

Clock Technology: The Next Frontier

Liming Xiu

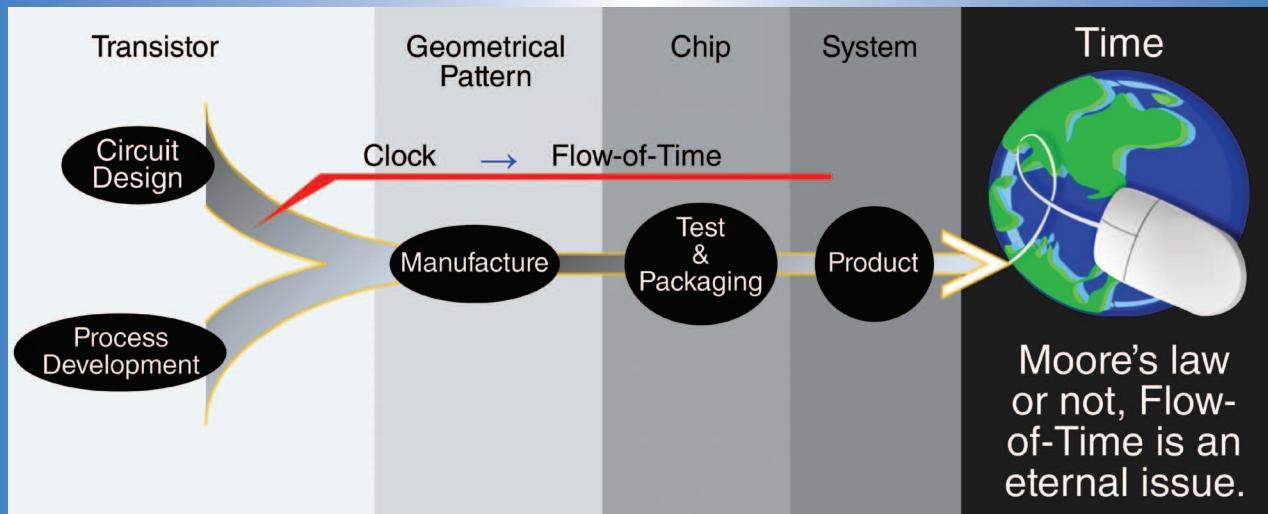


IMAGE CREATED BY AUTHOR LIMING XIU, IMAGE COURTESY OF LIMING XIU

Abstract

Clocking of electrical circuit is a crucial issue since clock signal is used to establish the *flow-of-time* inside electronic world. Before, during and after the “Moore’s law”, *flow-of-time* is an eternal issue. Alongside processor, memory/storage and analog/RF technologies, IC clocking could be regarded as the 4th major IC design technology. In the past several decades, clock is mostly used in the form of fixed-frequency with high frequency stability. For future system, however, this type of clock signal is not sufficient because its usage environment is not expected to be stationary but dynamic. To meet this challenge, innovation in IC clocking is required. This paper first discusses the difficulties in creating *flow-of-time*; then the two long-lasting problems in clock generation are identified, and then new challenges in the design of future system are summarized. Afterwards, the Time-Average-Frequency based flexible clock generator is introduced and its potential to confront these challenges is addressed. Several major issues in modern design are discussed. The paper concludes with a vision that, for electronic system to improve its information processing efficiency to next level, clock technology is the next frontier to be explored.

Part I: The Problem

I.1 A Critical Issue Inside Electronic World: The Difficulty in Dealing with Time

We need two variables, voltage level and time, to describe an electrical signal, either analog or digital. The waveform of *voltage vs. time* completely illustrates (for an output) or defines (for an input) the behavior of a given signal. Therefore, it is immensely important to discuss how all electronic devices recognize the variables of voltage level (hereafter referred to as voltage) and time. The four fundamental elements in the family (resistor, capacitor, inductor and memristor), which form the basic building blocks of the entire electronic world, all follow the law of proportionality¹: the force is proportional to the volume of objects. In other words, the strength of a device’s response to a stimulant depends on the number of electrons involved in the process. This

Digital Object Identifier 10.1109/MCAS.2017.2689519

Date of publication: 19 May 2017

¹In this paper, law of proportionality is used as in the field of mathematics, not field of law. An example, ohm’s law.

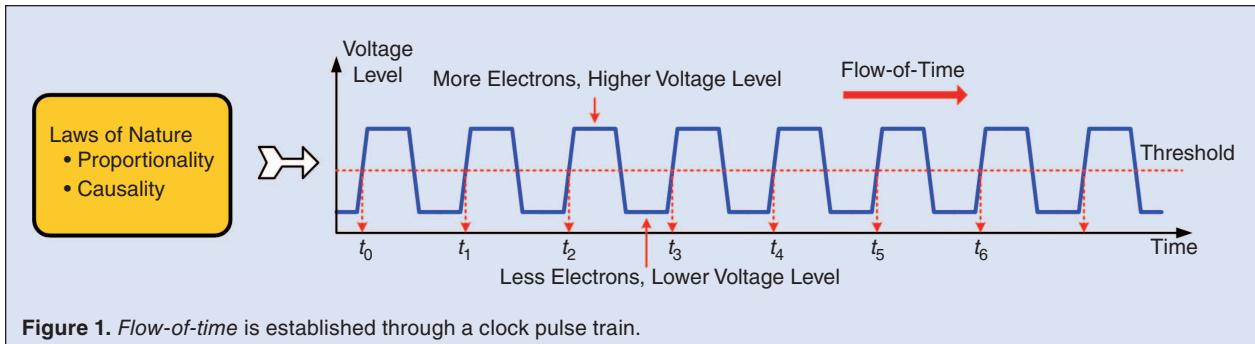


Figure 1. Flow-of-time is established through a clock pulse train.

property makes it possible for us to use voltage, inside electronic world, to represent information².

When *time* is concerned, unfortunately, none of these devices recognize the *concept of time*. They do not know the ideas of *past*, *present* and *future*. However, they do respect the causality: the law of cause & effect. As an example, for a resistor, a meaningful electrical voltage appears only *after* an electrical current is injected into it, not *before*. This property enables us to establish an *order of sequence*, which is the foundation for creating *flow-of-time* inside electronic world. Therefore, inside electronic world, voltage and current can be naturally associated with the basic building elements. On the other hand, a nontrivial mechanism is needed to establish the *flow-of-time*.

I.2 In The Making of Clock Signal:

The Two Long Lasting Problems

For all chips, the execution of intended task is accomplished only with the help of electrical signals, millions of them in a sophisticated chip. Among those signals, the most important one is the clock signal. **Clock signal is the mechanism that establishes the flow-of-time.** Without it, the order for all events cannot be arranged and, consequently, no useful tasks can be performed. Together, the two properties of **proportionality** and **causality** provide a possibility for us to create this special clock signal. In engineering practice, as illustrated in figure 1, a clock signal is created as an electrical pulse train whose voltage level alternates between high and low. *Moments-of-time* are created when the pulse train makes low-to-high (or high-to-low) transitions. These *moments*, t_0, t_1, t_2, \dots , are the points on the horizontal axis that correspond to the *crossover-points* when the transitions cross a predefined threshold. The most important parameter of the clock pulse train is the time span between any two adjacent *moments*, such as t_1 and t_0 . This is the period whose inversion, frequency, is commonly used to quantify the “speed” of a clock pulse train. The crucial requirement in the construction of clock signal is the precision associated with the locations of those *moments-of-time*. Those moments must occur at the locations as close to the prediction as possible. The quality of this implementation is quantified by the parameters of phase noise and jitter.

In the past several decades, tremendous effort has been spent on processor, memory/storage and analog/RF technologies. Significant advances have been made in these areas. However, among the four major IC design technologies depicted in figure 2, clock technology has fallen behind. As illustrated, clock is fundamentally important; it serves as the driver of all the others. Clock technology has three subjects of study: clock usage, clock distribution and clock generation. Clock generation further includes four key issues, as shown. The task of clock generation is often referred as frequency synthesis since the most distinguishing characteristic of a clock signal is its frequency. In the history of frequency synthesis, there are three major approaches: Direct Analog Synthesis (the mix-filter-divide approach) [1], Direct Digital Synthesis (the look-up-table approach) [2], and Phase Locked Loop (PLL, the compare-then-correct approach, the indirect approach) [3]–[4]. Among them, the first and second ones are routinely found as stand-alone solutions. For on-chip clock generation, PLL is the most widely used approach due to its easy integration with other on-chip circuits.

From a user perspective, when a clock signal is evaluated, the concerns include output frequency range (tuning range), frequency resolution (frequency granularity), frequency stability (phase noise, jitter) and frequency switching speed (settling time). From a hardware design perspective, unfortunately, these requirements are often mutually contradictory. Abstractly, these requirements can be itemized as four targets: high frequency, low phase noise/jitter, small frequency granularity and fast frequency switching. In the past, most clock-generation-related

²For information processing purpose, electrical current is usually converted into voltage before being processed.

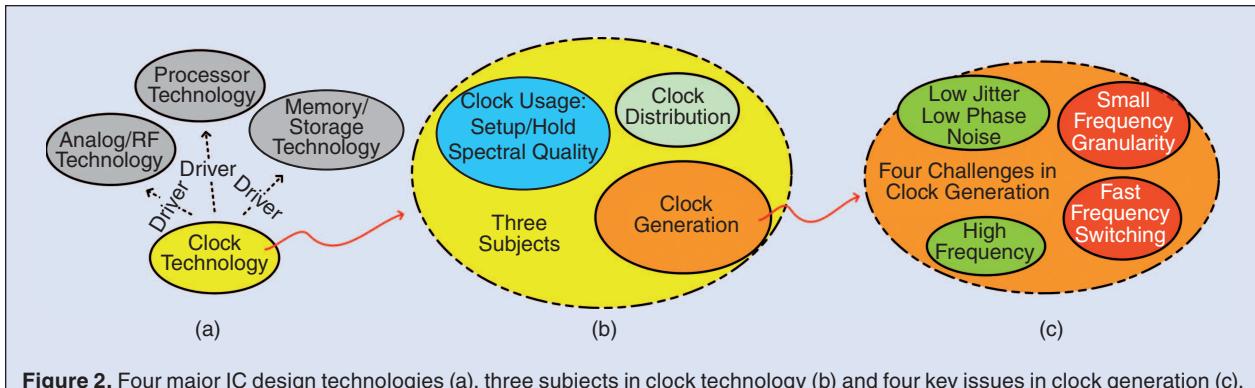


Figure 2. Four major IC design technologies (a), three subjects in clock technology (b) and four key issues in clock generation (c).

work has been focused on the first two targets, and it would be fair to say that these two issues have been well understood and decent solutions are available. The third and fourth targets, however, remain to be problems that have not been solved to our complete satisfaction. In other words, *small frequency granularity* and *fast frequency switching* are the two long lasting problems in the field of on-chip clock generation.

I.3 New Challenges in Electronic System Design: The Call for Flexible Clock

In the past, a fixed-frequency clock signal would very likely be sufficient for most applications. For this reason, clock generators of the past have focused their attention on just generating, for any given application, a selected few frequencies with high frequency stability. In addition, fast frequency switching speed is not considered as a high design priority. It is reasonable to term this kind of clock signal as *rigorous* (or *rigid*) clock. After more than a half century's evolution, the sophistication of IC design has reached a very high level. Moving forward, the key challenges that lie at the forefront of future electronic system design are listed as below.

- Connect every electronic device to a network (Internet of Things)
- Move ever-increasing amount of data among heterogeneously clocked systems
- Reduce power consumption, improve energy efficiency (= performance/power)
- Alleviate EMI (electromagnetic interference) problem
- Improve network time synchronization accuracy (e.g. to ns range)
- Enhance sensing capability, improve measurement accuracy
- Reconfigurable hardware, improve hardware programmability
- Embed security into electronic devices

- Miniaturization challenge
- Interoperability challenge
- Heterogeneous system architecture: multi-core, parallel computing, reconfigurable computing
- Operate under rougher environment (larger variation on supply voltage, bigger swing on temperature and higher degree of EM interference).

As can be understood from aforementioned challenges, the design of future electronic system is more complex and thus requires additional features from the clock, *ample frequency* and *quick frequency switching* in particular. A clock signal of such characteristics could be called *flexible* clock, in contrast to the rigorous clock. A clock generator capable of generating such flexible clock signal is termed Field Programmable Frequency Generator (FPFG).

The value of FPFG could be appreciated from the illustration in figure 3 and 4. The operating environment of modern electronic systems is constantly changing rather than stationary. To reach an optimum solution, it demands an adaptive clock signal that can quickly respond to the change occurring in the environment. During this process of optimization, the finer the frequency tuning step (frequency resolution, frequency granularity) is, the better; the quicker the frequency can be switched from one value to another, the better. This feedback based optimization process is further illustrated in figure 4 where the *sense-compare-adjust* mechanism can be compared to that of PLL. In contrast to the VCO (Voltage Controlled Oscillator, an analog circuit component) in a PLL³ [5]–[6], the response speed of the FPFG must be precisely quantifiable. This will lead to a quantifiable loop latency of the feedback loop, in terms of clock cycle. For a given application scenario, this mechanism can enhance the possibility for the system to reach an optimal solution.

³In a Phase Locked Loop, the loop latency is hard to be precisely calculated since the VCO's response speed is difficult to be quantified.

In conclusion, modern electronic system design calls for flexible clock.

