WestminsterResearch
http://www.westminster.ac.uk/westminsterresearch

Energy efficient implementation of multi-phase quasi-adiabatic Cyclic Redundancy Check in near field communication

Maheshwari, S., Bartlett, V. and Kale, I.

Energy Efficient Implementation of multi-phase Quasi-Adiabatic Cyclic Redundancy Check in Near Field Communication

Sachin Maheshwari, V.A.Bartlett and Izzet Kale

Applied DSP and VLSI Research Group, Department of Engineering,
University of Westminster,
London, W1W 6UW, United Kingdom
Email: w1412503@my.westminster.ac.uk, {v.bartlett, kalei}@westminster.ac.uk

Abstract

Ultra-low power operation in power-limited portable devices (e.g. cell phone and smartcard) is paramount. Existing conventional CMOS consume high energy. The adiabatic logic technique has the potential of rendering energy efficient operation. In this paper, a multi-phase quasi-adiabatic implementation of 16-bit Cyclic Redundancy Check (CRC) is proposed, compliant with the ISO/IEC-14443 standard for contactless smart cards. In terms of a number of CRC bits, the design is scalable and all generator polynomials and initial load values can be accommodated. The CRC design is used as a vehicle to evaluate a range of adiabatic logic styles and power-clock strategies. The effects of voltage scaling and variations in Process-Voltage-Temperature (PVT) are also investigated providing an insight into the robustness of adiabatic logic styles. PFAL and IECRL designs using a 4-phase power-clock are shown to be both the most energy-efficient and robust designs.

Keywords— quasi-adiabatic logic; CRC; near field communication; energy consumption; robustness; smartcard.

1. Introduction

CRC is widely used in all data-communication, transmission and memory devices as a powerful method for detecting errors. One of the traditional hardware solutions for the CRC calculation is a bit-serial approach using a Linear Feedback Shift Register (LFSR), consisting of XOR gates and flip-flops [1]. A general diagram for CRC using an LFSR is shown in Fig.1. Depending on the application, a generator polynomial is used which gives a high probability of error detection [2]. For very high-speed data transmission, researchers have proposed many hardware and software-based CRC implementations. These include parallel software implementations based on look-up algorithms [3] and hardware implementations based on the z-transform [4], matrix formulation [5] and pipelining [6]. These parallel approaches focus mainly on fast error detection when processing large data messages. Software solutions have several drawbacks: they are slow, they occupy processor resources, and require ROM storage for the lookup table. Nevertheless, in the references cited above, the energy consumption has not been considered.

1.1. Motivation

Due to the increased usage of battery-less applications (e.g. a smartcard) and rising energy density due to the technology shrinkage, energy-efficiency has become a major concern in the design of large systems. To address this, a circuit technique, "Quasi-Adiabatic Logic" based on the CMOS technology, has the potential for low energy operation albeit at some cost in terms of performance speed. However, adiabatic logic can provide sufficient performance to be used to design energy-efficient communication protocol which has low data rate such as Radio Frequency Identification (RFID) and NFC operating at 13.56MHz. Although, adiabatic logic is in existence for more than two decades, still, its full potential has not been discovered.
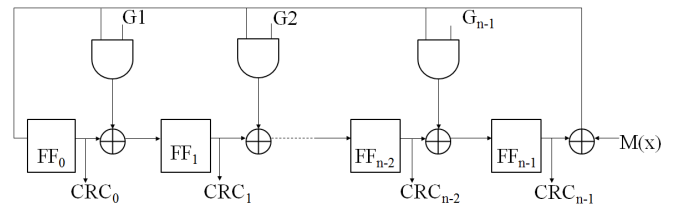

Fig. 1. A bitwise serial LFSR for n-bit CRC generator.

In the literature, researchers have mostly demonstrated the low energy benefits of adiabatic logic using implementations such as counters [7], multiplexers, adders and multipliers [8]. At the system level, very few papers exist [9, 10], demonstrating the energy benefits in comparison to non-adiabatic (static CMOS). In this paper, we compare the performance of multi-phase adiabatic logic designs in particular energy dissipation, throughput, latency, area, robustness and complexity based on the circuit and the power-clocking scheme. Practically, it is difficult to design an optimum adiabatic system but the trade-offs between energy, speed, area, complexity and robustness can be established that enables the designer to design an optimum adiabatic logic system. The main motivation of this work is to design an energy-efficient 16-bit CRC based on the standards and protocol of the NFC frame format outlined in ISO/IEC 14443-3 [11].

1.2. Contribution of this paper

The focus of this paper is to compare a CRC implemented using five energy-efficient quasi-adiabatic logic designs namely: Efficient Adiabatic Charge Recovery Logic (EACRL), Improved Efficient Charge Recovery Logic (IECRL), Positive Feedback Adiabatic Logic (PFAL), Complementary Pass-transistor Adiabatic Logic (CPAL) and Clocked Adiabatic Logic (CAL) and analyse the performance trade-offs over a wide range of external constraints as discussed above. Here the multi-phase is referred to as the power-clocking scheme used

1

by adiabatic logic designs. The main contributions of this paper are numbered as follows;

1) We present a hardware implementation of 16-bit multi-phase adiabatic CRC for NFC application.
2) We present a CRC design which can be scaled up or down by adding or removing the CRC slices in the datapath and flip-flops in the register unit for an application other than the NFC.
3) A methodology is proposed to minimize the design time and synchronization issue by implementing a CRC design which is suitable for a range of adiabatic clocking strategies, specifically 4-phase, 2-phase and single phase.
4) A system level implementation of CRC comprises of a power-clock generator for different adiabatic clocking strategy was implemented and compared on the basis of energy consumption.
5) Finally, we analyse the performance trade-offs in terms of energy benefits, throughput, latency, complexity, robustness and area between multi-phase adiabatic CRC implementations. We also compare these with a non-adiabatic CMOS design.

1.3. Structure of the paper

The structure of the paper is as follows. Section 2 reviews the energy dissipation of the quasi-adiabatic logic due to adiabatic loss and non-adiabatic losses. Then the five chosen quasi-adiabatic logic techniques are discussed in short. Section 3 presents the application of CRC in NFC. The design methodology is presented in Section 4. Implementations of 16-bit CRC according to ISO/IEC 14443 standard are presented in section 5. Section 6 presents the simulation results and performance comparison using five adiabatic logic designs and non-adiabatic logic design. Finally, the paper is concluded in section 7.

2. Quasi-Adiabatic Logic Families

The term "Quasi" describes the logic that involves some theoretical losses arising due to the threshold voltage degradation. Such losses are termed as Non-Adiabatic Loss (NAL). For low energy operation, adiabatic logic uses a slowly changing power-clock which allows approximately constant current charging/discharging and by avoiding current surges, the circuit dissipates less energy [12]. In addition, the power-clock also makes possible the recovery of charge by pumping the stored energy back to the power supply during the discharging process. The power-clock generator can be implemented either using a stepwise charging circuit [13-14] or an inductor based generator [15-16]. For more than two decades, adiabatic logic

has been widely studied and various energy efficient logic families have been proposed [7-10]. Since the implementation and the distribution of multiphase power-clocking scheme requires additional area, energy consumption and increases complexity, logic families with more than 4-phases are not taken into account

Single-phase and 4-phase power-clocks are broken down into four equal time periods namely evaluation (E), hold (H), recovery (R) and idle (I). On the other hand, due to the non-overlapping power-clock requirement of 2-phase, its idle time period is three times than the rest of each three time periods. Fig.2 shows the corresponding multi-phase power-clocking schemes along with the relationship of the power-clock period, $T_{clk, phase}$, with the ramping time, T_r.

The mathematical relationship for the energy dissipation also known as adiabatic loss (AL), E_D, using a ramp during charging phase is given as;

$$E_D = \frac{R_{ON}C_L}{T_r} C_L V_{DD}^2 \qquad (1)$$

Where C_L is the lumped load capacitance at the output node of the circuit, R_{ON} is the resistance of the charging path and V_{DD} is the maximum supply voltage. The detailed derivation of (1) is given in [17]. According to (1), it is possible to reduce the energy dissipation to an arbitrary degree by increasing the ramping time to ever-larger values. However, there is a practical lower limit to the ramping time value due to the increased leakage at longer ramping times.

