Aaron J Arthurs
Low Power Design
November 1, 2003
Class Project

A Theoretical Comparison Between Static-CMOS and Hybrid-RERL

Abstract:

This paper presents a theoretical comparison between a static-CMOS and a hybrid-RERL design of an inverter. The measurements taken into account are the total power dissipation, the performance in terms of propagation delay, and the size of each design. The hybrid-RERL architecture uses the RERL format except it reduces the 6-phase clock presented in earlier papers to a 5-phase clock by replacing the isolation switch on the output of the logic function with a non-gated, isolating device (i.e. a diode). Still, the hybrid-RERL logic makes further use of reversible logic. Although, the idea behind reversible logic (and adiabatic circuits) is to eliminate all non-adiabatic losses, this paper places a secondary goal of simplifying the hybrid-RERL inverter, which results in some non-adiabatic loss. Overall, the hybrid-RERL inverter significantly reduces power consumption at the cost of complexity and performance.

Introduction:

As personal computers become more compact and portable (e.g. laptops, PDA's, cell phones, etc.), power consumption becomes a larger factor. Although the energy dissipation from digital devices has declined over the last 50 years, it is expected that this tread line will slow down and/or stop within the next decade or two [3]. The reason, as mentioned by a number of papers on low power design, is that conventional designs do not save energy, which prevents current digital designs from dissipating energy as low as kT Joules (k is Boltzman's constant and T is the temperature of operation, usually 300 Kelvins). Furthermore, as digital devices pack more logic per volume and operate at higher frequencies, more power is consumed.

Two solutions come to mind such as operating computers at lower temperatures and designing circuits that conserve energy. The problem with the former solution is that the overall power may not be reduced because of the power consumed by cooling units (e.g. a cooling fan and a refrigerator). Even with non-power consuming cooling units such as a heat sink, the amount of area taken (especially for a portable device) justifies the need for an alternative solution. The next solution is to change the way traditional digital circuits are designed, reducing the amount of energy unnecessarily dissipated as heat. To explain what is meant by "energy unnecessarily dissipated" gives attention to two types of energy dissipation: adiabatic and non-adiabatic [1, 2].

Adiabatic energy loss occurs due to an inherit characteristic in the circuit such as the turn-on resistance of a MOSFET transistor. If a digital circuit is clocked, the adiabatic loss can be adjusted. According to Joonho Lim et. al., lowering the transition time of a clock signal will reduce adiabatic loss [1]. Non-adiabatic loss, on the other hand, occurs whenever a non-zero potential exists across a transistor (from source to drain) while the transistor is switching on; furthermore, non-adiabatic loss is the "energy unnecessarily dissipated" that deserves change in traditional circuit designs. To reduce the potential difference between the source and the drain on a transistor, reversible logic is used to recover the energy from the output to the input.

Several approaches that integrate reversible logic have been taken; however, those approaches have their shortcomings (explained later). One popular design approach is known as RERL (Reversible Energy Recovery Logic), which uses a 6-phase clock to control the energy path from input to output (energy charging), then from output to input (energy recovery). In this paper, two different designs are used for an inverter: a conventional CMOS inverter and a hybrid-RERL inverter. These two designs are compared with respect to energy dissipation, performance, and size.

Reversible Logic – General Idea:

Reversible logic is simply any logic such that each unique value of the outputs maps to a unique value of the inputs. In mathematical notation, if Output = F(Input) and there exists the inverse function F^{-1} such that Input = F^{-1}(Output), then the function F is reversible. For example, an inverter is reversible because if the value of the output is known, then the value of the input can be determined. On the other hand, a 2-input, 1-output XOR gate is irreversible because the value of the inputs cannot be computed given the value of the output.

However, it is possible to make irreversible logic gates reversible. In conventional designs, irreversible logical devices typically have a few outputs and many inputs (e.g. N-input AND gate with a single output), which, according to Michael Miller et. al., makes a logical device irreversible [4]. In the case of the 2-input, 1-output XOR gate, adding an additional "garbage" output that directly maps one of the inputs will make a reversible 2-input, 2-output XOR gate, illustrated in Figure 1.

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

(a)

A	B	Z	A'
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	1

(b)

Z	A'	A	B
0	0	0	0
0	1	1	1
1	0	0	1
1	1	1	0

(c)

Figure 1: (a) shows an irreversible truth table for XOR. (b) is the same except a "garbage" output is added, thereby making the XOR function reversible. (c) shows how the output maps back to the inputs using the inverse functions from A' and Z (A = A' and B = A' XOR Z).