Part II: A Solution

II.1 The Concept of Clock Frequency: Rethinking

Given clock signal's immense influence on the structure and the performance of a chip and its associated system, all the aforementioned challenges are related to the clocking in one way or another. It is clear that innovation in clocking is required to confront the challenges. A flexible clock generator is defined as the one that has the following capabilities: *arbitrary frequency*

generation and instantaneous frequency switching. In addition, for a given application, these two features have to be achieved at the same time. Our goal is to be able to generate any frequency that we want, and that the clock's frequency can be changed quickly at our command. This is, however, not an easy task since *flow-of-time* is indirectly created from voltage transitions, as illustrated in figure 1.

Ever since clock signal was introduced into the field of VLSI system design, the golden rule has been that "all the cycles in a clock pulse train of certain frequency have to have same length-in-time". All the clock generation techniques, including integer-N PLL, fractional-N

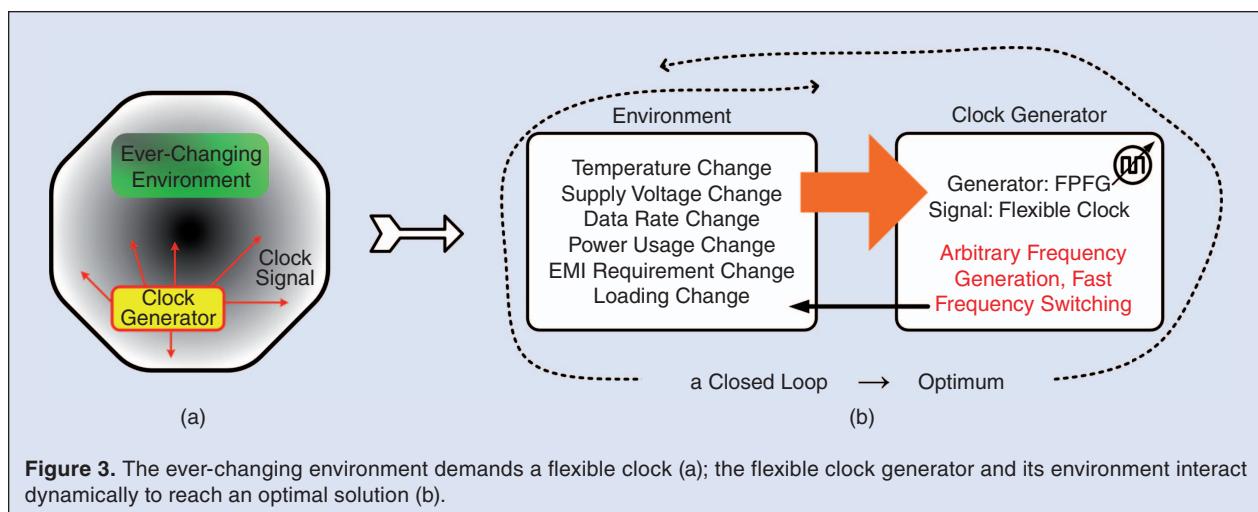


Figure 3. The ever-changing environment demands a flexible clock (a); the flexible clock generator and its environment interact dynamically to reach an optimal solution (b).

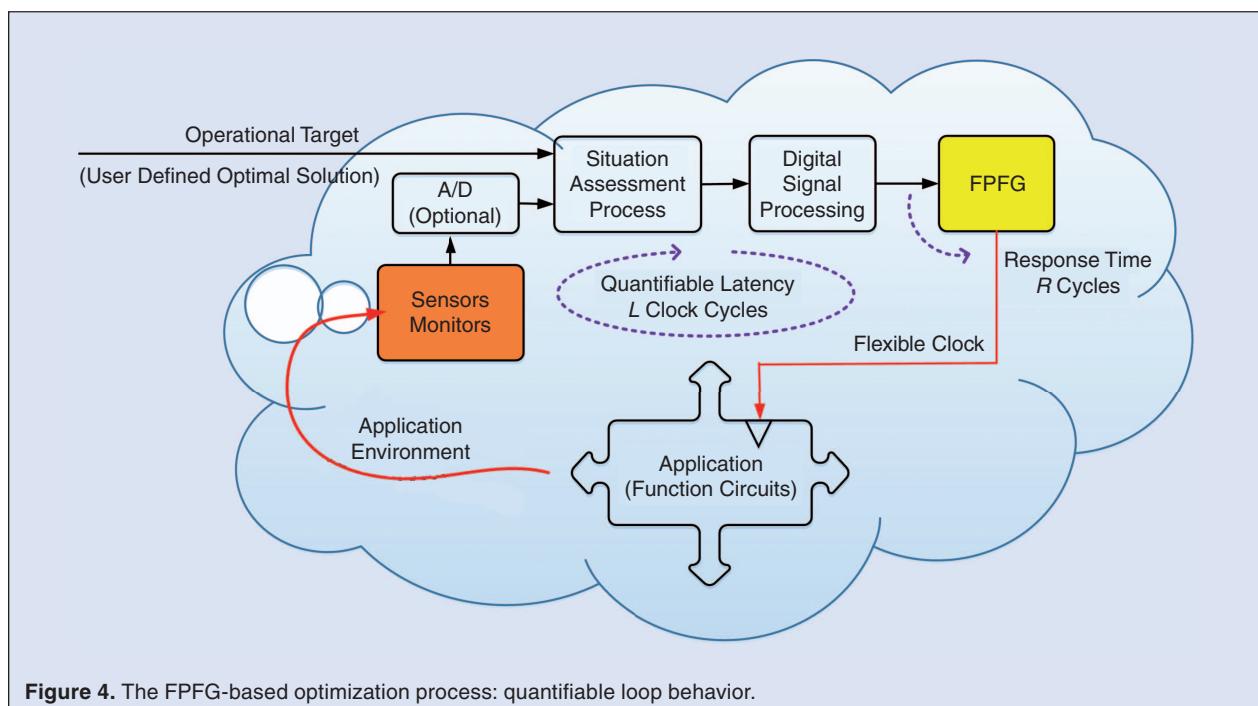


Figure 4. The FPFG-based optimization process: quantifiable loop behavior.

PLL, DDS and others, use this criterion to guide their design practices. In circuit implementation, any resulting cycle whose length-in-time is different from that target value is considered as imperfect cycle. A clock signal consisting of such cycles is viewed as a “jittery clock”. It is believed, however, that this “golden rule” is the key reason for the persistence of the aforementioned two long-lasting problems. It constrains our hand when we are creating clock signal. In order to make a breakthrough in IC clocking, now is the time to ask the following questions.

- What does the concept of clock frequency really mean?
- Can this golden rule be challenged?
- What could be gained if this constraint of “all cycles have to be equal in their lengths” is removed?

II.2 Time-Average-Frequency Based Flexible Clock Generator

The concept of clock frequency is defined as “the number of clock cycles existing within a time frame of one second”. When a clock signal is used to drive a circuit, this definition can be translated into “the number of operations carried out within the time frame of one second”. For example, a processor running at a clock frequency of 1 GHz requires one billion operations to be executed within one second. This is the only hard requirement regarding the 1 GHz clock frequency. As long as one billion operations are successfully finished within one second, the processor does not care how each operation is carried out in detail. As a matter of fact, at a higher level,

the processor does not have the capability to tell what each cycle exactly looks like. This understanding leads to the belief that, in the definition of clock frequency, the key point is the *number of operations*, not the *structure of each cycle*. This opens up the possibility that the “golden rule” does not necessarily have to be honored if there are other more important design concerns. It is our desire to use this opportunity to attack those two long-lasting problems.

In recent years, a novel concept of Time-Average-Frequency has been proposed [7]. It removes the constraint that all cycles’ lengths have to be equal. It is rigorously defined by the number of operations. It intentionally uses multiple types of cycles to achieve some hard-to-reach frequency values. To our best knowledge, that was the first time that clock frequency concept is reinvestigated since clock was introduced into IC design to drive circuit. From this concept, a new type of clock generator, Time-Average-Frequency Direct Period Synthesis (TAF-DPS), emerged [8]–[9]. The block diagram at the top of figure 5 depicts the TAF-DPS architecture showing its work principle. Starting from a base time unit Δ (e.g. 25 ps), TAF-DPS creates two type of cycles $T_A = I \cdot \Delta$ and $T_B = (I+1) \cdot \Delta$, where I is an integer. Then, these two types of cycles are used in an interleaved fashion to make the output clock pulse train. The possibility of T_A (and thus T_B) occurrence is determined by a fraction r , $0 \leq r < 1$. As a result, the output period can be calculated using a function of $T_{\text{TAF}} = 1/f_{\text{TAF}} = F \cdot \Delta = (I+r) \cdot \Delta$, where $F (= I+r)$ is termed as frequency (more precisely, period) control word.

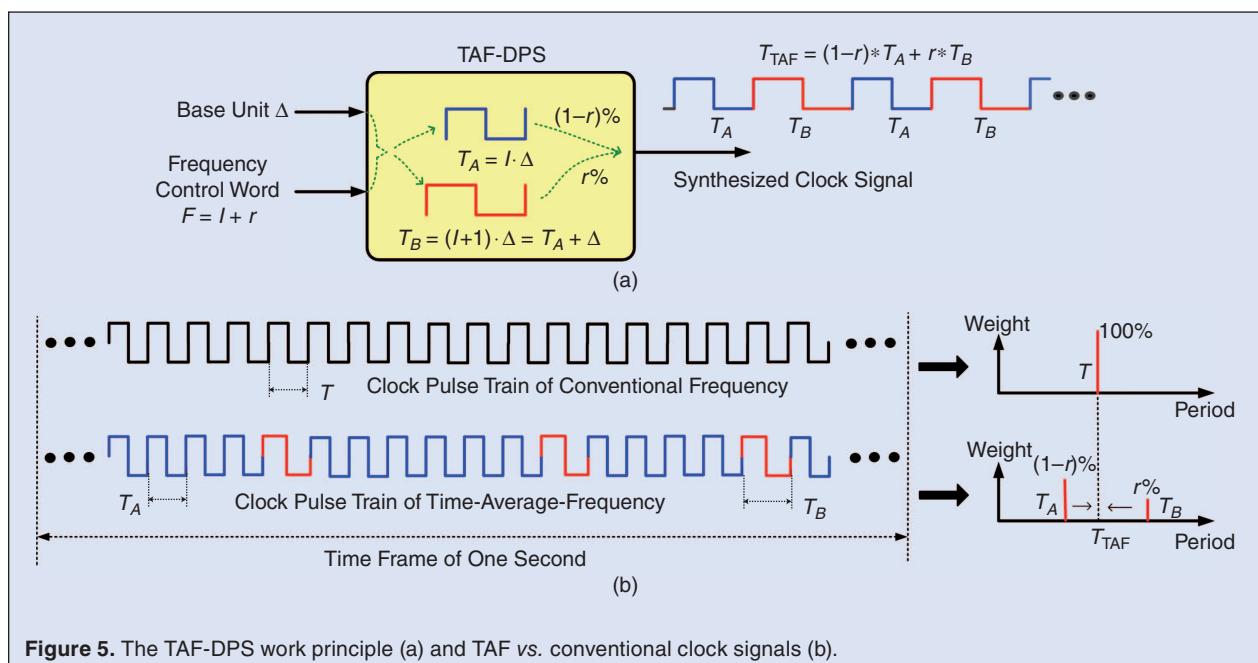


Figure 5. The TAF-DPS work principle (a) and TAF vs. conventional clock signals (b).

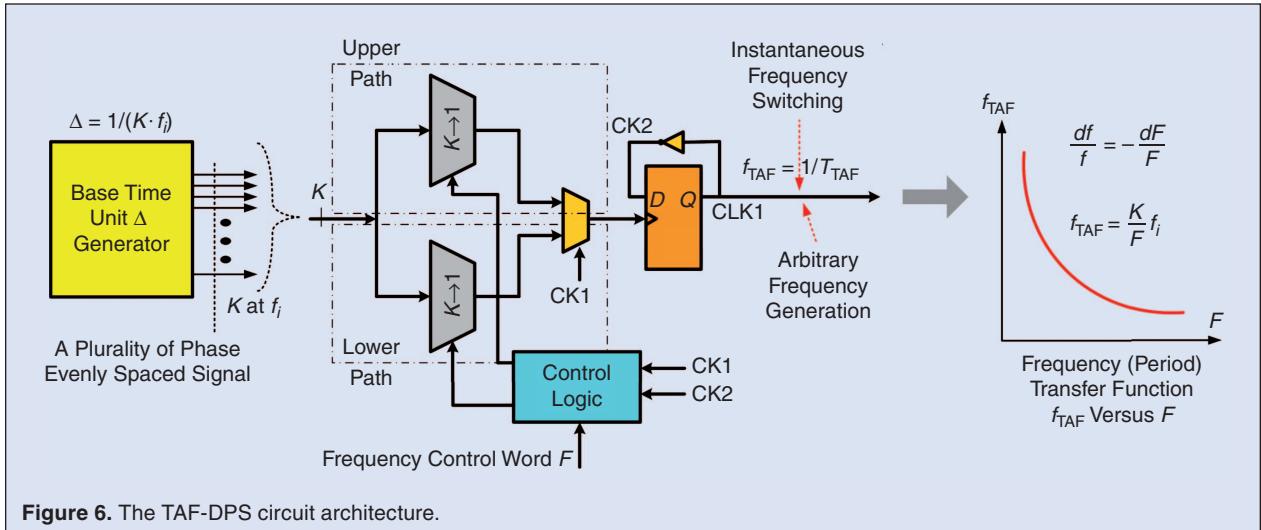


Figure 6. The TAF-DPS circuit architecture.

