

Minimum-Energy Sub-Threshold Self-Timed Circuits: Design Methodology and a Case Study

Omer Can Akgun, Joachim Rodrigues
Lund University,
Dept. of Electrical and Information Technology,
Box 118, 221 00 Lund, Sweden
E-mail: {omercan.akgun, joachim.rodrigues}@eit.lth.se

Jens Sparsø
Technical University of Denmark,
Dept. of Informatics and Mathematical Modelling,
Building 322, DK 2800 Lyngby, Denmark
E-mail: jsp@imm.dtu.dk

Abstract—This paper addresses the design of self-timed energy-minimum circuits, operating in the sub- V_T domain. The paper presents a generic implementation template using bundled-data circuitry and current sensing completion detection. To support this, a fully-decoupled latch controller has been developed, which integrates the current sensing circuitry. The paper outlines a corresponding design flow, which is based on contemporary synchronous EDA tools, and which transforms a synchronous design, into a corresponding self-timed circuit. The design flow and the current-sensing technique is validated by the implementation of an asynchronous version of a wavelet based event detector for cardiac pacemaker applications in a standard 65 nm CMOS process. The chip has been fabricated and the area overhead due to power domain separation and completion detection circuitry is 13.6 %. The improvement in throughput due to asynchronous operation is 52.58 %. By trading the throughput improvement, energy dissipation is reduced by 16.8 % at the energy-minimum supply voltage.

Keywords—minimum-energy, subthreshold, self-timed, case study, methodology

I. INTRODUCTION

Power density and consumption of complex digital systems have become a major concern during the recent years, both due to thermal concerns and limited battery life-time in mobile applications. Significant reduction in power consumption is achieved by lowering the supply voltage of the circuits [1]. This is possible by relaxing the constraints of classical strong-inversion operation of MOSFETs, and by accepting the notion that transistors are operated well below threshold, in the sub-threshold (weak-inversion) regime.

In sub-threshold (Sub- V_T) mode, the supply voltage may be scaled aggressively and thus power consumption is decreased by magnitudes. Sub-threshold operation of static CMOS logic has been analyzed using the EKV model in [2]. In this analysis, it is shown that static CMOS logic may be operated with a supply voltage as low as 50 mV at ambient temperature. There are several successful implementations of digital circuits operating in the sub-threshold regime in the literature such as, an FFT processor that is operational down to 180 mV [3] and a sub-threshold SRAM which operates

with a supply voltage of 160 mV [4]. Circuits operating at these extreme low supply voltages work at much lower speeds, as an example, the FFT processor presented in [3] works with a maximum clock frequency of 10 kHz with a power supply of 350 mV. Their extremely low power consumption results in excellent power delay product (PDP) values, making such circuits very interesting candidates for ultra-low power applications which do not have very high processing requirements.

In the sub-threshold regime, leakage current of the transistors are used for computation. The sub-threshold leakage current depends on the supply voltage exponentially, resulting in exponential increase in the circuit delay and leakage energy dissipation for lower supply voltages. Due to an exponential dependence of the leakage energy and quadratic dependence of the switching energy on the supply voltage in the sub-threshold regime, sub-threshold operation has an energy-minimum operating voltage (EMV). This minimum operating voltage may be lowered by decreasing the time a circuit spends in idle mode, i.e., the circuit is not operating but leaking. Thus, both leakage and dynamic energy are effectively reduced.

An attractive technique to reduce leakage energy is the application of asynchronous circuits. Because their performance is determined by actual case (rather than worst case) latencies, they can provide a higher throughput compared to the synchronous counterparts. However, if a gain in throughput is not necessary, the supply voltage may be lowered, which in turn reduces energy dissipation. Recently in [5] asynchronous circuits were studied from a low power and energy efficient operation perspective. Average case performance property allows asynchronous sub-threshold circuits to work with a higher energy efficiency [6].

There are many approaches to design asynchronous circuits, some of which have inherent completion detection capabilities. However, asynchronous circuit families with inherent digital completion detection incur both area and energy dissipation overhead. Especially in the sub-threshold regime energy increase due to higher leakage is more

pronounced [7]. In this work we focus on the implementation and de-synchronization flow development of single-rail circuits that operate with current sensing completion detection. The implemented external completion detection system was first introduced in [8].

The contributions of this paper are as follows: A de-synchronization flow employing commercial EDA tools is developed. A new fully decoupled latch controller that integrates the completion detection system to an asynchronous circuit implementation is designed and analyzed. Finally, using the developed de-synchronization flow and the latch controller, a self-timed version of the sub-threshold event detector for cardiac pacemaker applications is designed and fabricated, and functionality is verified by measurements. The remainder of this manuscript is organized as follows. Section II introduces advantages of asynchronous sub-threshold operation in terms of energy efficiency, as well as our reference design. Section III presents the current sensing completion detection system used in our asynchronous circuit implementation. In Section IV we present the de-synchronization flow and in Section V simulation results and preliminary measurements are given. Finally, in Section VI conclusions are drawn.

II. MOTIVATION AND BACKGROUND

The total energy dissipation of static CMOS digital circuits is given by the following well-known formula:

$$E_{total} = \underbrace{\alpha C_{load} V_{DD}^2}_{E_{dynamic}} + \underbrace{I_{leak} V_{DD} t_{leak}}_{E_{leakage}} + \underbrace{I_{peak} t_{sc} V_{DD}}_{E_{short-circuit}} \quad (1)$$

where $E_{dynamic}$ is the total dynamic energy consumed while charging the load capacitance C_{load} , with a switching probability of α . During the operation of the circuit, i.e., the circuit is powered, there exists leakage energy $E_{leakage}$ that is consumed during the leakage time t_{leak} . In addition, when a switching event occurs, for the duration t_{sc} , when both nMOS and pMOS transistors are conducting, some short circuit energy $E_{short-circuit}$ will be dissipated. In our energy dissipation analysis the contribution of the short circuit energy in the sub-threshold regime is neglected, as it is known to contribute only a small portion of the overall energy dissipation [2].

From (1) it is immediately clear that the energy dissipation of digital circuits can be reduced by lowering the supply voltage. It was first shown by Swanson as early as 1972 that CMOS digital operation can be realized with ultra-low supply voltages [9]. When the supply voltage is lowered aggressively, below the threshold voltage (V_T) of the MOS transistors, the digital circuit operates in the sub-threshold regime. In [2] and [9] it was proven that the minimum operating voltage for obtaining an absolute gain of more than 1 from a simple inverter and for guaranteeing bi-stability with sufficient voltage swing the lower limit of

supply voltage scaling was $4U_T$, where U_T is the thermal voltage whose value is 26 mV at 300 K.

