■ 1019

Energy Consumption Saving in Embedded Microprocessors Using Hardware Accelerators

Gian Carlo Cardarilli, Luca Di Nunzio*, Rocco Fazzolari, Marco Re, Francesca Silvestri, Sergio Spanò
University of Rome Tor Vergata, Via del Politecnico 1, 00133 Rome, Italy
*Corresponding author, e-mail: cardarilli, di.nunzio, fazzolari, re, f.silvestrig, spanò@ing.uniroma2.it

Abstract
This paper deals with the reduction of power consumption in embedded microprocessors. Computing power and energy efficiency are becoming the main challenges for embedded system applications. This is, in particular, the caseof wearable systems. When the power supply is provided by batteries, an important requirement for these systems is the long service life. This work investigates a method for the reduction of microprocessor energy consumption, based on the use of hardware accelerators. Their use allows to reduce the execution time and to decrease the clock frequency, so reducing the power consumption. In order to provide experimental results, authors analyze a case of study in the field of wearable devices for the processing of ECG signals. The experimental results show that the use of hardware accelerator significantly reduces the power consumption.

Keywords: low power architectures, embedded systems, hardware accelerator

1. Introduction

Energy consumption in electronic systems is one of the most discussed issues in the last years. This aspect has been dealt by researchers at different abstraction levels from the physical to the application one [1]-[3] and for different technologies such as IoT and cellular equipment [4]-[6] by exploiting dedicated efficient algorithms [7]-[8]. Also for embedded systems power consumption represent a crucial aspect. These systems are often used under operating conditions where power supply cannot be provided by the electrical grid. This is the case of medical wearable devices [9]. The development of advanced wearable systems makes possible to track patient health conditions outside hospital setting for several days [10]. These devices avoid extra costs for hospitals and uncomfortable distress for patients. On the other hand, wearable devices often need to operate powered by batteries for a very long time. Frequently, such batteries cannot be easily replaced. In this scenario, power consumption is one of the most important issues in order to guarantee a long service life. Thanks to their low cost, their flexibility and their easy programmability (that impacts on the applications develop time), embedded microprocessors represent the main choice in embedded systems. There are three power dissipation components in CMOS digital circuits and consequently in microprocessors [11]:
 a. Switching Power
 b. Short-Circuit Power
 c. Static Power.

Among these contributions, the switching power represents the main one [10] and it is defined in equation 1, where a is the switching activity, C is the switching capacitance, f is the clock frequency and Vdd the supply voltage.

$$P = a\ C\ f\ V^2_{dd} \tag{1}$$

The second contribution, the short-circuit power, is related to the short-circuit currents flowing through the MOS transistors in the gate at each switching. It is strongly dependent on the parameters present in equation 1 (switching activity, clock frequency, and supply voltage) [13], but it also depends on the design (the transistor ratios and the node waveforms). Finally, the static power depends on the leakage currents and it is related to the circuit design, the technology, and the supply voltage [12].

In the last few years, with the scaling of the device sizes and the supply voltage, microprocessor vendors provided devices with increased energy efficient [13]. In wearable devices, a typical embedded microprocessor application consists in the processing of biomedical signals coming from the ADC. In this scenario, in which real-time acquisition represents a crucial feature, the microprocessor must be able to process data in a time smaller than the ADC sample time. For this reason, the CPU clock frequency is usually much higher than the ADC sample rate. With reference to Figure 1, the computation time must be smaller than the ADC rate. During the computation time, the microprocessor requires an energy that in Figure 1 is represented by the area of the rectangle (for sake of simplicity, we assume that the power consumption in the computation time is constant). In order to reduce the energy consumption, the area of the rectangles must be reduced.
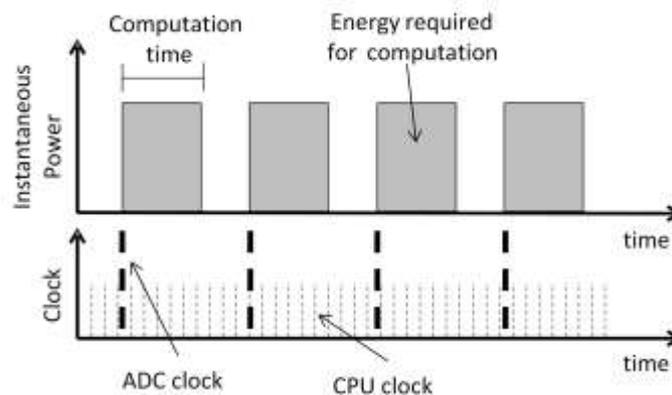
Figure 1. Energy consumption of embedded microprocessor in a typical application.