The waveforms shown at the bottom illustrate the difference between conventional frequency and Time-Average-Frequency clock signals⁴. It is understandable that, for a given design and given enough resource, almost any period (frequency) can be generated since both I and r are controllable by the designer. The frequency resolution/granularity is determined by r (the LSB of r). Frequency granularity of <2 ppb has been achieved in experiment. For this reason, TAF-DPS is regarded as a clock generator capable of “arbitrary frequency generation”.

The second problem of fast frequency switching is also addressed by TAF-DPS. This can be elucidated with the assistance of figure 6. As shown, the base time unit Δ is generated from a plurality of K phase-evenly-spaced signals of known frequency. The value of Δ is the time span between any two adjacent signals. The said plurality of signals are fed into the direct period synthesizer that synthesizes each output pulse’s low and high portions⁵. In contrast to PLL which is an indirect approach, each TAF-DPS output cycle’s structure is directly constructed and individually controlled. Thus, its period (and frequency) can be instantly changed. Further, the frequency switching speed is quantifiable⁶. This leads to the so-called “instantaneous frequency switching”.

In the right hand side of figure 6, the frequency transfer function is graphically illustrated. The transfer function takes the $1/x$ format since TAF-DPS’s output period is linearly proportional to the value of control word F . In this curve, almost all the frequency point can be

reached as long as enough bits are used in the fraction r . The frequency resolution/granularity can be calculated as $df/f = -dF/F$, where dF is the LSB of r . As an example of a real case, $r = 2^{-27}$ and $I = 4$ leads to a frequency granularity df/f of 1.89 ppb (from calculation). In lab measurement, the real number obtained is 1.99 ppb. Such a small frequency granularity has never been reported from other techniques. Another important characteristic of TAF-DPS is that the frequency transfer function is precisely predictable (traceable with 100% fidelity). This is extremely valuable while it is used in applications.

The most well known TAF-DPS implementation is the Flying-Adder frequency synthesis architecture [10]–[14], which has been used in commercial products for over a decade. “Arbitrary frequency generation” and “instantaneous frequency switching” are its key advantages, valued in the design-decision-making process. With frequency granularity in ppb range and frequency switching speed of two cycles, TAF-DPS is the type of clock generator that is qualified as the FPFG in figure 4. When this FPFG is used in the optimization loop, its response time (the FPFG latency) is $R = 2$ cycles. Consequently, the loop latency L can be calculated in terms of cycle since all other elements in the loop can be likely designed with fixed and known response time. This calculable loop behavior is fairly valuable to system designer.

As evidenced from figure 5, Time-Average-Frequency clock signal is different from conventional clock signal. Thus, theoretical work is required to understand its impact both in time and frequency domains. Since the majority of clock usage is to drive digital circuits, the impact of TAF-DPS on driving digital circuits has been studied and is well understood [8]–[9]. Its usage has been proven in commercial products for over a decade (an example can be found in [15]). When TAF-DPS is

⁴TAF clock signal can be safely used to drive circuit as long as T_d is used as the setup constraint (please refer to section 4.22 of [8]). It is also possible that more than two types of cycles be used in a TAF clock signal.

⁵The TAF-DPS circuit detail is extensively discussed in section 4.4 of [8].

⁶In TAF-DPS implementation, switching speed is typically constructed as two cycles, please refer to section 4.5 of [8].

used in applications where spectral purity is a concern, theoretical studies have also been carried out to rigorously derive its spectrum in various scenarios [16]–[20]. Furthermore, theoretical work on the subject of manipulating the TAF clock signal's spectrum to our advantage has begun as well [21].

Part III: The New Frontier

After over a decade's circuit practice and the recent advancement on theoretical work, TAF-DPS is ready as the clock generator for producing the much desired flexible clock. It is time to apply it to system level for enabling innovation in a higher domain. In the following sections, new frontier emerged from the novel TAF-DPS clock source will be discussed. Because of the omnipresence of clock signal in VLSI system, the discussion will be cross-disciplines including VLSI system's logic and physical implementations, computer architecture, network time synchronization, sensor design, frequency signal source construction and etc. In these applications, all the innovations are discussed around the newly available features from the flexible clock: *arbitrary frequency generation* and *instantaneous frequency switching*.

III.1 Flexible Clock and Electronic System's Precision, Accuracy and Stability

Electronic system is used for processing information. Inside chip, information presents itself as data flows, which is controlled by clock (frequency) source. In figure 7, a generic transceiver architecture is depicted. The TX and RX are controlled by their clock sources CLKT and CLKR, respectively. CLKT and CLKR are usually independent of each other (they don't share a common reference). Their real time frequencies are $f_t(t)$ and $f_r(t)$, which vary with time due to environmental disturbances. Therefore $f_t(t) = f_{t0} + \delta f_t(t)$ and $f_r(t) = f_{r0} + \delta f_r(t)$ are not equal most of the times, where f_{t0} and f_{r0} are the stationary target values and $\delta f_t(t)$ and $\delta f_r(t)$ are their variations, respectively. To ensure successful data transmission, the average frequencies of the two sides must be equal: $f_{t_avg} = f_{r_avg}$. The quality of

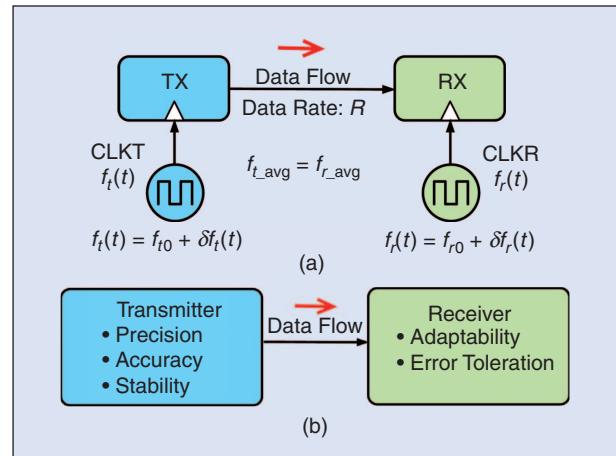


Figure 7. Successful data transmission is influenced by frequencies sources (a) the desirable qualities on transmitter and receiver (b).

an electronic system can be judged by the characteristic of $\delta f(t)$. For transmitter, as illustrated in the right hand side of figure 7, it is preferable to be *precise*, *accuracy* and *stable*. For receiver, *adaptability* and *error toleration* are the desirable qualities.

Figure 8 illustrates the meanings of *precision* and *accuracy* of an electronic system. They are defined from the perspective of data rate, which are solely influenced by the quality of the clock (frequency) source: the $\delta f(t)$. The *stability* is defined as the system's capability to preserve its precision and accuracy under external environmental disturbances. Figure 9 illustrates the process of improving the system's stability by the actions of calibration and compensation. In the figure, the ideal source represents the desired frequency value of a real clock generator. It is unavoidable that some sort of imperfection could be introduced in the manufacture process, which leads to error in frequency value. Further, the source's output could vary due to environment change, which will negatively impact the precision and accuracy. After the calibration and compensation, hopefully, the system's stability can be improved as illustrated. In this processes of calibration and compensation, the flexible

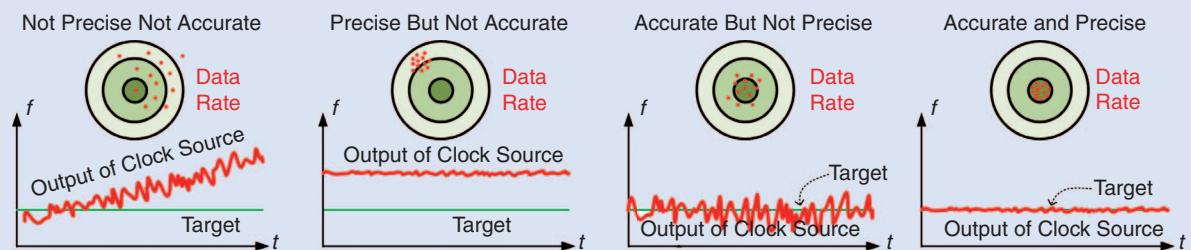
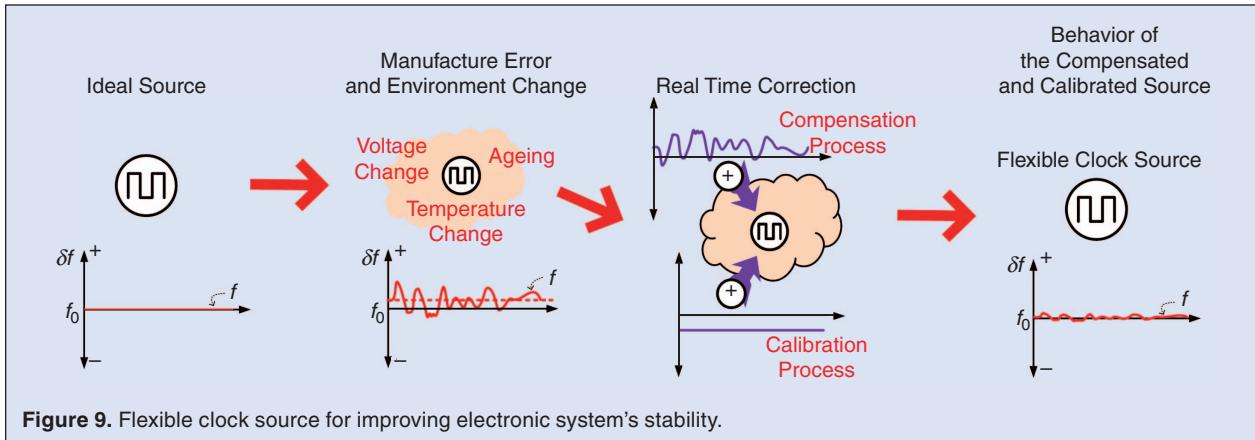


Figure 8. Electronic system's precision and accuracy are defined by its frequency source.



clock source of *arbitrary frequency generation* and *instantaneous frequency switching* is the key.

III.2 Flexible Clock for Controlling Data Flow: FIFO Memory and Multi-Cores

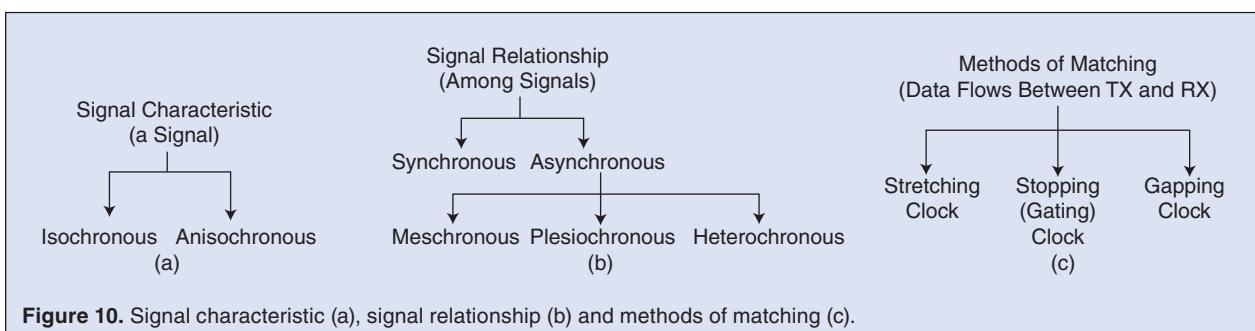
The discussion around figure 7–9 illustrates the importance of the quality of the clock source on an electronic system's operation, where data flow is the main concern. Before our further investigation on data flow, it is beneficial to review the terminology describing signal, signal relationship and the method of matching data flows. As depicted in the left of figure 10, a digital signal can be characterized by the time interval that it uses to represent one bit of information. In the case of *isochronous*, all the bits use the same length of time interval. Otherwise, it is *anisochronous*: the time interval separating two significant instants of a signal does not necessarily relate to the time interval separating two other significant instants.

When more than one signal is concerned, as depicted in the middle of figure 10, the relation among the signals can be classified as *synchronous* and *asynchronous*. Synchronous signals are generated by same clock, or by clocks sharing a common reference (which have exact same frequency but might have phase difference of fixed value). Asynchronous signals are driven by differ-

ent clocks, which naturally bear different frequencies. Inside asynchronous, there are three cases: *meschronous*, *plesiochronous* and *heterochronous*. For meschronous, the driving clocks have same frequency but with a random and static phase difference. In plesiochronous case, the driving clocks have same *average* frequencies but different instantaneous frequencies. Finally, for heterochronous, even the average frequencies are different.

When data is transferred between TX and RX, most likely, the mode will be asynchronous since TX and RX usually use their own clocks. To match the data flows, clock signals have to be adjusted from time to time. In right hand side of figure 10, three methods of matching are listed. Stretching prolongs or shortens an individual clock pulse at selected times. Stopping stops the clock pulses when needed (also referred as clock gating). In the case of gapping, one or more clock pulses are intentionally removed. In all the cases, the goal is to match the RX data flow to that of TX.

