Abstract—This paper presents preliminary results regarding system-level power awareness for FPGA implementations in wireless sensor networks. Re-configurability of field programmable gate arrays (FPGA) allows for significant flexibility in its applications to embedded systems. However, high power consumption in FPGA becomes a significant factor in design considerations. We present several ideas and their experimental verifications on how to optimize power consumption at high level of designing process while maintaining the same energy per operation (low-level methods can be used additionally). This paper demonstrates that it is possible to estimate feasible power consumption savings even at the high level of designing process. It is envisaged that our results can be also applied to other embedded systems applications, not limited to FPGA-based.

Keywords—Power optimization, FPGA, system-level designing, wireless sensor networks.

I. INTRODUCTION

A typical node of sensor networks (for either military or civilian applications) consists of a wireless communication unit, a processing unit, a sensing unit, and a power unit [12], [13], [14]. Moreover, sensor nodes are often used in such applications where their power source, e.g. battery, is irreplaceable in field. This accounts to sensor node constraints such as performance and power consumption. Therefore, processing units with fixed architecture, e.g. microcontrollers (MCU), digital signal processors (DSP), or their special combinations, are the majority of processors used.

However, with technology advancement, reconfigurable architectures become more powerful and popular, e.g. Altera FPGA chips, Xilinx FPGA chips [15], [16]. Hence, there is significant increase in attention paid to reconfigurable architectures as the processing units currently, e.g. software-based processors (LatticeMico, Nios, MicroBlaze, PicoBlaze, XTensa) [15], [16], [17], [18]. It is envisaged that employing such reconfigurable processing unit to sensor node may improve flexibility and suitability for handling a wider range of scenarios (including some unpredictable ones) in applications like surveillance, monitoring, etc. Up to our knowledge, there are few wireless sensor node applications where FPGA chip is employed. However, FPGA is only used as a supporting processing unit, and its advantageous reconfigurability is almost neglected [28], [29], [30], [31].

Field programmable gate arrays (FPGA) are the natural candidates for such applications. FPGA typically incorporates (apart from its main array of slices and I/O blocks) a number of other hard cores, e.g. memory blocks, digital clock managers, encryption circuitry, and custom multipliers [3].

Although power and performance of FPGA are often compared to their standard-cell ASIC counterparts, [3], [5], programmability of FPGA results in the interconnection structures with larger loading than custom circuits, [10]. The capacity load of signal nets over dedicated metal wires is significantly increased by signal buffers, pass transistors, and other programmable switching structure.

Therefore, the flexibility of FPGA accounts for a significant power consumption increase compare to other processing units with almost fixed architecture. Carefully analyzed tradeoffs between power consumption and performance may mitigate these drawbacks of FPGA while maintaining the same energy per operation.

Technology advancements of high level designing techniques, e.g. compilers (Quartus, ISE), hardware description languages, e.g. Verilog, VHDL, and high level languages for hardware description, e.g. Handel-C, allow synthesizing and prototyping processing units in shorter time and without tedious low level designing techniques (assuming some power and hardware resources overheads) [15], [16], [19]. In this article, we show that we are able to estimate power consumption savings, with acceptable precision, even at the highest level of designing process implementing relevant designs with Handel-C.

The paper presents results of several experiments on power optimization in FPGA. We analyze the issues of clock frequency, the size of design, multi-domain implementations (with various clock frequencies), and chip area constraints. The following sections are structured as follows. Section 2 introduces to power consumption of FPGA and means of power consumption reduction. Section 3 addresses the experiment setup and preliminary results. Section 4 concludes this paper.
II. POWER CONSUMPTION IN FPGA

Power consumption of components fabricated in CMOS technology, e.g. FPGAs, comprises of static and dynamic parts [3], [5], [7], [8], [9], [11].

A. Dynamic Power

The dynamic power of a CMOS device is caused by signal transitions at the device transistors [7], [9], [11]. Frequencies of signal transitions are obviously related to the clock frequency. Thus, the dynamic power consumption is generally modeled as

\[ P = \sum_i C_i \cdot V_i^2 \cdot f_i \]  

(1)

where \( C_i \), \( V_i \), and \( f_i \), represent capacitance, the voltage swing, and clock frequency of the resource \( i \), respectively [3], [5], [9], [11]. The total dynamic power consumed by a device is the summation of the dynamic power of each resource.