In later sections, it will be shown how reversible logic can be integrated into a design to reduce the power consumption.

Background of Reversible Logic in Digital Circuits:

Every digital design presented below consumes less power than a conventional design. The trade-off for reduced power consumption, however, is speed and complexity (or real estate).

One implementation of a digital circuit using reversible logic is known as SCRL (Split-Charge Recovery Logic), contributed by Saed Younis and Thomas Knight, Jr. [6]. The SCRL circuit contains two logical paths. The first logical path computes the logical function and tri-states the output of the function block with a transmission gate. The other logical path computes the input using the output from the first logical path; it too contains a tri-stated output back to the inputs. Initially, every node (except the zero-voltage, enable signals to the transmission gates) holds a voltage of $V_{dd}/2$. The operation of the circuit that follows is similar to the operation of a RERL circuit, which is explained in the next section. An advantage to the SCRL circuit is that it employs existing technology (e.g. a CMOS inverter) and adds additional components (e.g. a transmission gate with clocks) to reduce some of the non-adiabatic loss. The drawback is that it requires 24 unique clock signals, an increase in complexity.

Another implementation is a variation of RERL called nRERL, proposed by Lim et. al. [1]. The entire circuit is implemented with NMOS transistors, which have the source and drain terminals coupled with capacitors. The nRERL circuit is competitive as it reduces the complexity of a reversible logic circuit; however, clamp devices are used to drive un-driven nodes to a constant ground (i.e. every input/output node has an inverted version), which produce non-adiabatic loss.

Finally, Yiben Ye et. al. explained the realization of QSERL (Quasi-Static Energy Recovery Logic) [5]. A QSERL circuit uses two, complementary sinusoidal clock signals to control the logic. Unlike the SCRL and nRERL, QSERL does not have a path from the output back to the input; it evaluates and holds the output in each stage. While QSERL has the advantage of a CMOS circuit (no voltage degradation in the output) and yet reduces the amount of switching activity, it still dissipates non-adiabatic energy due to the switching while non-zero potentials across the PMOS and NMOS trees are present.

Before analyzing the inverter using CMOS and hybrid-RERL, the next section describes a common RERL design.

General Architecture of RERL:

RERL is based on each logic device having forward and backward logic paths that are controlled by a 6-phase clock (for a timing diagram of the 6-phase clock, the reader is referred to [1] on page 870). Figure 2 shows the architecture of a RERL circuit. In the architecture, F is the logic function, F^{-1} is the inverse function of F, I_1 and I_2 are both isolation switches, and the Φ inputs are clock signals. The F block performs a complex function of the inputs, and Φ_1 is used as the power source (or ground) for each transistor in the block. The I_1 block tri-states the output from the F block using Φ_0 as the gate controller. The F^{-1} and I_2 blocks use a similar data path.

The RERL architecture operates in the following manner. First of all, the assumption made is that all nodes are initially discharged and the input lines contain a valid value (this assumption turns out to be a characteristic of a RERL circuit when it is part of a chain of RERL circuits). So, the circuit computes the output based on the input. Once the output is determined, the input is no longer needed so the circuit discharges its input through the Φ_2 signal. The following explains the operation in more detail.

During the first step in the RERL circuit's operation, Φ_0 is the first clock signal to rise, connecting the output of the F block to the output lines; concurrently, the input lines turn on certain transistors inside the F block. Because the lines feeding into the I_1 block and the lines feeding out of the same block both have low voltage when the transistors in the block are switching on, there is no non-adiabatic loss occurring from those transistors. The same happens for the transistors inside the F block because Φ_1 and the F block's output both have low voltage.

In the second step and after Φ_0 is at a high voltage, Φ_1 begins to rise, charging the output lines with the value F(Input); also, certain transistors in the F^{-1} block are turned on. The clock signal Φ_2 is at a low voltage as well as the lines feeding out of the F^{-1} block, so no non-adiabatic loss occurs from those transistors.