In asynchronous data transfer, it is expected that some sort of temporary storage is required to temporarily store data when it is moved from one place to another. For this reason, FIFO (First In First Out) memory is widely used in chip design. As depicted in figure 11.a, a FIFO is inserted between two data domains of different operating speeds. A data stream flows from left to right. The two domains



are clocked by CK_L and CK_R , respectively. CK_L and CK_R are not necessarily equal in frequency. As a result, the data rates of R_L and R_R can be different. This FIFO mechanism has two issues that are problematic. First of all, the size of FIFO shall be chosen as proportional to the rate difference of R_L and R_R , which could be quite large for certain designs. Secondly, to ensure no-data-lost, either the CK_L or CK_R has to be stopped, stretched or gapped occasionally. Therefore, the data flow is not continuous but interrupted from time to time.

Flexible clock is a new approach of controlling data flow. In principle, it is similar to the fact of stretching the clock. In practice, it is different from the stretching clock that the task of stretching is carried out not in transistor circuit level but in system control level, which is transparent to circuit designer and convenience to function/block/logic designer. In other words, the stretching capability is already built-in in the TAF-DPS clock generator; it is the underlying support for dynamical data flows matching.

In figure 11.b, a flexible clock based scheme is proposed. As shown, the data domain in the receiving side is clocked by a TAF-DPS flexible clock CK_R . The clock generator is informed with the status of CK_L . Thus, it continuously adjusts its output frequency to make R_R match to R_L . With the help of this adaptive clock generator, the size of FIFO can be much reduced. For an optimum result, the storage size can be reduced to the minimum of two data units [22]. Furthermore, the data flow becomes continuous now because there is valid data in every clock cycle. Since FIFO is used in almost all modern designs, this flexible clock enabled data communication scheme can have a profound influence.

This adaptive clock generator can also be used in assisting multi-cores architecture as depicted in figure 11.c. In multi-cores design, the cores are usually heterogeneously clocked. In this architecture, to accommodate the rate differences, each core can be accompanied with an interface adapter and a routing module. They are used to interface the core with other cores. Inside the interface adapter, the scheme for controlling the data flow is similar to the one presented in figure 11.b. The routing module is responsible for directing the data to the desired destination through the link. Compared to the conventional PLL based clocking, it is believed that this method can improve power and area efficiency. In this approach, *arbitrary frequency generation* and *instantaneous frequency switching* is the enabler for this data transfer mechanism.

III.3 Flexible Clock and IoT: The Task of Establishing “A Common View of Time”

The Internet of Things (IoT) is the third wave in the development of the Internet, following the development of

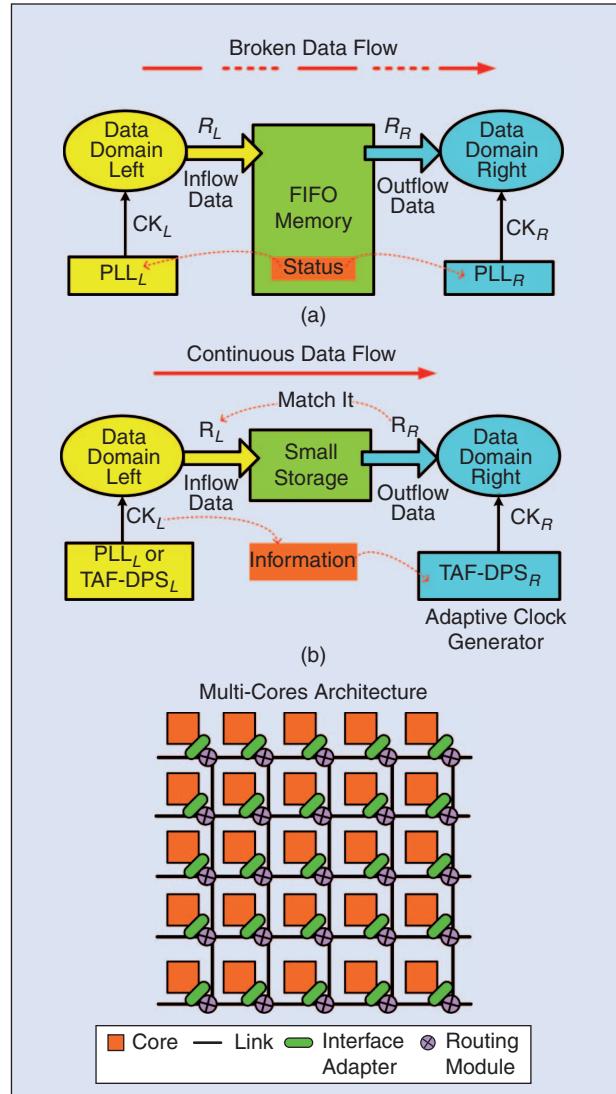


Figure 11. Flexible clock for controlling data flows in FIFO memory and multi-cores architecture.

network connectivity for desktops and then for mobile devices. IoT makes everyday objects connected and smarter. Due to IoT’s capability to collect and report real-time data in healthcare, education, entertainment, utility, insurance, business, and industrial environments, it is projected that it will revolutionize the way we live. The key challenges in IoT are protocols and standards that ensure that IoT technology can understand, for any given situation, the intention of the human, the applicability of security and the governance of laws. The distinguishing characteristic of IoT is connection: everything is connected to every other thing. For this reason, one of the cornerstones of IoT is to establish “a common view of time” among all the connected devices in the

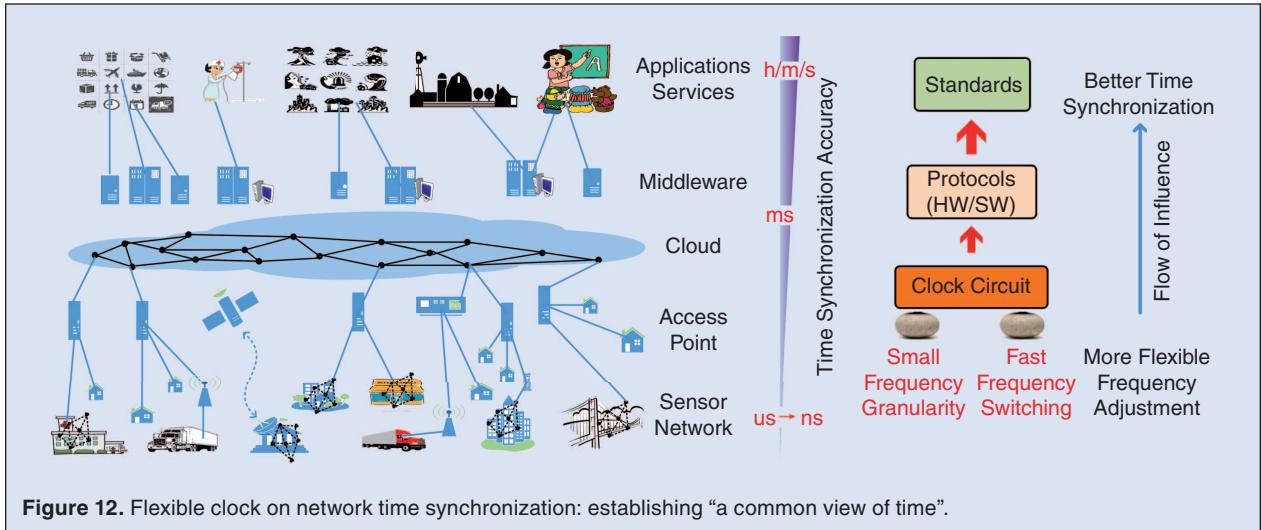


Figure 12. Flexible clock on network time synchronization: establishing “a common view of time”.

given network⁷. Since direct clock link (more precisely, direct frequency link) is impossible, this “common view of time” can only be established through the process of network time synchronization [23], which requires a good frequency source in each device connected to the network.

Time synchronization depends on the statistical characteristic of the network (network delay and etc.). Based on the cost and the required synchronization target, the task of synchronization can be carried out in software, in hardware or in hybrid mode. In all these methods, the fundamental building blocks of the time synchronization mechanism are techniques of *synchronization event detection*, *remote clock estimation* and *local clock correction*. They all have impact on the achievable synchronization precision (and accuracy when synchronizing to an external time reference). The synchronization precision is influenced by: $\pi = c_1 \cdot \varepsilon + c_2 \cdot P + c_3 \cdot G + c_4 \cdot u + c_5 \cdot G_s$, where π is the precision, ε is the transmission delay uncertainty when reading the remote clock, P is the clock drift (due to local oscillator frequency drift), G is the clock reading granularity, u is the rate adjustment granularity, G_s is the clock setting granularity and $c_{1,2,3,4,5}$ are the weighing factors [24]. The granularities G , G_s and u are all related to the frequency granularity of the time source in each device. In general, the smaller the frequency granularity is, the better the synchronization precision will be.

In IoT environment as illustrated in figure 12, the devices are required to be designed and manufactured under the constraints of extremely low cost, low power consumption and small form factor. Under those constraints, it would be difficult to build high quality frequency

source for each device using conventional approaches. Low quality frequency source however leads to bad time synchronization accuracy, which consequently degrades the quality of IoT service. Flexible clock provides a solution to this dilemma. In every IoT device, a low cost TAF-DPS synthesizer can be incorporated in the system to accompany the low quality (due to low cost as said) frequency source. Its purpose is, by using the small frequency granularity and fast frequency switching, to quickly and constantly adjust itself and adapt to its environment (with the help of adequate protocols) so that better time synchronization can be achieved [25]. In this direction, the TAF-DPS has already enabled the possibility at the circuit level. With additional work in protocol and standard levels, the pursuit of better time synchronization (and thus better IoT service) with low cost implementation could become feasible.

III.4 Frequency Granularity for High Precision Time-Based Signal Processing in Sensor

Sensors are crucial building blocks of IoT. Figure 13 depicts the general architecture of a sensor system. To detect the changes in our surrounding environment, many different types of sensors are required. The environmental changes we are interested in are electromagnetic field, voltage & current, temperature, pressure, liquid & gas flow, light intensity, Time-of-Flight, chemical composition, among others. Any such change can only present itself to us through one of the following types of mediums: electromagnetic radiation (light & radio wave), acoustic radiation (sound & ultrasound), particle radiation, mechanical force, heat, or a type of material. For a sensor system, the change is detected by a sensing element of particular type. The output from the sensing element must be in one of the formats: change of

⁷An analogy: the Greenwich Mean Time, if the whole world is considered as a network and each individual is considered as a node.

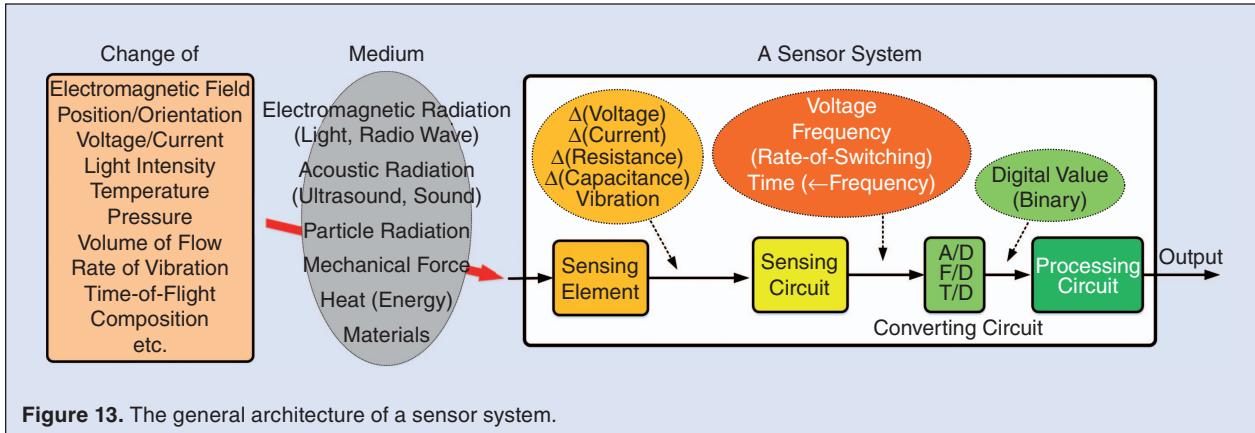
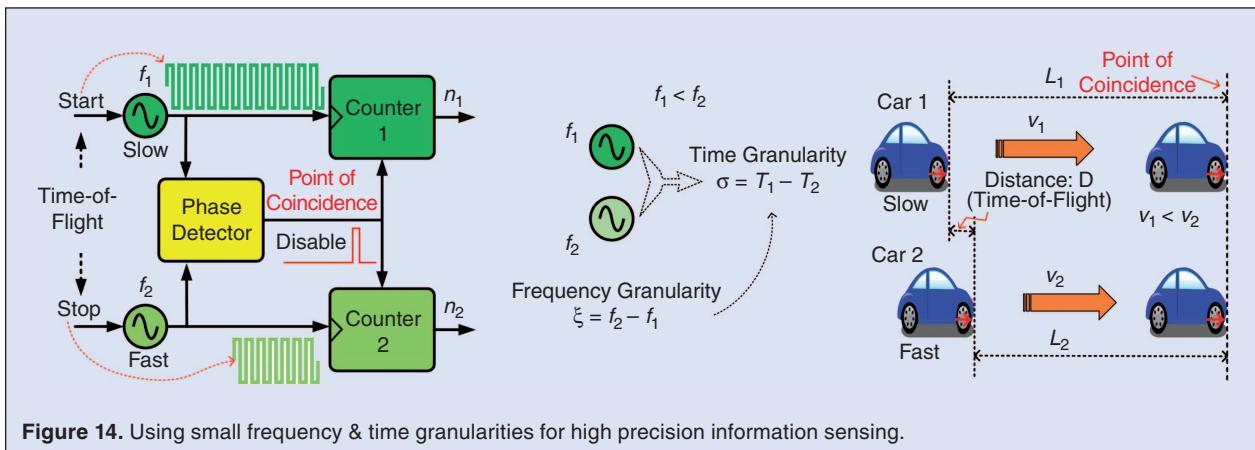


Figure 13. The general architecture of a sensor system.



