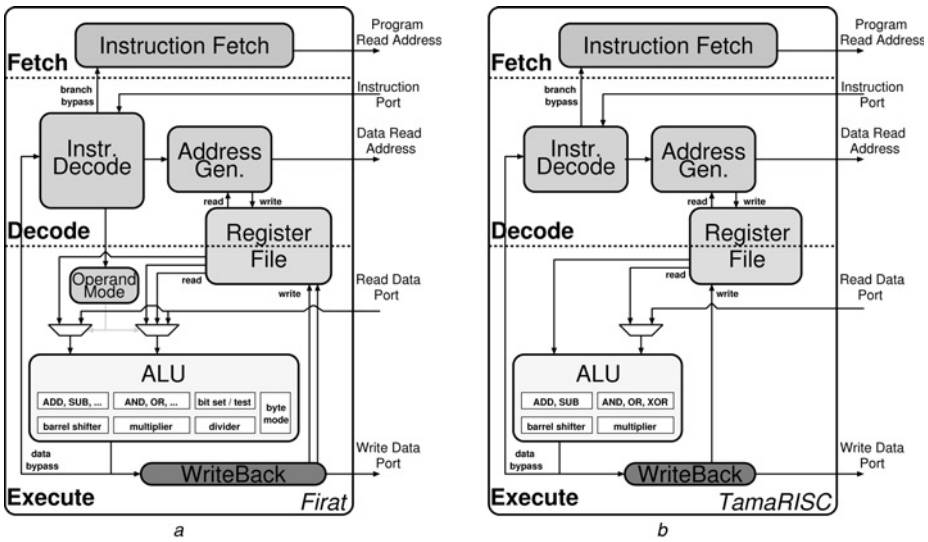

Fig. 1 *Processing core architectures*

a Firat
b TamaRISC

into an 8-bit opcode, which specifies the instruction type (up to three different opcodes per operation), and one or more operands, which further specify the operation of the instruction. The instructions operate on either two or three operands. The core uses a dual-port data memory (DM) with 16-bit wide addresses, and can hence read as well as write one 16-bit data word each, in the same cycle. The architecture additional comprises a specific PIC24 feature that allows addressing of the register file inside the DM address space, that is, a mapping of all processor registers onto DM addresses. Especially in the context of data bypassing (hazards), combined with address generation, this feature requires additional register ports.

The options for the instruction operands depend on the corresponding operation, and can be divided into 24 distinctive operand mode groups. In general, the operand modes can be described as follows. Both operands of the two-operand instructions can be either a register (with different addressing modes for the first operand) or a direct memory address or a literal of various sizes (4, 6, 8, 10, 14 or 16 bits). The instructions with three operands have always the first operand as a working register, the second operand can be either a register with different addressing modes or a literal, and the third operand is always a register with different addressing modes. The supported addressing modes are register direct, register indirect (with pre- or post-increment and decrement), as well as register indirect with signed offset. Almost all operand addressing modes work in byte or word mode, some also in double (32-bit) mode.

The ISA is a subset of the PIC24H/F ISA [14], and includes totally 66 instructions, which is still rather extensive and complex for a RISC-like ISA. The instructions can be divided into various groups, such as arithmetic logic unit (ALU) operations, program flow and control operations, bit-oriented operations, single- or multi-bit shifts and data-move operations. More specifically the ALU instructions comprise addition and subtraction with/without carry, logic operations (XOR, AND, OR), 16-bit signed/unsigned multiplication as well as signed/unsigned 32/16-bit (16/16-bit also possible) integer division. The shift operations offer both arithmetic and logic single- or multi-bit right/left shifts with the help of a barrel shifter. The program flow and control operations include CALL and RETURN instructions which can address the IM with a direct mode or relative to the program counter. Moreover, branching is possible in direct or indirect modes, with different condition modes dependent on the status register flags: carry, zero, negative and overflow. Data-move instructions enable single or double data move from register/memory to register/memory. Most of the instructions are executed in only one clock cycle, except for some instructions, such as CALL and RETURN instructions (requiring three clock cycles), double data-move instructions (two clock cycles), and division (17 clock cycles).

The core bypasses result data from the execute stage to the decode stage for matching register destinations of two subsequent instructions. For memory destinations however, no bypassing mechanism exists, which can result in additional stall cycles for read-after-write hazards on destinations with the same DM address.

