

# A Digitally Controlled Oscillator for Fine-Grained Local Clock Generators in MPSoCs

Guilherme Heck, Leandro S. Heck, Matheus T. Moreira,

Fernando G. Moraes and Ney L. V. Calazans

Pontifícia Universidade Católica do Rio Grande do Sul - Porto Alegre, Brazil

{guilherme.heck, leandro.heck, matheus.moreira}@acad.pucrs.br,

{fernando.moraes, ney.calazans}@pucrs.br

**Abstract**—The evolution of technology into deep submicron domains leads to increasingly complex timing closure problems to design multiprocessor systems. One natural alternative is to resort to the globally asynchronous, locally synchronous paradigm (GALS). This work proposes a generic architecture for very low power- and area-overhead local clock generators (LCG) to drive individual modules of a multiprocessor, *e.g.* network on chip routers and other elements. As main original contribution it details the design of a digitally controlled oscillator (DCO), the core of the clock generator architecture. This DCO can produce at least 16 distinct frequencies between 117 MHz and 1 GHz and supports clock gating and glitch-free frequency changes. Its design is robust to PVT variations and takes less than 1000  $\mu\text{m}^2$ .

**Index Terms**—Local clock generator, LCG, PVT, MPSoCs.

## I. INTRODUCTION AND RELATED WORK

Network-on-chip (NoC) based multiprocessor systems-on-chip (MPSoCs) can deal with semiconductor market evolving requirements, which has for many years capitalized on the synchronous paradigm. The latter assumes the use of a single clock signal to control the design sequencing of events, which drastically reduces complexity. However, in modern technologies the assumption of global synchronicity in MPSoCs leads to problems in timing closure, as process variability stresses design constraints to unmanageable levels.

Today, the globally asynchronous, locally synchronous (GALS) design paradigm [1] is a practical design option for MPSoCs, allowing the use of synchronous elements that communicate asynchronously at system level. But in such a system, there are at least two new problems to solve: system partitioning into frequency domains and the choice of a clock generation scheme. This work addresses the second problem. It proposes a generic architecture for local clock generators (LCGs) targeted to GALS NoC-based MPSoCs. The architecture focus on low power and area, to minimize overheads. It enables the use of up to one distinct clock generator per MPSoC module, either an intellectual property core (IP) or an NoC router.

As main original contribution the paper details the design of a digitally controlled oscillator (DCO) for the architecture. The DCO can produce any of 16 distinct operating frequencies

between 117 MHz and 1 GHz, with process, voltage and temperature (PVT) compensation capability, clock gating and glitch-free frequency switching support. Traditional techniques to design clock generators employ PLLs or similar structures, to achieve highly precise phase and frequency, leading to high costs in power and area. Our approach trades off precision, power and area. External controllers can add fidelity if needed.

Although the ITRS [2] states that intrachip asynchronous global signaling will be increasingly needed, few works address DCOs or LCGs for MPSoCs. In [3], authors propose a standard cell-based DCO with clock gating support using inverters and NANDs. Results show a 22% uncertainty in the generated frequency but process variations can produce up to 70% deviations. References [4] and [5], propose the suppression of clock pulses to reduce a base frequency. However, the maximum supported operating frequency determines timing constraints that must be respected globally, which may be unreasonable for MPSoCs. Albea et al. [6] propose an LCG for DVFS on GALS MPSoCs which uses a DCO with R-2R binary coded networks for DACs. It can produce 256 different clock values, but R-2R DACs are based on voltage references and on DVS circuits, being this dependence undesirable. The gap between corner cases can achieve around 3.5 GHz on a given frequency. Besides, the approach does not support clock gating. Höppner et al. [7] propose a clock generator that produces 83 distinct frequencies between 83 MHz and 666 MHz for IPs and 2 GHz and 4 GHz frequencies for NoC routers. Although it allows different frequencies for IPs and routers, the circuit has high cost in complexity, area and power.