A. Sub-threshold Operation

The drain current of an n-channel MOS transistor operating in this regime is specified as

$$I_{DS} = I_S \exp \frac{V_{GS} - V_T}{nU_T} \left(1 - \exp \frac{-V_{DS}}{U_T} \right) \quad (2)$$

where n is a process dependent term called slope factor and is typically in the range of 1.3 – 1.5 for modern CMOS processes [2]. The value of n depends on the depletion region characteristics of the transistor, i. e., $n = 1 + C_d/C_{ox}$. V_{GS} and V_{DS} are the gate-to-source and drain-to-source voltages, respectively. The parameter I_S is the specific current which is given by,

$$I_S = 2n\mu C_{ox} U_T^2 \frac{W}{L} \quad (3)$$

where μ is the mobility of carriers, C_{ox} is the gate oxide capacitance per unit area, and $\frac{W}{L}$ is the aspect ratio of the transistor.

As equation (2) shows, the drain current of a MOS transistor operating in the sub-threshold regime shows exponential dependence on the gate-to-source, drain-to-source voltages, slope factor, and the operating temperature. This exponential dependence of the drain current on the node voltages causes near-exponential changes in the operating speed of the circuit as the supply voltage varies [2]. As the supply voltage is lowered in the sub-threshold regime, the circuit delay as well as the leakage energy dissipation increase exponentially and the switching energy decreases quadratically, resulting in an energy-minimum operating point to occur. This is in contrast to the super-threshold operation where an energy-minimum operating voltage cannot be found. By operating asynchronously, both leakage and dynamic energy components can be reduced. The reduction in the leakage energy is due to the reduction of idle time of the circuit and the reduction in the dynamic energy is due to the moving of the energy-minimum supply voltage to a lower value as will be explained in Section II-C.

B. Asynchronous Operation

Components in an asynchronous system circuit operate largely autonomously. They are not governed by clock circuitry or a global clock signal, but instead need only wait for the signals that indicate completion of instructions and operations. These signals are specified by simple data transfer protocols. The data through the stages propagate by means of handshake signals that signal propagation of the data. This digital logic design is contrasted with a synchronous circuit which operates according to clock timing signals.

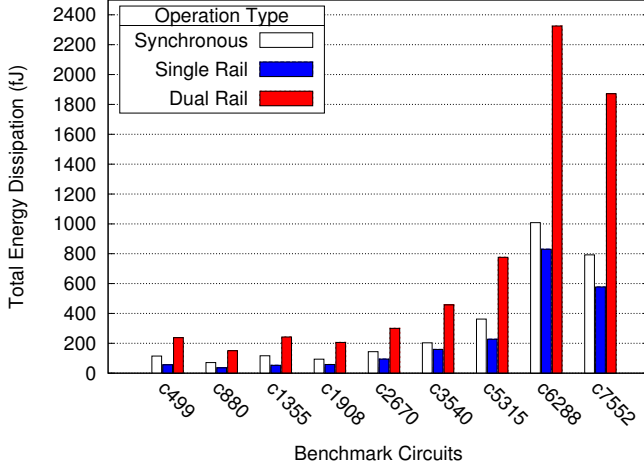


Figure 1. Energy dissipation comparison of ISCAS85 benchmark circuits for synchronous, single-rail and dual-rail implementations.

There are several reasons why asynchronous circuits are not as common as synchronous designs. First of all asynchronous circuits are more difficult to design than their synchronous counterparts. Synchronous designers are not concerned with what is going on between the latches/registers of a design as long as the data at the input of the memory elements is stable before the next clock signal. In contrast, asynchronous designs should be free of logic hazards in all the levels of abstraction [10] and switching activity must be properly ordered not to cause wrong data propagation. Although this is not an issue for the datapath of a bundled data design, logic hazards still need to be avoided in the asynchronous control circuitry. Second, asynchronous design methodologies are not fully supported by commercial Electronic Design Automation (EDA) tools, necessitating custom modifications to EDA tools for asynchronous design implementation. In our de-synchronization flow we employ commercial EDA tools for the implementation of the circuits, thus allowing the designers to work in a familiar environment.

Asynchronous circuit design techniques employing completion detection are very attractive for digital circuits operating in the sub-threshold regime because of their better than worst case operation speeds, resulting in higher average throughput and lower leakage energy dissipation. Efficient asynchronous operation requires a reliable method to determine the time required to complete processing of information. There are many methods to realize asynchronous circuits, some of which have inherent completion detection capability such as methods based on n -out-of- m coding. In practice, implementing digital circuits with digital completion detection properties bring circuit overhead due to the inherent redundancy for generating the completion detection signal. Recently in [7] Lotze et al. examined self-timed asynchronous circuits operating in the sub-threshold regime.

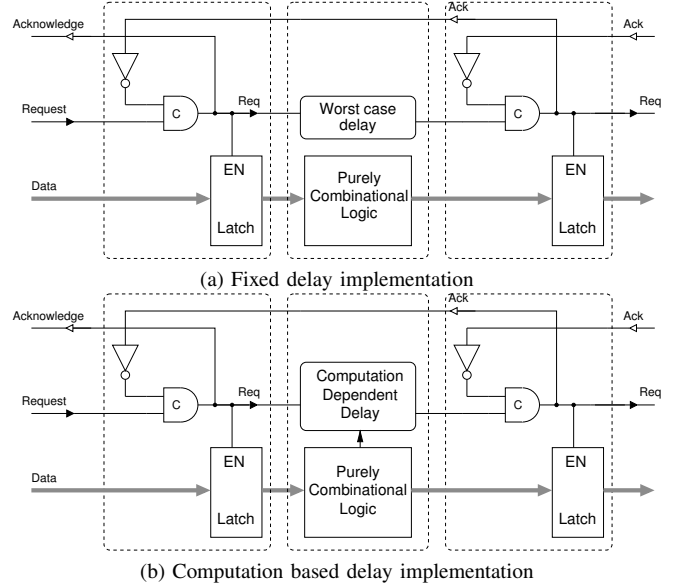


Figure 2. 4-phase bundled data pipeline (After [12]).

They implemented and simulated dual-rail multipliers with different bit-widths. For the 8-bit case their simulations showed an increase by a factor of 1.95 and 3.05 for the area and the switching energy, respectively.

To compare a purely digital self-timed implementation, i.e., dual-rail implementation, with the single rail implementation with external completion detection mechanism, we used ISCAS85 benchmark circuits [11]. Based on the findings in [7], we updated the circuit model parameters from the energy model in [6] to reflect dual-rail operation. The results of our numerical simulations showing the energy profile of ISCAS85 benchmark circuits for synchronous, single-rail and dual-rail implementations are shown in Figure 1. Single-rail and synchronous implementation are the same except average case operation is assumed for the single-rail implementation and the model parameters are updated accordingly. It is seen that for all the circuits, due to the increase in leakage and switching energy, dual-rail implementation has the highest energy dissipation. This confirms the result in [7] where it was suggested that only the critical path of the circuit should be designed using dual-rail techniques.

