International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com
Volume 2, Issue 7, July 2013 ISSN 2319 - 4847

Factors Causing Power Consumption in an Embedded Processor - A Study

Anju S. Pillai[1], **Isha T.B.**[2]

[1&2] Department of Electrical and Electronics Engineering,
Amrita Vishwa Vidyapeetham, Coimbatore.

Abstract

Processors are the computing elements of most of the embedded systems. To cater the increased performance demand, more and more transistors are fused into a single chip, which indirectly accelerates the power consumption of the system. Most of the embedded systems are portable battery powered systems. The usefulness of the system depends on the battery life, which is determined by the power consumption of the entire system. This paper presents a detailed study on the factors that influence and cause power consumption of an embedded processor/controller. The knowledge about the elements that influence the processor functioning leading to power consumption, is vital to devise newer mechanisms for power consumption reduction. The main parameters and issues that have an impact on the power consumption of an embedded processor is also discussed.

Keywords: Power consumption, Embedded processor, Power saving, Power reduction techniques

1. INTRODUCTION

Processors of different size and model are embedded into devices for a variety of reasons viz. perform computations, provide automatic control, development of applications like embedded, biomedical, multimedia and control, provide communication and control over internet and many more. Processors are the brain of these systems. Since the scope of the processors vary widely, their computation power also changes from few hundred instructions to millions of instructions per second. And this makes it hard to compare different processors and evaluate their performance. The existing competition between different manufactures of processor makes it very competitive in terms of increased performance, reduced size, increased clock frequency etc. which puts a tradeoff between the performance and the power consumption. The consumer demands namely compactness in size with all sophistications added up, need of prolonged battery life etc. are hard to attain. With the advancements in technology, the transistor count on a single CPU has crossed 2.5 billion transistors. Integrating these transistors for performance enhancement will also have an impact on the power consumption, because adding more and more transistors give rise to increase power consumption, thereby reducing the battery life due to the increase in the heat dissipated in the device; which reduces the usefulness of the portable embedded system. Due to this, power management has become a design constraint today, for most of the computationally intensive and sophisticated applications.

In order to improve the utility of the embedded system, the power consumption of the entire system has to be minimized. Since the major computations are done by the embedded processor/controller, energy minimization of the processor is also vital for the total power reduction, even though it is relatively small. The Intel corp. has cancelled one of its new generation processor Tejas Pentium 4 processors due to the power related issues [1]. But, for large and complex applications like smart grid, sensor network etc, where thousands and lakhs of processors are present and are performing computations for almost all the time, the cumulative energy consumption of the entire nodes in the network cannot be neglected. This paper focuses on the energy consumption issues on a processor level rather than on the system level. There are available plenty of techniques for power consumption reduction on a system level and relatively there is only little work which aims at power reduction of the embedded processor. The main objective of this paper is to identify the different factors that are affecting the execution of the processor and yielding to power consumption. A detailed study is done to identify various factors and understanding the impact of these factors on the processor performance.

The rest of the paper is organized as follows: Section 2 gives a brief explanation about the necessity of low power processors followed by the overview of factors affecting power consumption of a processor in Section 3. Section 4 details the different levels of abstraction and the power reduction techniques and finally Section 5 presents the conclusion of the paper.

2. NECESSITY OF LOW POWER PROCESSORS

The power management and optimization issues were not very demanding and considered as a crucial aspect in the domain of analog circuit design. But, now it is emerged as a prevailing factor in the digital design community. At present, with the ever growing market, there is a steady and increasing requirement for low power electronic devices. In order to incorporate the power issues in the design, many other significant factors associated with the system are to be looked into viz. the delay associated with the system, Quality of Service (QoS), performance, functional and temporal requirements of the system, throughput, reliability, area, cost etc. So, before advocating a particular low power design, these major criteria has to be looked into, to find a better system design meeting all the specifications after examining each design alternative.

