

A Current Shaping Methodology for Low EMI Asynchronous Circuits

Dhanistha PANYASAK*, Gilles SICARD, Marc RENAUDIN

TIMA Laboratory, Concurrent Integrated Systems Group
46, avenue Félix Viallet – 38031 GRENOBLE cedex – FRANCE
<http://tima.imag.fr/cis>

e-mail : dhanistha.panyasak@imag.fr

(*)Ph.D. work in co. with STMicroelectronics - 850 rue Jean Monnet - 38926 CROLLES cedex - FRANCE

Abstract – The paper describes a design methodology for reducing current peaks in asynchronous digital circuits. Two existing methods influence this methodology, which deals with circuits at the architecture level. It spreads the current activity inside the circuit by controlling communication delays and events scheduling. A 4-taps FIR filter, synthesized in a 0.18 μ m CMOS technology, proves the methodology efficiency obtaining 20% peak current reduction and no significant area overhead before layout.

1. INTRODUCTION

Currently, digital circuits may contain many millions of transistors. In the same time, the clock can reach the frequency of GHz as well. In synchronous circuits, all signals switch at the clock frequency. Their simultaneity may draw tremendous amount of current in a very short time. It is well known that the more the peak current is high and brief, the more EM noise it generates. With the SOC (System On Chip) generation where digital and analog circuits cohabit, this parameter could become a crucial obstacle in preventing the system from its functioning.

Since they do not use clock signal, asynchronous circuits produce less electromagnetic radiation [5]. The purpose of this study is to minimize electromagnetic interference (EMI) phenomena in asynchronous digital circuits by using a specific methodology. This methodology applies at the design level during the synthesis stage. Current peaks are minimized by delaying signals from one another and scheduling data processing.

An overview of asynchronous circuits can be found in [1]. However the section 2 of this paper, we briefly introduce ones that interest us and the terminologies relevant to this work. Section 3 deals with Clock Skew Optimization [11] and Power-profiler [12] devoted to peak current reduction in synchronous circuits. The Current Shaping methodology [2] proposed in this work is presented in section 4 and detailed through paragraph 5 and 6. Finally, section 7 reports a 4-taps FIR filter implementation, which demonstrates the effectiveness of the methodology.

2. ASYNCHRONOUS CIRCUITS

In synchronous circuits, the clock controls the sequencing of processing and the communications between circuit elements. In opposite, in asynchronous circuit, also called self-timed circuit, handshaking communication protocols based on request and acknowledgement signals are in charge both communications and sequencing of processing (Figure 1).

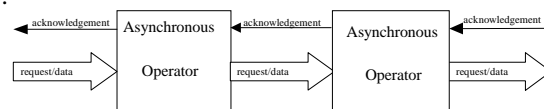
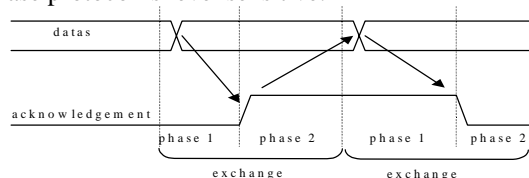


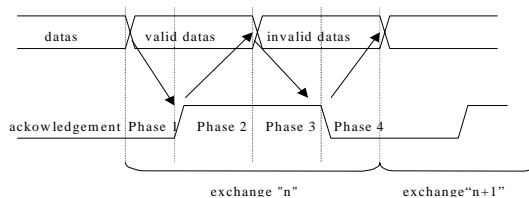
Figure 1 : Communication asynchronous operators

These communications use 2 or 4 phase protocols described in Figure 2 and Figure 3. The 2-phase protocol is events (signal edges) sensitive whereas the 4-phase protocol is level sensitive.



2 phase protocol. When the receiver is active, it treats the information and produces the acknowledgment signal (Phase1). Then, when the transmitter is active, it detects the acknowledgment signal and sends the new data (Phase2).