Because of programmability of FPGA the dynamic power is design-dependent [2], [3]. Thus, there are design-dependent factors that contribute to the dynamic power: the effective capacitance of resources, the resources utilization, and the switching activity of resources [2], [3], [5].

The effective capacitance corresponds to the sum of parasitic effects due to interconnection wires and transistors. Since FPGA architecture usually provides more resources than required to implement a particular design, some resources are not used after chip configuration and they do not consume the dynamic power (this is referred to as resource utilization). Switching activity represents the average number of signal transitions in a clock cycle. Though generally it depends on the clock itself, it may also depend on other factors (e.g. temporal patterns of input signals). Hence, (1) can be rewritten as

\[ P = V^2 \cdot f \cdot \sum_i C_i \cdot U_i \cdot S_i \]  

(2)

where \( V \) is the supply voltage, \( f \) is the clock frequency, and \( C_i \), \( U_i \), and \( S_i \), are the effective capacitance, the utilization, and the switching activity of each resource, respectively [3].

B. Static Power

The static power is caused mainly by the leakage current between power supply and ground. The leakage current consists of the reverse biased PN-junction current, the sub-threshold leakage, the gate induced drain leakage, the punch through, and the gate tunneling [11]. The sub-threshold leakage current (depending on temperature and the threshold voltage \( V_{th} \)) dominates the leakage current.

Until recently, the leakage current was not taken into consideration because of its negligible level. However, the static power is currently expected to increase dramatically along with shrinking transistor size.

Moreover, the design shrinking of the features affects significantly the thermal characteristics. On-chip temperature of processing unit may vary among the whole chip area. The maximum on-chip temperature is related to the chip area and the maximum power dissipation. Therefore, reduction of the total maximum power may decrease the maximum temperature, so influences the static power.

Recent researches [3] show that the static power of modern FPGAs, e.g. the Virtex-II family (SRAM-based FPGA, 0.15 \( \mu \)m technology), ranges between 5 and 20% of the total dissipated power, depending on the temperature, the clock frequency, and the implemented design.

C. Reduction of Power Consumption

There are three major strategies in FPGA power consumption reduction [2]. First, changes can be done at the system level (e.g. simplification of the algorithms used). Secondly, if the architecture of FPGA is already fixed, a designer may change the logic partitioning, mapping, placement and routing. Finally, if no changes at all are possible, enhancing operating conditions of the device may be still promising (this includes changes in the capacitance, the supply voltage, and the clock frequency).

III. EXPERIMENT SETUP AND PRELIMINARY RESULTS

Typical sensor nodes perform operations such as sensing, detection, and classification [20], [21], [22], [23], [24], [25], [26], [27]. Popular sensing algorithms include the exponential weighted moving average (EWMA) filter used as the low-pass filter, the auto regressive moving average (ARMA) filter used as the high-pass filter, and a simple moving average (SMA) filter also used as the low-pass filter [20], [21], [22], [23], [24], [25], [26], [27]. Due to their different complexity, they differ in hardware requirements and logic delays.

After sensing, the processed data may be passed to the detection and/or classification modules. In wireless sensor networks, the important issue is how to effectively perform all the operations under existing constraints, i.e. the limited capacity of power sources of the sensor node and limitations of the communication mechanism (power, range and bandwidth). Thus, an important step is to analyze how effectively various sensing algorithms (filters) can be implemented within the node FPGA. In the following sections we discuss various aspects of this issue: such as clock frequency, the size of design, multi-domain implementations (with various clock frequencies), and chip area constraints. We choose EWMA filter implementation as an exemplary case study.

The purpose of this experiment is to use the logic of FPGA effectively. Hence, feasible functionality of the implemented logic may not account significantly to the quality of results.

