Asynchronous Event-Driven Clocking and Control in Pipelined ADCs

Benjamin Hershberg, *Member, IEEE*, Barend van Liempd, *Member, IEEE*, Nereo Markulić, *Member, IEEE*, Jorge Lagos, *Member, IEEE*, Ewout Martens, *Member, IEEE*, Davide Dermit, *Member, IEEE*, and Jan Craninckx, *Fellow, IEEE*

***Abstract*— An asynchronous event-driven approach to clocking and timing control is explored in the context of pipelined ADCs. It is shown how a conventional global clock tree can be replaced by localized control units coordinated through inter-stage communication protocols. The approach is found to yield many compelling advantages in terms of power efficiency, speed, robustness, and reconfigurability. It is shown how these benefits are particularly well leveraged when used in combination with dynamic-power residue amplifiers such as ring amplifiers. Several challenges also arise: re-synchronization of the digital outputs, mitigation of possible deadlock scenarios, and robust timing control configuration. Solutions to these problems are presented. Two single-channel 11-bit 1.5-bit/stage pipelined ADC designs are fabricated in a 16nm CMOS technology, each with a different implementation approach to the asynchronous control units. The trade-offs of both approaches are considered. At 1 GS/s the fastest prototype achieves 59.5 dB SNDR and 75.9 dB SFDR at Nyquist, consuming 10.9 mW including reference regulator. Due to fully-dynamic operation, it maintains a near-constant Walden Figure of Merit (FoM) of 14 fJ/conversion-step from 1 MS/s to 1 GS/s.**

***Index Terms*— ADC, A/D, asynchronous, clock tree, dynamic, event-driven, high speed, low power, pipeline, pipelined ADC, deep pipeline, ringamp, ring amplifier.**

I. Introduction

TIMING control in a pipelined ADC is traditionally implemented with a synchronous hierarchical clock tree. The clocking system illustrated in Fig. 1 represents the minimum set of timing signals required for the operation of a typical SHA-less pipelined ADC. At the global level, a non-overlapping clock generator takes the *master clock* as an input and synthesizes two non-overlapping clock phases, *clk1* and *clk2*, which are distributed to all stages in the pipeline. At the stage level, a local clock generator converts these two clocks into the signals needed for timing control within the stage. It also serves as a buffer, isolating the global clock network from the load of the stage circuitry. Each stage requires at minimum a *track*, *track delay*, *quantize*, and *amplify* control signal. In practice, pipelined ADC implementations are often more complex than this, including additional control signals and/or layers of distribution and buffering.

The practical implementation of this approach results in a highly distributed network, with parasitics and mismatch creating skew between the different branches of the clock tree. In the presence of these non-idealities, sufficient margin in the form of extra time delay between phases of operation must be incorporated into the clock tree design such that all non-overlap and causal relationships are maintained across process, voltage, and temperature (PVT) variation. The overhead associated with this becomes increasingly costly as clock frequency is increased, since many mismatch and skew related delays do not scale proportional to clock period [1]. In high speed pipelined ADCs, this leads to a difficult set of related design trade-offs in terms of power, speed, jitter, and reliability.

Meanwhile, advances in scalable amplifier technology such as ring amplification [2] have led to considerable improvements in residue amplifier power efficiency, such that in some cases clock power exceeds amplifier power in pipelined ADCs [3], [4]. This suggests that total system efficiency will become increasingly dependent on clocking and control optimization in the future.

It is already well known that asynchronous event-driven clocking techniques can provide speed and power advantages in SAR and pipelined-SAR ADCs [5]–[9]. For SAR ADCs in the hundreds of MHz and even low GHz, this approach becomes particularly attractive, providing a solution to a fundamental problem. SAR ADCs require many sub-operations to complete a conversion, and if clocked synchronously, require a master clock running much higher than the sampling rate. For example, a 10 bit 1GS/s SAR ADC with 5 sampling sub-periods, 10 conversion sub-periods, and 1 reset sub-period requires a 16 GHz master clock to be synthesized and distributed, which is not trivial or energy efficient to do. Furthermore, the minimum allowable DAC settling time and comparator decision delay often varies from MSB to LSB in a SAR, and when constrained by equally spaced synchronous phases, will result in sub-optimal timing utilization and reduce the maximum achievable sampling rate. A self-timed asynchronous event-driven control scheme can solve these problems, allowing for a master clock frequency equal to the sampling

Manuscript received February 21, 2021; revised April 11, 2021; accepted April 28, 2021. Date of publication May 27, 2021; date of current version June 4, 2021. This article was recommended by Associate Editor R. Rieger. *(Corresponding author: Benjamin Hershberg.)*

The authors are with the Interuniversity Microelectronics Center (IMEC), 3001 Leuven, Belgium (e-mail: benjamin@hershberg.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSI.2021.3077881.

Digital Object Identifier 10.1109/TCSI.2021.3077881

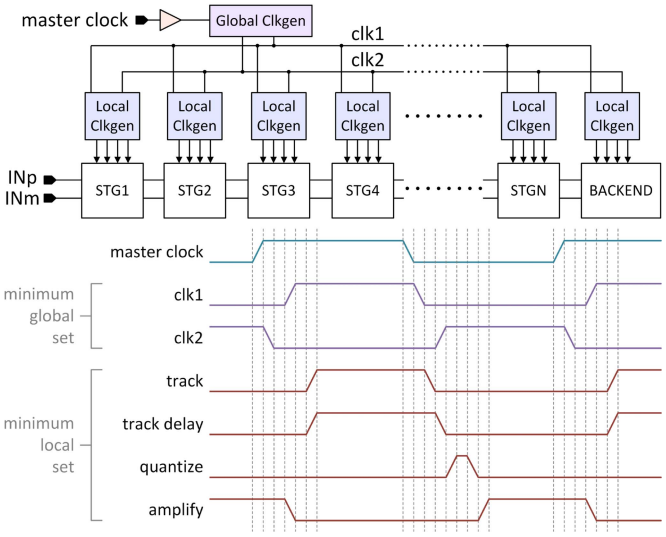


Fig. 1. The minimum set of signals required for synchronous hierarchical timing control in a conventional SHA-less "deep" pipelined ADC.

rate (e.g., 1GHz in the example above) and better utilization of the available timing budget. In pipelined SAR ADCs, usually consisting of two or three stages, the advantages become even more compelling, as it also allows for the independent partitioning of the SAR quantizer and residue amplifier timing budgets [8], [10], [11].

And yet, in "deep" pipelined ADCs that consist of many stages, the asynchronous event-driven approach remains largely unexplored at the system level. A few examples do exist for asynchronously triggered sensor applications running at low speed [12], [13], but these highly specialized architectures come with major speed penalties that make them unsuitable for most applications. The most relevant prior work is found in pipelined SARs where key elements of the approach have been utilized [10], but there is still much unexplored from the perspective of deep pipelines. Unlike the clear limitations that immediately confront a designer attempting synchronous clocking of a SAR ADC, the two-phase clocking scheme of Fig. 1 appears deceptively simple by comparison, and the justification for an asynchronous approach is less obvious. However, this intuition is often misleading, and obscures a surprising number of opportunities for innovation in deep pipelined ADCs.

In this paper we will explore such a scheme, where a hierarchical clock tree is replaced by localized control units that interact with each other using event-driven triggers and handshake protocols. The result is a modular, correct-by-construction timing system that minimizes routing complexity and maximizes performance. The numerous advantages of the approach are combined with the efficiency of ring amplifiers to build two single-channel 11-bit pipelined ADCs, each with a different approach to timing control implementation. The faster of the two prototypes achieves 59.5 dB SNDR and 75.9 dB SFDR at 1 GS/s with fully dynamic power consumption that maintains 14 fJ/conversion-step Walden FoM from 1 MS/s to 1 GS/s.

Section II introduces a pipelined ADC architecture that implements asynchronous event-driven timing control, and

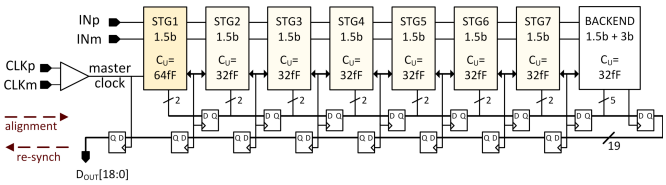


Fig. 2. Top-level architecture of the pipelined ADC used to demonstrate asynchronous event-driven timing control.

explains how this approach expands design freedom at both block and system levels. Section III dives deeper into understanding additional advantages of the approach. Section IV considers challenges and drawbacks of the approach, with proposed solutions. Section V digs into practical details of the timing control logic, considering the pros and cons of different implementation strategies. Relevant measurements of the two related pipelined ADCs, each with a different timing control implementation, are presented in Section VI. Finally, in Section VII we draw conclusions and discuss future work.

II. Illustrative Architecture

We will begin our exploration of the topic with a specific implementation example, and show how event-driven control is incorporated into all levels of this particular system, including sub-blocks and circuits. Note that this is just one architecture among many possibilities, but by starting with a practical implementation example first, it provides a reference for the more generalized and abstracted discussions that follow later.

A. ADC Overview

The deep pipeline architecture to be considered is shown in Fig. 2. It is an 11-bit ADC composed of seven 1.5-bit stages followed by the 1.5-bit + 3-bit backend stage described in [14]. Stage scaling is performed only once in this pipeline: the capacitors and circuity in stage 2 are downscaled by a factor of two relative to stage 1. Stage 2 is then duplicated for stages 3 to 7. This simple architecture is chosen to minimize design complexity at the expense of some power efficiency that could be achieved with a more careful backend optimization. The SHA-less first stage implements a passive-hold MDAC technique [14], and it is for this reason that the sub-ADC of stage 1 is connected to the amplifier summing nodes in Fig. 3.

B. Timing Control Blocks

The global clock generator and distribution network of Fig. 1 is entirely eliminated in this design. Rather, the clock buffer drives a single clock line, the *master clock*, directly into the first pipeline stage. This *master clock* triggers the self-timed chain reaction of processing events inside the pipeline that converts an analog input into a digital output. Unlike a synchronous system that often requires a specific duty cycle, usually 50%, the duty cycle of the *master clock* is not important because only one edge of the *master clock* initiates the chain reaction. This helps to simplify the input clock buffer design.