(voltage, or current, or resistance, or capacitance) or mechanical vibration. Following the sensing element is the sensing circuit that converts the amount of change or the vibration into voltage, frequency or time. For electrical circuit, only voltage, frequency and time can be directly manipulated in quantifiable fashion and be used for signal processing⁸.

In electrical circuit, there are two approaches for signal sensing: voltage-based and time-based. In voltage-based signal sensing, electrical voltage is used to represent information and time is just for indexing. In time-based sensing, however, time is specifically used to convey messages. In circuit design, an important method of manipulating *time* is through frequency source. In a rudimentary level, a digital counter running at a known frequency can be used to serve this purpose. For sub-T time resolution, two frequency sources can be employed as shown in the left of figure 14. Two pulse trains run at slightly different frequencies can be used as a Vernier

caliper to take the measurement into sub-T range. This technique is useful for Time-of-Flight (TOF) measurement, which is demanded in many sensor applications of high measurement accuracy. The frequency and time granularities are illustrated in the middle of figure 14. Small frequency granularity $\xi = f_2 - f_1$ leads to small time granularity $\sigma = T_1 - T_2$, which subsequently leads to finer time resolution in the TOF measurement.

An analogy can be made by measuring the distance (corresponding to the TOF) between two cars, as illustrated in the right hand side of figure 14. To measure the distance, a simple method is to let car 1 run to car 2 (car 2 is stationary). The number of turns of its wheel is the indication of the distance $D = \text{number-of-turns} \times \text{circumference-of-the-wheel}$. However, the measurement accuracy is only good to the degree of one circumference (similarly, the time resolution of a digital counter is one cycle). To reach sub-circumference (sub-T) accuracy, we can let the two cars run at slightly different speeds of v_1 and v_2 , with $v_1 < v_2$. Since car 2 goes slightly faster, it will eventually catch up to car 1. When *point of coincidence* is reached, $D = L_1 - L_2$ is the measured result. This is the well-known Vernier scale for

⁸When frequency is used to directly convey message, it is more appropriate for it being termed as “rate-of-switching”: the number of zero-crossings within a given time window.

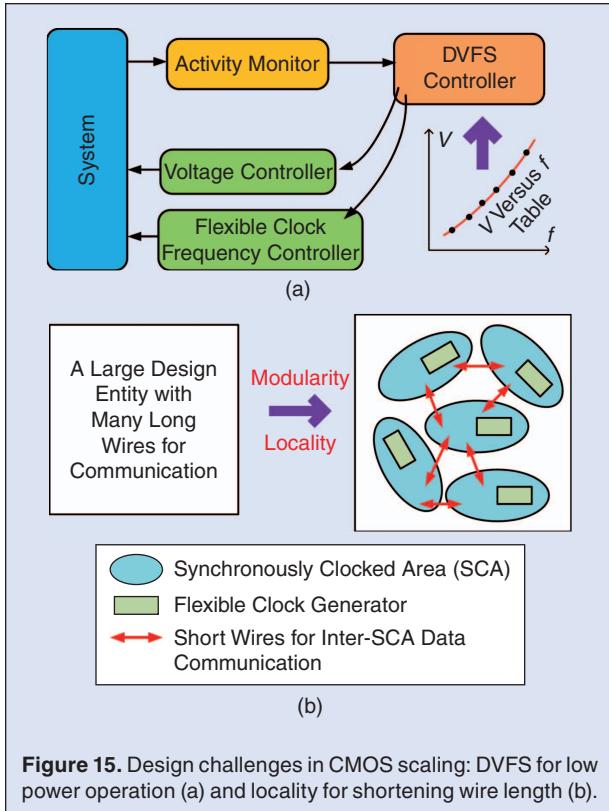


Figure 15. Design challenges in CMOS scaling: DVFS for low power operation (a) and locality for shortening wire length (b).

fine measurement. In the example, the closer v_2 is to v_1 , the finer the measurement can be made (but the absolute values of L_1 and L_2 will be larger). Therefore, a car with flexible speed control is desirable (i.e. we could have whatever speed we want). Similarly, in the world of electrical circuit, a flexible clock source of small frequency granularity is highly preferred and TAF-DPS can meet the challenge⁹.

It is worth mentioning that, when voltage is used as the medium for high precision information sensing, the processing resolution is limited by voltage headroom. Devices' noise floor does not go down with semiconductor process scaling. As a result, when supply voltage is lowered for every new process node, the voltage-based processing resolution will be degraded since the headroom is reduced. On the other hand, for time-based approach, there is no such limitation since there is no end in time.

III.5 “Time” and Clocking in the Trend of CMOS Scaling

During the past several decades, the semiconductor industry has grown in an astonishing pace predicated by the Moore's law. This growth is enabled by the continuous process scaling, which is characterized by the re-

duction of transistor channel length and power supply voltage. One negative impact of supply voltage reduction is the shrunken voltage headroom. This is disadvantageous to voltage based analog signal processing. On the other hand, a key benefit of this scaling is the increased transistor switching speed. This leads to finer resolution in time. Therefore, the continuation of CMOS scaling favors time based signal processing. In other words, *time*¹⁰ is expected to play a more important role as the medium for representing information in future. Time-Average-Frequency based direct period synthesizer can be a useful tool in this emerging game.

The CMOS scaling has increased the gate density (number of transistors packed in a unit area) in many folds during the course of past several decades. This leads to high power consumption and high heat density. Therefore, in modern design, reducing power consumption is a major concern. In many cases, lowering power usage takes priority over chasing performance. One of the useful techniques of reducing power is the dynamical voltage and frequency scaling (DVFS). This is based on the fact of $P \propto V^2 \cdot f$, where P represents the dynamic power. It is seen that lowering f and V can both reduce power consumption. And lowering V has a bigger impact. It is a well-known fact that transistor can operate at lower supply voltage when its operating speed is reduced. This leads to the DVFS scheme as shown in the top of figure 15. A V vs. f table can be established from experiment and stored in the DVFS controller. An activity monitor constantly monitors the system's computation loading. Its output is fed to the DVFS controller. From the loading status, DVFS controller makes decision on voltage and frequency settings. This decision is then passed to the voltage and frequency controllers. As a result, the system could operate at a state of using minimum power. In DVFS scheme, fine resolution in V vs. f table can enhance the DVFS efficiency. In this respect, TAF-DPS based flexible clock generator has advantage over conventional PLL since it can generate much more frequencies.

In the trend of CMOS scaling, wire for interconnect does not scale in the same pace as transistor does. As a result, instead of transistor switching speed, interconnect delay becomes the performance bottleneck. In other words, long wire is problematic in modern IC design. From chip implementation perspective, one solution to this problem is to reduce the wire length. In the bottom of figure 15, a large design entity with many long wires is replaced with a design style of multiple local Synchronously Clocked Areas (SCAs). The size of each SCA is constrained under certain limit so that no significant long

⁹In TAF-DPS Vernier method, the TAF signal needs to be converted into conventional signal by using a 1x PLL.

¹⁰Time in here is used to represent “rate-of-switching”, or number of zero-crossings in a given time window.

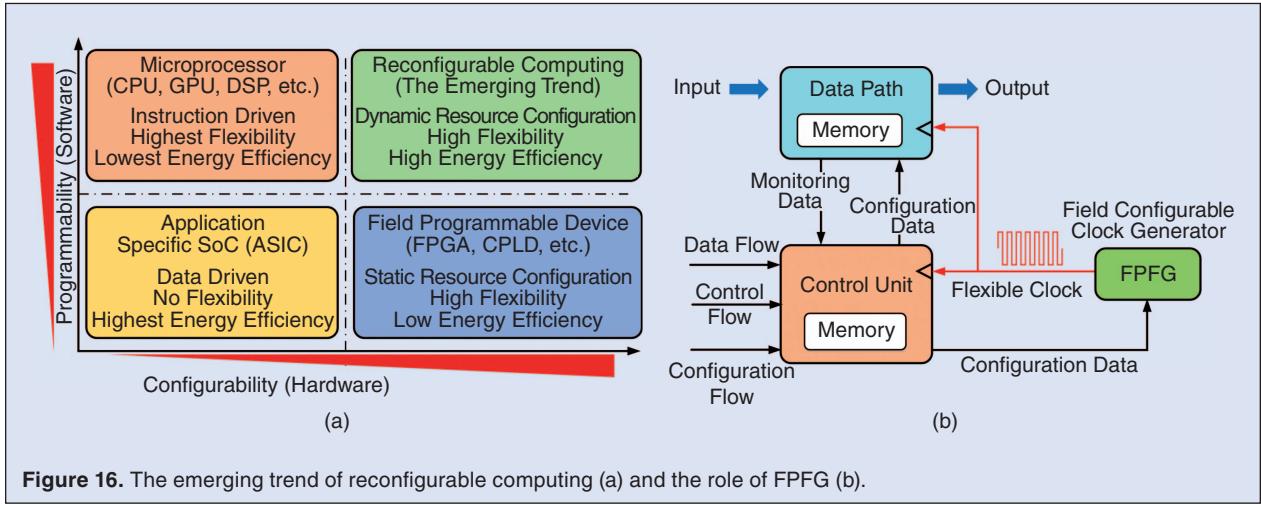


Figure 16. The emerging trend of reconfigurable computing (a) and the role of FPFG (b).

wire is required for data communication. The communications among the SCAs are accomplished through short wires as indicated. This is the modular design philosophy that aligns well with modern VLSI system's characteristic: strong in computation and weak in communication. In this scheme, each LSB is driven by its local clock. To enhance the performance and the communication efficiency of LSBs, the local clocks have to be flexible so that their frequencies and phases can be adjusted as needed. In this front, TAF-DPS based flexible clock generator again has advantage over conventional PLL.

Another side effect of CMOS scaling is that the increasing variability on transistor manufacturability jeopardizes circuit predictability and chip yield. In this environment of larger uncertainty, flexibility incorporated inside circuit design can help ameliorate the situation. Flexibility on clock frequency is an important part of the whole picture of IC design flexibility.

III.6 Flexible Clock and Reconfigurable Computing & Parallel Computing

The driving force for exploring new computer architecture is to improve information processing efficiency (= performance/power), which is measured in MOPS/W. In the left hand side of figure 16, different computing architectures are compared in term of programmability (software) and configurability (hardware) [26]. ASIC style has the lowest programmability and configurability. There is hardly any built-in flexibility. But it has the highest energy efficiency. Unlike ASIC which is usually data driven, microprocessor is instruction driven, it has high programmability (for general purpose computing). On the other hand, because of the generality for software programming, its energy efficiency is low. FPGA is known for its hardware configurability. Similarly, because of the flexibility for hardware configurability, its energy

efficiency is hard to reach high level. It is seen that reconfigurable computing could be the potential winner as the architecture of future. This is evidenced by the recent emergence of HPRC (High Performance Reconfigurable Computing), which combines reconfigurable computing based accelerator (such as FPGA) with CPU or multi-core microprocessor. The increased processing power in a FPGA has enabled larger and more complex algorithms to be programmed into it. The attachment of such FPGA to a CPU over a high speed bus enables the configurable logic to act more like a coprocessor rather than a peripheral.

Reconfigurable computing is an architecture that combines the flexibility of software with the reconfigurable high speed computing fabrics (such as those found in FPGA). In this computer architecture, the main processor controls the behavior of the reconfigurable hardware that are tailored to perform a specific task, such as image processing or pattern matching, as quickly as a dedicated piece of hardware. Once done, the hardware can be reconfigured to do some other tasks. This results in a hybrid computer structure combining the flexibility of software with the speed of hardware. The key difference between reconfigurable computing and conventional microprocessor is its ability to make substantial changes to the data path hardware, in addition to just handling the control flow. On the other hand, its main difference with custom designed hardware (i.e. ASIC or SoC) is the possibility of adjusting hardware during runtime by loading/unloading circuits to/from the reconfigurable fabric [27]–[28].

