A Current Sensing Completion Detection Method for Asynchronous Pipelines Operating in the Sub-threshold Regime

Omer Can Akgun, Frank K. Gürkaynak, Yusuf Leblebici
Swiss Federal Institute of Technology (EPFL)
Microelectronic Systems Laboratory (LSM)
Lausanne, CH-1015, Switzerland
E-mail:{omercan.akgun, frank.gurkaynak, yusuf.leblebici}@epfl.ch

Abstract

Digital circuits operating in the sub-threshold regime are able to perform minimum energy operation at a given delay. In the sub-threshold regime the circuit delay, and hence, the leakage energy consumption depend on the supply voltage exponentially. By reducing the idle time of the circuit, the energy-minimum supply voltage can be reduced, resulting in lower energy consumption.

This paper first presents an energy model for comparing synchronous and asynchronous sub-threshold operation. After the presentation of the model, design and simulation results of a novel current sensing based completion detection circuit that is applicable to coarse-grained single-rail asynchronous pipelines working in the sub-threshold regime is shown. The system works by sensing the changes in the current consumption of static CMOS combinational blocks and generates completion signals delayed in proportion to the computation taking place.

The method is applied as a proof of concept to the operation of a static CMOS 16-bit adder logic block that have been designed using conventional EDA tools for a $0.18\,\mu m$ CMOS process. When operating with a supply voltage of 220 mV, speed improvements of up to 19.3% have been observed in the operation of the circuit. If the system is operated at the asynchronous energy-minimum supply voltage, an energy consumption reduction of 17% can be realized for the same computation load.

I. Introduction

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

In sub-threshold mode of operation, the supply voltage can be scaled aggressively and power dissipation can be decreased significantly. Sub-threshold operation of static CMOS logic has been analyzed using the EKV model in [2]. In this analysis, it was shown that static CMOS logic can 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. Even so, 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 the leakage energy of the circuit increases exponentially with decreasing supply voltage as explained in Section II. At the same time the dynamic energy consumption of a circuit operating in the sub-threshold regime decreases quadratically with decreasing supply voltage. Therefore for circuits operating in the sub-threshold regime a supply voltage that minimizes the sum of the dynamic and leakage energy can be found. We will call this supply voltage the *energy-minimum operating voltage* throughout the paper. We will show that the key to reduce the energy-minimum operating voltage will be to lower the leakage energy of the circuit.

One solution to reduce the leakage energy is to use asynchronous circuits, which can be designed to work at average case performance. Recently in [5] asynchronous circuits were studied from a low power and energy efficient operation perspective. At least in theory, average case performance property would allow asynchronous sub-threshold circuits to work at a higher energy efficiency operating point. There are many approaches to design asynchronous circuits, some of which have inherent completion detection capabilities. In this paper we investigate a specific subset of asynchronous circuits that are obtained around coarse grained combinational blocks. Traditionally such circuits require fixed delay lines to account for the worst case critical path [6].

These fixed delay lines can be replaced by circuits that monitor the dynamic current consumption of individual blocks and generate appropriate completion detection signals that augment the asynchronous control circuits as presented in [7]–[9]. The circuits presented in these earlier papers are designed for normal operating regime with currents in the μA-to-mA range. In this paper, we propose a current sensing based completion detection method that is suitable for sub-threshold operation and can detect current variations in the pA-to-nA range. The proposed methodology is demonstrated on the operation of a 16-bit adder where improvements in the average throughput of 19.3% and energy consumption reduction of 17% can be shown.

The remainder of this paper is organized as follows. Section II provides background information and motivation for this work and presents an analytical model for minimum energy operation of asynchronous circuits in general. In Section III, our sub-threshold completion detection solution is explained, and in Section IV, the circuit-level implementation is presented. Simulation results are given in Section V, and finally conclusions are drawn in Section VI.

II. Motivation and Background

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

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

where $E_{dynamic}$ is the total dynamic energy consumed while charging the load capacitance C_{load}, with a switching probability of α. When the circuit is not switching there is some leakage energy $E_{leakage}$ that is consumed during the leakage time t_{leak}. In addition, when a switching event occurs, for the duration of the switching time t_{sc}, when both nMOS and pMOS transistors are conducting, some short circuit energy $E_{short-circuit}$ will be consumed . In our energy consumption analysis we will neglect the contribution of the short circuit energy in the sub-threshold regime, as it is known to contribute only a small portion of the overall energy consumption [2].

From (1) it is immediately clear that the energy consumption 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 [10]. 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.

The current equations of MOS transistors operating in the sub-threshold regime are different than those working in the super-threshold regime. For an nMOS transistor, the drain current in the sub-threshold regime is given in [2] by

$$I_{DS} = I_S e^{\frac{V_{GS}-V_T}{nU_t}} \left(1 - e^{\frac{-V_{DS}}{U_t}}\right) \tag{2}$$

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. 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 μ is the mobility of carriers, C_{ox} is the gate oxide capacitance per unit area, U_t is the thermal voltage whose value is $26\,\text{mV}$ at $300\,\text{K}$ and $\frac{W}{L}$ is the aspect ratio of the transistor.

Due to the second term in (2), the drain current is 0 when $V_{DS} = 0$ but reaches its maximum value for a given V_{GS} and saturates with V_{DS} values higher than a few U_t. As it is apparent from (2), 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 consumption 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-B.

