Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Ultra-Low-Energy DSP Processor Design for Many-Core Parallel Applications

*B. Soltani-Farani [1], H. Dorosti [2], M. E. Salehi [1, *], S.M. Fakhraie [1]*

[1] *School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran.*

[2] *Computer Engineering Department, Shahid Rajaee Teacher Training University, Tehran, Iran.*

Article Info

Article History:
Received 04 February 2019
Revised 31 May 2019
Accepted 01 December 2019

Keywords:
DSP processor
Ultra-low-energy
sub-threshold circuits
Many-core architectures

[*] Corresponding Author's Email Address: mersali@ut.ac.ir

Extended Abstract

Background and Objectives: Digital signal processors are widely used in energy constrained applications in which battery lifetime is a critical concern. Accordingly, designing ultra-low-energy processors is a major concern. In this work and in the first step, we propose a sub-threshold DSP processor.
Methods: As our baseline architecture, we use a modified version of an existing ultra-low-power general purpose processor. Afterwards, we make some modifications to add new instructions to the processor instruction set for better adapting to signal processing applications. In the second step, employing sub-threshold cores in many-core architectures, we use the proposed processor as simple basic cores in a many-core architecture.
Results: In comparison with the baseline architecture, these modifications reduce the program memory size about 42% in average. In addition, data memory accesses are reduced about 60% in average, and more than 90% speed-up is achieved. According to the improvements in total execution time (93%) and power consumption (27%), the total consumed energy is reduced about 95% in average with at most 2.6% area overhead and without increasing the process variation effects on processor specifications.
Conclusion: The results show that for parallel applications, such as FFT in LTE standard, exploiting sub-threshold processors in a many-core architecture not only can satisfy the required performance, but also reduce the power consumption about 50% or even more.

Introduction

Energy constrained applications such as cell phones, wireless networks, and RFID tags are widely used in recent years. Power/energy consumption, performance, and reliability are the main concerns in designing such systems [1], [2], [3]. Moreover, the use of deep sub-micron technologies has made designs even more challenging due to increased variations in process parameters such as gate oxide thickness, channel length, and threshold voltage [2], [4].

Designing processor architecture in the sub-threshold region, where the supply voltage is less than the threshold voltage, can efficiently reduce energy and power consumption [6], [5].

Sub-threshold design provides energy harvesting capability for long-term applications such as health-care signal processing and monitoring wireless networks [2].

However, sub-threshold computing degrades performance and escalates variation problems [6]. Therefore, it is necessary to design custom architectures and utilize alternate techniques to accompany sub-threshold design. Application specific processors are potential candidates to meet the mentioned requirements.

To customize an architecture for a specific application, parameters such as computation width,

pipeline depth, ISA definition, memory organization design, and addressing modes need to be considered [7]. Among these parameters, ISA definition is more effective and will affect the others. The complexity of custom instructions can change the execution timing and memory access rate of an application. Hence, custom instructions are useful in improving processor performance and reducing power/energy consumption [7]-[9].

Most of the low-power processors introduced in the literature have been designed to work in the super-threshold or near-threshold regions [10]-[15]. Moreover, processors introduced for the sub-threshold region [8],[16]-[18] are not optimized for computationally-intensive signal processing applications. The growing demand for DSP processors in energy constrained applications motivates our work, where we have simplified an existing ultra-low power general purpose processor [18] according to Pollack's rule [19]. We have added an extra unit corresponding to a new instruction added to the processor's ISA for accelerating signal processing applications. In this paper, we investigate the effects of the added custom instruction on power and energy consumption using specific signal processing applications. The results show that although adding the extra unit increases processor area, which is equivalent to more static energy dissipated in the sub-threshold region, the added custom instruction enables a smaller program memory footprint which in turn significantly reduces power and energy consumption. The proposed processor, to the best of our knowledge, is the first sub-threshold DSP processor; nevertheless, we do not claim that it has the best architecture.

Another usage of ultra-low-energy processors can be found in many-core architectures. These systems may exploit hundreds or thousands of small cores in parallel [20]. According to Pollack's rule [19], the performance of a core is proportional to the square root of the core's complexity (area). Although simplifying the core structure decreases its performance, it can increase the power efficiency. With many ultra-low-power cores one can make, through parallelism, a many-core architecture and then achieve a desirable throughput. According to the Amdahl's law, the serial part of a code will limit the speed-up. Thus parallel applications rationalizes the usage of many-core architectures. An integral part of high-speed wireless networks are Orthogonal Frequency-Division Multiplexing (OFDM) systems, in which signal processing applications such as Fast Fourier Transform (FFT) and Finite Impulse Response (FIR) filtering are exploited as the main operations [21], [22].

In this work, we proposed a novel processor architecture to achieve required performance with higher energy efficiency for target energy-constrained applications such as IoT. The proposed architecture is a general purpose processor with customized instructions for Digital Signal Processing (DSP) domain to improve performance and energy efficiency. Using the proposed DSP processor, one can aggregate many simplified very-low-power cores in parallel to construct very high-speed systems such as OFDM, filter bank, and etc.

The remainder of the paper is organized as follows: Section II reviews related work from the literature and discusses previously designed processors. In Section III the proposed processor architecture is presented along with a detailed description of the added extra unit and the related custom instructions. Section IV introduces our experimental setup and analyzes the obtained results, using specific signal processing applications. The usage of the proposed processor in many-core architectures and its experimental results are described in Section V. Finally, the paper is concluded in Section VI.

Related Works

In this section, we review the various ultra-low power processors introduced in the literature [8], [10]-[18]. For each processor, its main specifications and the techniques exploited by designers to reduce power and energy consumption are discussed.

In [10], a DSP processor known as uAMPS (micro Adaptive Multi-domain Power aware Sensors) has been designed, based on a load-store 16-bit RISC architecture. The processor is designed to work in the near-threshold region and contains an instruction cache and extra units including Multiply-ACcumulate (MAC) unit and custom hardware accelerator cores for FIR filters and FFT operations. Moreover, power gating is used to reduce power and energy consumption. The results show that uAMPS works at 4 MHz with a 0.45V power supply and consumes 10 pJ/Instruction in 90-nm technology.

Kelly *et al.*[23] have proposed a Sensor Network Asynchronous Processor (SNAP). To achieve a low power design, the SNAP designers employed asynchronous circuits by which dynamic power is reduced due to lower switching activities. SNAP/LE, the low energy version of SNAP that is proposed in [11], is based on an event-driven 16-bit RISC processor. Their single-issue processor core executes instructions in order and uses two separate on-chip memory banks for instructions and data. Their results show that in 180-nm technology, SNAP/LE works with 28 MIPS speed and consumes 24pJ energy per instruction at 0.6V. While SNAP/LE has a parallel datapath, another extension of SNAP called BitSNAP [12], has a bit-serial datapath leading to lower area and lower leakage power. In addition, the designers used compression techniques on data to improve the processor's performance. According to their report, BitSNAP consumes 17 pJ/Instruction in 180-nm

technology at 0.6V and provides 6 MIPS execution speed.

Another ultra-low power processor, called Smart Dust [13], has been devised based on a load-store RISC architecture, Harvard architecture for memories, and single cycle per instruction but with no pipelining in the datapath. Furthermore, techniques such as component-level clock gating and guarded ALU inputs were exploited to reduce power. Smart Dust was designed in 250-nm technology to work with a 1.0V power supply at 500 kHz and consumes 12 pJ/Instruction.

Hempstead et al. [14], [15] have used event-driven processing and hardware accelerators to improve power consumption and performance. The accelerators are designed for specific tasks which are common in many wireless sensor network applications. There is an event processing unit investigating events, and based on the event's type, determines which accelerator must be activated. Therefore, the tasks are offloaded to the accelerators and the microcontroller can be power gated. This technique reduces dynamic and leakage power dramatically. Thus Hempstead's processor consumes only 0.44 pJ per equivalent instruction at 12.5 MHz working frequency using a 0.55V power supply in 130-nm technology.