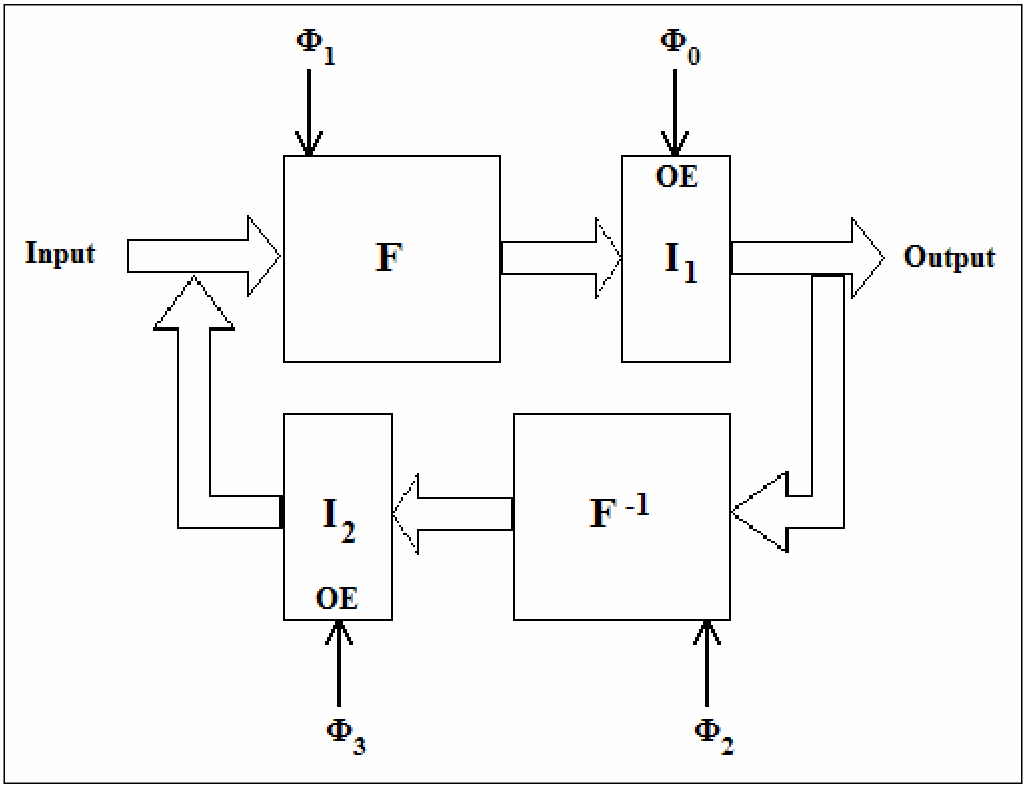

Figure 2: General architecture of a RERL circuit

During the third step and after Φ_1 is at a high voltage, Φ_0 begins to fall, cutting the output lines from the F block. At the same time, Φ_2 begins to rise and charges the lines feeding from the F^{-1} block with the value F^{-1}(Output), which equals the value stored on the input lines. No non-adiabatic loss occurs because no transistors are being turned on.

In the next step, Φ_0 is at low voltage, Φ_1 begins to fall and discharges the output from the F block, Φ_2 is at high voltage, and Φ_3 begins to rise, connecting the output from the F^{-1} block to the input lines. Once again, no non-adiabatic loss results because the lines feeding into the I_2 block have the same potential as the input lines.

During the fifth step, Φ_3 remains at high voltage and Φ_2 begins to fall and discharges the output from the F^{-1} block as well as the input lines. Then, Φ_3 falls and disconnects the input lines from the low voltage Φ_2; both the input lines and the output from the F^{-1} block are at low voltage, preventing non-adiabatic loss. Finally, the operation of the circuit starts over.