These adiabatic logic families also suffer from NAL arising in the evaluation and recovery phase depending upon the circuit topology. NAL occurs because of the threshold voltage degradation. In the evaluation phase, the output follows the power-clock only when the source-to-gate voltage of pMOS transistor is greater than or equal to its threshold voltage $|V_{t,p}|$. Similarly, during the recovery phase of the power-clock, when the supply voltage goes below the threshold voltage, through one of the pMOS transistors, it is turned off and a residual charge remains on the output node. This residual charge gets discharged non-adiabatically at the start of the next cycle when new input is evaluated. This part of non-adiabatic discharge is independent of the frequency but can cause high energy dissipation for large system designs with high fan-out. It represents the main part of the NAL and is equal to

$$E_{NAL} = \frac{1}{2}C_L V_t^2 \qquad (2)$$

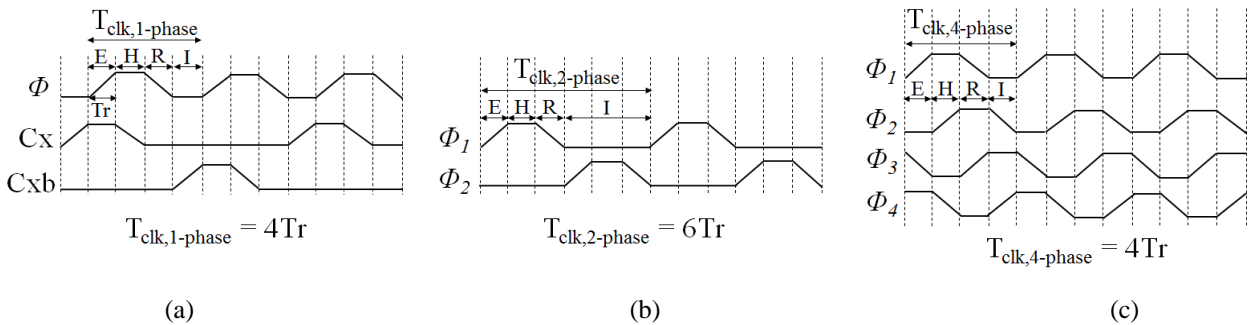Another part of the NAL occurs from coupling effect which is

Fig. 2. Power-clocking Scheme (a) Single-phase (b) 2-phase (c) 4-phase.

circuit topology dependent. The low-level output goes to a negative voltage value during the complete recovery phase of the power-clock due to the absence of cross-coupled nMOS transistors. The topology which has cross-coupled transistors, the coupling effect occurs for a part of recovery phase when the supply voltage goes below the pMOS threshold voltage.

In 1994, Denker [18] introduced a high performance improved ECRL logic circuit shown in Fig. 3, also called 2N-2N2P, an improvement over conventional ECRL [19]. The major difference is that IECRL has a pair of cross-coupled nMOS transistors in addition to the cross-coupled pMOS as in ECRL. This leads to the reduction of coupling effect for a large part of the recovery phase until the power-clock reaches the threshold voltage.
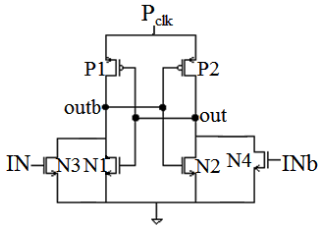

Fig. 3. IECRL buffer [18].

In 1996, A. Vetuli et al. [20] has presented a new adiabatic logic family, known as PFAL, which makes use of a CMOS positive feedback amplifier. It is very similar to IECRL, but its evaluation tree is connected between the power-clock and the outputs [20] as shown in Fig. 4 (a).
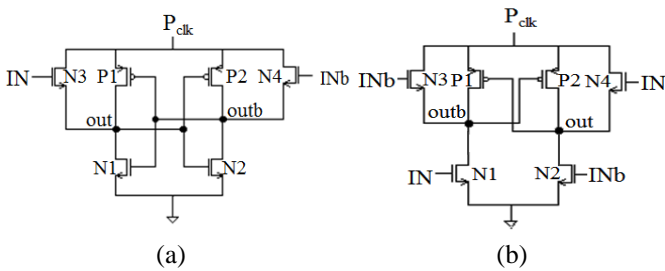

(a)   (b)

Fig. 4.   (a) PFAL buffer [20] (b) EACRL buffer [21]

In 2001, Varga et al. [21] proposed a novel dual-rail energy efficient adiabatic charge recovery logic (EACRL). The EACRL buffer uses a pair of cross-coupled PMOS transistors and duplicate evaluation trees; one connected between the output and ground and the other, driven in anti-phase, connected between the power-clock and the output as shown in Fig. 4(b). Both PFAL and EACRL logic is an improvement over IECRL as they completely eliminate NAL during the evaluation phase of the power-clock. Also, both logic has reduced equivalent resistance at the two output nodes due to the formation of transmission gate pairs (N3, P1 and N2, P2) as shown in Fig 4 (a) and (b). However, due to the absence of the cross-coupled nMOS transistors in EACRL, an additional non-adiabatic dynamic loss occurs due to the coupling effect. All the above three adiabatic logic families use 4-phase power-clocking scheme for cascaded logic.

Due to the complexity of generating 4-phase power-clock, Maksimovic et al. [22] proposed a logic topology called CAL, which uses the single-phase power-clocking scheme as an improvement over the logic using 4-phase power-clocking scheme. The CAL buffer is similar to IECRL but has clocked nMOS transistors (N3, N4) between the evaluation nMOS

transistors (N5, N6) and the output as shown in Fig. 5 (a). The clocked nMOS transistors use a pair of auxiliary-clocks which allow operation from a single-phase power-clock. A more detailed description can be found in [22]. Although this topology results in the simplified power-clock generator but the use of the auxiliary clock signals for cascaded logic, make it complex and result in extra area and energy overhead. Also due to the stacking of transistors at the two output nodes, it has higher NAL arising because of large threshold voltage degradation.

As both, single-phase and 4-phase designs suffer from NAL, a new logic based on Pass-transistor Adiabatic Logic (PAL) was introduced by Oklobdzija and Maksimovic [23] in 1997. It uses 2-phase power-clocking scheme for cascaded logic. Then in 2003, Jaiping et al. [24] presented an improved PAL circuit, called CPAL which uses 4-phase power-clocking scheme for cascaded logic. But later in 2005, he demonstrated that the cascaded CPAL logic can be driven using 2-phase non-overlapping power-clocking scheme [25]. Fig. 5 (b) shows the CPAL buffer circuit which uses a PFAL buffer, with the main part of the evaluation tree (N5-N8) designed using the pass-transistors to connect to the gates of the nMOS pull-ups (N3-N4), also called bootstrapped transistors. Despite completely eliminating the NAL at the two output nodes, CPAL suffers from NAL on internal nodes X (or Xb). The more detailed description of its NAL on internal nodes are analysed in [25].

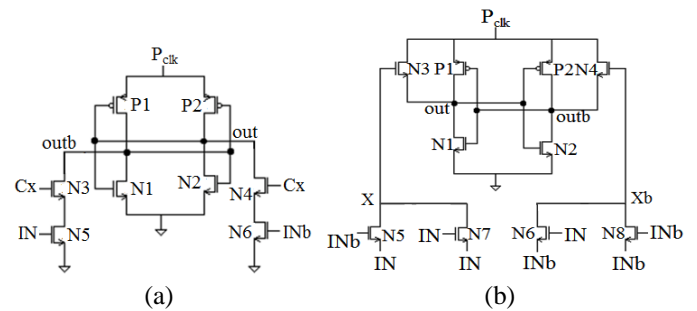
(a)   (b)

Fig. 5.   (a) CAL buffer [22] (b) CPAL buffer [24]

Though various multi-phase energy-efficient adiabatic logic families have been proposed in two decades, where each encompasses many novel ideas and saves considerable energy compared to the static CMOS, still literature lacks to address the issues related to the adiabatic logic in a complex system design; i) Selection of an adiabatic multi-phase logic for an application specific design; ii) Buffer insertion for handling synchronization issue incurring area and latency overhead; iii) Decreased throughput as multiple power-clock phases require different computation time; iv) and the non-adiabatic loss compromising the energy efficiency.

3. Application of CRC in NFC

NFC is an emerging Radio Frequency (RF) technology and is based on Radio Frequency Identification (RFID) standards for short-range wireless data-exchange between a reader and a target. NFC devices operate at 13.56 MHz, a frequency low enough to allow a bit-serial CRC approach to be used. Before transmitting the message bit stream, CRC value is calculated and appended to the message. At the receiver end, the CRC is calculated again to check errors in transmission. Thus, this continuously active sub-system makes it one of the most power-hungry modules [26]. The frame format of the data transmission in the NFC protocol of ISO/IEC 14443-3 [11] uses 16-bit CRC generator polynomial, $x^{16} + x^{12} + x^5 + 1$, k-bit payload (message

3

bit stream) and works at three different bit-rates. Out of the three bit-rates, two (212 kbps and 424 kbps) have 0x0000 as the pre-set value, whereas the slowest bit-rate, 106 kbps has 0x6363 pre-set value.

The CRC calculation is cyclic, which incorporates the current CRC value of the data (MSB first) and the CRC value of the previous data bytes. Let $M(x)$, $G(x)$, $Q(x)$ and $R(x)$ represent the message polynomial, generator polynomial, quotient polynomial, and remainder polynomial respectively. The message, $M(x)$ is a k-bit payload which is operated upon to form an n-k bit CRC detection block, where n is the length of the complete block. The algorithm for the CRC calculation for NFC is described in the following steps.

Step 1: The original k-bit payload, $M(x)$ is multiplied by x^{n-k} to shift the data and the pre-set value is appended.

Step 2: The result is then divided by the generator polynomial $G(x)$ to form the quotient $Q(x)$ and remainder $R(x)$.

Step 3: The transmission polynomial $T(x)$ is formed by appending the payload, $M(x)$ and the remainder, $R(x)$.

Step 4: At the receiver, the CRC calculation on the transmitted block, $T(x)$ is done to check for errors in transmission.

Step 5: After the transmission, the received message is processed with step 1-2 albeit with the received message replacing $M(x)$. If the remainder, $R'(x)$ produced is zero, the transmission is assumed to be error-free.

The more detailed description of CRC algorithms is specified in [27].

4. Methodology

We present a modification in conventional LFSR which fulfils the criteria for an ISO/IEC-14443 contactless card. Using the conventional CRC only a single bit-rate with an initial value of zeros can be loaded whereas, our proposed design is valid for a multiple data bit-rate and every initial load values. The proposed CRC also has the flexibility to be used for other power-clocking schemes without modifying the design. Although the CRC is implemented for NFC application it can be easily modified to accommodate different CRC application like mobile networks, Ethernet, USB, high-level data link control, etc [2]. A wide range of generator polynomials is presented in [3] along with their applications. With our strategy, an n-bit CRC can be implemented by replicating n "slices" of circuitry. This approach enables CRCs of every number of bits to be readily created, thus decreases design time and synchronization issues [28].

