Clock Distribution Strategy for IP-based Development

Rui L. Aguiar and Dinis M. Santos[1]
Universidade de Aveiro / Instituto de Telecomunicações
3810 AVEIRO, Tel:+351.34.370200, Fax:+351.34381128
email: ruilaa@ua.pt, dms@ua.pt

Key words: Clock Distribution, IP-based development, Controlled Delay Lines, Clock Jitter

Abstract: This paper presents a high-level clock distribution strategy for usage in a design-and-reuse environment. This strategy allows for controlled clock distribution across an arbitrary number of blocks through the usage of controlled delay lines. A new clock frequency multiplication structure optimised for this clock distribution strategy is finally proposed, since multifrequency clock support is highly desired.

1. INTRODUCTION

Increased integrated circuits sizes are presenting enlarged possibilities to circuit designers. Current estimates [1] indicate e.g.. microprocessor die sizes on the 4.5 cm^2 range in a three-year time frame, with nearly 100 million transistors. CMOS semiconductor technology has reached a point where the concepts of "system-in-a-chip" are becoming reality [2]. Systems previously mounted in printed circuit boards or large multi-chip modules are now to be implemented in a single chip. This trend brings increased technical benefits, as most of the input-output ports of previous circuits disappear, leading to lower power consumption with improved performance (often measured in working clock frequency).

Unfortunately, this same integration level is bringing increased problems to circuit designers, such as power and packaging issues [3]. Furthermore,

[1] This work has been sponsored partially by project Genclock, funded by Praxis XXI.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35498-9_57

L. M. Silveira et al. (eds.), *VLSI: Systems on a Chip*
© IFIP International Federation for Information Processing 2000

integrated system development is not possible due to increased design complexity. Circuit partitioning (and correspondent floorplanning) has been a reality for several years now: a team of designers separates the circuit in high-level blocks, and each block is developed independently. With increased system-in-a-chip developments, even this hierarchical development strategy has been showing its shortcomings. The complexity of CMOS circuits, with multiple functions, has grown in such an order that companies are finding hard to create development teams able to cover with realistic time-to-market constraints all the functions desired in the circuits.

Thus these systems are increasingly using "off-the-shelf" predesigned components, frequently coming from different sources. These components - often called Intellectual Property (IP) blocks, or Virtual Components (VC) - can be delivered in three different formats: soft IP, where a simple logical and synthesisable description is provided; firm IP, where this description is enhanced by some floorplanning and routing information; and hard IP, which provides complete physical models (e.g. layout) of the block. This approach is becoming popular even inside development teams, which are starting to handle their own blocks in a hard-IP format (and providing these blocks in this format to other groups for further design reuse) both due to company policy and to increasing needs for design reuse. Thus VCs will become progressively more common in future generations of chips.

This chip development strategy does provide fast development of complex circuits, especially as it fits naturally to current circuit partitioning methods. However, it brings a novel set of problems to clock distribution.

Even with fairly complex circuits, clock distribution is frequently handled as a problem by itself, covering the whole circuit development: it is common for a 2-million transistor circuit to be handled and analysed as a single clock distribution tree, resorting to modelling and complex simulation tools (frequently developed in-house). This clock distribution tree can present varying degrees of complexity [4], but nevertheless bounds on clock phase are usually provided [6] and maximum operating frequencies are derived for these circuits.

This approach is becoming inadequate as semiconductor technology evolves. For one-hand, as circuit sizes increase (and technology dimensions decrease), interconnect effects are becoming a dominant effect on the clock distribution tree. Furthermore, these are becoming (in percentage) more sensitive to process and environment parameter variation across the chip. Both effects increase the complexity of designing a clock distribution tree. On the other hand, no detailed control exists for the clock distribution inside reused hard-IP blocks. The system designer is effectively incapable of changing/controlling clock distribution in these blocks. This may also happen with most firm blocks, where changes in its clock distribution may

affect the block performance (it is usually recommended that detailed clocking information should be delivered with firm IP [2]). Thus an increasing number of IP-blocks with its own clock distribution tree will be used as a "black-box" block by design teams of future chips.

Two questions arise naturally in this type of environment: i) how to distribute the clock to these blocks such that with minimum jitter and skew is achieved, and ii) how to handle the effects of these hard-IP blocks in terms of jitter/skew.