In this paper, authors address the issue of the energy consumption reduction in embedded microprocessors, using hardware accelerators [14]. The idea is to reduce the overall energy dissipation of the microprocessor, using the speed-up factor introduced by a suitable hardware accelerator. In fact, the speed increase allows reducing the processing time (corresponding to a reduction of the number of switching per input sample) and, in addition, to scale the clock frequency. Consequently, if the power dissipated in the accelerator is small, the overall power consumption is reduced. In order to provide experimental results, authors considered a case of study in wearable device field, a real-time algorithm for detection of QRS complexes in ECG signal. In this context, two different implementations of the algorithm were proposed in order to estimate the energy saving. In the first implementation, the algorithm was executed only by the microprocessor. In the second one, the algorithm was executed by a system composed of a microprocessor and a hardware accelerator. The paper is organized as follows: in section 2 the issue of the power consumption, in a system composed of a microprocessor and a hardware accelerator, is discussed. In section 3 the Pan and Tompkins algorithm is introduced and described. In section 4, details about the experimental setup are given. In section 5 results are provided, and finally, in section 6, conclusions are discussed.

2. Microprocessor and Power Consumption
The energy required by the microprocessor for executing an algorithm is provided in equation 2, where PPR is the mean dynamic power (that includes the switching and the shortcircuit contributions) dissipated inside the microprocessor, and T is the execution time.

$$E_{PR} = P_{PR} T \tag{2}$$

Coupling the microprocessor with a hardware accelerator, the energy required for the algorithm execution is shown in equation 3. The equation contains P_A, the mean dynamic power consumption of the hardware accelerator, and $\alpha = 1/S$, the reciprocal of the speed-up factor (S).

Using the accelerator, the execution lasts $T_A = \alpha\, T$. In the analysis, we suppose that in the idle interval, of length $T(1-\alpha)$, the system power consumption can be neglected. The term α cannot be equal to 0, because this value would imply an execution time equal to 0, and must be less than 1, because $\alpha = 1$ would imply no acceleration in the computation time. For these reasons $0 < \alpha < 1$.

$$E_{TOT} = (P_{PR} + P_A)\, T\, \alpha \tag{3}$$

In order to introduce power saving, we must have:

$$E_{TOT} < E_{PR} \tag{4}$$

Substituting the equation 3 in equation 4 we obtain equation 5.

$$(P_{PR} + P_A)\, T\, \alpha < P_{PR}\, T$$

$$\alpha < \frac{P_{PR}}{P_{PR} + P_A} \tag{5}$$

Defining $K = P_A / P_{PR}$ as the power ratio, we obtain:

$$\alpha < \frac{1}{1+K} \tag{6}$$

If the power consumption of the hardware accelerator is negligible with respect to the power consumption of the microprocessor, the power saving is obtained for any value of α. This is the case of Bit Manipulation Units (BMUs), Reconfigurable-Functional Units (RFUs) and, in general, of the hardware accelerators characterized by a reduced area occupation [15]-[21]. In this case, the energy saving is proportional to α. Alternatively, the power consumption can be lowered reducing the clock frequency. If the initial execution time T satisfies the time constraints, a hardware accelerator introducing a speedup factor S, can be used to reduce the clock frequency. It is possible to scale the clock frequency from f to a value $\tilde{f}$, such that execution time $T_A(\tilde{f})=T$. In this way, no speedup is obtained but the dynamic power, that is proportional to the clock frequency, is reduced. If we assume static power negligible with respect to the dynamic power, we obtain equation 7 and equation 8, where $\beta = \tilde{f}/f$ is the frequency scaling coefficient (tipically $\alpha = \beta$).

$$\beta\, (P_{PR} + P_A)\, T < P_{PR}\, T \tag{7}$$

$$\beta < \frac{1}{1+K} \tag{8}$$

In conclusion, we have two possibilities to reduce the energy consumption of a microprocessor using a hardware accelerator:
  a. Direct Energy reduction: reduction of the execution time and consequently, the energy required for the algorithm execution.
  b. Indirect Energy reduction: reduction of the power consumption decreasing the clock frequency of the system, leaving the execution time unaltered.

