#### Purdue University Purdue e-Pubs

**ECE Technical Reports** 

**Electrical and Computer Engineering** 

3-1-1995

# Ultra Low Energy Computing Using Adiabatic Switching Principle

Yibin Ye Purdue University School of Electrical Engineering

Kaushik Roy Purdue University School of Electrical Engineering

Follow this and additional works at: http://docs.lib.purdue.edu/ecetr

Ye, Yibin and Roy, Kaushik, "Ultra Low Energy Computing Using Adiabatic Switching Principle" (1995). *ECE Technical Reports*. Paper 114.

http://docs.lib.purdue.edu/ecetr/114

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

## Ultra Low Energy Computing Using Adiabatic Switching Principle

Yibin Ye Kaushik Roy

TR-EE 95-7 March 1995



School of Electrical Engineering Purdue University West Lafayette, Indiana 47907-1285

## Ultra Low Energy Computing Using Adiabatic Switching Principle

Yibin Ye and Kaushik **Roy** Electrical Engineering Purdue University West Lafayette IN 47907-1285, USA.

Contact person: Kaushik Roy Ph: 317-494-2361 Fax: 317-494-6440 e-mail: kaushik@ecn.purdue.edu

## Ultra Low Energy Computing Using Adiabatic Switching Principle

#### Abstract

This paper presents a new family of logic gates for ultra low energy computing using pulsed power CMOS logic. The logic gates use the principles of adiabatic switching principle and results show that in typical cases 90% of the energy can be recovered with operating frequency around 1MHz. Constant capacitance condition is enforced in our designs so that signals' energy can be efficiently recycled in the chip. We also present a detailed analysis and modeling of energy dissipation. The models were experimentally validated using the circuit simulator SPICE. We also simulated a serial adder (mod 2) implemented using the reversible logic principle. The design can recover 85% of energy while operating at a frequency of 1.67MHz. For a naturally reversible buffer chain, 95% of energy can be recovered at 1.1 MHz.

#### **1** Introduction

With the recent trend toward portable communication and computing, power dissipation has become one of the major design concerns along with area and performance. Research to reduce power dissipation at various levels of design abstraction has started in earnest to achieve ultra-low power with optimum performance [8, 9, 10, 11, 12, 13]. In this paper we will consider circuit level techniques for low-energy computation using the principles of adiabatic switching. Energy-recovery using adiabatic switching is a relatively new idea [5, 6, 7]. Although the concept of reversible logic and zero-energy computing can be traced back to the early 1970s [1, 2, 3, 4], the attempt to realize the concept in electronic circuits is a new endeavor.

In the early papers, researchers largely focused on discussing the possibility of having physical machines which consume zero energy while computing and tried to find the lower bound of energy consumption. If the physical processes associated with the computing are non-dissipative, the natural law requires that the physical entropy is conserved. Entropy conservation means that the processes are physically reversible. One of the conclusions from the earlier studies is that the abstract logic operations composing the computing tasks must be reversible, i.e., the information entropy must be conserved, in order to be performed by physically non-dissipative hardware. Nevertheless, logical reversible or nonreversible hardware.

A non-dissipative circuit requires that the switching activities in each element doing computation is non-dissipative. Under certain conditions, an element of a circuit can work in a nearly non-dissipative fashion (see Section 2). However, these conditions are not easy to satisfy when all elements work together to perform the desired tasks. Theoretically, it is possible to achieve non-dissipative computing for reversible logic. However, in general, circuits are not reversible.

In adiabatic circuit designs presented in [5] and [7], every switching operation is enforced to be nearly non-dissipative. However, inverse logic functions are required to perform the energy-recovery. Since many conventional logic primitives are irreversible, addressing the design issue at gate level is not sufficient. The circuit topologies can be radically different.

In [6], a logic family has been presented, which dissipates a small amount of energy per operation. Inverse logic is not required, and hence, the design can largely simulate the conventional CMOS circuits. However, we believe that the dissipation can be significantly decreased. We present our designs in Section 3, which consume less energy in general.

By explicitly defining control schemes in the energy-recovery path, we present a logic primitive family, which does not require inverse logic and hence functions like conventional logic gates. The logic primitives work in a nearly adiabatic-switching fashion. We do not reject the reversible logics. In fact, we also present the reversible logic primitives which can be used with other partially reversible logic gates. We also give a detailed analysis and modeling of energy dissipation principle. The models are experimentally validated using the analog simulations. Measurement on a 6-buffer chain and a serial adder (mod 2) with feedbacks have been performed to validate our designs.

The rest of the paper is organized as follows. Section 2 discuss the basic operations and specific problems in a energy-recovery circuit. In Section 3 we present logic designs with partially reversible logic. Section 4 considers designs with reversible logic. The analysis of the dissipative mechanisms in our designs is presented in section 5. Section 6 presents the results of energy dissipation using SPICE simulations for the basic circuit blocks developed in Sections 3 and 4. The conclusions are given in Section 7.