Figure 2-phase protocol



4 phase protocol. Transmitter is active and receiver is passive during the first phase. Then Transmitter is active, receiver is active in the second phase. In the third phase Transmitter is passive and receiver is active. Finally, transmitter is passive and receiver passive in the last phase.

Figure 3: 4-phase protocol

TAST (TIMA Asynchronous Synthesis Tool) [3], developed by the CIS group at the TIMA laboratory, is one of the tools, which generates asynchronous circuits of both protocols. This tool does not eliminate the current peak problem, which can arise inside the circuit. Our methodology is to fulfill the gap.

Among the various asynchronous circuit categories [1], our interest focuses on micropipelines.

Figure 4 presents general structure of micropipelines. Logic combinational blocks accomplish data processing. Latch stages control data, which move through them. In the control blocks, delays are equal to critical time of glue logic in order to match combinational bloc latency. For more details, the reader can refer to [3] where I. Sutherland introduces and describes micropipeline circuits.

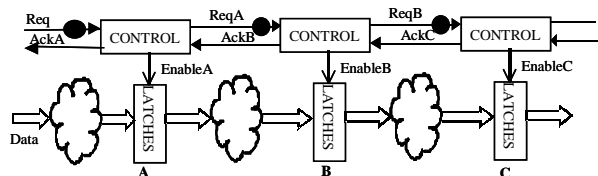


Figure 4: Micropipeline circuit

EM noise from signal switching is much more in synchronous circuit than in self-timed circuits. In [4], K. V. Berkel and al. deal with asynchronous circuit and their low electromagnetic emission. In their study, they compare frequency spectra of synchronous and asynchronous 80c51 microcontroller [5]. For several harmonics, the difference between synchronous and self-timed version reaches 30dB. Philips Semiconductors have exploited the low emission level of the asynchronous 80c51, using it in a pager. In this pager, asynchronous controller may be active during message reception.

In [6], a self-timed DSP demonstrates its low emission in comparison with the synchronous equivalent. The current spectrum of the first DSP provides a peak component 1.8 times lower than the one of the synchronous device. AMULET2e [7] is an embedded system which contains a 32-bit ARM-compatible asynchronous processor. Measures on its radiated emission show that it complies with the EMC standard constraints unlike the synchronous version.

Many asynchronous circuit experiments have shown that they induce less EMI than synchronous ones. However, industrial widespread of these circuits could be possible only if specific tools for asynchronous design are available. Furthermore, design method which reduce EMI in asynchronous circuits does not exist. Being aware of this, we elaborate the design method for low EMI circuit with the intention of integrating it in the CIS tool for synthesis asynchronous circuits: TAST.

3. REDUCING PEAK CURRENT IN SYNCHRONOUS CIRCUIT

In electronics systems, Spread-Spectrum Clock [9] is efficient to reduce EMI. A frequency modulates the clock and minimizes peak spectral current by distributing the energy of each fundamental and harmonic. As shown in [10], this methodology can be used to reduce EMI, radiated from a microcontroller, by using this kind of modulation in the system master clock. In this study we will use similar methodologies for designing integrated circuits as soon as their conception starts.

L. Benini and al. propose Clock Skew Optimization [11] in synchronous digital circuits. By this optimization, they reduce by 30 % the current peak of a circuit after layout. Knowing the circuit architecture at Register-Transfer-Level (RTL), a genetic algorithm calculates the clock arrival time at sequential elements (flip-flop) in each cycle. Clustering sequential elements extends the method and allows to drive several flip-flops by the same clock driver.

The assumption made on current activity is: the total current is a sum of current contributions represented as triangular shapes:

$$I_{tot}(t,T) = \Delta_c(t) + \sum_0^N \Delta_i^r(t,T_i) + \sum_0^N \Delta_i^f(t,T_i)$$

where t represents the time at which the total current is considered and T is the combined clock arrival times at flip flops. Supposing they do not have any control on combinational logic $\Delta_c(t)$ and considering only the current activity inside sequential element. Each sequential element (i) may be modeled by 2 triangles which correspond respectively to the current peak produced by a rising edge ($\Delta_i^r(t,T_i)$) and falling edge ($\Delta_i^f(t,T_i)$) of the clock. They evaluate the cost function, which approximates the current peak: $F(t) = \max_{t \in [0, Clkperiod]} \{I_{tot}(t,T)\}$ and reduce by finding the optimum clock schedule T .

