Swimming pool like distributed architecture for clock generation in large many-core SoC

Chuan Shan, François Anceau, Dimitri Galayko, Eldar Zianbetov

”Swimming pool”-like distributed architecture for clock generation in large many-core SoC

Chuan Shan, François Anceau, Dimitri Galayko, Eldar Zianbetov
Laboratoire d’informatique de Paris 6 (LIP6)
University of Pierre and Marie Curie (UPMC), Paris, France
Email: {First name.Last name}@lip6.fr

***Abstract*—Synchronization is an issue of significant importance in large-scale, distributed and high-speed systems. Traditional globally synchronous approach is no longer viable due to severe wire delay. Solutions such as ”Globally Asynchronous, Locally Synchronous (GALS)” approaches suffer from metastability risk limiting their use in many-core SoC for critical applications, such as aerospace, military or medical equipment. This paper presents a distributed clock generator based on a network of oscillators. A great advantage of this architecture is its high stability and immunity to perturbations. This architecture also makes possible to design large fully synchronous SoC. A 10×10 network supplying clock sources for 100 clock domains has been modeled in VHDL and is under design in silicon. Simulation results show ± 40 ps peak-to-peak phase error between two neighboring clock signals and ± 50 ps between two clocks in distance.**

I. Introduction

Advances in CMOS technology have led to an exponential increase of chip complexity and transistor numbers (Moore’s law). The modern SoC can be regarded as micro-networks. Synchronization between their different areas becomes a research subject of utmost importance.

There are many methods for synchronous operation. Traditional clock distribution in complex circuits uses tree or grid structures [1] [2]. These centralized distribution techniques are very expensive, mainly in terms of energy consumption and area. They also suffer from uncontrollable skew and jitters.

To solve these problems, large digital chip are partitioned into local clock areas [3] [4]. Each zone has its own local clock. These zones are small enough so that the clock distribution inside the zones can be achieved by conventional techniques. The communication between blocks in different zones can be asynchronous (GALS: Globally Asynchronous and Locally Synchronous) or synchronous (GSLS: Globally Synchronous and Locally Synchronous). In asynchronous circuits, the reliability is difficult to guarantee at design stage because of a large state space volume of the designed system: the time is continuous and not quantized as in synchronous circuits. For this reason, there is a motivation for pursuing researches on alternative techniques of synchronization in large SOCs.

This study focuses on study and design of distributed clock generators, whose basic idea was presented in [5] [6]. Each local clock area has its clock generator coupled in phase with the clock generators in neighboring areas. The main advantage of this kind of synchronous architecture is that the communication between two adjacent areas obey the same rules as inside a single clocking area. Compared with a centralized clock generator, the distributed clock generator requires shortest paths to carry clock signals, reducing the accumulation of delay and jitters on distribution lines.

Several implementations of clock generator prototypes were achieved with this technique. In particular, recent work using a network of all-digital PLLs for on-chip local clock synchronization proved feasibility of this approach and compatibility with digital environment of SoCs [7].

In a tissue of coupled and spatially distributed oscillators, questions about global stability and robustness towards local perturbations are of paramount importance [8] [9]. This paper presents study result of an original solution aiming to limit wave propagation in a coupled PLL network, and to prevent an advent of standing wave. The proposed technique has been validated by behavior simulation.

This paper is organized as follows. In Section II, we start by studying one local clock generator in a large unlimited network. Then, we introduce an analogy between the surface of the phase error in a network of ADPLL and a water surface. In Section III we study the phenomena of error wave propagation and reflection in an ADPLL network with limited surface, and we also explain the advantage of proposed ”Swimming pool”-like architecture in preventing error wave reflection. Simulation results are presented in Section IV to demonstrate the performance of proposed architecture.

II. Structure and behavior of a single ADPLL

A many-core large SoC is partitioned into multiple local clocking areas. Each of these areas has its own clock generator (represented by the squares in Fig. 1(a)). The locally generated clock is synchronized in phase with its four neighbors by coupling links (represented by the lines in Fig. 1(a)). The goal of the distributed PLL network is to synchronize all oscillators both in frequency and in phase. In steady state, such a network is the source of fully synchronous local clocks.

