# Minimum Energy Sub-Threshold Self-Timed Circuits Using Current Sensing Completion Detection

Journal:	IET Computers & Digital Techniques	
Manuscript ID:	Draft	
Manuscript Type:	Research Paper	
Date Submitted by the Author:	n/a	
Complete List of Authors:	Akgun, Omer Can; MystIC; Lund University, Electrical and Information Technology Rodrigues, Joachim; Lund University, Electrical and Information Technology Sparsoe, J.; Technical University of Denmark, Informatics and Mathematical Modelling - Computer Science and Engineering	
Keyword:	ASYNCHRONOUS CIRCUITS, VLSI, APPLICATION SPECIFIC INTEGRATED CIRCUITS, CMOS INTEGRATED CIRCUITS, DIGITAL CIRCUITS	
	·	



# Minimum Energy Sub-Threshold Self-Timed Circuits Using Current Sensing Completion Detection

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

Abstract-This paper addresses the design of self-timed energy-minimum circuits, operating in the sub-V<sub>T</sub> domain. The paper presents a generic implementation template using bundled-data circuitry and current sensing completion detection (CSCD). To support this, a fully-decoupled latch controller has been developed, which integrates with the current sensing circuitry. Different configurations in which the latch controller can be used are highlighted. The paper also outlines a corresponding design flow briefly, which is based on contemporary synchronous EDA tools, and which transforms a synchronous design, into a corresponding self-timed circuit. Different use cases for the CSCD system are examined. The design flow and the current-sensing technique is validated by the implementation of a self-timed version of a wavelet based event detector for cardiac pacemaker applications in a standard 65 nm CMOS process. The chip has been fabricated and verified to operate down to 250 mV. 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.

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

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 currents 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 circuits 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. Since their performance is determined by actual case (rather than worst case) latencies, they may provide a higher throughput compared to the synchronous counterparts. However, if a gain in throughput is not utilized, 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 and in [6] it was shown that average case performance property allows asynchronous sub-threshold circuits to work with a higher energy efficiency.

The use of minimum energy circuits in energy constrained applications, e.g., sensor networks [7] or biomedical applications, is of greatest interest. A typical application scenario could be a sensor network that is measuring various conditions, triggered by an event or by a fixed time-interval. The goal of these sensor networks is to perform measurements (no time-constraints), process and eventually transmit the data by dissipating as little energy as possible, since the

)

systems are supplied by energy sources which may utilize a continuous but weak (energy harvesting) [8], [9] or limited (batteries) energy source. After the requested operations are complete, the system is put in sleep mode where only a wake-up circuitry is still active.

There is a growing interest in (wireless) sensor networks possibly powered using energy harvesting techniques and there are many challenges in this field. One set of challenges relate to the design of data-processing circuits which can be used to implement these systems efficiently [10]. Focus here is on minimizing energy dissipation and on being robust towards variations in the supply voltage. The circuits are characterized by periods of activity and long periods of standby in-between. The circuit techniques presented in this paper represents a match to these challenges. Our primary focus is on sub-threshold operation in order to achieve minimum energy but the use of asynchronous techniques also brings some robustness towards variations in the supply voltage as well.

In this work a cardiac event detector was chosen as a reference design since a synchronous counterpart was already fabricated in 65 nm low-leakage high-threshold (LL-HVT) CMOS technology [11]. Consequently, we will be able to compare throughput and energy dissipation of two competitive design strategies by measurement (not scope of this study).

The contributions of this paper are as follows: Use cases and possible applications of a current sensing completion detection (CSCD) system are presented. A new fully decoupled latch controller that integrates the completion detection system to an asynchronous circuit implementation is designed and analyzed. Different operating modes for the controller are explained. A de-synchronization flow employing commercial EDA tools is shown. Finally, using the developed de-synchronization flow and the latch controller, a selftimed version of the sub-threshold event detector for cardiac pacemaker applications is 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. Section III gives an overview of the CSCD system used in our self-timed circuit implementation with possible use cases. Section IV gives details of the CSCD implementation and explains the designed latch controller in detail. In Section V we present the de-synchronization flow briefly and in Section VI simulation results and preliminary measurements are given. Finally, in Section VII conclusions are drawn.

### II. MOTIVATION AND BACKGROUND

The section gives a brief overview on sub- $V_T$  domain, before introducing asynchronous operation, and an analysis of energy reduction enabled by asynchronous operation.