In [16], Nazhandali *et al*. proposed an energy-efficient sub-threshold processor, called Subliminal. The architecture is an accumulator-based CISC architecture with single operand instructions. Nazhandali et al. investigated 21 different architectures by changing the pipeline depth, datapath width, memory organization (von-Neumann vs. Harvard), and use of an explicit register file. The best performing architecture with lower energy consumption and acceptable performance was found using the pareto-optimal curves. The investigated processor specifications are summarized in Table 1. Subliminal is designed in 130-nm technology, works at 182 kHz, and consumes 1.38 pJ/Instruction at 235mV. In [17] and [8] Zhai *et al*. also analyzed process variation in sub-threshold circuits and showed that dynamic frequency scaling is more important than dynamic voltage scaling. In addition, Zhai *et al*. have designed a robust SRAM memory that made the Subliminal processor work correctly from 0.2V to 1.2V. The minimum energy consumption of the processor is 2.6 pJ/Instruction at 0.36V while working at 833 kHz.

The second generation of Subliminal is described in [18]. Subliminal2 is a load-store RISC architecture with two-operand 12-bit instructions. In Subliminal [16], the authors have shown that a smaller code size (with more complex control logic) is more energy-efficient than using a simpler control logic (with larger code size) and therefore, have selected CISC architecture with variable instruction lengths. However, in Subliminal2, Nazhandali *et al*. [18] proposed a compact 12-bit instruction set architecture with complicated addressing modes satisfying both requirements: simple control logic and also a dense code. Table 1 shows the Subliminal2 specifications consuming the lowest energy per instruction (0.6pJ/Instruction) among the sub-threshold processors and works at 142 kHz and 0.2V in 130-nm technology. The low energy processor architectures of sub- and near-threshold processors are summarized in Table 1. Also there are other harware accelerators working for energy-constrained applications such as [24]-[27] which are customized to the application domain to achieve higher computation speed with lower energy consumption. These works are not listed in our candidate baseline architecture because it is not fair to compare a general purpose processor with custom hardware accelerators.

Table 1: Specifications of some ultra-low-power processors

Processor Specifications	Subliminal	Subliminal2	uAMPS
ISA	Accumulator CISC architecture with single operand instruction	load-store RISC architecture with 2-operand instruction	load-store RISC architecture with 3-operand instruction
Data Width (bits)	8	8	16
Instruction Length	32/16/8	12	16
CPI	1	1	1
Pipeline Stages	3	3	3
Registers	4 GPR(32 bits) + 4 pointer register (16bits)	8 GPR(8 bits)	8 GPR + 8 special purpose registers
Memory Architecture	Von-neumann	Harvard	Von-neumann
Program Memory	512*4 bits	128*12 bits	60KB
Data Memory		128*8 bits	
Out-of-Order Execution	no	yes	no
Branch Speculation	no	yes	no
Multiplier	no	no	Yes
MAC	no	no	yes

Processor Architecture

In this work, we use a modified version of the Subliminal2 architecture as the basis of our DSP processor architecture. The main features of Subliminal2

processor are summarized in Table I and our simplified implementation of the Subliminal2 (hereafter called SSL2) datapath is shown in Fig. 1. Here, we give a brief description of the Subliminal2 ISA as is used in our work and shown in Fig. 2.

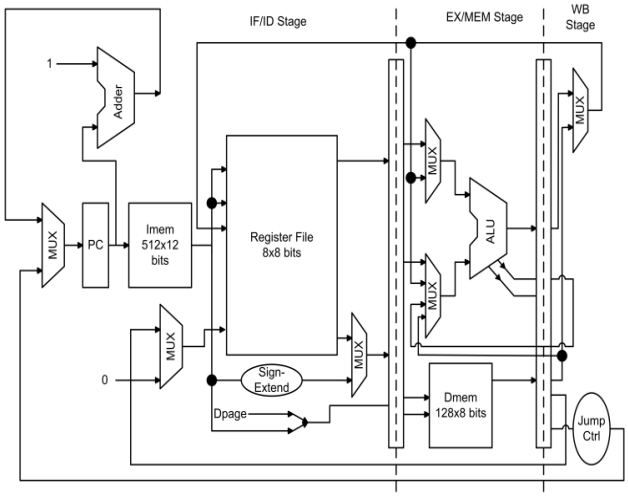


Fig. 1: SSL2 datapath, our simplified implementation of Subliminal2.

As the figure shows, in ALU instructions a P/U bit specifies a preserve/update mode. In the preserve mode the result of the ALU operation is saved in R0, while in the update mode operand A is updated by the ALU result. Using this, Subliminal2 benefits from the advantages of two-operand instructions, namely, smaller instruction length, and three-operand instructions in which source operand will not be overwritten. Note that operand B (OPB) includes five cases:

- MEM: The operand is one of the memory entries which can be accessed through direct memory access.
- REG: It denotes one of the 8 registers in the register file.
- PTR: In SSL2, PTR is used to denote one of the registers (R0 to R3) and it can access memory through indirect access.
- IMM: This is a two-bit immediate value.
- C: Shows a carry operand.

This feature shows that micro-operations are included in the ISA, which leads to smaller code size. For example, to add a register to a memory operand, instead of two instructions: LOAD followed by ADD, only one instruction is used.

The instruction and data memory size in Subliminal2 are 128x12 bits and 128x8 bits, respectively. In SSL2 we changed the instruction memory size to 512x12 bits, adapted to the size of most DSP codes running on. Instruction and data memories are divided into 32 and 8 pages, respectively. Each page has 16 entries. To access data memory, we need 3 bits to indicate the page, and 4 bits to access one of the 16 entries in that page. The PAGE instruction represents the page number, and then with the mode MEM of OPB, as shown in Fig. 2, we can access data memory directly.

In Subliminal2, the JDST part in JUMP instructions indicates the lowest 6 bits of jump target address. In SSL2 we modified the JUMP instruction as follows: For all jump conditions, except CALL, the jump target address will be added to the PC address, thereby we can access -32 to 31 instructions away from the current instruction using 6 bits; however for CALL condition, JDST indicates the lowest 6 bits of jump target address and the other three bits are determined by the PAGE instruction. The details of the instructions are presented in [18]. In Subliminal2, indirect memory access takes two clock cycles. If the load instruction is followed by a dependent ALU operation, then the implemented out-of-order execution feature will monitor the next instruction and if it is independent of ALU result, it will be executed before the ALU operation. We designed the instruction and data memory in the form of flip-flops as in [24]. Therefore, memory access takes only one cycle and there is no need to implement the out-of-order execution feature. Also in SSL2, unlike Subliminal2, the branch speculation scheme has not been implemented for sake of more simplicity. The instruction next to jump is always executed; therefore the compiler has been optimized to support this feature.

The main focus of our work is on designing an ultra-low power DSP processor. FFT operations and FIR filtering are frequently used in wireless networking and signal processing applications. Conventional FFT algorithm requires nlog n multiplications which makes it an important feature of any DSP processor. In SSL2, there is no multiply unit and no multiply instruction, and we need to carry out multiply operations using add instructions.

Here we propose the next version of SSL2 called SSL2Mult. In SSL2Mult we have added a multiply unit to the SSL2 datapath and a MULT instruction to its ISA. Figures. 5 and 3 show the SSL2Mult datapath and MULT instruction, respectively. The MULT instruction has two modes for 8-bit unsigned and signed multiplications whose result will be saved in a 16-bit register. In the first cycle, OPA will be overwritten by the lowest 8 bits of the result and in the next cycle, the highest 8 bits of the result are saved in OPB.

Moreover, as shown in Fig. 7, FIR filters use multiply-accumulate (MAC) operations.

Thus, as our next step we have added a MAC unit and a MAC instruction to SSL2. We call this processor SSL2MAC. The SSL2MAC datapath and MAC instruction are shown in Figs. 6 and 4, respectively.