TAST generates asynchronous circuits at RTL level description. For this reason the Current Shaping methodology refers to circuits at their architecture level as Clock Skew Optimization methodology. However, since communication protocols are local and depend on element latency, it is not possible to calculate clock arrival time at each element.

Using their Power-Profiler, Raul San Martin and John P. Knight suggest reducing peak power at the behavioral synthesis [12] level. Behavioral synthesis is a task which maps an abstract behavioral description of a circuit onto RTL level. During this mapping, an operation called *scheduling* determines the sequences of operations.

A library contains characteristics (silicon area, power consumption, delay) of different operators (multiplier, adder, etc.) and the plot of the average profile of their power dissipation. Different time slots divide the computation cycle. Depending on the operator assignation and the operation sequences, an algorithm calculates the total peak power at each time slot. Consequently, the power-profiler simultaneously finds the best operators and schedules operations in order to reduce the peak power. Application of the power-profiler on a DCT decreases the highest current peak by 66%.

At the RTL description level, the resources are known and fixed. But, it is possible to manage their concurrency by adapting communication between operators. We suggest reducing peak current by scheduling the current activity of concurrent blocks.

4. CURRENT SHAPING METHODOLOGY

Clock Skew Optimization [11] determinates arrival clock time at each sequential block in order to spread the current activity. Thus, it reduces the current peaks. Here we deal with asynchronous circuits. Their blocks communicate locally. We suggest shaping the global current by controlling these handshaking, i.e. request and acknowledgment arrival time. We determinate delays within communications for minimizing simultaneous signal switching. In this way, the Current Shaping methodology [2] controls the execution overlapping of concurrent blocks within the circuit:

- Identifying distinct blocks liable to be processed at the same time
- Distributing the activity of the blocks.

A model of the architecture of the circuit is necessary to identify the concurrent blocks. At the behavioral synthesis time, the Power-Profiler [12] chooses the operators and the sequences to perform in order to reduce the peak power. Our methodology is applicable to circuits at the architecture level. At this level, operators and their order are fixed. But, by taking account of the slowest operation, we may delay concurrent processing. Consequently, we insert delays without slackening the functioning of the circuit. Furthermore, we decompose the current activity of a block in a sequence of different current activities. An optimization algorithm uses these models to schedule the sub current activities. As a result of simultaneously delaying processing and scheduling current activities, the Current Shaping methodology distributes the power consumption. The study is composed of four fundamental points:

- Modeling the architecture
- Estimating architecture block latency
- Modeling the Current Profiles

- Shaping the global current by using an optimization algorithm

Using defined models (section mark 5), our methodology can be automated.

Taking care of circuits at their description (RTL), the methodology circumscribes conception time. The section of the methodology described further on is for asynchronous micropipeline circuits (Figure 4) using 4-phase protocol (Figure 3).

5. CIRCUIT MODELING

A graph representation of the circuit architecture makes the functioning analysis easier. Control Data Flow Graphs (CDFG) or Petri Nets graphs fit the analysis. The example of the Figure 5 maps the CDFG representing the behavior of an asynchronous circuit.

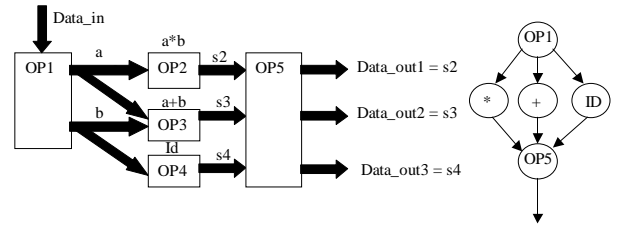


Figure 5: An asynchronous example (left) and its CDFG (right)

Graph representation provides useful information. In particular sequential, parallel compositions and choices are of interest. Figure 6 represents the patterns which have to be identified in a CDFG.

