Title	A Concept of Synchronous ADPLL Networks in Application to Small-Scale Antenna Arrays
Authors(s)	Koskin, Eugene; Galayko, Dimitri; Blokhina, Elena
Publication date	2018-02-09
Publication information	IEEE Access, 6 : 18723-18730
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/9684
Publisher's statement	© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/ACCESS.2018.2804324

A Concept of Synchronous ADPLL Networks in Application to Small-Scale MIMO Antenna Arrays

EUGENE KOSKIN[1] (STUDENT MEMBER, IEEE), DIMITRI GALAYKO[2] (MEMBER, IEEE), AND ELENA BLOKHINA[1] (SENIOR MEMBER, IEEE)

[1] School of Electrical and Electronic Engineering, University College Dublin, Belfield, Dublin 4, Ireland

[2] LIP6 Lab, Sorbonne Université, 4 place Jussieu, 75252 Paris Cedex 05, France

ABSTRACT In this paper, we introduce a reconfigurable oscillatory network that generates a synchronous and distributed clocking signal. We propose an accurate model of the network to facilitate the study of its design space and ensure that it operates in its optimal, synchronous mode. The network is designed and implemented in a fully integrated 65nm CMOS system-on-chip that utilises coupled all digital phase locked loops interconnected as a Cartesian grid. The model and measurements demonstrate frequency and phase synchronisation even in the presence of noise and random initial conditions. This network is proposed for small-scale multiple input multiple-output systems that require complete synchronisation both in frequency and in phase.

INDEX TERMS Antenna arrays, Distributed clocks, Networks of oscillators, Synchronous circuits, Internet of Things, Systems-on-a-chip.

I. INTRODUCTION

A Growing number of mobile data customers leads to an exponential increase in internet traffic worldwide. According to a recent Cisco report [1], the number of mobile-connected devices will increase beyond 11 billion within the next 5 years, reaching 50 exabytes of global data traffic per month by 2021. The predicted traffic will be approximately seven times greater compared to the rate we had at the end of 2016. Conventional wireless technologies such as 3G and 4G will remain dominant, with 19% and 79% of global mobile traffic respectively. However, a further increase in traffic would require the development of 5G technologies.

The fifth generation (5G) of wireless communications is a front edge research topic, which has attracted increasing attention over past few years, accelerating the development of networks [2]–[13] and devices for communication [14]–[21]. The main advantage of a 5G network is to support significant amounts of mobile data from spatially distributed devices (the Internet of Things) and a huge number of wireless connections between them. In addition, 5G networks are promising for achieving better cost-energy performance, as well as the quality of service in terms of communication delay, reliability and security.

Proposed carrier frequencies for 5G communications will lie in the millimetre-wave band (e.g., 28 GHz) [22], [23] where the spectrum is less crowded and greater bandwidths are available. These carrier frequencies correspond to wavelengths of 1 cm and below, requiring an antenna size less of than 2.5 mm. As a result, a single antenna becomes inefficient in transmitting signals with enough power to communicate with nearby receivers and cannot be used on its own.

A promising approach to overcome this difficulty relies on an array of spatially distributed antennas transmitting synchronous signals, which can be classified as a multiple-input multiple-output (MIMO) system. Depending on the application, some MIMO antennas will be required to transmit fully coherent signals, while others may be required to transmit signals with specific phase delays. In both cases, the precise control over the frequency and phase of the transmitted signals is important.

With regard to antenna arrays operating synchronously, globally-modular architectures, where each module drives a small group of antenna units, are commonly used (see Fig. 1). For the successful operation of the system, phase-frequency synchronicity must be achieved so that each antenna receives the same clocking signal. It is a known challenge to deliver such a clocking signal to a large chip. One of the solutions may consist of utilising a phase-locked loop to synthesise the
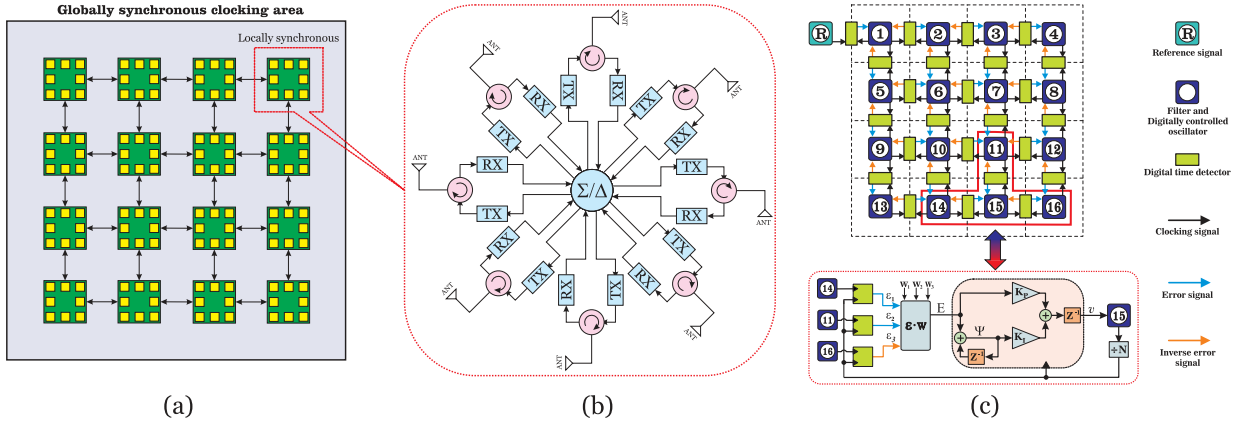
FIGURE 1: (a) General concept of a distributed RF microchips (green squares) on a ADPLL oscillatory network. Each RF microchip drives eight antennas (yellow squares) within its locally synchronous area (outlined as dashed red square). (b) Antennas distributed on a RF microchip that includes receiver and transmitter chains. (c) Schematics of an oscillatory network including principal blocks of an ADPLL: digital time detector (DTD), PI control block, digitally controlled oscillator (DCO) and reference signal.

required frequency and deliver the signal. Another approach proposes using synchronised coupled phase-locked loops (PLLs). The concept, based on Digital Phase-Locked Loops, has been recently considered [2], [3]. However, a practical implementation of such a network to deliver a distributed signal for antenna arrays has not been offered to date to the knowledge of the Authors.