It is envisaged that other filter or algorithm implementations should not account significantly to the quality of results.
A. Experiment Setup

In our experiment we investigate various implementations of EWMA filter. Algorithms are coded in Handel-C using DK Design Suite (Complete design environment for C-based algorithmic design entry, simulation and synthesis) and are compiled for Xilinx Virtex-II FPGA (xc2v3000fg676-4). We also incorporate relevant development board libraries to integrate the implementations of algorithms into real designs. This is required to obtain essential design timing constraints for map, place and route tools. Power efficiency of each particular design is investigated by XPower (one of the accessories of Xilinx Integrated Software Environment (ISE)) that provides estimations of power consumption.

Development board libraries, which are incorporated into design implementation, provide support for audio-to-digital (ADC) and digital-to-audio (DAC) converters. In our experiment, audio data is in-sampled, processed by the relevant filtering algorithm, and out-sampled.

Algorithm verification is performed on simulation level (Handel-C) and the hardware validation is conducted on RC203 development board.

B. Power Consumption and Clock Frequency

According to equation (1) the dynamic power consumption is supposed to increase linearly with changes of clock frequency and size of a design.

Power estimation tools such as XPower require simulation data describing activity of the implemented design. However, embedded systems such as sensor nodes are often deployed in hard-to-predict environments. Therefore, we somehow arbitrarily assume that activity rate of our designs is 50%.

To simulate various sizes of the design, we have implemented designs with 3, 6, 9, and 24 copies of the same EWMA filter. These filters were implemented in parallel. The designs occupy 7%, 11%, 16%, and 36% of available slices within FPGA, respectively. They include overheads of ADC and DAC circuits (that occupy 3% of available FPGA slices).

As a reference, we have also implemented a design with only ADC-DAC circuits (that occupy 3% of available FPGA slices). The usage of multiple clock domains is a well known technique that allows performance increase, and power and/or energy decrease. However, up to our knowledge this is only discussed at low level of designing process. Therefore, we believe that there is strong need to investigate applicability of such a technique at higher level of designing process for power consumption savings.

D. Multiple Clock Domains

The usage of multiple clock domains is a well known technique that allows performance increase, and power and/or energy decrease. However, up to our knowledge this is only discussed at low level of designing process. Therefore, we believe that there is strong need to investigate applicability of such a technique at higher level of designing process for power consumption savings.

As we see from this figure, with the clock frequency increase the effect of the design size on power consumption is decreasing. For example, for 2MHz clock frequency the total dynamic power consumption for designs with 3 and 24 copies of EWMA filter is equal to 26mW and 40mW, respectively. However, for the clock frequency of 20MHz, the total dynamic power consumption for the same designs is equal to 63mW and 238mW, respectively. Smaller design with 3 copies of EWMA filter occupies 1079 slices (with ADC-DAC overheads of 482 slices), and design with 24 copies of EWMA filter occupies 5294 slices (with ADC-DAC overheads) that is almost 5 times more. For low clock frequencies, e.g. 2MHz, a fivefold increase of the occupied area gives only 1.5 times increase of the dynamic power consumption. However, if the clock frequency is much higher, e.g. 20MHz, the dynamic power consumption is increased 3.8 times. From this figure we can conclude that FPGA designs can be enlarged with a disproportionally low dynamic power increase as long as the device operates at low frequencies. Only at the highest frequencies, the dynamic power changes proportionally to the design area.
The results are still satisfactory. map utilization problems with achieving desired performances in with 48 copies of EWMA filter. The latter result is not power consumption overheads, even for the largest design Moreover, we can also notice that generally there are no hardware overheads due to channels are rather minor. such a large design that occupies 76% of available slices and 904 slices, respectively. It is worth noticing that even for designs with 2, 3, 4, and 5 clock domains, the first domain contains only ADC-DAC circuits, while the other clock domains contain equal numbers of EWMA filters. Each design incorporates the same number of EWMA filters altogether. For example, the single-clock domain design consists of 24 copies of EWMA filter, the two-domain design consists of 24 copies of EWMA filter in the second domain (the first one contains only ADC-DAC circuits), the three-domain design consists of 12 copies of EWMA filter in the second and the third domain, etc.