In this paper, we concentrate on a subset of asynchronous circuits that are based on asynchronous micro-pipelines first introduced by Sutherland [13]. The asynchronous circuit model that we will use in the remainder of this paper is a 4-phase bundled-data circuit shown in Figure 2 and is taken from [12]. In this type of asynchronous circuits, consecutive pipeline stages are separated using latches or registers controlled by an asynchronous finite state machine (AFSM). The *req* line is used to signal that new data is available for processing. Once the pipeline stage is ready to

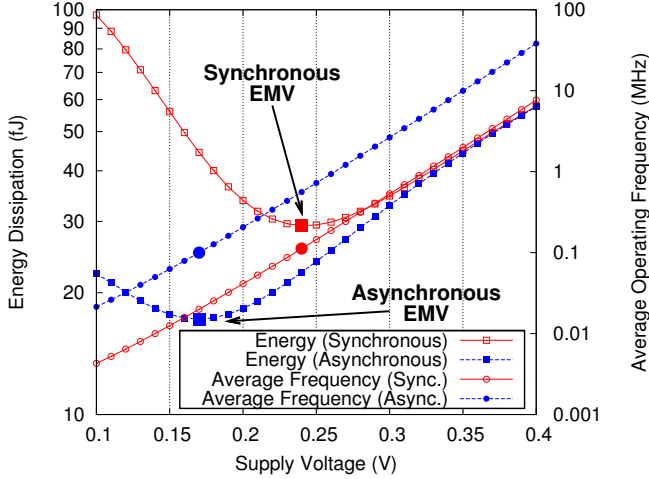


Figure 3. Total energy dissipation and average operation frequency for changing supply voltage values for a switching/delay mean value of 0.1. The energy-minimum operating points occurs at different voltage values for synchronous and asynchronous cases.

process new data, the AFSM will acknowledge this request by using the *ack* line. This will enable the latch/register, and new data will become available for processing by the combinational circuit. The completion of this operation will generate a new *req* signal to the following stage. Implementations differ depending on the signaling scheme used between AFSMs. Without loss of generality we will use the four-phase signaling scheme in our examples.

Traditional implementations of this circuit frequently use a *matched delay line* that has been engineered to have a delay that corresponds to the *worst case* delay through the combinational circuit as shown in Figure 2a. There are obvious disadvantages of using a fixed delay element for performance reasons, especially for coarse grained pipeline stages, where there is substantial variation in the operating speed depending on the input data switching probability. By operating such systems in a fixed delay fashion, unnecessary leakage energy consumption and throughput degradation will occur. Our solution for this problem is to employ an external completion detection circuit for sub-threshold operation. An overview of this solution will be given in Section III.

C. Energy Reduction By Asynchronous Operation

In [6] a comparison of synchronous and asynchronous circuits in terms of energy efficiency was performed. Energy dissipation of synchronous and asynchronous circuits are given as

$$E_T = C_{inv} V_{DD}^2 \left[\mu_e k_{cap} + k_{crit} k_{leak} e^{-V_{DD}/(nU_t)} \right], \quad (4)$$

and

$$E_T = C_{inv} V_{DD}^2 \left[\mu_e k_{cap} + k_{crit} k_{leak} (\mu_d + k_{com_{oh}}) e^{-V_{DD}/(nU_t)} \right], \quad (5)$$

for the synchronous and asynchronous cases, respectively. In the equations, C_{inv} is the total switched capacitance of an inverter, μ_e is the activity factor of the circuit and k_{crit} is the critical path delay normalized to the delay of an inverter. The parameter k_{leak} is obtained by normalizing the total average leakage current of the circuit by the average leakage of an inverter, and k_{cap} is obtained by normalizing the total capacitance of the circuit by the capacitance of an inverter. When equations (4) and (5) are compared, it is seen that the leakage energy contribution parts differ by a factor of $\mu_d + k_{com_{oh}}$, where μ_d is a parameter which denotes the average computation time of the asynchronous circuit and is in the range $[0, 1]$, and $k_{com_{oh}}$ is the asynchronous communication overhead. This difference in the leakage energy part of the equations results in lower leakage energy dissipation for asynchronous circuits as long as the value $\mu_d + k_{com_{oh}}$ is below 1.

When leakage energy of a circuit is reduced, the EMV moves to a lower value where the circuit dissipates lower switching energy while operating at a lower speed. From our numerical simulations based on the energy model, Figure 3 shows the energy and frequency profile of a sample design at a switching/delay mean of 0.1 for both asynchronous and synchronous operation. The k-parameters of the sample design were chosen such that the circuit has an energy dissipation equivalent to 1000 inverter gates with a drive capability of 1, and the critical path was chosen to be 25 inverter delays. In the simulations unless otherwise noted, the communication overhead parameter $k_{com_{oh}}$ is taken as 0.1. The EMV of the same circuit for synchronous and asynchronous operations for the specified mean values occur at 240 and 170 mV, respectively. Energy is reduced by 41.3 % from 29.3 fJ to 17.2 fJ by operating asynchronously.

Advantages of operating in an asynchronous manner in the sub-threshold regime are twofold. First, leakage energy is lowered by reducing the average time during the circuit is in idle mode, i.e., the time the circuit purely leaks. Second, lower leakage energy shifts the EMV to a lower value. This reduction in the supply voltage effectively reduces the switching energy. This can be seen in the plot of energy-minimum supply voltage values and their respective throughput values at those voltage values. In Figure 4 energy-minimum supply voltages and the operating frequencies for changing switching/delay properties are shown. The energy-minimum supply voltage of the asynchronous operation is lower, thus reducing the switching energy. The throughput worsens due to lower operating voltage but it is negligible

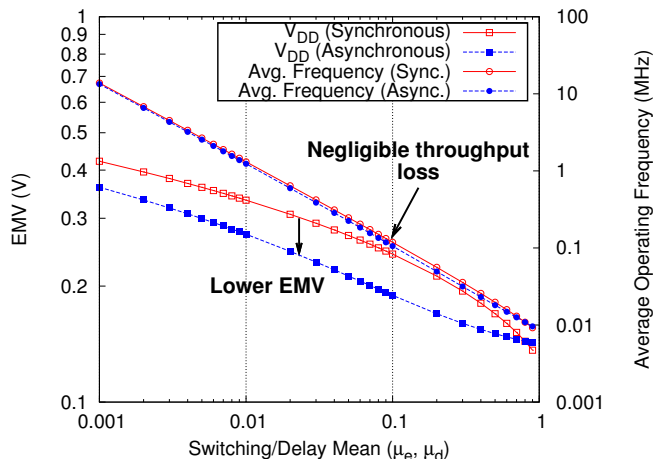


Figure 4. Energy-minimum operating voltage and the respective throughput change for varying switching/delay mean values. The throughput loss due to the lower energy-minimum operating voltage is negligible.

in asynchronous operation under the assumption that better-than-worst-case operation is possible.