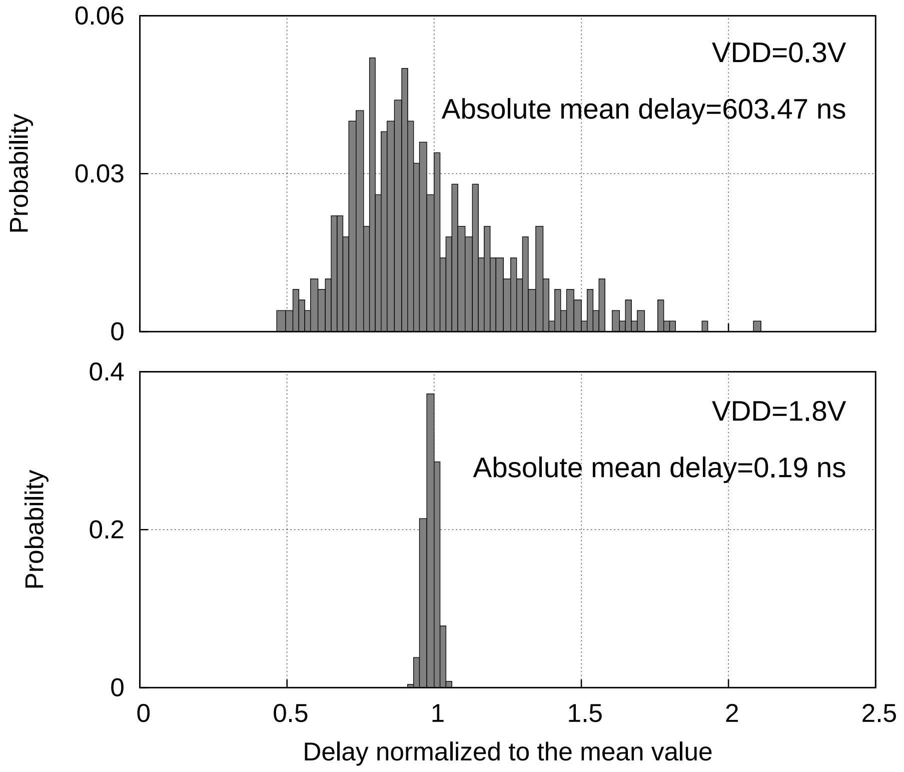


Fig. 1: Comparison of the delay times for a single inverter under V_T variation. The variation of the average delay is gathered from SPICE level Monte Carlo simulations.

Another problem with the synchronous operation in the sub-threshold regime is the required excessive timing margin due to process variations. Because the sub-threshold current depends on the V_T exponentially, the sub-threshold current and parameters that have a first order relationship with the sub-threshold current such as leakage energy and circuit delay show a log-normal distribution under the assumption of normal distribution of V_T mismatch. The log-normal distribution, when compared to the normal distribution, has broader tail, meaning that synchronous circuits operating in the sub-threshold regime needs to be operated with a larger safety margin when compared to super-threshold operation.

The delay variation for a single inverter from a standard cell library in a 0.18μm process is shown in Fig. 1. The data for the delay variation is gathered from 1000 point Monte Carlo simulations. As it can be seen from the figure, the delay variation is minimal at the nominal supply voltage of the process (VDD=1.8V) and has a normal distribution as expected. On the other hand in the sub-threshold operating regime the delay has a log-normal distribution with a long tail on the right (VDD=0.3V). This kind of distribution curve implies that the below average delays deviate slightly from the expected value while above average delays can be as high as several times the mean. Because of this excessive safety margin on the operating speed of the circuit, synchronous circuits operating in the sub-threshold regime consume more leakage energy, moving the energy-minimum operating voltage to higher values, resulting in higher energy consumption.

A. Asynchronous Operation

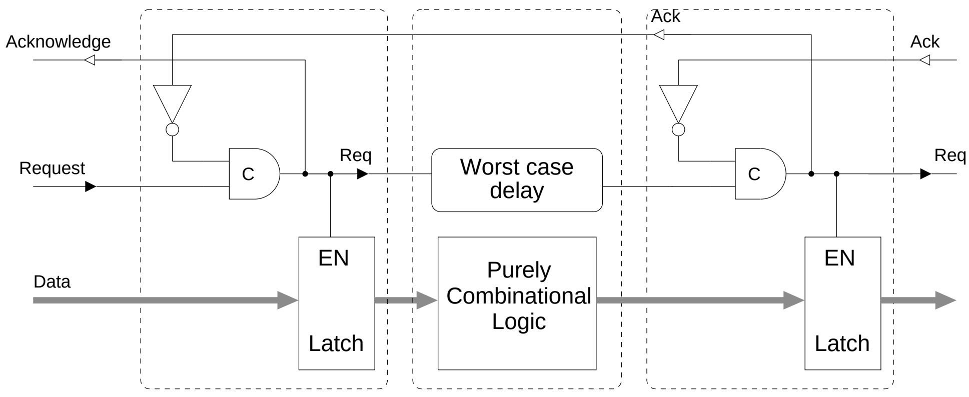


Fig. 2: 4-phase bundled data asynchronous pipeline (after [11]).

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 consumption. Asynchronous circuits rely on handshaking signals to communicate between functional units. Once an asynchronous block has been triggered (by the arrival of new data for example), it will start processing. At the end of the operation, the asynchronous sub-block will signal that it has completed its operation. After this time operation will continue depending on the specific asynchronous protocol that has been used.

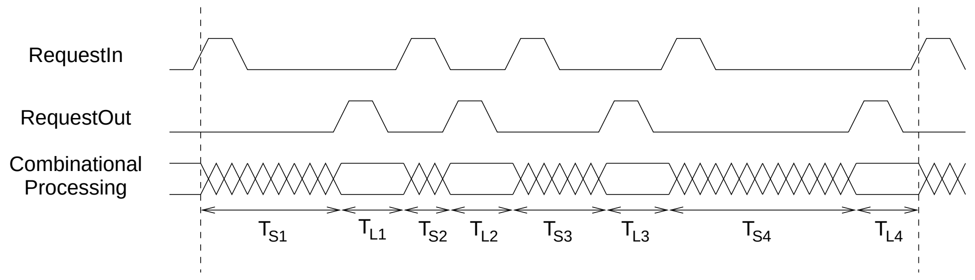
Efficient asynchronous operation requires a reliable method to determine the time required to complete processing of information. There are many methods to realize asynchronous circuits, some of which have inherent completion detection capability such as methods based on *n-out-of-m* coding. In this paper, we will concentrate on a subset of asynchronous circuits that are based on asynchronous micro-pipelines first introduced by Sutherland [12]. The asynchronous circuit model that we will use in the remainder of this paper is a bundled-data circuit shown in Fig. 2 and is taken from [11].

In this type of asynchronous circuits, consecutive pipeline stages are separated using latches 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. This will enable the latch, 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.

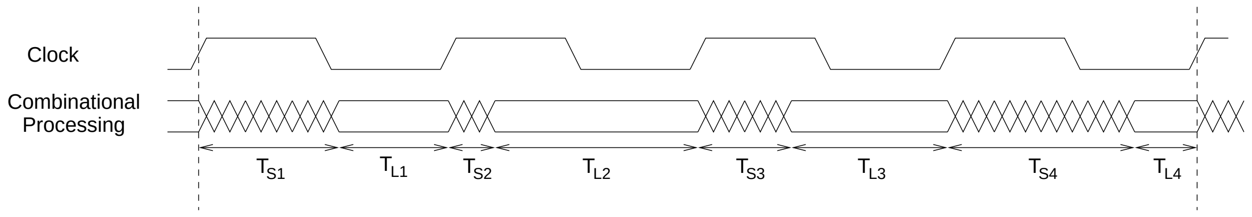
Traditional implementations of this circuit frequently use a *matched delay line* that has been engineered to have a delay that corresponds to the *worst case* delay through the combinational circuit as shown in Fig. 2. There are obvious disadvantages of using a fixed delay element for performance reasons, 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 develop a completion detection circuit for sub-threshold operation. Details of this solution will be given in Section III.