## II. THE LCG ARCHITECTURAL ASSUMPTIONS

The LCG proposed herein addresses MPSoCs like those discussed in [8], [9]. Fig. 1 presents the target MPSoC structure. A router R and an IP compose a processing element PE. Routers and/or IPs have an operating frequency provided by the low power-, low area-overhead LCG that enables using fine-grained clock domains. Here, all LCGs employ an external low frequency reference clock (*ref\_clk*) to provide overall control of the locally generated frequencies. The *ref\_clk* signal drives the LCGs only, consuming negligible power and requiring no skew control, as signal phase is irrelevant. Routers and IPs require synchronization interfaces [10] in all interconnection points, *e.g.* all ports of a router. Other clock schemes are also allowed, where groups of routers and/or IPs are in a

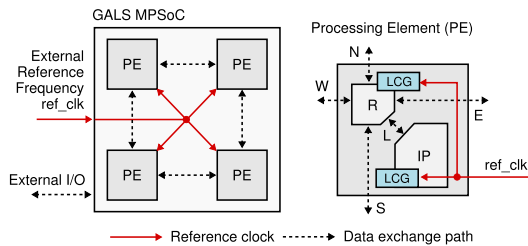


Fig. 1. Block diagram for the target MPSoC structure and detailed view of a processing element PE. LCGs are the local clock generators.

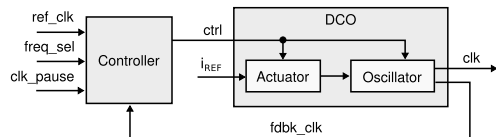


Fig. 2. The proposed LCG architecture.

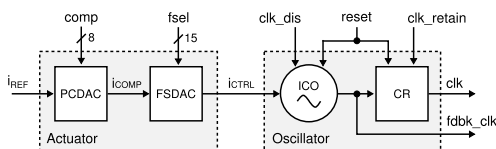


Fig. 3. Block diagram and block contents for the DCO.

same frequency island. This relaxes synchronizing interface requirements, but may prevent design space exploration of fine-grain frequency domains, and is application specific.

### III. THE PROPOSED LOCAL CLOCK GENERATOR

The LCG structure comprises a Controller and a DCO, which in turn comprises an Actuator and an Oscillator. Fig. 2 shows its architecture, which generates the adjustable clock (clk). The ref\_clk signal furnishes a time base for the generation process, while signals freq\_sel and clk\_pause respectively allow choosing the clock frequency and requesting the clock to pause. The Controller produces a set of binary control signals, generically designated ctrl in the Figure.

The Actuator receives part of ctrl and a current source  $i_{REF}$  output, a reference to produce the main frequency control signal. This module compensates PVT variation effects. The external Controller receives as feedback the generated frequency, and verifies if it corresponds to the expected value. If not, it reconfigures the DCO to reach the requested operating frequency. Typically, NoC routers do not need a precise clock frequency to operate; their usual role is to deliver packets in a rate close to that of the IP injecting data in it. Each router can individually select the best available frequency to operate in. Thus, the Controller for routers LCGs can be extremely simple, reducing power and area overheads. The Controller for IPs LCGs is typically more complex, to improve precision. This paper will not focus the design of the Controller, only on the DCO design.

### IV. THE DCO DESIGN

The LCG output frequency range is set between 117 MHz to 1 GHz, to cover typical operating frequencies of embedded

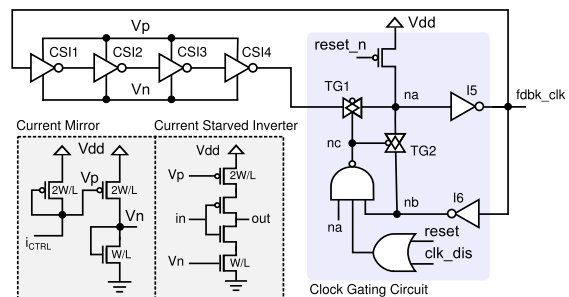


Fig. 4. Circuit diagram of the ICO. The current mirror feeds the current-starved inverters (CSI). The Clock Gating Circuit shares the I5 inverter with the oscillator.

systems. Coupled to an assumption to use 16 frequency values, this determines frequency steps of around 59 MHz, considering the ideal use of a constant difference between any two successive frequency steps. The number of distinct frequencies resulted as the maximum value in an analysis of the noise and supply voltage margins offered by the chosen technology (65 nm). This assumes the use of minimum-size transistors in the DCO, to reduce the area overhead. To enable significant reductions in power when idle, the DCO embeds a clock gating capability to stop the oscillation.