#### 2 Background

In CMOS circuits charges are fed from the power supply, steered through MOSFET devices, and then dumped into the ground terminal. To change a node's voltage with associated capacitance C, as shown in Figure 1(a),  $V_{dd}Q(=CV_{dd}^2)$  of energy is extracted from the  $V_{dd}$  terminal. Half of the energy,  $\frac{1}{2}CV_{dd}^2$ , is stored in capacitance temporarily, and the other half is dissipated in the path. Although energy is dissipated in the channel resistance and wire resistance, the amount of dissipated energy,  $\frac{1}{2}V_{dd}Q$ , only depends on the voltage and the amount of charge which flows through, and is independent of the resistance. Later, when this node is connected to the ground, the stored energy is again dissipated. In a cycle, all  $V_{dd}Q$  of energy is converted into heat.

Whenever there is a conducting path, energy is dissipated if there is a potential difference between the endpoints of the path (e.g. power supply terminal and the internal node in the circuit). The amount of energy dissipated is  $\Delta V_{avg}Q$ , where  $\Delta V_{avg}$  is the average potential difference between the endpoints and Q is the amount of charge which flows through the path. This implies that the circuit should switch adiabatically to avoid the energy dissipation. Let us consider Figure 1(b), where the power supply terminal swings gradually from 0 to  $V_{dd}$ , stays at  $V_{dd}$  for a while and then swings back to 0. If x = 0, and the initial charge on capacitance C is 0, then node y follows the power supply to  $V_{dd}$  during its upward swing. Similarly, while discharging, potential of node y follows the power supply terminal swinging gradually from  $V_{dd}$  to 0, so that there is little potential difference across the path (from the power supply terminal to node y) in the whole transition process. Hence only a small amount of energy is dissipated. The question is: Can an internal node's voltage level follow the change in the power supply terminal? Let At be the transition time of the power supply terminal from 0 to  $V_{dd}$  (or from  $V_{dd}$  to 0), and the time constant in a conducting path is RC, where R is the effective resistance in the path and C is the effective load capacitance. If the transition at the power supply terminal is sufficiently slow, i.e.,  $At \gg RC$ , then the voltage drop between the power supply terminal  $\Phi$  and node y is small at any time instant



Figure 1: Basic recovery processes

during the transition, and hence, there is very low power dissipation. A simple model to estimate the power dissipation in this case is (for proof see section 4):

$$E_{dissipation} = \left(\frac{RC}{\Delta t}\right) C V_{dd}^{2} \tag{1}$$

Since  $RC \sim \ln s$  for a moderate fanout, and  $At \sim 1/f$ ,  $E_{dissipation}$  is very small when the operating frequency f < 10MHz.

So far, the load is simply viewed as a dummy capacitance. In a circuit, an output of a gate drives other gate(s). Let us consider Figure 1(c). The power supply/clock waveform  $\Phi_1$  can be divided into four phases.  $\Phi_1$  is in the *idle phase* when  $\Phi_1 = 0$  and is in the *evaluation phase* when  $\Phi_1$  goes up from 0 to  $V_{dd}$ . The time interval in which  $\Phi_1$  remains high, i.e.  $\Phi_1 = V_{dd}$  is defined as the *hold* phase. Finally, the phase when  $\Phi_1$  goes down from  $V_{dd}$  to 0 is defined as the restoration phase. After evaluation: the output.  $y_1$  of the first stage must

hold for a while in order to be sampled by the second stage. Meanwhile, there must exist some device, say, a transmission gate denoted by a "T-gate" in Figure 1(c), to isolate the input x and the output  $y_1$ . Otherwise x can not change its value as long as  $y_1$  needs to hold its value, and  $y_1$  can not change its value as long as  $y_2$  need to hold its value, which implies that an input has to be held constant until the signal propagates all the way to the last stage of the logic level. A consequence of the isolation is that the charge may not flow back to the power/clock terminal along the original charging path. Thus, another path has to be created to let the potential of  $y_1$  follow  $\Phi_1$  going down to 0. However, unlike the charging path which is controlled by the inputs, the turning on of the discharging path depends on the logic value of  $y_1$ . When  $y_1$  is 1, charge can flow back to power terminal through the path shown in Figure 1(c) during the restoration phase. If  $y_1$  is 0, this path must be open to prevent current leaking from  $\Phi_1$  to  $y_1$ , which not only will consume energy but might set a wrong logic value to node  $y_1$  as well.

Then the new question is: How to detect the voltage level at the output of a gate and how to control the recovery path? It is basically the control schemes which differ in the various designs currently available. The output voltage to control the restoration path is simple and is shown in Figure 1(d). Such control schemes are referred to as the self-control schemes.