B. Sub-threshold Minimum Energy Operation



(a) Asynchronous Operation



(b) Synchronous Operation

Fig. 3: Timing diagram showing the same logic block operating in synchronous and asynchronous modes. Note that the overall processing time can be significantly reduced by shortening the time lag between switching events T_{Si}.

In this section an energy model for asynchronous systems working in the sub-threshold regime is presented. In the asynchronous systems, the operation of the system is both dictated by the switching and delay properties of the system under investigation and the external request and acknowledge signals. During the development of the model it will be assumed that as soon as the asynchronous block finishes processing the current data, a new data can be applied.

The conceptual operation of an asynchronous block is shown in Fig. 3a. In this example, the circuit is observed for an arbitrary time frame T, where four distinct sets of input data are processed. The time spans where the circuit is purely *leaking* (waiting for handshake completion) are denoted with T_{Li} and the time spans where the circuit is both *switching* (processing input data) and *leaking* are denoted with T_{Si}. In this diagram without loss of generality we assume that as soon as the *RequestOut* signal is lowered *RequestIn* goes high and the stage begins the *switching* phase; so all the purely leakage time spans (T_{Li}s) are fixed, equal, and represent the asynchronous communication overhead.

In the development of the asynchronous system energy model, we assume a total of N operations take place during the arbitrarily long observation time frame T. Each switching time T_{Si} within the time frame can be different than each other depending on the switching properties of the circuit and the applied data to the circuit input.

Elaborating general energy consumption equation (1), the dynamic energy consumption during the ith time span can be expressed as

$$E_{dynamic_i} = e_i C_{tot} V_{DD}^2 \tag{4}$$

where e_i is a scaling parameter that defines the switching property of the circuit for a specific input data transition and C_{tot} is the maximum possible switched capacitance of the circuit. The switching energy scaling parameter e_i is in the range $[0,1]$ and without loss of generality can be expressed as a single value in a random process e. By defining the e as a random process, we can define a mean μ_e for it and using this mean the average dynamic energy for N calculations can be expressed as

$$E_{dynamic} = N\mu_e C_{tot} V_{DD}^2 \tag{5}$$

In (4) and (5), the total capacitance C_{tot} can be normalized in terms of the total inverter capacitance using a capacitance scaling factor $k_{cap-logic}$ as $C_{tot} = k_{cap-logic} C_{inv}$ where C_{inv} is the switched capacitance of an inverter.

Assuming that even during the *switching* time, most of the cells in the circuit are leaking, the leakage energy consumption during the observation period T can be defined as

$$E_{leak} = k_{leak} I_0 V_{DD} T \tag{6}$$

where k_{leak} is the average leakage scaling factor of the circuit, I_0 is the leakage current of a single inverter. From (6) total average leakage current of the circuit can be calculated as $k_{leak} I_0$. In this equation, the average leakage parameter k_{leak} can be obtained from the synthesis results by summing the individual average leakage currents of the digital gates, where average leakage current is the mean of the leakage current for all the combinations of input vectors applied to the logic gate, and normalizing the result to the average leakage current of a single inverter.

Combining (1), (5) and (6), total energy consumption during the monitoring time frame T can be defined as

$$E_T = N\mu_e k_{cap-logic} C_{inv} V_{DD}^2 + k_{leak} I_0 V_{DD} T \tag{7}$$

From Fig. 3a, the total time spent during switching is

$$T_S = \sum_{i=1}^{S} T_{s_i} \tag{8}$$

and based on the switching/timing statistics of the circuit we are analyzing any switching activity (T_{s_i}) in Fig. 3a can be defined as

$$T_{s_i} = d_i k_{crit} T_{sw\_inv} \tag{9}$$

where d_i is a scaling parameter that defines the delay properties of the circuit processing the current data, k_{crit} is a coefficient that defines the critical path delay of the circuit in terms of the inverter delay and T_{sw_inv} is the delay of an inverter. The scaling parameter d_i can take any value in the range $[0,1]$. Similar to process e, if the process d is modeled as a random process with the mean μ_d, the total time spent during switching can be calculated approximately as

$$T_S = N\mu_d k_{crit} T_{sw\_inv} \tag{10}$$

During our observation frame T, N switchings and N handshakes take place, so T can be expressed as

$$T = T_S + T_L = N\mu_d k_{crit} T_{sw\_inv} + N k_{com_{oh}} k_{crit} T_{sw\_inv} \tag{11}$$

where $k_{com_{oh}}$ is a parameter defining the overhead caused by the asynchronous communication in terms of the critical path delay of the purely combinational logic block. The delay of an inverter working in the sub-threshold regime is given in [2] as

$$T_{sw\_inv} = \frac{C_{inv} V_{DD}}{I_0 e^{V_{DD}/(nU_t)}} \tag{12}$$

By introducing (12) into (11), we get the total observation time as

$$T = N k_{crit} \frac{C_{inv} V_{DD}}{I_0 e^{V_{DD}/(nU_t)}} (\mu_d + k_{com_{oh}}) \tag{13}$$

and by introducing (13) into (7), we get the final total energy consumption equation for N switching events as

$$E_T = N C_{inv} V_{DD}^2 \left[\mu_e k_{cap-logic} + k_{crit} k_{leak} (\mu_d + k_{com_{oh}}) e^{-V_{DD}/(nU_t)} \right] \tag{14}$$

By setting $N = 1$ in (14), the average energy consumption per operation can be found. The optimal operating voltage for minimum energy operation can be found by taking the derivative of (14) with respect to V_{DD}, equating the result to 0, and solving for V_{DD}. The energy-minimum operating voltage is given in (15)

$$V_{opt-async} = 2nU_t - nU_t W_{-1} \left[-\frac{2e^2 k_{cap-logic} \mu_e}{k_{crit} k_{leak} (k_{com_{oh}} + \mu_d)} \right] \tag{15}$$

where W_{-1} is the -1 branch of the LambertW function. LambertW function is examined thoroughly in [13]. All the k-parameters in (14) and (15) can be found from the synthesis results of the digital circuit, and the μ-parameters can be found after running switch level (synthesized Verilog) simulations. Hence the total simulation time for characterizing the sub-threshold performance of the circuit can be reduced greatly when compared to the SPICE-level simulation.