Fig. 3 depicts the proposed DCO structure. A set of external signals commands the frequency generation process. These correspond to the decomposition of the ctrl control bus of Fig. 2 in five components: comp, fsel, clk\_dis, clk\_retain and reset. Signal comp acts on the reference current source to compensate PVT variations; fsel selects the DCO frequency after PVT calibration, helping in the generation of the current that commands the Oscillator,  $i_{CTRL}$ . Clock gating is achievable in two different ways: *i*) activating clk\_dis freezes both the feedback clock (fdbk\_clk) and the clock (clk) when these are 0, opening the oscillation ring; *ii*) activating clk\_retain holds just the clock signal level (clk), to allow the Controller to adjust the DCO without interfering in the driven circuit.

The DCO produces a clock signal from its digital inputs, which control the delay of a ring oscillator. It contains: *i*) a DAC to control PVT variations (PCDAC); *ii*) a DAC to handle frequency selection requests (FSDAC); *iii*) a current-controlled oscillator (ICO), to produce the output frequency; and *iv*) a clock retention circuit (CR), to disable clock propagation to external entities, when requested. The following Sections detail the design of the Oscillator, the Actuator and their internal components.

#### A. The Oscillator

Fig. 4 details the structure of the proposed oscillator (ICO). It relies on a ring formed by current-starved inverters (CSIs) [11]. Besides providing small area footprint, this structure allows manipulating oscillation time constants of the ring elements by using current instead of voltage control, which improves stability [12]. Given the power supply, the transistor nominal current and input capacitance for the technology, it is possible to determine the appropriate number of delay stages for the selected frequency constraints. In our case, five delay stages were enough to achieve the required frequency range.

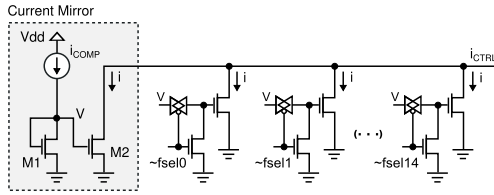


Fig. 5. Structure of the DCO Frequency Selector DAC (FSDAC). Transistor M2 is always on, and fsel controls the current mirrors.

One original contribution added to the well-known current-starved architecture appears in the fifth stage: a glitch-free Clock Gating Circuit. An energy-efficient way to stop a clock consists in opening the ring. To avoid stability problems, modifications on the last stage of the oscillator allow it to operate under clock gating as a latch, using a pair of transmission gates, as Fig. 4 shows. The ring opens when TG1 opens and TG2 closes. The memory structure formed by inverters I5, I6 and TG2 holds the output logic level. The NAND gate freezes the output clock signal in 0 without glitches, when asserting  $clk\_dis$ , na and nb. Note that I6 is a high-drive inverter. If na starts going to 0 when signal  $clk\_dis$  activates (a condition that could lead to a glitch) there are two cases to consider. In the first one, na goes fast to 0, which will delay the clock gating to the next cycle. In the second case, na is slower than  $clk\_dis$  and its propagation to node nc, which will close TG2 and make na return to 1. In this way, clock gating is free of hazards. The Clock Gating Circuit allows keeping the output clock signal and enables DCO power reduction, by setting the comp and fsel signals to minimal values.

