International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:1, No:4, 2007

Power Optimization Techniques in FPGA Devices: A Combination of System- And Low-Levels

Pawel P. Czapski, and Andrzej Sluzek

***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.

Manuscript received May 10, 2007.

P. P. Czapski is with the School of Computer Engineering, Nanyang Technological University, Singapore (e-mail: pczapski@pmail.ntu.edu.sg).

A. Sluzek is with the School of Computer Engineering, Nanyang Technological University, Singapore (e-mail: assluzek@ntu.edu.sg).

II. Power Consumption in FPGA

Power consumption of components fabricated in CMOS technology, e.g. FPGA, 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

$$P = \sum_i C_i \cdot V_i^2 \cdot f_i \qquad (1)$$

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

$$P = V^2 \cdot f \cdot \sum_i C_i \cdot U_i \cdot S_i \qquad (2)$$

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 μm 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 have any applications.

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 and DAC circuits. The validity of equation (1) has been fully confirmed and experimentally obtained coefficients describing the ratio between power consumption and the clock frequencies, for designs with only ADC-DAC circuits, 3, 6, 9, and 24 copies of EWMA filter, are equal to 1.0934, 2.0731, 3.1132, 4.4961, and 11.5, respectively.

C. Power Consumption, Clock Frequency and Size of Design

Results from the previous experiment were used to prepare Fig. 1 describing relations between the power consumption ratio and the clock frequency. This ratio is obtained by dividing the total dynamic power consumption of the design with a number of EWMA filter copies by the total dynamic power consumption of the smallest design, e.g. the design with only ADC-DAC.

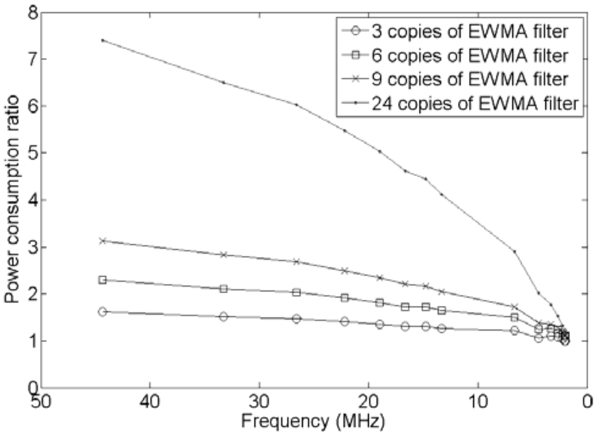

Fig. 1 Relations between power consumption, clock frequency and size of design

As we see from this figure, with the clock frequency decrease 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.

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.

Therefore, this experiment was conducted to investigate benefits and limitations of dividing a design into a number of clock domains (using various frequencies) using a high level language.

The concept of multiple clock domains was implemented at a high-level of designing process, i.e. Handel-C. We have used channels and interfaces for communication between clock domains. The former mechanism (channels) is built into Handel-C and synthesized with essential synchronization and hand-shaking (data integrity) circuits. The latter mechanism (interfaces) that is also built into Handel-C presents only signal interconnections, and essential hand-shaking must be implemented by a designer. Moreover, with little effort at a low-level of the design process, interfaces allow for much more flexibility to the concept of multiple clock domains.

The single-clock domain design consists of a certain number of copies of EWMA filter and ADC-DAC circuits. In the 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 channels 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.

TABLE I
DESIGN WITH 12 COPIES OF EWMA FILTER

No of clock domains	Total dynamic power consumption (mW)	No of used slices (utilization)
1	296	2900 (20%)
2	310	3054 (21%)
3	315	3295 (22%)
4	291	3289 (22%)
5	296	3352 (23%)

TABLE II
DESIGN WITH 24 COPIES OF EWMA FILTER

No of clock domains	Total dynamic power consumption (mW)	No of used slices (utilization)
1	533	5294 (36%)
2	486	5444 (37%)
3	536	5769 (40%)
4	498	5896 (41%)
5	495	5999 (41%)

TABLE III
DESIGN WITH 48 COPIES OF EWMA FILTER

No of clock domains	Total dynamic power consumption (mW)	No of used slices (utilization)
1	833	10052 (70%)
2	847	10204 (71%)
3	874	10775 (75%)
4	890	10904 (76%)
5	868	10956 (76%)

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 frequency. For designs with only 3, 6, and 9 copies of EWMA filters, these equations are, respectively, as follows,

$$p_3(f) = 0.9797 \cdot f + 0.1858 \quad (3)$$

$$p_6(f) = 2.0198 \cdot f + 1.1699 \quad (4)$$

$$p_9(f) = 3.4027 \cdot f - 1.3416 \quad (5)$$

TABLE IV
DESIGN WITH 3 COPIES OF EWMA FILTER PER DOMAIN

CFDF	Total dynamic power consumption (mW)	Dynamic power consumption decrease
11111	289	-
12111	266	23
12211	236	30
12221	209	27
12222	184	25

TABLE V
DESIGN WITH 6 COPIES OF EWMA FILTER PER DOMAIN

CFDF	Total dynamic power consumption (mW)	Dynamic power consumption decrease
11111	489	-
12111	472	17
12211	377	95
12221	324	53
12222	284	40

TABLE VI
DESIGN WITH 9 COPIES OF EWMA FILTER PER DOMAIN

CFDF	Total dynamic power consumption (mW)	Dynamic power consumption decrease
11111	678	-
12111	671	7
12211	522	149
12221	458	64
12222	381	77

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), VHSIC 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. floorplaning. 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.