A similar modeling approach can be also taken for modeling a synchronous system operating as shown in Fig. 3b. By assuming only one set of data is processed during a clock period ($N = 1$) and the clock period is equal to the critical path of the logic circuit ($T = k_{crit} T_{sw\_inv}$), the energy per operation can be derived from equations (13) and (14) as

$$E_T = C_{inv} V_{DD}^2 \left[\mu_e k_{cap-logic} + k_{crit} k_{leak} e^{-V_{DD}/(nU_t)} \right] \tag{16}$$

As in the asynchronous case, by taking the derivative of (16), equating the result to 0 and solving for V_{DD}, we get the optimum voltage that realizes the minimum energy operation as

$$V_{opt-sync} = 2nU_t - nU_t W_{-1} \left[-\frac{2e^2 k_{cap-logic} \mu_e}{k_{crit} k_{leak}} \right] \tag{17}$$

To be able to compare the energy consumption of the synchronous and asynchronous circuits at their energy-minimum operating points, the V_{DD} parameters in (14) and (16) are replaced by the optimum voltages $V_{opt-async}$ and $V_{opt-sync}$, respectively. The resulting energy equations depend only on the circuit implementation related k-parameters and the switching/delay properties of the circuit.

Fig. 4 shows the energy profile of our randomly generated test circuit at a switching/delay mean of 0.1 for both asynchronous and synchronous operation. The k-parameters of the test circuit were chosen such that the circuit has energy consumption equivalent to 1000 inverter gates with a drive capability of

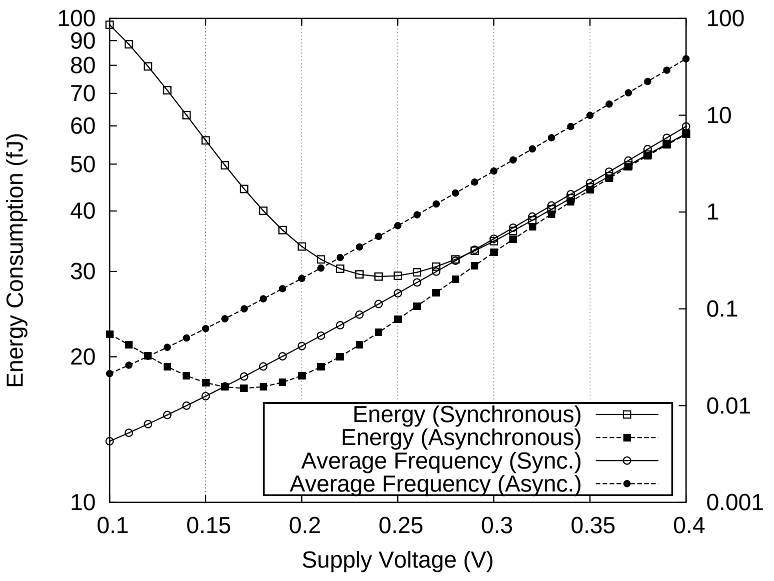


Fig. 4: Total energy consumption and average operation frequency for changing supply voltage values for a switching/delay mean value of 0.1. The energy-minimum operating points occurs at different voltage values for synchronous and asynchronous cases.

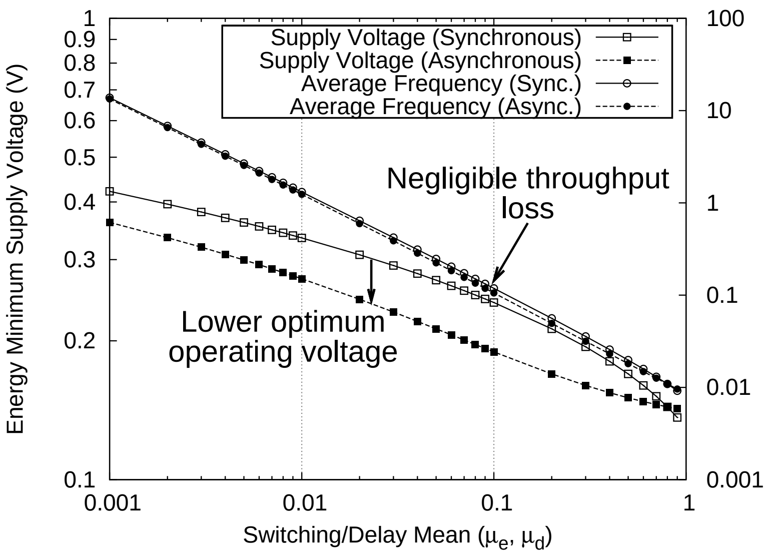


Fig. 5: Energy-minimum supply voltage and the respective throughput change for varying switching/delay mean values. The throughput loss due to the lower minimum energy operation supply voltage is negligible.

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 optimum operating voltages of the same circuit for synchronous and asynchronous operations for the specified mean values occur at $210mV$ and $170mV$, respectively.

Advantages of operating in an asynchronous manner in the sub-threshold regime are two fold. First, the leakage energy is reduced by reducing the average time during which the circuit leaks. Lower leakage energy consumption reduces the energy-minimum supply voltage value. This reduction in the supply voltage effectively reduces the switching energy consumption. This can be seen in the plot of energy-minimum supply voltage values and their respective throughput values at those voltage values (Fig. 5). In the figure, 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 in asynchronous operation because of better-than-worst-case computation delay.

III. Completion Detection System for Sub-threshold Operation

Let us consider the 4-phase bundled data asynchronous micro-pipeline shown in Fig. 2. As mentioned earlier, the main problem with this configuration when operating in the sub-threshold regime is the *matched delay line*. Due to process variations this delay line has to be severely 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 must reduce the time the asynchronous block spends *leaking*, both saving leakage energy and moving the minimum energy operating voltage to a lower value.

Instead of using a a fixed delay line, one solution is to detect the completion of the operation. One way to implement a completion detection circuit is to monitor the current consumption of the combinational block. As long as the combinational block is *switching* there will be dynamic power consumption in the circuit, which will be 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 [7]–[9]. 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.