Self-control is not perfect because the voltage controlling the channel of M is also varying along with the voltage levels being controlled. The gate, the source, and the drain ends can not have the same potential at the same time when the channel is conducting. The voltage drop across the channel of the control transistor,  $V_{ds} \approx V_t$ , causes as much as  $CV_tV_{dd}$  of energy dissipation in each restoration. Note that the control transistor M is equivalent to a diode since its gate terminal and drain terminal are connected together. The perfect controlling signal should hold a correct and constant voltage during the restoration process. Let us consider Figure 1(e). Such a controlling signal  $C_y$  can be inversely generated from the next stage signals, i.e., from the inverse logic function  $F_2^{-1}$ , if the logic function  $F_2$  itself is reversible. Since  $C_y$  is isolated from node y and is in a phase later than that of y, it holds constant ( in hold phase ) during the restoration phase of node y.

The power supply/clock waveforms can be generated using some simple schemes [5, 7] which consists of two stages, as shown in Figure 2. The first stage is the DC power supply, and the second stage generates alternating current/clock waveforms, which is controlled by external clock signal(s) to maintain the constant frequency. If the circuit viewed by the power/clock generator can be modeled as a simple and constant capacitive load, the entire system is effectively an RLC resonator. The dissipated energy is replenished by the first stage DC power supply by restoring the peak voltage of the second stage circuit to  $V_{dd}$  level in each cycle. However, there might be a problem in some existing designs since the number of gates charging and discharging varies cycle by cycle during computation, which violates the constant capacitance condition. When the amount of energy feeding into the circuit and



Figure 2: A framework for power supply and circuit

recovering from it behave like a random process, current waveforms and the frequency in the second stage power supply will be unstable, which in turn forces the first stage DC power supply to rectify the current. Restoration of the current waveform is accompanied by power dissipation. Constant capacitance condition will be enforced in all our designs as described in the next section.

#### 3 Designs with partially reversible logic

We have discussed in Section 2 that the adiabatic operation can be implemented if ideal charging and discharging paths are set up correctly. If every logic function primitive in the circuit is reversible, the controlling signal for the discharging path can be produced by an inverse function with correct timing. Theoretically, only reversible logic operations can be performed without accompanying energy dissipation [1, 2, 3]. Unfortunately, many common logic primitives, like NAND, NOR, and XOR are irreversible. Irreversibility does not mean that all energies involved have to be given up. The theoretical lower bound for an isolated irreversible operation is kT. However, in CMOS technology, the lower bound is related to the threshold voltage  $V_t$  and the amount of charges to build the voltage level to  $V_t$ .

We present two designs (in Figure 3 and Figure 4), and their modified versions, which are not based on reversible logic primitives. However, the designs recover most of the energy involved in the operations. The obvious advantage of these designs includes the simplicity as well as practicality for implementation. Both designs are essentially based on the self-control scheme, where recovery path is controlled by a signal generated from output. Our design goal is to achieve lowest possible dissipation without requiring inverse logic functions.

In [6], a pair of diodes are included in each gate to create a correct recovery path (similar to Figure 1(d).  $CV_tV_{dd}$  of energy is dissipated per operation, where C is the load capacitance it drives. Since the average fan-out in actual circuits is  $3 \sim 4$ , we modified the design to allow  $C_sV_tV_{dd}$  of energy to be dissipated per operation, where  $C_s$  is the smallest gate capacitance. On average C,  $\sim \frac{1}{3}C$ .

A basic logic gate (Buffer/Inverter) is shown in Figure 3(a). It uses differential signaling: both input signal and its complement are required and an output signal and its complement are generated. The circuit consists of two branches. In each cycle, one branch is charging and the other is not. Thus, the constant load capacitance condition is satisfied. Without loss of generality, let us assume x = 0. At the beginning of a cycle,  $\Phi_1$  is at 0 and the gate is in the idle phase. Both outputs y and  $\overline{y}$  are not valid and are equal to 0 at this phase. During the evaluation phase,  $\Phi_1$  swings to  $V_{dd}$ . Node y follows  $\Phi_1$  to high as IS 1 turns on while *ij* stays at 0. The output is now valid and can be sampled by other gates in the subsequent stages it is driving. Importantly, the inputs and the outputs are in effect isolated at this phase because the isolation transistor IS2 is turned off and IS1 remains on and doesn't affect the output y in its branch. Thus, the input need not remain valid and can restore to 0's while the output of the gate still hold its logic values. This is the hold phase. Subsequently, it is the restoration phase, in which y follows  $\Phi_1$  and returns to 0, since the controlling gate C1 is turned on. In the other branch, C2 is turned off to prevent leaking current from  $\Phi_1$ to ij. Note that the controlling signals C1 and C2 generated from y and  $\overline{y}$  are buffered and have the different timing with y and  $\overline{y}$ . Since the gate outputs y and ij are driving other gate(s) in the subsequent stages, which are also powered by  $\Phi_2$ , this control method does not increase the number of power supply/clock phases. Instead of directly putting a diode between  $\Phi_1$  and y as in [6], we use an augmented buffer to recycle energy stored in the load capacitance. However, a portion of energy allocated to the gate capacitance of controlling transistor (C1) is dissipated in the diode. One can observe that a smaller amount of energy,  $C_{g}V_{t}V_{dd}$ , is sacrificed to save a larger amount of energy, where  $C_{g}$  is the gate capacitance of the control transistor.