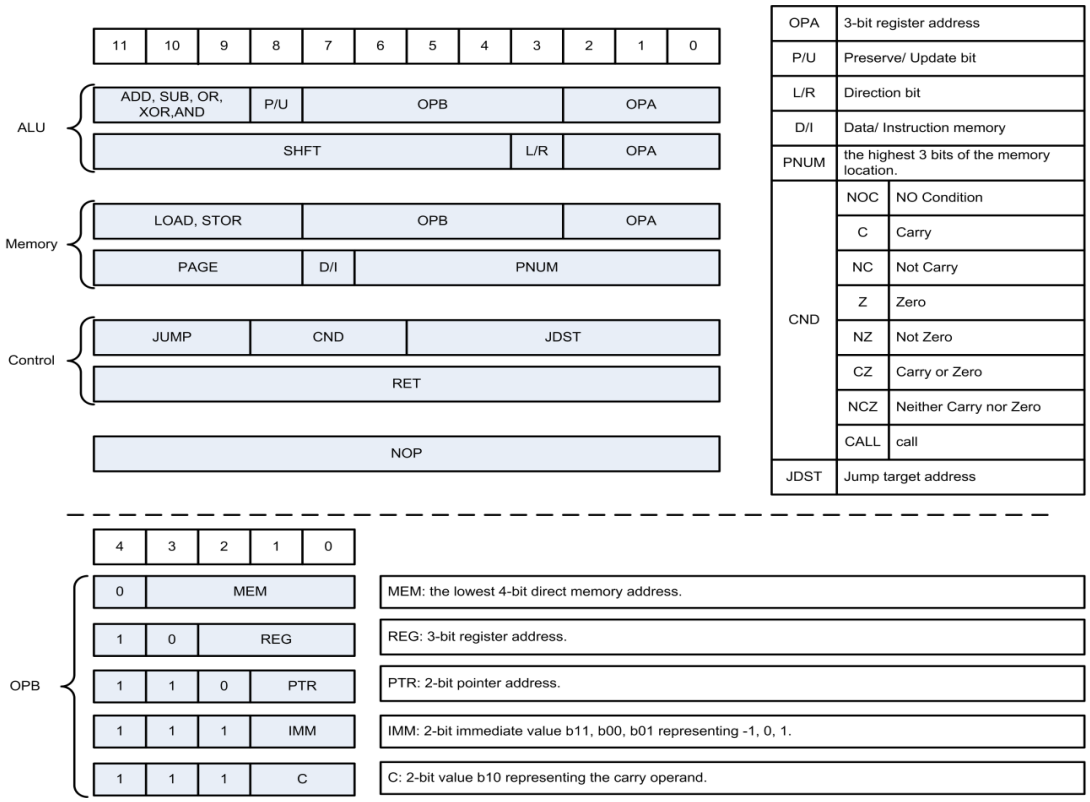


Fig. 2: SSL2 ISA encoding.

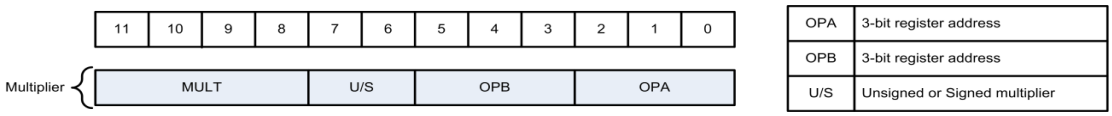


Fig. 3: The added MULT instruction toSSL2 ISA.

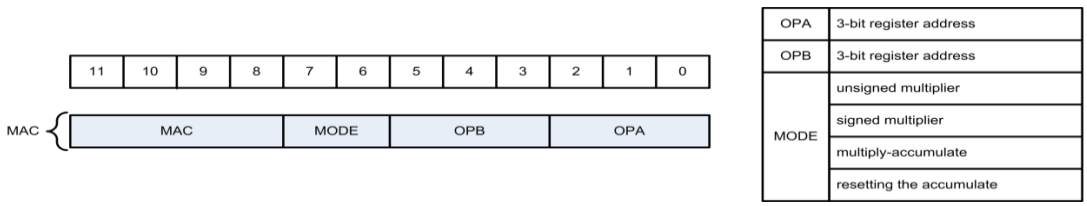


Fig. 4: The added MAC instruction toSSL2 ISA.

Table 2: Specifications of the proposed processors in comparison to subliminal2

Processor Specifications	Subliminal2	SSL2 (our implementation of subliminal2)	SSL2Mult	SSL2MAC
ISA	load-store RISC architecture with 2-operand instruction	load-store RISC architecture with 2-operand instruction	load-store RISC architecture with 2-operand instruction	load-store RISC architecture with 2-operand instruction
Data Width (bits)	8	8	8	8
Instruction Length	12	12	12	12
CPI	1	1	1	1
Pipeline Stages	3	3	3	3
Registers	8 GPR(8 bits)	8 GPR(8 bits)	8 GPR(8 bits)	8 GPR(8 bits)
Memory Architecture	Harvard	Harvard	Harvard	Harvard
Program Memory	128*12 bits	128*12 bits	128*12 bits	128*12 bits
Data Memory	128*8 bits	128*8 bits	128*8 bits	128*8 bits
Out-of-Order Execution	yes	Yes	yes	yes
Branch Speculation	yes	Yes	yes	yes
Multiplier	no	No	no	no
MAC	no	No	no	no

In MAC unit, the multiplier multiplies two 8-bit numbers and the result is added to a 24-bit accumulator. The MAC instruction has four modes: unsigned and signed multiplication, MAC operation, and accumulator reset. Table 2 summarizes the architectures of the proposed processors.

The multiply unit in SSL2Mult and the MAC unit in SSL2MAC increase the processor area, but on the other hand lead to smaller code size.

Furthermore, these units are power gated in applications that do not require them. We need to investigate the advantages and disadvantages of adding these extra units in sub-threshold circuits.

In the next section, we analyze the performance and energy results of the three discussed processors on two benchmarks: FFT and FIR.

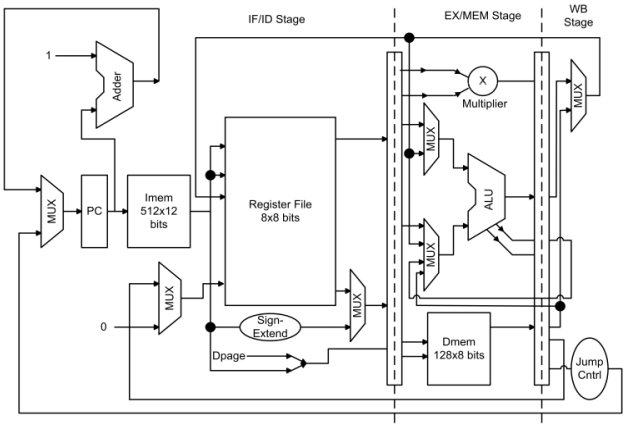


Fig. 5: SSL2Mult datapath.

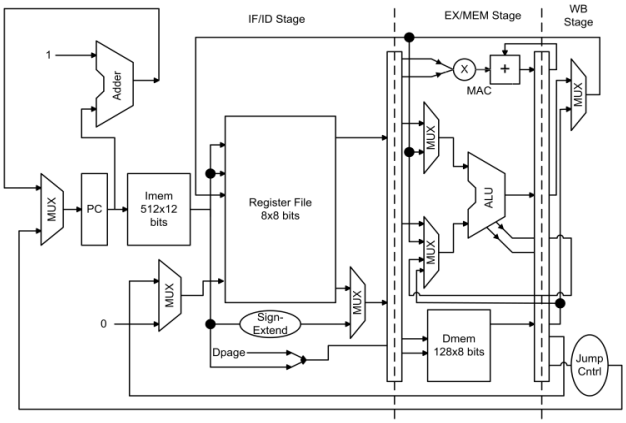


Fig. 6: SSL2MAC datapath.