A PLL network is a spatially distributed array of voltage or digitally controlled oscillators interacting with each other through exchange of error signals [24]–[27]. Various configurations of networks have been used for generating a distributed clocking signal in systems-on-chips (SoCs) [28]–[30]. However, achieving and sustaining synchronicity in phase is still a challenge [26], [31]–[34].

The aim of this work is to develop a network of All Digital Phase-Locked Loops (ADPLL) capable of generating stable, distributed and coherent clocking signals for synchronous transmission in antenna arrays achieving synchronisation in frequency and phase. We propose a model of such a network, investigate its design space to ensure its optimal operation and verify the concept on a 65nm CMOS chip. The paper is organised as follows. Section. II describes a structure of an ADPLL network. In Sec. II-C we introduce the model of the network on arbitrary topology. The experimental verification is provided in Sec. IV, a discussion on the benefits of the model is presented in Sec. III and Sec. V concludes the paper.

II. STATEMENT OF THE PROBLEM
A. ALL DIGITAL PHASE-LOCKED LOOP
An All Digital Phase-Locked Loop is a system with feedback control that generates a signal whose phase is related to that of an input signal. In the case of an all digital version of a PLL, all components of the system are implemented digitally. An ADPLL consists of convention CMOS components and includes a digitally controlled oscillator (DCO), a digital time detector (DTD) and a proportional-integral (PI) control block, all shown in Fig. 1c (lower). The role of the DCO is to

generate a square signal with a frequency F^{L} (and the period $T^{\mathrm{L}} = 1/F^{\mathrm{L}}$) with F^{L} that can be adjusted over a range of values. The frequency of the DCO changes linearly, starting from an initial point F_0, with a constant frequency resolution ΔF_{DCO}, varying from minimum and maximum values ($F_{\min}$, $F_{\max}$). Frequency division ($\div N$) may be applied to the DCO frequency F^{L}. The DTD compares the input signal with the signal generated by the DCO to determine their mismatch. It produces a timing error τ and a quantised timing error ε. The DTD used in this study represents a combination of finite-state machine (with state m) and time-to-digital converter (TDC). The TDC maps the timing error τ into the integer error ε using a quantised sigmoid function $\mathrm{H}(\tau, \tau_{\mathrm{TDC}}, N_{\mathrm{TDC}})$. The DTD has a quantisation range $[-N_{\mathrm{TDC}}; -1] \cup [1; N_{\mathrm{TDC}}]$ (with $N_{\mathrm{TDC}} = 7$ in this study) and a resolution step τ_{TDC}. The PI block adopted from control theory takes the quantised timing error ε and provides a control code $v = K_{\mathrm{p}}E + K_{\mathrm{i}}\Psi$ that adjusts the frequency of the DCO. In the latter expression, the signals E and Ψ are obtained from the proportional and integral paths of the PI block and K_{p} and K_{i} are the gains of those paths.

The major difficulty in the description of an ADPLL is deriving its accurate mathematical model: an ADPLL is a nonlinear and event-driven (discrete) system. For the completion of the model, we introduce the following quantities: the current instance (event) number n, the time L_n to the rising edge of the square signal generated by the DCO and the time R_n to the rising edge of the signal provided by the reference clock having period T^{R}. Therefore, the self-consistent set of equations modelling an ADPLL are as follows [32]:

$$T_n^{\mathrm{R}} = T^{\mathrm{R}}\zeta_n(\sigma^{\mathrm{R}})$$
$$T_n^{\mathrm{L}} = 1/[F_0 + \Delta F_{\mathrm{DCO}}(K_{\mathrm{p}}\varepsilon_n + K_{\mathrm{i}}\psi_n)]\zeta_n(\sigma^{\mathrm{L}})$$
$$L_{n+1} = T_n^{\mathrm{L}}\theta^-(L_n - R_n) + (L_n - R_n)\theta^+(L_n - R_n)$$
$$R_{n+1} = -(L_n - R_n)\theta^-(L_n - R_n) + T^{\mathrm{R}}\theta^+(L_n - R_n)$$
$$m_{n+1} = \mathrm{sign}[m_n + \mathrm{sign}(L_n - R_n)]$$

$$\tau_{n+1} = \tau_n|m_n| + m_{n+1}[L_{n+1}\theta^+(L_{n+1} - R_{n+1}) +$$
$$R_{n+1}\theta^-(L_{n+1} - R_{n+1})]$$
$$\varepsilon_{n+1} = \mathrm{H}(\tau_n, \tau_{\mathrm{TDC}}, N_{\mathrm{TDC}})$$
$$\psi_{n+1} = \psi_n + \varepsilon_n\theta^-(L_n - R_n) \tag{1}$$

Here, we also used the notations: $\theta^+(x)$ is the Heaviside step function and $\theta^-(x) = \theta^+(-x)$, $\zeta(\sigma)$ is a random log-normal variable to model the noise (jitter) in the reference and DCO signals with variance $\sigma^{\mathrm{R,L}}$.