Fig. 1(b) presents the structure of one local clock generator (ADPLL) in the network [7]. It is composed of phase-frequency detectors (PFD) placed between two synchronous clock areas (SCA), measuring the phase/frequency difference between this locally-generated clock and its neighboring clock. Each PFD generates a 5-bit signed binary code. Then, the errors with neighbors are added and processed by the loop filter (LF) to generate a control word for the digitally controlled oscillator (DCO). The loop filter is a digital proportional-integral (PI) one. The output of the DCO is divided by 4 in order to be used in the feedback path of the PLL.

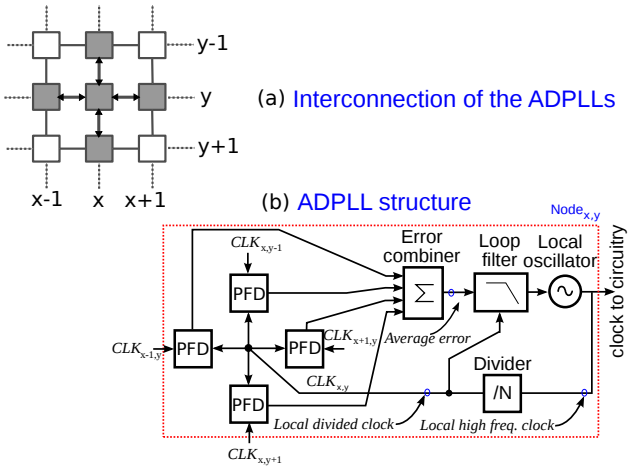

Fig. 1. Interconnection and structure of the ADPLLs

A. From a discrete network to a continuous surface

Each ADPLL has a closed loop behavior given by a differential equation. A network of nodes represented by such a differential equation can be seen as a discretization of liquid surface, or an elastic membrane surface.

The transfer function of each block in an ADPLL can be expressed in Laplace domain:

$$\begin{aligned} H_{\text{PFD}} &= K_{\text{PFD}} = \frac{1}{\delta_{PFD} \cdot 2\pi f_s} \\ H_{\text{Filter}} &= (K_p + \frac{K_i}{s}) \cdot e^{-s\tau} \\ H_{\text{DCO}} &= \frac{K_{\text{DCO}}}{s} \end{aligned} \tag{1}$$

where K_{PFD} and K_{DCO} are the gains of PFD and DCO. δ_{PFD} is the timing resolution of PFD and f_s is the sampling frequency. K_p and K_i are the gains of the proportional and integral paths in the PI filter, respectively. τ is the delay in loop filter. The closed loop transfer function of feedback system can be expressed as:

$$\begin{aligned} H &= \frac{H_{\text{PFD}} \cdot H_{\text{Filter}} \cdot H_{\text{DCO}}}{1 + H_{\text{PFD}} \cdot H_{\text{Filter}} \cdot H_{\text{DCO}}} \\ &= \frac{M(K_p s + K_i)e^{-s\tau}}{s^2 + M(K_p s + K_i)e^{-s\tau}} \end{aligned} \tag{2}$$

where $M = K_{\text{PFD}} K_{\text{DCO}}$.

The phase comparison part of ADPLL consists of four PFDs and generates the average value of phase errors (Fig. 1(b)). The input of feedback loop for the node (x,y) can be regarded as $(\phi_{x+1,y} + \phi_{x-1,y} + \phi_{x,y+1} + \phi_{x,y-1})/4$. Each ADPLL satisfies the following equation:

$$\phi_{x,y} = \frac{\phi_{x+1,y} + \phi_{x-1,y} + \phi_{x,y+1} + \phi_{x,y-1}}{4} \cdot H \tag{3}$$

The sum of phase error is:

$$\begin{aligned} \Sigma_{\text{error}} =& (\phi_{x+1,y} - \phi_{x,y}) + (\phi_{x-1,y} - \phi_{x,y}) + \\ & (\phi_{x,y+1} - \phi_{x,y}) + (\phi_{x,y-1} - \phi_{x,y}) \end{aligned} \tag{4}$$