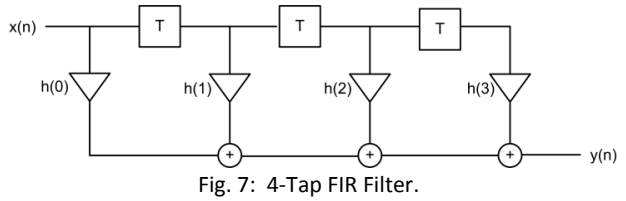


Fig. 7: 4-Tap FIR Filter.

Results and Discussion

Benchmark suits are small sized programs derived from real applications to evaluate the efficiency of processor architectures. SenseBench [29] and WiSeNBench [30] are available suits for wireless sensor networks and other energy-constrained applications. Aside from the benchmarks, well-defined efficiency metrics are essential for detailed analysis of processor performance. These metrics should be measurable and clear in order to be able to evaluate architecture efficiencies. Conventionally, energy per instruction and clock per instruction (CPI) plus clock frequency are used to describe energy consumption and processor performance, respectively [7]. However, using these parameters, we cannot provide effective comparison between RISC and CISC architectures. Therefore, as other essential metrics, we need to consider the total amount of energy consumed to execute the application (energy per data bundle), the total number of cycles to execute the application (clock cycles per data bundle), and the program size (memory footprint) for an appropriate and fair analysis [29]. Table 3 shows the composition of our benchmark applications used to compare the aforementioned efficiency metrics.

Table 3: Benchmark composition algorithms

Benchmark Application	Comments
FFT	16-point Complex-Valued Fast Fourier Transform
FIR	20-Tap FIR Digital Filter

All architecture configurations are synthesized using standard synthesis tools and custom libraries, designed and characterized with 25 basic cells in different supply voltages from 0.25V to 1V in 0.05V steps. Here we present the results of experiments on SSL2Mult and SSL2MAC in comparison to our baseline processor, SSL2.

A. Area

Table 4 shows the area results for all configurations obtained from synthesis tools with the same constraints. According to this table, about 1.5-2.6% area overhead has been added to the baseline design due to multiply and MAC units.

Table 4: Total cell area of different architecture configurations all with 512x12 bits memory

Architecture	SSL2	SSL2Mult	SSL2MAC
Area(µm2)	242228	244861	248631

B. Memory Footprint

The expected memory size of each processor

configuration differs for different benchmark applications due to ISA modifications. Table V shows the memory size categories with their corresponding memory reduction results. This table shows that the custom instruction, added to SSL2's ISA, affects the memory size (code size) only if it is used in the algorithm. For example, when FIR is executed on SSL2Mult or SSL2MAC, it needs less program memory in comparison to the baseline design. Note that for each benchmark the required program memory pages are activated and the unused pages are power gated.

According to Table 5, complex instructions reduce the data memory access (considering reads as well as writes) to 50% and 30% relative to that of SSL2 for both FFT and FIR applications, respectively. Table 5 shows that the total number of executed instructions for FFT and FIR benchmarks in both modified versions are less than that of the baseline version; therefore, the total number of clock cycles required to execute these benchmarks are reduced. As expected, these reductions in memory access reduces the memory access power consumption and total execution time.

Table 5: Memory size and number of instruction and data memory accesses for each benchmark on different configurations (unit = 12bits). The values for SSL2Mult and SSL2MAC are reported relative to those of SSL2.

Benchmark	FFT			FIR		
Architecture	SSL2 (base)	SSL2Mult (Relative to Base)	SSL2MAC (Relative to Base)	SSL2 (base)	SSL2Mult (Relative to Base)	SSL2MAC (Relative to Base)
Memory (unit)	240	0.73	0.73	112	0.57	0.43
Number of Instruction Memory Accesses	66157	0.08	0.08	47505	0.08	0.06
Number of Data Memory Accesses	5377	0.52	0.52	4324	0.32	0.29

C. *Performance*

Working frequency for different architectures vs supply voltages is shown in Fig. 9. This figure reveals that the working frequency is the same for the three configurations, that is, the critical path has remained the same.

As mentioned in the previous subsection, the total number of clock cycles needed to execute FFT and FIR benchmarks were reduced in SSL2Mult and SSL2MAC (See Fig. 8).

This fact, with no change in frequency, implies a reduction in total execution time and accordingly an improvement in performance. However, as shown in Table 6 the required clock cycles per single instruction (CPI) has increased due to complex instructions.

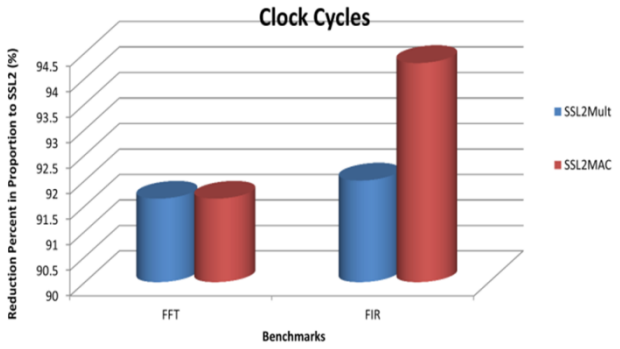


Fig. 8: Number of clock cycles reduction, in proportion to SSL2, for each benchmark running on different configurations.

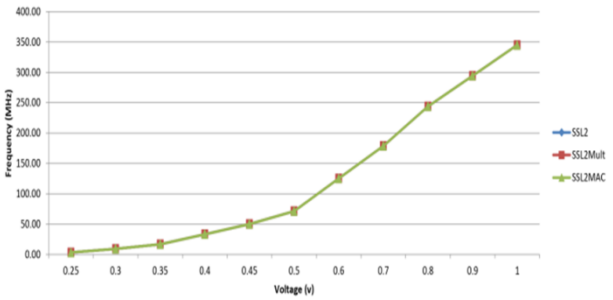


Fig. 9: Working frequency for different architectures vs supply voltage.

Table 6: Clock per instruction for each benchmark running on different configurations

Benchmark	SSL2	SSL2Mult	SSL2MAC
FFT	1.010	1.055	1.055
FIR	1.039	1.088	1.070

D. *Power and Energy Consumption*

Power consumption is estimated through simulation environment and captured VCD files in different supply voltages separately. Figures 10a and 10b depict the average dynamic and static power consumption, respectively, for benchmark applications vs. supply voltages. According to Fig. 10a, modified versions of the instruction set reduce the amount of dynamic power consumption due to memory access reduction (both for instruction and data). Core leakage has a direct relation with core area, increased with the additional units for

the complex instructions, and therefore can increase static power consumption. On the other hand, the added instruction leads to smaller program memory (smaller total area) and then lower static power consumption (see Fig. 10b). The total power consumption depicted in Fig. 10c shows that adding complex custom instructions has improved power efficiency of processor architecture. Figure 10c also shows decreasing of the supply voltage scales down power consumption for all architectures.

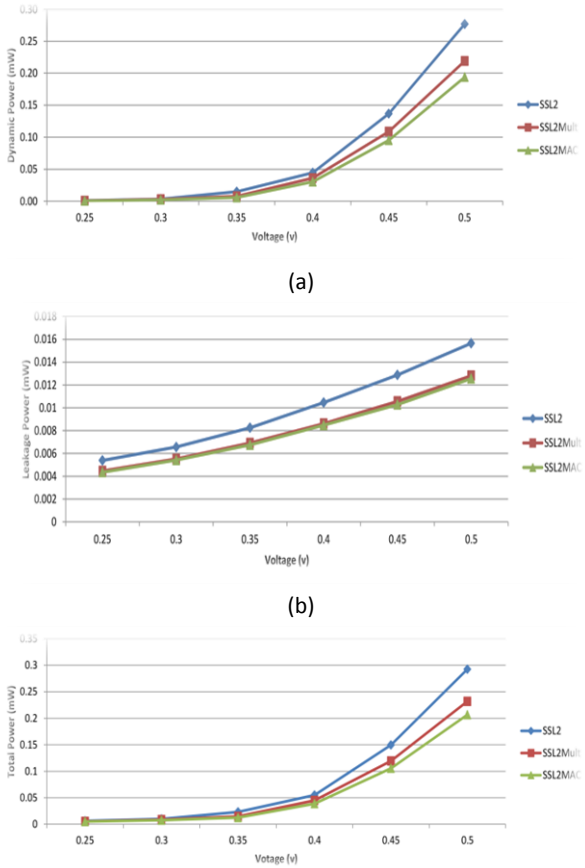


Fig. 10: Power consumption for the three architecture configurations: (a) Average dynamic power consumption; (b) Average static power consumption; (c) Average total power consumption.