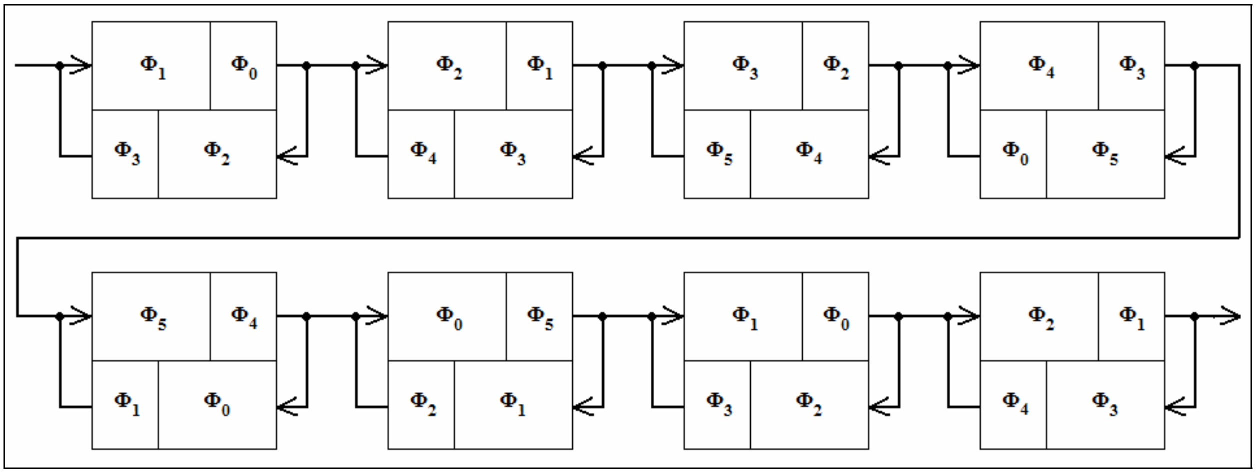

Figure 3: Each block represents a RERL circuit, and each sub-block represents one of the four major components in the RERL circuit (upper-right is I_1, upper-left is F, lower-right is F^{-1}, lower-left is I_2). The clock names, Φ, in each sub-block represents which clock signal in the 6-phase clock is applied to that sub-block.

Throughout the entire operation of the circuit, there is never non-adiabatic loss from any transistor because there is never a non-zero potential across any transistor while that transistor switches on. Also, at the end of the operation, every node except the output lines is discharged to low voltage. The output lines of this circuit will be discharged by a circuit that follows this circuit (i.e. the output lines become the input lines of the next RERL circuit). However, the question is how to clock a chain of RERL circuits; Figure 3 shows such a chain. As it turns out, a RERL circuit that follows another RERL circuit performs the same operations one clock phase later. For example, after a RERL circuit charges its output with a valid value, the next RERL circuit starts energizing its output.

Before showing a an inverter can be implemented using the RERL architecture, a static-CMOS inverter is presented to demonstrate the non-adiabatic loss that occurs when its input switches.

Analysis of the Static-CMOS Inverter:

A static-CMOS inverter, as in any static-CMOS design, consists of a PMOS and an NMOS network; Figure 4 shows the design of the inverter. Assuming the output is zero (input is V_{dd}), the voltage across the NMOS transistor is zero volts while the voltage across the PMOS transistor is $(V_{dd} – 0) = V_{dd}$. As the input switches to zero volts, the NMOS transistor disconnects and the PMOS transistor conducts. However, due to the initial voltage, V_{dd}, across the PMOS transistor, the non-adiabatic power consumed by the transistor is related by ($Power_{pmos} = V_{dd}^2 / R_{pmos}$), which is non-zero. The same happens for the NMOS transistor ($Power_{nmos} = V_{dd}^2 / R_{nmos}$) when the input switches back to V_{dd}. So for a static-CMOS inverter, the average non-adiabatic power consumption per switching is ($Power_{avg} = (Power_{pmos} + Power_{nmos}) / 2 = V_{dd}^2 * ((R_{nmos} + R_{pmos}) / (R_{nmos} * R_{pmos}))$)); for more complex static-CMOS circuits, the $Power_{avg}$ formula can be applied by computing the total resistance of the PMOS network and the NMOS network (not taking into account the average switching within each network as well as the signal probabilities of the inputs). For simplicity, the power consumption of a circuit will be computed by (Power Cost = Σ {each transistor i}($V_{SW(SD),i}$)) where $V_{SW(SD),i}$ is the voltage from source to drain across transistor i when the transistor is switched on (i.e. $V_{SW(SD),i}$ is a possible non-adiabatic loss from a transistor i). In the case of this inverter, Power Cost = 2 * V_{dd}.

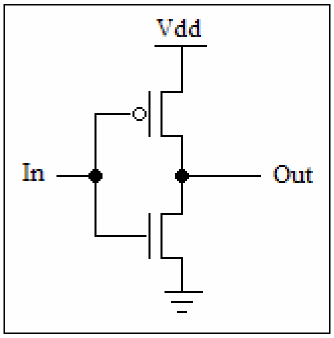

Figure 4: Static-CMOS Inverter

The performance of a static-CMOS inverter can be related the rising time, $T_{pmos} \alpha V_{SW(SD),pmos}$, or the fall time, $T_{nmos} \alpha V_{SW(SD),nmos}$. So, the performance cost of the inverter per switching of the input is half the power cost (Performance Cost = V_{dd}).