An n-bit CRC is designed using n-blocks of CRC slices in the datapath. Each block of CRC slices has four logic gates connected in a cascaded manner. Out of the n-blocks, n-1 are identical blocks having same logic gates connected in same order. Whereas, the Least Significant Bit (LSB) of CRC slice have the position of the XOR gate different than that of the identical blocks. This is due to the synchronisation of the feedback signal with the input message bits. A single block slice requires three stages or phases but due to the recursive nature of the CRC implementation, the number of stages should either be a multiple of two in the case of single-phase (because of auxiliary clock signals) and 2-phase designs or a multiple of four in the case of 4-phase logic designs. Thus, a buffer is added

in each slice to have an even number of stages for correct synchronisation and functionality. Each slice in the CRC datapath implemented using 4-phase designs take one power-clock cycle, whereas single-phase and 2-phase designs take four and two power-clock cycles respectively. The controller generates the synchronization signals for the CRC design. The CRC starts the computation when the 'New message' and 'R_count' signal values are logic '1'. The input message, $M(x)$ is provided to the CRC datapath using a multiplexer, used as a test circuitry for the CRC design. The counter outputs, part of the controller, act as the select lines for this multiplexer which provides serial input to the CRC datapath and the register unit.

The speedup technique is used as described in [29] to increase the throughput. The buffers in the counter are replaced with the functional logic gates (AND/OR/XOR). Thus, the throughput and latency of 4-phase designs are improved by ½ of the power-clock cycle whereas, in the case of 2-phase and single-phase CRC design, an improvement of one power-clock cycle and two power-clock cycles respectively is achieved. In addition, it also reduces the buffers required for synchronisation by four in the counter unit.

For a message word-length of 16 bits, the 16-bit CRC datapath requires 64 power-clock cycles using a single-phase power-clocking scheme, whereas, 32 and 16 power-clock cycles are needed if we are using 2-phase and 4-phase adiabatic logic respectively. In general, for the message word-length of k-bits, an n-bit CRC datapath requires 4k, 2k and k power-clock cycles for single-phase, 2-phase and 4-phase adiabatic logic respectively. Where k is always greater than or equal to n. Since the presented work is in accordance with ISO/IEC 14443 standard for NFC, a 16-bit CRC is designed based on the methodology and strategy used in describing n-bit CRC. The CRC is implemented in all the five adiabatic logic families and tested for its functionality and robustness against PVT variations. All the components including the multiplexer (providing input serially) are designed using adiabatic logic.

5. Implementation of 16-bit CRC in NFC

A block diagram of the 16-bit CRC design is shown in Fig.6. All the adiabatic logic designs have differential input and output signals, see Fig. 3-5, but for the simplicity and better understanding, complementary signals are not shown in Fig. 6. The complementary signals are denoted by a letter 'b' following a signal. The main part of the CRC design is its datapath which is responsible for computing the CRC value. The 16-bit input message ($M(x)$) is provided to the CRC datapath through the 16:1 multiplexer at every count of the counter. To be consistent with the protocol, MSB is the first bit transmitted as shown in Fig. 6. Since each block in the datapath has a latency of 4 power-clock phases, a delay cell is added at the output of the 16:1 multiplexer to synchronise the final CRC values from the CRC datapath and the input message ($M(x)$).

The CRC is initialised using the reset input 'RES' which clears the CRC unit, the register unit, the controller unit, resets the counter and load the pre-set value '0x6363'. When 'RES' signal is set false and the 'new message' bit is true the CRC starts the computation. The CRC value is calculated when the last message bit is sent and the counter reaches the value '1111'. Then the calculated CRC value from the datapath and the message bits gets appended to the register unit while the counter returns to value '0000'. The appended CRC value and the
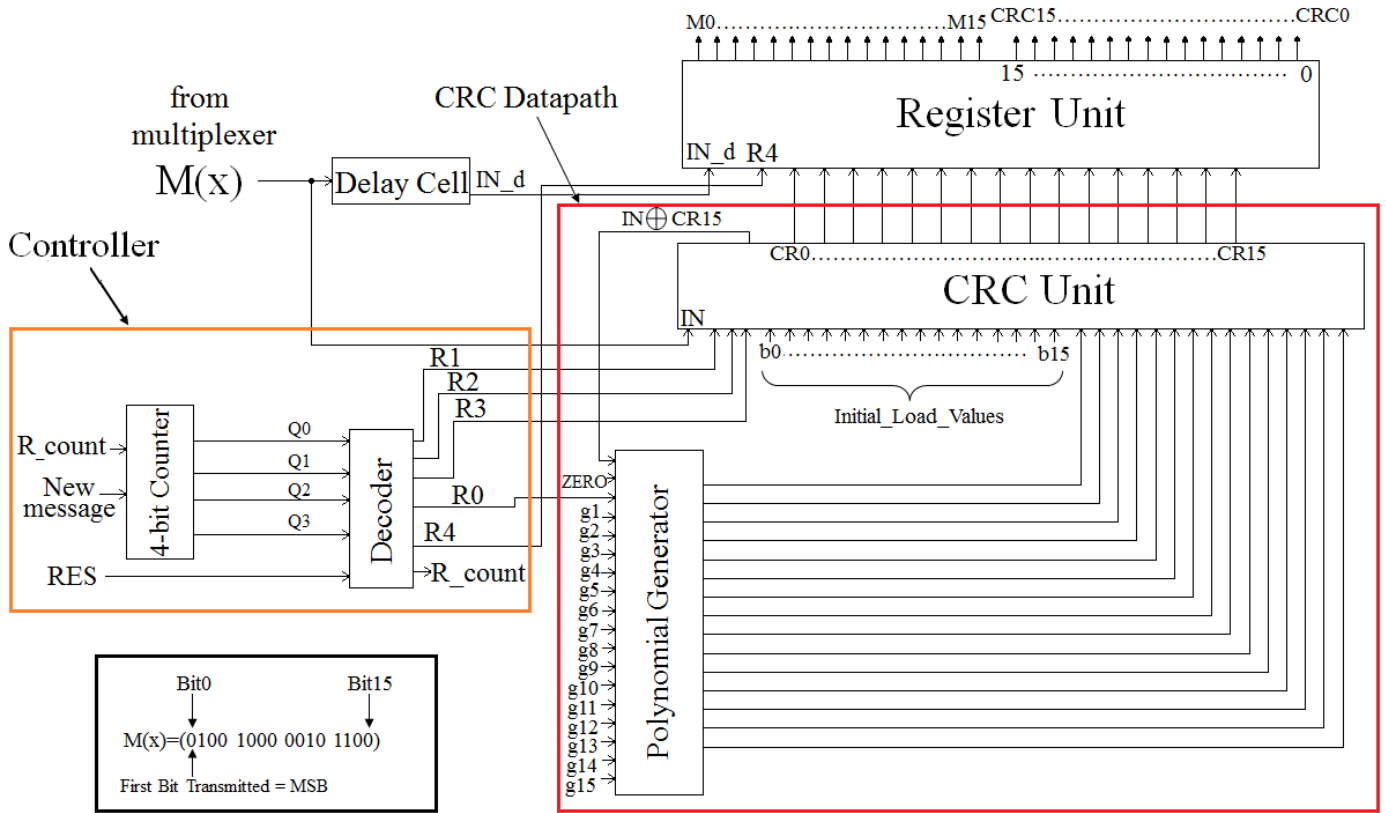
Fig. 6. Block diagram of the 16-bit CRC.

message word are retained during the wait period in the specially designed register unit, while the values in the CRC datapath are cleared to zero. The wait period lasts for two power-clock cycles and after that, the counter starts counting again automatically allowing the CRC to re-calculate its value. To calculate the new CRC value either the new message bits or the generator polynomial along with the load values can be provided during the wait period.

The CRC design has a number of advantages. Firstly, it can be used for the different power-clocking schemes as shown in Fig. 2. Secondly, all the control signals remain same for multi-phase adiabatic logic designs. Thus, the designer has only to pick the required adiabatic logic and replace it with their design saving design time and eliminating synchronisation issues. Thirdly, the use of polynomial generator unit and initial load value makes it reusable for other applications of 16-bit CRC.

In order to have the reusable CRC design for multi-phase clocking scheme and for application other than NFC, the implementation has associated hardware cost. Firstly, the generator polynomial unit incurs an area overhead of twelve 2-input AND gates and twelve 2:1 multiplexer. Secondly, for the range of adiabatic clocking CRC design, the register unit of the single-phase and 2-phase implementations use approximately 50% more buffers.