Clock domains are functionally (and physically) connected so that data sampled in the first clock domain are sent to the second clock domain, processed there, and sent to another clock domain for further processing. The last clock domain also performs processing and then sends data back to the first clock domain, where data are out-sampled. Channels and interfaces used for clock domain interconnections are 18bit wide.

1) Channels Overhead

To investigate power consumption and hardware resources overheads due to clock domain interconnections, we have implemented designs with 1 up to 5 clock domains, and with 12, 24, and 48 copies of EWMA filter in total. Designs with 2 up to 5 clock domains have the same amount of copies of EWMA filter in each domain. Originally, all designs are clocked with the same frequency, i.e. 44.3MHz. Results are presented in Tables I, II, and III.

By comparing the first row of each table to the other rows, hardware resources overheads due to channels can be estimated. For designs with 5 clock domains, and with 12, 24, and 48 copies of EWMA filter the overheads are 452, 705, and 904 slices, respectively. It is worth noticing that even for such a large design that occupies 76% of available slices hardware overheads due to channels are rather minor. Moreover, we can also notice that generally there are no power consumption overheads, even for the largest design with 48 copies of EWMA filter. The latter result is not straightforwardly expected, since for such a high resource utilization problems with achieving desired performances in map, place and route tools might be encountered. However, the results are still satisfactory.

<p>| TABLE I | DESIGN WITH 12 COPIES OF EWMA FILTER |</p>
<table>
<thead>
<tr>
<th>No of clock domains</th>
<th>Total dynamic power consumption (mW)</th>
<th>No of used slices (utilization)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>296</td>
<td>2900 (20%)</td>
</tr>
<tr>
<td>2</td>
<td>310</td>
<td>3054 (21%)</td>
</tr>
<tr>
<td>3</td>
<td>315</td>
<td>3295 (22%)</td>
</tr>
<tr>
<td>4</td>
<td>291</td>
<td>3289 (22%)</td>
</tr>
<tr>
<td>5</td>
<td>296</td>
<td>3352 (23%)</td>
</tr>
</tbody>
</table>

<p>| TABLE II | DESIGN WITH 24 COPIES OF EWMA FILTER |</p>
<table>
<thead>
<tr>
<th>No of clock domains</th>
<th>Total dynamic power consumption (mW)</th>
<th>No of used slices (utilization)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>533</td>
<td>5294 (36%)</td>
</tr>
<tr>
<td>2</td>
<td>486</td>
<td>5444 (37%)</td>
</tr>
<tr>
<td>3</td>
<td>536</td>
<td>5769 (40%)</td>
</tr>
<tr>
<td>4</td>
<td>498</td>
<td>5896 (41%)</td>
</tr>
<tr>
<td>5</td>
<td>495</td>
<td>5999 (41%)</td>
</tr>
</tbody>
</table>

<p>| TABLE III | DESIGN WITH 48 COPIES OF EWMA FILTER |</p>
<table>
<thead>
<tr>
<th>No of clock domains</th>
<th>Total dynamic power consumption (mW)</th>
<th>No of used slices (utilization)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>833</td>
<td>10052 (70%)</td>
</tr>
<tr>
<td>2</td>
<td>847</td>
<td>10204 (71%)</td>
</tr>
<tr>
<td>3</td>
<td>874</td>
<td>10777 (75%)</td>
</tr>
<tr>
<td>4</td>
<td>894</td>
<td>10994 (76%)</td>
</tr>
<tr>
<td>5</td>
<td>868</td>
<td>10956 (76%)</td>
</tr>
</tbody>
</table>

2) Clock Domains and Reduction of Power Consumption – Channels

In this experiment we wanted to investigate usefulness of this approach for power consumption reduction. To conduct this experiment we have implemented various designs with 5 clock domains using channels and interfaces. Functionally, these designs are the same as designs described in previous sections. Nevertheless, each implementation consists of different number of copies of EWMA filter in each domain. We have implemented designs with 3, 6, and 9 copies of EWMA filter in each domain, respectively. Moreover, these designs were also implemented with various clock frequencies in each domain.