This equation coincides with the discretization (Eq. 6) of the Laplacian (Eq. 5) of a scalar field Φ in 2-dimensional space.

$$\Delta\Phi_{x,y} = \frac{\partial^2 \Phi}{\partial x^2} + \frac{\partial^2 \Phi}{\partial y^2} \tag{5}$$

$$\begin{aligned} \Delta\Phi_{x,y} \simeq& (\phi_{x+1,y} - \phi_{x,y}) - (\phi_{x,y} - \phi_{x-1,y}) + \\ & (\phi_{x,y+1} - \phi_{x,y}) - (\phi_{x,y} - \phi_{x,y-1}) \end{aligned} \tag{6}$$

Hence, from Eq. (4) and (6), Eq. (3) becomes

$$\Phi_{x,y} = (\frac{\Delta\Phi_{x,y}}{4} + \Phi_{x,y}) \cdot \frac{M(K_p S + K_i)e^{-s\tau}}{S^2 + M(K_p S + K_i)e^{-s\tau}} \tag{7}$$

By performing a reverse Laplace transform on Eq. (7), we arrive at the following differential equation:

$$\frac{\partial^2 \Phi}{\partial t^2} = \frac{MK_p}{4} \frac{\partial \Delta\Phi(t-\tau)}{\partial t} + \frac{MK_i}{4} \Delta\Phi(t-\tau) \tag{8}$$

To simplify the equation, we make two approximations. First, during a very small variation of time t, $\Delta\Phi(t-\tau) \simeq \Delta\Phi(t) - \frac{\partial \Delta\Phi(t)}{\partial t} \cdot \tau$; Second, a variation of local clock phase Φ introduces a change of phase differences $\Delta\Phi$ with its neighboring nodes clocks. Since the loop filter processes the mean value of four phase errors, according to Eq. (6), during a small variation of time, $\frac{\partial \Delta\Phi}{\partial t} \simeq -4\frac{\partial \Phi}{\partial t}$.

Hence, Eq. (8) can be approximated as:

$$\frac{\partial^2 \Phi}{\partial t^2} \simeq -\frac{M(K_p - K_i\tau)}{1 - MK_p\tau} \frac{\partial \Phi}{\partial t} + \frac{MK_i}{4(1 - MK_p\tau)} \Delta\Phi \tag{9}$$

B. An analogy with damped wave equation

Up here, we have performed a reverse discretization passage from ADPLL mesh ϕ to a continuous surface of phase errors Φ. Eq. (9) is the same as the damped wave equation describing water surface movement with dissipation [10]:

$$\frac{\partial^2 h}{\partial t^2} = -k\frac{\partial h}{\partial t} + c^2 \Delta h \tag{10}$$

Here h is the height of the water, c is the wave speed and k is the damping constant. We can make an analogy between the level of water and the phase error of synchronous network. By comparing Eq. (9) and Eq. (10) we get the k and c parameters of the synchronization errors surface.

$$k = \frac{M(K_p - K_i\tau)}{1 - MK_p\tau}, \quad c = \sqrt{\frac{MK_i}{4(1 - MK_p\tau)}} \tag{11}$$

The transient process of an unlimited ADPLL network in phase domain can be seen as analogous to the wave movement

in a vast expanse of water. In equilibrium, the whole water surface is flat. Similarly, when the ADPLL network is synchronized, all the locally generated clocks are in phase. However, if there is a local perturbation, a wave may appear, propagating through the network. This is an undesirable phenomena. The solution aimed to limit it is proposed in the next section.

III. ADPLL NETWORK WITH LIMITED SURFACE

A large synchronous surface satisfies continuity condition: In a local micro region, the difference of level between the nodes (x+1,y) and (x,y) approximates the inverse of level difference between nodes (x-1,y) and (x,y), thus the Laplacian of local phase approaches zero, which means the local region $\langle \phi_{x+1,y}, \phi_{x,y+1}, \phi_{x-1,y}, \phi_{x,y-1} \rangle$ can be regarded as flat.