5.1. Controller (Counter and decoder)

The controller comprises a counter, generating states and a decoder (combinational logic) that generates the synchronisation signals for the CRC. The counter is designed using D flip-flops. It has two inputs, 'R_count' (coming from the decoder) and the 'New message'. The 'New message' input is an active high external input. Initially, it is zero when the counter is in the reset state. The counter starts counting when
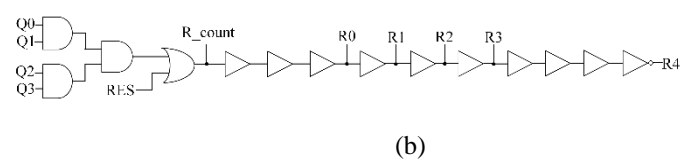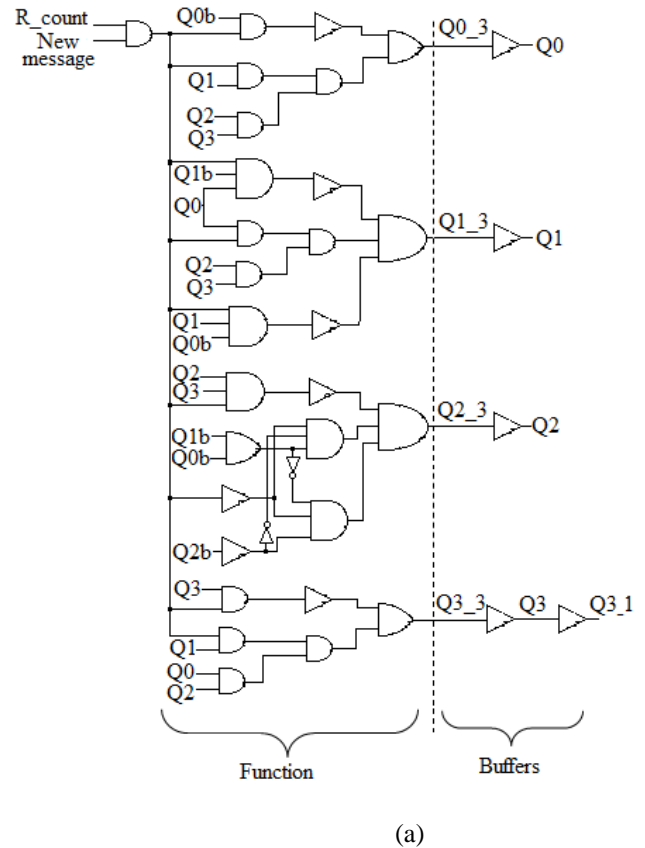

(a)

(b)

Fig. 7. Controller (a) 4-bit Counter (b) Decoder.

both the 'New message' and 'R_count' signal values are logic '1'. In general, the adiabatic D-flip-flop is structured using a cascaded buffer chain, but in this case, the buffers are replaced with the logic gates (AND/OR/XOR) which saves exactly twelve buffer gates. For the test purpose, the 16-bits new message is provided to the CRC datapath using 16:1 multiplexer. The counter outputs Q0_3, Q1_3, Q2 and Q3_1 are the select input to the multiplexer. Fig. 7 (a) shows the functional part and the synchronisation buffers used in the 4-bit counter. The inputs Q0b, Q1b and Q2b are the complementary signals of Q0, Q1 and Q2 which are not shown for clarity.

The outputs of the counter Q0, Q1, Q2 and Q3 are the inputs to the decoder along with the external reset input 'RES' as shown in Fig. 7 (b). The decoder provides automatically activation reset signals (R_count, R0, R1, R2) to the counter and the datapath. The signal, R0 is the input to the AND gate in generator polynomial bit blocks of the CRC datapath. Whereas, the signal, R3 is the select input for the 2:1 multiplexer in the CRC bit blocks which selects the initial load value when active high. The new generator polynomial along with the load values can be provided during the wait period. The signal, R4 is an inverted signal of R3 which is delayed by four buffer gates. It is used as a wait signal to the register unit that generates a wait period of two power-clock cycles. The decoder performs three tasks; firstly, it generates a retain signal which helps to retain the final CRC value in the register unit. Secondly, it reset the CRC datapath, the counter unit, and the register unit before the computation begins and after the final CRC value is computed. Lastly, the buffers in the decoder serve the purpose of synchronising the decoder output with different units of the CRC design for correct calculation of the CRC value.

The use of the signals from the decoder makes the CRC design to calculate the CRC value continuously after the counter reaches the value '1111'. Because each bit blocks in the CRC unit is having four logic gates connected in a cascaded manner, the implementation of the controller remains fixed for all the power-clocking schemes.

5.2   CRC Datapath

The CRC datapath consists of the CRC unit and the generator polynomial unit. The CRC unit computes the CRC value based on the generator polynomial (g1…..g15). The generator polynomial, G(x) for NFC applications, is $x^{16}+x^{12}+x^5+1$. Since the binary value of the MSB and LSB of the generator polynomial is always one, the polynomial generator unit consists of fifteen 2-input AND gates each followed by 2:1 multiplexers. The hex value '0x8810' corresponds to G(x) (g1, g2, ……., g15) is fed along with the reset signal, R0. The output of the AND gate triggers the multiplexer to select either a zero or the XOR function of the input message bit with the MSB bit of the CRC Unit (CR15) as shown in Fig. 8. The outputs from the generator polynomial bit blocks are fed into the XOR gates of the respective CRC bit blocks.

A 16-bit CRC has sixteen bit blocks with one LSB bit block and fifteen identical blocks (1 to 15) as shown in Fig. 8. Each identical block uses four logic gates which incorporate a synchronisation buffer, a resettable buffer for resetting the datapath, XOR gate for generator polynomial representation and 2:1 multiplexer for initial bit loading for different bit-rates (b0, b1,…., b15). The initial load value (0x6363) is loaded in the CRC datapath during reset operation when 'R3' signal is logic '1'. Two different resettable signals, R1 and R2 are used to synchronise the CRC unit due to the different position of resettable buffers in the CRC bit blocks and the LSB of CRC. The design can be reused either for a higher bit or for lower bit CRC, depending on the application by adding the identical CRC bit blocks or by eliminating it. Fig. 8 shows two feedforward paths and a feedback path. Both the feedforward paths comprise of four cascaded gates. Since the feedforward path 2 has a fixed latency of four logic gates (two XOR gates and two MUXs), a buffer is added in the feedforward path 1 for synchronisation. Thus, the n-bit CRC datapath implementation has a fixed overhead of n-buffer logic gates due to the synchronisation.

5.3.   Register Unit

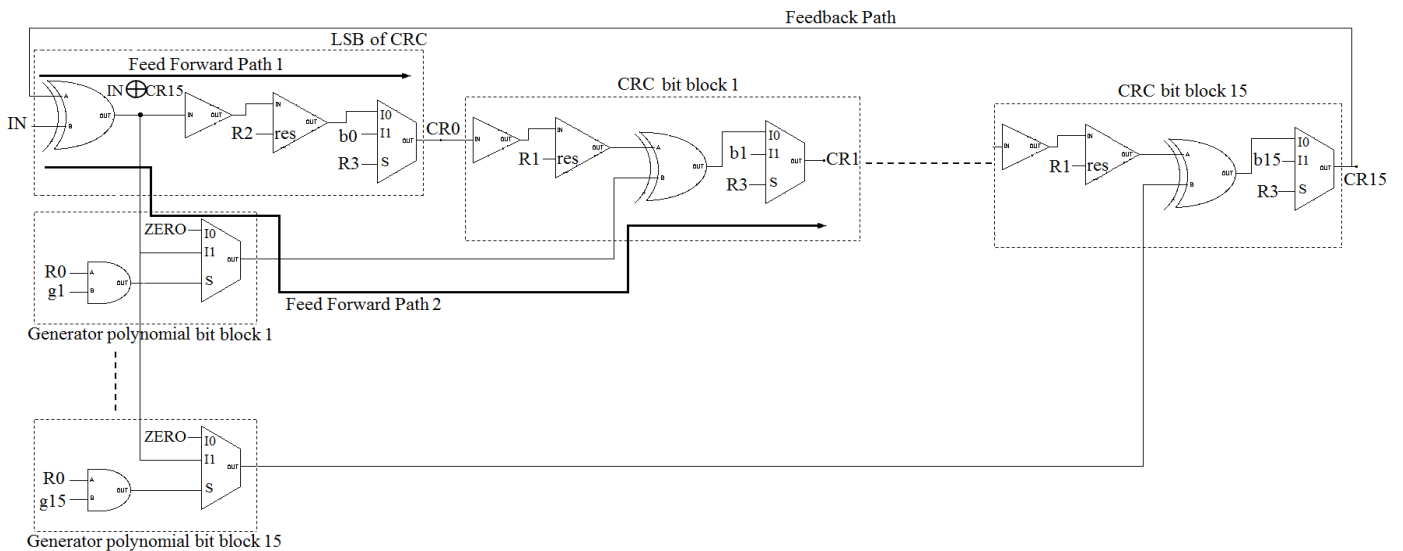The CRC value is appended to a message bit stream in the
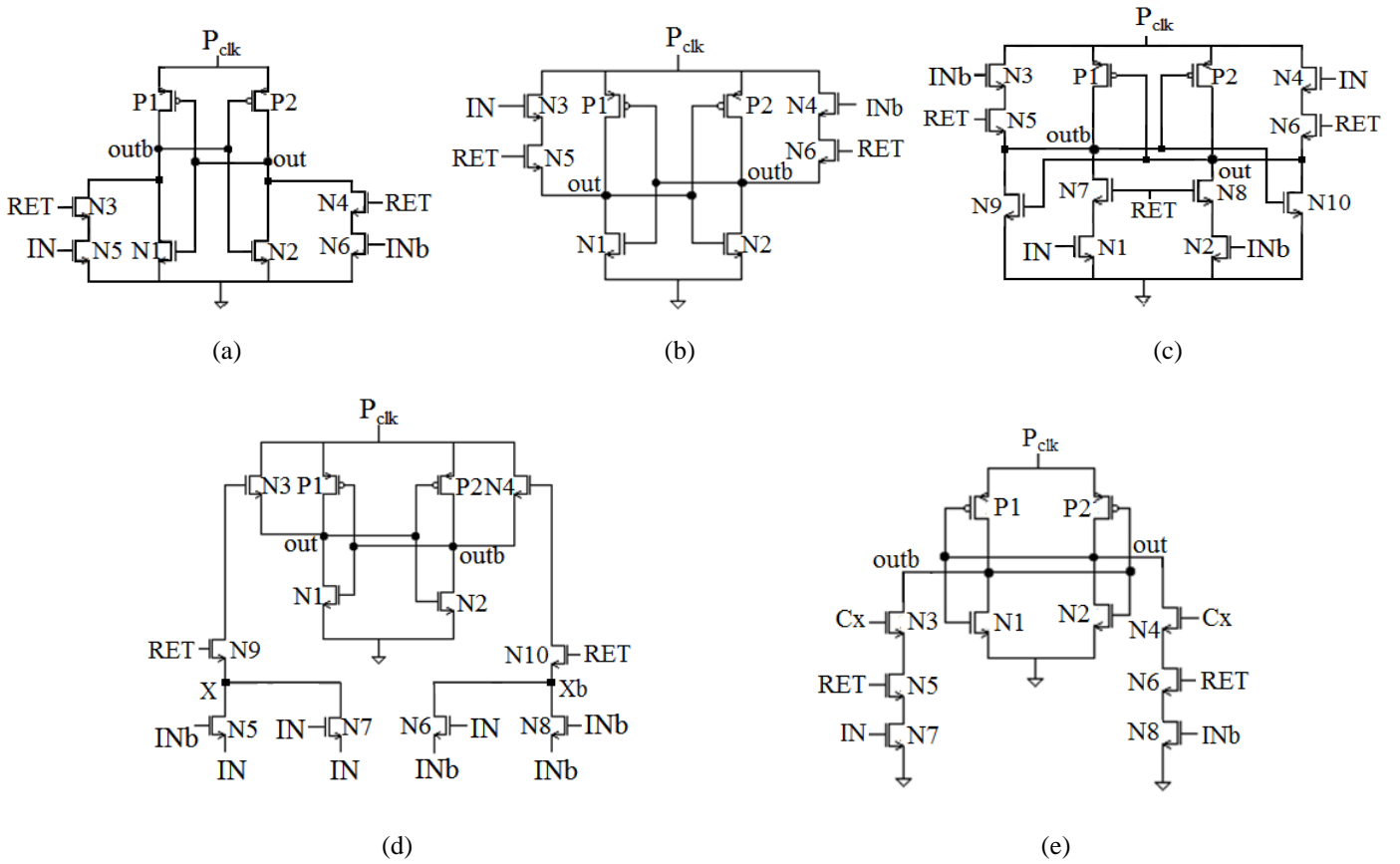