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

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

where  $E_{dynamic}$  is the total dynamic energy consumed while charging the load capacitance  $C_{load}$ , with a switching probability of  $\alpha$ . 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 [12]. 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 [12] it was proven that the minimum operating voltage for obtaining an absolute gain of more than 1 from a simple inverter and for guaranteeing bistability 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

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

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

where  $\mu$  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 energyminimum 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.

There are several reasons why asynchronous circuits are not as common as synchronous designs. First of all asynchronous circuits are more difficult to design then 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 [13] and switching activity must be properly ordered not to cause wrong data propagation. Although this is not a 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.

In this paper, we concentrate on a subset of asynchronous circuits that are based on asynchronous micro-pipelines first introduced by Sutherland [15]. The asynchronous circuit model that we will use in the remainder of this paper is a 4-phase bundled-data circuit shown in Figure 1 and is taken from [14]. In this type of asynchronous circuits, consecutive pipeline stages are separated using latches or



Figure 1. 4-phase bundled data pipeline (After [14]).

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 process new data, the AFSM will acknowledge this request by using the *ack* line, which in turn enables 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 1a. 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 dissipation and throughput degradation will occur. By employing circuits with completion detection capabilities and by realizing average-case operation speed, leakage energy component, which is inversely proportional to the operating speed of the circuit, may be reduced.

There are many approaches to design asynchronous circuits with 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 [16]. Our solution for this problem is to employ a supplemented



Figure 2. 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.

completion detection circuit for sub-threshold operation. In this work we focus on the implementation details and use cases of single-rail circuits that operate with current sensing completion detection. The implemented completion detection system was first introduced in [17] and an overview of this solution with possible use cases 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

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

and

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

for the synchronous and asynchronous cases, respectively. In the equations,  $C_{inv}$  is the total switched capacitance of an inverter,  $\mu_e$  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  $\mu_d$  is a parameter which denotes the average computation time of the asynchronous circuit



Figure 3. 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.

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 2 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 is shown in the plot of energy-minimum supply voltage values and their corresponding throughput values. In Figure 3 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 switching energy. The throughput worsens due to lower operating voltage but it is negligible in asynchronous



Figure 4. General block diagram of the completion detection system. The system consist of an asynchronous finite state machine, completion detection circuitry and a sensor transistor.

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

# III. CURRENT SENSING COMPLETION DETECTION SYSTEM

In this section we provide an overview of the completion detection system that was first presented in [17] by Akgun et.al. This method may be applied to either the complete design using one completion detection circuit, or the individual stages using dedicated completion detection circuitry for each stage. In this paper we present the former as an implementation example since the implementation is more generic (presents the basic idea) and allows us to prove the concept of current sensing completion detection in the subthreshold regime on a complex design.

#### A. CSCD System

In general, single-rail asynchronous circuits operate by delaying the control signals by an amount equal to the critical path delay [14]. 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 a circuit, we need to reduce the time the circuit spends *leaking*, both saving leakage energy 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 domain. As long as the combinational circuitry is *switching* there will be dynamic power consumption in the circuit, which is detectable through the supply current  $I_{VDD}$  of the circuit. There are several implementations of completion detection circuits that use current sensing in the literature [18]–[20]. 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  $\mu$ A-to-mA range. In this work we apply our de-synchronization flow to implement a current sensing completion detection system as presented in Figure 4. 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. Use Cases for CSCD

In this section different use cases for a current sensing completion detection system which realizes average case performance are presented. Use cases are examined for different environments and architectures:



Figure 5. Minimum energy processing scenario.

1) Minimum Energy per Sample Processing: As shown in Section II-C, self-timed operation realizes minimum energy per operation. However, the operating speed is dictated by the EMV and minimum energy per sample operation is only realizable within an asynchronous frame. Such an implementation is shown in Figure 5. In this implementation the core is governed by a single CSCD controller. Whole system is triggered by a single signal and the current drawn by the entire circuit is monitored. Implemented system works at EMV with average-case throughput, which is the most energy efficient.

2) Asynchronous Operation in a Synchronous Environment: For using the CSCD in a synchronous environment, the digital system must be surrounded by FIFOs such that asynchronous operation speed can be matched to that of the synchronous one. One such example was presented in [21]. In this work, supply voltage was scaled so that justin-time operation was realized. This cannot be applied to sub-threshold circuits for lower energy operation because as shown in Figure 2, operating at any other point other than the EMV results in non-minimal energy dissipation. Thus, we propose a system, similar to that of in [21] as shown in Figure 6.