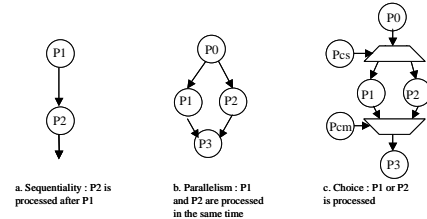


Figure 6: Composition patterns

For instance, a pattern of concurrency of 3 blocks is determined in the CDFG of Figure 5 (the multiplication, the addition and the identity are concurrent).

The composition pattern enables to automate the analysis of the graph representation of the circuit.

To model the current activity within each asynchronous operator, we study the 4-phase protocol. Control path and data path divide the micropipeline asynchronous operator (also called half buffer) (Figure 4).

The current activity of the operator may be decomposed in 3 phases (Figure 7).

- Phase 1: The valid data arrives from the precedent half buffer A, which sends the request signal (ReqA). The combinational logic computes the received data. The end of the computation marks the phase achievement.
- Phase 2: The enable signal switches and lets the data go through the latches. The half buffer B sends a request (ReqB) to the following half buffer C and an acknowledgment (AckB) to the previous half buffer A.
- Phase 3: The half buffer B wait for the acknowledge signal (AckC) of the following half buffer C and the reset of the request signal (ReqA) of the half buffer A. The enable signal switches and locks the latches. Then, the half buffer B stops sending the request signal (ReqB) to the half buffer C and the acknowledge signal (AckB) to half buffer A.

These 3 phases are displayed in a CDFG of 3 serial blocks of sub current activity (Figure 7):

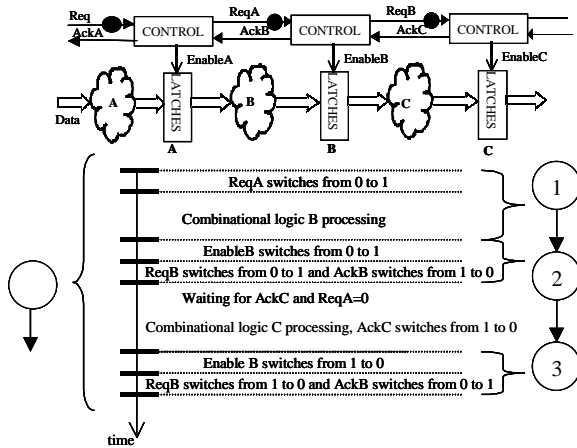


Figure 7 : Decomposition of the current activity in an half buffer and its CDFG

After analysing the communication between the half buffers, the model can be simplified (Figure 8).

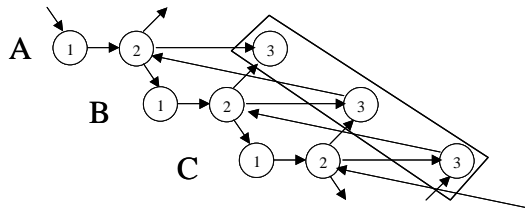


Figure 8: Phase 2 and 3 are synchronized

The phase 3 of the half buffer A and the phase 2 of the half buffer B are synchronized. The phase 3 of the half buffer B and the phase 2 of the half buffer C are synchronized. In order to simplify the model, we keep unchanged the protocol but we force the phase 3 since this place is synchronized with the phase 2.

Thus, Phase 1 and 2 are considered as serial blocks of current activity. The Figure 9 maps the CDFG of the characteristic pattern refined and simplified.

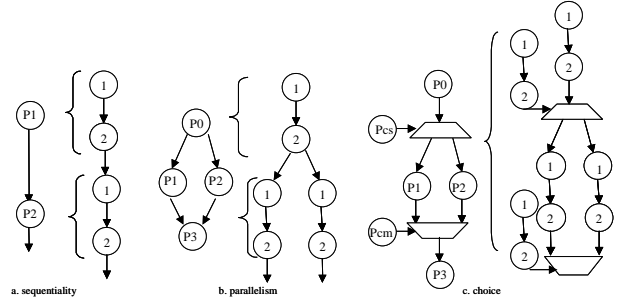


Figure 9 : phase composition CDFG

Such as L. Benini and al. [11], we use the triangle shape to roughly modelize the phases (Figure 10).