Fig. 8.   CRC Datapath.

Fig. 9. Adiabatic Retain Buffer Logic (a) IECRL (b) PFAL (c) EACRL (d) CPAL (e) CAL.

register unit. Typically, a message bit stream is stalled using a delay cell comprises of four adiabatic buffer logic gates to synchronise it with the CRC value which has a latency of four gates. A single-bit register comprises of four buffer logic stages connected in a cascade manner. The first three stages consist of a buffer logic as shown in Fig. 3-5 and the last stage consist of a novel retain buffer logic. Fig. 9 shows the retain buffer logic circuits for all the five adiabatic designs. The 'RET' is an active low input. It performs a function of retaining the final CRC value using the wait signal, R4 from the decoder. As soon as the computation is over, the RET input is zero and cut-off the two output nodes from the power-clock and the ground respectively. Thus, the logic value gets retained because of the cross-coupled nMOS and pMOS transistors.

In the case of a dual-rail EACRL logic, duplicate retain transistors were not sufficient. As the logic suffers from coupling effect, due to the absence of nMOS cross-coupled transistors, both the output nodes get coupled, when RET input goes low. Thus, two extra cross-coupled nMOS transistors, N9 and N10 are used as shown in Fig. 9 (c). The cross-coupled transistors pair P1, N9 and P2, N10 reduces the coupling effect and helps in providing the complementary output signals at the two output nodes. Conventionally, to construct a 1-bit register using a single-phase and 2-phase adiabatic logic, two buffer stages are required. Due to the synchronisation issue and using the design for the multi-phase power-clocking scheme, the number of stages used in a single bit register of CRC is twice the conventional case. Nevertheless, the overhead is not twice since the number of transistors used to implement a logic gate is more than 4-phase design [30].

6. Simulation Results

For meaningful simulations and to compare CRC implementation using different adiabatic logic designs, the transistor sizes were set to the technology minimum for high energy efficiency [31]. The simulations were done using Spectre simulator in Cadence EDA tool based on TSMC 180nm CMOS process technology at 'Typical-typical (TT) process corner.

For a single-phase and 4-phase, each power-clock is generated using the trapezoidal wave, ramping from 0V to V_{DD}, having equal duration of Evaluation (E), Hold (H), Recovery (R) and Idle (I) periods as shown in Fig. 2. Hence the ramping time (T_r) of the power-clock is one-quarter of the power-clock time-period ($T_{CLK,1-phase/4-phase}$). In case of 2-phase clocking, due to the non-overlapping requirement of the power-clock the Idle period (I) is three times that of the Evaluation, Hold or Recovery period. Hence the ramping time (T_r) of the 2-phase power-clock is one-sixth of the power-clock time-period ($T_{CLK,2-phase}$). Because the adiabatic and non-adiabatic design do not share the same ramping time, the clock frequency of the non-adiabatic implementation, is chosen such that its frequency of operation is same as that of an adiabatic implementation, keeping the rise time and fall time constant across the chosen frequency range. For example, for a ramping time of 2.5ns, the time period of one power-clock cycle is 10ns, thus, the clock period for the non-adiabatic implementation is taken as 10ns with constant rising and falling time of 10ps. To measure the energy dissipation and avoiding excessive data dependencies, the average energy per computation was measured for ten random input combinations. It is measured at various frequencies ranging from 1MHz to 100MHz, load capacitances, supply voltage scaling and PVT variations for all the five adiabatic and non-adiabatic CRC implementations. Also, the computation time in terms of clock-cycles for various message word-lengths was extrapolated. In the end, a comparison at the system level, comprises of the

7

power-clock generator, between adiabatic logic families were done and energy saving percentage is calculated for each of them.

6.1 Impact of Frequency on Energy Dissipation

The energy per computation at varying power-clock frequencies are measured for an output load capacitance of 10fF connected at the output of the register unit. Fig. 10 shows that the energy of all the adiabatic implementations outperforms the non-adiabatic implementation significantly. Energy Saving (ES) is calculated which is defined as the difference in the energy consumption of non-adiabatic and adiabatic implementations divided by the energy dissipation of the non-adiabatic implementation. The formula for "*Energy Saving Percentage*" (ESP) is given by (3)
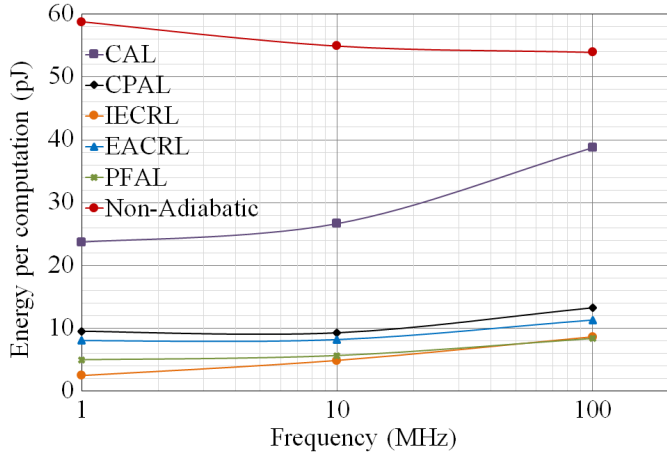
Fig. 10. Energy per computation at various power-clock frequencies.

$$ESP = \frac{E_{non\text{-}adiabatic} - E_{adiabatic}}{E_{non\text{-}diabatic}} \times 100 \qquad (3)$$

The energy saving is calculated excluding the energy dissipated by the power-clock generator circuit. Out of the five adiabatic logic designs, PFAL exhibits the maximum ESP at 100MHz of approximately 84.5%. Whereas, for 10 MHz and 1MHz frequencies, IECRL implementation exhibits the maximum ESP approximately 91% and 96% respectively. The energy consumption per computation and the ESP of the five adiabatic logic families working at frequencies are reported in Table 1. The values are calculated under the same condition as reported above.

The single-phase, CAL design is least beneficial in comparison to the other adiabatic implementations. Unlike 2-phase and 4-phase power clocking schemes, in a single-phase cascaded CAL logic, the incoming inputs from the previous stages are always the same phase as the power-clock, except with a small delay. As the wait signal, R4 is supplied to the 'RET' input of 32 retain transistors, its propagation delay increases as the power-clock speed is increased (shorter ramping time). As a result, the input reads the wrong value which gets propagated to the register outputs. Hence, for a shorter ramping time (higher frequency), the sizing of the logic gate generating the 'R4' signal in CAL controller, was done leading to an increase in energy dissipation. On the other hand, the IECRL design shows the minimum energy per computation at a frequency lower than 25MHz approximately, whereas, above 25MHz, PFAL consumes the minimum energy.

Table 1
Energy dissipation per computation for adiabatic logic families and non-adiabatic at various frequencies.