To avoid undesirable harmonic frequencies by possible bubbles in the ring, a reset process is necessary. Initially, the Controller block must generate the reset for the DCO and set both comp and fsel signals to their maximum values. This sets the feedback clock (fdbk\_clk) signal and propagates this value faster to each ICO stage. Next, the DCO reset release occurs before removing reset signals from all other modules. The DCO architecture also allows the possibility of interrupting the clock without opening the ring, through the clock retain (CR) circuit, which is an exact copy of the DCO Clock Gating Circuit. The  $clk\_retain$  signal assertion does not stop the oscillation, instead, it disconnects the oscillation ring from the clock output. In this way, the CR circuit allows adjusting the clock to PVT variations effects, by disabling the clock signal propagation. The external clock signal release happens as soon as the adjustment to PVT variations ends.

### B. The Actuator

The Actuator converts the frequency selection digital input to an analog value that drives the ICO. The reason for using two DACs is functional: PCDAC enables limiting the maximum achievable operating frequency, while FSDAC manages the excursion between maximum and minimum frequencies, in a glitch-free manner. Since the ICO assumes a current-controlled structure, DACs manipulate current values as well. The Frequency Selector DAC (FSDAC) produces the  $i_{CTRL}$  input to the ICO, and its internal structure appears in Fig. 5.

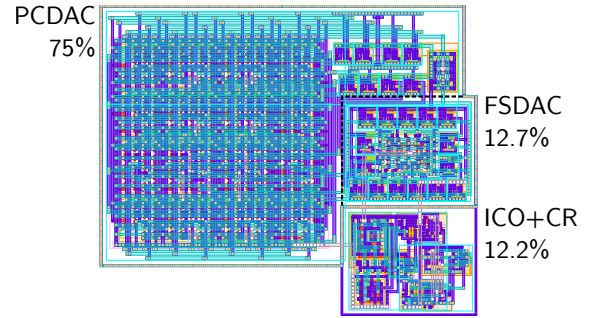


Fig. 6. DCO Layout. Visible layers are Poly, Diffusions, Metal1 and Metal2.

The FSDAC receives as input a 15-bit thermometer-like encoded signal named fsel that generates one out of 16 distinct current values in  $i_{CTRL}$ . Based on an input current source ( $i_{COMP}$ , from PCDAC) the number of active current mirrors that are connected in parallel defines the value of  $i_{CTRL}$ . All current mirrors are identical, i.e. supply the same amount of current and each bit of fsel activates one of the current mirrors when set to 1. This code guarantees a monotonic conversion process, which provides continuously increasing or decreasing frequencies values. PVT variations can cause irregularities (e.g. frequencies above the maximum allowed).

A gate-based switching scheme allows producing the required frequencies, and substitutes the classical drain-based switching of current mirrors (see Fig. 5). This guarantees that the output current varies between minimum and nominal values, never exceeding the later. Thereby, it is possible to limit the maximum frequency generated to stay below or at a specified value. Transistor M2 is always on, to keep a minimum operating frequency. As input, the FSDAC has a reference current source (the structure to the left of transistor M2 in Fig. 5) and the frequency selection channel, fsel. The output is a controlled current ( $i_{CTRL}$ ), which drives the ICO.

The PVT compensator DAC (PCDAC) enables fine tuning the reference current source ( $i_{REF}$ ), generating  $i_{COMP}$ , which feeds the FSDAC. An external controller can explore the direct manipulation of  $i_{REF}$  to provide extra precision, since its current adjusts the ICO delay. The PCDAC allows defining 256 distinct  $i_{COMP}$  levels to the FSDAC. In fact, this mechanism allows  $\pm 10\%$  variations in  $i_{REF}$  without affecting the correct functionality of the DCO. The PCDAC has a circuit topology similar to that of the FSDAC (current mirrors connected in parallel). However, contrary to the later, it employs a binary code input (comp) and a current mode output. A binary code is possible here because during DCO calibration, where the PCDAC has switching activity in its inputs, the generated clock signal is cut from the external entity. Thus, the LCG confines any possible hazard. Just as in the FSDAC, the PCDAC has a dedicated transistor to keep a minimum reference current.