3.1.2 TamaRISC: TamaRISC is a custom-designed RISC architecture, shown in Fig. 1*b*. The core architecture focuses on minimising the instruction set complexity, while still providing enough hardware support, especially regarding addressing modes, for efficient execution of the target biomedical signal-processing applications. The processor has a three-stage pipeline (fetch, decode and execute stages). The core operates on a data word length of 16-bit, comprises 16 working registers and three external memory ports, for one instruction read, data read and data write each in the same cycle. The register file has three read ports and four write ports to provide 32-bit double word writeback support. The core architecture is therefore similar compared to the Firat architecture, but with reduced complexity.

The instruction word length is 24 bits, and every instruction has a single-word size. All instructions are executed in one cycle, which is guaranteed by the complete data bypassing inside the core for register- and memory-write-back data.

The main reduction of complexity lies in the ISA, which comprises a total of 11 unique instructions, with eight ALUs, one general data-move and two program flow

instructions. The ALU supports addition, subtraction (each with optional carry/borrow), logical AND, OR and XOR, right (arithmetic or not) and left shift (barrel shifter), as well as full 16-bit by 16-bit multiplications (32 bit-result) on unsigned and signed data. All ALU instructions work on three operands, using the exact same addressing mode options for each instruction, which reduces the complexity of the architecture, since the operand fetch logic and the arithmetic operation are completely decoupled. Additionally, the instruction word encoding is designed as regular (fixed bit positions) and as simple as possible to allow for very efficient decoding of the operands and the different instruction words in general. The supported addressing modes are register direct, register indirect (with pre- or post-increment and decrement) as well as register indirect with offset. The second operand also supports the use of 4-bit literals. Regarding program flow instructions, branching is possible in direct and register indirect mode, as well as by offset with 15 different condition modes (dependent on the processor status flags: carry, zero, negative and overflow).

3.2 Processing platforms

The single-core and multi-core configurations include the same processing unit (PU) and a DM). However, the multi-core processing platform also involves a central data crossbar interconnect (D-Xbar), connecting the PUs with the shared DM.

Processing unit: A PU comprises a processing core and a 24-bit wide IM for 4 k instruction words (12 kBytes) which is sufficient for many typical biomedical applications on WBSNs such as delineation and CS data compression [3, 16]. One of the introduced processing core architectures, Firat or TamaRISC, is used in the PUs.

Data memory: The processing core (both the Firat and the TamaRISC architectures) can access the DM for reading and writing in the same clock cycle. Therefore the DM requires two separate access ports, one for reading and another one for writing. The 64 kByte of DM, required for multi-lead biomedical signal analysis, is split into $M = 16$ memory banks with 2 k words per bank. This configuration corresponds to the maximum available from our 2-port memory generator and it allows partial shutdown for leakage power reduction for applications with reduced memory requirements.

Data crossbar interconnect: The D-Xbar is a mesh-of-trees interconnection network to support high-performance communication between PUs and memories [18]. The interconnect is intended to connect a number of processing cores (in our case eight cores) to a multi-banked memory (i.e. 16 banks). The total memory access latency is one clock cycle; however, in case of multiple conflicting requests, for fair access to memory banks, a round-robin scheduler arbitrates access and a higher number of cycles is needed depending on the number of conflicting requests, with no latency in between.

3.2.1 Single-core processor architecture: The single-core processor architecture is shown in Fig. 2*a*. A simple selection logic connects the single PU to the individual memory banks and multiplexes the data. The system processes the 8-lead ECG signals sequentially.