Extension to complex gates is straight forward, and is shown in Figure 3(b). The logic function F is constructed by a T-gate tree. For example, for a NOR gate,  $F = \overline{x1 + x2}$  is constructed by two transmission gates in series and  $F^d$  is the dual form of function F which is constructed by two transmission gates in parallel.

Figure 4 shows another design. We use transmission gates, instead of p-mos trees or nmos trees, to construct logic functions to avoid order of  $CV_t^2$  of energy being dissipated due to the threshold voltage (see section 5). Let us examine how the recovery paths are set up. The transmission gates in the path are turned on only when  $\Phi$  line is restoring its voltage to 0. Without loss of generality, assume that y is charging high and  $\overline{y}$  remains low. The controlling signal C1 is from *ij* and is equal to 0. It turns on the transistor and discharges node y in the restoration phase. The other controlling signal, C2, is from y and hence is



(c) Timing diagram

Figure 3: A design with adiabatic switchings

changing. Since y follows  $\Phi$  down to 0 volts, the voltage level at y and C2 is not less than  $\Phi$ . Thus, the p-mos controlling transistor, C2, will not turn on in the whole discharging phase. The energy dissipation comes from the threshold voltage of the isolation transistor IS1 and control transistor C1, which are in cut-off region until  $V_{ds} \ge V_t$ . The dissipation due to the threshold voltage is modeled and computed in section 5. The construction of a NAND (AND) gate is also shown in Figure 4(b).

The power/clocks for the second design is simpler. Four phases are required instead of six phases of the first design. As an example to show how a circuit can be built using logic primitives, let us consider a serial adder (mod 2) in Figure 5. Three NAND gates form the logic function. By using different signaling, all NAND, NOR, AND and OR gates have the same structure with different connections at the inputs and the outputs. Two additional buffers are used to fit the four-phase power/clock. Note that the transmission gate control signals, P1 and  $\overline{P1}$  only turns on in the restoration phase and therefore can be overlapped with  $\Phi_2$  and  $\Phi_4$ . i.e.,  $P1 = \Phi_2$ ,  $\overline{P2} = \Phi_4$  and so on. In the first design, P and  $\overline{P}$  can not be overlapped with other power/clock phases in a four phase power/clock and hence six phase power/clock are required. Clock phases will be further discussed in the next section.

We can modify both designs to further reduce power dissipation by replacing the isolation transistor by transmission gate. Dissipation due to the threshold voltage of isolation transistor is avoided and the channel resistance is reduced. However, due to additional control signals, six phase power/clock are now required in both designs.

## 4 Designs with reversible logic

Reversible logic is desired since non-dissipative operations can only be realized in this type of logic. As discussed in the previous sections, we need the inverse logic function to control the discharging path of the original gate.

We begin with the naturally reversible logic, the buffer (inverter) chain. Figure 6(a) shows the design of such a buffer (inverter). One can observe that it is constructed with transmission gates, which reduces the channel resistance and has no threshold voltage when it is "on". The functionality of the four transmission gate in each branch is as follows: The input  $x_i$  and  $\overline{x_i}$  control the charging path to set up the logic value of the gate output. The isolation gate controlled by IS; and  $\overline{IS_i}$  has been discussed in the modified version of the previous two designs, which essentially controls the charging path be "on" only during certain time instants. And the gate in the discharging path ( controlled by signal C and  $\overline{C}$ ) function similarily, which turns on during the discharging phase. The discharging path is set up by another transmission gate, controlled by the next gate outputs  $x_{i+2}$  and  $\overline{x_{i+2}}$ , instead of the self-control scheme of the previous two designs. In a buffer chain, this set-up gate can be controlled by the output of the subsequent buffer, which has the same value and timing (holding constant during the restoration of  $x_{i+1}$  and  $\overline{x_{i+1}}$ ). Thus, a buffer chain can



Figure 4: Another design with adiabatic switchings





Figure 5: Serial adder (mod 2) constructed by our logic primitives

be constructed as in Figure 6(c).