In this proposed system asynchronous circuit core is surrounded with FIFOs to be able to match the operating speed of the synchronous environment. As mentioned, for



Figure 6. Self-timed operation embedded in a synchronous environment.

energy minimum operation the system needs to be operated at EMV. While working in a synchronous environment as in our example, energy may be saved by employing power shutdown. There are multiple scenarios that needs to examined:

- Synchronous environment slower than the self-timed domain: In such a case, if the speed difference between the environments is high, only 1-sample FIFOs can be used for synchronization and the self-timed system can be shutdown as soon as the operation is complete, thus saving leakage energy. If the energy savings due to power-shutdown are not substantial and power-shutdown overhead is dominating the savings, the depth of the FIFOs can be increased and the number of power-shutdowns may be reduced significantly, i. e., by a factor of N for N-sample FIFOs.
- Synchronous environment speed is similar to that of the self-timed domain: In this case N-sample FIFOs are needed around the self-timed core. The FIFOs have two purposes: i) Guaranteeing that no data is missed if asynchronous core processes multiple worstcase samples in succession, and ii) allow the self-timed core to shutdown after processing a certain number of samples. By operating in such a manner, the cost of the power-shutdown on the overall energy figure is reduced, i. e., less power-shutdowns as explained in the previous scenario. In a configuration such as shown in Figure 6, a supervisor block that monitors the fullness of the FIFOs is needed for guaranteeing correct operation. In a situation where the FIFOs are full, supervisor block should generate a signal to increase the supplied voltage, which will result in a higher processing speed, and thus moving away from EMV for guaranteeing that the self-timed domain do not stall the rest of the system. Once the FIFOs get less occupied, supply

# voltage should be returned to EMV.

3) Reactive systems: In this scenario the system is active for a brief time and shut-down for a much longer time. Such an application is the sensor networks, where the system performs a sequence of operations and sleep for a much longer period than the active time, and these applications usually do not have hard real-time (worst-case) speed requirements. The self-timed system works at EMV and goes to sleep again. The benefits of employing a self-timed implementation in this case is operating at a lower EMV than the synchronous counterpart, hence lower energy dissipation. It should also be noted that all the energy dissipation overhead elements, such as power-shutdown, FIFOs (if needed), etc. also exist in a synchronous implementation for such a use case.

# IV. CSCD IMPLEMENTATION

This section describes the implementation details of the sub-threshold current sensing completion detection system. Building blocks of the system are explained with emphasis on the designed latch controller.

Figure 4 is a general block diagram showing how the completion detection circuit is added in an asynchronous pipeline. The detailed design will be explained later. As seen, a *stage* is a latch/register followed by some logic. This is opposite to the usual convention where a pipeline stage is defined as some logic followed by a latch/register. The use of an input register is necessary because the latch controller AFSM, in addition to generating clock-ticks to the latch, also generates ticks which starts the current sensing completion detection. Moreover, the latch controller AFSM and the following completion detection circuit is implemented in an integrated fashion. It must be emphasized that the (non-standard) use of input latches/registers is mostly an aesthetic issue; from a functional point of view it makes no difference to the operation of the system.

The pipeline in Figure 4 can be implemented using latches as well as flip-flops. The designed latch controller AFSM generates *ticks* which can control an edge-triggered register directly, as well. The ticks are pulses and to control a latch it needs to be ensured that the pulse has a sufficient width. This can be done using a special pulsing circuit or as explained later in the design of the latch controller, one can use the "fixed pulse generator" shown in Figure 9.

Figure 4 showed a fragment of a simple linear pipeline. It must be emphasized that arbitrary circuit structures using conventional data-driven asynchronous handshakecomponents, e.g. latch, logic, join, fork, merge, mux, demux, [14] can be built. A design constraint to keep in mind is that when smaller combinatorial blocks are composed into larger combinatorial blocks (without latches in-between), the entire combinatorial block should have only one single current sensing circuit.

Finally, it must be noted that the latch controller AFSM has the property that a ring with only one edge-triggered

pipeline stage can iterate. This can be exploited to transform a synchronous circuit into self-timed circuit using a global but a-periodic clock which is derived using current sensing completion detection in the combinatorial circuit. This represents an interesting and simple form of de-synchronization and in the next section we present a corresponding design flow and a prototype circuit which has been designed in this way.