The Power-Delay product of a processor is known as energy consumption and indicates the lifetime of battery-powered applications. Dynamic energy consumption depends on dynamic power consumption and on total execution time. Custom instructions improve energy efficiency in FFT and FIR applications due to execution time reduction. Figure 11a shows the average dynamic energy consumption for each configuration vs. supply voltage. Static energy consumption depends on total area and total execution time of applications. Custom instructions improve both total execution time and area, leading to lower static energy consumption for FFT and FIR applications. As shown in Fig. 11b, decreasing the supply voltage increases leakage energy consumption because of delay dominant effect. The Total energy consumption of the processor is the sum of static and dynamic ingredients and is depicted in Fig. 11c for all architectures.

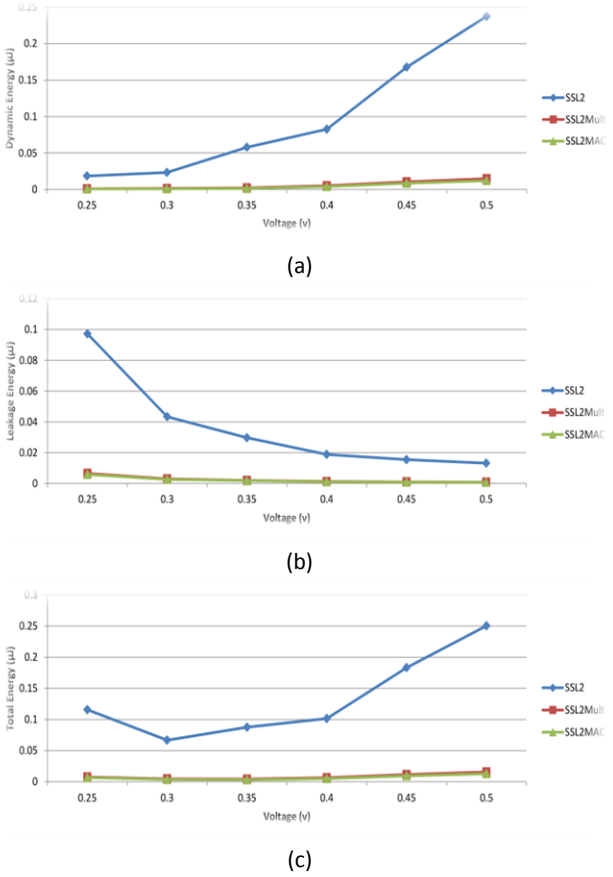


Fig. 11: Energy consumption for the three architecture configurations: (a) Average dynamic energy consumption; (b) Average static energy consumption; (c) Average total energy consumption.

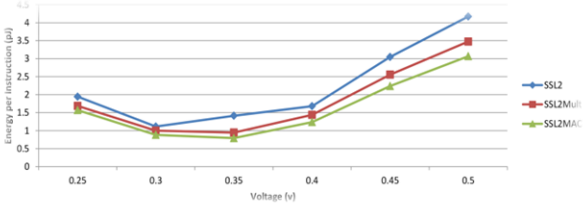


Fig. 11: Energy per instruction consumption for different architectures vs supply voltages.

This curve has an optimal point with maximum energy efficiency known as minimum energy point which is located at 0.3V for SSL2 and 0.35V for SSL2Mult and SSL2MAC. On the right side of this point, dynamic energy is dominant, and on the left side the static part dominates. According to the above results, adding complex instructions, which are commonly used in

applications, reduces energy consumption and improves performance and energy efficiency in sub-threshold region. The average consumed energy per instruction is depicted in Fig. 12.

E. Variations Analysis

Besides performance and power/energy consumption of processor, variability is another important factor needing to be considered at design time. Variation-aware design needs careful analysis based on statistical static timing analysis (SSTA) to prevent timing error occurrence due to process variations [27], [28]. Equation (1) depicts the mean (μ) and standard deviation (σ) of a normal distribution of a critical path:

$$\mu_{critical-path} = \sum_{i=1}^{n} \mu_i, \qquad \sigma^2_{critical-path} = \sum_{i=1}^{n} \sigma_i^2 \tag{1}$$

where n denotes the number of gates, and

$$\mu_{critical-path} + 3\sigma_{critical-path} = delay_{critical-path} \tag{2}$$

We note that the traditional worst-case analysis assumption led to $\sigma_{critical-path} = \sum_{i=1}^{n} \sigma_i$, yielding a much higher $\sigma^2_{critical-path}$.

Analyzing the effects of process variations on processor specifications is accomplished via Monte-Carlo simulations of the critical paths in SPICE environment for different supply voltages. Critical path delays are randomly assigned and simulated for 1000 iterations and the histogram is fitted to normal distribution to get mean (μ) and standard deviation (σ). According to Figures 13 and 14 these parameters are similar for expected architecture configurations with negligible difference. Figure 15 depicts the normal distribution of critical path for different architectures at 0.45V power supply. The results show that adding complex instructions does not change the critical path delay and so does not worsen the variability status of design around minimum working frequency.

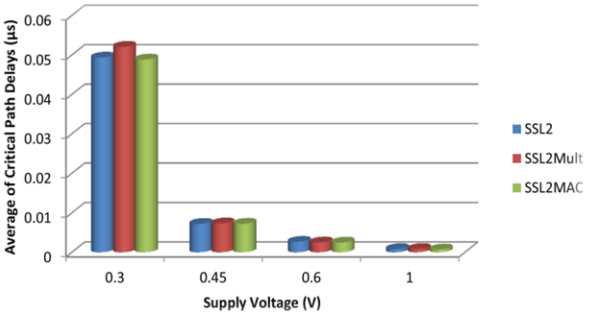


Fig. 2: Average of critical path delays in normal distribution.

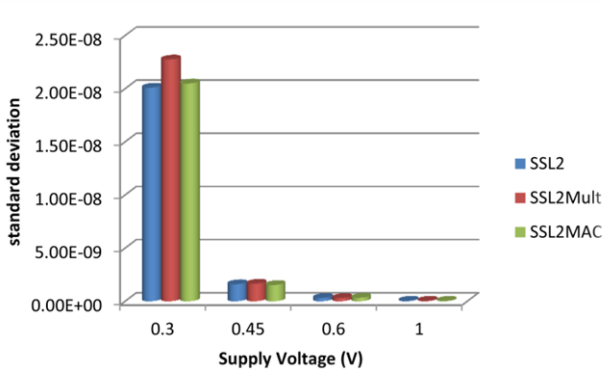


Fig. 13: Standard deviation of critical path delays in normal distribution.

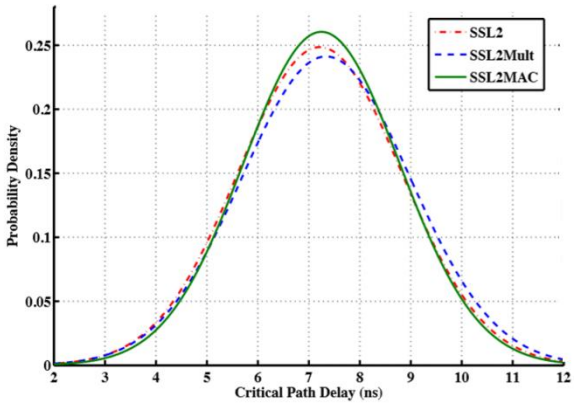


Fig. 14: Normal distribution of critical path for different configurations at 0.45V power supply.