Finally, the size of the inverter is two transistors and four literals (In, Out, V_{dd}, and Ground).

Analysis of the Hybrid-RERL Inverter:

Before explaining the design of a hybrid-RERL inverter, there is an issue that must be resolved: voltage degradation, illustrated in Figure 5. What would cause the voltage of a node to degrade is when that node is connected to the drain terminal on a transistor, depending upon the transistor type and the node connected to the source terminal. A PMOS transistor, for instance, conducts when there is a voltage of V_{dd} from the source terminal (V_{dd}) to the gate (zero volts); the drain terminal is driven to V_{dd}. However, as the source terminal on a PMOS transistor is driven to zero volts with the gate remaining at zero volts, the transistor stops conducting when the source terminal reaches $|V_{th,p}|$; therefore, once the source terminal reaches zero volts, the drain terminal remains at $|V_{th,p}|$. So in short, when the source terminal is driven to a strong low (zero volts) and the gate is driven to a strong low, the output (drain terminal) is driven to a weak low ($|V_{th,p}|$). In the case of the NMOS transistor, the opposite occurs; when both the source terminal and the gate are driven to a strong high (V_{dd}), the drain terminal is driven to a weak high ($V_{dd} - V_{th,n}$). These weak low and high voltages stored in an output are undesirable and must be resolved.

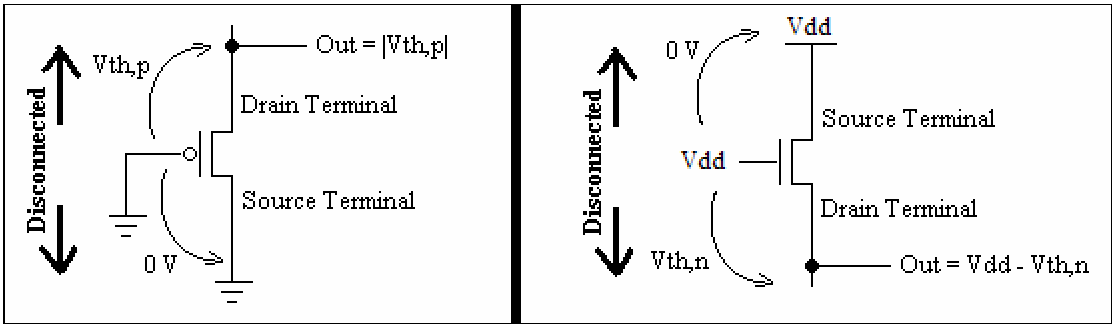

Figure 5: For each transistor shown, voltage degradation occurs due to the zero potential across the source terminal and the gate, causing the initially un-driven output (drain terminal) to store a weak signal.

There are two general ways to resolve a weak signal. One way is to prevent such signals by applying a constant voltage to the source terminal depending on the transistor. The CMOS inverter presented in the previous section eliminates degraded signals by connecting a constant V_{dd} to the source terminal of the PMOS transistor and a constant zero volts to the source terminal of the NMOS transistor; the output connects to the drain terminals of both transistors. The other way is to add an isolating path from the drain terminal to the source terminal that will assure equal voltage across the transistor while it is conducting; for example, Figure 6 shows how a diode can be used to eliminate voltage degradation. The drawback to this method is that the drain terminal is driven by the source terminal despite the input into the gate (e.g. the drain terminal on the PMOS transistor in Figure 6 is always driven to ground). Fortunately, for a RERL circuit, this drawback imposes no problem because the voltage to the source terminal on any transistor varies over time.

As explained later, the motivation for designing a RERL circuit with diodes in order to eliminate voltage degradation is to simplify the circuit. In fact, the 6-phase clock presented by earlier papers can be reduced down to a 5-phase clock. Also, transmission gates can be used instead, but at the cost of complexity because transmission gates require an additional control signal. Plus, there are certain aspects in the RERL design that makes a gate-controlled device unnecessary because each node in the circuit is sampled for a short period of time before being discharged to ground. The need for zero non-adiabatic loss in a circuit requires additional complexity and control. As it turns out, the hybrid-RERL inverter (presented later) does not impose this requirement and instead allows some non-adiabatic loss to simplify the circuit.