However, on the boundary of a limited network, the error wave reflects on the border and increase the perturbation inside the network.

To suppress this reflection, we can change the connections to force $\frac{\partial^2 \Phi_{x,y}}{\partial x^2}$ to be zero. This can be obtained by removing the link from node (x+1,y) to the node (x,y) (Fig. 2(a)).

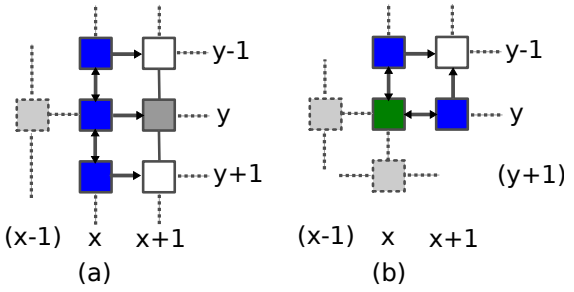

Fig. 2. (a) border (b) corner (→: unidirectional; ↔, −: bidirectional)

It means that the variation in x-direction is not taken into consideration on the vertical border. $\Delta\phi_{x,y}$ turns out to be

$$\Delta\Phi_{x,y} = \frac{\partial^2 \Phi_{x,y}}{\partial y^2} \simeq \phi_{x,y+1} + \phi_{x,y-1} - 2\phi_{x,y} \qquad (12)$$

Similarly, Fig. 2(b) shows the case for a corner node. Therefore, this network topology (Fig. 3) isolates a border only distributing its clock to a kernel surface in which the nodes are connected as in the case of unlimited network.

All ADPLLs (border and kernel) are used to generate local clocks. The network border can be regarded as an independent and synchronous ring exciting the inner kernel as a membrane. This ring, with the reference clock signals at its 4 corners, produces a reference for ADPLLs in the kernel and absorbs the error waves. In the "Swimming pool"-like analogy, the ring of the ADPL network acts as the overflow channels of a pool.

IV. SIMULATION RESULTS

A 10×10 network as shown in Fig. 3 is modeled in VHDL. A PFD with a resolution of 20 ps and a DCO with a nominal frequency of 1 GHz and mean frequency step of 2.26 MHz are used in this work. The simulations allow studying the behavior of network with different parameters and validating the theoretical analysis.

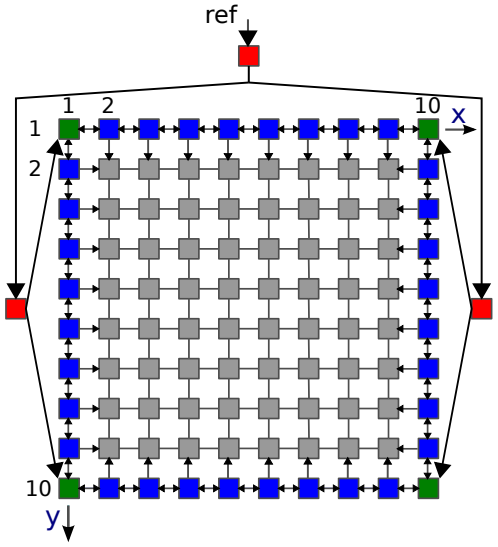

Fig. 3. Proposed network topology (→: unidirectional; ↔, −: bidirectional)

To observe the transient process of error attenuation, phase errors of local clocks with respect to the reference clock are sampled each cycle and used to create a 3-D animation of the transient process. Fig. 4 shows the phase error surface before the whole network is in phase. We observe that the border is very stable with a relatively low amplitude of errors, while the kernel acts like a membrane fluctuating up and down with an amplitude smaller and smaller until the whole network gets in phase.