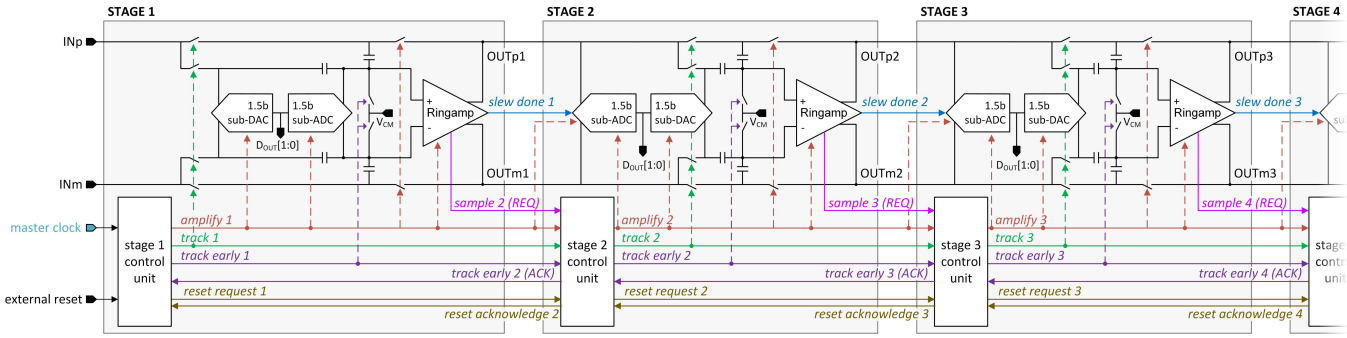


Fig. 3. Simplified schematic of the front pipeline stages. Inter-stage buses connect the "control unit" state machines.

As shown in Fig. 3, each stage is equipped with a control unit containing a state machine (whose implementation is discussed in Section V). Each control unit is connected to the control units of adjacent stages through inter-stage buses. These buses are used for the dual-purpose of controlling the core pipeline circuitry and communicating with other control blocks.

Trigger signals are not always dispatched directly from the control unit. Sometimes a triggering event originates within an analog block, and in those cases it is often more optimal to pass communication directly from the analog block to a receiving control unit, minimizing latency. For example, the *sample REQ* and *slew done* events generated by the ringamp sub-system in Fig. 3 follow this approach.

C. Ring Amplifier

As we will see in Section III, many of the advantages of an asynchronous event-driven approach are enhanced by a residue amplifier capable of dynamic operation. It should be able to rapidly switch on and off, and when off, not consume static power. Many state-of-the-art designs in recent years eschew conventional amplifiers for techniques that fall into this category, including Gm-C integrators [8]–[10], Gm-R amplifiers [11], and ring amplifiers [2]–[4], [14], [15], including emerging ringamp topologies such as self-quenching "floating inverter" structures [16]–[18].

In the case of a ring amplifier, detecting the moment during residue amplification when slewing finishes and settling begins is often useful as an event trigger in timing control. Two examples are: using it to implement the early quantization technique presented later in Section II-D, and using it to switch between regulated voltage replica lines in the "split-reference" dynamic reference regulation technique of [19].

Fig. 4 presents a simple and effective method for detecting this "slew done" event in a ringamp. Consider the binary value of the voltages at nodes A, B, C, and D. During slewing, the logic equation AB'CD' will never equal "1", because A=B and C=D. During settling, the dynamic formation of a dead-zone forces the four nodes apart such that A=C="1" and B=D="0" and the logic equation evaluates to "1". Based on this principle, a simple detection circuit in the "Event-Trigger Generation" block of Fig. 4a generates the *slew done* signal when D_{MODE}="1". The logic function output is fed into a gated latch, which captures and holds the first detection edge. To avoid a false-positive detection during startup transients, the latch enable can be slightly delayed by t_{d1}. Alternatively, when D_{MODE}="0" the sensor is bypassed and *slew done* is set directly with digitally tunable delay t_{d1}.

Maximization of both speed and scalability in ringamps revolves around the principle of placing the internal poles at the highest frequency possible. This typically requires a philosophy of minimalism: never adding capacitance or limiting current flow of the inner stages unless absolutely necessary. Although the *slew done* detector violates this cardinal rule, a careful and compact layout of the NAND gates that attach to nodes A, B, C, and D can result in an acceptable performance loss that is more than compensated for by the time won by eliminating quantization from the critical timing path, by using the early quantization technique described in the next subsection.

Both the analog-detection and static-delay options chosen by D_{MODE} for "slew done" event triggering, and even the idea of a "slew done" event at all, are restricted to a subset of residue amplifier topologies. For some dynamic amplifier topologies such as Gm-R, the digitally tunable static delay option is still viable, but the analog detector option cannot be used. In other topologies that continuously integrate, such as Gm-C, the notion of a "slew done" moment doesn't even exist, and requires different architectural choices altogether. For amplifier topologies where certain internal nodes undergo voltage collapse during settling, such as self-quenching ringamps or charge-steering amplifiers [20], [21], a number of analog-based detection schemes that are different from the circuit in Fig. 4a become possible. Ultimately, it is important to keep in mind that the architecture and implementation presented in this section is just one instructive example inside a larger set of asynchronous event-driven pipelined ADC possibilities.

D. Early Quantization

The instant the ADC samples its input onto a capacitor, all the information it will ever digitize about that sample is already available in analog form. For the first stage of a SHA-less pipeline, this observation is of little practical use due to causality constraints: the order of operations must still be *track*, *quantize*, and *amplify*.[1] But for all subsequent

[1]Some exceptions have been proposed for certain niche applications [22].

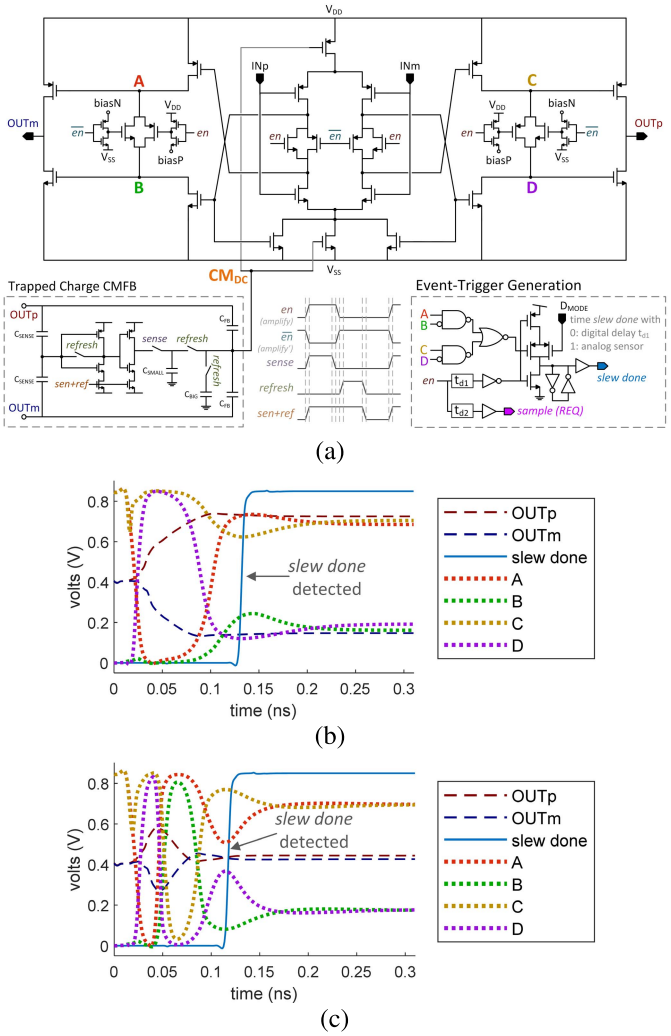


Fig. 4. In (a) the slew done detector is shown attached to the ringamp of [14]. Transient operation is depicted for (b) a small output voltage and (c) a large output.

stages, this insight suggests the possibility of removing the quantization phase from the critical timing path. The two main approaches to this are "look-ahead quantization" techniques that combine over-quantization with sub-ranging to predict future residues [23], [24] and "early quantization" techniques based on early sampling of the partially settled residue of the previous stage [25], [26]. Although both approaches have their merits, early quantization tends to win in a full system-level analysis provided that one critical issue is addressed, which we will now discuss.

The principle of early quantization can be understood by considering the ringamp charging waveforms of Fig. 4b and 4c: shortly after slewing, the output residue accuracy is already within the error bounds of the next pipeline stage's sub-ADC redundancy. Thus, it is not necessary to wait until the end of residue amplification to quantize; the sub-ADC can sample early and process the partially settled residue in parallel with continued fine settling of the main MDAC path. The key challenge of this technique is determining when the sub-ADC should sample. Sampling too soon will result in error exceeding the redundancy limits, and sampling too late will

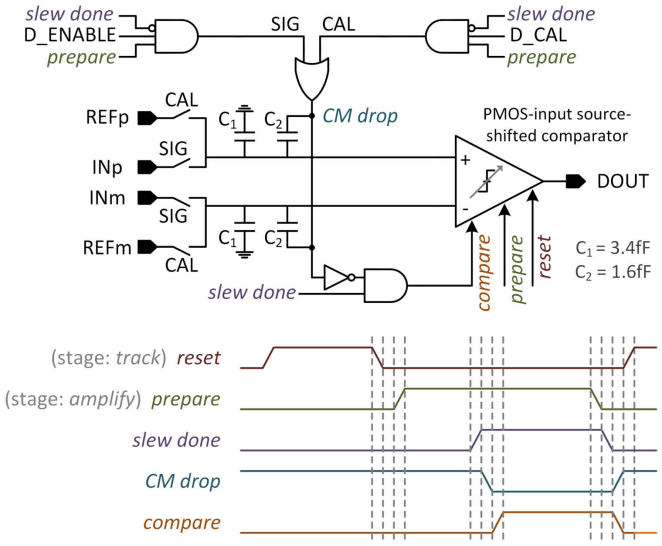


Fig. 5. Self-timed early-quantization sub-ADC used in pipeline stages 2 to 7.

impede optimal settling due to disturbances in the amplifier output caused by kickback from the sub-ADC sampler. The ideal moment to sample is therefore immediately after slewing. The ringamp's *slew done* detector provides an event-driven solution to this.