Unlike an ordinary microprocessor wherein operation is instruction driven, the operation of reconfigurable computing architecture is driven by data flow, control flow and configuration flow as illustrated in the right hand side of figure 16. This fact motivates the idea of incorporating

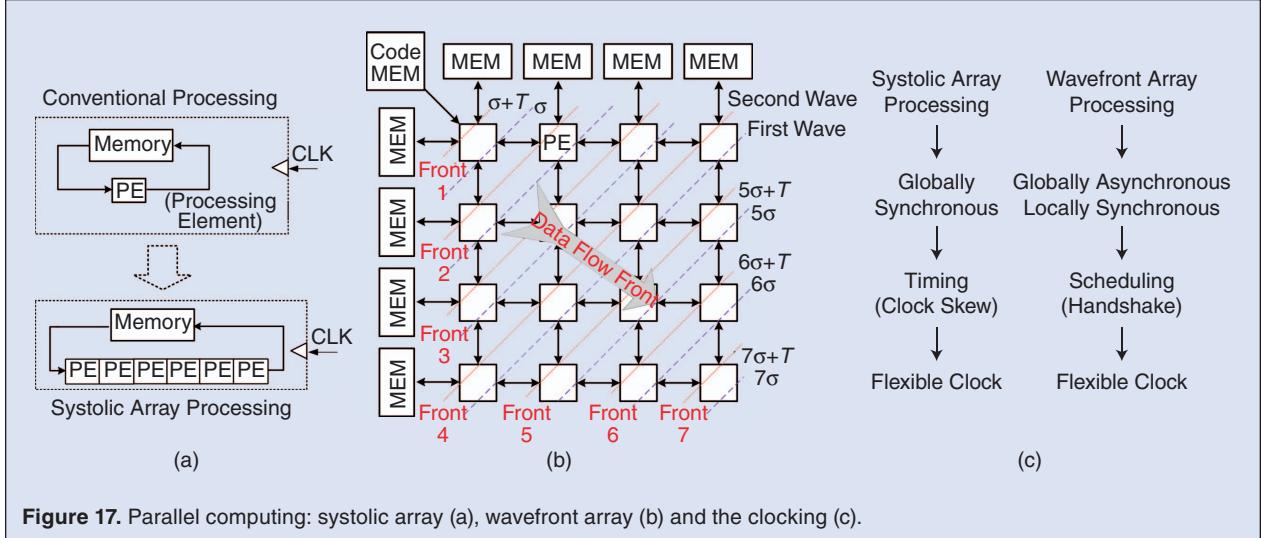


Figure 17. Parallel computing: systolic array (a), wavefront array (b) and the clocking (c).

the clock generator as part of the reconfigurable hardware. TAF-DPS can be designed entirely in digital domain and, in practice, it has been implemented on FPGA using reconfigurable fabric [29]. This makes it a viable candidate for being included in the reconfigurable computing architecture. The benefit of including such FPFG is to *program the driving clock's frequency* on runtime. It is believed that this can further improve the information processing efficiency.

Modern applications requires many specialized algorithms that bear the characteristics of regularity, recursiveness and locality. This fact provides the base for parallel computing. The architecture selection for parallel computing must be studied from both computation and implementation perspectives. It must consider the strength and the weakness of VLSI system: the intensive computing power and the restricted communication capability. More than two decades ago, among others, systolic array processing [30] and wavefront array processing [31]–[32] have been proposed for addressing this issue. As depicted in the left of figure 17, a particular data unit is processed by multiple PEs (processing element) after/before it is released/returned from/to memory. Therefore, the data throughput is increased multiple times in principle. In wavefront approach, data flow is recognized as data wavefront. In the middle of figure 17, data is shown as moving in the fashion of water wave. Each PE receives input from its neighbor and propagates the processed data into the PEs of next level. In this process, the wavefront moving speed is influenced by σ and T , where σ is the time used for data transfer and T is the time for performing arithmetic operation in each PE.

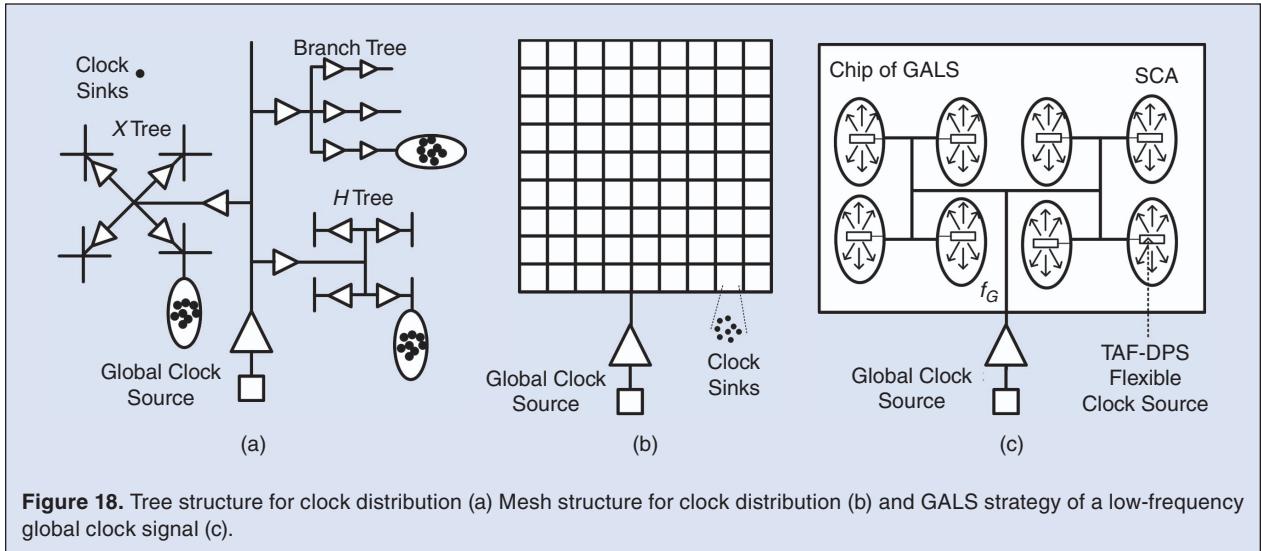
Both the systolic and wavefront architectures take the advantage of localized communication [31]–[33]. But they have significant difference in clocking. As shown in the right hand side of figure 17, systolic uses globally synchro-

nous design methodology. Thus, timing (clock skew) is a big challenge when array size is large. On the other hand, wavefront is Globally Asynchronous Locally Synchronous (GALS). It uses scheduling (based on handshake) to handle the inter-PE communication problem. Flexible clock can assist systolic architecture in addressing the global clock distribution and skew problems (please refer to section III.7). Flexible clock is beneficial to wavefront as well since its frequency flexibility can enhance wavefront architecture's scalability (to dynamically compensate various σ and T values). From a system design perspective, the programmability on frequency provides another dimension for designers to explore and innovate in parallel computing domain.

III.7 Flexible Clock and GALS Strategy

The advance in CMOS technology has led to an exponential increase in chip complexity. The number of transistors in large chips has reached billions. From clocking perspective, chip architecture can be classified as Globally Asynchronous Locally Synchronous (GALS) and Globally Synchronous Locally Synchronous (GSLS). In GSLS approach, a clock signal drives all on-chip modules. This requires the distribution of a global clock signal. There are several design considerations when distributing a clock signal globally: the skew caused by different distribution paths, the jitter accumulated along the distribution path, the silicon & metal resource required for routing the clock signal and the power used by the distribution network. Figure 18 illustrates the clock distribution methods commonly used in chip design practice: tree structure in the left and mesh structure in the middle.

As process technology advances, the tree structure faces difficult challenges. Circuit operating frequency becomes higher due to the reduction in transistor gate



delay. Chip size becomes larger since more transistors are packed in a chip and, as a result, the global clock signal has to travel further. Furthermore, both the gate and the interconnect delay variations induced by PVT change become larger. Moreover, the interconnect delay does not scale well with process advance. All these factors have made skew take larger percentage of the clock period. They also make the variation of skew hard to be controlled. To make it even worse, the delivery of clock signal crossing chip in high frequency requires large amount of metal resource (for shielding) and high consumption of energy (could be as high as $\sim 1/3$ of the total power). The mesh architecture can provide better skew performance than that of tree. However, in most cases, the mesh is over-designed which wastes large amount of resource (power & area).

GALS is a design strategy that aims at ameliorating the situation. Large design is divided into several smaller modules. Each individual module uses synchronous design methodology with a local-global-clock signal as the driving clock. Since the size of each module is usually within a controlled range, the local clock distribution problem is not a severe issue. At chip level, the modules can run asynchronously among each other, which eliminates the global clock distribution problem. The price to pay is the interface circuits for asynchronous communication among the modules (e.g. the handshake mechanism). The wavefront array processor discussed in previous section is a good example of such design philosophy.

TAF-DPS based flexible clock generator could be a useful tool for assisting GALS. As shown in the right hand side of figure 18, a large complex design is divided into several Synchronously Clocked Area (SCA). Each SCA is small enough that can be designed in synchronous style

comfortably. Among SCAs, they run asynchronously. The clock signals for each SCA are generated by TAF-DPSs residing inside each SCA. All the TAF-DPSs share a common reference, which is a low-frequency clock signal globally distributed to each SCA. From a distribution perspective, for two reasons, this low-frequency reference signal does not present problem: 1) it is a low frequency signal (e.g. 100 MHz) and 2) its fan-out loading is limited.

The advantages of this TAF-DPS assisted GALS strategy can be appreciated from the following facts. First of all, the clock frequency of each SCA can be individually controlled (synthesized). Secondly, if desirable, the clock phases of those SCAs can be aligned. Thirdly, the clock phase of each SCA can be individually moved forward or backward for assisting data transfer. These features can be helpful for designing the GALS asynchronous interface circuit (the handshake circuit). Figure 19 shows that two SCAs' clock phases can be aligned after an enable signal. Figure 20 illustrates the fact of moving the clock phase forward or backward under a control signal.

III.8 Flexible Clock and Resonant Clock Distribution & Generation

The most characterizing feature of clock network is its large capacitive loading presented to clock source. During operation, the clock source is responsible for the charge and discharge of this large capacitance. Therefore, an effective way of lowering power usage is to recycle the energy used by the charging and discharging (charge recovery clock distribution). This approach is realized by using the principle of LC resonance in clock distribution. In this idea, inductance is intentionally introduced into chip physical structure. The large capacitance associated with the clock distribution network functions as the C of

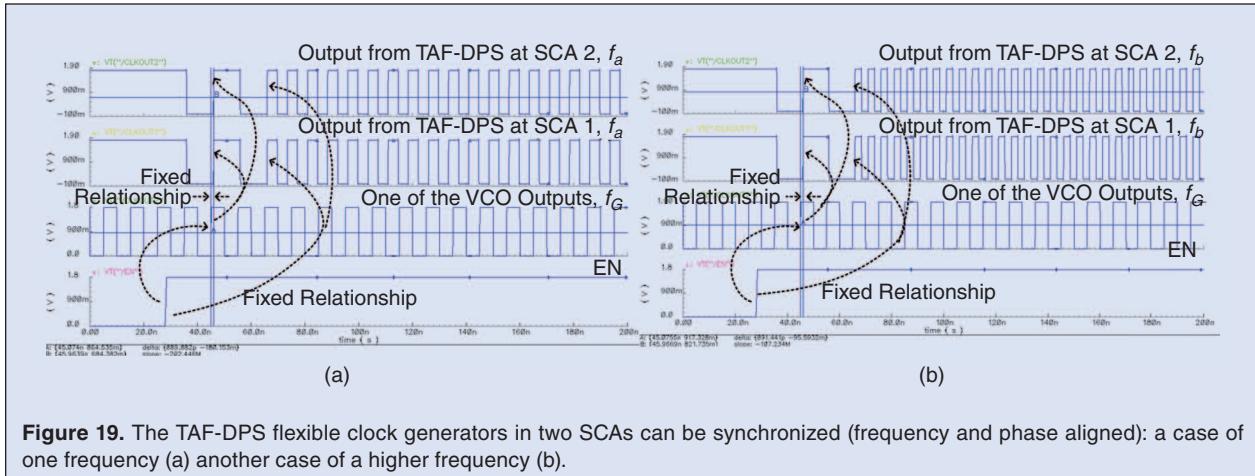


Figure 19. The TAF-DPS flexible clock generators in two SCAs can be synchronized (frequency and phase aligned): a case of one frequency (a) another case of a higher frequency (b).

the LC oscillator. During the charge and discharge process, the energy is stored and released periodically. Ideally, 100% of the energy can be recycled and the electrical oscillation (the clock waveform) can be self-sustained. In practice, due to the parasitic resistance associated with the inductor and the clock sinks, some portion of the energy is lost as the generated heat. Hence, compensation circuitry has to be incorporated on-chip to provide the energy supporting the oscillation. It is expected that the energy used by this approach is much lower than that consumed in conventional methods since, instead of $C \cdot V^2 \cdot f$, the consumed power now is $I^2 \cdot R$ where R is the total parasitic resistance. This power is frequency independent and hence this approach is a good candidate for distributing clock in high frequency (e.g. GHz range).

$$A \cos(\omega t - \beta z) + B \cos(\omega t + \beta z + \varphi) \\ = 2B \cos\left(\omega t + \frac{\varphi}{2}\right) \cos\left(\beta z + \frac{\varphi}{2}\right) + (A - B) \cos(\omega t - \beta z) \quad (1)$$

When two waves of same frequency travel in opposite directions and interact with each other, standing

and traveling waves will be formed. Equation (1) demonstrates the result when two waves of $A \cos(\omega t - \beta z)$ and $B \cos(\omega t + \beta z + \varphi)$ travel in opposite directions within the same transmission line, where t and z are independent variables for time and location, respectively. The resultant first term is the standing wave which has uniform phase in all the locations along the line. Its amplitude is location dependent. The second term is the traveling wave whose phase is location dependent but its amplitude is constant.