## V. IMPLEMENTATION AND QUANTITATIVE DATA

The DCO layout was developed using *Cadence IC Design* and the STM CMOS 65 nm technology with low-power, standard and low  $V_{th}$  transistors. The layout employs the three first metal layers, and *Metal4* is used as a shield. As Fig. 6 shows, the layout of the DACs relies on the use

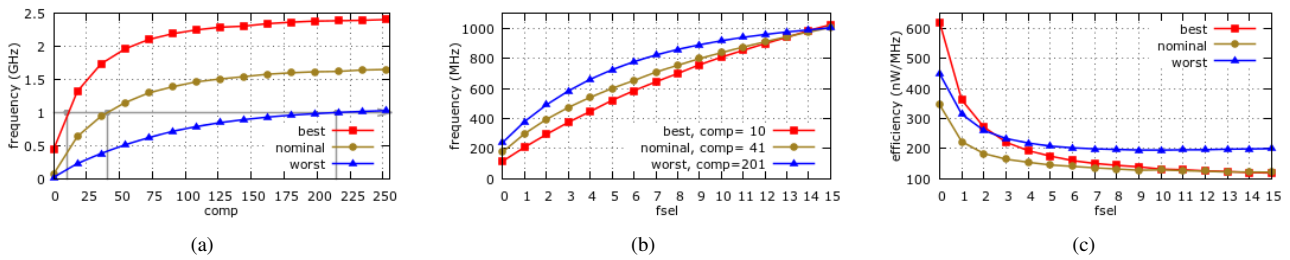


Fig. 7. a) frequency varying the PCDAC with the  $fsel$  at maximum value, b) frequency at corners cases varying the  $fsel$  and c) DCO power efficiency.

TABLE I  
SIMULATION CORNERS FOR TESTS ON THE PROPOSED DCO.

Variable	Worst	Nominal	Best
Process	Slow-Slow	Typ-Typ	Fast-Fast
Voltage ( $V_{dd}$ )	1.08V	1.20V	1.32V
Temperature	-55°C	25°C	125°C
Current ( $I_{REF}$ )	12 $\mu$ A	13.5 $\mu$ A	15 $\mu$ A
Parasitics	$RC_{max}$	$RC_{typ}$	$RC_{min}$

TABLE II  
COMPARISON WITH STATE OF THE ART.

Work	[3]	[4]	[6]	[7]	This
No. of Freqs	4	16	255	33	16*
Glitch-free	No	No	No	Yes	Yes
Clock Gating	Yes	Yes	No	No	Yes
Technology (nm)	90	45	32	65	65
Area ( $\mu m^2$ )	189.7	-	1600	7800	850.6
Power ( $\mu$ W)	500	-	-	2700	$\leq 197$

\*Number of frequencies that can be changed in a glitchless manner.

of the common-centroid technique, widely used on current mirror and differential pair layouts to reduce mismatches [13]. Guard rings were created around DACs, ICO and CR to avoid external noise from digital circuitry [13]. The ICO has 75 transistors and the CR has 45 transistors, which amount to a total area of  $850.6 \mu m^2$ . This corresponds roughly to seventy 2-input NANDs of the technology core library. Clearly, the DCO is a small module, easily included as part of NoC routers and processing elements.

The DCO experiments were performed using *Mentor Calibre PEX* to extract the parasitic capacitances. Three simulation scenarios were considered: *worst*, *nominal* and *best* cases – each one described in Table I. The *best* and *worst* corner temperatures were determined by experiments and are in accordance with [14] for deep submicron technology nodes. Simulations employed the *Cadence Spectre*. Fig. 7(a) illustrates the behavior of the PCDAC. Here, the  $comp$  signal is varied along all values, while  $fsel$  is set to its maximum (all 1s). It is possible from this to precisely define the maximum frequency that can be obtained for each value of  $comp$ , which enables calibrating the DCO. In all scenarios a 1 GHz frequency could be reached, even in the *worst* corner. Fig. 7(b) shows the sixteen different frequencies provided by  $fsel$  after setting  $comp$ . The DCO power efficiency appears in Fig. 7(c), and is between 618 nW/MHz and 120 nW/MHz. Power consumption goes from 62  $\mu$ W, to 197  $\mu$ W, and drops to 0.45  $\mu$ W under clock gating.