This system of discrete time event-driven equations is used to build a model of an ADPLL network and perform the exploration of its design space and finding its optimal configuration. The approach to describe a network is presented in the next Section.

B. ADPLL-BASED OSCILLATORY NETWORKS

The proposed network represents a Cartesian grid, as shown in Fig. 1c (upper), where every node is an ADPLL. When interconnected in a network, phase-locked loops require a number of DTDs, equal to the number of their neighbours, to generate an averaged error signal. The network is driven by a reference oscillator that provides a reference signal to which the whole network will synchronise. When the network synchronises, all the DCOs have the same frequency and phase.

We describe the network through a directed graph $G = (V, E)$ with vertices V and edges E. All DCOs combined with their PI control blocks are placed in vertices of the graph. For example, in the case of a 2×2 network including a reference signal we can determine a graph as follows:

$$G(V, E) = \begin{cases} V = \{1, 2, 3, 4, 5\} \\ E = \{1 \rightarrow 2, 2 \leftrightarrow 3, 2 \leftrightarrow 4, \\ \quad 3 \leftrightarrow 5, 4 \leftrightarrow 5\} \end{cases} \tag{2}$$

It can be seen that this graph has five directed edges: four of them are symmetrical ($i \leftrightarrow j$) and one is asymmetrical ($1 \rightarrow 2$) to describe the influence of the external reference signal to the DCO at node 2. Hence, five DTDs are required to interconnect the DCOs in such a network.

To describe a network with a larger number of nodes, we adopt the following extension of the simple network example (2). We assume that the reference oscillator is placed at the first vertex of the graph having index $i = 1$. All the others DCOs are placed at the vertices having indices $i = 2 \ldots N_{\mathrm{OSC}}$, where $N_{\mathrm{OSC}} = N_{\mathrm{DCO}} + 1$ is the total number of DCOs in the network. The network graph is then presented in terms of the adjacency matrix $\mathbf{A}$:

$$\mathrm{A}_{i,j} = \begin{cases} 1 & \text{if } i \text{ is reference for } j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

It can be seen that that $\mathrm{A}_{i,j} = \mathrm{A}_{j,i}$ for almost all i, j except from $\mathrm{A}_{1,2}$. As we mentioned above, it points the fact that the external reference signal influences the first DCO but not vice versa. In an ADPLL network, all elements which allocated at the main diagonal of the matrix ($\mathrm{A}_{i,i}$) are equal to zero.

Algorithm 1: GETDETECTORS finds a set of detectors to describe an ADPLL network

Input: An adjacency matrix $\mathbf{A}^{N_{\mathrm{OSC}} \times N_{\mathrm{OSC}}}$
Output: A set of digital time detectors $\mathcal{D}$

1 $\mathcal{D} \leftarrow \emptyset, p \leftarrow 0$
2 **for** $i \leftarrow 1$ **to** $N_{\mathrm{OSC}} - 1$ **do**
3   **for** $j \leftarrow i + 1$ **to** N_{OSC} **do**
4          $p = \max(\mathrm{A}_{i,j}, \mathrm{A}_{j,i})$
5   **if** $p \neq 0$ **then**
6              $\mathcal{D} \leftarrow \mathcal{D} \cup \{(i, j)\}$

7 **return** $\mathcal{D}$

We note that the kth edge in graph, $k : \{i \rightarrow j\}$, connects two oscillators via DTDs: $\mathrm{DCO}_i \rightarrow \mathrm{DTD}_k \leftarrow \mathrm{DCO}_j$. For every DTD_k, the oscillator having the lower index $\min(i, j)$ is defined as reference, while the oscillator with the higher index $\max(i, j)$ is defined as local. Having the adjacency matrix determined, we can define a set of DTDs $\mathcal{D}$ with the help of the Algorithm 1.

As each DCO is designed to be driven by a mean quantised timing error E_i, the signals coming from multiple DTDs must be averaged. For this purposed, we create a matrix of weight coefficients $\mathbf{W}$. For a ith DCO in the network, the total number of its neighbours g_i is defined from the adjacency matrix $\mathbf{A}$: $g_i = \sum_{j=1}^{N} \mathrm{A}_{j,i}$. The weight matrix can be then introduced as

$$W_{i,j} = \begin{cases} \mathrm{A}_{i,j}/g_j & \text{if } g_j \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

For the oscillators that have three neighbours, division by four is used to obtain the averaged timing error E. Finally, using the coefficients $\mathbf{W}$ we calculate the error that is supplied to the PI controller of each DCO:

$$E_j = \sum_{i}^{N} \mathrm{sgn}(j - i)\varepsilon_i \mathrm{W}_{i,j} \tag{5}$$

The mathematical foundations to describe a network of ADPLLs presented in this Section are used to create computation algorithms for network simulations. The algorithms are presented in the next Section.

C. ALGORITHM FOR ADPLL NETWORK SIMULATIONS

In this Section, we suggest a complete event-driven model of a network with an arbitrary topology and its iterating algorithm. We use the adjacency matrix $\mathbf{A}$, introduced in the previous Section, to describe connections between oscillators and detectors in a network. This model allows us to perform a fast optimisation of parameters of the network ensuring its optimal synchronised operation.