This paper presents a clock distribution strategy that covers these two issues. We present a high-level clock distribution architecture independent of the individual clock distribution trees in each block in Section 2. This strategy allows for precise multi-point synchronization. In Section 3, IP blocks are characterized in terms of "clock uncertainty", and this measure can also be applied to the clock distribution mechanism being used. From these values, maximum clock uncertainty bounds and simple design specifications can be achieved. This strategy can be applied in a hierarchical approach, allowing for further reuse of the circuit. Section 4 expands this approach in order to provide for local frequency multiplication, due to power consumption considerations.

2. HIGH-LEVEL CLOCK DISTRIBUTION

2.1 Clock Distribution Model

The traditional design paradigm to date is the synchronous design. This is a deterministic approach, robust and easy to implement. It is based on a state-machine concept, where states are represented by register output values, and state transitions are controlled by the results of combinatorial logic. Latching data into a new state is controlled by a set of clock signals.

Although asynchronous design has gained popularity in the last years [7], it is not the focus of this paper, as currently self-timed methods are not often used in complex circuits in a design-and-reuse environment. Self-timed approaches present some properties that seem to hold much promise for future complex chips (namely low power and heterogeneous timing capabilities) but design methodologies and supporting tools still need to evolve in order for their widespread usage in very complex systems.

Thus our synchronous reference model assumes synchronization units (I/O registers) at the input and output of every IP-block. (If synchronization points are not available, it is not possible to evaluate clock delay without detailed knowledge of circuit functions and implementation). A multi-point clock distribution tree then drives these blocks (Fig. 1).

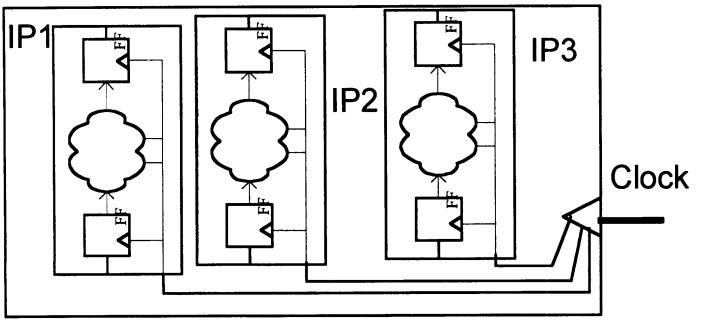

Figure 1. Reference model for clock distribution, with three hard-IP components

Synchronous design methodology forces the time reference (the clock) to be constrained by the slowest path in the whole circuit [6]. The critical path delay (T_c) can be related to the smallest clock period possible ($Tclk_{min}$) by:

$$Tclk_{min} - \delta_{max} = Tgate_{max} + Tset_{max} + Tp_{max} = T_c \quad (1)$$

where $Tgate_{max}$ is the maximum propagation delay in the critical path gates, $Tset_{max}$ is the maximum setup and hold times of the registers, and Tp_{max} is the maximum signal propagation time in the interconnects of the critical path (which can be estimated by several propagation models); δ_{max} is the modulus of maximum clock uncertainty (due to jitter and/or skew). Although more complex clock strategies may be used [4, 6], such as e.g. cycle borrowing, two-phase clocking or pipelining, their applicability is either infeasible at the top design partitioning level we are considering, or can be readily incorporated in our discussion.

Clock phase differences between registers inside each IP block are neglected at this stage. All blocks are considered synchronization domains, with null clock phase difference between registers (The next section will remove this assumption). Then the problem can be stated as the delivery of a clock signal at the input of each block in such a way as to assure the same phase reference at all synchronization points, or, using expression (1) to minimize clock uncertainty δ_{max}. Note that as the clock uncertainty increases, clock frequency will have to be decreased.

Two different (but interrelated) issues affect clock uncertainty: skew and jitter. Skew can be defined as the deterministic delay difference between two supposedly synchronous signals, caused by such effects as different interconnect line lengths or different driver strengths. Jitter is the random variation of clock phase around its average point, caused by effects such as circuit noise or (dynamic) load variations on clock propagation lines.

Both effects are disadvantageous for clock synchronization purposes, and they represent only different aspects of clock uncertainty. For clock synchronization purposes both perfect timing between different synchronization points and perfect regularity in clock period is sought (unless complex clock cycle management is deliberately used).

Traditional clock distribution resorts to the design of a clock tree network [4, 5], or a two level tree network, with hierarchical clock buffering schemes and local skew equalization [6] with careful interconnect routing. Nevertheless, these approaches suffer from significant process and environment problems, even when they theoretically allow for perfect skew cancellation.