3. Microprocessor and Power Consumption A Case Of Study: The Pan and Tompkins Algorithm

In this paper, the case study is the well-known Pan and Tompkins algorithm, for the detection of QRS complexes in ECG signals [22]-[23]. Figure 2 shows a normal ECG signal. It has different segments, the P wave, the QRS complex and the T wave. Among them, the QRS complex is the most important part of the waveform and is related to the electrical activity of the heart during the ventricular contraction.
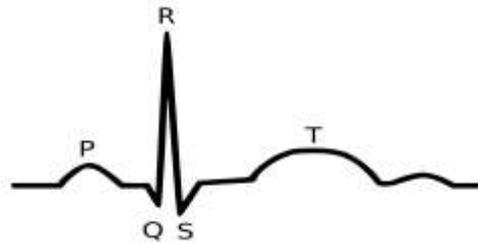
Figure 2. ECG signal

The real-time algorithm is composed of a Digital Signal Processing (DSP) section and a final decision element. The first two operations of the DSP algorithm consist in the application of two IIR filters, a 15 Hz low-pass filter followed by a 5 Hz high-pass filter. The resulting band-pass filter removes the noise due to power line interference, baseline wander, motion artifacts, muscle contraction, and electrode contact disturbs. Then, the signal is differentiated to extract the slope information. The differentiated output is then squared to maximize the amplitude difference of QRS complex with other peaks. Finally, the squared output signal passes through a moving window integrator to smooth the signal by removing the fluctuations in signal peaks. For a frequency sampling of 200 Hz, the typical window width is 32. The filtered ECG signal is shown in Figure 3a. After the signal is filtered, QRS peaks are detected. The detection rules used by the algorithm, determine the peak height, the peak location, and the maximum derivative to classify peaks. When a peak occurs, it is classified as either a QRS complex or noise. At each peak, higher than detection threshold and classified as QRS complex, the algorithm associates a spike. These spikes are shown in red in Figure 3b. The detection threshold is automatically calculated using the estimate of the average QRS peak and the average noise. It is shown in green in Figure 3b.
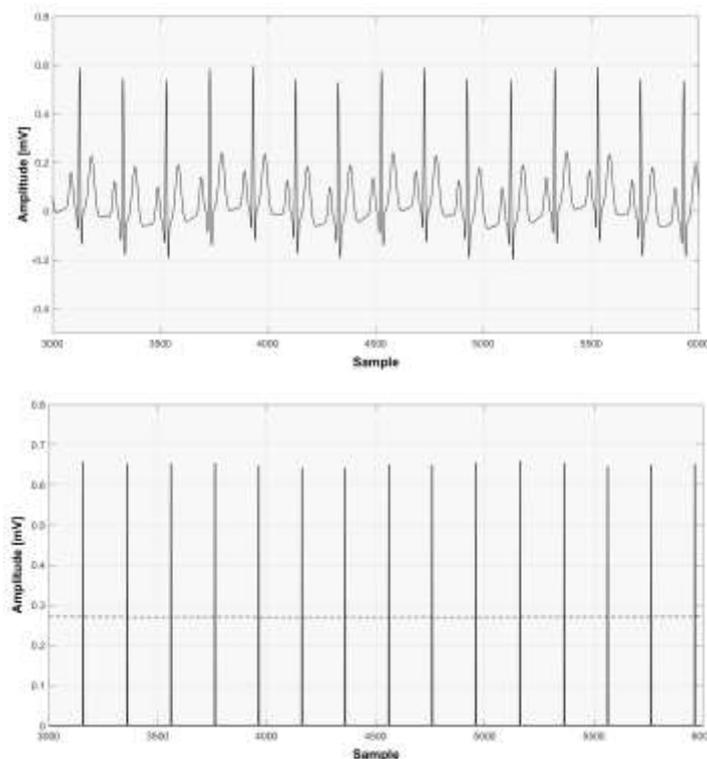
Figure 3. Processed ECG signal. (a) Filtered ECG signal (b)QRS detected (solid line) and detection threshold (dashed line).