The following section briefs the need for low power processors. In the past, the need for low power devices were not very demanding as the clock frequency and the device density were not as high as present time, and was enough to meet the requirements. Because of this, power issues were not a major concern during that time. But today, because of humans crave for more sophistication and entertainment, most of the devices are packed with thousands and lakhs of processors in a single chip for increased performance with reduced size. This indirectly aids to the power consumption with a steady growth in clock frequency. This is supported by Moore's law which states that the transistor density doubles every 18 months [2]. Therefore, high performance computing adds to increased power dissipation in the system. Building automation systems and other associated devices along with the computers, which are the largest and rapidly increasing electricity load, introduces an additional need of low power processors. Another factor is the escalating requirement of portable battery operated mobile devices in the market.

The duration of functioning of a system depends on the battery life which is affected by the power dissipation in the system. Even though there is an appreciable growth in the device density, the battery technology has not undergone a similar growth which always puts an ever growing gap called as the *battery gap* [3]. This factor also forces the need of power consumption reduction in embedded processors. Though more compact, less weight and sturdy design is demanded by the consumers, it makes it hard to achieve these requirements without an increase in power consumption complexity of the system. As the system becomes more complex, it affects the cooling and the packing cost. The power dissipation of high computing processors is in the order of Watts (W) and the average current consumption in the range of Ampers (A) and the transient current will be a few fold of the average current which puts difficulty in the design of power supply rails. Also, unsusceptibility to digital noise is a problem to be addressed. These factors demand the need of low power processors.

3. Overview of Factors Affecting Power Consumption of a Processor

In order to devise new techniques for energy consumption reduction of the processor, knowledge about the factors affecting processor execution and influencing the performance is essential. There are a variety of factors on which power consumption of a processor depends on. It is hard to analyze the dependency of each factor on power consumption. In this paper, some of the significant factors are studied. The analysis needs to be started with CMOS circuit level. The two major types of power dissipations occurring in a CMOS circuit are:

- *Static dissipation*: Due to leakage currents
- *Dynamic dissipation*: Due to the charging and discharging of capacitance or due to the switching activities of the circuit. As switching activities increase with increased clock frequencies, processors with high operating clock will have more power dissipation [4], [5].

The power dissipation of a CMOS chip can be represented as in Eqn. (1)

$$P = \sum_i C_i * v_i^2 * f_i \qquad (1)$$

where, P is the power consumption of CMOS circuit, C is the effective load capacitance, v the supply voltage and f the clock frequency. From this relation, it is clear that the power consumption of the processor depends linearly on the effective capacitance, supply voltage and clock frequency. Thus, by changing the voltage and frequency, the power dissipation of the processor can be controlled.

3.1 Reducing capacitance

Reducing the parasitic capacitance of the circuit will result in power consumption reduction. Reduction of capacitance should be done with appropriate frequency scaling to get better results and performance increase. High frequency signals have to be routed through low capacitance and vice versa to conserve power. To reduce the capacitance for low power design, select a least gate size enough to meet the speed constraints of the signal switching [4].

3.2 Switching voltage

Due to the quadratic relation of voltage term, reducing the supply voltage can result in substantial savings. The well known technique is the Dynamic Voltage Scaling (DVS) used to save power [6]. But, when the supply voltage is reduced to conserve power, there are some major concerns to be addressed. The performance will vanish due to the slow CMOS transistor. This may lead to system failure in meeting the timing constraints which may result in catastrophic effects for hard real time systems [7]. Therefore, enough precaution must be taken to ensure that power reduction is achieved without violating the temporal and functional constraints of the system. Also, at low voltages, noise immunity becomes difficult.

3.3 Clock frequency

Reducing the clock frequency is beneficial for power consumption reduction and has the same effect as that of reducing the capacitance of the circuit. Eliminating the unwanted logic switching will help to reduce the clock frequency. The other factors that can help to reduce the switching frequency are: changing the number representation and coding forms, finding an alternate logic design for the same circuit, etc. Reducing the switching frequency will improve the system reliability [4].