3.2.2 Multi-core processor architecture: The multi-core processor design, shown in Fig. 2*b* consists of

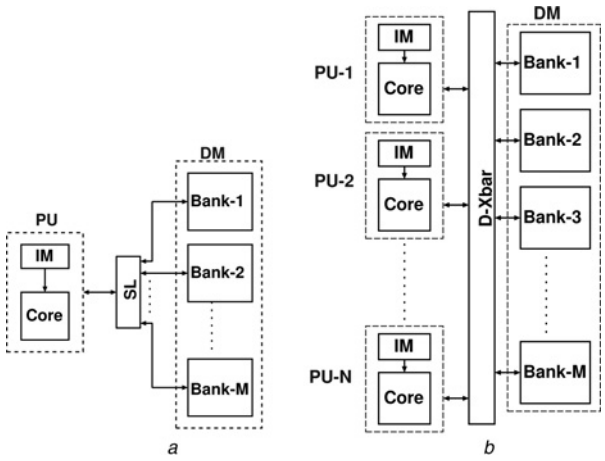

Fig. 2 *Processing platforms*

a Single-core architecture
b Multi-core architecture

$N = 8$ PUs with individual IMs. Each PU accesses the shared DM with 16 banks through the D-Xbar [18] to enable full access to the entire memory space for each PU. This architecture is different from the one proposed by Dreslinski *et al.* [12] in which several slower cores share a cache that is proportionally faster and thus requires a higher supply voltage. Compared to their architecture, which relies on a fully shared memory-block configuration, our proposed architecture simplifies the clock-network design [As seen in Table 3, the clock tree in the proposed architecture consumes only 5.7% of the whole power consumption.] and neither requires an additional faster clock, nor level-shifters between the cores and the shared cache. Furthermore, the ability to operate with only a single supply voltage considerably simplifies the overall system design and can result in additional energy savings, because multiple weakly loaded DC/DC-converters can be avoided. The drawback of our approach are the occasional access conflicts when two or more PUs access the same MB on the same port. In this case, a round-robin scheduler arbitrates access, while the waiting PUs are stalled using clock gating to avoid unnecessary active power consumption. As opposed to the single-core architecture, the multi-core architecture processes the eight-lead ECG signals in parallel (one lead per core).

4 Experimental results

Our experiments comprises mainly two phases: (i) analysis of the introduced processing cores in terms of energy efficiency for biomedical signal processing and (ii) exploring the power/performance trade-offs between the single- and multi-core architectures. To analyse the energy efficiency of the processing cores, we have built two single-core architectures with Firat and TamaRISC. This analysis is explained in detail in Section 4.2. In the second phase of our experiments, to explore the power/performance trade-offs between the single- and multi-core architectures, we have built the single- and multi-core designs with the more energy-efficient processing core. The reference benchmarks are executed on the designs for various workloads requirement (Ops/s) while exploiting voltage scaling to accomplish minimum power solutions. The scaling of the operating voltages is limited to the transistor threshold voltage level to avoid performance variability and

doi: 10.1049/iet-cds.2012.0011

functional failure issues occurring mainly at sub-threshold voltages. The power values at scaled voltages are calculated regarding the fact that the power decreases with the square of the supply voltage. In addition, we also analyse the architectures with respect to the ECG sampling rates corresponding to our application requirements. All the designs (two single-core and one multi-core design) are implemented in a 90-nm low-leakage process technology trading peak performance for significant leakage power reduction, especially in the memories.

4.1 Power characterisation framework

The evaluation and implementation flow for the architectures is shown in Fig. 3. The left and right sides of the figure show the parts of the design flow specifically related to the use of Firat and TamaRISC as the processing cores in the PUs. Moreover, the common flow when, either Firat or TamaRISC is used, is presented in the central part of the figure. Once the HDL code is prepared, it is integrated into the single- and multi-core architectures written in VHDL, providing the memory banks as well as the crossbar interconnect for the multi-core architecture. The complete system is then synthesised, placed, routed and optimised to have a full layout design. This design is then post-layout simulated using the memory contents extracted from the compiled benchmark program binary. The resulting trace file is used to perform an accurate power analysis of the complete system.

As indicated on the left side of Fig. 3, the processing core Firat is described manually in VHDL. However, the C compiler (MPLAB C-compiler) provided by Microchip Technology is used for compiling benchmarks, since the instruction set of Firat is a subset of PIC24H/F [14]. Moreover, the ISA of Firat is verified with the help of MPLAB integrated development environment [19] provided by Microchip Technology.

As opposed to Firat, the TamaRISC architecture is described in an automated design tool LISA (Language for ISAs) [20], which enables rapid design space exploration for the software as well as hardware aspects of the system. Synopsys processor designer (PD) is used to generate the RTL description of the core, a cycle accurate instruction set simulator as well as the necessary software tools (assembler, linker) for creating program binaries from the LISA specification. Additionally, the tool chain is extended by a custom C compiler, which is based on the PD built-in CoSy compiler development system. The C compiler allows for easier benchmark development. The design flow contains a custom regression test for cycle accurate