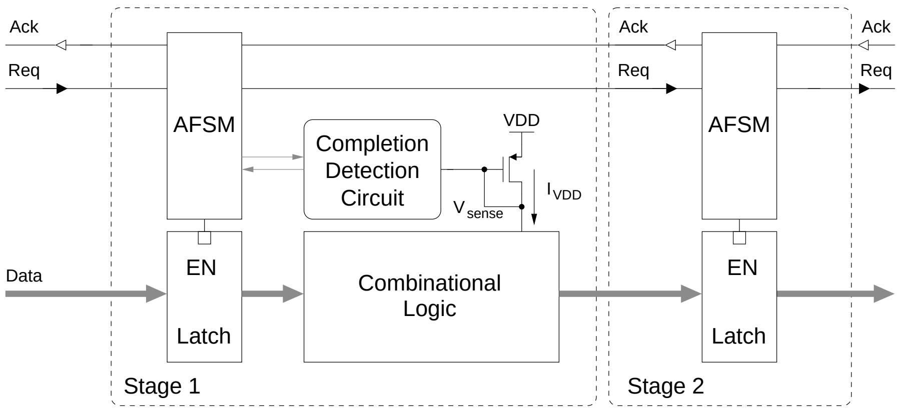


Fig. 6: General block diagram of the completion detection system. The system consist of an AFSM, completion detection circuitry and a sensor transistor.

Figure 6 shows the general block diagram of the current sensing based completion detection system. The completion detection system consist of an AFSM, completion detection circuit, which consist of pulse generators and an AC-coupled amplifier, and a single pMOS transistor.

A. Sensing and Sensed Signal Amplification

We propose a current sensing technique where the supply node (V_{DD}) of each combinational macro block in the asynchronous micro-pipeline will be driven by a diode-connected pMOS transistor. While the combinational circuit is idle (i.e. is only *leaking*) the supply voltage is at its nominal value. As soon as the *computation* phase starts the combinational block will start drawing current from the pMOS transistor while the internal nodes begin to switch. The gate-to-source voltage (V_{sense}) of the pMOS transistor will change to accommodate the current change as shown in Fig. 7. By detecting the change in this current, it is possible to detect the *computation* time of the combinational block.

Two operation regions can be discerned in the figure while a computation takes place. One region corresponds to the time when the actual switching of the internal nodes is taking place and the other to

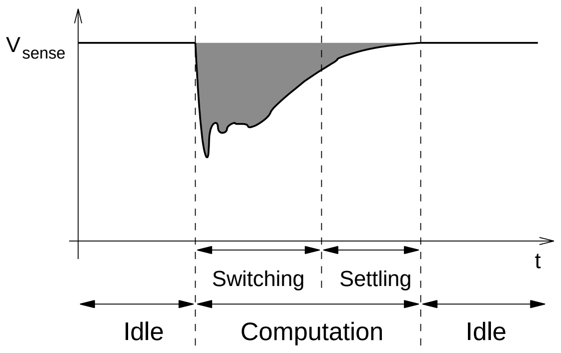


Fig. 7: Computation signature detected as the temporary drop of the supply voltage at the drain node of the current sensor device. Two main operation regions are shown, computation and idle. The computation frame is divided into switching and settling regions.

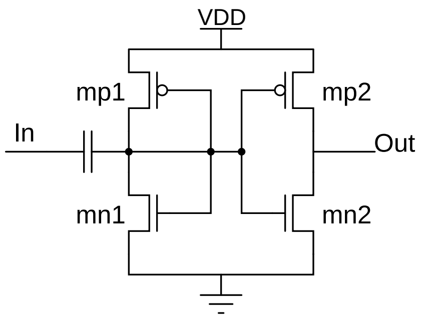


Fig. 8: AC-coupled amplifier used to amplify the detected signal.

the settling of the supply node of the combinational logic block. All the internal nodes that will settle to a logic high value follow the settling of the supply node. The settling time of the supply node depends on the capacitance of the supply node of the logic block and the resistance of the diode connected pMOS transistor. In order to reduce the serial resistance and hence the excessive settling delay as much as possible, a low-V_T pMOS transistor is used to supply the combinational logic block. In [14] the computation time comparison of an 16-bit ripple carry adder with and without the sensor transistor was made and it was shown that the adder with the sensor transistor takes about 40% more time for computation, on average.

Another challenge involves the magnitude of the signals. The supply voltage in the sub-threshold regime will be lower than 400 mV. Consequently, the change in the sensed signal will be in the range of a few tens of mVs at most. After sensing the current signal and converting to a log-compressed voltage signal, amplification of this signal is necessary before feeding it into the completion pulse generator. For the amplification of the sensed signal a class AB CMOS inverter-amplifier is used. Implemented AC-coupled amplifier used for amplification is shown in Fig. 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. The sizing optimization depends on the implementation characteristics of the logic block. If lower timing overhead is required for lower energy operation, the sensor transistor's aspect ratio can be increased to reduce the voltage drop (sensed signal) on the sensor transistor. Because of the lower amplitude of the sensed signal, more gain is required in the AC-coupled amplifier, hence higher power consumption . On the other hand 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.

B. Completion Signal Generation

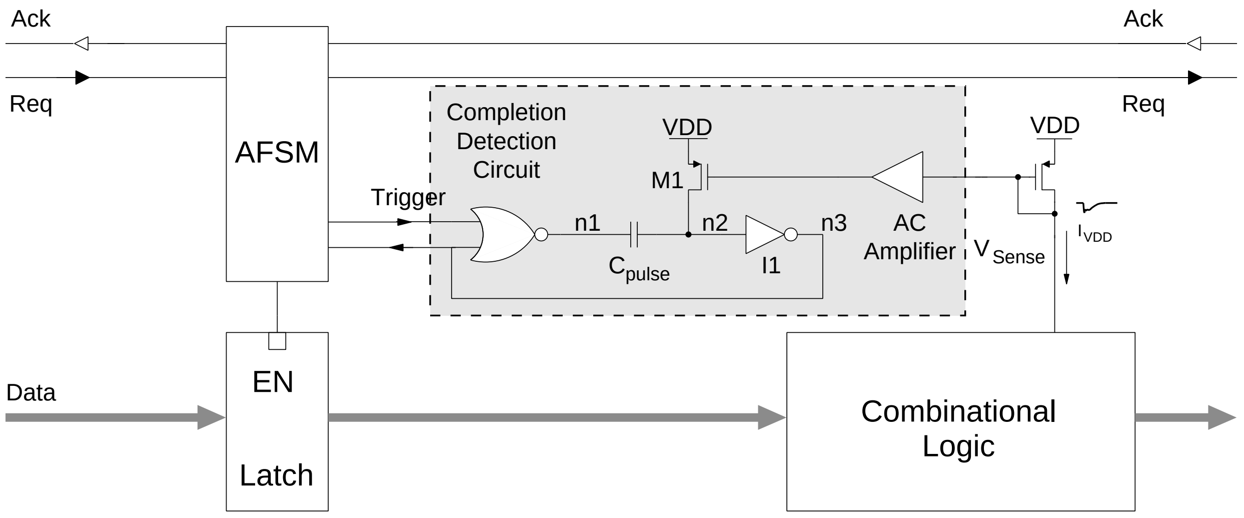


Fig. 9: Principle of the completion detection system using a triggering circuit and a variable pulse generator.