3.4 Frequency switching, Preemption and Context switches

The main factors on which the processor power dissipation directly depends on are the load capacitance, supply voltage and clock frequency. For low power consumption, the processor voltage and frequency has to be reduced without violating any other specifications. The key technique for power consumption reduction of an embedded processor is Dynamic Voltage Frequency Scaling (DVFS), in which the processor is operated at different operating points which is a duple of supply voltage, clock frequency. But, frequent switching between the operating points is also not desirable, as power wastage is associated with each switching. Another factor which if not controlled can lead to an increase in power consumption of the processor is the preemptions and context switches between tasks or processes. Preemptions and associated context switches are essential for the implementation of an embedded application. If preemptions are not allowed, when a high priority task is ready to execute, it may not get the control of the processor due to execution of a low priority task. When preemption occurs, the context of the task namely, the program counter, temporary register values, etc. are to be saved into the stack. This is context switching and if these are not controlled can even fail the schedulability of the system and can lead to considerable amount of energy wastage [8], [9].

3.5 Cache misses

Cache is the smallest and the fastest memory unit associated with every embedded device. The essential data which is needed for the future computations are stored in the cache and is used by the processor. But, due to the limitation of the memory capacity, all the data cannot be stored as required by the processor, and can lead to cache misses. Cache misses have a great impact on the processor energy consumption due to the additional clock cycles wasted to find the data and the delay incurred. This leads to an increased instruction time and will add to energy penalty [10].

3.6 Resource constraints

Any system design will focus on how effectively the resources can be utilized to get the maximum or optimal performance. Always there exists a tradeoff between the performance achievable and the available resources. The major impact of limited resources is the stalls introduced in the pipeline which degrades the performance of the system, by the inclusion of additional instruction cycles needed to complete the set of instruction execution. Therefore, this factor also has an adverse effect on the energy consumption [10], [11].

3.7 Branch Prediction Overhead

In order to increase the performance of processors with any architecture support, or especially in multi-core systems, branch prediction technique is widely used for reducing the overhead associated with conditional instruction execution. Every branch instruction has to have a penalty of few clock cycles to find the correct address to which the Program Counter (PC) should point next. This is one of the major pipeline hazards which introduces stalls in the pipeline and increase the energy wastage of the system [11].

3.8 Insertion of *WAIT* States

Wait states are inserted in a code when there is an off-chip memory access. When some data has to be accessed from an external memory unit, the delay associated will be more and to incorporate that into the program code, WAIT states has to be inserted, due to the lower operating frequency of the processor execution [11].

3.9 Register File Size

The register file size i.e. the number of registers required by a program code while executing an application task, has an adverse effect on power consumption. As the register file size increases, the amount of memory usage increases, increasing the chip area. Increase in chip area will increase the power consumption of the processor. The use of an optimum number of registers will result in shorter instruction word to reduce the power consumption [11].

3.10 Power-aware Instruction Selection

Another factor which can be considered for power consumption reduction is the selection of appropriate instruction sequence. By identifying a power-aware instruction sequence and avoiding costly external memory accesses, some wasteful power consumption can be reduced [12].

3.11 Program execution time

The execution time or computation time of a code has a direct influence on the energy consumption of the processor. As the code size increases, the number of processor cycle also increases; increasing the duration of active state of the processor. Being in the active state, the processor consumes more energy when compared to an idle state. Therefore, code optimization becomes essential for any embedded application development [4].

3.12 Instruction Scheduling

By re-ordering the instruction execution, without changing the functionality, some amount of power saving can be claimed. By doing so, the circuit can be made to retain in similar states with slight changes only. With reduced state changes, the circuit saves some power dissipation. Along with instruction scheduling, other techniques like: register pipeline, memory layout, jump optimization etc can also be used for better power saving [12].