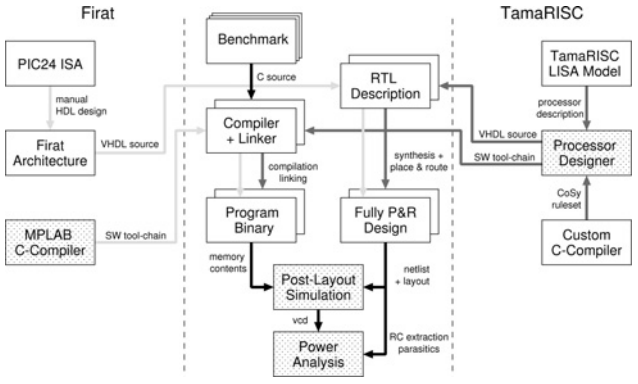

Fig. 3 *System evaluation and implementation flow*

Table 1 Firat against TamaRISC: Required number of cycles and energy consumption for the single lead reference benchmarks at 1.2 V

	Single-core with Firat		Single-core with TamaRISC	
	No of cycles	Energy of the core, μJ	No. of cycles	Energy of the core, μJ
CS	114 k	3.4	89.8 k	1.7
DWT	1.85 M	55.5	2.11 M	36.3
ECG2	348.6 k	10.8	304.3 k	4.9

verification of the LISA model simulation against the behavioural simulation of the generated HDL code.

4.2 Processing core selection

To compare the power against performance trade-offs between the processing cores (Firat and TamaRISC), we have built two different single-core architectures with Firat and TamaRISC. Table 1 shows the required number of clock cycles and energy consumption of the cores to execute the single-lead reference benchmarks for the single-core architectures with Firat and TamaRISC. As seen from Table 1, TamaRISC requires comparable number of clock cycles with Firat to execute the benchmarks, slightly higher number of cycles for some applications (i.e. DWT) and slightly lower number of cycles for some other (i.e. CS and ECG2). Moreover, at 1.2 V, Firat consumes 3.4 μJ, 55.5 μJ and 10.8 μJ whereas TamaRISC requires 1.7, 36.3 and 4.9 μJ to execute the single-lead CS, DWT and ECG2 benchmarks, respectively. TamaRISC achieves up to 54% energy saving with respect to Firat when executing the reference benchmarks. This advantage is achieved because of the reduced instruction set, as well as the efficient decoding of instructions and addressing modes in the TamaRISC architecture. TamaRISC is more energy efficient than Firat for our targeted application groups, thus hereafter we only use TamaRISC as the processing core in the PUs of the processing platforms.

TamaRISC also achieves higher energy efficiency as compared to the other state-of-the-art cores, developed for biomedical signal processing. On average, TamaRISC consumes only 12.1 pJ/Ops at 1.0 V while executing our reference benchmarks. For the same supply voltage level (1.0 V), yet a 130-nm process, Kwong *et al.* [21] report 47 pJ/cycle energy consumption for their 16-bit core where the number of clock cycle per instruction is higher than one. In another work, Ickes *et al.* [22] introduce a 32-bit core implemented in 65 nm, and the energy consumption of the core [22] is estimated for 1.0 V between 19.7 pJ/Ops and 27.0 pJ/Ops. Compared to these state-of-the-art processing cores, our customised core consumes less energy per operations notably because of its simple architecture as well as reduced instruction set, as explained in Section 3.1.2.

4.3 Processing platform architectures comparison

4.3.1 Design point exploration: Figs. 4*a* and *b* show the power consumption of the single- and multi-core architectures optimised with different clock constraints for workloads lighter than 50 MOps/s. The power values are normalised to the one of the design optimised for the highest maximum clock frequency. For this experiment, each design is supplied by the minimum voltage level

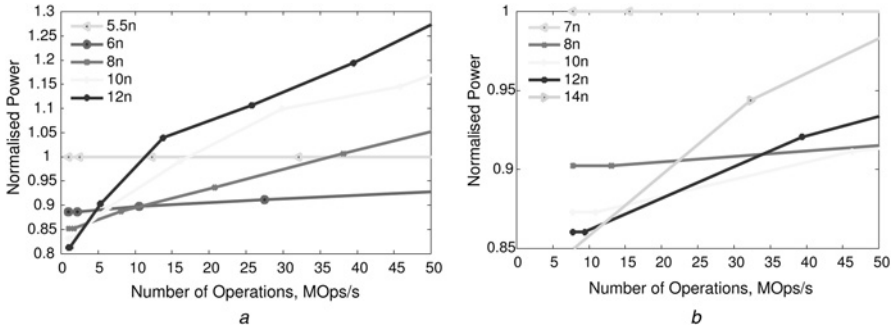

Fig. 4 *Power consumptions for various clock constraints*

a Single-core design
b Multi-core design

required for the respective throughputs. The single- and multi-core architectures operate up to 5.5 ns and 7.1 ns clock periods, respectively when optimised for maximum speed. This difference is due to the D-Xbar, leading almost to 1.6 ns additional delay in the longest delay path of the multi-core architecture. However, the targeted application groups do not require such high clock frequency, thus the delay penalty because of the D-Xbar does not raise any vital timing issue.