## A. Current Sensing

In any CSCD implementation, 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- $V_{\rm T}$  pMOS transistor, see Figure 4 [17]. In this implementation, the current signal is sensed by the diode connected low- $V_{\rm T}$  pMOS transistor and is converted to a voltage signal.

## B. Latch Controller

A conventional bundled data pipeline, see Figure 1a, 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 [22] 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 [14], [23]. 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 Ain signal to clock the registers. Moreover, the use of current sensing completion detection leads to a situation where a signal event, Ain+, 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 Ain+ 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



Figure 7. Signal transition graph for the designed controller.

Rout+ includes a delay which matches the subsequent combinatorial circuit.

Figure 4 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 7. 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 (Ain+) and which then produces a variable width pulse (T- - > 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 (Ain+) 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) Rin connected to Ain through an inverter and Rout connected to Aout, and (ii) Rout connected to Rin and Aout connected to Ain (i.e., the output port connected to the input port). These modes and use cases are explained in the next section.

# C. Operation Modes for the Latch Controller

The controller described in the previous section can be employed for multiple operation modes:

- *Standalone controller (Figure 8a):* A single CSCD controller is used to control the entire core. In this implementation, due to the specific CSCD implementation presented in this work, only the current dissipation of the combinational gates is monitored (see Figure 5).
- *Pipelined Operation:* CSCD controllers can be connected in pipelined fashion as any other



Figure 8. Usage cases for the controller circuitry.

asynchronous controller. Depending on the size of the circuit two different implementations are possible: i) Pipelined systems: This implementation is similar to the standalone controller. Combinational current consumption of the systems are sensed by individual controllers. ii) Smaller pipeline stages separated by memory elements: As explained previously, combinational current consumption of the pipeline stage is sensed, and the memory elements at the input of the pipeline stage are controlled. From a system's perspective, both implementations are the same, see Figure 8b.

• Self-Oscillating Configuration - 1: The controller is connected as shown in Figure 8c, and after the reset phase, the circuit will start self oscillating. This configuration may be used in a system which realizes minimum energy per sample operation. By self-oscillating, as soon as the computation is completed, new processed data will be available to



Figure 9. AFSM including the pulse generating completion detection circuitry.



Figure 10. AC-coupled amplifier.

the environment and the system can request and begin processing the following sample immediately. It should also be noted that only one controller is required to realize self oscillation, unlike other examples in the literature which require multiple controllers.

• Self-Oscillating Configuration - 2: Like the previous mode, this mode (Figure 8d), also allows self-oscillating operation. In case one of the data sides, either input or output, require different operating speed, then the loop may be cut and the system becomes controlled only from one side, operating at the maximum speed possible at the uncontrolled side.

#### D. Completion Signal Generation

The completion detection circuit used to realize the asynchronous cardiac event detector is presented in Figure 9. Unlike the current sensing implementation presented in [17], we removed the variable pulse-width generator, which was possible due to the increased complexity and hence higher current drawn by our reference circuit. The sensed signal is strong enough to drive the AC-coupled amplifier to voltage values which are just below the supply voltage. Consequently, it is possible to shape this analog signal such that is 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 10. 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 is adjusted 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 is connected in parallel to the AC-Coupled amplifier and the shaping circuitry, which is connected to the output of the amplifier and consist of an inverter, is implemented as well. Fixed pulse generator is necessary to generate a pulse signal for cases where no combinational switching occurs, or for minimal amount of switching occurring in the circuit that cannot be amplified enough by the AC-coupled amplifier for being converted to a logic level change. 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 and the width of the pulse generated is set equal to the minimum pulse width generated by the AC-coupled amplifier.

# V. ASYNCHRONOUS SYSTEM IMPLEMENTATION EXAMPLE

This section presents the reference design and employed self-timed cardiac event detector implementation methodology briefly. Design flow is briefly introduced in this section and more details about the flow can be found in [24].

## A. Digital Event Detector for Cardiac Pacemakers

As a reference circuit, we use a CMOS implementation of a digital 3-scaled wavelet-based filter in combination with so-called hypothesis testing [25] for detecting the R-wave in a cardiac pacemaker. 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. The event detector consists of 727 registers and 4200 NAND2 equivalent logic cells.

#### B. 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-toasynchronous conversion in the literature. In [26] 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 [27] is similar to the DLAP approach. A fully automated synthesis flow that does not change the structure



Figure 11. Separated current waveforms of the combinational gates and registers of the reference circuit.