Six phases of power/clock is required for this design and is shown in Figure 6. IS; has the phase earlier than that of x; in order to perform the isolation, and  $x_i$  has the phase earlier than  $\Phi_1$ . Hence IS; has two phases earlier than  $\Phi_i$ . Moreover, IS; is still in hold phase when  $\Phi_i$  is in evaluation phase. One can immediately observe that six phases are required in this design. Although it is possible to reduce the number of phases by signal encoding[5] for a buffer chain, the six phase clock can be generally applied in reversible logics. Also note that there is no need to introduce additional control signals for the isolation gate and control gate by observing that IS;  $= \Phi_{i-2}$ ,  $\overline{IS_i} = \Phi_{i+1}$ ,  $C_i = \Phi_{i+2}$ ,  $\overline{C_i} = \Phi_{i-1}$ , etc. The measurent of the power dissipation of the 6 buffer chain with the six phase clock is presented in section 6.

The generalization to other gates is straight forward, however, the signal to set up the discharging path is not available unless reversible logic is used.

There are two directions to design reversible logic circuits. One is to introduce reversible logic primitives, as the Fredkin gate in [2]. The Fredkin gate has nice properties since it is not only logically reversible but also conservative. However, besides the difficulty to implement the gate in electronic circuit, reversible logic primitives may have more output than one may need, i.e., redundant outputs are produced.

Although conventional logic primitives are mainly irreversible, it is easy to modify them to be reversible, e.g., simply buffer the input signals and augment them to the outputs. Hence, one can obtain reversible logic circuit from synthesis of conventional logic gates. But again the reversible logic circuit based on some conventional gate will usually produce redundant output signals, and the energy involved in the redundant output signals is wasted. However, in theory, we can always find a reversible logic design in which the number of garbage signals is at most the number of primary input bits for the whole circuitry [4, 2]. We believe that not only new reversible logic primitives are needed but new logic synthesis style is also required to solve this problem.

## 5 Analysis and measurement of energy dissipation

In our designs, energy consumption is from the channel resistance and the threshold voltage. We first use a linear RC model to analize the energy dissipation due to the channel resistance. Then a nonlinear RC model is used to calculate the dissipation due to threshold voltage.

#### 5.1 Energy dissipation in the transistor channel in a RC model

In this section, we use a simple RC model to compute the energy dissipation in a transistor channel while working in the linear region. When the voltage at the power/clock terminal swings from 0 to  $V_{dd}$  to charge node capacitance through a transistor channel, there is a voltage drop (and hence energy dissipation) in the channel due to channel resistance. The



Figure 6: A design example with reversible logic – the buffer(inverter) chain

model is shown in Figure 7(a). Let us consider the amount of energy dissipated when charging  $V_c$  from 0 to  $V_{dd}$  in time T with a linear power supply voltage of Figure 7(b). We have:

$$RC(\frac{dV_c}{dt}) + V_c = \Phi \tag{2}$$

where,

$$\Phi = \begin{cases} 0, & t < O \\ (\frac{V_{dd}}{T})t, & 0 \le t < T \\ V_{dd}, & t \ge T \end{cases}$$

The solution of the above equation is given by:

$$V_{c} = \begin{cases} 0, & t < O \\ \Phi - (\frac{RC}{T}) V_{dd} (1 - e^{-\frac{t}{RC}}), & 0 \le t < T \\ \Phi - (\frac{RC}{T}) V_{dd} (1 - e^{-\frac{T}{RC}}) e^{-\frac{(t-T)}{RC}}, & t \ge T \end{cases}$$
(3)

The energy dissipation in the above charging process can be calculated as follows:

$$E_{linear} = \int_0^\infty i V_R dt = \int_0^T i V_R dt + \int_T^\infty i V_R dt \tag{4}$$

The first term of equation(4) can be written as

$$\int_{0}^{T} iV_{R}dt = \int_{0}^{T} \frac{(\Phi - V_{c})^{2}}{R} dt$$
  
= 
$$\int_{0}^{T} [\frac{V_{dd}}{T} RC(1 - e^{(-\frac{t}{RC})})]^{2}/R dt$$
  
= 
$$\frac{R^{2}C^{2}}{T} CV_{dd}^{2} \int_{0}^{\frac{T}{RC}} (1 - e^{-\frac{t}{RC}})^{2} d(\frac{t}{RC})$$
  
= 
$$(\frac{RC}{T}) CV_{dd}^{2} [1 - \frac{3}{2}(\frac{RC}{T}) + 2(\frac{RC}{T})e^{(-\frac{T}{RC})} - \frac{1}{2}(\frac{RC}{T})e^{(-\frac{2T}{RC})}]$$