E. Self-Timed Sub-ADC

Asynchronous control at the system level necessitates that timing inside of lower level blocks also follow an event-driven approach. One example is the 1-bit quantizer shown in Fig. 5, which operates based on a chain-reaction of self-timed events. The signals *slew done*, *prepare*, and *reset* are derived from external stage-level controls. Internally, the triggering of *slew done* initiates a local chain-reaction that samples (*SIG*="1"), sets the desired common-mode level (*CM drop*="0"), and latches the comparator (*compare*="1"). This example illustrates how causality leads to optimality: despite single-gate-delay spacings between many of these operations, the correct order of operations is still guaranteed (correct-by-construction). The comparator of Fig. 5 is the wide-range tunable threshold topology introduced in [27]. Due to the versatility of this comparator (high efficiency, fast decision speed, very wide tuning range, and minimum input capacitance) it is used for all sub-ADCs throughout the pipeline, including the 3-bit backend flash.

F. Timing Example

Fig. 6 shows a simplified version of the event-driven control that occurs during normal operation for the pipeline stages shown in Fig. 3. Similar to the approach established by digital asynchronous pipelines [28], it uses a combination of handshakes and event-triggers to initiate state-transitions in the stage control units.

We begin in the system reset state, where all stages are idle and either tracking their input or waiting for permission to begin tracking. The arrival of the master clock initiates a chain

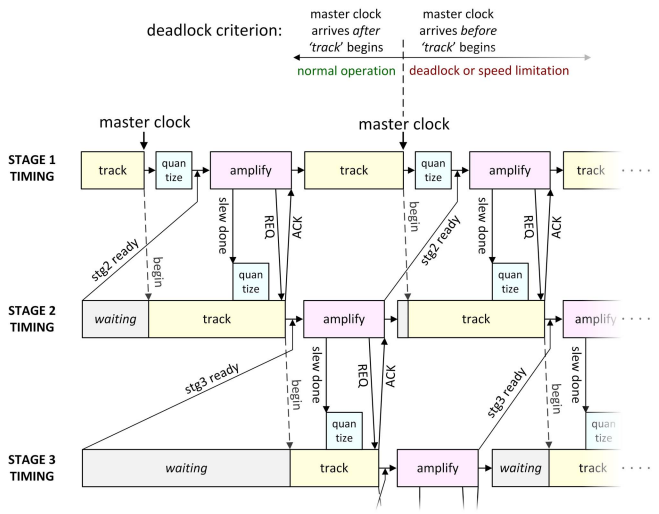


Fig. 6. The chain-reaction of processing steps in the pipeline propagates based on event-triggers generated in the system.

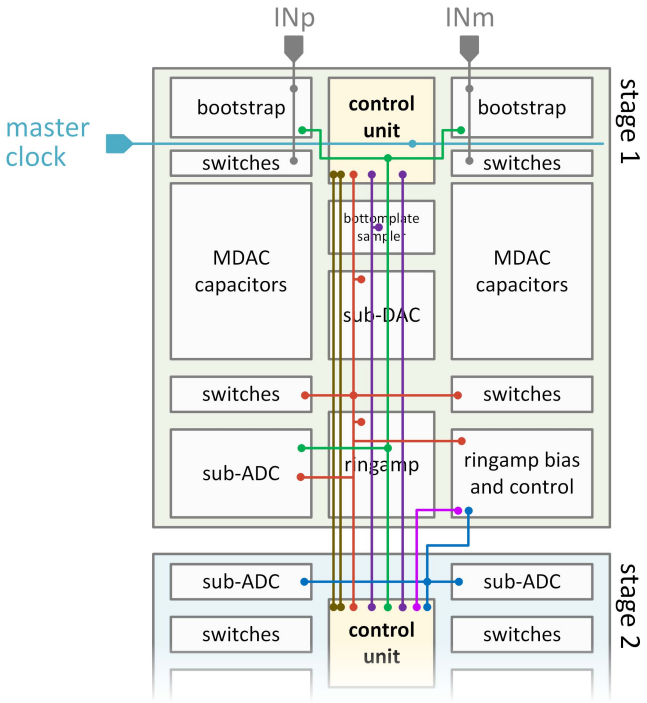


Fig. 7. Layout floorplan of the pipeline described in Section II. The control line coloring corresponds to the scheme of Fig. 3.

reaction. First, stage 1 samples its input and begins quantizing. When quantization is complete and stage 2 indicates that it is ready, stage 1 amplifies the residue onto the sampling capacitor of stage 2. At some point during amplification, the *slew done* signal will be generated by the ringamp sub-system. After this point, additional time is allocated for continued fine settling, set by the delay cell t_{d2} in Fig. 4a. When settling is complete, sampling request *sample 2 (REQ)* is sent to stage 2. Stage 2 then samples and returns an acknowledgment *track early 2 (ACK)*, allowing stage 1 to safely power down its ringamp and return to the track state.

Stage 2 now begins a similar exchange with stage 3, with a few notable differences. First, as discussed earlier, quantization happens in parallel with tracking. Second, when stage 2 is ready to resume tracking, it will wait until the *begin* signal of stage 1 indicates that it is safe to reconnect. This prevents stage 2 from reconnecting to stage 1's output just as stage 1 samples, and safeguards against kickback related errors. Like many aspects of the timing behavior shown here, this is a programmable option that can be disabled through firmware.

Note that the *ready* and *begin* signals in Fig. 6 are not physical. They are implicit in the information contained in the physical *track* and *amplify* signals of Fig. 3.

III. Advantages

In this section we discuss the main advantages of an asynchronous event-driven approach, not only as they apply to the specific implementation discussed in Section II but also in a more general sense, including interleaved systems.

A. Layout

Localized timing control results in the simplest possible top level clocking scheme, and this carries with it a number of physical layout advantages. As shown in the example floorplan of Fig. 7, the *master clock* is fed directly from the top level clock buffer into the control unit of the pipeline's first stage. For an interleaved layout, the *master clock* may also be fed into a global interleave controller, as shown in Fig. 8. This direct-feed approach eliminates the global clock generator of Fig. 1 and all associated fan-out. The larger the system, the more profound this advantage becomes. For example, the ADC described in [14], depicted by Fig. 8, has 4 interleaved channels each with 10 stages, and yet the *master clock* is only tapped at five locations (the four stage 1 control units and the global controller), all closely spaced, and connected using a single straight wire. By contrast, if this system were clocked according to Fig. 1, the layout would require at minimum two wires, each connecting to 40 stages across a much larger area. Power efficiency is obviously improved as a result, due to significantly reduced routing, loading, and parasitics. But even more importantly, by confining the jitter sensitive *master clock* to only the very first stage of the pipeline layout, it is physically isolated from all other stages, which minimizes coupling and cross-talk induced jitter. The task of ensuring adequate shielding is thus greatly simplified, reduced to only a few sensitive areas in the layout such as the two places in Fig. 7 where input signal and clock lines cross.

Localization of clocking and control has important layout advantages at the stage level as well. Minimization of routing distance becomes increasingly important in deep nanoscale CMOS technologies where interconnect parasitics can have a significant impact on clock propagation delays, which increase non-overlap timing overheads in the system. This problem is addressed by reducing the physical length between transmitter and receiver of all control lines in the system to one stage at maximum. Observe the example layout of the inter-stage control bus of Fig. 3 in the floorplan of Fig. 7. The wires

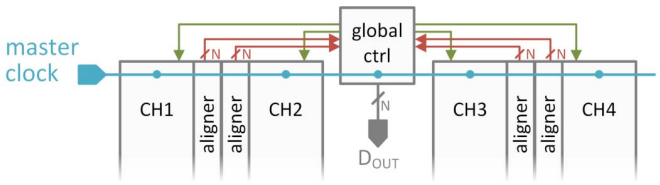


Fig. 8. Floorplan and top level control layout for the interleaved design of [14].

connecting the two control units are routed directly over the stage's primary circuitry, making it possible to tap these lines for local sub-block control without increasing routing length significantly. The approach of using the same wire for communication and line driving minimizes the total number of high speed control lines in the system, saving power and compacting the layout. By contrast, the conventional scheme in Fig. 1 handles clock distribution on a global level, separately from line driving which occurs locally only after re-buffering.

Although conventional wisdom often frowns upon running un-shielded clock lines over sensitive analog circuits, in practice it is fairly straightforward to ensure that coupling is signal-independent and does not impact performance. This is because in an event-driven system we know the exact relationship between all control lines and analog blocks and we know which phases of operation they will have interactions during. For example, if a control line related to sampling runs over the ringamp layout, coupling is irrelevant because the ringamp is powered down during the tracking and sampling events.

B. Correct-by-Construction

The unification of wiring used for communication and wiring used for controlling functional analog blocks into a single set of physical wires also conveys another important advantage. Notice how the two control units in Fig. 7 function as transceivers, and how all of the circuity that loads the control lines lies physically between these two transceivers. As a consequence, there is an inherent self-accounting for delay and rise/fall time non-idealities in each "event-trigger" signal when it arrives at the input of a control unit receiver.

Combined with the inherent causality of an event-driven approach, this leads to the important property of correct-by-construction control, where safe and valid phase relationships are guaranteed. By contrast, in the traditional approach of Fig. 1 there is no self-accounting for loading, and skew mismatch can accumulate to significant levels due to the highly distributed nature of the layout. This can be partially compensated by precision modeling and post-layout analysis, but ultimately there is still some amount of safety margin that must be included in the timing budget.

The advantage offered by the event-driven approach relies on one key assumption: that the underlying state machines are themselves logically correct. The validation effort required to confirm this is in most cases an acceptable trade-off; whereas clock tree validation is a laborious process heavily dependent on post-layout modeling accuracy and variables that cannot be fully controlled, logical validation of control units can be done at design time (pre-layout) and is technology and model independent.

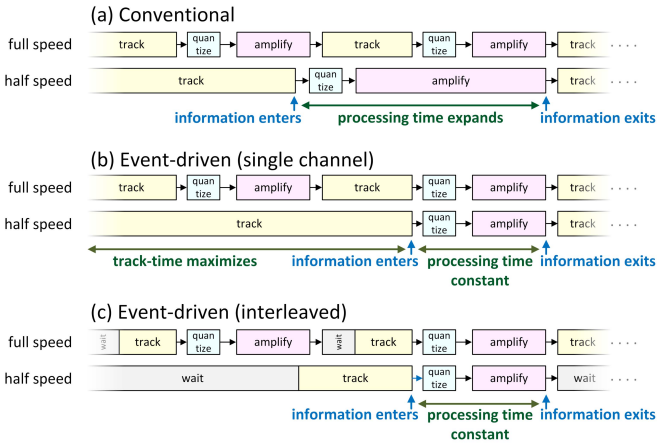


Fig. 9. Frequency scaling behavior of sub-phases for (a) conventional clocking, (b) event-driven single-channel, and (c) event-driven interleaved.