## VI. CONCLUSION AND FUTURE WORKS

According to Table II, our DCO can operate with a reasonable number of frequencies while guaranteeing glitchless changes. Area occupation is higher than that reported in [3], but it is at least 50% lower than similar works built with DCOs [6], [7]. Also power consumption is at least 60% lower than all other solutions and can still be improved by the clock gating mechanism. In addition to the PVT adaptation features that Sec. V described, it is viable using individual LCGs based on the proposed DCO for modules such as routers and other IPs of an MPSoC. The proposed DCO needs an external current source, which implies additional area overhead. However, it is possible to share this element among several LCGs. Ongoing work includes the design of more complex controllers to take advantage of the 4096 frequencies which are in fact generated by the DCO (16 from the FSDAC  $\times$  256 from the PCDAC).

## REFERENCES

- [1] D. Chapiro, "Globally-Asynchronous Locally-Asynchronous Systems," Ph.D. dissertation, Stanford University, 1984.
- [2] "International Technology Roadmap for Semiconductors," <http://www.itrs.net>, Accessed in Oct 2014.
- [3] A. Sobczyk, A. Luczyk, and W. Pleskacz, "Controllable Local Clock Signal Generator for Deep Sub-micron GALS Architectures," in *Design & Diagnostics of Electronic Circuits & Systems DDECS*, 2008.
- [4] M. Yadav, M. Casu, and M. Zamboni, "DVFS Based on Voltage Dithering and Clock Scheduling for GALS Systems," in *Int. Symposium on Asynchronous Circuits and Systems (ASYNC)*, 2012, pp. 118–125.
- [5] T. Rosa, V. Larrea, N. Calazans, and F. Moraes, "Power consumption reduction in MPSoCs through DFS," in *Symposium on Integrated Circuits and Systems Design (SBCCI)*, 2012.
- [6] C. Albea, D. Puschini, P. Vivet, I. M. Panades, E. Beigné, and S. Lesecq, "Architecture and Robust Control of a Digital Frequency-Locked Loop for Fine-Grain Dynamic Voltage and Frequency Scaling in Globally Asynchronous Locally Synchronous Structures," *Journal of Low Power Electronics (JOLPE)*, vol. 7, no. 3, pp. 328–340, 2011.
- [7] S. Hoppner, H. Eisenreich, S. Henker, D. Walter, G. Ellguth, and R. Schuffny, "A Compact Clock Generator for Heterogeneous GALS MPSoCs in 65-nm CMOS Technology," *IEEE Trans. on VLSI Systems*, vol. 21, no. 3, pp. 566–570, 2013.
- [8] U. Ogras, R. Marculescu, D. Marculescu, and E. G. Jung, "Design and Management of Voltage-Frequency Island Partitioned Networks-on-Chip," *IEEE Trans. on VLSI Systems*, vol. 17, no. 3, pp. 330–341, 2009.
- [9] E. Carara, R. Oliveira, N. Calazans, and F. Moraes, "HeMPS - a framework for NoC-based MPSoC generation," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2009, pp. 1345–1348.
- [10] R. Ginosar, "Metastability and Synchronizers: A Tutorial," *IEEE Design & Test of Computers (IDTC)*, vol. 28, no. 5, pp. 23–35, 2011.
- [11] R. Baker, *CMOS: Circuit Design, Layout, and Simulation*. Wiley, 2011.
- [12] C. Klapf, A. Missoni, W. Pribyl, G. Holweg, and G. Hofer, "Analyses and Design of Low-Power Clock Generators for RFID TAGs," in *Ph. D. Research in Micro. and Electronics (PRIME)*, 2008, pp. 181–184.
- [13] B. Razavi, *Design of Analog CMOS Integrated Circuits*. Tsinghua University Press, 2001.
- [14] J. Bhasker and R. Chadha, *Static Timing Analysis for Nanometer Designs: A Practical Approach*. Springer, 2009.