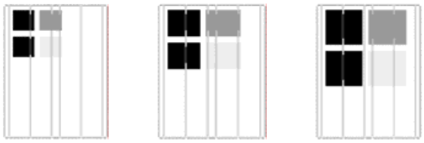

Fig. 2 Area constraints for clock domains with 3, 6, and 9 copies of EWMA filter per domain, respectively

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

CFDF	Total dynamic power consumption (mW)	Dynamic power consumption decrease
11111	267	-
12111	235	32
12211	202	33
12221	162	40
12222	127	35

TABLE VIII
DESIGN WITH 6 COPIES OF EWMA FILTER PER DOMAIN

CFDF	Total dynamic power consumption (mW)	Dynamic power consumption decrease
11111	450	-
12111	383	67
12211	311	72
12221	238	73
12222	172	66

TABLE IX
DESIGN WITH 9 COPIES OF EWMA FILTER PER DOMAIN

CFDF	Total dynamic power consumption (mW)	Dynamic power consumption decrease
11111	645	-
12111	548	97
12211	438	110
12221	330	108
12222	222	108

As wee 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 pro-longing 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

[1] N. Chang and K. Kim, "Real-time per-cycle energy consumption measurement of digital systems", *Electronics Letters, Volume 36, Issue 13*, 22 June 2000. Page(s):1169 - 1171.

[2] H. G. Lee, S. Nam, and N. Chang, "Cycle-accurate energy measurement and high-level energy characterization of FPGAs", *Quality Electronic Design, 2003. Proceedings. Fourth International Symposium* on 24-26 March 2003, Page(s):267 – 272.

[3] L. Shang, A. S. Kaviani, and K. Bathala, "Dynamic Power Consumption in Virtex-II FPGA Family", *Proceedings of the 2002 ACM/SIGDA 10th International Symposium on Field-Programmable Gate Arrays*, pages 157 – 164. ACM Press, 2002.

[4] K. Weiß, C. Oetker, I. Katchan, T. Steckstor, and W, Rosenstiel, "Power Estimation Approach for SRAM-based FPGAs", *International Symposium on Field Programmable Gate Arrays archive, Proceedings of the 2000 ACM/SIGDA eighth international symposium on Field programmable gate arrays table of contents*, Monterey, California, United States, Pages: 195 – 202.

[5] V. Degalahal and T. Tuan, "Methodology for high level estimation of FPGA power consumption", *Design Automation Conference, 2005. Proceedings of the ASP-DAC 2005. Asia and South Pacific, Volume 1*, 18-21 Jan. 2005 Page(s):657 – 660 Vol. 1.

[6] M. French, "A Power Efficient Image Convolution Engine for Field Programmable Gate Arrays", *2004 MAPLD International Conference*, September 8-10, 2004.

[7] Steven J. E. Wilton, Su-Shin Ang, and Wayne Luk, "The Impact of Pipelining on Energy per Operation in Field-Programmable Gate Arrays", *Field Programmable Logic and Application: 14th International Conference, FPL 2004*, Leuven, Belgium, August 30-September 1, 2004. *Proceedings. Volume 3203 / 2004*, Chapter: pp. 719 – 728.

[8] G. J. M Smit and P. J. M. Havinga, "A survey of energy saving techniques for mobile computers", internal report University of Twente, 1997, http://www.home.cs.utwente.nl/~havinga/papers/energy.pdf

[9] P. J. M. Havinga and G. J. M Smit, "Low power system design techniques for mobile computers", internal report University of Twente, 1997, http://www.home.cs.utwente.nl/~havinga/papers/energy.design.pdf

[10] N. Rollins and M. J. Wirthlin, "Reducing Energy in FPGA Multipliers through Glitch Reduction", *2005 MAPLD International Conference*, September 7-9, 2005.

[11] O. S. Unsal and I. Koren, "System-level power-aware design techniques in real-time systems", *Proceedings of the IEEE, Volume 91, Issue 7*, July 2003 Page(s):1055 – 1069.

[12] K. Römer and F. Mattern," The design space of wireless sensor networks", *Wireless Communications, IEEE, Volume 11, Issue 6*, Dec. 2004 Page(s):54 – 61.