D. Digital Event Detector for Cardiac Pacemakers

We use a CMOS implementation of a digital 3-scaled wavelet-based filter in combination with so-called hypothesis testing [14] for detecting the R-wave in a cardiac pacemaker as our reference circuit. The R-wave detector qualifies for pacemaker applications with reliable detection performance in noisy environments, and is validated on cardiograms recorded and digitized during pacemaker implantation. The architecture is optimized by register minimization, internal wordlength optimization, and numerical strength reduction. From now on, we refer to the cardiac event detector as our reference design in this paper.

III. CURRENT SENSING COMPLETION DETECTION SYSTEM

In this section we give an overview of the completion detection system that was first presented in [8] by Akgun et.al. This method may be applied to both the whole design using one completion detection circuit and the individual stages using dedicated completion detection circuitry for each stage. The first case, one completion detection circuit for the whole design, is presented in this paper.

A. Variable Delay Generation

In general, single-rail asynchronous circuits operate by delaying the control signals by an amount equal to the critical path delay [12]. Due to process variations this delay line has to be over-constrained, reducing the operating speed, and thereby also directly reducing the energy efficiency of the circuit. In order to be able to harvest the maximum energy efficiency out of this circuit we have to reduce the time the circuit spends *leaking*, both saving leakage energy

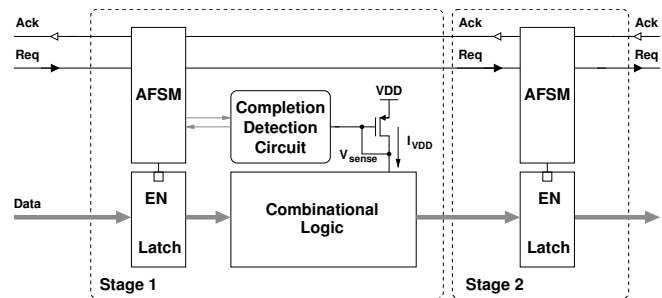


Figure 5. General block diagram of the completion detection system. The system consist of an asynchronous finite state machine, completion detection circuitry and a sensor transistor.

and moving EMV to a lower value. Hence switching energy is reduced in a quadratic manner.

Instead of using a fixed delay line, the completion of an operation may be detected. A technique to realize completion detection is to monitor the current drawn by the combinational block. As long as the combinational block is *switching* there will be dynamic power consumption in the circuit, which is detectable through the supply current I_{VDD} of the block. There are several implementations of completion detection circuits that use current sensing in the literature [15]–[17]. These methods rely on bipolar transistors, and/or resistors with high values, both of which are not always available in a standard process, or come as a process option with additional cost. The requirements on the bipolar transistors and resistors in these solutions set practical limits for the detection of current values in the μA -to- mA range. In this work we apply our de-synchronization flow to implement a current sensing completion detection system as presented in Figure 5. The completion detection system consist of an asynchronous finite state machine (AFSM), a completion detection circuit, which consist of a pulse generator and an AC-coupled amplifier, and a single pMOS transistor used for sensing the current. Implemented system is suitable for sub-threshold operation and can sense the current changes in the pA - nA range. Due to the simplicity of the system, the area overhead is very small.

B. Current Sensing

To be able to detect the operation phase of the circuit in the current sensing completion detection method, instantaneous current drawn by the circuit needs to be monitored. Thus, a circuit with low energy and area overhead that acts as an ammeter is required. We use a current sensing technique where the supply node (V_{DD}) of the combinational macro block is driven by a diode-connected low V_T PMOS transistor, see Figure 5 [8]. In this implementation, the current signal is sensed by the diode connected low V_T PMOS transistor and is converted to a voltage signal.

C. Latch Controller

A conventional bundled data pipeline, see Figure 2a, is constructed from handshake latches (each composed of a latch controller AFSM and a conventional enable-latch) and handshake combinatorial circuits (each composed of a matched delay element and a conventional combinatorial circuit).

Based on the handshaking on its input and output sides, a latch controller [18] produces a signal which opens and closes the latch. The signal transitions which opens and closes the latch are normally interwoven with the handshaking, in order to ensure that the latch is opened for a safe and sufficiently long period. Different data validity schemes may be obtained in this way [12], [19]. Finally, it is worth mentioning that the delay element used to match the latency in the combinatorial circuit delays both the rising and falling transitions on the request signal.

In our design we modify the latch controller AFSM as follows. The targeted sequential elements for implementation are positive-edge triggered flip-flops. Therefore, we use the A_{in} signal to clock the registers. Moreover, the use of current sensing completion detection leads to a situation where a signal event, A_{in+} , causes the completion detection circuit to generate a pulse, where the width matches the switching in the combinatorial part; the leading edge is caused by A_{in+} and the trailing edge of this pulse happens when the combinatorial circuit has settled. This situation, where a signal transition causes a pulse (i.e. two signal transitions) means that the completion detection circuit may not directly substitute a matched delay element. Therefore, we merge this pulse generating completion detection circuit into the latch controller, such that the outgoing request R_{out+} includes a delay which matches the subsequent combinatorial circuit.

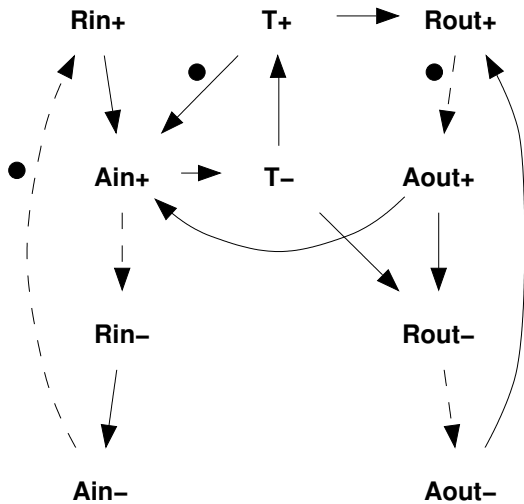


Figure 6. Signal transition graph for the designed controller.

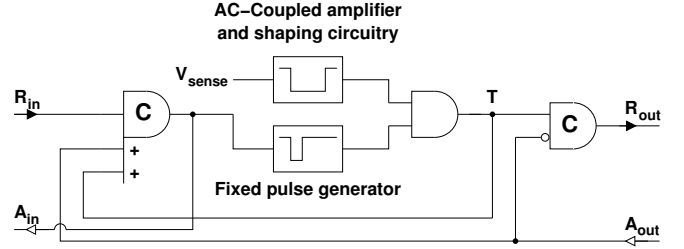


Figure 7. AFSM including the pulse generating completion detection circuitry.

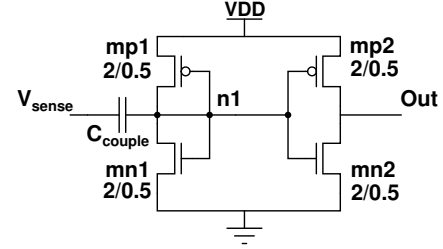


Figure 8. AC-coupled amplifier.