And the second term can be written as  $(5 - 30)^2$ 

$$\int_{T}^{\infty} iV_{R}dt = \int_{T}^{\infty} \frac{(\Phi - V_{c})^{2}}{R}dt$$
  
$$\equiv \frac{\pi \sigma}{T^{2}} CV_{dd}^{2} (1 - e^{-\frac{T}{RC}})^{2} \int_{T}^{\infty} e^{-2\frac{(t-T)}{RC}}dt$$
  
$$= (\frac{RC}{T})^{2} CV_{dd}^{2} [\frac{1}{2} (1 - e^{-\frac{T}{RC}})^{2}].$$

Finally we have,

$$E_{linear} = \left(\frac{RC}{T}\right) C V_{dd}^2 \left[1 - \frac{RC}{T} + \frac{RC}{T} e^{-\frac{T}{RC}}\right]$$
(5)

Let us consider the two extreme cases. When  $T \gg RC$ ,

$$E_{linear} = \left(\frac{RC}{T}\right) C V_{dd}^2 \tag{6}$$

and when  $T \ll RC,$  as in normal CMOS,

$$E_{linear} = \left(\frac{RC}{T}\right) C V_{dd}^{2} \left[1 - \frac{RC}{T} + \frac{RC}{T} \left(1 - \frac{T}{RC} + \frac{1}{2} \left(\frac{T}{RC}\right)^{2}\right)\right] \\ = \frac{1}{2} C V_{dd}^{2}$$
(7)

The response voltage  $V_c$  over time is shown in Figure 8(a) and the dissipated energy versus RC/T is shown in Figure 8(c).



Figure 7: An RC model



Figure 8: Linear and Non-linear Power dissipation with RC model

#### 5.2 Energy dissipation from non-linear mechanism

The linear RC model assumes that the transistor is always in the linear region while charging or discharging. In our designs of Figure 3 and Figure 4, due to the consideration of simplicity in the power supply/clock phases, we use a single transistor for the "isolation" between the input and output instead of a transmission gate controlled by clock signal with proper phase. Also, a single transistor is used to control the recovery path. Due to the threshold voltage, the isolation transistor in the charging path and control transistor in the discharging path are not always working in the linear region. We modeled a channel with threshold voltage  $V_T$  in Figure 7(c). The voltage  $V_C$  is then given by:

$$V_{c} = \begin{cases} 0, & t < t_{0} \\ \Phi - (\frac{RC}{T}) V_{dd} (1 - e^{-\frac{t-t_{0}}{RC}}) + V_{t} e^{-\frac{t-t_{0}}{RC}}, & 0 \le t_{0} < t < T \\ \Phi - (\frac{RC}{T}) V_{dd} (1 - e^{-\frac{T-t_{0}}{RC}}) e^{-\frac{(t-T)}{RC}} - V_{t} e^{-\frac{t-t_{0}}{RC}}, & t \ge T, \end{cases}$$
(8)

where  $t_0 = (\frac{V_t}{V_{rd}})T$ . Let us consider Figure 3. While charging y, there is a current peak when *IS1* jumps from cut-off region to linear region with a voltage drop  $V_{ds} \approx V_t$ . Since an amount of  $CV_t$  charge is required to build the voltage to  $V_t$  level, the energy dissipated due to this non-linear mechanism is approximated by

$$_{\text{Enon-linear}} = \frac{1}{2}CV_t^2 \tag{9}$$

Derivation similar to eq. 5 gives

$$E_{dissipated} = \frac{1}{2}CV_t^2 + (\frac{RC}{T})CV_{dd}^2 \left[1 - \frac{RC}{\beta T} + \frac{RC}{\beta T}e^{-\frac{\beta T}{RC}}\right] + (\frac{RC}{T})CV_tV_{dd} \left[\frac{RC}{\beta T} - e^{-\frac{\beta T}{RC}} - \frac{RC}{\beta T}e^{-\frac{\beta T}{RC}}\right] = (\frac{RC}{T})CV_{dd}^2 + \frac{1}{2}CV_t^2 + O((\frac{RC}{T})^2) \approx E_{linear} + E_{non-linear}$$
(10)

where  $\beta = 1 - \frac{V_L}{V_{dd}}$ . Figure 8(b) and 8(d) show the exact solutions of this RC model with threshold voltage. Assume  $V_t = 1V$  and  $V_{dd} = 5V$ . Therefore,  $(\frac{1}{2}CV_t^2)/(\frac{1}{2}CV_{dd}^2) = 0.04$ . Discharging consumes the same amount of energy and hence 8% of energy is consumed due to the non-linear mechanism. Since the non-linear dissipation is independent of the transition time, it dominates the power consumption when the operating frequency is low, while linear dissipation is more significant in the higher frequency region.

There is energy dissipation due to other factors, such as the source to bulk capacitance, which causes a voltage drop in the channel if the voltage level at two ends of the transmission gate are not the same when a transmission gate turns on. These factors are technology-dependent, and in our SPICE simulation program, they are small but observable. One of the effect that complicates the analysis – "the energy transfer effect" – is discussed in the following section.