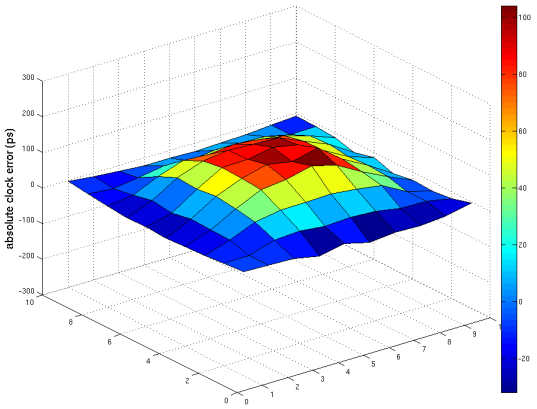

Fig. 4. Phase errors of all clocks in the network at a given moment

When the whole network is in phase, the phase difference between two neighboring clocks is within ±40 ps, which is two steps of PFD resolution. We measure the phase error of each clock with respect to the reference clock REF so as to obtain the clock error distribution histogram of nodes in the kernel (Fig. 5(a)) and that of nodes in the ring border (Fig. 5(b)). It is obvious that the border clocks have smaller errors than kernel clocks, which agrees with our previous analysis.

A significant advantage of the proposed circuit is its good performance of perturbation attenuation. To prove it, we compare the proposed architecture shown in Fig. 3 with

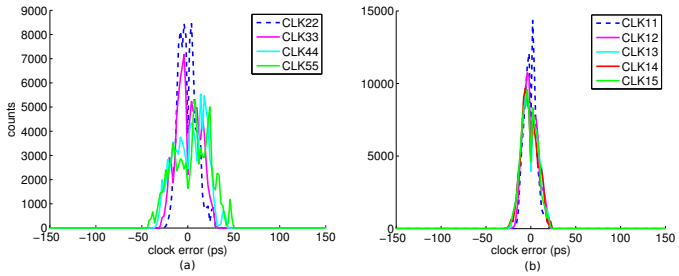

Fig. 5. Histogram of absolute phase errors at steady state: (a) inner clock signals; (b) border clock signals

a conventional 10×10 fully connected topology without ring. When the network is in phase, we add an artificial perturbation on CLK35 at the node (3,5), and observe the transient response on nodes (3,5), (2,5) and the nearest border node (1,5).

In the conventional circuit, it is obvious that CLK15 is affected by the perturbation put on CLK35. The reflection produces some wavelet on node (1,5) (Fig. 6(a)). In the proposed "Swimming pool"-like topology, we can observe that thanks to the strong ring border, CLK15 is not affected and there is no more wavelet on CLK35 (Fig. 6(b)).

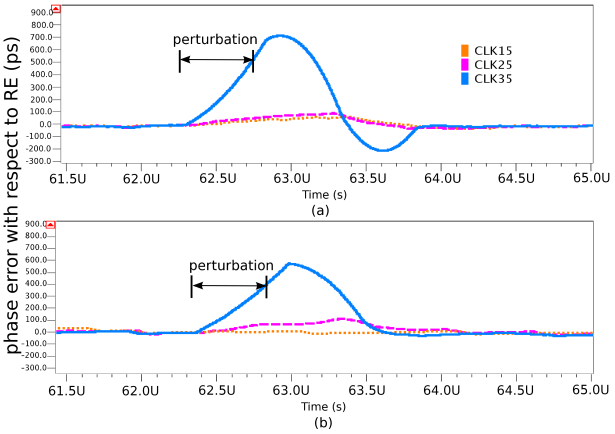

Fig. 6. Transient response of perturbation: (a) conventional 10×10 network (b) "Swimming pool"-like 10×10 network

According to Eq. (11), the wave speed and damping constant depend on design parameters (gains of PFD and DCO, filter coefficients, etc.). The reconfigurability of loop filter allows modifying features of the control system according to the specification [7]. Fig. 7 shows phase errors of clock signals in the principal diagonal of the proposed network with two different parameter sets. We can observe that if the system is overdamped, the system takes a shorter time to acquire the reference frequency, but the phase error in steady state is relatively larger (± 100 ps in Fig. 7(a)) compared to an underdamped system (± 50 ps in Fig. 7(b)). Designers can choose the appropriate parameters to meet requirements of convergence speed and maximum error limit.