SSL2 (Mult) in Many-Core Architectures

The idea behind many-core architectures is that hundreds to thousands small and simple cores run in parallel to achieve the desired throughput. In this section we propose a new idea: Employing ultra-low power processors as simple cores in many-core architectures. As discussed before, to reduce static energy consumption, designers try to design ultra-low power processors in small area. So the simplicity of these processors made them as good candidates for the cores in many-core architectures.

To show the efficiency of usage of our proposed processor in many-core architectures we select the FFT application in LTE standard [33]. According to this standard, a 2048-point FFT should be calculated in less than 66.6μs [34]. Thus, the desired throughput is 15.02 Ksymbol/sec and each symbol equals to a 2048-point signal. Therefore the many-core structure should provide this throughput. In addition to the existing parallelism in many-core architectures, if a pipeline structure is used, and meanwhile each pipeline stage has delay less than 66.6μs, then the desired throughput will be achieved.

Due to the data memory size of SSL2 and SSL2Mult,

these processors can execute at most a 32-point complex-valued FFT. Therefore, to execute a 2048-point FFT, the SSL2 or SSL2Mult cores should run in parallel. There are 37 ways that a 2048-point FFT can be constructed by 2,4,8,16, and 32-point FFTs. For example, assume an eleven-stage pipeline, in each stage 1024 cores running in parallel and each core executes a 2-point FFT. After 11 stages, a 2048-point FFT wil be calculated. Another example is a three-stage pipeline, in each of the first and the second stage there are 256 cores, each core executes an 8-point FFT and in the third stage, there are 64 cores each of which executes a 32-point FFT. After these three stages a 2048-point FFT will be calculated. We use this notation to represent a way: nS-mC-k + n'S-m'C-k' + ..., where n denotes the number of identical Stages and m is the number of Cores in each stage and each core executes a k-ponit FFT. The next different stages will be shown after plus symbol with the same definition. With this notation, 11S-1024C-2 represents the first mentioned example and the second one is represented by 2S-256C-8 + 1S-64C-32. Another example is shown in Fig. 16 and the corresponding notation is 1S-1024C-2 + 5S-512C-4.

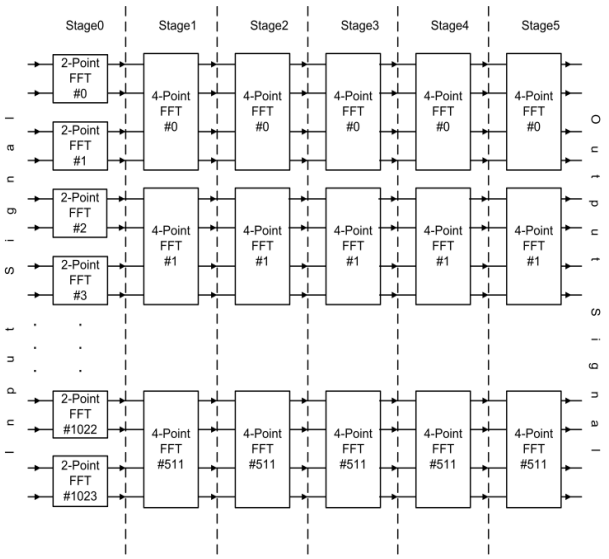


Fig. 15: One of the 37 ways that a 2048-point FFT can be constructed by 2,4,8,16, and 32-point FFTs. The representing notation for this way is 1S-1024C-2 + 5S-512C-4.

Each of the 37 ways has been investigated at 0.25V to 1.0V voltages. Among the investigated cases, those providing the desired throughput (15.02 Ksymbol/sec) are valid for us. For each aforementioned way among the valid cases, the one consuming the minimum power is called a "good case" and has been selected for that way. If we use SSL2 cores in a many-core architecture, among those 37 ways there will be only 6 good cases. On the other hand, if we use SSL2Mult cores instead, there will be 37 good cases implying that each of the 37 ways provides the desired throughput. Figures 17a and 17b show the experimental results of using SSL2 and SSL2Mult cores in a many-core architecture for a 2048-point FFT calculation in LTE standard. The vertical axis in the figure shows the required number of cores, and the power consumption has been labeled in horizontal axis. In Fig. 17a depicting the results of employing SSL2 cores, all 6 good cases providing the desired throughput have been shown. The points possessing the same throughout have been shown by a same marker. Figure 17b shows the results of employing SSL2Mult cores. For simplicity, only the good cases consuming less than 10mW power have been shown. There are 15 points in Fig. 17b. According to restrictions, namely, power or area (number of cores), the desired architecture can be selected. Among the investigated cases, those consuming the minimum power and minimum number of cores are reported in Table 7. According to this table, if the power consumption is the main concern, then the usage of SSL2Mult cores may reduce power consumption and number of cores down to 94.3% and 13.6%, respectively; and also if the main concern is the required number of cores (area), then employing SSL2Mult cores causes 84% and 85.7% reduction in power consumption and number of cores, respectively.

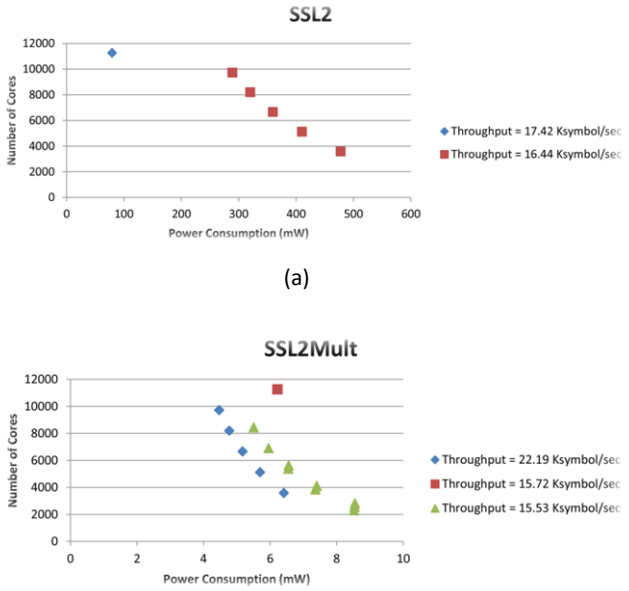


Fig. 16: Power consumption and needed number of cores for employing SSL2(Mult) in many-core architectures to calculate a 2048-point complex-valued FFT: (a) The cores are SSL2; (b) The cores are SSL2Mult.

So far, we have shown that employing SSL2Mult cores rather than SSL2 in a many-core architecture can improve power and energy efficiency. Now we show the results of employing SSL2Mult cores in a many-core

architecture and also the other implementations in literature, that are designed to calculate a 2048-point FFT in LTE standard, in order to compare the power consumption and to propose a new idea, namely, sub-threshold many-core architecture for future researches. Note that we are not proposing a many-core architecture for calculating FFT in LTE standard. Rather, we are suggesting many-core architecture with sub-threshold cores for parallel applications such as FFTs. We also show that, for example, the sub-threshold many-core architecture can reduce power consumption to about 50% or even more to calculate a 2048-point FFT in LTE standard; see Table 8.