CRC Implementation		Frequency (MHz)		
		1	10	100
Non-adiabatic	Energy	58.81	54.93	53.93
CAL	Energy	23.72	26.65	38.74
	ESP	59.67	51.48	28.17
CPAL	Energy	9.51	9.27	13.25
	ESP	83.83	83.13	75.43
IECRL	Energy	2.46	4.87	8.63
	ESP	95.83	91.13	83.99
EACRL	Energy	8.02	8.19	11.30
	ESP	86.36	85.10	79.05
PFAL	Energy	4.98	5.65	8.37
	ESP	91.54	89.71	84.48

Energy is expressed in pJ. ESP is the energy saving percentage as defined in the text.

6.2. Impact of Load Capacitance on Energy Dissipation

Fig. 11 shows energy per computation against load capacitance at 10MHz. It can be seen that the variation in energy dissipation of the CAL and the IECRL logic against load variation is steeper as compared to the rest of the logic designs presented. At load capacitance greater than 60fF, IECRL crosses the CPAL energy and becomes the second worst after CAL. Out of the five adiabatic designs, the CAL implementation consumes the most energy. It's also worth mentioning that the non-adiabatic design outperforms the CAL logic at load capacitance values greater than 100fF. On the other hand, PFAL consumes the least energy at load capacitance values greater than 20fF. Whereas the advantage of the low energy consumption of the 2-phase CPAL logic, due to zero NAL at the two output nodes, diminishes mainly because of the high computation time incurred by the CRC datapath.
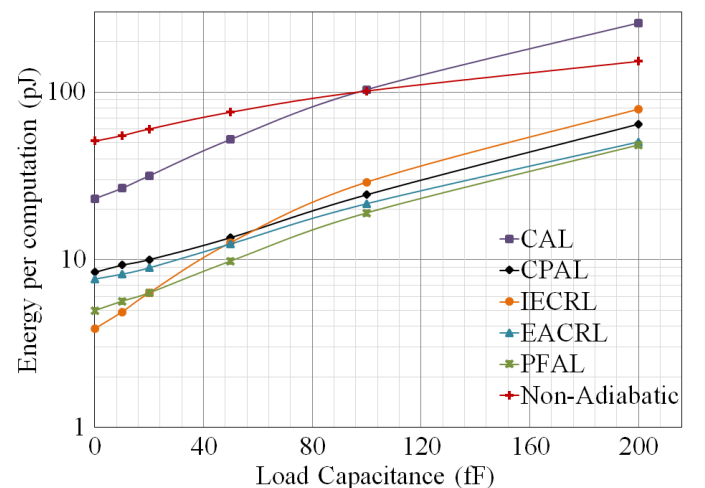
Fig. 11. Energy per computation at varying load capacitance.

Considering the EACRL design, it dissipates more energy in comparison to PFAL and IECRL at lower capacitive load as shown in Fig. 11. But as the load increases beyond 50fF, the advantage of zero NAL in the evaluation phase overpowers its disadvantages of higher input/output node capacitances (due to

8

dual-rail logic) and the coupling effect. Thus, it dissipates less energy than that of IECRL at higher capacitive loading. In addition, when compared to PFAL, due to more transistors EACRL consumes approximately 55% more energy at zero load capacitance. But at 200fF, the load capacitance dominates the internal node capacitance of EACRL and consequently, the difference in PFAL and EACRL energy dissipation reduces, dissipating approximately 4.3% more energy than PFAL.

6.3. Impact of Supply Voltage Scaling on Energy Dissipation

Energy in both adiabatic and non-adiabatic implementations can be reduced by supply voltage scaling according to the quadratic dependence of the energy dissipation on the supply voltage (1) and (4).

$$E_{non\text{-}adiabatic} = \frac{1}{2}\alpha C_L V_{DD}^2 \qquad (4)$$

However, in adiabatic logic, reducing V_{DD} also increases the ON-resistance, R_{ON}, of the transistor in the charging path (5), thus increases the energy dissipation [29]. Hence, the energy benefits of the reduced supply voltage in adiabatic circuits are less.

$$R_{ON} = \frac{1}{K(V_{GS} - V_t)} \qquad (5)$$

Where K is $\mu C_{ox}W/L$. As long as V_{DD} is above than V_t, the energy dissipation is given by

$$E_D = \frac{C_L^2 V_{DD}}{KT_r}\left(\frac{V_{DD}}{V_{DD} - V_t}\right)$$

$$E_D = \frac{C_L^2 V_{DD}}{KT_r}\left(1 + \frac{V_t}{V_{DD}}\right) \qquad (6)$$

Assuming negligible NAL and substituting (4) and (6) in (3), the effect of voltage scaling on ES in an adiabatic circuit can be derived (7).

$$ES = 1 - \frac{\beta}{V_{DD}}\left(1 + \frac{V_t}{V_{DD}}\right) \qquad (7)$$
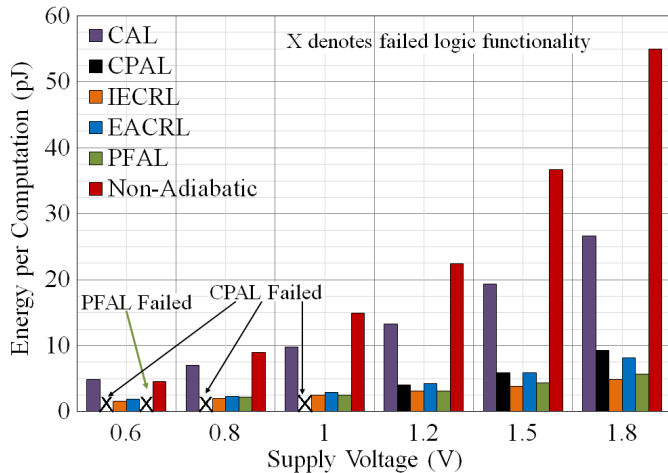
Where β is $2C_L/\alpha KT_r$.

Fig. 12. Energy per computation at varying supply voltage.

Fig. 12 shows the effect of voltage scaling on energy per computation for various adiabatic and non-adiabatic CRC

implementations at a 10MHz frequency and 10fF load capacitance. From (7) and Fig. 13, it can be seen that the adiabatic techniques largely suffered from voltage scaling in terms of ESP and functionality. PFAL and IECRL show a similar reduction in ESP as the voltage is scaled down, except the former malfunction at 0.6V (voltage closer to the threshold voltage). Also, due to the higher voltage drop of pass transistors in CPAL, it malfunctions at 1V and less. Thus, it makes CPAL highly vulnerable logic at lower voltages. As expected CAL logic shows minimum ESP and goes below zero, approximately 5% at 0.6V meaning that the energy dissipation of the non-adiabatic implementation becomes less than that of the CAL logic.

It can be summed up that, the ESP of using adiabatic logic shows a steeper response at supply voltage less than 1.2V. In addition, the reduction in supply voltage will also degrade the noise margin both in non-adiabatic and adiabatic implementations. Thus, for adiabatic logic families we propose an optimal range for the supply voltage scaling, that we call "*Adiabatic Voltage Scaling Range*" for better ESP and proper functionality which is stated as;
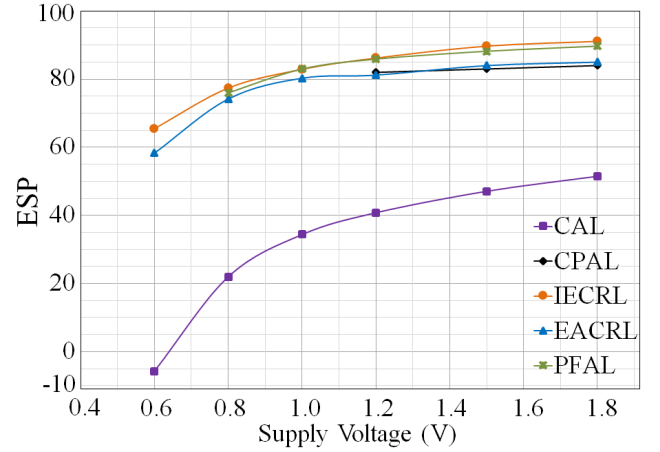
$$V_{DD} \geq 2V_t \qquad (8)$$

Fig. 13. ESP at varying supply voltage.

6.4. Impact of Process Corner Variation on Energy Dissipation

The robustness of the CRC design using adiabatic logic implementations against the process, voltage and temperature variation is investigated by running the corner analysis in Analog Design Environment (ADE). All the CRC implementations were simulated for five corners to ensure correct operation. Fig. 14 shows the energy per computation measured for the adiabatic and non-adiabatic designs at 10MHz and 10fF load capacitance.

Temperature plays an important role in energy dissipation of the adiabatic circuits. Due to the dependency of R_{ON} on adiabatic energy dissipation, the increase in temperature causes R_{ON} to increase. Thus, causing the adiabatic logic to dissipate more at a higher temperature. The worst case energy dissipation was measured for the Fast-Fast 'FF' process corner at a 1.98V supply voltage and 100°C temperature. Similarly, for the best case, slow-slow 'SS', 1.62V and 0°C is considered. Whereas for the skewed corners slow-fast 'SF', and fast-slow 'FS', we simulated the designs for 1.62V and 100°C temperature giving energy dissipation close to the SS corner and for the FS corner 1.98V

9

and 0°C, close to the FF corner. For typical-typical corner (TT), 1.8V voltage and the 27°C temperature is the default value.
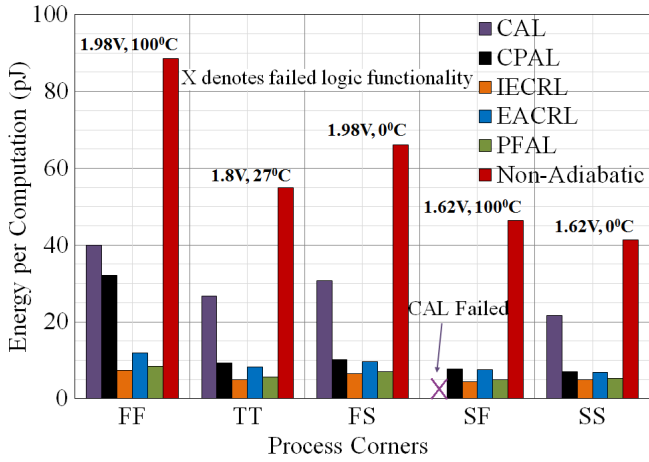

Fig. 14. Energy per computation at five process corners.