C. Sampling-Rate Invariant Processing

From the moment an input sample is captured, all processing inside the pipeline is based on a chain-reaction of internally generated events. Information always propagates down the pipeline at the desired (programmed) internal rate regardless of the external clock rate. Several advantages arise from this property.

Foremost among these is the possibility of constant energy-per-conversion independent of clock rate. In other words: fully-dynamic power consumption. This is particularly useful for reconfigurable multi-standard applications, enabling high efficiency across a range of conversion speeds. Fig. 9 compares internal phase transitions in pipeline stage 1 when the externally driven *master clock* frequency is varied by a factor of two. In the conventional case, the time span from when a packet of analog information enters the stage to when it exits is a function of clock frequency. Even if all circuits have zero quiescent current in their off-state and the residue amplifier is only powered on during *amplify*, the width of *amplify* and consequent amplifier on-state static power consumption is still proportional to clock rate. By contrast, for the two event-driven cases in Figs. 9b and 9c, all phases related to actual processing tasks (*amplify*, *quantize*) are *master clock* invariant, and the energy-per-conversion always remains constant.

Fig. 9b also shows how this behavior allows for an automatic and optimal maximization of track time. As soon as the *amplify* phase has finished, the ADC can return immediately to tracking. In the conventional case, the track time is always related to both the rising and falling edges of the *master clock* and can never be optimized without negatively affecting the timing of other phases. A similar constraint also appears in interleaved systems that only connect one ADC channel to the input at a time (Fig. 9c). Thus, this benefit applies primarily to single-channel systems only.

If pursuing multi-rate reconfigurability, signal corruption due to device leakage must also be considered. The event-driven approach offers a key advantage in this regard: because there is effectively only one internal speed of propagation down the pipeline, all phases related to residue processing are only subject to one leakage condition. For example, using large bottom-plate sampling switches in the MDAC is desired for high bandwidth, but during processing-related phases these large switches also introduce leakage current that can corrupt the charge information stored on the MDAC capacitors. In Fig. 9a where the processing time expands for lower clock rates, this poses a problem, since leakage will integrate for a proportionally longer span of time. In Figs. 9b and 9c, the amount of integrated leakage current is always the same, because the elapsed time during which charge-based information is stored and processed is clock-rate invariant.

However, the event-driven approach does not remove leakage concerns for the clock-rate dependent *track* and *waiting* phases. Fortunately, simply by virtue of these phases' primary function (listening, not processing), leakage effects will tend to lie outside of the main signal path. For example, in the design of Section II the bootstrap circuit and the ringamp common-mode feedback (CMFB) circuit (shown in Fig. 4a, described in [14]) are sensitive to leakage during these phases. In both cases the sensitive nodes hold trapped charge used for DC-biasing, and simply using standard-Vt doped devices and sizing transistors appropriately is found to be sufficient to ensure acceptable leakage across a large range of operating speeds without performance trade-offs

The invariance of the internal behavior to external conditions also makes the approach well suited to non-uniform sampling applications, such as compressed sensing and event-triggered biomedical.

D. Jitter and Track Time Optimality

As illustrated in Figs. 9b and 9c, when a given pipeline stage is done processing and has handed off its residue to the next stage, it automatically returns to either a tracking or waiting state. From this behavior arises a subtle but important advantage in terms of input sampling jitter.

Causality mandates that all edge transitions in a clocked system be derived from an initial driven source, the *master clock*. Fig. 10a shows the most ideal two-phase non-overlapping clocking in this regard, where the edge transitions are spaced as close as possible while still obeying all relationships required for intra-stage and inter-stage timing correctness.[2] A trade-off between jitter and track-time can be observed here. The *track* phase can't possibly rise until instant A4, but it can already end as early as B2. From a jitter perspective, this is ideal because the falling (sampling) edge of *track* has minimum delay with respect to the *master clock* and thus minimum accumulated clock-path jitter. But from a track-time perspective, this is sub-optimal because it will be less than the theoretical maximum track time width. Track time can be expanded by adding 2 extra units of delay to all of the edges in the "B" region, such that *track* rises on A4 and falls on B4, but this increases jitter of the sampling edge due to more inverters in the critical path. At very high speeds where every picosecond matters, this subtle trade-off can become quite consequential.

[2]Despite the popular convention of labeling sampling edges as "early", a "delay" terminology is used here to emphasize the causal link between the *master clock* edge and all derived clock edges in the system.

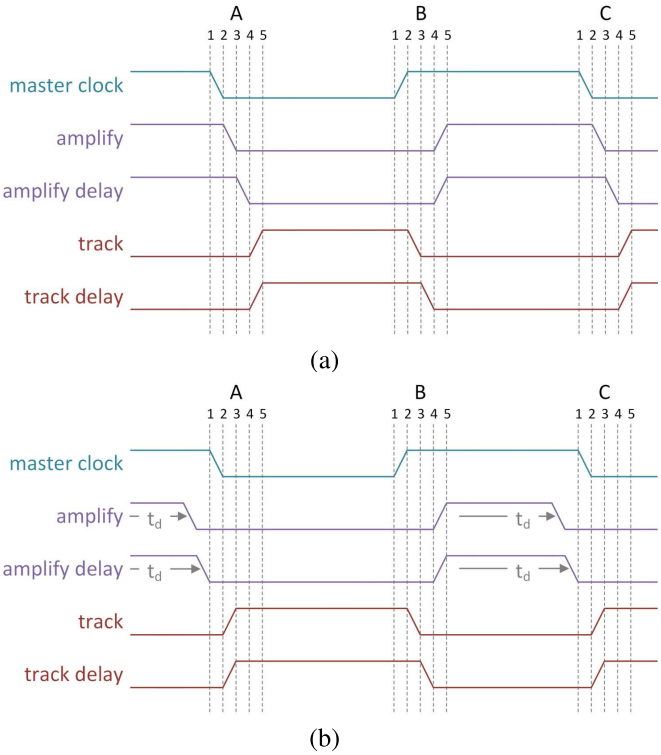


Fig. 10. Largest possible track time duration when sampling jitter is prioritized for (a) synchronous and (b) event-driven clocking schemes.

Ultimately, the root cause of this problem is that all edges of *track* and *amplify* are derived from the externally driven *master clock*. As shown in Fig. 10b, by removing one of these constraints and instead allowing the width of *amplify* to be freely set with an internal time delay (derived from a previous *master clock* edge), the jitter / track-time trade-off can be eliminated. The *track* phase is now free to rise at A2 and fall at B2, maximizing track-time and minimizing jitter.

E. Reconfigurability

The sequence and duration of processing steps orchestrated by a stage control unit is always partitioned optimally regardless of the particular state-flow. Any increase in duration of an active processing phase will correspond to an automatic and equal decrease in the duration of an idle phase (e.g., *track* and/or *wait*). In synchronous clocking this is not possible because both the rising and falling edges of the *master clock* are involved in phase generation, which constrains the available options for how phases can be ordered and partitioned.

Furthermore, entire phases of operation can be added or removed, simply by reconfiguring state-flow logic in firmware. For example in the stage 1 control unit implementation of [29], an optional phase can be inserted between *track*

and *amplify* that will pre-clear the stage 1 MDAC capacitors before connecting them to the ADC input. The specific criteria that initiate state transitions in the control logic can also be reconfigured. For example, the "stg ready" blocking conditions illustrated in Fig. 6 can optionally be disabled with the *D_clearCareful* digital control bit shown later in Fig. 12.

Deep pipelines are naturally imbued with an inherent modularity that few other classes of ADC enjoy; increasing or decreasing the target resolution is mostly a matter of adding or subtracting stages. Event-driven control simplifies this further by eliminating the clock tree, and if a standardized inter-stage communication interface is used, it leads to modular design re-use. Building blocks that implement the standardized interface can be chained and interleaved together regardless of the internal method of operation. For example, with an appropriately defined interface protocol, the last six stages of the pipeline in Fig. 2 can be replaced with an asynchronous SAR, or even several SARs interleaved together, despite significantly different internal operation and clocking compared to the backend of Fig. 2.

This resolution reconfigurability is complemented by the speed reconfigurability of fully-dynamic power consumption. For example, if a "design library" of modular building blocks can all operate at a maximum speed of 1 GS/s, and a 12-bit 400MS/s ADC is needed, it is simply a matter of chaining together the correct blocks and clocking at the desired rate, with no sacrifice in efficiency or performance.

These reconfigurability concepts are demonstrated by the design presented in [14] compared to the ones in [19], [29], [30], where a 13-bit channel, interleaved four times, was designed for direct RF-sampling base-station applications and then, with fairly minor modifications, was reconfigured as an 11-bit single channel ADC targeting mobile terminal applications.

A key impediment to true "plug-and-play" building blocks is the required agreement between stages regarding the load capacitance that the residue amplifier will drive. This can often be less problematic than it initially appears because a number of different amplifier drive strengths are usually already available in the design library as a product of pipeline stage scaling. With proper planning for swappable layouts between the varying sizes of residue amplifier, adaptation to different loading conditions when joining blocks together is possible with modest design effort.

IV. Challenges

A. Re-Synchronization

As shown in Fig. 2, alignment of the digital outputs to form the final ADC output code can be done using a standard approach with the *sample REQ* signals in Fig. 3. The digital output of any stage is guaranteed to be valid when *sample REQ* of that same stage asserts, and it is also guaranteed through causality that this assertion will not happen at the same time as the *sample REQ* signals of directly adjacent stages. However, once aligned at the end of the pipeline, the output data stream is still asynchronous with respect to the external *master clock*. For the architecture in Fig. 2 we solve this problem with the simple approach of shifting all data back up to the first stage using the same causally related *sample REQ* edges. When it arrives at stage 1, the output data can be safely re-synchronized with the *master clock*. As seen in Fig. 8, this results in an atypical layout where the digital output emerges at the front rather than the back of the pipeline. Proper shielding can prevent data-dependent coupling into the input and clock lines. Although this solution is simple and easily verifiable, it also more than doubles the power of the aligner sub-system and requires shielding. Alternative re-synchronization methods that avoid both of these drawbacks are also possible, but it is left for future work.

B. Deadlocks

Under certain boundary conditions, an illegal system state can arise in some control implementations where the chain-reaction of self-timed events in the pipeline halts indefinitely in a so-called "deadlock". A deadlock can be either digital or analog in origin.