For simplicity of problem description, we have assumed that clock frequency of each domain is described by the clock frequency division factor (CFDF). E.g. 11111 means that each domain is clocked with the same basic frequency, and 12111 means that the second domain is clocked with a half of the basic frequency. In our experiment basic clock frequency is equal to 44.3MHz. This is the maximum clock frequency that the implemented circuits of EWMA filter can be clocked with.

Initially, we tried to estimate power consumption savings before the experiment has been conducted. We had assumed that the total dynamic power consumption related only to relevant number of copies of EWMA filter could be simply obtained by subtracting power consumption of the design with only ADC-DAC from power consumption of the design with these filters. We decided that because of negligible hardware resources for channel overheads such approach is justifiable. Using the results from section III.B we derived empirical equations describing power consumption against clock
filter in each domain) that utilize 41% and 59% of available
tables, even for large designs (with 6 and 9 copies of EWMA
power consumption since additional routing resources must be
available for inter-domain interconnections) are given in Tables IV, V,
respectively.

According to the above equations, if the clock frequency is
reduced by half, the estimated power consumption savings for
each domain (clocked by the lower frequency) with 3, 6, and 9
copies of EWMA filter, should be 22mW, 45mW, and 75mW,
respectively.

Results of the experiment (designs with use of channels for
clock domains interconnections) are given in Tables IV, V,
and VI.

As we see from the tables, results obtained in our
experiment, for the design with 3 copies of EWMA filter in
each domain, are close to the estimated power consumption
savings. However, for much bigger designs, i.e. with 6 and 9
copies of EWMA filter in each domain, only some results are
close to our expectations.

Apparently for designs occupying a large portion of FPGA
resources, map, place, and route tools try to achieve desired
performances by spreading the logic of each particular clock
domain over a wider area of the chip. This may increase the
power consumption since additional routing resources must be
used to interconnect relevant logic. However, as seen in the
tables, even for large designs (with 6 and 9 copies of EWMA
filter in each domain) that utilize 41% and 59% of available
slices, a significant power consumption reduction is still
possible.

3) Clock Domains and Reduction of Power Consumption

Interfaces

One of the reasons of using interfaces at a high-level of
designing process is to integrate a particular design with other
designs that are already synthesized on lower-levels into, e.g.
Electronic Design Interchange Format (EDIF), VHDL
Hardware Description Language (VHDL), other hardware
description languages, or provided as IP cores. Moreover,
such separately synthesized designs can be assigned chip area
constraints for mapping, placing and routing. This can be done
at a lower-level of designing process, e.g. floorplanning. However, we wanted to investigate usefulness of above
approaches for power consumption reduction.

To conduct this experiment we have implemented various
designs with 5 clock domains. Assumptions to this experiment
as well as design and clock domains functionality are the same
as in section III.D.2. However, we have physically divided our
design into separate modules each constituting a particular
clock domain. Such a division was done at a high-level of
designing process, i.e. Handel-C, and each module was
synthesized into separate EDIF files used for mapping, placing
and routing in next. Moreover, we decided to use a top-level
design as a wrapper for all modules. The top-level is
functional equivalent of the first clock domain from
experiment in section III.D.2. Hence, the top-level design is
in-sampling data, passing data to the physically separated
second clock domain, receiving data from the last physically
separated clock domain, and out-sampling data. Moreover,
this wrapper defines interconnections between other clock
domains by defining relevant interfaces. However, the top-
level is not involved in the order of inter-domain
communication. The top-level is synthesized with relevant
FPGA family chip settings (Xilinx Virtex-II FPGA), and other
designs are synthesized as cores. This is required by
map, place, and route tools.

When all relevant design files are imported into map, place,
and route tools, we perform translation. Then, we can apply
relevant area constraints to each of the modules, Fig. 2. However, we do assign area constraints only for the clock
domains containing implementation of EWMA filters, since
they constitute major hardware resources utilization.

These clock domains are located close to each other to
avoid routing overheads due to data flow. However, we leave
some space between them and to chip borders for additional
routing within a particular clock domain. Moreover, we assign
area constraints to about 125-130% of required space for a
particular clock domain to allow for efficient routing.