As shown in Figs. 4*a* and *b*, 6- and 10-ns clock constraints provide an energy efficient design point for the single- and multi-core designs, respectively. The single-core design optimised with 6-ns clock constraint consumes less power than the other single-core designs for workloads higher than 11 MOps/s, and consumes only slightly higher power for the workloads lighter than 11 MOps/s. Similarly, the multi-core design optimised with 10-ns clock constraint consumes less power than the other multi-core designs for the workloads higher than 22 MOps/s, and consumes only slightly higher power for the workloads lighter than 22 MOps/s. To obtain the respective minimum power solutions for the performance range of interest we optimised the single-core and multi-core designs with clock constraints of 6 and 10 ns, respectively.

The occupied silicon area of the single- and multi-core design is given in Table 2. As expected, the total area of PUs in the multi-core design is almost eight times the area of the PU in the single-core design. However, the total area of the multi-core design is only 1.72 times of the total area of the single-core design as the DM is responsible for most of this area.

Table 2 Area results of the architectures (1 GE = 3.136 μm^2)

	Single-core, kGE	Multi-core, kGE
total	644.2	1111.3
PUs	66.4	513.3
DM	576.7	576.7
D-Xbar	–	21.3

Table 3 Dynamic power distributions at 8 MOps/s and 1.2 V

	Single-core, mW	Multi-core, mW
total	0.76	0.70
PUs	0.59	0.55
DM	0.07	0.07
D-Xbar	–	0.03
clock tree	0.05	0.04

4.3.2 Experimental results with benchmark applications: The single-core processes the eight-lead ECG signals sequentially by using 1403 cycles, 32969 cycles and 4754 cycles on average per sample for CS, DWT and ECG2 benchmarks, respectively. On the other hand, the multi-core architecture processes eight-lead ECG signals in parallel (one lead per core) and requires 196 cycles, 4395 cycles and 637 cycles on average per sample for CS, DWT and ECG2 benchmarks, respectively. When accounting for the 8-times parallel processing, these correspond to a penalty between 6.6 and 11.8% penalty in terms of execution time because of stall-cycles compared to the corresponding number of cycles required for a single lead in the single-core architecture. We always adjust the clock frequency of the single- and multi-core design to correspond to processing requirement. In particular, Table 3 shows the distribution of the power consumption of single- and multi-core designs running at 8 MOps/s at 1.2 V. The total power consumption of the DM and IMs for the architectures are equal since the power consumption of a memory bank is proportional of the total number of access times. However, as seen from the table the PU in the single-core architecture consumes more power than the total power consumption of the PUs in the multi-core architecture. This is due to the power consumption of the processing cores. The total power consumption of the processing cores in the multi-core design is less than the corresponding one in the single-core design since the single-core design optimised for a higher clock frequency as explained in Section 4.3.1. As seen from Table 3, the overhead of the D-Xbar in terms of power consumption is insignificant, only 4.3% of the entire multi-core design. At the nominal voltage, the multi-core design consumes 27.9 μW leakage power whereas the leakage power consumption of the single-core design is 14.9 μW. This difference is mainly because of the individual MI banks for each PUs.

At the nominal voltage (1.2 V) the single-core approach achieves 167 MOps/s whereas the multi-core approach can accomplish up to 750 MOps/s. As shown in Fig. 5, for a given computational workload requirement, the multi-core can operate at significantly lower supply voltages as compared to the single-core architecture. However, the multi-core architecture operates at near to the transistor threshold voltage for 10.9 MOps/s workload, and do not benefit from voltage scaling for workload requirements lower than 10.9 MOps/s, since the voltage scaling is limited to the transistor threshold voltage level to avoid performance variability and functional failure issues. On the

doi: 10.1049/iet-cds.2012.0011

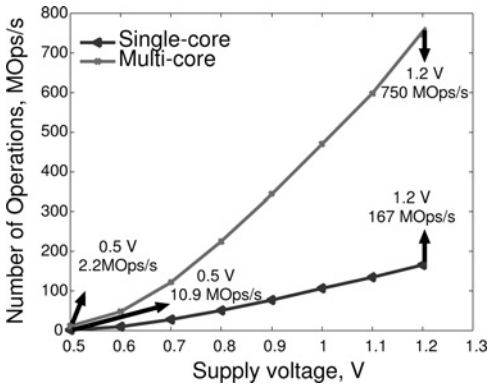

Fig. 5 *Single-core and multi-core designs: number of operations for various supply voltages*

other hand, the single-core benefits from voltage scaling until 2.2 MOps/s workload requirement.