4. Experimental Setup

Power consumption experiments were performed implementing the Pan and Tompkins algorithm on a microprocessor and on a system composed of a microprocessor and a hardware accelerator. Given the need to have on the same chip a microprocessor and a hardware accelerator, the experiments were performed on a FPGA. The FPGA used for the experiments is a Xilinx Artix 7 and the microprocessor is a Microblaze soft processor. This choice assures that both microprocessor and hardware accelerator are implemented using the same technology. This aspect is very important in order to obtain valid results.

The design flow was the following:
a. Software implementation of the algorithm on the microprocessor.
b. Profiling of the software to individuate in which portion of the algorithm the microprocessor spends the most of the time.
c. VHDL implementation of the hardware accelerator.
d. Integration of the hardware accelerator with the microprocessor.
e. Realization of the energy consumption tests.

The software profiling shows that the microprocessor spends the greatest part of the time for executing the digital filtering of the Pan and Tompkins algorithm. For this reason, a hardware accelerator was realized for implementing these operations. The hardware accelerator performs the following operations: a low-pass filtering, a high-pass filtering, a derivative and moving window integration. This accelerator was implemented in VHDL and integrated into the Microblaze microprocessor using the AXI-Lite interface. The board used for the experiments is the "Xilinx SoC ZC706 Evaluation Kit". This board provides the possibility to measure the power consumption using a Texas Instruments probe (TI USB Interface Adapter [24]), that continuously measures and monitors the power supplies. In order to evaluate the effects induced by the presence of the hardware accelerator in terms of energy saving, the two methods for the reduction of energy consumption explained in section 2 were implemented.

5. Experimental Results

The estimation of the energy saving was performed through a series of tests. The first step was the estimation of the speedup factor S introduced by the hardware accelerator. From the results shown in Table 1, it is possible to notice that $S \cong 10$.

Table 1. Clock Cycles Required for Computation

SE P	CLOC CY LES
MICRO	63 942.478
MICRO+ACC	6.632.720

Successively, the power consumption of the two systems (microprocessor and microprocessor plus the hardware accelerator) was measured using the TI USB Interface Adapter. The results were collected by the TI Fusion Digital Power Designer Graphical User Interface. Starting from above measurements, the direct and indirect energy reduction methods were applied to the circuit. In order to evaluate the dynamic power, a preliminary evaluation of static power consumption was performed. In this measurement, we observed a large value of the static power with respect to the dynamic one. This is due to the use of a big FPGA, if compared to the complexity of the implemented system. For this reason, the effect of static power was removed in the following experimental results.

5.1. Direct Energy reduction

Power consumption graphs are shown in Figure 4 and in Figure 5. As shown in these graphs, in this case, we have K<<1 and consequently the energy saving is obtained for any value of α and it is proportional to α. The very small value of the power ratio K was obtained introducing the hardware optimization presented in [25], in which all multipliers have been replaced by shifters and area occupancy was reduced optimizing the wordlengths of the fixed-point representation.