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 employing, 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.

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, 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 wave-



Figure 13. Design flow for the test chip

forms for the combinational logic and the registers are shown in Figure 11. 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 12. Positive edge triggered registers are driven by the Ain signal of the completion detection circuit shown in Figure 9. The separation process is automated and incurs little overhead in terms of routing and area.

The main advantage of the employed flow (Figure 13) is that stable and well known EDA-tools/flows for synchronous circuit design are used. Detailed information about the employed design flow and the tools is given in [24].

# VI. 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 currentsensing completion detection system, the self-timed cardiac



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



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

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 20000 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.

The histogram in Figure 15 shows the normalized processing time of the circuit while processing real data. The

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

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  $\mu_d$  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 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 detection pulses generated by the completion detection circuitry while operating at a supply voltage of 300 mV. As



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.



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

it is observed from the figure, the width of the generated pulses vary according to the current consumed by the selftimed cardiac event detector. Output signals from the chip are up-converted by on-chip level converters.

#### VII. CONCLUSIONS

In this manuscript the design of self-timed, energyminimum circuits operating in the sub- $V_T$  domain are presented. Different use cases for a current sensing completion detection system are examined. 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 is briefly introduced. A fully-decoupled latch controller has been developed for integrating with the current sensing completion detection circuitry. Different configurations for the latch controller together with the CSCD circuitry are presented.

The area overhead due to de-synchronization is 8.2%

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

- 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] H. Gharavi and S. Kumar, "Special issue on sensor networks and applications," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1151 – 1153, aug. 2003.
- [8] B. Warneke, M. Scott, B. Leibowitz, L. Zhou, C. Bellew, J. Chediak, J. Kahn, B. Boser, and K. Pister, "An autonomous 16 mm3 solar-powered node for distributed wireless sensor networks," in *Proc. IEEE Sensors*, vol. 1, no. 2, 2002, pp. 1510–1515.
- [9] R. Amirtharajah, J. Wenck, J. Collier, J. Siebert, and B. Zhou, "Circuits for energy harvesting sensor signal processing," in *Design Automation Conference*, 2006 43rd ACM/IEEE, 2006, pp. 639 –644.
- [10] X. Zhang, H. Jiang, L. Zhang, C. Zhang, Z. Wang, and X. Chen, "An energy-efficient asic for wireless body sensor networks in medical applications," *Biomedical Circuits and Systems, IEEE Transactions on*, vol. 4, no. 1, pp. 11–18, feb. 2010.
- [11] J. Rodrigues, O. C. Akgun, and V. Owall, "A <1 nJ Sub-VT Cardiac Event Detector in 65 nm LL-HVT CMOS," in Proceedings of 18th IEEE/IFIP International Conference on VLSI and System-on-Chip (VLSI-SOC), in press.

- [12] 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.
- [13] S. Hauck, "Asynchronous design methodologies: An overview," *Proceedings of the IEEE*, vol. 83, no. 1, pp. 69–93, 1995.
- [14] J. Sparsø and S. Furber, Eds., Principles of Asynchronous Circuit Design - A Systems Perspective. Kluwer Academic Publishers, 2001.
- [15] I. Sutherland, "Micropipelines," *Communications of the ACM*, vol. 32, no. 6, Jul. 1989.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] E. Grass, R. Morling, and I. Kale, "Activity-Monitoring Completion-Detection (AMCD): a new single railapproach to achieve self-timing," in Advanced Research in Asynchronous Circuits and Systems, 1996. Proceedings., Second International Symposium on, 1996, pp. 143–149.
- [20] 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.
- [21] L. Nielsen, C. Niessen, J. Sparsø, and K. Van Berkel, "Lowpower operation using self-timed circuits and adaptive scalingof the supply voltage," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 2, no. 4, pp. 391–397, 1994.
- [22] 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.
- [23] A. M. G. Peeters, "Single rail handshake circuits," Ph.D. dissertation, Eindhoven University of Technology, 1996.
- [24] O. C. Akgun, J. Rodrigues, and J. Sparsø, "Minimum-energy sub-threshold self-timed circuits: Design methodology and a case study," in *Proceedings of 16th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)* 2010, 2010, pp. 41–51.
- [25] 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.

- [26] R. Kol and R. Ginosar, "A doubly-latched asynchronous pipeline," in *Proc. International Conf. Computer Design* (*ICCD*), 1997, pp. 706–712.
- [27] 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.