Figure 5 illustrates how the latch controller AFSM and the completion detection circuit work together. The behavior of the combined latch controller AFSM and completion detection circuit is specified in the signal transition graph (STG) in Figure 6. When focusing on the STG and the synthesis of the AFSM, it is sufficient to represent the completion detection circuit as a black box which is triggered by a positive signal transition (A_{in+}) and which then produces a variable width pulse ($T- \rightarrow T+$).

Looking at the STG it is seen that the handshaking on the input and output sides are totally decoupled; except for the fact that the latch controller will not accept new data (A_{in+}) until after $T+$. If only one AFSM is required, i.e., the circuit is working as a stand-alone processing unit and there are no external timing constraints, AFSM will work in two different oscillating settings: (i) R_{in} connected to A_{in} and R_{out} connected to A_{out} through an inverter, and (ii) R_{out} connected to R_{in} and A_{out} connected to A_{in} (i.e., the output port connected to the input port).

D. Completion Signal Generation

The completion detection circuit used to realize the asynchronous cardiac event detector is presented in Figure 7. Unlike the current sensing implementation presented in [8], we removed the variable pulse-width generator. This is possible due to the increased complexity and hence higher current drawn by the reference circuit. The sensed signal is strong enough to drive the AC-coupled amplifier to voltage values close to the supply voltage. Thus it is possible to shape this analog signal such that it is treated as digital and used to control the AFSM.

The implemented AC-coupled amplifier that is suitable for sub-threshold operation is shown in Figure 8. Diode connected transistors $mp1$ and $mn1$ bias the transistors $mp2$ and $mn2$, which are acting as an amplifier, at the maximum gain point for a given size and DC level. By changing the transistor sizes, the frequency response of the amplifier can be changed and there is a trade off between the gain required from the AC-coupled amplifier and the delay caused by the sensor transistor. If greater delay caused by the sensor transistor (larger spikes in the supply node of the combinational logic block) can be tolerated, the gain, thus the power consumption of the AC-coupled amplifier can be reduced.

A fixed pulse generator in parallel is connected to the AC-Coupled amplifier and shaping circuitry is implemented as well. This is necessary to generate a pulse signal for cases such as no combinational switching occurring, or for minimal amount of switching occurring in the circuit that cannot be amplified by the AC-coupled amplifier for converting to a logic level. Thus this fixed pulse generator both realizes the *time-out* feature and guarantees correct operation for the cases where the sensed signal is not strong enough. The sizing of the fixed pulse generator is done based on the HSPICE simulation results is explained in the next section.

IV. DE-SYNCHRONIZATION FLOW

To realize a self-timed version of the cardiac event detector circuit, a de-synchronization flow has been developed. There are multiple examples of synchronous-to-asynchronous conversion in the literature. In [20] the authors proposed the Doubly-Latched Asynchronous Pipeline (DLAP) approach. In this implementation the circuit is first synthesized into a synchronous structure by using commercial EDA tools. Then each register in the design is replaced by a pair of latches and the asynchronous controller. Flow described in [21] is similar to the DLAP approach. A fully automated synthesis flow that does not change the structure of the synchronous datapath is introduced. In this approach only the synchronization network is modified by replacing the clocking network of a synchronous circuit by a set of asynchronous controllers.

In the de-synchronization flow we are proposing the synthesized circuit remains unchanged, and only during the placement and routing step the registers are separated from the combinational gates and they are assigned a different power domain. Area overhead is due to the separation required between the power domains and, thus, is less than the previously proposed approaches.

A. Power Domain Separation

The current sensing completion detection concept is applicable to sense the current drawn by a digital circuit. In circuits where the majority of the gates are combinational,

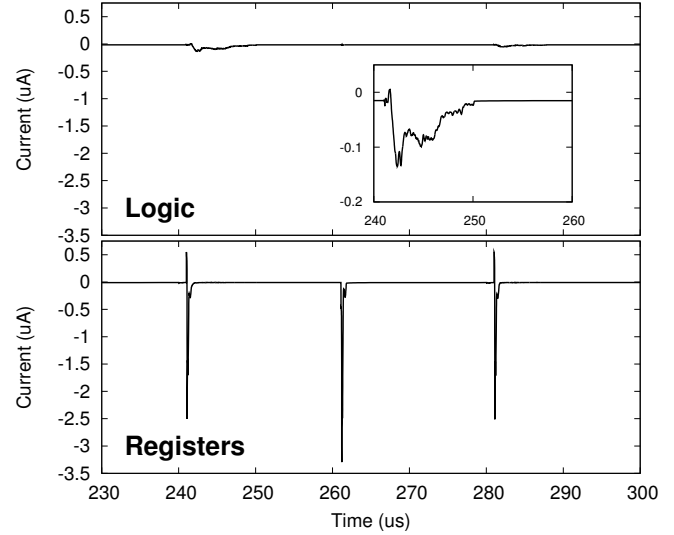


Figure 9. Separated current waveforms of the combinational gates and registers of the reference circuit.

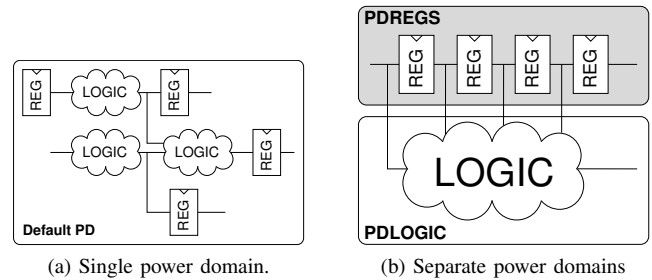


Figure 10. Concept of power domain separation for completion detection circuit implementation.

the same power domain may be used for both combinational and sequential elements. Due to the majority of the combinational gates, the current drawn by the whole circuit may be sensed with the sensor transistor without saturating the following AC-coupled amplifier.

However, the reference design uses a substantial number of sequential gates, i.e., registers. Separated current waveforms for the combinational logic and the registers are shown in Figure 9. The current waveform of the memory elements has sharp and instantaneous changes while the current waveform of the combinational logic part is spread over time with a smaller amplitude. Hence, separation of the combinational and memory elements of a complex circuit are crucial for proper operation of the current sensing completion detection system.

Asynchronous implementation of the reference design employs the separation of the power domains as shown in Figure 10. Positive edge triggered registers are driven by the A_{in} signal of the completion detection circuit shown in Figure 7. The separation process is automated and incurs little overhead in terms of routing and area.