Fig. 6*a* shows the total power consumption of the single- and multi-core design for various workload requirements. As can be seen from the figure, the multi-core approach is the only viable solution for workloads between 167 MOps/s and 750 MOps/s. Moreover, when the workload requirement is between 1.7 MOps/s and 167 MOps/s, the multi-core is more energy efficient than the single-core design, because the multi-core design can meet the workload requirements at a lower operating voltage compared to the single-core design (c.f. Fig. 5). In particular, to meet a high workload requirement (167 MOps/s) the single-core design operates at 1.2 V and consumes 15.8 mW, whereas the multi-core design operates at 0.75 V and consumes only 6.0 mW. Thus, the multi-core solution consumes 62% less power than the single-core design. Even though both designs are supplied around the transistor threshold voltage level (c.f. Fig. 5) at 2.2 MOps/s, the multi-core design still consumes less power then the single-core design because of its lower dynamic power consumption (c.f. Table 3). However, if the required workload is light (lower than 1.7 MOps/s) the single-core design consumes less power than the multi-core design, because of its lower leakage power consumption compared to the multi-core design. The power saving in the single-core design with respect to the multi-core design maximises when the designs only leak, 46% power saving.

Fig. 6*a* also shows the power consumptions of the architectures for our reference benchmarks with an ECG sampling rate of between 125 Hz and 1 kHz. Moreover, Fig. 6*b* shows the power savings in the multi-core design with respect to the single-core design for the benchmarks with different ECG sampling rate requirements. The CS benchmark workload requirement ranges from 175 kOps/s to 1.4 MOps/s, thus the single-core approach is the better solution in terms of power consumption. More specifically, the multi-core design consumes 1.6 and 37% more power than the single-core design for the CS benchmark with ECG sampling rate of 1 kHz and 125 Hz, respectively. The ECG2 benchmark requires a workload between 594.2 kOps/s to 4.75 MOps/s, thus the multi-core design consumes 11.9% more power than the single core design for 125 Hz ECG sampling rate, whereas the multi-core achieves 26.4% power saving for 1 kHz of ECG sampling rate. However, the DWT benchmark requires heavier workloads than the ECG2 and CS benchmarks, the DWT workload requirement ranges from 4.12 to 32.97 MOps/s, and thus the multi-core approach is a better solution. More precisely, the multi-core design accomplishes 23.6%and 41.4% power savings with respect to the single-core design for 125 Hz and 1 kHz sampling rate, respectively.

Another interesting point is the comparison between dynamic and leakage power consumptions in the two designs. Fig. 7 shows the dynamic and leakage power consumptions of the logics and the memories, including both IM and DM, for various workload requirements for the single-core and multi-core designs. As shown in the figures, the memories dynamic power consumption becomes comparable with the leakage power consumptions of the memories when the workload is 250 and 450 kOps/s for the single-core and the multi-core designs, respectively.

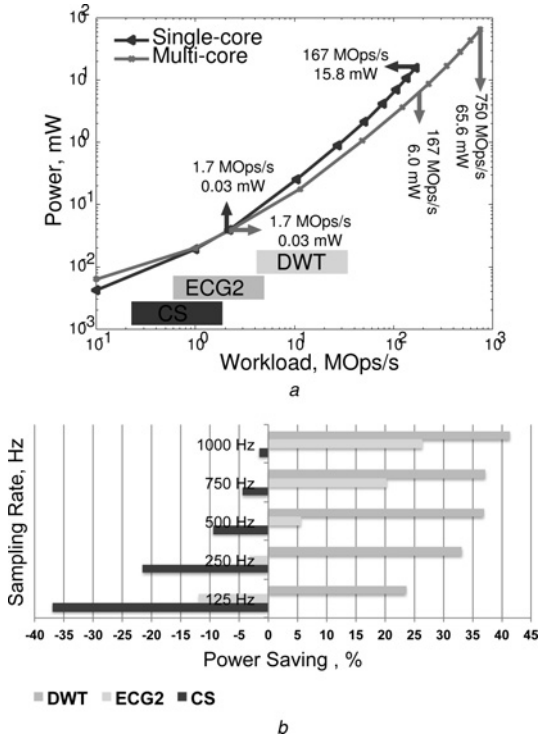

Fig. 6 *Power consumptions of the architectures and power savings in the multi-core design*

a Total power consumptions for various workloads
b Power savings in the multi-core architecture with respect to the single-core architecture