3.13 Impact of Real Time Operating System (RTOS)

Real time operating systems are used in most of the real time embedded applications to manage the software and hardware resources. Due to this nature, RTOS dissipates a significant amount of power in the system. It was found that on an average; around 32% of the total energy was drawn by the OS alone for a particular workload [13]. But, the impact of RTOS to the application and to the processor/controller is mostly neglected. The user gets estimation about the computation time of different parts of the RTOS when a particular hardware configuration is used. But, there is no knowledge about the power impact. Actually, there is an effect of use of RTOS on the performance and power consumption of the system. The major power consumption happens when the RTOS is running the application software. From the literature, it can be observed that the percentage of energy consumed by the RTOS and the Board Support Programs (BSP) can vary on average from 1% to 99% based on the level of software dependence on the RTOS [12].

3.14 Switching of processor states

Over the time of execution, there is a continuous transition of the processor state, which is a factor to be considered for power consumption. For energy saving, normally processors are assigned with different states like stand-by, power down, idle, sleep, nap etc. When the processor is not executing any job, it is switched to sleep, stand-by or any other idle state to save energy. But, there is an average of some few milliseconds needed to switch the processor from an idle state to an active state. So, by incorporating some predictive power management techniques in the processor, the future re-activation time can be predicted well in advance to avoid the delay and associated power wastage [14].

3.15 Mass Data Storage and Handling

While implementing large and complex applications, huge amount of data is needed to be handled and also stored. This is highly energy inefficient. Generally in data grid systems, the data storage and data computing units are placed far away, and therefore, there is a necessity for huge data transfer. As an alternative, data are replicated at many different locations for reducing the accessing time; avoid loss of data and to reduce the huge amount of data transfer. So, there should be good techniques to support data placement and replacement to have better energy efficiency [10], [15].

3.16 Addressing modes

Any processor architecture will support different addressing modes to facilitate ease of data/operand fetch. The different addressing modes allow the processor to fetch the data in various manners to ease the data procurement and the coding. But, there is an influence of different addressing modes on the power consumption of the processor. One reason is the variations in the length of code aiding the power consumption when the processor expends more time to find the data for executing a particular instruction [10].

3.17 External memory access

Embedded systems, being developed for meeting specific functionality, the associated resources of the system are also limited. Most of the embedded systems have limited on-chip memory including the caches. But, when running complex and large applications, the memory requirement will be more, leading to external memory accesses, which is a major source of energy consumption in embedded systems. The reason is external memory accesses incur additional energy dissipation in terms of increased memory access time and delay. So, during the development stage of the application, enough measures have to be taken to fix the memory capacity of the system [16].

3.18 Input data values

The execution time of a set of code depends on the values of input data. Similarly, there is a firm dependence of energy consumption of different instructions on the data on which it is operating. Because, based on the input data, the conditional instructions which will be executing and therefore the total number of instructions executed will vary; varying the energy consumption. The other energy susceptible parameters are: the value of operands, operand and fetch address of the instruction, register number and values [15].

3.19 Transistor size

Transistors are the basic building block in circuit level and the main factor that influences the power dissipation, area and performance of the circuit is the size of the transistor. Reducing the transistor size is beneficial because it results in a linear reduction in capacitance, thereby reducing the dynamic power [17]. Also, on analyzing the signal probabilities, transistor restructuring in a combinational circuit can help to achieve superior power efficiency. Small techniques like equivalent pin reordering can be exploited because of the fact that logically identical pins may not have similar delay or power consumption [18]. Another important component which is very frequently used in synchronous systems is flip flops and latches. A large amount of power is dissipated to operate latches and flip flops, so reducing the clock power dissipation of these elements can conserve some power [4].