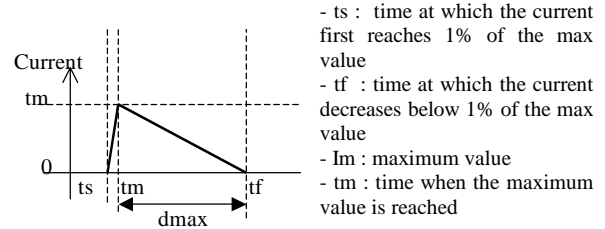


Figure 10 : Model of a current phase

The triangle, which represents the first phase $\Delta_i^{phase1}(t)$ contains the combinational logic activity. The triangles which represent second $\Delta_i^{phase2}(t)$ and third phase $\Delta_i^{phase3}(t)$ essentially contain communications current profile. The total current inside the circuit is :

$$I_{tot}(t) = \sum_0^N \Delta_i^{phase1}(t) + \sum_0^N \Delta_i^{phase2}(t) + \sum_0^N \Delta_i^{phase3}(t)$$

Consequently, each current activity of asynchronous operator has a CDFG model. In the graph representation of the circuit, each asynchronous operator may be replaced by this model, which refers to phases. These phases can be roughly represented by triangles. Relying on that we can estimate the current profile of the circuit.

6. CURRENT SHAPING

After identifying parallel handlings in the architecture model, we spread the activity of current by using refined current models.

Inserting delays permits to control the processing of the concurrent blocks. We can state that the current activity is $\Delta_i(t, Di)$ and the global current in the circuit is: $I_{tot}(t, Di) = \sum_0^N \Delta_i^{phase1}(t, Di) + \sum_0^N \Delta_i^{phase2}(t, Di) + \sum_0^N \Delta_i^{phase3}(t, Di)$ Where t is the time and D the set of delays.

Reducing current peak matches the cost function shown in [11]. It approximates the current peak:

$$F(t) = \max_{t \in [0, \text{CircuitLatency}]} \{I_{tot}(t, D)\}$$

We consider the latency of the slowest element in the concurrency as a constraint. During this latency the other concurrent blocks operate. We suggest slicing this time period into slots suitable for scheduling. Arbitrarily, we decompose the period in regular steps equal to the highest common factor of sub current blocks latency. Figure 11 details the slicing operation for the example of the Figure 5.

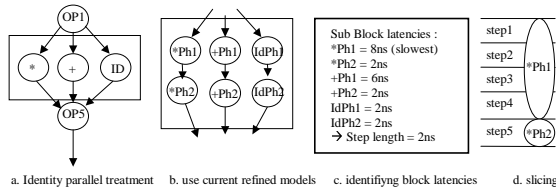


Figure 11: Applying the methodology to the example

Depending of the data and the type of operation the current activity in a phase of an operator may be higher than in another phase. For instance, a 32-bit multiplication corresponds to a heavy phase1 whereas an identity corresponds to a phase1 equivalent to a phase 2 or 3. According to the nature of the operator, we may attribute weight to each phase. This weight characterize the block current consumption. Moreover, knowing t_m , we determine in each step whenever the current peak is placed. Thus the weight of the phase is higher in these steps.

To schedule the sub current blocks, we opt for the Force Directed Scheduling (FDS) [13]. This algorithm minimizes the concurrency of current peaks that occur in one slot by distributing current activity among all slots. Considering concurrency of phases, and their weight, scheduling is applied in order to spread the current activity.

The Figure 12 shows the distribution obtaining after scheduling for the example.