In order to apply the model of a single ADPLL oscillator, developed in Sec. II-A, we define the total number of DTDs and their connections to other oscillators in the network. We

---

Algorithm 2: DETECTORSUPDATE updates the state of a detector

Input: Detectors' state before update: $m, \varepsilon, \tau, t_{\min}$
Output: Detectors' state after update: m, ε, τ

1 **for** $k \leftarrow 1$ **to** N_{D} **do**
2      $i \leftarrow \mathcal{D}_{k,1}, \quad j \leftarrow \mathcal{D}_{k,2}$
3   **if** $m_k \neq 0$ **then**
4        $\tau_k \leftarrow \tau_k + t_{\min}$
5      $choice \leftarrow 0$
6   **if** $R_k \leq 0$ *and* $L_k > 0$ **then**
7        $choice \leftarrow 1$
8   **else if** $R_k > 0$ *and* $L_k \leq 0$ **then**
9        $choice \leftarrow 2$
10   **else if** $R_k \leq 0$ *and* $L_k \leq 0$ **then**
11       $choice \leftarrow \mathrm{random}\{1, 2\}$
12   **switch** *choice* **do**
13   **case** *1*
14   **if** $m_k = -1$ **then**
15           $\varepsilon_k \leftarrow -H(\tau_k, \tau_{\mathrm{TDC}\,k}, N_{\mathrm{TDC}\,k})$
16           $\tau_k \leftarrow 0, \quad m_k \leftarrow 0$
17   **else**
18           $m_k \leftarrow 1$
19         $R_k \leftarrow 1/F_i$
20   **case** *2*
21   **if** $m_k = 1$ **then**
22           $\varepsilon_k \leftarrow H(\tau_k, \tau_{\mathrm{TDC}\,k}, N_{\mathrm{TDC}\,k})$
23           $\tau_k \leftarrow 0, \quad m_k \leftarrow 0$
24   **else**
25           $m_k \leftarrow -1$
26         $L_k \leftarrow 1/F_j$
27 **return** $\{m, \varepsilon, \tau\}$

---

also use the definition of the reference and local oscillators introduced in Sec. II-B.

Each ADPLL in the network is described by the set of equations (1). As a result, we obtain a vector of state variables $(T^{\mathrm{R}}[n], T^{\mathrm{L}}[n], R[n], L[n], m[n], \tau[n], \varepsilon[n], \psi[n])^T$ giving the state of one oscillator. In order to simulate a network and take into account all interconnections between the oscillators, when generating the control code to update DCO frequencies, we use the neighbour-average errors $E[n]$ and $\Psi[n]$ (see formula (5)) instead of individual errors $\varepsilon[n]$ and $\psi[n]$. The pseudocode that is used for discrete-time event-driven simulations of an ADPLL network with an arbitrary topology is summarised in Algorithms 2 and 3. In the next Section, we describe the results of numerical simulations and discuss the synchronises mode of network operation.

III. FINDING THE OPTIMAL BEHAVIOUR: SYNCHRONISATION OF THE NETWORK IN FREQUENCY AND PHASE

In this Section, we study the network behaviour through numerical simulations, employing the equations and algorithms we introduced earlier. The aim of this Section is to show that the ADPLL network is capable of synchronisation in frequency and phase. In addition, we aim to show that synchronised regimes are globally stable (i.e., there is no mode-locking or co-existing regimes that depend on initial conditions). The parameters used for simulations are listed

---

Algorithm 3: ADPLLNET simulates a network of AD-PLLs

Input: Iteration steps S; topology: $\mathbf{A}^{N_{\mathrm{osc}} \times N_{\mathrm{osc}}}$,
     $\mathcal{D} \leftarrow \text{GETDETECTORS}(\mathbf{A}), N_{\mathrm{D}} \leftarrow \text{Length}(\mathcal{D})$
     Oscillator parameters: $\Delta F_{\mathrm{DCO}}^{N_{\mathrm{osc}} \times 1}$,
     $K_{\mathrm{p}}^{N_{\mathrm{osc}} \times 1}, K_{\mathrm{i}}^{N_{\mathrm{osc}} \times 1}, F^{N_{\mathrm{osc}} \times 1}, \phi^{N_{\mathrm{osc}} \times 1}, \sigma^{N_{\mathrm{osc}} \times 1}$
     Detector parameters: $\tau_{\mathrm{TDC}}^{N_{\mathrm{D}} \times 1}, N_{\mathrm{TDC}}^{N_{\mathrm{D}} \times 1}$
Output: $\mathcal{F}^{S \times (N_{\mathrm{osc}}+1)}$