3.20 Clock signals

At the device level, the major power dissipation which occurs in CMOS circuit is the *capacitive power dissipation.* Due to the charging and discharging of node capacitance, *internal switching power* is the dynamic power dissipated inside the logic cell which is a sum of charging and discharging of internal nodes and the short circuit power. Third is *static power,* which is based on the state of the logic circuit. These power dissipations depend on various factors like: operating voltage, temperature, signal slope, output load capacitance, fabrication process etc. Also, some other factors that influence the power dissipation is the logical encoding, Boolean function description and information representation. Clock signal contributes to around 40% of total system power dissipation. Different techniques like clock gating, reduced swing clock etc. can be adopted to reduce the power dissipation associated with the clock signals [4].

3.21 Correlation between samples

Register Transistor Level is also known as architecture level or block level. In this level, the high level functions are performed by the basic units such as: registers, buses, multiplexer, adders, multipliers, memories, state machines etc. In this level, power dissipation can be expressed as a function of the number of bits of the components and the operating frequency. It does not consider the data dependency of power dissipation. The correlations between successive samples are beneficial because there may be more number of bits which are in common. In some cases negative correlation also exist. Anyway, the correlation has an impact on power dissipation because of the switching activities of the data path. The switching activities has to be reduced as minimum as possible to conserve power [19]. Buses are the mode of data exchange between different parts of the circuit; it is a major source of power dissipation in digital circuits. Therefore, applying reduced voltage swing can help to reduce the power dissipation [4].

3.22 Memory

Memory is a vital part of any circuit and the technological growth has allowed the integration of more on-chip memory and data caches. For low power dissipation, bulk Random Access Memory (RAM) units have to be operated at low voltages. But this may affect the speed of operation of the system and to balance, the threshold voltage can be reduced. Better means like using multiple threshold devices, dynamic adjusting of threshold voltage by back bias voltage control, memory bank partitioning etc. can be adopted [4]. Memory unit is considered as one of the greatest consumers of power. For example, in the case of STRONG ARM controller, the cache consumed around 43% of the total power [20]

3.23 Exploiting parallelism

Different power management techniques are also in common for conserving power like using different power down modes, deactivating the part of the circuit which is not functional etc. Another attractive technique is to exploit parallelism which helps to reduce the clock frequency and supply voltage of the system. Pipeline techniques are now very versatile and a uni-processor system with pipeline is *n* times better to improve the power efficiency [21]. Loop unrolling techniques can aid in lowering the operating voltage and clock frequency [4].

The factors discussed are only some of the factors which has an impact on power consumption. It is also desirable to consider power consumption issues at different levels of abstraction. The following section gives a discussion on the power reduction techniques possible at different levels of abstraction.

4. DIFFERENT LEVELS OF ABSTRACTION AND POWER REDUCTION TECHNIQUES

The power minimization can be attempted at different levels of abstraction viz, at circuit level, device level, logic level, architectural level and at algorithmic level. A brief description about each level of abstraction is included in the following section.

4.1 Algorithmic level

In this higher level of abstraction, energy minimization is attempted by reducing the number of operations by analyzing the cost of each operation. It mainly includes: by comparing the arithmetic operations and logical operations, cost of memory accesses and incorporating ways to maximize the spatial and temporal locality of references. Also, there can be energy saving by considering the number representations and selection of different data representations like two's complement, sign magnitude etc. Another similar factor that can be beneficial is the bit length. Power reduction can be achieved at the cost of accuracy loss in the results [22].

4.2 Architectural level

In the architectural level, different techniques for instruction set design and performance increase by software approaches can be dealt with. The major in this group is exploiting the parallelism at instruction level, thread level and data level. Also, another widely implemented technique is pipelining; which can be useful for power minimization at the architectural level. Here, processor voltage is lowered to conserve power and then parallelism is exploited for enhancing the performance of the system. When the processor supply voltage is reduced, the speed of the system will come down. To supplement this, software techniques like, pipelining can be used to maintain the throughput of the system. There are many other software techniques that can be possibly incorporated into the architecture: speculation and prediction, loop unrolling, lop termination, procedural inlinig, product specific selection, global variable localization and assembly language in lining etc. to save energy and to enhance parallelism [22].