The current consumption of a combinational block depends entirely on the input data switching probability. In the extreme case, where the input data does not change for two consecutive cycles, there will be no dynamic current consumption, and the system will not be able to detect a change in the current. For this reason the system must provide an automatic *timeout* feature.

We address this problem by using a circuit with inherent *timeout* feature that consists of a trigger circuit and a variable length pulse generator as shown in Fig. 9. We implemented the variable pulse generator as a monostable multivibrator as shown in [15]. The trigger circuit inside the AFSM is activated with the arrival of new data to the combinational block. This starts the pulse generator by setting nodes *n1* and *n2* to logic-0. At this point, the pMOS (M1), which acts as a variable resistor, starts charging node *n2* until this node reaches the inversion threshold of the inverter (I1). This causes the node *n3* to switch finalizing the completion detection pulse generation. The time required to switch node *n2* is dictated by the value of the charging current I_{charge} and the value of the pulse capacitor C_{pulse}. The charging current I_{charge} is inversely proportional to the change in V_{sense}, thereby generating a pulse that directly corresponds to the *switching* time of the combinational logic block. The correct computation completion signal generation is achieved by matching the time required to charge node *n2* to the actual computation time. The generated pulse width of the variable pulse generator is given by

$$T = C(R + R_{on}) \ln \left[\frac{R}{R + R_{on}} \frac{V_{DD}}{V_{DD} - V_{th}} \right] \tag{18}$$

where R is the average resistance of the pMOS transistor during pulse generation, R_{on} is the resistance of the NOR gate and V_{th} is the switching threshold of the inverter. Assuming $R_{on} << R$ and $V_{th} = V_{DD}/2$, equation (18) can be simplified as $T = 0.69CR$. This means the pulse generated will be compressed by a factor of 0.69 when compared to the actual computation time in the ideal case.

In the standard implementation of the variable pulse generator, node $n3$ is used as the output. For some cases, the resistance of the pMOS transistor might be too low. A very fast switching occurs at node $n2$ and no switching occurs at node $n3$, resulting in no signal generation. To prevent this situation, the signal at node $n1$ was used as the output. Figure 10 shows the node voltage waveforms of the pulse generator for two different control voltage waveforms. The signal at $n1$ goes low when the trigger signal goes high and if there is no change in the signal at $n3$, it goes high as the trigger signal goes low. This operation guarantees that a completion detection signal is generated for each trigger input, hence the *timeout* feature. The minimum width of the completion detection pulse is thus set by the trigger pulse width, setting the minimum delay of the *Request* signal of the preceding stage.

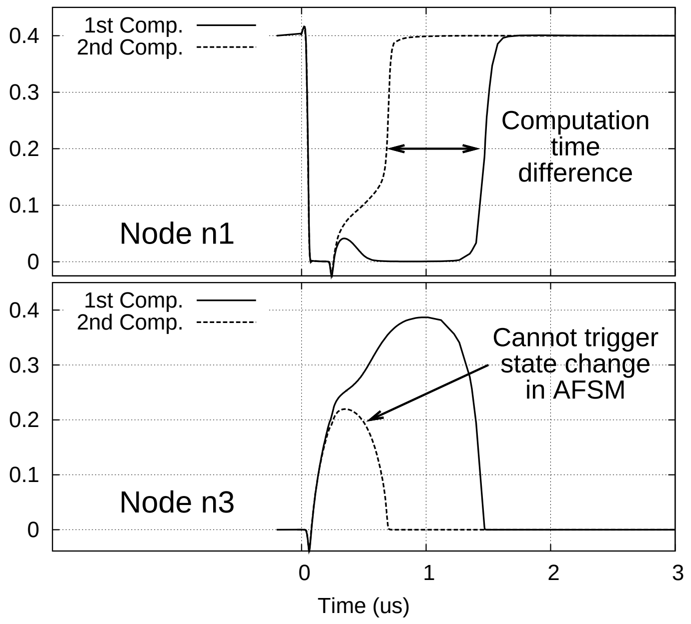


Fig. 10: Variable pulse generator node voltages. The pulse width of the signal at node *n1* is proportional to the control voltage, and hence, to the actual computation completion time of the combinational block.

We have used MOS-based capacitors instead of the more common metal-insulator-metal (MIM) to realize the pulse capacitor C_{pulse}. In this way, the capacitor value is made more dependent on the PVT variations, improving the immunity of the circuit against such PVT variations. One disadvantage of MOS-based capacitors is their non-linearity. In the parallel compensated configuration presented in [16], all MOS transistors remain in the depletion mode for all the input and output voltage combinations . Within the operation range of a few hundred mVs this configuration gives a capacitance value that varies less than 4%. While this value is by no means spectacular, for the pulse generation circuit where the current is averaged over time, it is more than sufficient.

IV. Implementation

To implement the completion detection circuit, several components need to be designed. The first stage is to design a suitable AC amplifier to increase the voltage swing of the sense voltage V_{sense}. In the second step the RC constant of the pulse generator needs to be matched to that of the combinational circuit for varying switching/delay probabilities. To determine the delay of the combinational circuit, we first extracted the SPICE-level netlist of the critical path of each combinational circuit from the back-end design database using Synopsys PathMill static analysis tool. Later HSPICE is used to simulate the netlist using detailed models. From the simulation results, the appropriate resistance and sizing for the pMOS transistor is derived. We later merged the completion detection circuit block with the AFSM described in the following subsection.