1 $\mathcal{F} \leftarrow \emptyset, \quad \mathbf{W} \leftarrow \mathbf{A}, \quad F \leftarrow \tilde{F}, \quad t \leftarrow 0, \quad t_{\min} \leftarrow 0$
2 $\{\mathbf{R}, \mathbf{L}, \varepsilon, \tau, m\} \leftarrow 0^{D_{\mathrm{N}} \times 1}$
3 $\{\mathrm{E}, \Psi, newCycle, g\} \leftarrow 0^{N_{\mathrm{osc}} \times 1}$
4 **for** $i \leftarrow 1$ **to** N_{OSC} **do**
5   **for** $j \leftarrow 1$ **to** N_{OSC} **do**
6        $g_i \leftarrow g_i + \mathrm{A}_{j,i}$
7 **for** $i \leftarrow 1$ **to** N_{OSC} **do**
8   **for** $j \leftarrow 1$ **to** N_{OSC} **do**
9        $\mathrm{W}_{j,i} \leftarrow \mathrm{W}_{j,i}/g_i$
10 **for** $k \leftarrow 1$ **to** N_{D} **do**
11    $i \leftarrow \mathcal{D}_{k,1}, \quad j \leftarrow \mathcal{D}_{k,2}$
12    $R_k \leftarrow \tilde{\phi}_i/(2\pi \tilde{F}_i), \quad L_k \leftarrow \tilde{\phi}_j/(2\pi \tilde{F}_j)$
13 **for** $s \leftarrow 1$ **to** S **do**
14    $\mathcal{F} \leftarrow \mathcal{F} \cup \{(t, F_1, F_2, \ldots, F_{N_{\mathrm{osc}}})\}$
15    $\{newCycle, \mathrm{E}\} \leftarrow 0^{n \times 1}$
16    $t_{\min} \leftarrow \min\{\mathbf{R} \cup \mathbf{L}\}$
17   **for** $k \leftarrow 1$ **to** N_{D} **do**
18      $i \leftarrow \mathcal{D}_{k,1}, \quad j \leftarrow \mathcal{D}_{k,2}$
19      $R_k \leftarrow R_k - t_{\min}, \quad L_k \leftarrow L_k - t_{\min}$
20   **if** $R_k \leq 0$ **then**
21        $newCycle_i = 1$
22   **if** $L_k \leq 0$ **then**
23        $newCycle_j = 1$
24      $\mathrm{E}_j \leftarrow \mathrm{E}_j + \mathrm{W}_{i,j}\varepsilon_k, \quad \mathrm{E}_i \leftarrow \mathrm{E}_i - \mathrm{W}_{j,i}\varepsilon_k$
25   **for** $i \leftarrow 1$ **to** N_{OSC} **do**
26   **if** $newCycle_i = 1$ **then**
27        $F_i \leftarrow \tilde{F}_i + \Delta F_{\mathrm{DCO}\,i}(K_{\mathrm{p}\,i}\mathrm{E}_i + K_{\mathrm{i}\,i}\Psi_i)$
28        $F_i \leftarrow F_i e^{N(0,\sigma_i)}, \quad \Psi_i \leftarrow \Psi_i + \mathrm{E}_i$
29    $\{m, \varepsilon, \tau\} \leftarrow \text{DETECTORSUPDATE}(m, \varepsilon, \tau)$
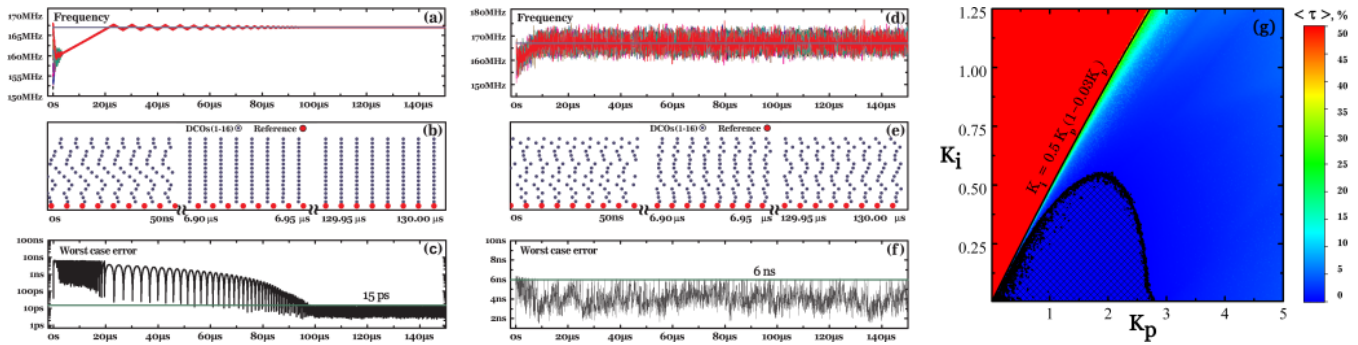30 **return** $\mathcal{F}$

FIGURE 2: Simulation results for a 4×4 ADPLL network with random initial phase and frequency conditions. The network is driven by a signal at 167 MHz. Figures (a) through (c) correspond to $K_{\mathrm{p}} = 0.5$ and $K_{\mathrm{i}} = 0.1$ and show (a) frequency acquisition dynamics, (b) occurrence of clocking events and (c) network jitter dynamics. Figures (d) through (f) correspond to $K_{\mathrm{p}} = 0.5$ and $K_{\mathrm{i}} = 0.3$ and also show (d) frequency acquisition dynamics, (e) occurrence of clocking events and (f) network jitter dynamics. Figure (g) shows network jitter in the plane spanned by K_{p} and K_{i}. Synchronisation in frequency and phase and the minimal network jitter can be achieved at K_{p} and K_{i} lying in the shaded region.

in Table 1 and correspond to an experimental implementation of the ADPLL network (which will be discussed in the next Section). The *control* parameters for the network are the gains K_{p} and K_{i} that can be programmed and that significantly affect the frequency acquisition rate and overall synchronisation process.

As a measure of synchronisation in a ADPLL network, we define the worst-case absolute network jitter $\langle |\Delta\tau| \rangle$:

$$\langle |\Delta\tau| \rangle = \lim_{t \to \infty} t^{-1} \int_0^t \max_{k=1\ldots N_{\mathrm{DTD}}} |\tau_k(t')| \, dt'$$

This quantity represents the averaged time error $\tau_k(t)$ taken across all the detectors in the network. When it is equal to zero, we say that the network is synchronised in frequency and phase, meaning that all the rising edges arrive at the same time instances. In order to understand how small or large the network jitter with respect to the time scale of the network dynamics, we also define relative network jitter, which is the absolute jitter normalised to the period of the reference (driving input) signal: $\langle |\Delta\tau| \rangle / T^{\mathrm{R}} \cdot 100\%$.