Deadlocks of digital origin primarily pertain to the power-on state of the digital control units. To ensure proper operation, all state machines should be initialized to a known and valid configuration before *master clock* edges are allowed to initiate conversions. As illustrated in Fig. 6, this entails forcing the state machine back into the *track* or *waiting* state, depending on the stage. After system reset, all stages are idle and waiting to receive a new packet of analog information.

Deadlocks of analog origin arise when an analog circuit involved in event-trigger generation is in an invalid state. Although not all systems will necessarily contain such circuits, in the design of Section II we do find an example of this in the ringamp sub-system. Observe how the detector circuit in Fig. 4 that generates the *slew done* event-trigger operates based on the assumption that the ringamp eventually reaches its target value and that the outputs are not saturated. If they are saturated, the slewing operation never finishes, *slew done* will never be dispatched, and the chain-reaction of events in the pipeline will halt. Saturation is indeed possible under certain boundary conditions. Differential-mode saturation can occur, for example, as the result of an input signal amplitude beyond full-scale or sub-ADC errors exceeding the redundancy range. Common-mode saturation can occur during system power-on when the CMFB circuit of the ringamp is not yet settled to a valid bias point, or when the input common-mode voltage is outside the range tolerated by the CMFB.

Both the digital and analog causes of deadlock can be resolved by following a simple strategy: 1) monitor for deadlocks, 2) if one occurs, "flush" the pipeline by forcing it back into a known reset state, 3) design any sensitive analog systems such that they can reach correct bias convergence through repeated forced returns to the "reset" state.

Fortunately, the first step (deadlock detection) is trivial. If a deadlock occurs anywhere in the pipeline, the effects of this blockage will eventually appear at stage 1. For example, if a deadlock occurs in stage 6, stages 1 through 5 will still temporarily be able to accept data, but the next sample propagating down the pipeline will never get clearance to enter stage 6 and halt at stage 5. The sample after that will

halt at stage 4, and so on. When this backlog appears at stage 1, it is possible to detect a deadlock based on the simple criterion illustrated in Fig. 6: under normal operation, stage 1 should always be in the *track* state at the moment that the *master clock* arrives to initiate sampling. If stage 1 is in any other state during this event, we know that either 1) a deadlock has occurred or 2) the external clock rate is higher than the internal processing rate can support. We will discuss this second possibility in the next sub-section, but for now, let us assume the first case.

To force the system back into its reset state following detection of a deadlock, the deadlock detector in stage 1 dispatches a system-reset command which is relayed to all stages in the pipeline through the *reset request* lines in Fig. 3. When this request reaches the last stage, it responds with an acknowledgment that is relayed back up the pipeline to stage 1 through the *reset acknowledge* lines in Fig. 3. After the acknowledgment is received, the original request is released. It is important that the stage 1 control logic be designed to block the *master clock* trigger during this entire process and only allow a new conversion when the acknowledgment at stage 1 has finally de-asserted again.

Finally, any analog, signal-dependent, or meta-stability sensitive circuits involved in generating event-triggers must be designed such that repeated returns to the system reset state will eventually lead to proper bias convergence. Appendix VII-A provides an example of this, describing how the ringamp in Section II that generates the *slew done* event-trigger is designed for robust DC bias convergence at power-on. Such solutions may come at the cost of slower bias settling at power-on due to repeated deadlock and reset events.

In the architecture of Section II, *slew done* can also optionally be generated with a constant time delay that triggers some fixed time after *amplify* begins by setting D_{MODE}="0" in Fig. 4a. This option eliminates analog-based event triggers in the system altogether, which can be useful in systems where certain analog circuits cannot be designed to gracefully recover from deadlock, where saturated analog input signal levels must still be tolerated and correctly processed, or where very fast startup convergence is required.

C. Timing Control

Beyond some maximum clock rate, the throughput of internal processing will not be fast enough to keep up with input sampling requests and a backlog will eventually form. Like deadlocks, this scenario can be detected by the criterion of Fig. 6. However, unlike a deadlock, returning to system reset will not resolve the underlying problem. This can only be done by either increasing the internal processing speed or by reducing the external clock rate.

To avoid system resets and corrupted cycles in the ADC digital output, measures should be taken to avoid "hard" timing violations that trigger deadlock detection. One approach is to make internal timing controls PVT invariant by-design and provide sufficient margin for any remaining variability. A variety of well established techniques already used in industry can enable this, such as PVT-insensitive delay cells and temperature-aware digital control. Another option is to monitor the excess timing margin available in each pipeline stage and adjust delay elements adaptively. One possible implementation of this concept is described in Appendix VII-B. In addition to providing timing optimization, by relaxing PVT stability requirements, adaptive control allows tunable delay cells to be integrated directly into the digital standard cell layouts of stage control units.

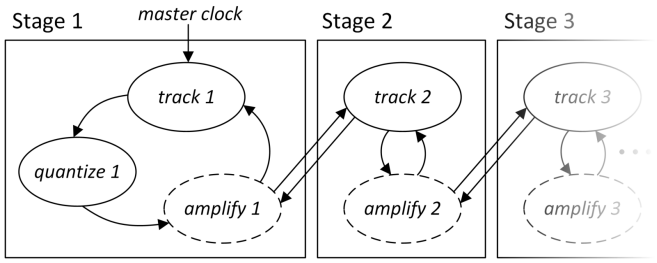


Fig. 11. Simplified state flow under normal operation. Implicit (redundant) states are denoted by dashed lines.

V. Control Unit Implementation

A simplified model of control unit state flows and interactions in the architecture of Section II is shown in Fig. 11. Hardware implementation can follow a number of strategies. One option is to design the control unit as a self-contained state machine. This maximizes modularity, enabling the generic interfacing protocol discussed in Section III-E. However, a self-contained implementation is often not the most timing-efficient. For example, note how *amplify 1* of stage 1 can be inferred implicitly from other states in the system: if *quantize 1* has already been visited and stage 2 is currently in *track 2*, we can deduce that stage 1 should also be in *amplify 1*. The generation of control signals corresponding to the *amplify 1* state can therefore be generated using only combinational logic. This is usually faster than the transition delay of a flip-flop memory element, and allows for less wasted non-overlap time.

Ultimately, the optimal balance between modularity and timing efficiency comes down to system level motivations. We have experimented with both: the control units of [29], [31] are modularity optimized and the control units of [14], [19] are speed optimized. The stage 1 and stage 2 speed-optimized control units of [19] are shown in Fig. 12. The control units of stages 3-7 are identical to stage 2 and the backend stage is similar but not identical. Simulated waveforms of key nodes in the Fig. 12 control units are plotted in Fig. 13 alongside other relevant event triggers and analog residue voltages from Fig. 3 and Fig. 4a.

The signal *(track early 1)'* in Fig. 12 is the jitter-sensitive sampling edge connected to the gate of the PMOS bottom plate sampling switch in the stage 1 MDAC. The jitter-critical path between *master clock* and this signal consists of only four gates, all sized for noise. All other logic gates in the system are sized only with regard to loading. The control logic in Fig. 12 contains a number of speed optimizations that

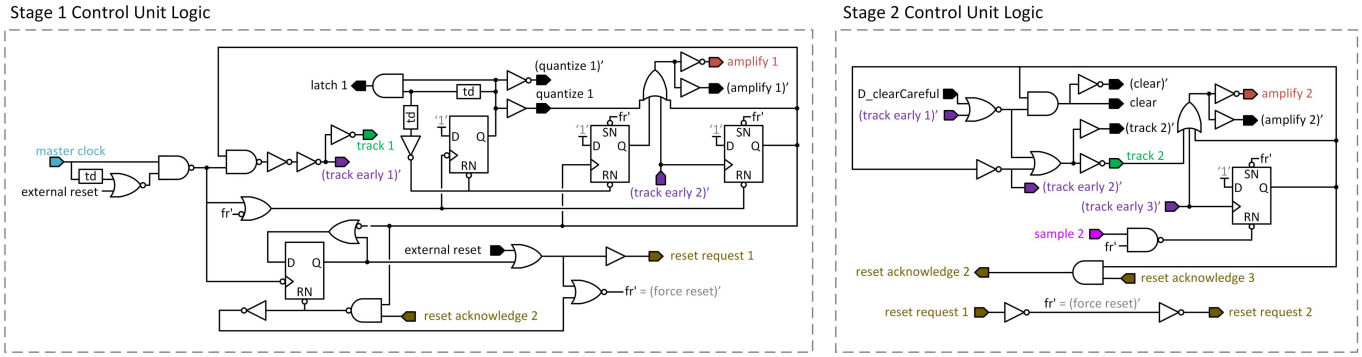


Fig. 12. Control unit implementation for the design of [19]. See Fig. 3 for inter-stage connections and nomenclature.

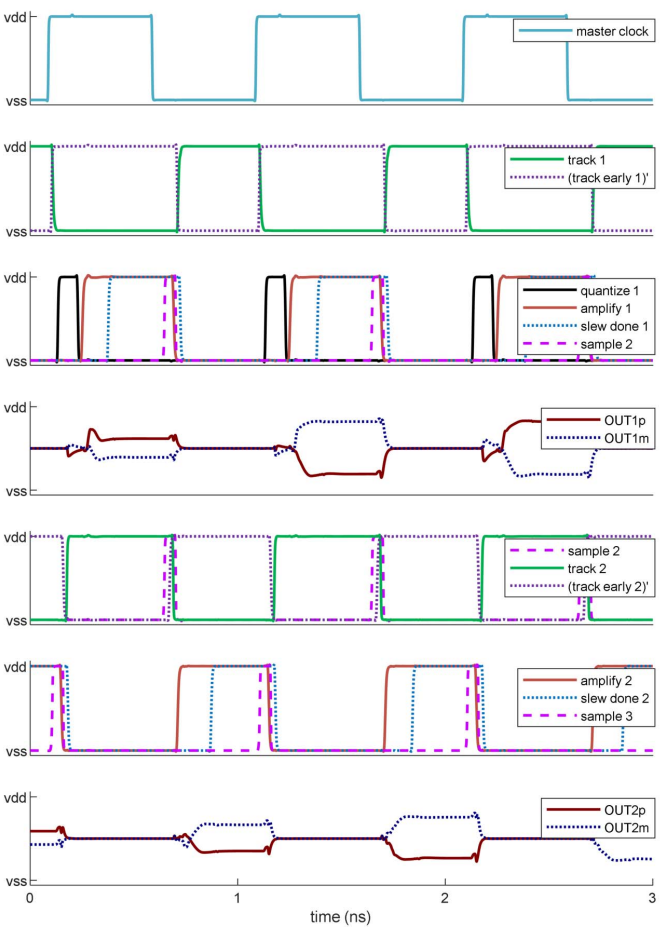


Fig. 13. Simulated waveforms of key signals in the control units of Fig. 12 as well as the analog outputs of the two pipeline stages they control.