Table IV

<table>
<thead>
<tr>
<th>CFDF</th>
<th>Total dynamic power consumption (mW)</th>
<th>Dynamic power consumption decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>289</td>
<td>-</td>
</tr>
<tr>
<td>12111</td>
<td>266</td>
<td>23</td>
</tr>
<tr>
<td>12211</td>
<td>236</td>
<td>30</td>
</tr>
<tr>
<td>12221</td>
<td>209</td>
<td>27</td>
</tr>
<tr>
<td>12222</td>
<td>184</td>
<td>25</td>
</tr>
</tbody>
</table>

Table V

<table>
<thead>
<tr>
<th>CFDF</th>
<th>Total dynamic power consumption (mW)</th>
<th>Dynamic power consumption decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>489</td>
<td>-</td>
</tr>
<tr>
<td>12111</td>
<td>472</td>
<td>17</td>
</tr>
<tr>
<td>12211</td>
<td>377</td>
<td>95</td>
</tr>
<tr>
<td>12221</td>
<td>324</td>
<td>53</td>
</tr>
<tr>
<td>12222</td>
<td>284</td>
<td>40</td>
</tr>
</tbody>
</table>

Table VI

<table>
<thead>
<tr>
<th>CFDF</th>
<th>Total dynamic power consumption (mW)</th>
<th>Dynamic power consumption decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>678</td>
<td>-</td>
</tr>
<tr>
<td>12111</td>
<td>671</td>
<td>7</td>
</tr>
<tr>
<td>12211</td>
<td>522</td>
<td>149</td>
</tr>
<tr>
<td>12221</td>
<td>458</td>
<td>64</td>
</tr>
<tr>
<td>12222</td>
<td>381</td>
<td>77</td>
</tr>
</tbody>
</table>

Fig. 2 Area constraints for clock domains with 3, 6, and 9 copies of EWMA filter per domain, respectively.
Results of the experiment for designs with use of interfaces and area constraints are given in Tables VII, VIII, and IX.

TABLE VII
DESIGN WITH 3 COPIES OF EWMA FILTER PER DOMAIN

<table>
<thead>
<tr>
<th>CFDF</th>
<th>Total dynamic power consumption (mW)</th>
<th>Dynamic power consumption decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>267</td>
<td>-</td>
</tr>
<tr>
<td>12111</td>
<td>235</td>
<td>32</td>
</tr>
<tr>
<td>12211</td>
<td>202</td>
<td>33</td>
</tr>
<tr>
<td>12221</td>
<td>162</td>
<td>40</td>
</tr>
<tr>
<td>12222</td>
<td>127</td>
<td>35</td>
</tr>
</tbody>
</table>

TABLE VIII
DESIGN WITH 6 COPIES OF EWMA FILTER PER DOMAIN

<table>
<thead>
<tr>
<th>CFDF</th>
<th>Total dynamic power consumption (mW)</th>
<th>Dynamic power consumption decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>450</td>
<td>-</td>
</tr>
<tr>
<td>12111</td>
<td>383</td>
<td>67</td>
</tr>
<tr>
<td>12211</td>
<td>311</td>
<td>72</td>
</tr>
<tr>
<td>12221</td>
<td>238</td>
<td>73</td>
</tr>
<tr>
<td>12222</td>
<td>172</td>
<td>66</td>
</tr>
</tbody>
</table>

TABLE IX
DESIGN WITH 9 COPIES OF EWMA FILTER PER DOMAIN

<table>
<thead>
<tr>
<th>CFDF</th>
<th>Total dynamic power consumption (mW)</th>
<th>Dynamic power consumption decrease</th>
</tr>
</thead>
<tbody>
<tr>
<td>11111</td>
<td>645</td>
<td>-</td>
</tr>
<tr>
<td>12111</td>
<td>548</td>
<td>97</td>
</tr>
<tr>
<td>12211</td>
<td>438</td>
<td>110</td>
</tr>
<tr>
<td>12221</td>
<td>330</td>
<td>108</td>
</tr>
<tr>
<td>12222</td>
<td>222</td>
<td>108</td>
</tr>
</tbody>
</table>