As an illustration of the impact that K_{p} and K_{i} have on the system, Fig. 2 shows typical synchronisation patterns in a 4×4 ADPLL network driven by an input signal at 167 MHz. The figure is obtained by simulations, with no external or intrinsic noise included, starting from random initial conditions. While this is not shown in the figure, an investigation of the effect that initial conditions have on the performance of the network has been carried out and no mode-locking has been discovered.

The first column of this figure shows (a) the frequency acquisition dynamics, (b) occurrence of clocking events (rising edges of the digital oscillations generated by all oscillators in the network) and (c) the evolution of the timing error $\langle |\Delta\tau| \rangle$ at $K_{\mathrm{p}} = 0.5$ and $K_{\mathrm{i}} = 0.1$. This is an example of complete synchronisation that occurs following the standard scenario. Firstly, the frequencies generated by the oscillators in the networks approach the input reference frequency F^{R}

during a frequency acquisition stage. Next, the arrival times of clocking events begin to align during a phase acquisition stage. Finally, all signals generated in the network have the same frequency and phase, and synchronisation is observed.

The second column of Fig. 2 shows another pattern where synchronisation is not achieved (figured (d), (e) and (f) obtained at $K_{\mathrm{p}} = 0.5$ and $K_{\mathrm{i}} = 0.3$). It is seen that the frequency is not "stabilised" at F^{R}, the clocking events are not aligned and network jitter is large. When designing an ADPLL network, there is a trade-off in the gain K_{i}. Larger K_{i} ensure faster frequency acquisition rates (please refer to subfigures (a) and (d) to compare the duration of the frequency acquisition phases in both cases), but, on the other hand, they generally "de-stabilise" the network.

Hence, a detailed study of the behaviour of the network with respect to K_{p} and K_{i} is required. For this reason, Fig. 2g shows the plane spanned by the gains K_{p} and K_{i} with the relative network jitter coded with different colours. From the plane, we identify the region of optimal operation that occurs below the line $K_{\mathrm{i}} \approx 0.5K_{\mathrm{p}}$. In particular, the highlighted region below this line corresponds to the synchronisation of the network with the minimal network jitter. When operating in this region of the control parameters K_{p} and K_{i}, the network generates a *spatially distributed* synchronous signal, having the same frequency and phase with the least possible timing error (jitter). In the next Section, we show the experimental verification of the model and the synchronisation predicted by it.

IV. EXPERIMENTAL VALIDATION

This Section is dedicated to the verification of the network model and the synchronisation predicted by the model. The validation is achieved by comparing waveforms of simulated transient responses in a network vs. measured responses. The experimental measurements have been performed on a 2×2 network prototype fabricated in 65nm CMOS technology. The microphotograph of the chip is shown in Fig. 3 [29], [34]. The parameters of the implemented ADPLL network are summarised in Table 1.
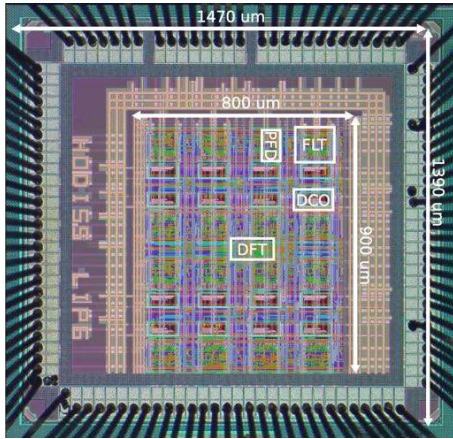
FIGURE 3: Die microphotograph of the distributed clock generator with 4×4 digitally controlled oscillators designed and manufactured in 65 nm CMOS technology.

TABLE 1: Parameters of the experimental implementation of an ADPLL network

Parameter	Value
Number of nodes	4×4
DCO frequency range, MHz	$4 \times (135 \sim 175)$
Division factor N	4
DCO gain Δf_{DCO}	4×150 kHz
TDC resolution step τ_{TDC}	20 ps*
DTD steps N_{DTD}	7
Timing error, ps	< 20
Power consumption, mW	< 1 per node
Technology, nm	65
Chip area, mm^2	≈ 2

* Estimated through additional simulations

The transient responses have been initiated by forcing the frequency of the input signal (reference frequency F^R) to change. In the experiment, the signal generator was programmed to synthesise rectangular signals with frequencies switching between two values. The responses shown in Fig. 4 are obtained by switching F^R between ~158 MHz and ~167 MHz. They illustrate the dynamics of the DCO frequencies in time measured from all the oscillators in the network. Since the oscillators evolve synchronously, they are seen as one signal in this figure. After each switching, the reference frequency remains constant long enough so the network is able to pass through the frequency and phase acquisition stages and settle down to a new reference frequency. This indicates that the synchronised regimes are robust.

The three different cases in Fig. 4 correspond to three different sets of the PI block gains K_p and K_i. As was mentioned earlier, there is a trade-off in selecting these gains. The gain K_i of the integral path of the PI control block is responsible for the frequency acquisition rate while the gain K_p of the proportional path controls signal overshoots during the frequency acquisition stage. However, as was shown by the simulations presented in Sec. III, a specific relationship

between the two must be kept to ensure that the network synchronises in frequency and phase. One can clearly see that since all the three sets of K_p and K_i belong to the optimal region from the plane in Fig. 2g, the network indeed synchronises to the reference signal, but the shape of the transient process and the resulting network jitter are slightly different. Hence, the selection of K_p and K_i is a designer's choice until they lie in the optimal region.

This comparison shows a very good agreement between the ADPLL network model we developed in this study and the measurements taken from the prototype.