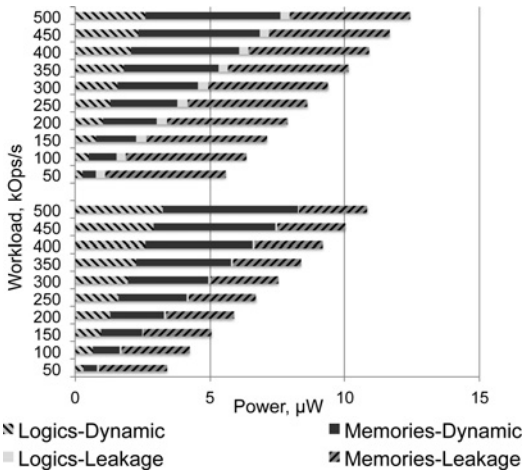

Fig. 7 *Leakage and dynamic power consumption comparison for various workload requirements*

As expected, the leakage power consumptions of the memories in the multi-core design becomes comparable with the memories dynamic power consumption at a higher workload, because the total memory leakage power is higher in the multi-core design. Furthermore, the overall leakage and dynamic power consumptions become comparable around 200 kOps/s for the single-core design while around 350 kOps/s for the multi-core design.

5 Conclusion

Health monitoring systems require energy-efficient processing platforms because of their limited power resources as well as long operational times. Online biomedical signal processing on such systems involves relatively low complexity and highly parallel computations on a low-rate physiological data, which creates the opportunity of low voltage operations as well as parallel processing. In this paper, to address the energy efficiency and data throughput requirements for biomedical signal processing on health monitoring systems, we have proposed: (i) an ULP processing core with a minimal instruction set; (ii) a single-core processor architecture; and (iii) a multi-core processor architecture with a DM shared through a crossbar interconnect. Our results have shown that an ISA with only several required instructions leads to a significant energy saving for biomedical signal analysis as compared to a well established, extensive ISA (in our case up to 54% energy saving compared to PIC24 ISA). We have also explored the power against performance trade-offs between the single- and multi-core architectures, including near threshold voltage computing, for different target workloads of online biomedical signal analysis. Our results have shown that the multi-core approach consumes 62% less power than the single-core approach for high biosignal computation workloads (i.e. 167 MOps/s). Moreover, as opposed to the common belief – single-core approaches are more energy efficient than multi-core ones for ULP domain since required workloads are typically light and can be handled effectively in single-core architectures – we have shown that a multi-core architecture, with a multi-bank DM shared by an interconnect between cores and DM, is a promising solution also in ULP parallel processing domain (in our case it achieves higher energy efficiency compared to the single-core approach for workloads as light as 1.7 MOps/s). This is because our multi-core solution neither requires memories with large number of read/write ports (using multi-port memories reason significant memory area density, and thus high leakage dissipation) nor a higher clock frequency compared to the rest of the circuit (over clocking reasons complexity and energy efficiency issues such as need of voltage level shifters and complex clock tree scheme). However, a multi-core architecture is still penalised because of its leakage power consumption at extremely light workloads, where the power consumption is dominated by leakage (in our case 46% more power consuming compared to the single-core approach). To alleviate this issue, as a future work, our proposed multi-core processing platform will include configurability such as turning of processing cores, and memories to reduce the leakage overhead when workloads are extremely light. The issues with low voltage operations such as process variability, functional failure, operating point temperature dependency issues occurring mainly at sub-threshold voltages are not examined in this paper; however, these issues will be also a subject of our future work.

6 Acknowledgments

This work was partially supported by the Swiss Confederation under the Nano-Tera.ch NTF Project BioCS-Node. The authors thank the Swiss NSF for their support under the project number PP002-119052. They also acknowledge the support of the ENIAC under the project JTI-END.