As we see from the tables, results obtained in this experiment, for various in size designs, are much better than the estimated power consumption savings from previous section. Moreover, power reduction is almost predictable regardless which domain has decreased clock frequency and its location within chip. This was possible by forcing in somehow map, place, and route tools to use hardware resources only within a particular area (area constraints). Nevertheless, it was also possible to achieve desired performance to relevant clock domains. Moreover, we have also observed that requirements for hardware resources are much lesser comparing to the approach with channels. They are 2413, 4237, and 6207 slices for designs with 3, 6, and 9 copies of EWMA filter, that is 39%, 42%, and 37% less, correspondingly.

IV. CONCLUSION

Modern FPGAs may consume up to few hundreds mW of static power, e.g. Xilinx Virtex-II FPGA (xc2v3000fg676-4) consumes 378mW of the static power. Though for moderate and large designs the dynamic power is usually higher, the static power still constitutes a significant part of the total power. For example, an exemplary design of moderate size utilizing 36% of available slices of the same Virtex-II consumes 533mW of the dynamic power. However, the dynamic power is dependent on the design activity, i.e. if the sensor node is not performing any computations, the dynamic power may be much lower, while the static power is still consumed. Thus, the size of the device is obviously a critical issue.

Nevertheless, the dynamic power is more important from the designer’s perspective as it can be controlled/reduced even at the system level by applying proper design methodologies. Our paper presents some ideas (and their experimental confirmations) in this area.

The obvious fact that lower clock frequencies reduce dynamic power can be supplemented with some additional observations. First, we have noticed that when the clock frequency decreases, the size of a design becomes almost unimportant (from the perspective of dynamic power consumption). Hence, certain approaches to the designing process can be recommended.

For example, we can conveniently maintain the same performance of a particular design by reducing its clock frequency and by introducing the increased level of parallelism that would allow more operations executed simultaneously within one clock cycle while maintaining the same energy per operation. Though the hardware resources would be significantly increased (due to additional hardware resources required by the parallelism introduced), the dynamic power consumption increase can be negligible (if a sufficiently low clock frequency is applied).

Although generally small FPGA chips should be used for small designs, the advantages of large FPGAs should not be ignored. They are almost like a “buffet lunch”; the ticket might look expensive (i.e. the static power is high) but you can eat as much as you can (i.e. a large amount of resources can be used for only an insignificant power increase) providing that you eat slowly (i.e. a low clock frequency is used). In wireless sensor networks the additional advantage of such an approach is that reconfigurability (which is always a burden for low-performance communication protocols) may not be necessary at all or at least reduced to very infrequent situations.

A particular design of wireless sensor node often consists of algorithms with highly diversified performances needed. For example, algorithms that process raw data must often deal with large amounts of data, while algorithms employed for detection may perform computation on data that are already filtered and thus delivered in much lower quantities. Our experiments indicate that using multiple clock domains with different clock frequencies is a feasible solution for such scenarios. The dynamic power consumption may be decreased by moving relevant parts of design logic to other domains and by clocking these domains with lower frequency. Although this method is not giving straightforward results in case of using channels (the actual power savings not always accurately correspond to the calculations) but power savings are still satisfactory. However, a combination of this approach with adequate low-level tools (area constraints) allow for much better power savings that are more predictable. Moreover, this is envisaged that employing the latter approach should not require significant changes to existing designs. We believe that functional blocks of a design (done at the system
level of designing process) can be straightforwardly mapped to particular clock domains. The only effort is to implement hand-shaking for relevant clock domains (for data integrity).

We have also proved that power, energy, and performance modeling is possible at the high level of designing process with acceptable precision.

Re-configurability and flexibility of FPGA offers a number of possibilities to embedded systems, enriching their features and prolonging the product lifetime. Hence, we believe that results of our experiments may be useful to other FPGA-based embedded systems, and they are not limited to wireless sensor networks.

REFERENCES