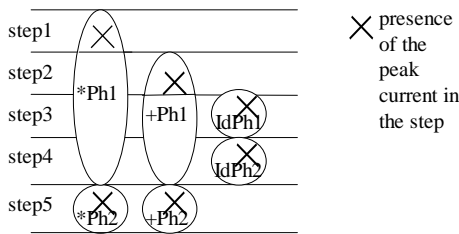


Figure 12: Scheduling the example

7. IMPLEMENTATION AND RESULTS

7.1 4-taps FIR filter

We designed and simulated synchronous and asynchronous 4-taps filters in HCMOS8 STMicroelectronics technology. In Figure 13, the synchronous filter is shown on the left and, on the right, the asynchronous equivalent in micropipeline architecture with 4-phase protocol [14].

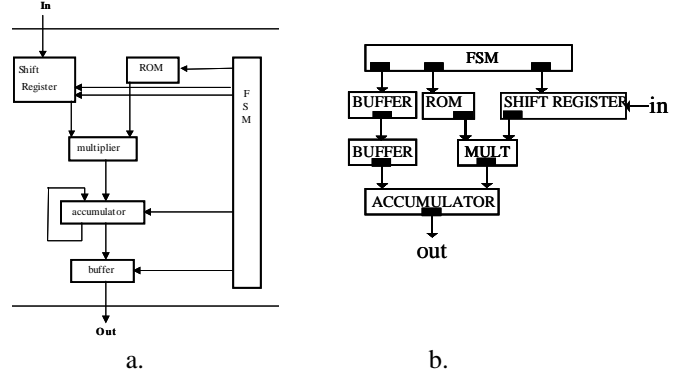


Figure 13: 4-taps FIR filter

a. Synchronous, b. Asynchronous micropipeline

The Design Compiler (Synopsys) extracts delays of combinational blocks. The current profiles of components were obtained by electrical simulation (Spectre). These profiles were employed to define current model of each component. Rom taps and stimuli were chosen in order to have an average current approximation of elements.

A CDFG was manually generated and all stages of the methodology, slicing, annotating, and scheduling, were processed.

7.2 Results

The Figure 14 shows current profiles of the synchronous and the asynchronous circuits.

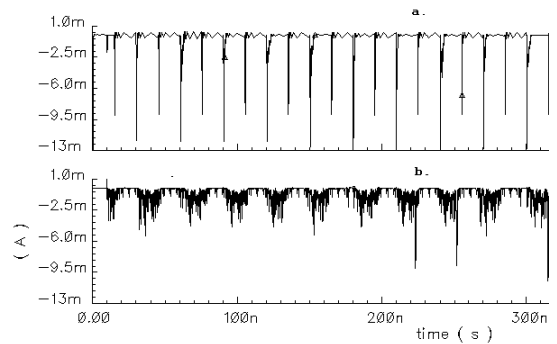


Figure 14: Current profiles of synchronous (a) and asynchronous (b) circuits

The synchronous circuit was simulated without clock tree. Despite any optimization, the maximum current was 22% lower in the asynchronous filter than in the synchronous one.

The Figure 15 compares current spectrum of the asynchronous circuit before and after using the Current Shaping methodology. The magnitude of the peak component was reduced by 20%.

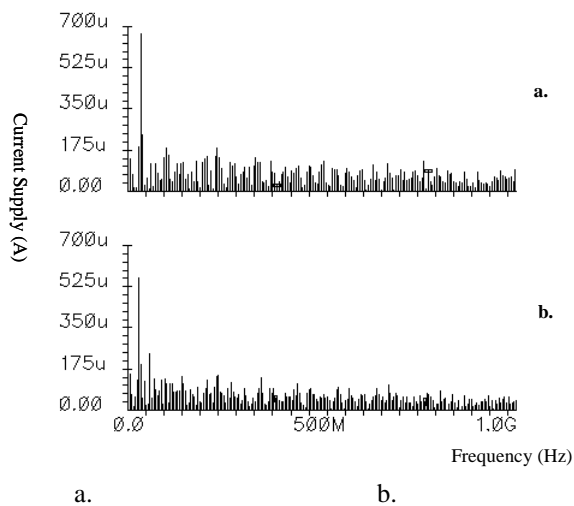


Figure 15: Current Spectrum of the asynchronous filter before (a) and after (b) using our methodology

8. CONCLUSION

A methodology of distribution of the current activity in asynchronous circuits was defined.