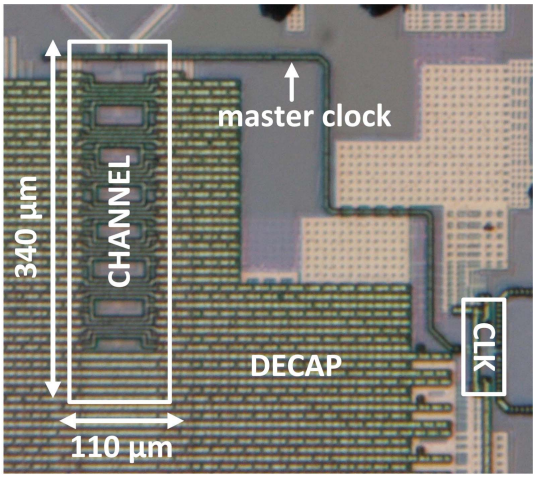


Fig. 14. Die photo of [29] that implements Section II.

break strict causality. For example, *(track early 2)'* initiates the de-assertion of *amplify 1* through the 4-input OR gate, but in parallel it initiates the assertion of *track 1* through a different logic path that includes a flip-flop transition. In a strictly causal system, this second path should be triggered from the falling edge of *amplify 1*. However, this would result in extra non-overlap time between *amplify 1* and *track 1*. By using a feed-forward path, we enhance speed by narrowing down the non-overlap time. Simulation analysis shows it is safe to do so: not only is there still some non-overlap time even with this feed-forward, but there are additional delays in the input sampler bootstrap circuit not shown here that further delay the actual start of tracking.

In interleaved systems such as [14] the stage 1 control unit requires additional sampling-related logic to coordinate with the global interleave controller as well as different deadlock resolution logic compared to Fig. 12.

VI. Measurements

This section presents measurements from two different designs that both implement the single-channel architecture of Section II [19], [29]. We focus here specifically on aspects of these designs relevant to timing control not reported in [19], [29]. Except where otherwise noted, all measurements are obtained using a fixed configuration of digital controls and fixed digital reconstruction coefficients (calculated off-chip). This includes all timing control settings.

To begin with the ADC of [29], a die photo is shown in Fig. 14. It occupies an active area of 0.037 mm^2 (110 μm $\times$ 340 μm), excluding decoupling. The *master clock* can be seen running from the clock input buffer directly into stage 1 of the pipeline. The ADC operates on an 850 mV supply with VREFp = 50 mV and VREFm = 800 mV. At 600 MS/s with a -0.1 dBFS Nyquist input tone it consumes 6.0 mW (including clock buffer) and achieves 60.2dB SNDR, 78.3dB SFDR, 12.0 fJ/conversion-step FoM$_W$, and 167.2 dB FoM$_S$. The

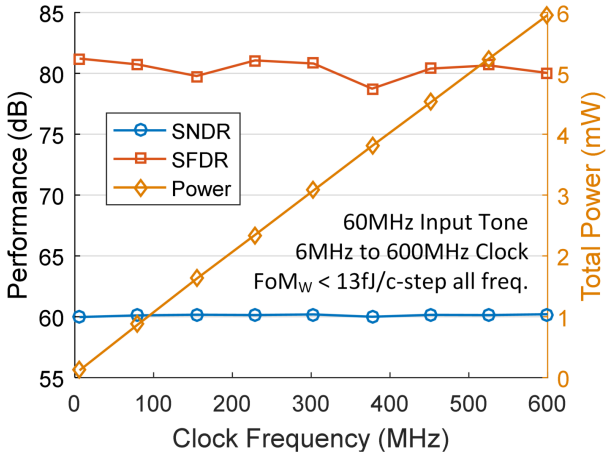


Fig. 15. Constant energy-per-conversion independent of clock of [29].

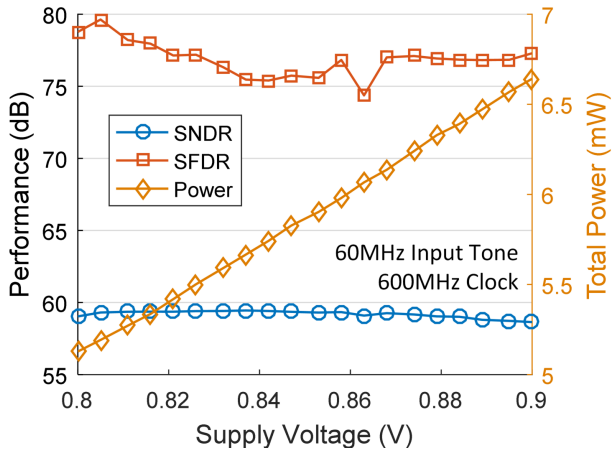


Fig. 16. Resilience of [29] to supply voltage variation.

fully-dynamic, linear scaling of power with respect to clock frequency is depicted in Fig. 15, maintaining better than 13 fJ/conversion-step FoM$_W$ from 6 MS/s to 600 MS/s. The supply sweep of Fig. 16 demonstrates the resilience of the fixed time delay elements in the control system to voltage variation.[3] A breakdown of simulated power consumption is depicted in Fig. 17. The "Control Unit and Line Driver" slice also includes all switching power (switches are driven directly by the control system). The "All Other Digital" slice includes the global clock buffer and digital output alignment and re-synchronization circuitry.

Measurements obtained from the ADC reported in [19] provide additional insights. As depicted in Fig. 4a, *slew done* can be generated using either the analog detection circuit (sensor) or the programmable delay t_{d1} (fixed delay). Unique to this particular design, the *slew done* signal is used to control critical events in both the early quantization system described in Section II-D and a reference regulation system described in [19]. In both cases, *slew done* initiates actions that disrupt ringamp settling and should ideally trigger immediately after slewing. Table I summarizes measured performance for

[3]To sweep the supply voltage down to 800 mV without saturating, VREFp is lowered to 750 mV for this test.

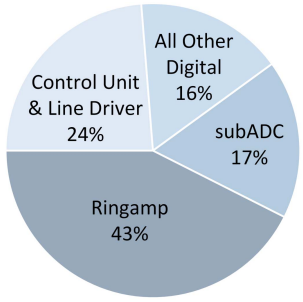


Fig. 17. Simulated power breakdown of [29].

TABLE I
DIFFERENT USES OF *slew done* FOR THE ADC OF [19]

Config	Sampling Rate	Reference Regulation	*slew done* Mode	ENOB	SNDR	SFDR	THD
A	1 GS/s	off (ideal)	sensor	9.63	59.7	72.9	69.7
B	1 GS/s	off (ideal)	fixed delay	9.64	59.8	75.9	70.4
C	1 GS/s	on	sensor	9.14	56.8	62.4	60.4
D	1 GS/s	on	fixed delay	9.56	59.3	75.7	70.0
E	800 MS/s	off (ideal)	sensor	9.61	59.6	72.6	69.0
F	800 MS/s	off (ideal)	fixed delay	9.61	59.6	72.6	69.3
G	800 MS/s	on	sensor	9.58	59.5	71.9	67.6
H	800 MS/s	on	fixed delay	9.59	59.5	73.0	68.9

different modes of operation. Comparing configurations A and B, we see that when *slew done* only controls early quantization, even at the maximum clock rate, using the self-triggered sensor yields performance almost as good as a well-configured fixed delay. However, comparing configurations C and D we find that when *slew done* also controls switching in the regulation system, performance is noticeably degraded, but only for the sensor case. As evidenced by configurations E, F, G, and H, when the clock frequency is reduced and the ringamps are given additional time to settle (by tuning t_{d2} in Fig. 4a), this discrepancy is again eliminated. A few conclusions can be drawn from these experiments. First, this illustrates a design trade-off: although the *slew done* sensor simplifies certain aspects of design by automatically tracking PVT, a properly configured fixed delay is shown to trigger at a more optimal time. Intuitively this makes sense, because causality dictates that the sensor-based event-trigger can only assert after the event actually occurs, whereas the fixed time delay can be tuned to trigger at the actual event instant (or even before). An additional concern of using the fixed-delay, however, is that it does not account for the signal-dependent component of slewing duration, and so to be safe, it must be configured for the longest slewing time possible (i.e. largest output swing). What waveforms like Fig. 4 suggest and these measurements support is that the variation in slewing duration is small enough to not overshadow the benefit of the fixed-delay, and a fixed delay may be the preferred choice for speed-critical designs. Still, there is no clear overall winner; each implementation option has pros and cons that must be evaluated in the context of the design at hand.

TABLE II
PERFORMANCE SUMMARY COMPARING [29] TO [19], [30]

Variant of Section II Architecture	[29] Hershberg ISSCC 2019		[19], [30] Hershberg VLSI 2020		[19], [30] Hershberg VLSI 2020	
Technology	16nm CMOS		16nm CMOS		16nm CMOS	
Supply	850 mV		850 mV		900 mV	
Sampling Rate	600 MS/s		800 MS/s		1 GS/s	
Resolution	11b		11b		11b	
Input Range	1.5 V pk-pk diff.		1.5 V pk-pk diff.		1.6 V pk-pk diff.	
Performance at 1GS/s:	*60 MHz input:*	*300 MHz input:*	*80 MHz input:*	*400 MHz input:*	*100 MHz input:*	*500 MHz input:*
ENOB	9.7 b	9.7 b	9.6 b	9.5 b	9.7 b	9.6 b
SNDR	60.4 dB	60.2 dB	59.4 dB	59.1 dB	59.8 dB	59.5 dB
SFDR	83.1 dB	78.3 dB	75.4 dB	75.1 dB	78.6 dB	75.9 dB
THD	76.2 dB	73.2 dB	69.7 dB	69.4 dB	71.8 dB	69.9 dB
Total Power	6.0 mW		7.3 mW		10.9 mW *	
Walden FoM	12.0 fJ/c-step		12.3 fJ/c-step		14.1 fJ/c-step	
Schreier FoM	167.2 dB		166.5 dB		166.1 dB	

*Includes reference regulation (0.9mW)

The two prototypes are compared in Table II. The design of [29] can reliably operate up to 600 MS/s with an 850 mV supply, whereas the design of [19] can reliably operate up to 800 MS/s with an 850 mV supply. Although there are several factors that, in total, are responsible for this discrepancy, differences in control unit implementation (i.e. optimization for modularity vs. speed) do play an important role. When the supply is increased to 900 mV the faster of the two designs can operate up to 1 GS/s, maintaining better than 14fJ/conversion-step FoM_W from 1 MS/s to 1 GS/s. When running each ADC continuously for approximately one week, no deadlocks are detected after the initial power-on transient in either one.