Since standing wave has the unique feature of uniform phase, it can be used for clock distribution with potentially zero skew. In the left of figure 21, a clock distribution network is constructed by a grid of metal structure with multiple virtual grounds (VG). The VGs are formed by shorting them together. Between two VGs, a standing wave oscillator (SWO) is formed since the electrical waves are reflected at the VGs. This leads to the interaction of waves, resulting in standing wave while the traveling wave is minimized by the fact that the two waves' amplitudes are almost equal, or $A \approx B$ in (1). A clock signal is injected into the network to initiate and maintain the oscillation. Clock buffers are attached

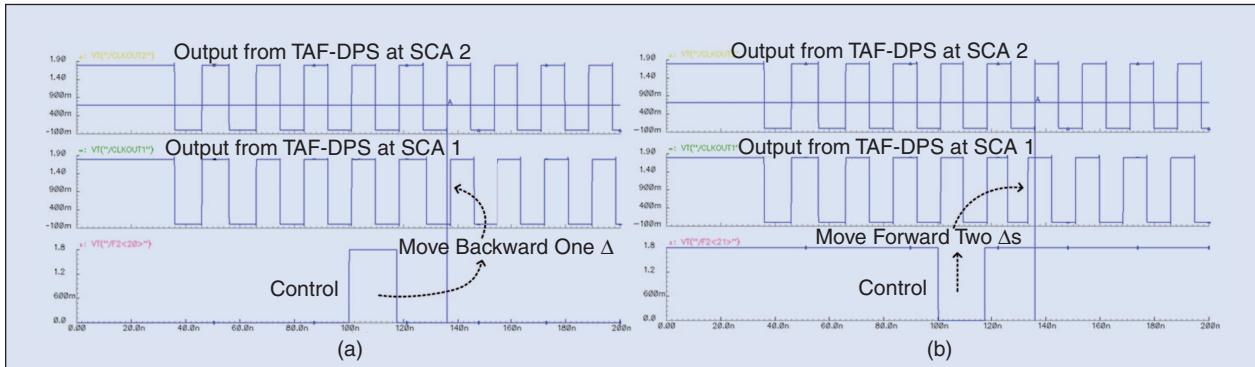


Figure 20. The phase relationship between TAF-DPS flexible clock generators can be adjusted: backward (a) and forward (b).

to the metal structure to form the local clock distribution network [34].

The drawing in the right hand side of figure 21 shows the configuration of using traveling wave to generate clock signal and to distribute clock signal across the chip. The traveling wave is produced by connecting one end of the transmission line to the other so that a loop is formed [35]. As shown, instead of tying the ends of the transmission line to virtual ground, the ends are cross connected (Möbius ring). In this way, no reflection is resulted and the standing wave is non-existing since now $B = 0$ in (1). Multiple antiparallel inverter pairs are inserted to the line to compensate energy lost and maintain rotation lock. The traveling wave has a constant amplitude but different phases along the points on the path. After startup, the wave can travel in either direction (usually the direction of lowest loss) unless it is intentionally biased. This traveling wave can be distributed to the whole chip as the global clock signal by connecting multiple such structures together, as illustrated in the drawing.

The LC resonance clock distribution methods described above have great potential to lower power consumption of the clock network. They all, however, lack the frequency flexibility required by functional circuit's operation. This is due to the fact that the oscillation frequency, for both the standing wave and the traveling wave, is determined by the physical structure of the network (i.e. the natural frequency of the LC resonator). As a flexible clock generator, TAF-DPS can be used to enhance its frequency flexibility in supporting LC resonance clock generation & distribution. For both the standing wave and traveling wave structures shown in figure 21, the ring structures can be used to generate multiple phases. Then, the base time unit Δ is derived from these phases. Subsequently, TAF-DPS clock generator can be built to provide the much needed frequency flexibility for LC resonance clock generation & distribution.

III.9 Flexible Clock for Generating Chirp Signal

A chirp is a signal in which its frequency increases or decreases with time. This type of signal is hard to be created using conventional design techniques. Chirp signal is commonly used in sonar and radar. It is also found in other applications such as spread spectrum communication, resonant converters, electronic ballasts, among others. Linear chirp signal generator is a key component in FMCW (frequency modulated continuous wave) radar for distance and altitude measurement, for vehicle collision warning systems and etc. In FMCW, a carrier frequency is varied linearly at a known rate over a predetermined range by using a chirp signal generator and a FM modulator. Instead of measuring the Time-of-Flight between transmission and reception directly, the change in carrier frequency is measured and divided by the rate of change. This gives the two-way travel time, which subsequently leads to the derivation of the distance/height. The carrier frequency f_c is modulated in a triangular fashion with modulation rate f_m and maximum deviation Δf . At any moment the beat frequency, $f_b = f_t - f_r$ where f_t and f_r are the frequencies of transmitting and receiving waves respectively, is detected using a mixer. The distance/height H can be derived as $H = f_b \cdot C / (4 \cdot \Delta f \cdot f_m)$ where C is the speed of light.

The TAF-DPS is a precise, and low-cost, tool for generating linear chirp signal due to the precisely predictable relationship between its control word and its output frequency. In the top of figure 22, a measured *frequency vs. time* plot from a TAF-DPS chirp signal generator is displayed. In this case, a center frequency $f_c = 60$ MHz is first generated by using a base control word value (e.g. $f = 16$). A triangular modulation pattern of $f_m = 1$ Hz is then produced by changing the output frequency once in every 50 ms. The maximum deviation Δf is chosen as 600 KHz (1% modulation). The change per step is $\Delta f/10 = 60$ KHz. Besides linear chirp signal, sinusoidal and exponential chirp signals are often used in applications as

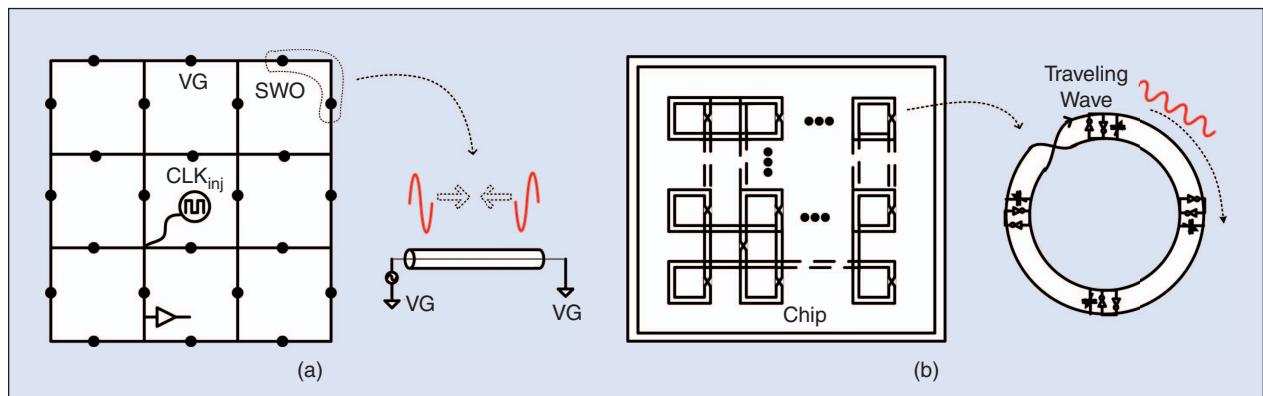
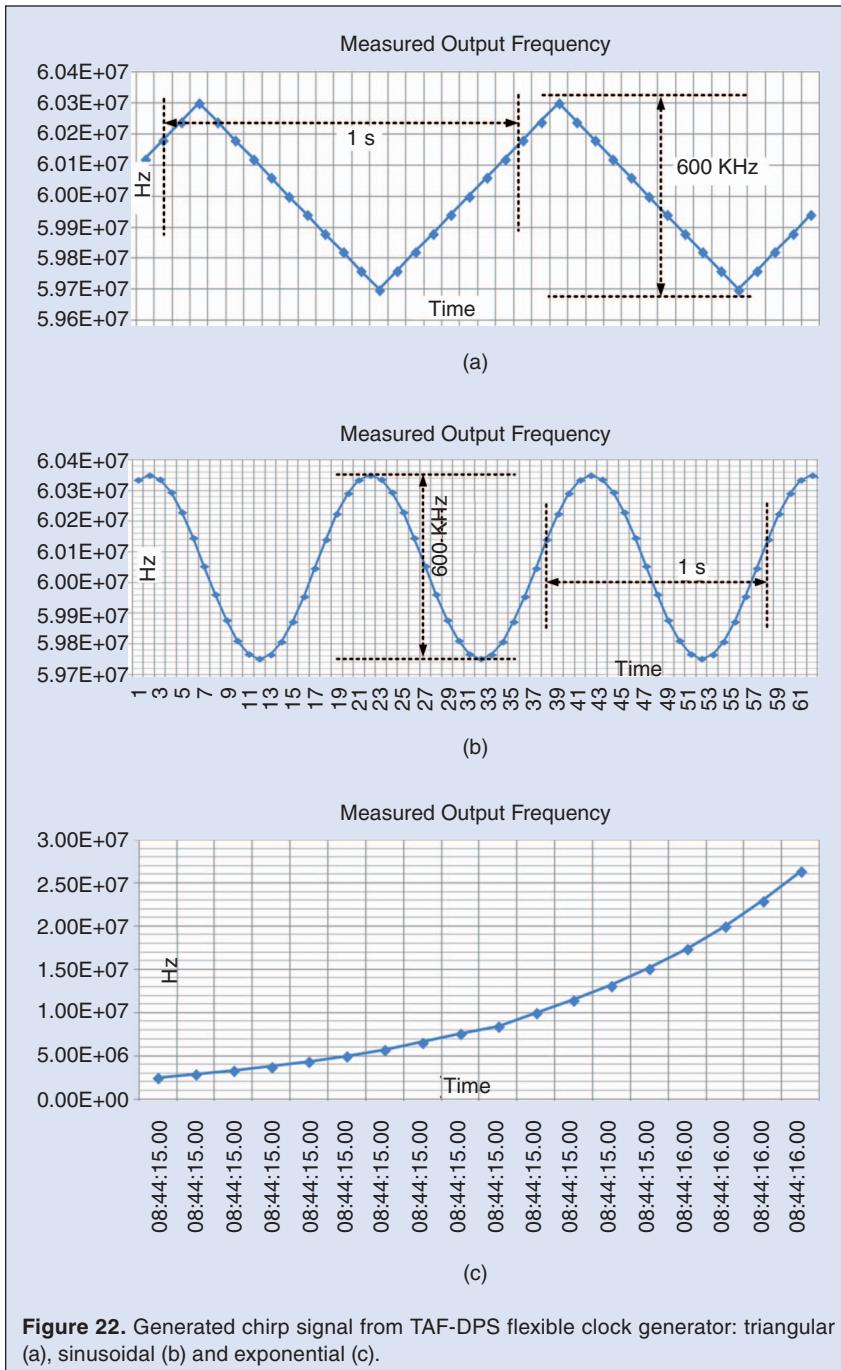


Figure 21. Standing-wave LC clock network (a) Traveling-wave clock network (b).

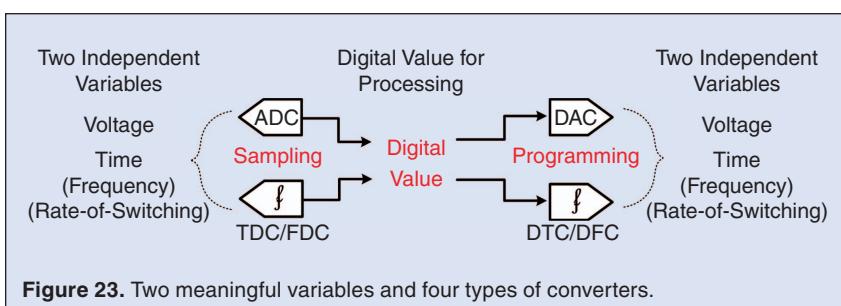


well. Thanks to $T_{\text{TAF}} = 1/f_{\text{TAF}} = F \cdot \Delta$, the TAF-DPS flexible clock generator can produce these signals by simply programming the frequency control word. In the middle of figure 22, measured result of a sinusoidal chirp is displayed. In the bottom of figure 22, an exponential chirp signal of $f = 2.5 \cdot 16^t$ ($t: s, f: \text{MHz}$) is generated, measured and displayed. In this application, “instantaneous frequency switching” makes the chirp signal generator possible.

III.10 Two Independent Variables and Four Types of Converters

As discussed in section I.1 we only need two independent variables, voltage and time, to fully describe any electrical signal. These two values have useful meaning in representing information. When processing information within electronic system, digital value in binary format is preferred for performing logical and arithmetic operations. Therefore, there is a need to convert those two variables to digital value, back and forth as shown in figure 23. When voltage is concerned, circuit techniques for ADC (analog-to-digital converter) and DAC (digital-to-analog converter) are already mature. Historically, they have played important role in the making of electronic system. On the time side, the TDC (time-to-digital converter) has also been well studied (the Vernier method discussed in section III.4 is an example). DTC (digital-to-time converter) can be simply constructed by using a counter. Vernier method can be used as well if better time resolution is required.

Often times in signal processing *time* is not directly used but its derivative, *frequency* or *rate-of-switching*, is used instead. Thus, we also need FDC (frequency-to-digital converter) and DFC (digital-to-frequency converter). FDC is rather



simple, it is simply a counter in principle. DFC, however, is a hard-to-make component since it is virtually a frequency synthesizer. The essential requirements here are *amply frequencies* and *fast frequency switching*. The TAF-DPS discussed in section II.2 is an ideal tool for making such DFC while PLL is not qualified for this task. As illustrated in figure 23, the ADC, TDC/FDC are used to perform the task of sampling. They convert the voltage, time and frequency/rate-of-switching into digital value. On the other hand, DAC, DTC/DFC carry out the task of programming. The programming on their input digital values will be reflected as designed-change-pattern on their outputs: the voltage, the time or the frequency/rate-of-switching.