Table7:The experimental results of usage SSL2(Mult) in a many-core architecture to calculate a 2048-point complex-valued FFT

Goal	Cores	Representing Notation of Way	Number of Cores	Supply Voltage (V)	Throughput (Ksymbol/sec)	Power Consumption (mW)
Minimum Power Consumption	SSL2	11S-1024C-2	11264	0.4	17.42	79.35
	SSL2Mult	9S-1024C-2 + 1S-512C-4	9728	0.35	22.19	4.47
Minimum Number of Cores	SSL2	1S-1024C-2 + 5S-512C-4	3584	0.6	16.43	477.68
	SSL2Mult	1S-256C-8 + 2S-128C-16	512	0.6	22.39	76.36

Table8: The Power Consumption Comparison

	Patyk[33]	Peng[34]	Yang[35]	Our Many-Core Architecture
Technology (nm)	130	180	65	90
Supply Voltage (V)	1.1	1.8	0.45	0.35
FFT Size	1024	2048	2048	2048
Throughput (Msymbol/sec)	140	35	20	0.022
Power Consumption (mW)	29.8	11.29	8.55	4.47
Energy Consumption (μJ)	1.09	2.65	0.103	2.01
Energy Consumption (nJ) / FFT Size	1.07	1.29	0.050	0.98

Conclusion

Due to advances in technology and the importance of power consumption, specially in energy constrained applications, it is necessary to make appropriate considerations in design time. In this work, we compared available ultra-low-power processors and selected the one with lower energy per instruction as our base and made some modifications in processor pipeline to achieve lower power consumption with higher throughput. Afterwards, based on our benchmark composition, we made two modified versions by adding two different custom complex instructions (Multiply and Multiply-Accumulate) as new candidates. These modifications increased the core area in one hand and also yielded smaller memory footprint for the code. In future deep sub-micron technologies, especially in sub-threshold region, due to the dominance of leakage, area efficiency implies energy efficiency. Higher density programs and smaller memory sizes improve performance and energy consumption simultaneously. Summary and analysis of results show that adding complex custom instructions to processor architecture and keeping the frequency constant will improve performance as well as energy efficiency of design without intensifying variation effects.

The results show that adding complex instructions in SSL2MAC reduce the program memory size and data memory access about 42% and 60% in average, respectively, relative to the baseline processor. The improvements in total execution time and power consumption, leads to 95% lower energy consumption in average.

Moreover, in the proposed processor, SSL2MAC, which, to the best of our knowledge, is the first sub-threshold DSP processor, the minimum energy per instruction has been improved about 29% in average compared to the baseline processor.

These improvements in power and energy consumption in conjunction with our proposed processors' simplicity led us to employ them as simple cores in a many-core architecture. Therefore, in addition to consuming less power, the performance reduction in sub-threshold circuits will be compensated. The results show that to calculate a 2048-point complex-valued FFT in LTE standard, the usage of SSL2Mult cores instead of SSL2 in a many-core architecture made 94% and 14% reduction in power consumption and number of cores, respectively.

Author Contributions

B. Soltani, H. Dorosti, and M. E. Salehi designed the experiments with the guidance of S. M. Fakhraie. B. Soltani collected the data with the help of H. Dorosti through proper simulations, and data analysis is carried out by B. Soltani and H. Dorosti. Finally B. Soltani,H. Dorosti, and M. E. Salehi interpreted the results and wrote the manuscript.

Acknowledgment

The authors would like to thank Alireza. Mazraee for developing sub-threshold library cells.

Conflict of Interest

Authors declare that there is no conflict of interests regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancy have been completely observed by the authors.

Abbreviations

DSP	Digital Signal Processor
ISA	Insctruction Set Architecture
FFT	Fast Fourier Transform
LTE	Long-Term Evolution
FIR	Finite Impulse Response
RFID	Radio Frequency Identification
OFDM	Orthogonal Frequency-Division Multiplexing
ALU	Arithmetic and Logic Unit
MAC	Multiply and Accumulate
RISC	Reduced Instruction Set Computer
CISC	Complex Instruction Set Computer
CPI	Clock Per Instruction
MIPS	Million Instruction Per Second
MULT	Multiply
VCD	Value Change Dump

References

[1] H. Iwai, “echnology Roadmap for 22nm and beyond (invited paper),” Microelectronic Engineering, 86(79): 1520-1528, 2009.

[2] ISSCC Trends, 2013.

[3] H. Nejatollahi. M. E. Salehi, “Voltage scaling and dark silicon in symmetric multicore processors,” Journal of Supercomputing, 71(10): 3958-3973, 2015.

[4] International Technology Roadmap for Semiconductors, 2013.

[5] H. Dorosti. A. Teymouri. S. M. Fakhraie. M. E. Salehi, “Ultralow-energy variation-aware design: adder architecture study,” IEEE transaction on Very Large Scale Integration (TVLSI), 24(3): 1165-1168, 2016.

[6] A. Wang, B. H. Calhoun, A. P. Chandrakasan, Design for Ultra Low-Power Systems, New York: Springer, 2006.

[7] J. L. Hennessy, D. A. Patterson, Computer Architecture: A Quantitative Approach, 4th Edition, San Francisco: Morgan Kaufmann, 2006.

[8] B. Zhai,S. Pant, L. Nazhandali, S. Hanson, J. Olson, A. Reeves, M. Minuth, R. Helfand, T. Austin, D. Sylvester, D. Blaauw, “Energy-efficient subthreshold processor design, IEEE Trans. On Very Large Scale Integration (VLSI) Systems, 17(8): 1127-1137, 2009.

[9] J. Constantin, A. Dogan, O. Andersson, P. Meinerzhagen, J. Rodrigues, D. Atienza, A. Burg, "TamaRISC-CS: An ultra-low-power application-specific processor for compressed sensing," in Proc. 2012 IEEE/IFIP 20th International Conference on VLSI and System-on-Chip (VLSI-SoC): 159-164, 2012.

[10] N. Ickes, D. Finchelstein, A. Chandrakasan, " A 10-pj/instruction, 4-MIPS micropower DSP for sensor applications," in Proc. 2008 IEEE Asian Solid-State Circuits Conference: 289-292., 2008.

[11] V. Ekanayake, C. Kelly, R. Manohar, “An ultra low-power processor for sensor networks,” SIGARCH Comput. Archit. News 32(5): 27-36, 2004.

[12] V. Ekanayake, I. Kelly, C., R. Manohar, "BitSNAP: Dynamic significance compression for a low-energy sensor network asynchronous processor," in Proc. IEEE International Symposium on Asynchronous Circuits and Systems: 144-154, 2005.

[13] B. Warneke, K. Pister, "An utra-low energy microcontroller for Smart Dust wireless sensor networks," in Proc. IEEE International Solid-State Circuits Conference: 316-317, 2004.

[14] M. Hempstead, N. Tripathi, P. Mauro, G.-Y. Wei, D. Brooks, "An ultra low power system architecture for sensor network applications," in Proc. IEEE International Symposium on Computer Architecture, ISCA: 208-219, 2005.

[15] M. Hempstead, D. Brooks, G.-Y. Wei, “An accelerator-based wireless sensor network processor in 130 nm CMOS,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems. 1(2): 193-202, 2011.

[16] L. Nazhandali, B. Zhai, J. Olson, A. Reeves, M. Minuth, R. Helf, S. Pant, T. Austin, D. Blaauw, “Energy optimization of subthreshold-voltage sensor network processors,” SIGARCH Comput. Archit. News, 33(2): 197-207, 2005.

[17] B. Zhai, L. Nazhandali, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, D. Blaauw, T. Austin, "A 2.60pj/inst subthreshold sensor processor for optimal energy efficiency," in Proc. 2006 Symposium on VLSI Circuits, 2006. Digest of Technical Papers.: 154-155, 2006.

[18] L. Nazhandali, M. Minuth, B. Zhai, J. Olson, T. Austin, D. Blaauw, "A second-generation sensor network processor with application-driven memory optimizations and out-of-order execution," in Proc. 2005 International Conference on Compilers, Architectures and Synthesis for Embedded Systems, CASES ’05, ACM: 249-256, 2015.

[19] F. J. Pollack, "New microarchitecture challenges in the coming generations of CMOS process technologies," in Proc. 1999 Annual ACM/IEEE International Symposium on Microarchitecture, MICRO 32, IEEE Computer Society, 1999.

[20] S. Borkar, " Thousand core chips: A technology perspective," in Proc. 2007 Annual Design Automation Conference, DAC ’07, ACM: 746-749, 2007.