#### 5.3 The voltage-dependence of gate capacitance and energy transfers

The transistor gate capacitance varies during the charging and discharging process and is seldom of importance in CMOS digital circuits. We found that such an effect is important in adiabatic-switching circuits since it leads to energy transfers between the controlling and controlled signals.

For an NMOS transistor, the gate capacitance  $C_g \approx C_{gb}$  in cut-off region, and  $C_g = C_{gd} + C_{gs}$  in linear region, where  $C_{gb}, C_{gs}$  and  $C_{gd}$  are the gate to bulk capacitance, gate to source capacitance, and gate to drain capacitance, respectively. Let us consider Figure 9, which shows a charging operation through an NMOS transistor. Assume the initial voltages at gate input  $V_g$  and the power supply line  $\Phi$  are both zero. Then  $V_g$  gradually swings to  $V_{dd}$ . Accompanying this,  $Q = C_g V_{dd}$  of charge flows into the gate terminal, which is associated with  $\frac{1}{2}CV_{dd}^2$  of energy. Following this,  $\Phi$  swings to  $V_{dd}$ , which results in  $V_{gd}$  and  $V_{gs}$  going down from  $V_{dd}$  to 0, (although  $V_g$  is still at  $V_{dd}$ ), causing a current back to the input from the gate terminal. The energy taken away by the current is approximately  $\approx (C_g V_{dd})V_{dd} = C_g V_{dd}^2$ . Thus, the net effect is that about  $\frac{1}{2}C_g V_{dd}^2$  of energy transfers from 9 to the input.

Figure 10 shows this effect from the SPICE simulation. The voltage and the current at the input of an nmos transistor and the power supply line are shown in Figure 10(a) and 10(b), respectively. When charging through a transmission gate, the pmos just works in an opposite way, transferring a certain amount of energy to the power supply. Thus, the net effect of a transmission gate controlling a path is that energy keeps transferring from pmos input line to nmos input line, which is shown in Figure 10(c).

#### 5.4 Measurement of "Net Energy Flow"

To measure the power savings in the circuits we designed, we compute the "Net Energy Flow" which is defined as follows:

$$E(t) = \int_0^t i(t)\Phi(t)dt \tag{11}$$

E(t) is the net energy flowing into the circuit from the power supply line. The voltage and the current waveforms are obtained from SPICE simulation.

As discussed in section 5.4, energy transfers between the controlling signals and the controlled signals. Therefore, we compute the "Net Energy Flow" by summing up the energy in all the input and power supply lines. As in Figure 11, within a cycle (charging and discharging), energy flows into the circuit and is recovered back from it. The level difference of E(t) in two consecutive cycle reflects the energy loss in a full cycle. The effect of energy-transfers between the controlling and the controlled signals lead to the energy transferring from one phase of power supply to another. However, the amount of energy transferred is



(e) Timing diagram

Figure 9: The effect of charging through an nmos transistor



Figure 10: An example of energy transfers between the controlling and controlled signals

steady in our design, not varying from cycle to cycle, and hence can be efficiently recycled. The impact of this effect will be our future consideration.

## **6** Results

The circuits were simulated using SPICE to obtain the voltage and current waveforms, from which we calculate the "Net Energy Flow<sup>n</sup> function E(t). The transistor model is from  $2\mu m$  CMOS processing technology from MOSIS. All transistors (both NMOS and PMOS) used in our analysis are of the same size with W = 3pm, and L =  $2\mu m$ . Hence the W/L ratio is fairly small, which is not chosen to favor our measurement.

Besides the two versions of designs in Figure 2 and Figure 3, we also measured a revised version, referred to as the "design three", which is identical to the design of Figure 3 except that the isolation transistor is replaced by a transmission gate. Six phases of power/clock is required for this design. For an inverter (buffer), the "Net Energy Flow" function E(t) from different designs is shown in Figure 11. The load in each branch is a 0.04pf, which is fairly large considering the fact that the gate capacitance  $C_g < 0.01pf$  for the transistors used in our design.

The difference of the "Net Energy Flow" between two consecutive cycles is the amount of energy dissipated in the circuit in such a cycle. We define the "Energy dissipation percentage" as the ratio of the amount of dissipated energy and the amount of energy entering the circuit from the power supply in a complete cycle. For different transition times, we compared the "Energy dissipation percentage" from different designs in Figure 12. Results show that this percentage is around 10% at 1MHz, and increases when the frequency goes higher. The revised version has the smallest power dissipation, but needs six phases of power supply/clock, while design two needs only four.