7 References

1 World Health Organization: 'Cardiovascular diseases', available at http://www.who.int/cardiovascular_diseases, accessed May 2012
2 Yang, G.Z.: 'Body sensor networks' (Springer-Verlag, London, UK, 2006)
3 Mamaghanian, H., Khaled, N., Atienza, D., Vandergheynst, P.: 'Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes', *IEEE Trans. Biomed. Eng.*, 2011, **58**, (9), pp. 2456–2466
4 Hanson, M.A.: 'Body area sensor networks: challenges and opportunities', *IEEE Comput.*, 2009, **42**, (1), pp. 58–65
5 Powell, H.C., Barth, A.T., Lach, J.: 'Dynamic voltage-frequency scaling in body area sensor networks using COTS components', *BodyNets*, 2009, **15**, pp. 1–8
6 Hanson, S., Zhai, B., Seok, M., *et al.*: 'Exploring variability and performance in a sub-200 mV processor', *IEEE J. Solid-State Circuits'*, 2008, **43**, (4), pp. 881–891
7 Zhai, B., Nazhandali, L., Olson, J., *et al.*: 'A 2.60 pJ/Inst subthreshold sensor processor for optimal energy efficiency'. Symp. on VLSI Circuits Digest of Technical. Papers, 2006, pp. 154–155
8 Wang, A., Chandrakasan, A.: 'A 180 mV FFT processor using sub-threshold circuit techniques'. IEEE Int. Solid-State Circuits Conf. on Digest of Technical Papers, 2004, pp. 292–529
9 Dreslinski, R.G., Wieckowski, M., Blaauw, D., Sylvester, D., Mudge, T.'Near-threshold computing: reclaiming moore's law through energy efficient integrated circuits', *IEEE Proc.*, 2010, **98**, (2), pp. 253–266
10 Chen, G., Fojtik, M., Kim, D., *et al.*: 'Millimeter-scale nearly perpetual sensor system with stacked battery and solar cells'. Solid-State Circuits Conf. on Digest of Technical Papers, 2010, pp. 288–289
11 Hanson, S., Seok, M., Lin, Y.-S., *et al.*: 'A low-voltage processor for sensing applications with picowatt standby mode', *IEEE J. Solid-State Circuits*, 2009, **44**, (4), pp. 1145–1155
12 Dreslinski, R.G., Zhai, B., Mudge, T., Blaauw, D., Sylvester, D.: 'An energy efficient parallel architecture using near threshold operation'. PACT, 2007, pp. 175–188
13 Pu, Y., Pineda de Gyvez, J., Corporaal, H., Ha, Y.: 'An ultra-low-energy multi-standard JPEG co-processor in 65 nm cmos with sub/near threshold supply voltage', *IEEE J. Solid-State Circuits*, 2010, **45**, (3), pp. 668–680
14 Microchip Technology: available at www.microchip.com/en_us/family/16bit/architecture/PIC24H.html, accessed May 2012
15 Jocke, S.C., Bolus, J.F., Wooters, S.N., *et al.*: 'A 2.6-mW Sub-threshold Mixed-signal ECG SoC'. Symp. on VLSI Circuits, 2009, pp. 60–61
16 Rincon, F., Recas, J., Khaled, N., Atienza, D.: 'Development and evaluation of multilead wavelet-based ECG delineation algorithms for embedded wireless sensor nodes', *IEEE Trans. Inf. Technol. Biomed.*, 2011, **15**, (6), pp. 854–863
17 Sun, Y., Chan, K., Krishnan, S.M.: 'ECG signal conditioning by morphological filtering', *Comput. Biol. Med.'*, 2002, **32**, (6), pp. 465–479
18 Rahimi, A., Loi, I., Kakoee, M.R., Benini, L.: 'A fully-synthesizable single-cycle interconnection network for Shared-L1 processor clusters'. DATE, 2011, pp. 1–6
19 Microchip Technology 'Development environment and compilers', available at http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGEnodeId=1406dDocName=en019469part=SW007002, accessed May 2012
20 Synopsys: available at http://www.synopsys.com/Systems/BlockDesign/processorDev, accessed May 2012
21 Kwong, J., Chandrakasan, A.P.: 'An energy-efficient biomedical signal processing platform', *IEEE J. Solid-State Circuits*, 2011, **46**, (7), pp. 1742–1753
22 Ickes, N., Sinangil, Y., Pappalardo, F., Guidetti, E., Chandrakasan, A.P.: 'A 10 pJ/cycle ultra-low-voltage 32-bit microprocessor system-on-chip'. ESSCIRC, 2011, pp. 159–162