VII. CONCLUSION

This paper provides the first in-depth exploration of asynchronous event-driven timing control techniques as applied to deep pipelined ADCs. Initial implementations are shown to be enhanced by the associated benefits, exhibiting state-of-the-art performance and efficiency. The freedom and flexibility offered by a fully programmable control system creates new opportunities for design creativity, opening the door to a wider array of techniques than were previously possible in the conventional two-phase clocking paradigm. Opportunities for future work include alternative solutions to the challenges discussed in this paper, such as re-synchronization, timing self-optimization, and deadlock resolution.

APPENDIX

A. Ensuring Ringamp DC Bias Convergence

This appendix explains how the biasing of the ringamp in Fig. 4a will converge correctly, even when starting from an unknown state during system power-on. The ringamp contains three CMFB feedback paths: two fast, and one slow [14]. The slow path sets the DC bias and is contained in the "Trapped Charge CMFB" block of Fig. 4a. It operates by sensing the common-mode error at the ringamp output and sampling the sign of this error onto C_SMALL at the end of the *amplify* phase.

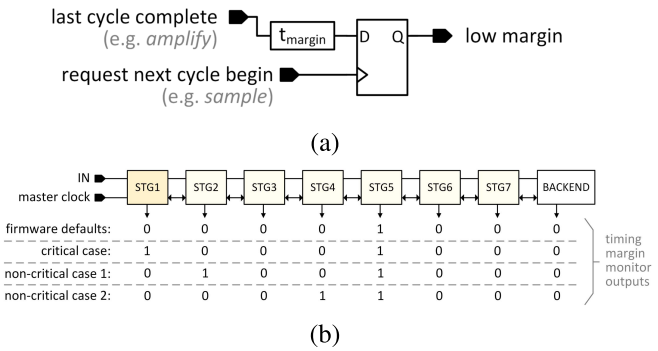


Fig. 18. (a) A monitor circuit to detect relative timing margins, with example input signals from the stage 2 control unit of Fig. 12. (b) Monitor outputs for different timing scenarios.

The charge on C_SMALL is used to update the trapped charge stored at node CM_DC, and thus sets the DC bias condition of the ringamp.

In the event of an analog-based deadlock, the pipeline stage always halts in the *amplify* phase (EN=1 in Fig. 4a), and because of this, the ringamp still generates a residue and the sign of the common-mode error in this residue is still charged onto C_SMALL. When the deadlock is detected, the stage will be forced back into reset, C_SMALL will still sample the correct error information, and CM_DC will still be updated correctly. This error feedback will eventually push the ringamp into a valid bias condition and out of saturation after a sufficient number of system resets. This procedure converges even when multiple stages in the pipeline are in invalid states.

B. Methods for Timing Violation Detection and Correction

This appendix describes a possible approach to using adaptive control to configure tunable time delays in the system. Consider the simple monitor circuit in Fig. 18a that observes how much time exists between the end of one processing cycle and the beginning of the next. If the pipeline stage is not ready and waiting at least t_margin before the next sampling request arrives, then the *low margin* flag will assert. This "early warning" system allows internal timing to be adjusted in the background before a backlog actually occurs.[4] Due to the presence of the *master clock* in stage 1, a monitor placed there can provide a measure of absolute timing margin. In all other stages, only relative comparisons are possible. For example, if a monitor in stage 5 indicates *low margin*, this only tells us that stage 4 processes data packets faster than stage 5. It says nothing about whether stage 5 might cause a backlog.

There are a number of possibilities for using monitoring for background control. The simplest option is to act based on stage 1's information about absolute timing margin only. If *low margin* asserts there, decrement the total processing delay in all stages of the pipeline by one digital code. Although we don't know which stage is the timing bottleneck,

[4]Referring to the internal control unit labels of stage 2 in Fig. 12, for stages 2-7, t_margin is the desired safety margin, *last cycle complete* is *amplify*, and *request next cycle begin* is *sample N*. In stage 1 of an interleaved system *request next cycle begin* is *master clock*. In stage 1 of a single channel system such as this one, t_margin should be set to the minimum acceptable track time and *request next cycle begin* should be signal *(track early)'*.

by decrementing all of them it is certain that the problem will be resolved.

It is often possible to more accurately pinpoint the origin of timing bottlenecks. Consider the four monitor output cases in Fig. 18b. The "firmware defaults" case represents an initial power-on configuration where the control settings are deliberately chosen so that the *low margin* flags will all output "0" nominally[5] but due to some process variation in this particular piece of silicon stage 5 outputs a "1" anyway (indicating that it is slower than stage 4). The other three output cases are hypothetical scenarios encountered due to further PVT drift. In the "critical case", it is clear that stage 1 is the bottleneck. Likewise, in "non-critical case 1" it is clear that stage 2 is slowing down relative to stage 1. For "non-critical case 2", there is some ambiguity: either stage 4 or stage 5 could be the culprit. Using this type of information, it is possible to implement adaptive control that can better pinpoint which stages need to be adjusted. However, the specific implementation and control algorithm is left for future work.

REFERENCES

[1] B. Razavi, "Design considerations for interleaved ADCs," *IEEE J. Solid-State Circuits*, vol. 48, no. 8, pp. 1806–1817, Aug. 2013.

[2] B. Hershberg, S. Weaver, K. Sobue, S. Takeuchi, K. Hamashita, and U.-K. Moon, "Ring amplifiers for switched capacitor circuits," *IEEE J. Solid-State Circuits*, vol. 47, no. 12, pp. 2928–2942, Dec. 2012.

[3] J. Lagos, B. Hershberg, E. Martens, P. Wambacq, and J. Craninckx, "A single-channel, 600-MS/s, 12-b, ringamp-based pipelined ADC in 28-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 54, no. 2, pp. 403–416, Feb. 2019.

[4] J. Lagos, B. P. Hershberg, E. Martens, P. Wambacq, and J. Craninckx, "A 1-GS/s, 12-b, single-channel pipelined ADC with dead-zone-degenerated ring amplifiers," *IEEE J. Solid-State Circuits*, vol. 54, no. 3, pp. 646–658, Mar. 2019.

[5] L. Kull *et al.*, "A 3.1 mW 8b 1.2 GS/s single-channel asynchronous SAR ADC with alternate comparators for enhanced speed in 32 nm digital SOI CMOS," *IEEE J. Solid-State Circuits*, vol. 48, no. 12, pp. 3049–3058, Dec. 2013.

[6] P. J. A. Harpe *et al.*, "A 26 μW 8 bit 10 MS/s asynchronous SAR ADC for low energy radios," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1585–1595, Jul. 2011.

[7] T. Jiang, W. Liu, F. Y. Zhong, C. Zhong, K. Hu, and P. Y. Chiang, "A single-channel, 1.25-GS/s, 6-bit, 6.08-mW asynchronous successive-approximation ADC with improved feedback delay in 40-nm CMOS," *IEEE J. Solid-State Circuits*, vol. 47, no. 10, pp. 2444–2453, Oct. 2012.

[8] B. Verbruggen, M. Iriguchi, and J. Craninckx, "A 1.7 mW 11 b 250 MS/s 2× interleaved fully dynamic pipelined SAR ADC in 40 nm digital CMOS," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2012, pp. 466–468.

[9] E. Martens, B. Hershberg, and J. Craninckx, "A 69-dB SNDR 300-MS/s two-time interleaved pipelined SAR ADC in 16-nm CMOS FinFET with capacitive reference stabilization," *IEEE J. Solid-State Circuits*, vol. 53, no. 4, pp. 1161–1171, Apr. 2018.

[10] B. Vaz *et al.*, "A 13 b 4 GS/s digitally assisted dynamic 3-stage asynchronous pipelined-SAR ADC," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2017, pp. 276–277.

[11] W. Jiang, Y. Zhu, M. Zhang, C.-H. Chan, and R. P. Martins, "A 7.6 mW 1 GS/s 60 dB SNDR single-channel SAR-assisted pipelined ADC with temperature-compensated dynamic Gm-R-based amplifier," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 60–62.

[5]Determining a default setting that meets this criteria is usually possible without performance penalty because input-referred settling requirements of each stage progressively decrease down the pipeline (and thus require less processing time overall).

[12] S. P. Kopparthy, I. Makwana, and A. Gupta, "Asynchronous 8-bit pipelined ADC for self-triggered sensor based applications," in *Proc. Asia Pacific Conf. Postgraduate Res. Microelectron. Electron.*, Dec. 2012, pp. 205–210.

[13] L. Picolli, A. Rossini, P. Malcovati, F. Maloberti, F. Borghetti, and A. Baschirotto, "A clock-less 10-bit pipeline-like A/D converter for self-triggered sensors," *IEEE J. Solid-State Circuits*, vol. 43, no. 2, pp. 312–320, Feb. 2008.

[14] B. Hershberg *et al.*, "A 4-GS/s 10-ENOB 75-mW ringamp ADC in 16-nm CMOS with background monitoring of distortion," *IEEE J. Solid-State Circuits*, early access, Feb. 23, 2021, doi: 10.1109/JSSC.2021.3053893.

[15] T.-C. Hung, J.-C. Wang, and T.-H. Kuo, "A calibration-free 71.7 dB SNDR 100 MS/s 0.7 mW weighted-averaging correlated level shifting pipelined SAR ADC with speed-enhancement scheme," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 256–258.

[16] X. Tang *et al.*, "A 13.5-ENOB, 107-μW noise-shaping SAR ADC with PVT-robust closed-loop dynamic amplifier," *IEEE J. Solid-State Circuits*, vol. 55, no. 12, pp. 3248–3259, Dec. 2020.

[17] T.-H. Wang, R. Wu, V. Gupta, and S. Li, "A 13.8-ENOB 0.4 pF-CIN 3rd-order noise-shaping SAR in a single-amplifier EF-CIFF structure with fully dynamic hardware-reusing kT/C noise cancelation," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 374–376.

[18] X. Tang, X. Yang, J. Liu, W. Shi, D. Z. Pan, and N. Sun, "A 0.4-to-40 MS/s 75.7 dB-SNDR fully dynamic event-driven pipelined ADC with 3-stage cascoded floating inverter amplifier," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 64, Feb. 2021, pp. 376–378.