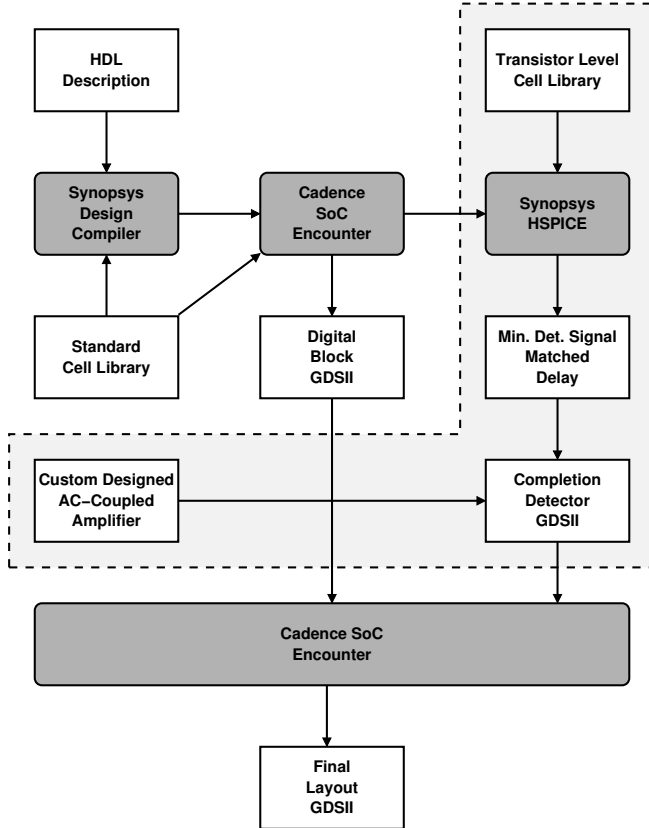


Figure 11. Design flow for the test chip

B. Using Commercial Design Tools for De-synchronization

The main advantage of the proposed flow is that stable and well known EDA-tools/flows for synchronous circuit design are used. The design flow to convert the synchronous version of the reference circuit is shown in Figure 11. The steps in the de-synchronization flow are as follows: After verifying the HDL description of the synchronous circuit is working properly, a synthesis using Synopsys Design Compiler with a standard cell library provided by the foundry is performed. Afterwards, the memory elements in the design, e.g., latches and/or flip-flops, are grouped as a separate module.

During the placement and routing process, this new module is assigned to a different power supply than the combinational gates. After the separation of logic and memory elements, the final gate level netlist is written, placed and routed in Cadence SoC Encounter. As mentioned, for only sensing the logic gates' current usage, the logic gates and memory elements need to be assigned and connected to different power domains. This is realized by creating multiple supply domains in SoC Encounter.

In the proposed flow, two power domains, PDLOGIC and PDREGS are created and assigned to different power nets, VDDLOGIC and VDDREG, respectively. Both power domains are assigned a common ground node, i.e., VSS because no

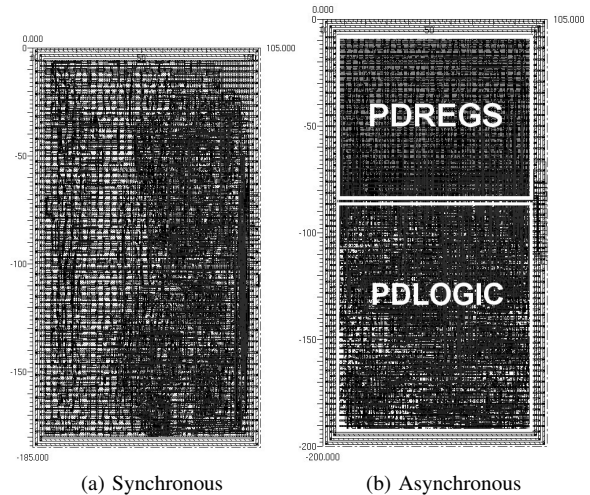


Figure 12. Visual comparison of synchronous and asynchronous versions of the reference circuit in terms of routing and placement.

voltage shifting occurs between different power domains in our design. Two versions of the reference circuit, i.e., synchronous and asynchronous, are shown in Figure 12. After the placement and routing, a Graphic Data System (GDSII) file is generated for the routed block.

To estimate the correct size of the parallel pulse generator, SPICE level simulations are run. From the routed block a Verilog netlist is generated, and converted to HSPICE netlist format using custom scripts. This netlist file is instantiated in the HSPICE testbench with the sensing transistor and the AC-Coupled amplifier. Transient simulations are run with real-life input data to find the minimum signal amplitude that may be sensed. By applying this simulated minimum signal value to the AC-coupled amplifier, the width of the required pulse that will operate in parallel to the AC-coupled amplifier is found. According to the simulation results, a minimum detectable signal matched delay circuit is designed. The GDSII file for this design may be created in two ways: It can be either (i) fully custom or (ii) standard ASIC design flow can be applied to generate the layout and the GDSII file. After the design of the matched pulse generator is completed, the AC-coupled amplifier is integrated with the pulse circuitry as in Figure 7. Finally, a GDSII file for the whole completion circuit is generated.

Finally, both the digital and analog (completion detection) GDSII files combined, placed and routed in SoC Encounter and a final layout GDSII is generated for the tape-out.

C. Area Overhead

The area overhead is due to de-synchronization and the external completion detection circuit. The asynchronous version of the reference design, which is basically the memory and logic elements separated, is slightly bigger than the synchronous version. The total area of the circuits

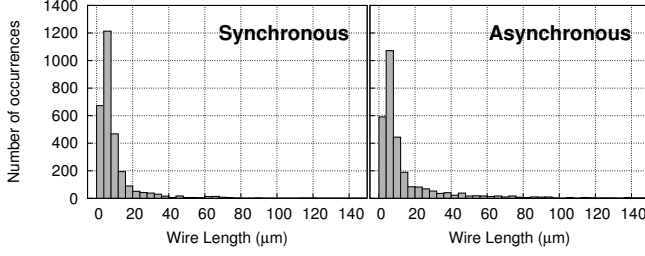


Figure 13. Routing capacitance distribution for synchronous and asynchronous implementations of the reference circuit.

are $19425 \mu m^2$ and $21000 \mu m^2$, for the synchronous and asynchronous versions, respectively. This increase (8.2%) in area of the core is necessary since a single row is required for dividing different power domains, and there is routing overhead due to sub-optimal placement of the memory elements in the circuit.

Area overhead due to the completion detection circuit is 5.4% of the reference design. Area overhead is mostly due to the coupling capacitance of the AC-coupled amplifier. Although in this specific implementation we used a Metal-Insulator-Metal (MIM) capacitor as the coupling capacitor, by using a denser capacitor type such as MOSFET capacitor (MOSCAP), area overhead of the completion detection circuitry may be reduced. Total area overhead due to de-synchronization and the completion detection circuit is 13.6% of the reference design.