Figure 4 shows the power vs time graph for the algorithm executed only by the microprocessor. It is possible to see that when microprocessor does not compute there is only static power dissipation. During the algorithm execution power increases for the dynamic power contribution. The measured dynamic power during the computation is 0.21 W at 100 MHz. Figure 5 shows the power vs time graph for the system composed of the microprocessor and the hardware accelerator. It is possible to see that the execution time has been reduced by the factor S. Because K is very small, the energy reduction is equal to S, that in this case is 10.
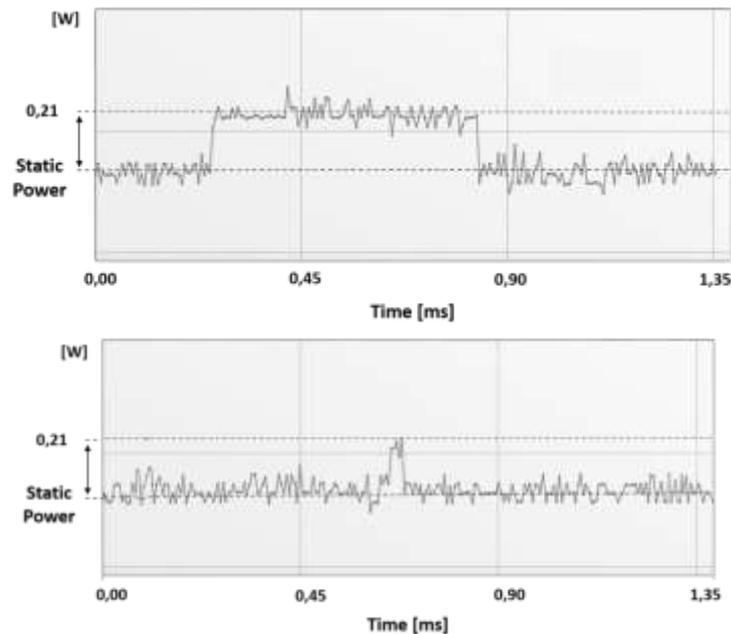
Figure 5. Power consumption of microprocessor plus hardware accelerator

5.1. Inirect Energy reduction

As explained in previous sections, if the initial execution time T satisfies the time constraints, a hardware accelerator introducing speed-up factor S can be introduced to reduce the clock frequency. In our experiments, the speed-up factor is S=10. It implies that it is possible to reduce clock frequency by a factor 10 ($\tilde{f}$=10MHz). In this way, the execution time is unaltered, but the power is reduced due to the clock scaling. In particular, the dynamic power measured during the computation is 0.21 W at 100 MHz, whereas reducing the clock frequency to 10 MHz the power measured is about 0.02 W.

6. Conclusions

In this paper, authors deal with the issue of the power consumption reduction in embedded microprocessors using hardware accelerators. Two different methodologies for the energy consumption reduction were analyzed and tested. The two methodologies were tested on a small system (microprocessor plus accelerator) implemented on a FPGA. The two methods give the same results, in terms of power consumption reduction. If the system is implemented using an ASIC methodology, the indirect energy reduction method can give additional advantages. In fact, the clock frequency reduction allows the decreasing of the voltage supply, quadratically reducing the dynamic power consumption as shown in equation 1.

References

[1] Iazeolla G, Pieroni A. Energy Saving in Data Processing and Communication Systems. *The Scientific World Journal* 2014; 2014: 1-11

[2] Iazeolla G, Pieroni A. Power Management of Server Farms, Applied Mechanics and Materials 2014; 492: 453-459

[3] Pieroni A, Iazeolla G. Engineering QoS and Energy Saving in the Delivery of ICT Services, Publisher: IGI Global 2016: 208-226.

[4] Petracca, M, Mazzenga, F, Pomposini, R, Vatalaro, F, Giuliano, R. *Opportunistic spectrum access based on underlay UWB signaling*. Proceedings - IEEE International Conference on Ultra-Wideband 2011: (6058822): 180-184

[5] Mazzenga, F, Petracca, M, Pomposini, R, Vatalaro, F, Giuliano, R. *Algorithms for dynamic frequency selection for femto-cells of different operators*. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC, 2010; (5671958): 1550-1555

[6] Giuliano, R, Mazzenga, F, Neri, A, Vegni, AM. Security access protocols in IoT capillary networks. *IEEE Internet of Things Journal*, 2017; 4(3), art. no. 7733119: 645-657

[7] Jiang, F, Hu, Y. Energy-efficient compressive data gathering utilizing virtual multi-input multi-output. *Telkomnika (Telecommunication Computing Electronics and Control)*, 2017; 15(1): 179-189

[8] Sari, L, Aditya, A. Raptor code for energy-efficient wireless body area network data transmission *Telkomnika (Telecommunication Computing Electronics and Control)*, 13 (1): 277-283

[9] Scarpato, N, Pieroni, A, Di Nunzio, L, Fallucchi, F. E-health-IoT universe: A review *International Journal on Advanced Science, Engineering and Information Technology*, 2017; (6): 2328-2336

[10] M Chana, D Estvea, JY Fourniols, C Escribaa, E Campoa. Smart wearable systems: Current status and future challenges, in *Artificial Intelligence in Medicine* 2012; 56, 137156.