2.2 Clock Feedback Design Philosophy

Clock distribution across several IP-based blocks should minimize both skew and jitter, as no information on the content of each block may be available. Traditional approaches look at clock distribution essentially as a passive problem. The clock is distributed from a source towards the sinks, without any further control. Clock networks are designed in such a way that minimize skew, but no dynamic control is done over this parameter. In this section we propose a radically different way of approaching these ideas, where feedback is an important part of the clock tree behaviour. The clock is still distributed to a set of clock drivers (sinks), but this distribution now uses feedback to set remotely the clock phase at a specific point. Several synchronization domains can be set in phase, regardless both of the number or characteristics of the domains (IP-blocks) and of interconnection lengths. The concept is simple and several clock distribution implementations have already presented some sort of feedback [8, 9, 10].

This approach requires the interconnect system (which may comprise both passive interconnect lines and driving buffers) to incorporate lines with controlled Delay Elements (DEs). Several structures for feedback can be used, according to the number of blocks to control, the size of the interconnections and the clock frequency. In our clocking strategy we adhere to a variant of a generic approach, initially proposed in [8].

Figure 2 shows a three-block active clock distribution network, where three points (Φ_1, Φ_2, Φ_3) will present the same phase, regardless of the number of DE in each interconnect and on the total interconnect delay - if the control blocks are able synchronize properly. Each one of these synchronization points can now act as a clock reference point for an IP-block, as in the situation depicted in Fig. 1. The clock network uses controlled delay lines with an even number of DEs, and establishes a loop with a symmetrical path between the clock synchronization reference and

each synchronization point. The synchronized points are located at the middle of these loops and will be in phase if the DEs are controlled to provide a line delay with an odd-or-even [8] number of clock periods in the total interconnection. Phase accuracy at the synchronized points depends mainly on the quality of the control blocks, on their matching, and on the matching of both clock interconnect lines for each block [8]. Control blocks should be placed together, as illustrated in Fig. 2. Both clock interconnect lines (to and from) between the clock reference point and each of the synchronization domains should be routed through the same routing channel, as the phase accuracy at the synchronization points depends on the symmetry of these two interconnections.

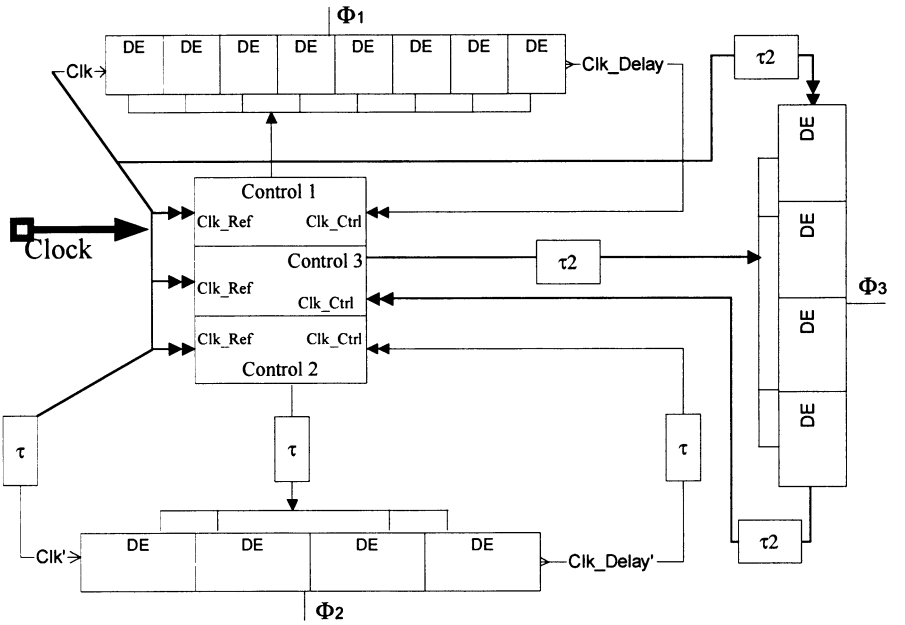

Figure 2. Three-point active clock distribution network. "Clock" is the clock reference point, and Φ_1, Φ_2, Φ_3 are the synchronised points