III.11 Frequency: A Programmable Variable

Enabling “Rate-of-Switching” Approach

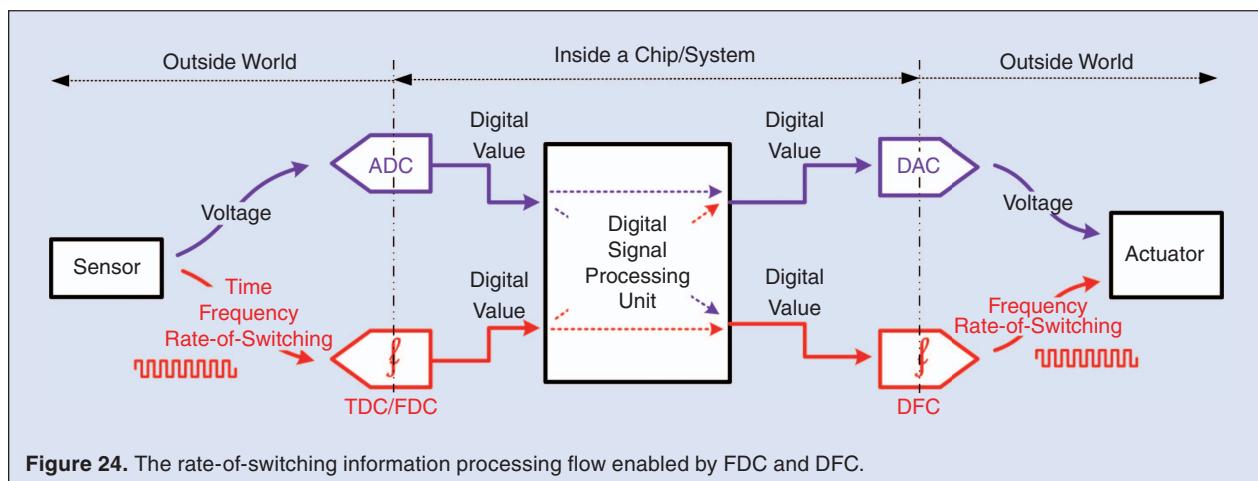
When frequency/rate-of-switching becomes a programmable variable, it opens a possibility for a new rate-of-switching based information processing flow. In this approach it is the *time elapse or number of switching activities occurred within a fixed time window*, not the voltage level, that represents the interested information. As depicted in figure 24, voltage level is used in the conventional processing method (the upper path). In this method, ADC and DAC plays the role of interfacing the chip/system with the outside world. For the new approach, FDC and DFC replaces them as the interface tools. For certain application, this approach might bear better processing efficiency. For instance, in the widely used charge pump PLL case, the PFD (phase frequency detector) is a TDC which converts the time difference between the edges of the input and reference pulses into rate-of-switching. This message is then fed into charge pump and be translated into voltage level to control the VCO (voltage controlled oscillator) output frequency. In here, the charge pump can be viewed as the rate-of-switching based actuator. In

this PLL example, the information is represented inside the system through rate-of-switching instead of voltage, and it serves its purpose better than voltage in this case.

Conclusion: It is Time for Time to Shine

Time is a major subject of religion, philosophy, and science. Among great thinkers, there are two distinct standpoints on time. One view is that time is part of the fundamental structure of the universe, a dimension in which events occur in sequence. The opposing view is that time does not refer to any kind of physical container that events and objects move through. Instead, time is part of a fundamental intellectual structure (made of space, number, and time) within which humans sequence and compare events. In this second view, time is a virtual subject, neither an event nor a thing, and thus is not itself measurable.

The chase for ever-increasing better control on time is one of the driving forces that advances our civilization. From a historical perspective, the previous three industrial revolutions all started with the pursuit of better understanding and control on time. For example, the “Theory of Relativity” is originated from the questioning on *simultaneity*. In the current 4th revolution, the same is true. But the battle field no longer is in our daily life but instead is inside electronic world. And furthermore, the focus is not on finer resolution but on control flexibility. For current information-central revolution, the key underlying support comes from the semiconductor industry that relies on creating various functional circuits from transistors. This field of IC design has seen major breakthroughs in many areas, such as processor architecture, data storing technology, analog circuit design practice and theory, computer algorithm, RF communication scheme, EDA tool development, among others. Additionally, in recent effort of “beyond Moore” and “More Than Moore”, novel material design concepts, new transport mechanisms, extreme electrostatics (such as



2D materials, quantum/spin devices, bi-layer graphene) have been actively explored. However, in-depth study of *time*, one of the two fundamental variables in describing signal, is still more or less a no-man's land.

Because of the importance of *time* in establishing both universal and local orders, great benefit can be gained by a more sophisticated treatment of *flow-of-time*. The idea of *flexible clock* emerges spontaneously from the fog of complexity characteristics of advanced chip design. In this direction, toward a higher application level, it leads to a new frontier for exploration. It is time for *time* to shine. Our ultimate goal is to lay a new cornerstone for the current information-age revolution.



Liming Xiu (M'95–SM'05) earned B.S. and M.S. degrees in physics from Tsinghua University, China, in 1986 and 1988, respectively. He earned his MEEE from Texas A&M University, USA, in 1995. In 1995–2009, he worked for Texas Instruments, Dallas, USA (Senior Member Technical Staff). In 2009–2012, he was chief clock architect of Novatek Microelectronics, Taiwan. From 2012–2015, he was VP of Kairos Microsystems, Dallas, USA. From 2015, he works for BOE Technology Group, Beijing, China, as chief scientist on IC technology. He served as VP of IEEE CASS in years 2009–2010. He is the inventor of Flying-Adder frequency synthesis architecture; the promoter of Time-Average-Frequency concept and theory. He has 24 granted US patents. He has published numerous IEEE journal and conference papers, published three books as solo author: «*VLSI Circuit Design Methodology Demystified*», «*Nanometer Frequency Synthesis beyond Phase-Locked Loop*» and «*From Frequency to Time-Average-Frequency: A Paradigm Shift in the Design of Electronic system*». He is also an invited author of a chapter in book «*Mixed-Signal Circuits*».

References

- [1] J. Popiel-Gorski, *Frequency Synthesis: Techniques and Applications*. Piscataway, NJ: IEEE Press, 1975.
- [2] V. F. Kroupa, *Direct Digital Frequency Synthesis*. Piscataway, NJ: IEEE Press, 1998.
- [3] F. M. Gardner, *Phaselock Techniques*, 3rd ed. New York: Wiley-Interscience, 2005.
- [4] W. F. Egan, *Phase-Lock Basic*, 2nd ed. New York: Wiley-IEEE Press, 2007.
- [5] G. A. Leonov, N. V. Kuznetsov, M. V. Yuldashev, and R. V. Yuldashev, "Hold-In, Pull-In, and Lock-In Ranges of PLL circuits: Rigorous mathematical definitions and limitations of classical theory," *IEEE Trans. Circuits Syst. I*, vol. 62, no. 10, pp. 2454–2464, 2015.
- [6] E. Ali, C. Hangmann, C. Hedayat, F. Haddad, W. Rahajandraibe, and U. Hilleringmann, "Event driven modeling and characterization of the second order voltage switched charge pump PLL," *IEEE Trans. Circuits Syst. I*, vol. 63, no. 3, pp. 347–358, 2016.
- [7] L. Xiu, "The concept of time-average-frequency and mathematical analysis of flying-adder frequency synthesis architecture," *IEEE Circuit Syst. Mag.*, pp. 27–51, Sept. 2008.
- [8] L. Xiu, *Nanometer Frequency Synthesis Beyond Phase Locked Loop*. New York: Wiley-IEEE Press, 2012.
- [9] L. Xiu, *From Frequency to Time-Average-Frequency: A Paradigm Shift in the Design of Electronic System*. New York: Wiley-IEEE Press, 2015.
- [10] H. Mair and L. Xiu, "An architecture of high-performance frequency and phase synthesis," *J. Solid-State Circuits*, vol. 35, pp. 835–846, June 2000.
- [11] L. Xiu and Z. You, "A flying-adder frequency synthesis architecture of reducing VCO stages," *IEEE Trans. VLSI*, vol. 13, pp. 201–210, Feb., 2005.
- [12] L. Xiu, W. T. Lin, and K. Lee, "A flying-adder fractional-divider based integer-N PLL: The 2nd generation Flying-Adder PLL as clock generator for SoC," *J. Solid-State Circuits*, vol. 48, pp. 441–455, Feb. 2013.
- [13] L. Xiu, K.-H. Lin, and M. Ling, "The impact of input-mismatch on Flying-Adder direct period synthesizer," *IEEE Trans. Circuit Syst. I*, vol. 59, pp. 1942–1951, Sept. 2012.
- [14] L. Xiu, "Direct period synthesis for achieving sub-ppm frequency resolution through time average frequency: the principle, the experimental demonstration, and its application in digital communication," *IEEE Trans. VLSI*, vol. 23, no. 7, pp. 1335–1344, 2015.
- [15] Texas Instruments Inc. (2015). TMS320DM816x DaVinci Digital Media Processors Technical Reference Manual. [Online]. Available: <http://www.ti.com/lit/ug/sprugx8c/sprugx8c.pdf>
- [16] P. Sotiriadis, "Theory of Flying-Adder frequency synthesizers, Part I: Modeling, signals' periods and output average frequency," *IEEE Trans. Circuits Syst. I*, vol. 57, pp. 1935–1948, Aug. 2010.
- [17] P. Sotiriadis, "Theory of Flying-Adder frequency synthesizers, Part II: Time and frequency domain properties of the output signal," *IEEE Trans. Circuits Syst. I*, vol. 57, pp. 1949–1963, Aug. 2010.
- [18] P. Sotiriadis, "Exact spectrum and time-domain output of Flying-Adder frequency synthesizers," *IEEE Trans. Ultrasonics, Ferroelectrics, Freq. Control*, vol. 57, pp. 1926–1935, Sept. 2010.
- [19] S. A. Talwalkar, "Quantization error spectra structure of a DTC synthesizer via the DFT axis scaling property," *IEEE Trans. Circuit Syst. I*, vol. 59, pp. 1242–1250, June 2012.
- [20] S. A. Talwalkar, "Digital-to-time synthesizers: Separating delay line error spurs and quantization error spurs," *IEEE Trans. Circuit Syst. I*, vol. 60, pp. 2597–2605, Oct. 2013.
- [21] P. Sotiriadis, "Spurs-free single-bit-output all-digital frequency synthesizers with forward and feedback spurs and noise cancelation," *IEEE Trans. Circuits Syst. I*, 2016.
- [22] F. Yang and T. Haider, "MPEG-2 transport stream packet synchronizer," U.S. Patent 8 249 171.
- [23] J. L. Ferrant, et al. *Synchronous Ethernet and IEEE 1588 in Telecoms: Next Generation Synchronization Networks*. New York: Wiley-ISTE, 2013.
- [24] M. Horauer, "Clock synchronization in distributed systems," Ph.D. dissertation, Vienna Univ. Technol., 2004.
- [25] L. Xiu, "A novel DCXO module for clock synchronization in MPEG2 transport system," *IEEE Trans. Circuit Syst. I*, vol. 55, pp. 2226–2237, Sept. 2008.
- [26] S. J. Wei, "Investigation on several fundamental issues regarding integrated circuit design," presented at the Microelectronics and Optoelectronic Technology & Science Forum, China Academy of Science, Shanghai, Nov. 3, 2015.
- [27] C. Bobda, *Introduction to Reconfigurable Computing: Architectures*. New York: Springer, 2007.
- [28] J. Cardoso, *Reconfigurable Computing: from FPGAs to Hardware/Software Codesign*. New York: Springer, 2011.
- [29] L. Xiu and P. L. Chen, "A TAF-DPS frequency synthesizer on FPGA achieving 2 ppb granularity and two-cycle switching speed," *IEEE Trans. on Industrial Electronics*, vol. 64, pp. 1233–1240, Feb. 2017.
- [30] H. T. Kung, "Why systolic architectures?," *Computer*, vol. 15, no. 1, pp. 37–46, 1982.
- [31] S. Y. Kung, K. S. Arun, R. J. Gal-Ezer, and D. V. Bhaskar Rao, "Wavefront array processor: Language, architecture, and applications," *IEEE Trans. Computers*, vol. C-31, no. 11, pp. 1054–1066, 1982.
- [32] S. Y. Kung, S. C. Lo, S. N. Jean, and J. N. Hwang, "Wavefront array processors: Concept to implementation," *Computer*, vol. 20, no. 7, pp. 18–33, 1987.
- [33] G. Causapruno, F. Riente, G. Turvani, M. Vacca, M. Ruo Roch, M. Zamboni, M. Graziano, "Reconfigurable systolic array: From architecture to physical design for NML," *IEEE Trans. Very Large Scale Integrat. Syst.*, 2016.
- [34] F. O'Mahony, C. P. Yue, M. A. Horowitz, and S. S. Wong, "A 10-GHz global clock distribution using coupled standing-wave oscillators," *J. Solid-State Circuits*, vol. 38, no. 11, pp. 1813–1820, Nov. 2003.
- [35] J. Wood, T. C. Edwards, and S. Lipa, "Rotary traveling-wave oscillator arrays: a new clock technology," *J. Solid-State Circuit*, vol. 36, no. 11, pp. 1654–1665, Nov. 2001.