4.3 Logic and Circuit level

At the logic and circuit level, methods namely reducing the voltage swing, reducing the effective load capacitance can be done to reduce the power dissipation. Also, due to the variations in the signal arrival time to the input of a gate, glitches may occur which will add to power consumption. Therefore, logic restructuring and path delay balancing techniques can be used to reduce the glitches to save power [23]. Mostly, static logic circuits are the sufferers of glitches when compared to dynamic logics. Also, high probability switching nodes are operated at reduced speed or hided to reduce the switched capacitance to conserve power. Use of hybrid library consisting of static CMOS gates and pass transistors for synthesis are found beneficial for power reduction. Equivalent pin reordering can be used for power reduction along with transistor reordering technique. Clock power minimization can be achieved with techniques like: clock gating, low swing clocking and clock distribution minimization to save power dissipation [22], [24].

4.4 Device level

Threshold voltage reduction or dual threshold techniques can be used to reduce the leakage power dissipation [25]. Use of high threshold devices for non-critical delay paths and low threshold devices for critical path can result in standby power consumption reduction [21].

The factors which are mentioned above include only a subpart of a variety of parameters on which the power consumption of the processor depends. There can be more factors which may influence the power dissipation of an embedded processor. But, the above discussed parameters are a few, which are essential to be known and considered during the design phase or later. Most of the above referred elements are software techniques which can be easily adopted while developing the program code. The power conservation can be attempted well with software techniques rather than hardware approaches. Therefore, which technique is to be used for a particular processor architecture has to be identified aptly by the system designer to alleviate the power dissipation of the processor.

5. CONCLUSION

This paper presents an exhaustive study on the factors that affect the power consumption of an embedded processor. The different factors under both hardware and software approaches are presented. Also, a brief discussion on different levels of abstraction and the power reduction techniques are presented in the paper. The knowledge about the detailed analysis is beneficial for finding new techniques for power reduction of the processor. Some of the techniques are suitable for a class of processor architecture and the knowledge about these different techniques becomes vital to choose them appropriately.

References

[1.] A. Wolfe, “Intel Clears Up Post-Tejas Confusion,” VAR Business magazine, May 17, 2004. http://www.varbusiness.com/sections/news/breakingnews.jhtml?articleId=18842588

[2.] G. E. Moore, Moor’s law, Available: http://www.mooreslaw.org/

[3.] Sung I. Park, “The Design of Power Aware Embedded Systems,” PhD Thesis, University of California, Los Angeleles, 2003.

[4.] Gary K. Yeap, “Practical Low Power Digital VLSI Design”, Springer, London, 1998.

[5.] T. Kuroda, “Low-power CMOS design in the era of ubiquitous computers,” OYO BUTURI, vol. 73, no. 9, pp. 1184-1187, 2004.

[6.] K. Usami, M. Horowitz, “Clustered voltage scaling technique for low power design,” in Proceedings of the International Symposium on Low Power Design, pp. 3-8, 1995.

[7.] P. Pillai, K. G. Shin, “Real-time dynamic voltage scaling for low-power embedded operating systems,”in Symposium on Operating Systems Principles, pp. 89–102, 2001.

[8.] C. G. Lee, J. Hahn, Y. M. Seo, S. L. Min, R. Ha, S. Hong, C. Y. Park, M. Lee, and C. S. Kim, “Analysis of cache-related preemption delay in fixed-priority preemptive scheduling,” IEEE Transactions on Computers, vol.47, no.6, pp.700-713, 1998.

[9.] H. Ramaprasad, F. Mueller, “Tightening the bounds on feasible preemption points,”in RTSS 06*:* Proceedings of the 27th IEEE International Real-Time Systems Symposium, Washington, DC, USA: IEEE Computer Society, pp.212-224, 2006.