V. Conclusion

An ADPLL network with "Swimming pool"-like topology is proposed for synchronization in large many-core SoC. This

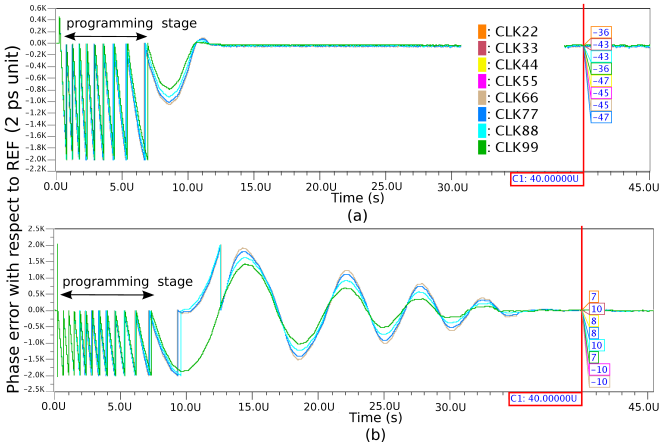

Fig. 7. Absolute phase errors of clock signals in proposed network with different parameters: (a) overdamped; (b) underdamped

architecture enhances the stability of system by improving its capacity of perturbation attenuation. Theory analysis presented in this paper supports design of the proposed architecture for different specifications. Simulation results proved the feasibility of a fully synchronized circuit with a large size. This possibility allows the use of large globally synchronous architecture, so offering a possibility of designing and verifying SoCs with well-established and secure design methods. The design of a large test circuit (10 × 10) nodes will be finished soon to demonstrate the validity of this approach.

Acknowledgment

This work has been funded by the French National Agency of Research (ANR) under grant ANR-10-SEGI-014-01.

References

[1] A. Abdelhadi et al., *Timing-driven variation-aware nonuniform clock mesh synthesis*, Proceedings of the 20th symposium on Great lakes symposium on VLSI, 2010, pp. 15-20

[2] V. Tiwari et al., *Reducing power in high-performance microprocessors*, Proceedings of the 35th annual Design Automation Conference, 1998, pp. 732-737

[3] Anceau, Francois, *A synchronous approach for clocking VLSI systems*, Solid-State Circuits, IEEE Journal of, vol. 17, n. 1, 1982, pp. 51-56.

[4] S. Rusu et al., *A 45 nm 8-core enterprise Xeon processor*, IEEE Journal of Solid-State Circuits, vol. 45, n. 1, 2010, pp. 7-14.

[5] G. A. Pratt et al., *Distributed Synchronous clocking*, IEEE transaction on parallel and distributed systems, vol. 6, n. 3, march 1995, pp. 314-328.

[6] , Maza, Manuel Salim and Aranda, Mónico Linares, *Analysis and verification of interconnected rings as clock distribution networks*, Proceedings of the 14th ACM Great Lakes symposium on VLSI, 2004, pp. 312–315.

[7] E. Zianbetov et al., *All-digital PLL array provides reliable distributed clock for SOCs*, IEEE international ISCAS conf., 2011, Rio de Janeiro, pp. 2589-2593

[8] , Korniienko, Anton and Scorletti, Gérard and Colinet, Eric and Blanco, Eric and Juillard, Jérôme and Galayko, Dimitri, *Control law synthesis for distributed multi-agent systems: Application to active clock distribution networks*, American Control Conference (ACC), 2011, pp. 4691–4696.

[9] , Akre, J and Juillard, Jérôme and Galayko, Dimitri and Colinet, Eric, *Synchronization analysis of networks of self-sampled all-digital phase-locked loops*, Circuits and Systems I: Regular Papers, IEEE Transactions on, vol. 59, n. 4, pp. 708–720, 2012.

[10] R.S. Johnson, *A Modern Introduction to the Mathematical Theory of Water Waves*, 1st edition, Cambridge University Press, October 28, 1997.