[21] M. Aliasgari. A. Abbasfar. S. Fakhraie. “Coding techniques to mitigate out-of-band radiation in high data rate OFDM-based cognitive radios,” Computers & Electrical Engineering 39(2): 373-385, 2013.

[22] A. Salari, S. Fakhraie, A. Abbasfar, “Algorithm and FPGA implementation of interpolation-based soft output MMSE MIMO detector for 3GPP LTE,” IET Communications 8(4), 21(3): 492-499, 2014.

[23] I. Kelly, C., V. Ekanayake, R. Manohar, “SNAP: A sensor-network asynchronous processor,” in Proc. International Symposium on Asynchronous Circuits and Systems, ASYNC ’03, IEEE Computer Society: 24-33, 2003.

[24] Y. Pu, G. Samson, C. Shi, D. Park, K. Easton, R. Beraha, J. Hadi, M. Lin, E. Arvelo, J. Fatehi, J. Kumar, M. Derkalousdian, P. Aghera, A. Newham, H. Sheraji, K. Chatha, R. McLaren, V. Ganesan, S.

Namasivayam, D. Butterfield, R. Shenoy, R. Attar, " Blackghost: “An ultra-low-power all-in-one 28nm CMOS SoC for Internet-of-Things,” in Proc. IEEE Symposium in Low-Power and High-Speed Chips (COOL CHIPS): 1-3, 2017.

[25] H. Cherupalli, H. Duwe, W. Ye, R. Kumar, J. Sartori, (2018).Bespoke Processors for Applications with Ultra-Low Area and Power Constraints IEEE Micro, 38(3): 32-39, 2018.

[26] S. Yin, P. Ouyang, J. Yang, T. Lu, X. Li, L. Liu, S. Wei, "An Ultra-High Energy-Efficient Reconfigurable Processor for Deep Neural Networks with Binary/Ternary Weights in 28nm CMOS," in Proc. IEEE Symposium on VLSI Circuits: 37-38, 2018.

[27] M. Wang, N. Yu, W. Ma, Q. Sheng, W. Zhang, Z. Huang, " An Ultra Low-power Processor with Dynamic Regfile Configuration," in Proc. 2018 IEEE International Conference on Solid-State and Integrated Circuits Technology (ICSICT): 1-3, 2018.

[28] P. Meinerzhagen, S. Sherazi, A. Burg, J. Rodrigues, “Benchmarking of standard-cell based memories in the sub-vt domain in 65-nm CMOS technology,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems 1(2): 173-182, 2011.

[29] L. Nazhandali, M. Minuth, T. Austin, "Sensebench: toward an accurate evaluation of sensor network processors," in Proc. IEEE Workload Characterization Symposium: 197-203, 2005.

[30] S. Mysore, B. Agrawal, F. Chong, T. Sherwood, "Exploring the processor and ISA design for wireless sensor network applications," in Proc. 21st International Conference on VLSI Design (VLSID 2008): 59-64, 2008.

[31] A. Srivastava, D. Sylvester, D. Blaauw, Statistical Analysis and Optimization for VLSI: Timing and Power. New York: Springer, 2005.

[32] S. Sarangi, B. Greskamp, R. Teodorescu, J. Nakano, A. Tiwari, J. Torrellas, VARIUS: A model of process variation and resulting timing errors for microarchitects. IEEE Trans. On Semiconductor Manufacturing. 21(1): 3-13, 2008.

[33] LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (3GPP TS 36.211 version 12.5.0 Release 12), ETSI, 2015.

[34] LTE in a Nutshell: The Physical Layer, Telesystem Innovations, 2010.

[35] T. Patyk, D. Guevorkian, T. Pitkanen, P. Jaaskelainen, J. Takala, "Low-power application-specific FFT processor for LTE applications," in Proc. IEEE International Conf. on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS): 28-32, 2013.

[36] S. Y. Peng, K. T. Shr, C. M. Chen, Y. H. Huang, "Energy-efficient 128~2048/1536-point FFT processor with resource block mapping for 3GPP-LTE system," in Proc. 2010 IEEE International Conference on Green Circuits and Systems: 14-17, 2010.

[37] C. H. Yang, T. H. Yu, D. Markovic, “Power and area minimization of reconfigurable FFT processors: A 3GPP-LTE example,” IEEE Journal of Solid-State Circuits. 47(3): 757-768, 2012.

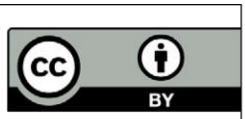
Biographies

Bushra Soltani-Farani was born in Tehran, Iran. She received her B.Sc. and M.Sc. degrees in Computer Architecture from the Department of Electrical and Computer Engineering at University of Tehran, Iran, in 2011 and 2014, respectively. Her main research interests are Computer Architecture, Digital Signal Processing, and Low-Power and Ultra-Low-Power Processor Design. She is currently embedded software at DESY, Hamburg Area, Germany.

Hamed Dorosti was born in Khoy, in 1986 and received the B.S. and M.S. degree in computer engineering from University of Tehran, Tehran, Iran, in 2009 and 2011, respectively. He received his Ph.D. in computer engineering (computer architecture) from University of Tehran in 2017.Since 2009, he was member of Silicon Intelligence and VLSI Signal Processing Lab., University of Tehran and co-operated in low-power ASIP project from 2010 to 2012. His research interest includes VLSI design, digital signal processing, adaptive timing error detection and correction and low-power high-throughput/performance processor architecture design considering static and dynamic variations. He is now an assistant professor of Shahid Rajaee University.

Mostafa Ersali Salehi Nasab was born in Kerman, Iran, in 1978. He received the B.Sc. degree in computer engineering from University of Tehran, Tehran, Iran, and the M.Sc. degree in computer architecture from University of Amirkabir, Tehran, Iran, in 2001 and 2003, respectively. He has received his Ph.D. degree in school of Electrical and Computer Engineering, University of Tehran, Tehran, Iran in 2010. From 2004 to 2008, he was a senior digital designer working on ASIC design projects with SINA Microelectronics Inc., Technology Park of University of Tehran, Tehran, Iran. He is now an Assistant Professor in University of Tehran. His research interests include novel techniques for high performance, low-power, and fault-tolerant embedded system design.

Sied Mehdi Fakhraie received the M.Sc. degree in electronics from the University of Tehran, Tehran, Iran, in 1989, and the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 1995. He was a Professor with the School of Electrical and Computer Engineering, University of Tehran. He was the Director of the Silicon Intelligence and VLSI Signal Processing Laboratory, the Director of Electrical and Electronics Engineering, and the Director of Computer Hardware Engineering with the School of Electrical and Computer Engineering, University of Tehran. He was a Visiting Professor with the University of Toronto, in 1998, 1999, and 2000, where he was involved in efficient implementation of artificial neural networks. He was with Valence Semiconductor Inc., Irvine, CA, USA, from 2000 to 2003. He was in Dubai, United Arab Emirates, and Markham, Canada Offices of Valence as the Director of Application Specified Integrated Circuit (ASIC) and System-on-a-Chip Design, and the Technical Leader of integrated broadband gateway and family radio system baseband processors. He was involved in many industrial integrated circuit design projects, including design of network processors and home gateway access devices, DSL modems, pagers, and digital signal processors for personal and mobile communication devices. He has co-authored a book entitled VLSI-Compatible Implementations for Artificial Neural Networks (Boston, MA, USA: Kluwer, 1997). He has authored or co-authored over 230 reviewed conference and journal papers. His last research interests include system design and ASIC implementation of integrated systems, novel techniques for high-speed digital circuit design, and system-integration and efficient VLSI implementation of intelligent systems. He passed away on December 7, 2014.



How to cite this paper:
B. Soltani-Farani, H. Dorosti, M. E. Salehi, S.M. Fakhraie, "Ultra-Low-Energy DSP Processor Design for Many-Core Parallel Applications," Journal of Electrical and Computer Engineering Innovations, 8(1): 71-84, 2020.

DOI: 10.22061/JECEI.2020.6969.350

URL: http://jecei.sru.ac.ir/article_1424.html