V. CONCLUSION

In this study, we have proposed a concept of an ADPLL network, operating in synchronised mode to deliver spatially distributed clocking signal. The synchronisation of the network was investigated in particular with application to drive an antenna array. In order to develop this concept, we have proposed algorithms that allow one to extend a discrete-time event-driven model of a single ADPLL oscillator to a
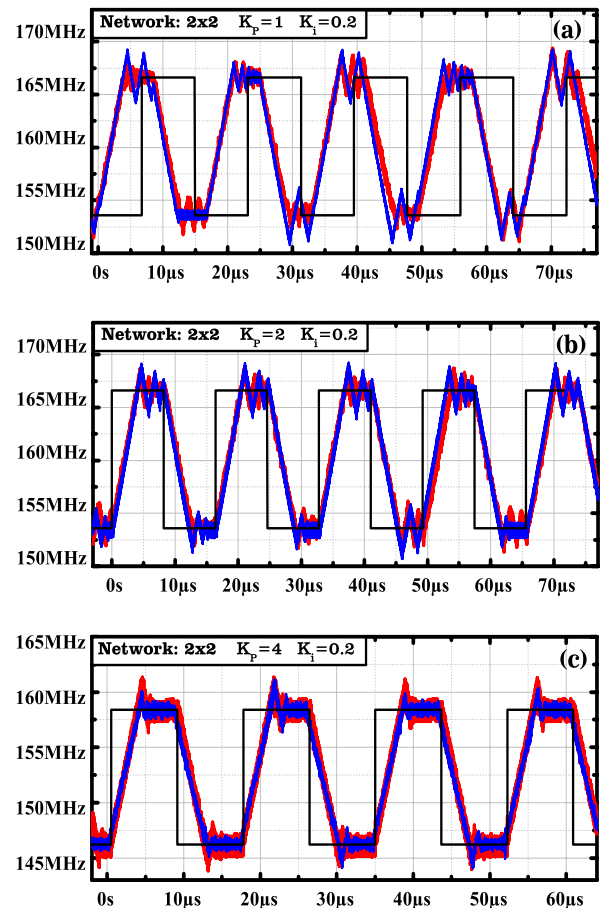

FIGURE 4: Comparison of simulations (blue) with experimental results (red) — transient waveforms showing the ability of the ADPLL network to synchronise to a varying reference frequency F^R switching between two values. The configurations of the network for the three presented cases are given in the legends.

network which, in principle, can have an arbitrary topology. The algorithms facilitate fast and accurate simulations of the network to find its optimal behaviour and the sets of control parameters that will ensure this behaviour. The proposed model has been verified using a 65nm CMOS 2×2 network prototype and shown a very good agreement with the experiment. Based on the numerical simulations and experimental results, we conclude that the ADPLL network is capable of achieving synchronisation if the control parameters are selected according to our study.

REFERENCES

[1] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update 2016–2021," White paper, Feb. 2016.

[2] D. J. Jörg, A. Pollakis, L. Wetzel, M. Dropp, W. Rave, F. Jülicher, and G. Fettweis, "Synchronization of mutually coupled digital PLLs in massive MIMO systems," in 2015 IEEE International Conference on Communications (ICC), June 2015, pp. 1716–1721.

[3] A. Pollakis, L. Wetzel, D. J. Jörg, W. Rave, G. Fettweis, and F. Jülicher, "Synchronization in networks of mutually delay-coupled phase-locked loops," New Journal of Physics, vol. 16, no. 11, p. 113009, 2014.

[4] A. L. Swindlehurst, E. Ayanoglu, P. Heydari, and F. Capolino, "Millimeter-wave massive MIMO: the next wireless revolution?" IEEE Communications Magazine, vol. 52, no. 9, pp. 56–62, Sept 2014.

[5] S. Ruffini, P. Iovanna, M. Forsman, and T. Thyni, "A novel SDN-based architecture to provide synchronization as a service in 5G scenarios," IEEE Communications Magazine, vol. 55, no. 3, pp. 210–216, 2017.

[6] H. Li, L. Han, R. Duan, and G. M. Garner, "Analysis of the synchronization requirements of 5G and corresponding solutions," IEEE Communications Standards Magazine, vol. 1, no. 1, pp. 52–58, 2017.

[7] D. A. Pokamestov, Y. V. Kryukov, E. V. Rogozhnikov, R. R. Abenov, and A. Y. Demidov, "Concepts of the physical level of the fifth generation communications systems," Radioelectronics and Communications Systems, vol. 60, no. 7, pp. 285–296, 2017.

[8] X. Li, M. Samaka, H. A. Chan, D. Bhamare, L. Gupta, C. Guo, and R. Jain, "Network slicing for 5G: Challenges and opportunities," IEEE Internet Computing, vol. 21, no. 5, pp. 20–27, 2017.

[9] D. Zachariah, S. Dwivedi, P. Händel, and P. Stoica, "Scalable and passive wireless network clock synchronization in LOS environments," IEEE Transactions on Wireless Communications, vol. 16, no. 6, pp. 3536–3546, 2017.

[10] D. T. L. Roche, B. Champagne, I. Psaromiligkos, and B. Pelletier, "On the use of distributed synchronization in 5G device-to-device networks," in 2017 IEEE International Conference on Communications (ICC), 2017, pp. 1–7.

[11] R. Gentz, A. Scaglione, L. Ferrari, and Y. W. P. Hong, "Pulsess: A pulse-coupled synchronization and scheduling protocol for clustered wireless sensor networks," IEEE Internet of Things Journal, vol. 3, no. 6, pp. 1222–1234, 2016.

[12] D. Kannapan and F. Bullo, "Synchronization in pulse-coupled oscillators with delayed excitatory/inhibitory coupling," SIAM Journal on Control and Optimization, vol. 54, no. 4, pp. 1872–1894, 2016.