[13] M. A. M. Vieira, C. N. Jr. Coelho, D. C. Jr. da Silva, and J. M. da Mata, "Survey on wireless sensor network devices", *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference. Volume 1*, 16-19 Sept. 2003 Page(s):537 – 544 vol.1.

[14] J. Feng, F. Koushanfar, and M. Potkonjak, "System-architectures for sensor networks issues, alternatives, and directions", *Computer Design: VLSI in Computers and Processors, 2002. Proceedings. 2002 IEEE International Conference* on 16-18 Sept. 2002 Page(s):226 – 231.

[15] http://www.xilinx.com/

[16] http://www.altera.com/

[17] http://www.latticesemi.com/

[18] http://www.tensilica.com/

[19] http://www.celoxica.com/

[20] G. Werner-Allen, J. Johnson, M. Ruiz, J. Lees, and M. Welsh, "*Monitoring volcanic eruptions with a wireless sensor network*", *Wireless Sensor Networks, 2005. Proceeedings of the Second European Workshop* on 31 Jan.-2 Feb. 2005 Page(s):108 – 120

[21] R. R. Brooks, P. Ramanathan, and A. M. Sayeed, "*Distributed Target Classification and Tracking in Sensor Networks*", *Proceedings of the IEEE, Volume 91, Issue 8*, Aug. 2003 Page(s):1163 – 1171

[22] T. He, L. Luo, T. Yan, L. Gu, Q. Cao, G. Zhou, R. Stoleru P. Vicaire, Q. Cao, J. A. Stankovic, S. H. Son, and T. F. Abdelzaher, "*An overview of the VigilNet architecture*", *Embedded and Real-Time Computing Systems and Applications, 2005. Proceedings. 11th IEEE International Conference* on 17-19 Aug. 2005 Page(s):109 – 114

[23] L. Girod and M. A. Roch, "*An Overview of the Use of Remote Embedded Sensors for Audio Acquisition and Processing*", *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium* on Dec. 2006 Page(s):567 – 574

[24] D. Li, K. Wong, Y. Hu, and A. Sayeed, "*Detection, Classification and Tracking of Targets in Distributed Sensor Networks*", *IEEE Signal Processing Magazine, 19(2):17--30*, March 2002

[25] L. Gu, D. Jia, P. Vicaire, T. Yan, L. Luo, A. Tirumala, Q. Cao, T. He, J. A. Stankovic, T. Abdelzaher, and B. H. Krogh, "*Lightweight Detection and Classification for WirelessSensor Networks in Realistic Environments*", SenSys'05, November 2–4, 2005

[26] Q. Wang, Wei-Peng Chen, R. Zheng, K. Lee, and L. Sha, "*Acoustic Target Tracking Using Tiny Wireless Sensor Devices*", *IPSN 2003, LNCS 2634, pp. 642–657*, 2003.

[27] A. Arora, P. Dutta, S. Bapat, V. Kulathumani, H. Zhang, V. Naik, V. Mittal, H. Cao, M. Demirbas, M. Gouda, Y. Choi, T. Herman, S. Kulkarni, U. Arumugam, M. Nesterenko, A. Vora, and M. Miyashita, "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking", *Computer Networks, Vol. 46, Issue 5, pp. 605-634*, *Elsevier Science*, Dec. 5, 2004

[28] B. O'Flynn, A. Barosso, S. Bellis, J. Benson, U. Roedig, K. Delaney, J. Barton, C. Sreenan, and C. O'Mathuna, "The Development of a Novel Miniaturized Modular Platform for Wireless Sensor Networks", *In Proceedings of the IPSN Track on Sensor Platform, Tools and Design Methods for Networked Embedded Systems (IPSN2005/SPOTS2005)*, Los Angeles, USA, IEEE Computer Society Press, April 2005.

[29] S. J. Bellis, K. Delaney, B. O'Flynn, J. Barton, K. M. Razeeb, and C. O'Mathuna, "Development of field programmable modular wireless sensor network nodes for ambient systems", *Computer Communications, Volume 28, Issue 13* , 2 August 2005, Pages 1531-1544, *Wireless Sensor Networks and Applications - Proceedings of the Dagstuhl Seminar 04122*.

[30] D. Bauer, S. Furrer, S. Rooney, W. Schott, H. L. Truong, and B. Weiss, Research Report, "The ZRL Wireless Sensor Networking Testbed", *RZ 3620 (# 99630)*, 07/20/2005, IBM Zurich Research Laboratory, 8803 Ruschlikon, Switzerland.

[31] V. Tsiatsis, S. A. Zimbeck, and M. B. Srivastava, "Architecture strategies for energy-efficient packet forwarding in wireless sensor networks", *Low Power Electronics and Design, International Symposium* on, 2001. 6-7 Aug. 2001, Page(s):92 – 95.