[19] B. Hershberg, N. Markulic, J. Lagos, E. Martens, D. Dermit, and J. Craninckx, "A 1-MS/s to 1-GS/s ringamp-based pipelined ADC with fully dynamic reference regulation and stochastic scope-on-chip background monitoring in 16 nm," *IEEE J. Solid-State Circuits*, vol. 56, no. 4, pp. 1227–1240, Apr. 2021.

[20] B. Razavi, "Charge steering: A low-power design paradigm," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2013, pp. 1–8.

[21] S.-H.-W. Chiang, H. Sun, and B. Razavi, "A 10-bit 800-MHz 19-mW CMOS ADC," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 935–949, Apr. 2014.

[22] Z. Zheng *et al.*, "A single-channel 5.5 mW 3.3 GS/s 6 b fully dynamic pipelined ADC with post-amplification residue generation," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2020, pp. 254–256.

[23] T. Matsuura *et al.*, "A 240-mbps, 1-W CMOS EPRML read-channel LSI chip using an interleaved subranging pipeline A/D converter," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1840–1850, Nov. 1998.

[24] O. Stroeble, V. Dias, and C. Schwoerer, "An 80 MHz 10 b pipeline ADC with dynamic range doubling and dynamic reference selection," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, vol. 1, Feb. 2004, pp. 462–539.

[25] B.-M. Min, P. Kim, F. W. Bowman, D. M. Boisvert, and A. J. Aude, "A 69-mW 10-bit 80-MSample/s pipelined CMOS ADC," *IEEE J. Solid-State Circuits*, vol. 38, no. 12, pp. 2031–2039, Dec. 2003.

[26] T. Sundstrom, C. Svensson, and A. Alvandpour, "A 2.4 GS/s, single-channel, 31.3 dB SNDR at Nyquist, pipeline ADC in 65 nm CMOS," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1575–1584, Jul. 2011.

[27] E. Martens, B. Hershberg, and J. Craninckx, "Wide-tuning range programmable threshold comparator using capacitive source-voltage shifting," *Electron. Lett.*, vol. 54, no. 25, pp. 1417–1418, Dec. 2018.

[28] S. M. Nowick and M. Singh, "High-performance asynchronous pipelines: An overview," *IEEE Des. Test. Comput.*, vol. 28, no. 5, pp. 8–22, Sep. 2011.

[29] B. Hershberg *et al.*, "A 6-to-600 MS/s fully dynamic ringamp pipelined ADC with asynchronous event-driven clocking in 16nm," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 68–70.

[30] B. Hershberg, N. Markulic, J. Lagos, E. Martens, D. Dermit, and J. Craninckx, "A 1MS/s to 1GS/s ringamp-based pipelined ADC with fully dynamic reference regulation and stochastic scope-on-chip background monitoring in 16nm," in *Proc. IEEE Symp. VLSI Circuits*, 2020, pp. 1–2.

[31] B. Hershberg *et al.*, "A 3.2 GS/s 10 ENOB 61 mW ringamp ADC in 16 nm with background monitoring of distortion," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 58–60.

Benjamin Hershberg (Member, IEEE) received the H.B.S. degree in electrical engineering and in computer engineering from Oregon State University in 2006 and the Ph.D. degree in electrical engineering from Oregon State University in 2012, with a focus on ADCs and scalable, and low-power switched-capacitor amplification solutions, including the invention of ring amplification.

In 2013, he joined the Interuniversity Microelectronics Center (imec), Leuven, Belgium. He is currently a Principal Member of the Technical Staff and conducts research on a variety of topics spanning the receiver chain from the RF frontend down to the ADC. He has authored or coauthored more than 50 articles, book chapters, and patents.

Dr. Hershberg was a recipient of the 2010 Analog Devices Student Design Award and the ISSCC 2019 Lewis Winner Award for Outstanding Paper. He was a winner of the 2012 Broadcom University Research Competition.

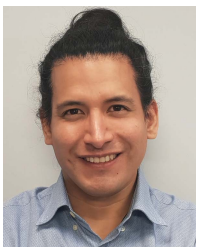
Barend van Liempd (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2009 and 2011, respectively, and the Ph.D. degree from the Vrije Universiteit Brussel, Brussels, Belgium, in 2017, in collaboration with the imec, Heverlee, Belgium. His Ph.D. dissertation concerned tunable, and highly integrated RF front-end circuits and modules in SOI CMOS.

From 2011 to 2014, he joined the imec, working as an Research and Development Engineer, on multi-standard transceivers, and became a Ph.D. Researcher and later a Senior Researcher in 2017. He was appointed as a Program Manager Radar ICs in 2018 and currently leads the imec's radar IC research and development activities. He has authored or coauthored more than 30 articles, patents, and patent applications. His research interests are analog, RF, and mm-Wave circuits for wireless and sensing applications.

Dr. Liempd was a co-recipient of the 2015 NXP Prize at the European Microwave IC (EuMIC) Conference and the 2019 Lewis Winner Award for Outstanding Paper at the International Solid-State Circuits Conference (ISSCC).

Nereo Markulić (Member, IEEE) received the M.Sc. degree in electrical engineering from the University of Zagreb, Croatia, in 2012, and the Ph.D. degree *(summa cum laude)* from the Vrije Universiteit Brussel, Belgium, in 2018. His Ph.D. work was in collaboration with the Interuniversity Microelectronics Center (imec), Leuven, Belgium, on digital subsampling phase-locked loops (PLLs) and polar transmitters.

He is currently a Research Scientist at imec, working on RF and mixed-signal circuits for radar applications and next generation connectivity. He has authored and coauthored publications and patents on PLLs and analog-to-digital converters, and a book on frequency synthesis. He was a co-recipient of the ISSCC 2019 Lewis Winner Award for Outstanding Paper. He currently serves on the Technical Program Committee of the IEEE VLSI Circuits Symposium.



Jorge Lagos (Member, IEEE) received the B.Sc. degree *(summa cum laude)* in electronics engineering from the Pontificia Universidad Católica del Perú (PUCP), Lima, Peru, in 2003, the M.Sc. degree *(cum laude)* in nanotechnologies for integrated systems from the Politecnico di Torino, Turin, Italy, the Institut Polytechnique de Grenoble, Grenoble, France, and the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2013, and the Ph.D. degree *(summa cum laude)* from the Vrije Universiteit Brussel, Brussels, Belgium, in 2019, with a focus on high-performance and power-efficient analog-to-digital converters, in collaboration with imec, Leuven, Belgium.

From 2004 to 2014, he occupied several Research Assistant positions with a focus on various topics. From 2004 to 2006, he was with PUCP, researching analog VLSI networks for channel decoding. From 2006 to 2011, he was with the Politecnico di Torino, working on chip-level electromagnetic compatibility and system-on-chip functional test. From 2012 to 2014, he was with the Fraunhofer Institute for Integrated Circuits, Erlangen, Germany, focusing on front-end design for MEMS sensors. In 2018, he joined the imec, where he is currently a Senior Researcher in analog and mixed-signal IC design.

Dr. Lagos was a co-recipient of the 2019 ISSCC Lewis Winner Award for Outstanding Paper.

Ewout Martens (Member, IEEE) received the M.Sc. degree in electronic engineering and the Ph.D. degree *(summa cum laude)* from the Katholieke Universiteit Leuven, Belgium, in 2001 and 2007, respectively, with a research on modeling techniques for delta-sigma modulators and RF front-ends. From 2007 to 2010, he worked at a spin-off company of KU Leuven, as a Chief Scientist, focused on analog design automation. In 2010, he joined the imec, Belgium, as a Researcher for WLAN transceivers, and is currently working as a Principal Member of the Technical Staff within imec's Department of the IoT. His interests and work are related to the development of innovative architectures for RF receiver front-ends, transceiver building blocks, like PLL, filters and LNA, and ADCs for various applications, including image sensors and wireless receivers. From 2017 to 2020, he served as a Technical Program Committee Member of the Symposium on VLSI Circuits.

Davide Dermit (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the University of Modena and Reggio Emilia, Italy, in 2005, 2007, and 2011, respectively.

In 2008, he was an Intership Student with Entropic Communications, Sophia-Antipolis, France, working on integrated high-speed digital PLL frequency synthesizers in 65 nm CMOS for cable-TV. From 2010 to 2015, he was an Analog and RF ICs Designer with STMicroelectronics, France, working on integrated I^2C shifters in 600 nm BCD for the HDMI standard and integrated 10–28 GHz receivers in 28–14 nm CMOS for multi-standard PHY and SerDes. In 2015, he joined the imec, Leuven, Belgium, where he is currently working as an Analog and RF ICs Designer. In imec, he worked on integrated Cartesian digital RF modulators in 28 nm CMOS for software defined radios, while his current main research interest is the design of integrated high-speed ADCs in 16 nm FinFET.

Dr. Dermit was a co-recipient of the 2019 Lewis Award for Outstanding Paper at 2019 ISSCC.

Jan Craninckx (Fellow, IEEE) received the M.S. and Ph.D. degrees *(summa cum laude)* in microelectronics from the ESAT-MICAS Laboratories, KU Leuven, in 1992 and 1997, respectively. His Ph.D. work was on the design of low-phase noise CMOS integrated VCOs and PLLs for frequency synthesis.

From 1997 to 2002, he worked with Alcatel Microelectronics (later part of STMicroelectronics) as a Senior RF Engineer on the integration of RF transceivers for GSM, DECT, Bluetooth, and WLAN. In 2002, he joined the IMEC, Leuven, Belgium, as a Principal Scientist, working on RF, analog, and mixed signal circuit design. He is currently a IMEC Fellow. He has authored and coauthored more than 200 articles, book chapters, and patents. His research focuses on the design of RF transceiver front-ends in nanoscale CMOS, covering all aspects of RF, mmwave, analog, and data converter design.

Dr. Craninckx was elected as a SSCS AdCom Member for the term 2017–2019. He is/was a regular member of the Technical Program Committee for several SSCS conferences and was the Chair of the SSCS Benelux chapter for the term 2006–2011. He currently serves as a RF Subcommittee Chair for the IEEE International Solid-State Circuits Conference. He was an Associate Editor from 2009 to 2016 and the Editor-in-Chief from 2016 to 2019 of the IEEE JOURNAL OF SOLID-STATE CIRCUITS. He was a SSCS Distinguished Lecturer for the term 2012–2013.