D. Comparison of Routing Capacitance for Synchronous and Asynchronous Implementations

We investigated the effects of power domain separation on the routing of wires in the reference circuit. Figure 13 shows our findings. These values are extracted from the post placement and routing of the circuit in Cadence SoC Encounter. When the wire-length distribution of two implementations is compared, it is found that the increase in wire lengths for the asynchronous case (with two power domains) is not significant. The maximum wire-length in the whole distribution has increased as expected, but overall distribution of the wire-lengths is similar with the same mean value. This shows that de-synchronization and separation of the power-domains do not incur significant routing overhead.

E. Possible Design Issues

One of the problems that may occur due to the separation of the power domains and sub-optimal placement of the leaf cells is routing congestion. With our reference design we did not notice any congestion in routing and placement density of the leaf cells were similar for both synchronous and de-synchronized designs. Moreover, because of the very low operating speeds, worse-than-optimal timing due to register placement is not an issue in the sub-threshold regime.

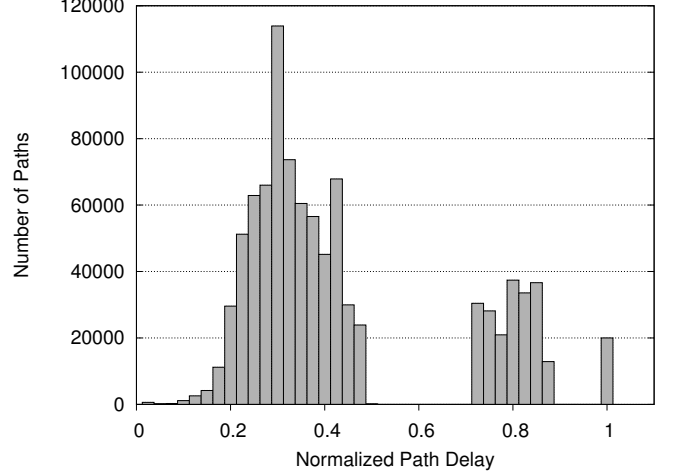


Figure 14. Normalized timing paths of the reference circuit. All path values are normalized to the critical path delay.

Another question that may arise is the performance of the completion detection circuitry for data samples that cause minimum amount of switching but takes a long time to process. For guaranteeing 100% correct operation the circuit may be analyzed for such data samples statistically but due to the multiple feedbacks in the designed circuit and dependence on multiple previous samples, this approach is not viable. However, the designed completion detection circuitry generated correct pulses during operation (both simulations and measurement) even for data samples that take as long as the critical path delay and consume minimal amount of dynamic current.

V. SIMULATION AND MEASUREMENT RESULTS

The energy dissipation and speed improvement results of the self-timed cardiac event detector are obtained by simulations. For verification of the proposed flow and current-sensing completion detection system, the self-timed cardiac event detector is fabricated in a 65 nm standard CMOS process.

A. Simulation Results

In a complex circuit such as the cardiac event detector, there may be many paths which have delays equal or close to the critical path delay of the circuit. It may be argued that, current sensing completion detection may not be as effective as the case where there is a single dominating critical path. Figure 14 shows the normalized path delay distribution of the reference design for all the paths in the circuit. All path delays are normalized to the critical path delay of the circuit. In the reference design, there are more than 20,000 paths that are close to the critical path value. Therefore, to see the possible gain in asynchronous operation, processing time of the circuit while processing real data needs to be investigated.

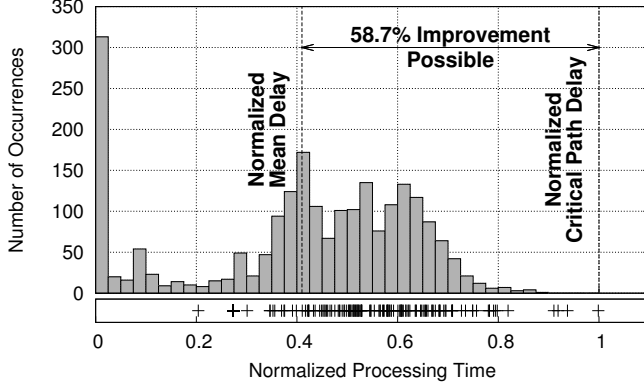


Figure 15. Normalized processing time histogram of the event detector. All processing time values are normalized to the critical path delay.

The histogram in Figure 15 shows the normalized processing time of the circuit while processing real data. The data presented in the histograms is obtained by processing the power waveforms generated by Synopsys PrimeTime for 2200 data samples. All the processing time values are normalized to the critical path delay as in the previous case. The distribution shows many small path delay values. These are due to repeated processing of the same data or due to the periods where the data at the input of the reference design does not change. Based on the processing times, a speed improvement of 58.7 % is possible with the applied real-life data set while operating asynchronously with the completion detection circuit.

Below the histogram, HSPICE simulation results of the completion detection circuit for 200 data samples is shown. According to the results of HSPICE simulation, completion detection circuit using the presented flow results in 52.58 % throughput improvement compared to the synchronous case while operating with a supply voltage of 0.35 V. In our implementation, pulse-width generated by minimum detectable current signal is 20.7 % of the critical path.

The improvement in the throughput results in moving of the EMV. This improvement is calculated by setting μ_d in equation (5) to 0.47, i.e., $1 - 0.53$. The change in EMV and reduction in energy dissipation resulting due to asynchronous operation is presented in Table I. By trading the throughput improvement and moving to a lower EMV, energy dissipation of the same circuit is reduced by 16.8 %. Energy dissipation of the completion detection circuit is 18.36 fJ, which is 2.3 % of the total energy dissipation in asynchronous mode.

B. Silicon Implementation

Both self-timed and synchronous versions of the cardiac event detector are fabricated in a 65 nm standard CMOS process. A chip micrograph is shown in Figure 16, where the event detectors are accommodated on a multi-project

Table I
ENERGY MINIMUM OPERATING VOLTAGE AND RESPECTIVE MINIMUM ENERGY DISSIPATION OF THE REFERENCE CIRCUIT IMPLEMENTED IN A COMMERCIAL 65 NM PROCESS FOR DIFFERENT OPERATION MODES.

Operation	EMV (mV)	Energy (fJ)
Synchronous	330.6	973.4
Asynchronous	296.1	809.7

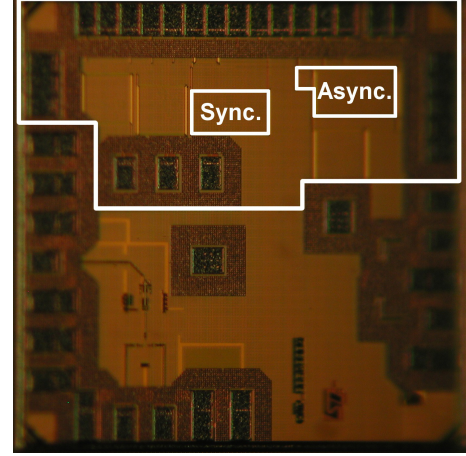


Figure 16. Chip micrograph. Designed circuits were implemented on a multi-project die. Periphery of the sub-threshold blocks are drawn, synchronous and asynchronous designs are emphasized.

die. The synchronous and self-timed versions of the cardiac event detector are highlighted in the figure.