2.3 Controlling Delay Lines

Some sort of phase comparison and control signal is required for setting the delay of the active line. This control should be done with digital techniques, which are more resilient to the inevitable substrate and power supply noises of high frequency digital systems. Then, using an active clock distribution, total clock interconnect delay can be expressed as:

$$\Delta_{clk} = T_{DE} + T_{clk\_prop} = M.(k.\delta_{DE} + T_{DEmin}) + T_{prop} \quad (2)$$

with Δ_{clk} as the total delay in the line, T_{DE} the (controlled) total delay in each DE, and T_{prop} the delay in the connection lines; M is the number of DEs in the line, T_{DEmin} is the minimum delay of the DEs and δ_{DE} is the minimum (controlled) delay step possible; *k* is the (integer) control variable. From (1), if no clock uncertainty exists then, at maximum operating frequency we get:

$$\Delta_{clk} = n * (Tgate_{max} + Tset_{max} + Tprop_{max}) \quad (3)$$

but this would require stringent (and generically unattainable) relationships between *M*, *k* and δ_{DE}. (In a first approximation, the integer value of *n* is not relevant.) Thus, using any digital control in this type of system will *require* a given clock uncertainty, as the total line delay will not be an exact multiple of the *maximum possible* clock period. This clock uncertainty is not equal to this minimum delay step δ_{DE}, but depends on other design parameters.

In practice the clock uncertainty Δ_{tg} will be the design target specification. It will be chosen as the maximum value that is able to fulfil the system design specifications for all parameter (viz. process and temperature) conditions.

The following methodology can be used for the design of such an active clock distribution system (the numbers achieved by this process provide design targets to be exceeded by the synchronization system):

a) Define the target clock uncertainty $\Delta_{tg}(= M.\delta_{DE})$; this will be related with the quality of the phase comparator in the control blocks.
b) Evaluate maximum (Δ_{clkMax}) and minimum (Δ_{clkmin}) possible time delays in the interconnect system, in function of interconnect size, number of clock buffers required to maintain sharp clock transitions and parameter variation.
c) Calculate the maximum control value, the number of DEs and the minimum delay step required with equation $\Delta_{clkMax} - \Delta_{clkmin} = \Delta_{tg}.k_{max}$ (where k_{max} is the maximum control value). The structure chosen for the DE will place bounds on "reasonable" values of *M* and δ_{DE}.
d) Evaluate the synchronization period (*n*) using (2) and (3) above, and keeping in attention that all control blocks should operate on the same parity (*n* is required odd or even for all blocks) [8].

Thus we will have a control system that can assure a maximum clock uncertainty in the phase relationship of the clock delivered at each IP-block. Unfortunately, clock uncertainly will be further increased by noise in the controlled delay line (including its delay elements). Several generic analyses have been made [11, 12], mostly for voltage controlled delay lines. It is often possible for the active control circuitry to be designed such that the uncertainty introduced by noise is much smaller than the uncertainty inherent to the digital delay control used.

3. IP BLOCK CHARACTERIZATION

In the previous section we discussed how to establish a multi-point clock distribution network with a given clock uncertainty, assuming that each controlled block was a perfect synchronization domain. In reality a hard-IP block may have an arbitrary complexity (e.g. a microprocessor), and thus may have its own uncertainties related with its internal clock distribution.

Thus a hard-IP block should be characterized by two parameters: a) its critical path delay; b) the maximum clock uncertainty that may appear between the I/O registers and the clock reference point, in all possible operation parameters. (Note that in our synchronous model, both inputs and outputs of IP-blocks are directly connected to registers: only phase differences between its clock reference point and the input and output registers are then relevant.)

The critical path delay is naturally required in order to quantify the maximum operating frequency. The critical path delay will be the maximum value of the critical paths of each IP-block ($T_c(i)$) and of the interconnection logic block ($T_c(int)$ (this block includes all logic implemented for the interconnection of the several IP-blocks, and is usually specifically designed by the global system development team):

$$T_c = \max\{T_c(1), T_c(2), \ldots. T_c(n), T_c(int)\} \quad (4)$$

The critical path delay of the interconnection block has to consider the clock uncertainties in each IP block ($\delta_{max}(i)$). Thus:

$$T_c(int) = Tgate_{max} + Tset_{max} + Tprop_{max} + \max\{\delta_{max}(1),..\delta_{max}(n)\} \quad (5)$$

Current development tools already supply the critical path of a block, and advanced clock development tools provide for bounds on clock distribution phase and on clock propagation delay across interconnect lines (e.g. [6]). These tools are able to immediately provide the values above referred.