[13] A. V. Proskurnikov and M. Cao, "Synchronization of pulse-coupled oscillators and clocks under minimal connectivity assumptions," IEEE Transactions on Automatic Control, vol. PP, no. 99, pp. 1–1, 2016.

[14] X. Gu, D. Liu, C. Baks, A. Valdes-Garcia, B. Parker, M. R. Islam, A. Natarajan, and S. K. Reynolds, "A compact 4-chip package with 64 embedded dual-polarization antennas for W-band phased-array transceivers," in 2014 IEEE 64th Electronic Components and Technology Conference (ECTC), May 2014, pp. 1272–1277.

[15] K. Zheng, L. Zhao, J. Mei, B. Shao, W. Xiang, and L. Hanzo, "Survey of large-scale MIMO systems," IEEE Communications Surveys Tutorials, vol. 17, no. 3, pp. 1738–1760, 2015.

[16] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K. K. Wong, R. Schober, and L. Hanzo, "User association in 5G networks: A survey and an outlook," IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1018–1044, 2016.

[17] T. E. Bogale and L. B. Le, "Massive MIMO and mmWave for 5G wireless hetnet: Potential benefits and challenges," IEEE Vehicular Technology Magazine, vol. 11, no. 1, pp. 64–75, March 2016.

[18] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 1617–1655, 2016.

[19] X. Ge, J. Yang, H. Gharavi, and Y. Sun, "Energy efficiency challenges of 5G small cell networks," IEEE Communications Magazine, vol. 55, no. 5, pp. 184–191, 2017.

[20] M. Koivisto, M. Costa, J. Werner, K. Heiska, J. Talvitie, K. Leppänen, V. Koivunen, and M. Valkama, "Joint device positioning and clock synchronization in 5G ultra-dense networks," IEEE Transactions on Wireless Communications, vol. 16, no. 5, pp. 2866–2881, 2017.

[21] C. Shepard, H. Yu, N. Anand, E. Li, T. Marzetta, R. Yang, and L. Zhong, "Argos: Practical many-antenna base stations," in Proceedings of the 18th annual international conference on Mobile computing and networking. ACM, 2012, pp. 53–64.

[22] N. Panwar, S. Sharma, and A. K. Singh, "A survey on 5G: The next generation of mobile communication," Physical Communication, vol. 18, Part 2, pp. 64–84, 3 2016.

[23] J. Zhang, P. Tang, L. Tian, Z. Hu, T. Wang, and H. Wang, "6–100 GHz research progress and challenges from a channel perspective for fifth generation (5G) and future wireless communication," Science China Information Sciences, vol. 60, no. 8, p. 080301, 2017.

[24] G. Pratt and J. Nguyen, "Distributed synchronous clocking," IEEE Transactions on Parallel and Distributed Systems, vol. 6, no. 3, pp. 314–328, 1995.

[25] E. G. Friedman, "Clock distribution networks in synchronous digital integrated circuits," Proceedings IEEE, vol. 89, no. 5, pp. 665–692, 2001.

[26] V. Gutnik and A. Chandrakasan, "Active GHz clock network using distributed PLLs," IEEE Journal of Solid-State Circuits, vol. 35, no. 11, pp. 174–175, 2000.

[27] A. Agrawal and A. Natarajan, "A scalable 28GHz coupled-PLL in 65nm CMOS with single-wire synchronization for large-scale 5G mm-wave arrays," in 2016 IEEE International Solid-State Circuits Conference (ISSCC), Jan 2016, pp. 38–39.

[28] C. Shan, E. Zianbetov, F. Anceau, O. Billoint, and D. Galayko, "A distributed synchronization of all-digital PLLs network for clock generation in synchronous SOCs," in New Circuits and Systems Conference (NEWCAS), 2015 IEEE 13th International, June 2015, pp. 1–4.

[29] C. Shan, D. Galayko, F. Anceau, and E. Zianbetov, "A reconfigurable distributed architecture for clock generation in large many-core SoC," in Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), 2014 9th International Symposium on, 2014, pp. 1–8.

[30] J.-M. Akre, J. Juillard, D. Galayko, and E. Colinet, "Synchronization analysis of networks of self-sampled All-Digital Phase-Locked Loops," Circuits and Systems I: Regular Papers, IEEE Transactions on, vol. 59, no. 4, pp. 708–720, 2012.

[31] J. Akré, J. Juillard, M. Javidan, E. Zianbetov, D. Galayko, A. Korniienko, and E. Colinet, "A design approach for networks of self-sampled All-Digital Phase-Locked Loops," in Proc. IEEE European Conf. Circuit Theory and Design (ECCTD), 2011, pp. 725–728.

[32] E. Koskin, E. Blokhina, C. Shan, E. Zianbetov, O. Feely, and D. Galayko, "Discrete-time modelling and experimental validation of an all-digital PLL for clock-generating networks," in 14th IEEE International New Circuits and Systems Conference (NEWCAS), June 2016, pp. 1–4.

[33] E. Koskin, D. Galayko, O. Feely, and E. Blokhina, "Semianalytical model for high speed analysis of all-digital PLL clock-generating networks," in 2017 IEEE International Symposium on Circuits and Systems (ISCAS), May 2017, pp. 1–4.

[34] E. Zianbetov, D. Galayko, F. Anceau, M. Javidan, C. Shan, O. Billoint, A. Korniienko, E. Colinet, G. Scorletti, J. Akré, and J. Juillard, "Distributed clock generator for synchronous SoC using ADPLL network," in Custom Integrated Circuits Conference (CICC), 2013 IEEE, Sept 2013, pp. 1–4.