Finding the actual energy savings of our design over static CMOS is non-trivial. The area increases by more than two fold, and hence the capacitance also increases. Also, signal switching activities are important for CMOS circuits, which depends on the applications under consideration. However, in our design, the signals are generated with their complements, and the signals are also self-buffered. And hence, some inverters and registers can be saved. Let us consider a serial adder (mod2) of Figure 5. Two inverters and one register can be saved in our design. Another interesting phenomenon is that three 2-input AND gates can form an 2-input XOR logic.

We computed E(t) for the serial adder (mod2) with three AND gates and two buffers running at 1.67*MHz*. The gates are from logic primitives of Figure 3. The two buffers are used to fit a simple four-phase power supply and the output load was 0.04pf in each branch. Results are shown in Figure 13. Approximately 85% of energy entering the circuit can be saved.

We also measured the reversible design of a six-buffer (inverter) chain, which is shown



Figure 11: Computed energy flows in the circuits



Figure 12: Dissipation computed from SPICE simulation



Figure 13: Power dissipation in a serial adder (mod2)



Figure 14: Power dissipation in six buffer(inverter) chain with reversible design



Figure 15: Power dissipation in different operating frequencies

in Figure 14. Only 5% of energy is dissipated at the operating frequency of 1.1MHz. We used 6 phase power supply/clock without external control signals. The output load at each branch was also 0.04pf.

The power dissipation of the six-buffer chain and the serial adder (mod 2) were also measured in different frequency regions, which is shown in figure 15. 70% of the energy can be saved for the reversible buffer chain when the frequency goes as high as 55.6MHz.

#### 7 Conclusions and future work

In this paper, we presented designs for energy-recovery logic primitives. Results from SPICE simulations are very encouraging. We have also presented a detailed analysis and modeling of adiabatic-switching, which is experimentally validated by the simulation.

From the results, one can observe that the partially reversible designs work well when the operating frequency is not too high(  $\leq 10MHz$ ) or when the capacitance a gate drives is not much larger than the internal gate capacitance. The reversible designs can save significantly more energy in the high frequency region, because of the fact that there is no non-linear dissipation due to the threshold voltage and the effective resistance is lower in the transmission gate constructs. We observe that the circuit topology is significantly different from the static CMOS circuits even when our logic primitives simulate the conventional gates. For future research, we will study the circuit topology with adiabatic switching logic primitives and the possibility of completely reversible logic designs.

#### References

- [1] C.H.Bennett, "The thermodynamics of computation- a review," Int. *Journal* of Theoretical Physics, vol. 21, no. 12, pp. 905-940,1982.
- [2] E. Fredkin and T. Toffoli, "Conservative Logic," Int. Journal of Theoretical Physics, vol. 21, no. 3/4, pp. 219–253,1982.
- [3] R.Landauer, "Dissipation and noise immunity in computation and communication," Nature, vol. 335, pp. 779-784, Oct. 27 1988.
- [4] R.P.Feynman, "Quantum mechanical computers," Foundations of Physics, vol. 16, No. 6, 1986.
- [5] W.C.Athas, L."J".Svensson, J.G.Koller, N.Tzartzanis, and E.Chou, "A Framework for Practical Low-Power Digital CMOS Systems Using Adiabatic-Switching Principles," Intl. Workshop on Low Power Design, 1994, pp. 189-194.
- [6] J.S.Denker, S.C.Avery, A.G.Dickinson, A.Kramer, and T.R.Wik, "Adiabatic Computing with the 2N-2N2D logic Family," Intl. Workshop on Low Power Design, 1994, pp. 183-187.
- [7] S.G.Younis and T.F.Knight, "Asymptotically Zero Energy Split-Level Charge Recovery Logic," Intl. Workshop on Low Power Design, 1994, pp. 177-182.
- [8] A.P. Chandrakashan, S. Sheng, and R. Brodersen, "Low Power CMOS Digital Design," IEEE Journal of Solid-State Circuits., vol. 27, No. 4, Apr. 1992, pp. 473-483.
- [9] K. Roy and S. Prasad, "Circuit Activity Based Logic Synthesis for Low Power Reliable Operations," IEEE Trans. on VLSI Systems, Dec. 1993, pp. 503-513.
- [10] C. Tsui, M. Pedram, and A. Despain, "Technology Decomposition and Mapping Targeting Low Power Dissipation," ACM/IEEE Design Automation Conf., 1993, pp. 68-73.
- [11] J. Monteiro, S. Devadas, and A. Ghosh, "Retiming Sequential Circuits for Low Power," IEEE Intl. Conf. on Computer-Aided-Design, 1993, pp. 398-402.
- [12] V. Tiwari, P. Ashar, and S. Malik, "Technology Mapping for Low Power," ACM/IEEE Design Automation Conf., 1993, pp. 74-79.

[13] P. Landman and J. Rabaey, "Power Estimation for High Level Synthesis," European Design Automation Conf., 1993, pp. 361-366.

\_\_\_\_\_