This approach creates a hierarchical development structure for clock distribution. Each block handles its clock distribution network in accordance with the techniques that finds more appropriate. For further design usage, the IP-block displays two properties (critical path and I/O clock uncertainty) that are used for the implementation of the clock distribution network of any circuit that uses this block. These values are then recursively used in the evaluation of the clock characteristics of this new circuit - which can become a new virtual component (IP-block) by itself.

4. MULTIPLE CLOCK FREQUENCIES

Previous sections have discussed the characterization of each hard-IP block in terms of clock performance, and a clock distribution strategy able to achieve a given clock uncertainty. In those sections the idea of a single clock frequency across the whole system was present.

However, clock distribution is power expensive. This is more significant as clock frequencies and interconnection distances increase. Future clock frequency estimates present quite different values, depending on interconnection distance [1]. Trying to distribute the same high-speed frequency across increasing distances brings propagation, power consumption and noise problems.

Moreover, there is no intrinsic need for the whole system to be driven by the maximum frequency. Several sub-blocks could use higher frequencies, but for power consumption and noise considerations, each sub-block should be attacked with as small a frequency as possible to achieve global frequency targets [13]. Global system synchronization would still be maintained at the maximum required clock frequency (which could be much lower than the maximum clock frequency in some of the IP-blocks). This transposes traditional digital development frameworks (e.g. [14]) for the "systems-in-a-chip" environment.

The clock strategy discussed in the previous sections allows for the inclusion of independent clock generators in each IP-block. These clock generators could provide for clock multiplication internally, and thus allow for much smaller clock frequencies to be distributed across the whole circuit. Both clock frequency and phase adjustment has to be performed in order for minimum clock uncertainty to be achieved in the IP-block.

Our proposed clock distribution strategy uses a novel synchronization method, depicted in Fig. 3. The internal clock generator is coupled to the clock reference point (Φ_1), as a phase detector controls the output of the clock generator (Φ'_1) in function of this point. The centralized control block adjusts the overall clock phase (as discussed before), controlling several DEs in the synchronization loop. However, some of the DEs in the loop are not under the control of this unit. The local phase control block assures that the multiplied local clock is "in phase" with the global clock signal. It does this by changing the delay in some DEs of the synchronization loop, in such a way that it compensates the timing delay of the clock multiplication block. For the synchronization loop, these locally controlled DEs act as another interconnect delay. As long as this phase control affects symmetrically the synchronization loop, the middle point (Φ_1) still keeps its phase reference, and will be the same across multiple similar connections. Although frequency multiplication is done, the internal clock is "in phase" with the

external distributed clock (i.e. the local high frequency clock has transitions at the same instants than the slower global clock at the reference point). All phase errors depend mainly on the performance of the phase control systems, and not directly on parameter variation. Thus this structure, associated with the clock distribution mechanism already discussed, allows the synchronization of multiple domains with multiple clock frequencies, while a precise common time reference is still assured across all domains.

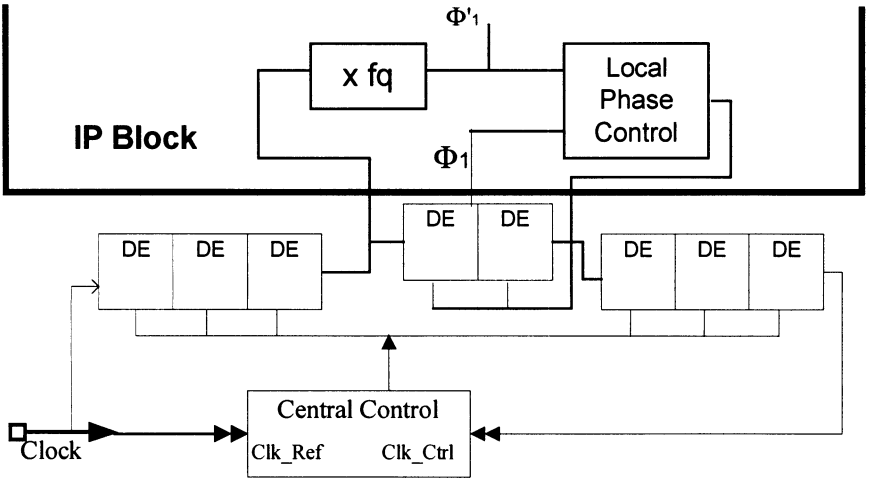

Figure 3. Local clock multiplication embedded in the synchronisation loop

In terms of expressions (4) and (5), the introduction of multiple clock frequencies does not present major changes, as simple additive processes may approximate jitter mechanisms on DLLs (for typical design parameters [15]). The major difference is that the clock uncertainty caused by any frequency multiplication has to be considered in the calculation of the clock uncertainty of that IP-block. However, this effect should only be evaluated at the global clock transition instants, when it should be small and dependent mainly on the IP-block local phase detector characteristics.