A. Asynchronous Finite State Machines

The asynchronous communication protocol between consecutive pipeline stages is a 4-phase bundled data protocol based on [17]. The AFSM that implements this protocol has been modified to include the completion detection circuit described in Section III. A gate level implementation of this AFSM is given in Fig. 11. Extensive HSPICE simulations were used to verify the functionality of this AFSM over process corners.

One of the main problems of sub-threshold operation is implementing state-holding elements. Such circuits usually depend on a feedback loop to preserve the desired state. High leakage currents in the

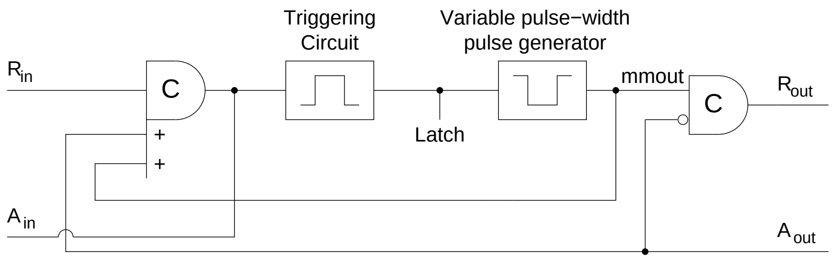


Fig. 11: AFSM including the pulse generating completion detection circuitry.

TABLE I: Comparison of Muller-C elements operating at $V_{DD} = 0.4\,$V.

	Dynamic	Martin	Sutherland	Van Berkel
Reference	[12]	[18]	[12]	[19]
Average Current [pA]	31.3	138.9	39.7	36.9
Total Energy [fJ]	12.5	59.5	15.5	14.8
Delay [ns]	63.2	101	67.9	45.4
EDP [pJ.ns]	0.79	6.01	1.05	0.67
State retention	No	Yes	Yes	Yes

sub-threshold operating regime may result in state loss in some of these circuits. In asynchronous circuit design, the main state holding primitive is the Muller-C element [20]. The correct operation of the Muller-C element is crucial for any asynchronous circuit. There are several published Muller-C implementations in the literature. We have compared four separate Muller-C implementations operating in the sub-threshold regime against each other: Dynamic and static versions suggested in [12], a dynamic implementation with feedback suggested by Martin [18], and the one designed by Van Berkel for low-power operation [19].

All four circuits were compared in terms of their power/energy consumptions and their operation speeds. The simulation results for a supply voltage of $0.4\,$V are given in Table I. It can be seen that the dynamic Muller-C element is not able to retain its state at this supply voltage. Of the remaining alternatives the Van Berkel Muller-C element consumes the least amount of power and is also the fastest implementation. The standard cell library of the target technology has been enhanced by implementing the Van Berkel Muller-C element.

The AFSM is a relatively small circuit and was easily implemented manually by placing and connecting standard cells. The only full custom additions were components of the pulse detector: C_{pulse}, resistor transistor M1, and the pMOS transistor supplying the combinational logic block with current.

V. Results

Extensive post-layout simulations have been performed on the completion detection system. The waveforms in Fig. 12 show the operation of the completion detection system for three consecutive cycles operating at $V_{DD} = 0.22\,V$. The supply voltage value of 0.22 V is the energy-minimum supply voltage for synchronous operation and for all the timing related simulations this value is used. In the figure the *switching* phase of the first computation continues for a long time (51.753$\,\mu s$). During the second computation, the input data does not change, and consequently there is no *switching* activity within the combinational block. Still due to the timeout feature, a fast completion detection signal is generated by the circuit (12.334$\,\mu s$). The width of this fast signal is set by the pulse-width of the *latch/trigger* signal as explained before. The last operation is much shorter than the first one and the sensed voltage signal V_{sense} settles more quickly to its final value, resulting in a shorter pulse (31.709$\,\mu s$) when compared to the first generated pulse.

Figure 13 shows the *switching* delay distribution extracted from the post-layout simulation of the 16-bit adder with completion detection for a large number of input vectors while operating with a supply voltage of $0.22\,V$. It can be seen that the mean value of this distribution is slightly larger than $40\,\mu s$. On the same graph the worst case path of the same combinational circuit is shown as well. This figure clearly shows

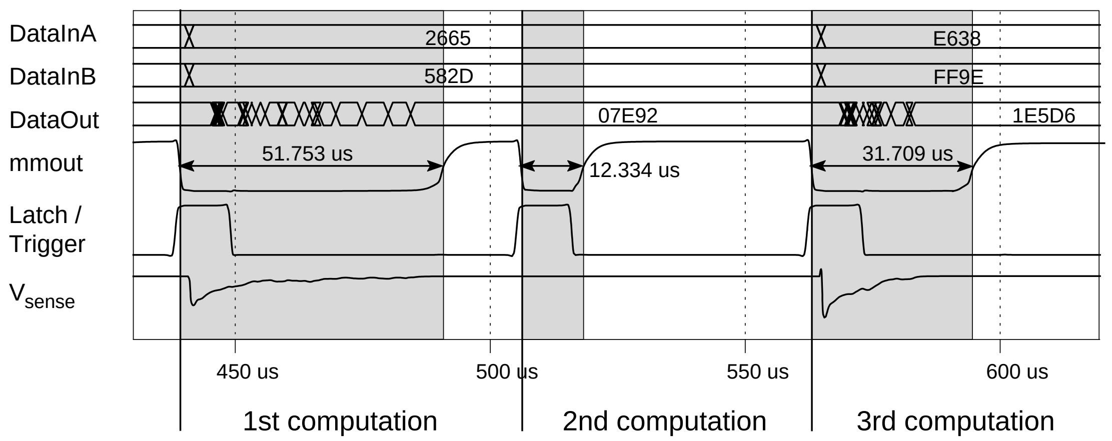


Fig. 12: Operation of the completion detection circuitry at $V_{DD} = 220\text{mV}$. Three consecutive computations are shown. The time difference between the computations and the automatic timeout feature are emphasized.