[10.] S. Nikolaidis, T. Laopoulos, "Instruction-level power consumption estimation embedded processors low-power applications," International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp.139-142, 2001.

[11.] T. Laopoulos, P. Neofotistos, C.A. Kosmatopoulos, S. Nikolaidis, "Measurement of current variations for the estimation of software-related power consumption [embedded processing circuits]", IEEE Transactions on Instrumentation and Measurement, vol.52, no.4, pp.1206-1212, Aug. 2003

[12.] R.P. Dick, G. Lakshminarayana, A. Raghunathan, N.K. Jha, "Analysis of power dissipation in embedded systems using real-time operating systems," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol.22, no.5, pp.615-627, May 2003

[13.] Li Tao, John Lizy Kurian, "Operating system power minimization through run-time processor resource adaptation," in Journal of Microprocessors and Microsystems, vol.30, no.4, pp.189-198, 2006.

[14.] Young-Si Hwang, Sung-Kwan Ku, Ki-Seok Chung, "A predictive dynamic power management technique for embedded mobile devices," *IEEE* Transactions on Consumer Electronics, vol.56, no.2, pp.713-719, May 2010.

[15.] S. Nikolaidis, T. Laopoulos, A. Chatzigeorgiou, "Developing an environment for embedded software energy estimation," Proceedings of the Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp.20-24, Sept. 2003.

[16.] Jose Nunez-Yanez, Geza Lore, “Enabling accurate modeling of power and energy consumption in an ARM-based System-on-Chip”, Journal of Microprocessors and Microsystems, vol.37, no.3, pp.319-332, May 2013.

[17.] D. Harris, “High Speed CMOS VLSI Design – Lecture 2: Logical Effort & Sizing,” November 4, 1997.

[18.] W. Shen, J. Lin, F. Wang, “Transistor Reordering Rules for Power Reduction in CMOS Gates,” in Proceedings of the Asia South Pacific Design Automation Conference, pp. 1-6, 1995.

[19.] M. Ito, D. Chinnery, K. Keutzer, “Low Power Multiplication Algorithm for Switching Activity Reduction through Operand Decomposition,” in Proceedings of the International Conference on Computer Design, pp. 21-26, 2003.

[20.] J. Montanaro, R.T. Witek, K. Anne, A.J. Black, E.M. Cooper, D.W. Dobberpuhl, P.M. Donahue, J. Eno, W. Hoeppner, D. Kruckemyer, T.H. Lee, P.C.M. Lin, L. Madden, D. Murray, M.H. Pearce, S. Santhanam, K.J. Snyder, R. Stehpany, S.C. Thierauf, “A 160MHz, 32-b, 0.5W, CMOS RISC Microprocessor,” Journal of Solid-State Circuits, vol. 31, no. 11, pp. 1703-1714, 1996.

[21.] T. Xanthopoulos, A. Chandrakasan, “A Low-Power IDCT Macrocell for MPEG-2 MP@ML Exploiting Data Distribution Properties for Minimal Activity,” Journal of Solid State Circuits, vol. 34, pp. 693-703, May 1999.

[22.] B. Moyer, "Low-power design for embedded processors", Proceedings of the IEEE, vol.89, no.11, pp.1576-1587, Nov 2001.

[23.] S. Nassif, “Delay Variability: Sources, Impact and Trends,” International Solid-State Circuits Conference, 2000.

[24.] T. Kitahara, F. Minami, T. Ueda, K. Usami, S. Nishio, M. Murakata, T. Mitsuhashi, "A clock-gating method for low-power LSI design," Proceedings of the ASP-DAC '98, Design Automation Conference 1998, Asia and South Pacific, pp.307-312, Feb 1998.

[25.] K. Usami, N. Kawabe, M. Koizumi, K. Seta, T. Furusawa, "Automated selective multi-threshold design for ultra-low standby applications," Proceedings of the 2002 International Symposium on Low Power Electronics and Design-ISLPED '02, pp.202-206, 2002.