5. CONCLUSIONS

Increased usage of Virtual Components (or IP-blocks) is a consequence of the raise in complexity of current CMOS microcircuits. These components are often provided in a hard-IP format, with little information on its internal clock distribution scheme. This increased complexity is furthermore being translated as greater circuit dimensions, while technology features decrease.

All these aspects bring new problems to traditional clock distribution strategies, as large blocks are placed inside a circuit without any control of its clock distribution while, simultaneously, interconnection lengths increase.

We have presented a clock distribution strategy adequate to these new environment characteristics. This strategy uses active delay lines as top-level clock distribution, assuring bounds on phase uncertainties for each IP-block. Furthermore, two simple parameters are associated to each block (critical path and clock uncertainty). These parameters present working bounds for system engineers to incorporate these components in new systems, creating a hierarchical development strategy of arbitrary complexity.

This strategy is flexible enough for allowing multi-frequency operation in the same system, reducing power consumption. We extended this strategy with a frequency multiplication method optimised for this clocking methodology. This method adheres to a global phase reference framework while presenting an internal frequency multiple of the reference clock.

6. REFERENCES

[1] Semiconductor Industry Association, *The International Technology Roadmap for Semiconductors, 1998 Update*, 1998.

[2] VSI Alliance, *Architecture Document, version 1.0,* 1997.

[3] Dennis Herrel, "Power to the package", *IEEE Spectrum*, n. 6, July 1999, pp.46-53.

[4] Eby G. Friedman, " Clock Distribution Networks in VLSI Circuits and Systems", *Clock Distribution Networks in VLSI Circuits and Systems*, IEEE Press, 1995, pag.1-36.

[5] H. Fair, D. Bailey, "Clocking Design and Analysis of a 600MHz Alpha Microprocessor", *IEEE Int. Solid-State Circ. Conf. ISSCC'98*, pp.398-399, 1998.

[6] Jon Rabaey, "Digital Integrated Circuits, a Design Perspective", Prentice Hall, 1996.

[7] C.H.Van Berkel, M. B. Josephs, S.M. Nowick, "Scanning the Technology – Applications of Asynchronous Circuit", *Proceedings of the IEEE*, vol. 87, n. 2, Feb. 1999, pp.223-233.

[8] Rui L. Aguiar, Dinis Santos, "Wide-Area Clock Distribution Using Controlled Delay Lines", *5th IEEE Int. Conf. on Electr., Circ. and Syst., ICECS'98*, pp.63-66, Sep. 98.

[9] G. Geannopoulos, X. Dai, "An Adaptive Digital Deskewing Circuit for Clock Distribution Networks", *IEEE Int. Solid-State Circ. Conf., ISSCC'98*, pp. 400-401, 1998.

[10] H. Sutoh, and K. Yamakoshi, "A Clock Distribution Technique with an Automatic Skew Compensation Circuit", *IEICE Trans. on Electronics*, E81-C, n. 2, Feb. 98.

[11] B. Kim, T.C. Weingandt, P.R. Gray, "PLL/DLL System Noise Analysis for Low Jitter Clock Synthesizer Design", *IEEE Int. Symp. on Circ. and Syst., ISCAS'94,* 1994.

[12] Rui L. Aguiar, Dinis Santos, "Modelling Charge-pump Digital Delay Locked Loops", *6th IEEE Intl Conf. on Electr., Circ. and Syst., ICECS'99*, Cyprus, Sep. 99.

[13] A.P. Chandrakasan, S. Sheng, R. W. Brodersen, "Low-Power CMOS Digital Design", *IEEE Journal of Solid-State Circuits*, vol. 27, n. 4, pp.473-483, Apr 1992.

[14] F. Anceau, "A Synchronous Approach for Clocking VLSI Systems", *IEEE Journal of Solid-State Circuits*, vol. 17, n. 1, Feb 1982.

[15] Rui L. Aguiar, Dinis Santos, "Simulation and Modelling of Charge-pump Digital DLLs", *42nd IEEE MidWest Symp. on Circ. and Syst., MWSCAS'99*, Las Cruces, Aug. 99.