In SF corner, the CAL implementation malfunctions, hence its value is not calculated. On the other hand, CPAL design shows a large variation in energy consumption at extreme corners (FF and SS) compared to the other adiabatic logic designs presented. However, out of the five adiabatic CRC implementations, PFAL and EACRL show a constant ESP approximately 90% and 85% respectively at all process corner. Whereas IECRL shows ESP of 85% at FS corner and 91% at rest of the four corners.

6.5. Impact of Message Word-Length on Computation Time

The datapath of the CRC for all the 4-phase and 2-phase 16-bit CRC designs, take 64 power-clock phases for the computation of 16-bit message word-length. An additional seven phases, four by the counter and three by 16:1 multiplexer are required for the message bits to arrive at the LSB bit block of CRC unit, XOR gate input. Another four phases are required by the CRC value to be appended with the message word in the register unit. Thus, the total of 75 power-clock phases, equivalent to 18.75 power-clock cycles, are required by the 4-phase designs for CRC computation. Whereas, for the 2-phase design, 37.5 power-clock cycles are required for computation. Although the single-phase design has the lowest power-clock complexity but requires 75 power-clock cycles in total. Thus, resulting in lowest throughput and highest energy dissipation.
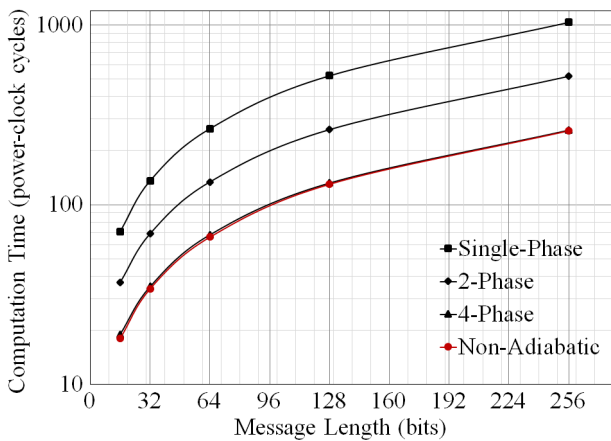

Fig. 15. Extrapolated throughput at varying message word-length.

The non-adiabatic design took 18 clock-cycles, 3/4th less as compared to the 4-phase adiabatic logic designs. This is because the adiabatic implementation of the multiplexer test circuit takes three power-clock phases whereas non-adiabatic takes none. Fig. 15 shows the extrapolated result of the computation time at varying message word-length using the multi-phase power-clock designs and the non-adiabatic design for 16-bit CRC code.

6.6. Power-Clock generation

Unlike static CMOS logic, adiabatic circuits are powered from the clock, requiring a separate "power-clock" supply. A power-clock will consume a significant amount of the energy (analogous to the clock generation in conventional CMOS). It is important to bear in mind that power-clock circuit will be able to supply considerably more circuitry than the CRC presented here. Nevertheless, it is appropriate to consider its energy too, which is often neglected in adiabatic papers. Power-clocks can be generated either using a capacitor-based Stepwise-Charging (SWC) circuit [13-14] or inductor-based resonant circuit [15-16]. Since the inductor based circuit occupies a large area it is not suitable for NFC application. Thus, an SWC based power-clock generator is used, as found in [32]. The complete adiabatic system was designed which comprises of the power-clock generator and the adiabatic core.

The adiabatic core contains the CRC. The required power-clock comes from the power-clock generator. We have designed single-phase, 2-phase and 4-phase power-clock generator using 2-step charging circuit. To generate 2-phase power-clock two 2-step charging circuits are required. Similarly, for 4-phase power-clock, four 2-step charging circuits are required. For a single-phase, only one 2-step charging circuit is required and the auxiliary clocks are supplied using a trapezoidal power source. What also has to keep in mind that generating power-clock of the same ramping time for 2-phase and single/4-phase clocking scheme, the power-clock frequency is different (see Fig. 2).

Table 2
Energy dissipation per computation by an adiabatic system including a power-clock generator and non-adiabatic design.

Adiabatic Logic Techniques	E_{PCG} (pJ)	$E_{TOTAL\ SYSTEM}$ (pJ)
Non-Adiabatic	--	54.93
CPAL	101.03	107.27
CAL	113.55	134.82
PFAL	44.17	48.53
EACRL	48.39	59.74
IECRL	29.36	36.93

The simulation was performed for a ramping time of 25ns for the power-clocks with supply-voltage 1.8V V_{DD} and 10fF capacitive load attached to the output of an adiabatic core. The reference CLK for generating the power-clock frequency is taken to be 40 MHz and 60MHz for single/4-phase and 2-phase clocking scheme respectively for generating a power-clock of 25ns ramping time. The frequency of operation for non-adiabatic is taken to be 10MHz. The tank capacitance chosen for all the logic families is 5pF. In the 2-step-charging circuit,

keeping the length of the switches minimum, the width of the switches are taken based on the logic families. For a single phase and 2-phase designs, the width of the pMOS and nMOS is chosen to be 1u and 0.5u respectively whereas for PFAL and IECRL, the width is taken as 0.25u for all the transistors. In case of EACRL, due to its dual evaluation network, the pMOS width is taken as 4u and nMOS width is taken as 2u.

Table 2 reports the energy consumed by the adiabatic system including the power-clock generator and the non-adiabatic design for computing the CRC value. In comparison to the non-adiabatic design, only PFAL and IECRL show a decrease in energy dissipation. It is also worth to be noted, that the energy consumption of the signal generator for SWC has not been considered. This is because its energy remains constant for all system size [29]. In addition, the energy of the adiabatic system can be made lower by using step charging circuits with more than 2-steps [14]. The comparison between adiabatic and non-adiabatic in Table 2 reported an unfavourable outcome for the adiabatic circuit since the dissipation of the clock generator and distribution network present in almost all the non-adiabatic circuits are not considered in this comparison. To undertake such a comparison would be beyond the scope of this paper.

Based on the simulation results for a 16-bit message word-length for 16-bit CRC, the performance trade-offs of the multi-phase adiabatic logic design is tabulated in Table 3. The only difference in the structure of PFAL and IECRL logic is in the connection of the evaluation network. They both have the same and a minimum number of transistor counts. On the other hand, the CPAL logic design uses approximately 40% more transistors compared to PFAL and IECRL whereas, CAL and EACRL design consume 25% and 20% more transistors respectively. This increase of CPAL transistor counts is because of the twice the number of buffers needed in the register unit due to the synchronisation issue.

The impact of increased message word-length is more on the throughput of single-phase and 2-phase designs rather than on the area for all the five adiabatic logic techniques. The area is mostly incurred by the register unit rather than the other CRC components. Since the CRC datapath implementation requires four cascade logic for a single bit CRC bit-slice, the advantage of single phase (CAL) and 2-phase (CPAL) designs in terms of transistor count and throughput diminishes. It can be seen that the 4-phase scheme are more efficient in terms of area and throughput. They also show high robustness against PVT variations. The power-clock complexity depends on the number of SWC circuits needed to generate the power-clock and the area utilized by the controller circuitry. Considering the

generation of single-phase power-clock generator requires one SWC circuits, two flip-flops and two 2-inputs logic gates for the controller. Whereas 2-phase power-clock generator requires two SWC circuits, three flip-flops and nineteen 2-input logic gates. On the other hand, all 4-phase power-clock generator is designed using four SWC circuits, two flip-flops and eight 2-inputs and single-input logic gates.

6. CONCLUSION

This paper presents a thorough comparison of the performance of single-phase, two-phase and four-phase adiabatic logic through the implementation of a 16-bit CRC for a 16-bit message word-length. A methodology for selecting generically "efficient" design is based on achieving optimum trade-offs between computational energy, area, computation time, complexity and robustness.

Though the CAL complexity is least due to the use of single-phase power-clocking scheme, its performance in terms of computation time, throughput, latency, energy dissipation of the core and system and robustness is least compared to the rest of the adiabatic logic presented. The 4-phase adiabatic logic designs outperform the single-phase and 2-phase designs. The 4-phase EACRL has the highest area due to the complex evaluation network compared to PFAL and IECRL.

Significant differences in functionality and robustness against voltage scaling and PVT variations among multiphase adiabatic implementations were found. The benefit of using adiabatic logic deteriorates for supply voltage less than 1.2V. Thus an optimal range for the supply voltage scaling is proposed for better ESP and proper functionality. The CRC implementation using CAL fails the functionality test at SF corner. The sensitivity to PVT variations of adiabatic designs shows that 4-phase designs are more robust and energy efficient compared to the single-phase and 2-phase designs. Overall, IECRL shows the best performance under voltage scaling followed by PFAL implementation.