First stage, it analyses the graph representation of the circuit. Second stage, it allocates the current activity by using current models. The automation of this methodology is possible as the analysis of the graph uses defined composition patterns and a library can contain the current models.

This methodology has been validated on a 4-taps FIR filter delivering 20% reduction on peak component and no significant area overhead due to the fact that only small delays are added.

Dealing with circuits at their architecture level, the methodology applies to others categories of asynchronous circuits as well. In the future, we will extend the methodology to Quasi Delay Insentive asynchronous circuits.

9. REFERENCES

[1] M.Renaudin. "Asynchronous circuits and systems: a promising design alternative", *Microelectronic*

Engineering Journal, vol. 54, No 1-2, pp.133-149, December 2000.

[2] D. Panyasak, G. Sicard, M. Renaudin, "Shaping Current Profile of Asynchronous Circuits", Communication to ACID Workshop, Munich, Germany, January 28-29, 2002.

[3] A. Dindhuc, J.-B. Rigaud, A. Rezzag, A. Sirianni, J. L. Fragoso, L. Fesquet, M. Renaudin. "TIMA Asynchronous Synthesis Tools", Communication to ACID Workshop, Munich, Germany, January 28-29, 2002

[4] I. Sutherland, "Micropipelines", *Communication of the ACM*, Volume 32, No 6, June 1989.

[5] K. V. Berkel, M. B. Josephs, S. M. Nowick. "Scanning the technology: Applications of asynchronous circuits", *proceedings of the IEEE*, 87(2): 223--233, February 1999.

[6] H. v. Gageldonk, D. Baumann, K. van Berkel, D. Gloor, A. Peeters, and G. Stegmann, "An asynchronous low-power 80c51 microcontroller", In *Intl. Symp. Adv. Res. in Asynchronous Circ. and Sys.*, 1998.

[7] N. C. Paver, P. Day, C. Farnsworth, D. L. Jackson, W. A. Lien, and J. Liu, "A low-power, low-noise configurable self-timed DSP", in *Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp. 32—42, 1998.

[8] S. B Furber, J. D. GarsideP. Riocreux, S. Temple, "Amulet2e: An asynchronous Embedded Controller", *Proceedings of the IEEE*, 87(2): 223--233, February 1999.

[9] K.B.Hardin, J.T. Fessler, D.R. Bush, "Spread Spectrum Clock Generation for the Reduction of Radiated Emissions" *Proc. of IEEE Int'l Symp. on EMC*, Chicago (IL), pp. 227-231, 1994.

[10]D.A. Stone, B. Chambers, N.J. Ryan "Radiated EMI reduction from microcontroller system using spread-spectrum modulation of the system clock", *International Symposium on Electromagnetic Compatibility*, Rome, Italy. 249-52, vol.1, 17-20 Sept. 1996.

[11]L. Benini, P. Vuillod, A.Bogliolo, G. De Michelli "Clock Skew Optimization for Peak Current Reduction" *Journal of VLSI Signal Processing*16, 117-130, 1997.

[12]R.S Martin, JP.Knight. "Optimizing power in ASIC behavioral synthesis", *IEEE Design and Test of Computer*, 13(2): 59-70, summer 1996.

[13]P. Paulin and J.P. Knight, "ForceDirected Scheduling for the Behavioral Synthesis of ASIC's", *IEEE Transaction on Computer-Aided Design*, Vol. 8, No. 6, page (s). 661-679, 1989.

[14] M. Renaudin, J.-B. Rigaud, A. Dindhuc, A. Rezzag, A. Sirianni, J. L. Fragoso, "TAST CAD Tools ", Tutorial at the ACiD Summer School, Grenoble, France, July 2002.