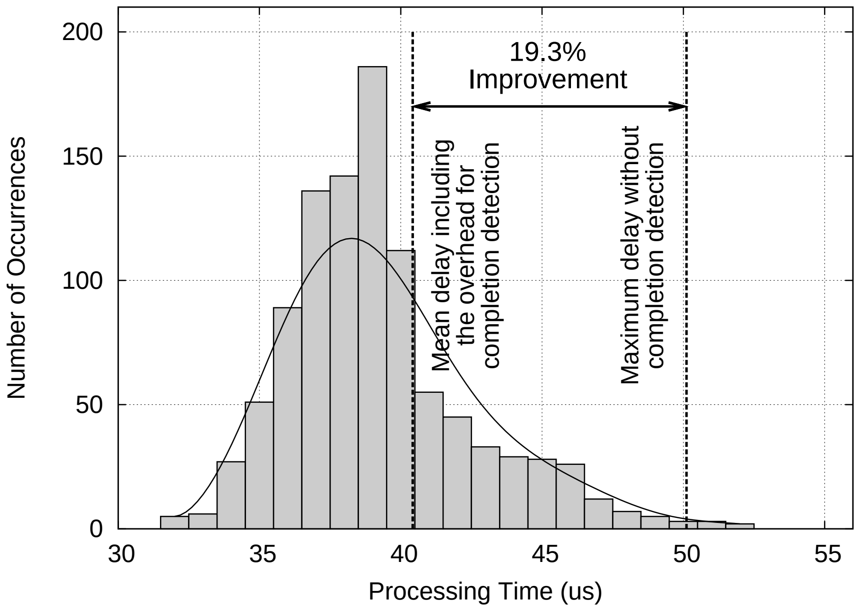


Fig. 13: Speed improvement in the operation of the 16-bit adder. Both the mean delay generated by the completion detection circuitry and the critical path delay are shown.

the potential benefit of the completion detection circuit for coarse grained pipeline stages. Including the delay overhead of the completion detection circuit and the sensor transistor, it can be seen that the average gain in throughput is still 19.3%.

In Section II-B it was shown that energy-minimum supply voltages for synchronous and asynchronous operations are different. We also investigated the energy reduction due to asynchronous operation at a lower supply voltage. Table II compares two implementations: *Synchronous* corresponds to the case where the adder is operating at the synchronous energy-optimum supply voltage value. The *Asynchronous* case shows the implementation with completion detection including the overhead for the completion detection circuitry while working at the asynchronous energy-optimum supply voltage. The energy consumption numbers given are for equal number of computations while operating on the same input data. For the 16-bit adder circuit, the synchronous energy-minimum supply voltage is $0.22\,V$ while for the asynchronous case the same value is $0.16\,V$. Due to the asynchronous operation and the lowering of the energy-minimum supply

TABLE II: Comparing Synchronous and Asynchronous Energy Consumption

Implementation	Supply Voltage	Energy consumption
Synchronous	0.22 V	31.48 pJ
Asynchronous	0.16 V	26.12 pJ

voltage value, asynchronous operation reduces the energy consumption by 17% for the same computation load.

Corner simulations were also run and the completion detection circuit functioned without any loss of performance for all the corner cases except the slow-slow corner. In corners other than the slow-slow one, completion detection circuit successfully tracked the variation in processing time of the logic block thanks to the MOS-based capacitors and pMOS resistance transistor. On the other hand for the slow-slow corner high correlation of the generated pulse width with the actual processing time worsens due to the slow AC-coupled amplifier response and slower charging of *n2* in Fig. 9.

VI. Conclusions

In this work, the advantages of operating a coarse-grained CMOS circuit asynchronously in the sub-threshold regime and a novel completion detection system for sub-threshold operation are presented. The completion detection concept is based on a current sensing and pulse generation method suitable for sub-threshold asynchronous pipelines. The proposed system is capable to detect supply current variations in the pA-to-nA range which are usually encountered in sub-threshold operation, it utilizes variable width pulse generation, and can be easily integrated into the standard CMOS design flow. The method is demonstrated as a proof of concept on a 16-bit CMOS adder using conventional EDA tools for $0.18\,\mu m$ CMOS technology. The completion signal generated by the proposed system allows throughput improvements on the operation of the adder circuit up to 19.3% compared to the synchronous operation. If the throughput is not a concern, by working at the energy-optimum supply voltage for the asynchronous operation, energy consumption reduction of 17% can be realized. The completion detection system presented in this work is also applicable to digital systems operating with above-threshold supply voltages for improving the throughput, without any modifications.

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," *Journal of Low Power Electronics*, vol. 3, no. 3, pp. 234–253, 2007.
6. M. Imai and T. Nanya, "A novel design method for asynchronous bundled-data transfer circuits considering characteristics of delay variations," in *Proceedings of the 12th IEEE International Symposium on Asynchronous Circuits and Systems*, 2006, pp. 68–77.
7. 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.
8. E. Grass and S. Jones, "Asynchronous circuits based on multiple localised current-sensing completion detection," in *Asynchronous Design Methodologies, 1995. Proceedings., Second Working Conference on*, May 1995, pp. 170–177.
9. 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.
10. 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.
11. J. Sparso and S. Furber, Eds., *Principles of Asynchronous Circuit Design - A Systems Perspective*. Kluwer Academic Publishers, 2001.
12. I. Sutherland, "Micropipelines," *Communications of the ACM*, vol. 32, no. 6, July 1989.
13. R. Corless, G. Gonnet, D. Hare, D. Jeffrey, and D. Knuth, "On the LambertW function," *Advances in Computational Mathematics*, vol. 5, no. 1, pp. 329–359, 1996.
14. O. C. Akgun, Y. Leblebici, and E. A. Vittoz, "Current sensing completion detection for subthreshold asynchronous circuits," in *Proceedings of the European Conference on Circuit Theory and Design 2007*, 2007, pp. 376–379.

15. A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 4th ed. Oxford University Press, 1998.
16. J. Sauerbrey, T. Tille, D. Schmitt-Landsiedel, and R. Thewes, “A 0.7-v MOSFET-only switched-opamp sigma-delta modulator in standard digital CMOS technology,” *IEEE Journal of Solid-State Circuits*, vol. 37, no. 12, pp. 1662–1669, Dec. 2002.
17. 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.
18. A. Martin, “Formal program transformations for VLSI circuit synthesis,” *Addison-Wesley University Of Texas At Austin Year Of Programming Series*, pp. 59–80, 1989.
19. M. Shams, J. Ebergen, and M. Elmasry, “A comparison of CMOS implementations of an asynchronous circuits primitive: the c-element,” in *ISLPED ’96: Proceedings of the 1996 international symposium on Low power electronics and design*. Piscataway, NJ, USA: IEEE Press, 1996, pp. 93–96.
20. D. E. Muller and W. S. Bartky, “A theory of asynchronous circuits,” in *Proceedings of an International Symposium on the Theory of Switching*. Harvard University Press, 1959, pp. 204–243.