[11] N Weste, D Harris. CMOS VLSI Design: A Circuits and System Perspective (4th Edition), in Addisin Wesley Publishing Company, USA, 2010.

[12] SR Vemuru, N Scheinberg. Short-Circuit Power Dissipation Estimation for CMOS Logic Gates, in *IEEE Transactions on Circuits and Systems I Fundamental Theory and Applications*, 1994; 41(11): 762-765.

[13] http://www.nxp.com/products/microcontrollers-and-processors/armprocessors Kinetis R L Series: Ultra-Low Power Microcontrollers (MCUs) based on ARM Cortex-M0+ Core

[14] Altera Corporation, Adding Hardware Accelerators to Reduce Power in *Embedded Systems*, September 2009, ver. 1.0, white paper.

[15] GC Cardarilli, L Di Nunzio, R Fazzolari, M Re. *Algorithm acceleration on LEON-2 processor using a reconfigurable bit manipulation unit*. 8th IEEE Workshop on Intelligent Solutions in Embedded Systems, 2010; (5548433): 6-11.

[16] Cardarilli, GC, Di Nunzio, L, Fazzolari, R, Pontarelli, S, Re, M, Salsano, A. *Implementation of the AES algorithm using a Reconfigurable Functional Unit* (2011) ISSCS 2011 - International Symposium on Signals, Circuits and Systems, Proceedings, art. (5978668): 97-100.

[17] GC Cardarilli, L Di Nunzio, R Fazzolari, M Re, *Fine-grain reconfigurable functional unit for embedded processors*, in Conference Record - Asilomar Conference on Signals, Systems and Computers, 2011; (6190048): 488-492.

[18] Razdan, Rahul, Brace, Karl, Smith, Michael D. *PRISC software acceleration techniques*, Proceedings - IEEE International Conference on Computer Design: VLSI in Computers and Processors, 1994: 145-149.

[19] Hilewitz, Y, Lee, RB. Fast bit gather, bit scatter and bit permutation instructions for commodity microprocessors", in *Journal of Signal Processing Systems*, 53 (1-2 SPEC. ISS.), 2008: 145-169.

[20] Hauck, S, Fry, TW, Hosler, MM, Kao, JP. The Chimaera reconfigurable functional unit, in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2004; 12(2): 206-217.

[21] GC Cardarilli, L Di Nunzio, R Fazzolari, M Re, RB Lee. *Integration of butterfly and inverse butterfly nets in embedded processors: Effects on power saving*, 3rd ed. in 46th Asilomar Conference on Signals, Systems and Computers, Article number 6489268, 2012: 1457-1459.

[22] J Pan, WJ. Tompkins. A real-time QRS detection algorithm, in *IEEE Trans. Biomed. Eng.,* 1985; (BME-32): 230-236. DOI: 10.1109/TBME.1985.325532

[23] Silvestri Francesca, Cardarilli Gian Carlo, Di Nunzio Luca, Fazzolari Rocco and Re Marco, Comparison of Low-Complexity Algorithms for Real-Time QRS Detection using Standard ECG

Database, *International Journal on Advanced Science, Engineering and Information Technology*, vol. 8, no. 2, 2018.

[24] Texas Instruments. USB Interface Adapter Evaluation Module-User's Guide. Aug. 2006, http://www.ti.com/lit/ml/sllu093/sllu093.pdf

[25] F Silvestri, S Acciarito, GC Cardarilli, GM Khanal, L Di Nunzio, R Fazzolari, M Re, *FPGA Implementation of a Low-power QRS extractor*, in Lecture Notes in Electrical Engineering, 2018 (ARTICLE IN PRESS). *Journal on Advanced Science, Engineering and Information Technology*, 2018; 8(2).