C. Preliminary Measurements

Preliminary measurements for verifying the functionality of the implemented cardiac event detectors as well as the completion detection circuitry have been performed. The reference design, i.e, cardiac event detector, and the completion detection circuitry are operational down to 250 mV.

Figure 17 shows the inverted version of the completion

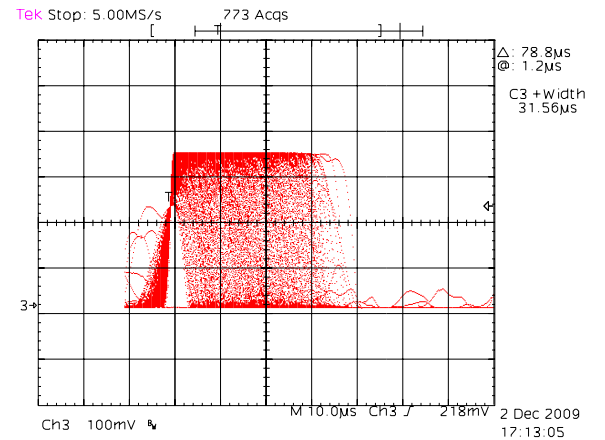


Figure 17. Measured inverted pulses generated by the completion detection circuitry.

detection pulses generated by the completion detection circuitry while operating at a supply voltage of 300 mV. As it is observed from the figure, the width of the generated pulses vary according to the current consumed by the self-timed cardiac event detector. Output signals from the chip are up-converted by on-chip level converters.

VI. CONCLUSIONS

In this manuscript the design of self-timed, energy-minimum circuits operating in the sub- V_T domain are presented. An event detector for cardiac pacemaker applications is used as a reference design. A de-synchronization flow for implementing a sub- V_T current sensing completion detection system has been developed. The developed flow uses commercial EDA tools for implementation. A fully-decoupled latch controller has been developed for integrating the current sensing completion detection circuitry. The area overhead due to de-synchronization is 8.2% in the core, while area overhead due to the completion detection circuit is 5.4% in a commercial 65 nm digital CMOS process. The improvement in throughput is 52.58% by operating asynchronously. Simulation results indicate that trading the throughput improvement reduces energy dissipation by 16.8%. The energy overhead of the completion detection circuit is 2.3% of the total energy dissipation of the reference circuit. The self-timed event detector has been fabricated and verified to operate down to 250 mV.

REFERENCES

- [1] A. Chandrakasan, S. Sheng, and R. Brodersen, "Low-power CMOS digital design," *Solid-State Circuits, IEEE Journal of*, vol. 27, no. 4, pp. 473–484, 1992.
- [2] E. Vittoz, *Low-Power Electronics Design*. CRC Press LLC, 2004, ch. 16.
- [3] A. Wang and A. Chandrakasan, "A 180-mV subthreshold FFT processor using a minimum energy design methodology," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 1, pp. 310–319, 2005.
- [4] J. P. Kulkarni, K. Kim, and K. Roy, "A 160 mv robust schmitt trigger based subthreshold SRAM," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 10, pp. 2303–2313, Oct. 2007.
- [5] P. Beerel and M. Roncken, "Low Power and Energy Efficient Asynchronous Design," *J. Low Power Electronics*, vol. 3, no. 3, pp. 234–253, 2007.
- [6] O. C. Akgun and Y. Leblebici, "Energy Efficiency Comparison of Asynchronous and Synchronous Circuits Operating in the Sub-Threshold Regime," *J. Low Power Electronics*, vol. 3, no. 3, pp. 320–336, 2008.
- [7] N. Lotze, M. Ortmanns, and Y. Manoli, "A study on self-timed asynchronous subthreshold logic," in *Computer Design, 2007. ICCD 2007. 25th International Conference on*, Oct. 2007, pp. 533–540.
- [8] O. C. Akgun, F. K. Gurkaynak, and Y. Leblebici, "A current sensing completion detection method for asynchronous pipelines operating in the sub-threshold regime," *International Journal of Circuit Theory and Applications*, vol. 37, pp. 203–220, 2009.
- [9] R. Swanson and J. Meindl, "Ion-implanted complementary MOS transistors in low-voltage circuits," *Solid-State Circuits, IEEE Journal of*, vol. 7, no. 2, pp. 146–153, 1972.
- [10] S. Hauck, "Asynchronous design methodologies: An overview," *Proceedings of the IEEE*, vol. 83, no. 1, pp. 69–93, 1995.
- [11] M. Hansen, H. Yalcin, J. Hayes, D. Syst, and I. Kokomo, "Unveiling the ISCAS-85 benchmarks: a case study in reverseengineering," *IEEE Design & Test of Computers*, vol. 16, no. 3, pp. 72–80, 1999.
- [12] J. Sparsø and S. Furber, Eds., *Principles of Asynchronous Circuit Design - A Systems Perspective*. Kluwer Academic Publishers, 2001.
- [13] I. Sutherland, "Micropipelines," *Communications of the ACM*, vol. 32, no. 6, Jul. 1989.
- [14] J. Rodrigues, L. Olsson, T. Sörnmo, and V. Öwall, "Digital implementation of a wavelet-based event detector for cardiac pacemakers," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 12, pp. 2686–2698, Dec. 2005.
- [15] M. E. Dean, D. L. Dill, and M. Horowitz, "Self-timed logic using current-sensing completion detection (CSCD)," in *Computer Design: VLSI in Computers and Processors, 1991. ICCD '91. Proceedings., 1991 IEEE International Conference on*, Oct. 1991, pp. 187–191.
- [16] E. Grass, R. Morling, and I. Kale, "Activity-Monitoring Completion-Detection (AMCD): a new single rail approach to achieve self-timing," in *Advanced Research in Asynchronous Circuits and Systems, 1996. Proceedings., Second International Symposium on*, 1996, pp. 143–149.
- [17] H. Lampinen and O. Vainio, "Dynamically biased current sensor for current-sensing completion detection," in *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, vol. 4, May 2001, pp. 394–397.
- [18] S. Furber and P. Day, "Four-phase micropipeline latch control circuits," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 4, no. 2, pp. 247–253, 1996.
- [19] A. M. G. Peeters, "Single rail handshake circuits," Ph.D. dissertation, Eindhoven University of Technology, 1996.
- [20] R. Kol and R. Ginosar, "A doubly-latched asynchronous pipeline," in *Proc. International Conf. Computer Design (ICCD)*, 1997, pp. 706–712.
- [21] J. Cortadella, A. Kondratyev, L. Lavagno, and C. Sotiriou, "Desynchronization: Synthesis of asynchronous circuits from synchronous specifications," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 10, pp. 1904–1921, 2006.