Energy saving deteriorates when the power-clock generator is considered. The results show that only IECRL consume less energy compared to the non-adiabatic design. It is anticipated that at high capacitive load (Fig. 13), PFAL will show best energy performance at the system level. The system energy comparison in Table 2 and performance Table 3 between adiabatic logic techniques, will enable the designers to use quantitative information in selecting the required n-phase adiabatic logic to design an effective feedback system.

Table 3
Performance Trade-offs between multi-phase adiabatic 16-bit CRC Implementation for a 16-bit message word-length.

Adiabatic Logic Techniques	Area (in terms of transistor counts)	Robustness against PVT Variations	Computation Time (power-clock cycles)	Circuit Complexity	Power-clock Complexity
CPAL	3012	Medium	75	High	Medium
CAL	2696	Low	37.50	Medium	Low
PFAL	2150	High	18.75	Low	High
EACRL	2582	High	18.75	High	High
IECRL	2150	High	18.75	Low	High

REFERENCES

[1] J. Pang, K. Andrews and Leigh Torgerson, Clarification for CCSDS CRC-16 Computation Algorithm, Section 332M-Communication Networks, 2006.

[2] P. Koopman, & T. Chakravarty, Cyclic redundancy code (CRC) polynomial selection for embedded networks. International Conference on Dependable Systems and Networks, (2004), 145–154.

[3] T. V. Ramabadran, & S. S. Gaitonde,. Tutorial on CRC Computations. IEEE Micro, 8(4), (1988) 62–75.

[4] G. Albertengo, & R. Sisto, Parallel CRC Generation. IEEE Micro, 10(5), (1990) 63–71.

[5] C. E. Kennedy and Mehran Mozaffari-Kermani, Generalized Parallel CRC Computation on FPGA 28th Canadian Conference on Electrical and Computer Engineering, (2015) 107-113.

[6] Walma, M., Pipelined cyclic redundancy check (CRC) calculation. Proceedings - International Conference on Computer Communications and Networks, ICCCN, (2007) 365–370.

[7] Sachin Maheshwari, Viv A. Bartlett, I. Kale, 4-phase resettable quasi-adiabatic flip-flops and sequential circuit design. 12th Conference on Ph.D. Research in Microelectronics and Electronics, PRIME 2016. (2016).

[8] M.C. Knapp, P. J. Kindlmann, M. C. Papaefthymiou, Design and Evaluating of Adiabatic Arithmetic Units, Analog Integrated Circuits and Signal Processing, 14, (1997) 71-79.

[9] G. Yemiscioglu, & P. Lee, Very-large-scale integration implementation of a 16-bit clocked adiabatic logic logarithmic signal processor. IET Computers & Digital Techniques, 9, (2015) 239–247.

[10] Ph. Teichmann, M. Vollmer, J. Fischer, B. Heyne, J. Götze, D. Schmitt-Landsiedel, Saving potentials of Adiab. Logic on system level: A CORDIC-based adiabatic DCT 12th International Symposium on Interated Circuits, (2009) 105-108.

[11] Identification cards - Contactless integrated circuit(s) cards – Proximity cards - Part 3: Initialization and anticollision, ISO/IEC Std. FCD 14443-3, 2001. http://www.icedev.se/proxmark3/docs/ISO-14443-3.pdf

[12] J. G. Koller and W. C. Athas, Adiabatic Switching, Low Energy Computing, and the Physics of Storing and Erasing Information, Proceedings of the 2nd Workshop on Physics and Computation, (1992) 267-270.

[13] L. Svensson, J. Koller, Driving a capacitive load without discharging fCV^2. Proceedings of the IEEE Symposium on Low-Power Electronics, (1994) 100-103.

[14] H. S. Raghav, V. A. Bartlett, & I. Kale, Investigation of stepwise charging circuits for power-clock generation in Adiabatic Logic. 12th Conference on Ph.D. Research in Microelectronics and Electronics, PRIME (2016), 1–4.

[15] W. C. Athas, J. G. Koller, and L. Svensson, An Energy-Efficient CMOS Line Driver Using Adiabatic Switching. Proc. Of IEEE Great Lakes Symposium on VLSI, (1994) 196-199.

[16] D. Maksimoviü and V. G. Oklobdžija, "Integrated Power Clock Generators for Low Energy Logic", 26 the Annual IEEE Power Electronics Specialists Conference (PESC'95), 1, (1995) 61-67.

[17] W. C. Athas, L. J. Svensson, J. G. Koller, N. Tzartzanis and E. Ying-Chin Chou, Low-Power Digital Systems Based on Adiabatic switching Principles, IEEE Transactions on VLSI Systems, 2(4), (1994) 398-407.

[18] John S. Denker, A Review of Adiabatic Computing, IEEE Symposium Low Power Design, (1994) 94-97.

[19] Y. Moon and D.K. Jeong, An Efficient Charge-Recovery Logic. IEEE Journal of Solid-State Circuits, 31(4), (1996) 514-522.

[20] A. Vetuli, S.D. Pascoli and L.M. Reyneri, Positive Feedback in Adiabatic Logic. Electronics Letters, 32(20), (1996) 1867-1869.

[21] L. Varga, F. Kovacs, & G. Hosszu, An efficient adiabatic charge-recovery logic. SoutheastCon 2001. Proceedings. IEEE, Clemson, SC, (2001) 17–20.

[22] D. Maksimovic, and V. G. Oklobd̆zija, Clocked CMOS adiabatic logic with AC power supply. Solid State Circuit Conference,(1995) 370-373.

[23] V. G. Oklobdzija, D. Maksimovic, and F. Lin, Pass-transistor adiabatic logic using single power-clock supply. IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 44(10), (1997) 842–846.

[24] Jianping, H., Lizhang, C., and Xiao, L., A New Type of Low-Power Adiabatic Circuit with Complementary Pass-transistor Logic. ASIC, Proceedings. 5th International Conference On, 2, (2003) 1235–1238.

[25] Yangbo Wu, Huiying Dong,Yi Wang, and Jianping Hu, Low-Power Adiabatic Sequential Circuits Using Two-Phase Power-Clock Supply. 6th International Conference on ASIC, (2005) 185-188.

[26] R. Tessier, D. Jasinski, A. Maheshwari, A. Natarajan, Weifeng Xu and W. Burleson, An energy-aware active smart card. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 13 (10), (2005) 1190-1199.

[27] W. W. Peterson, and D. T. Brown, Cyclic Codes for Error Detection. Proceedings of the IRE, 49(1), (1961) 228-235.

[28] Ph. Teichmann, J. Fischer, F. Chouard, and D. Schmitt-Landsiedel, Design Issues of Arithmetic Structures in Adiabatic Logic. Adv. Radio Science, 5, (2007) 291-295.

[29] P. Teichmann, Adiabatic logic: Future Trend and System Level Perspective, Springer Science & Business Media, 34, 2011, 76-77.

[30] Sachin Maheshwari, V.A.Bartlett and Izzet Kale, Adiabatic Flip-Flops and Sequential Circuit Designs using Novel Resettable Adiabatic Buffers. 23rd European Conference on Circuit Theory and Design (ECCTD), (2017) 1-4.

[31] J. Fischer, E. Amirante, F. Randazzo, G. Iannaccone and D. Schmitt-Landsiedel, Reduction of the energy consumption in adiabatic gates by optimal transistor sizing. International Workshop on Power and Timing Modelling, Optimisation and Simulation, (2003) 309-318.

[32] Himadri Singh Raghav, V.A.Bartlett and Izzet Kale, Energy efficiency of 2-step charging power-clock for adiabatic logic. 26th International Workshop on Power and Timing Modelling, Optimisation and Simulation (PATMOS), (2016) 176-182.

[33] G. Yemiscioglu and P. Lee, 16-Bit Clocked Adiabatic Logic (CAL) Leading One Detector for a Logarithmic

Signal Processor. 8th Conference on Ph.D. Research in Microelectronics & Electronics PRIME, (2012), 1-4.

[34] Standard ECMA-340 - Near Field Communication-Interface and Protocol (NFCIP-1), 3rd edition, 2013.

Sachin Maheshwari received the BTech Degree in electrical and electronics Engineering 2007 from ICFAI Tech, Hyderabad, India and Master of Engineering Degree in Microelectronics from Birla Institute of Technology and Science (BITS), Pilani, India in 2009. Currently, he is a Ph.D. student at University of Westminster, London, UK in Applied DSP and VLSI Research Group, Department of Engineering.
His research interest is in Low Power Digital VLSI Design.

Viv Bartlett obtained his BScEng (Hons) at Imperial College, London and his PhD at the University of Westminster. He is currently a Principal Lecturer at the University of Westminster, a post he has held for the last 10 years, teaching mainly in the fields of Digital Systems and VLSI design.
He has been designing integrated circuits since the 1980s and his research interests are centred on VLSI design and DSP architectures with an emphasis on asynchronous and low-power techniques.

Izzet Kale (M'87) received the B.Sc. (Hons.) degree in electrical and electronic engineering from the Polytechnic of Central London, the M.Sc. degree in the design and manufacture of microelectronic systems from Edinburgh University, Scotland, and the Ph.D. degree in techniques for reducing digital filter complexity from the University of Westminster, London. He is currently a Professor of applied DSP and VLSI systems, and Head of Engineering and Founder and the Director of the Applied DSP and VLSI Research Group, University of Westminster. He is currently working on efficiently implementable, ultralow-power DSP algorithms/architectures, sigma–delta modulator structures and Low Power VLSI Design for secure systems.