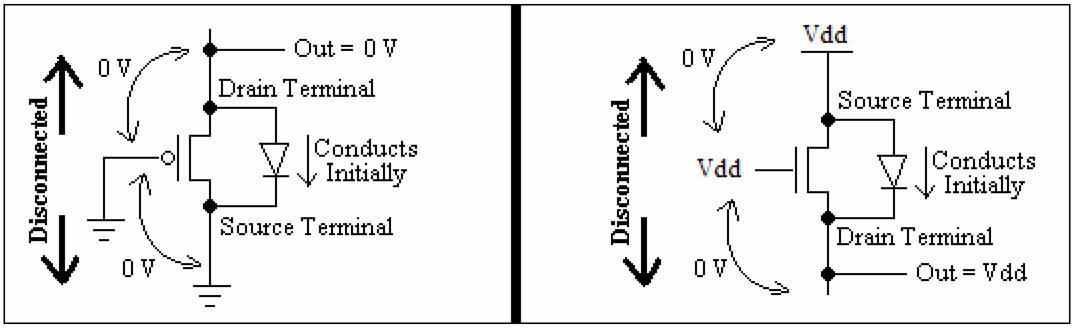

Figure 6: For each transistor shown, a diode is added such that it conducts towards the direction of a possible voltage drop (from ‘+’ to ‘-‘) to eliminate voltage degradation. Because the voltage on the drain terminal is driven to the same voltage to the source terminal as well as to the gate, the transistor is turned off or disconnected.

Following the architecture from Figure 2, an inverter can be implemented as shown in Figure 7. Unlike a common RERL circuit, the output of the logic function is controlled by a diode instead of a clock-gated isolation switch; therefore, a 5-phase clock signal is used for the circuit. The operation of the inverter begins with all nodes discharged to zero volts, except the input that contains a valid value. With the input being zero volts, Φ_0 is the first clock signal to rise and conducts through the upper PMOS transistor, charging both node n1 and the output line. So far, the output line contains the inverted value of the input and no non-adiabatic loss occurs. Now, as Φ_0 stays high, Φ_1 begins to rise, which has no effect on node n2 because the diode parallel to the lower PMOS transistor is reverse-biased and the transistor is disconnected due to the high voltage contained in the gate terminal. Next, Φ_0 falls and discharges node n1 through the diode parallel to the upper PMOS transistor (the diode between node n1 and the output line is now reverse-biased), Φ_1 stays high as well as the output line, and Φ_2 begins to rise and connects node n2 to the input line. In the next phase, Φ_1 falls and Φ_2 stays high, having no effect on the circuit. Finally, Φ_2 falls to zero volts and disconnects node n2 from the input line. As a result, when the input is zero, the output is driven to V_{dd} and every other node is at zero volts; no non-adiabatic loss occurs throughout the process.

A problem occurs when the input is V_{dd}; Figure 8 shows this problem. First, Φ_0 rises and has no effect on node n1 and the output line (both remain at zero volts). Then, Φ_0 stays high and Φ_1 rises, charging node n2 to V_{dd}. Now, Φ_0 falls, Φ_1 stays high, and Φ_2 rises. At this point, the NMOS transistor is not conducting because every terminal is at V_{dd} (i.e. there is zero voltage from the gate to the source terminal). Next, Φ_2 stays high and Φ_1 starts falling. Because the NMOS transistor is initially turned off, only node n2 is discharged through the diode parallel to the lower PMOS transistor; the input line remains at V_{dd}. As soon as Φ_1 reaches the voltage ($V_{dd} - V_{th,n}$), the NMOS transistor switches on; however, a voltage of $V_{th,n}$ exists across the transistor during switching, resulting in non-adiabatic loss.

A minor change that can be made to the inverter is to replace the NMOS transistor with a transmission gate. The advantage to this is that while Φ_1 discharges, the transmission gate is turned on the entire time, meaning the input will follow Φ_1 without a voltage build-up between the input and node n2. The disadvantage is that an inverted version of Φ_2 along with Φ_2 itself is required to control the transmission. Because the clock signals have a 33.3% duty cycle (each signal is at high voltage one-third of the time) instead of an even 50% duty cycle, inverters are required to create the inverted version of each clock signal. The addition of the inverters for the clock signals may or may not be self-defeating depending on how much logic is controlled by those signals.

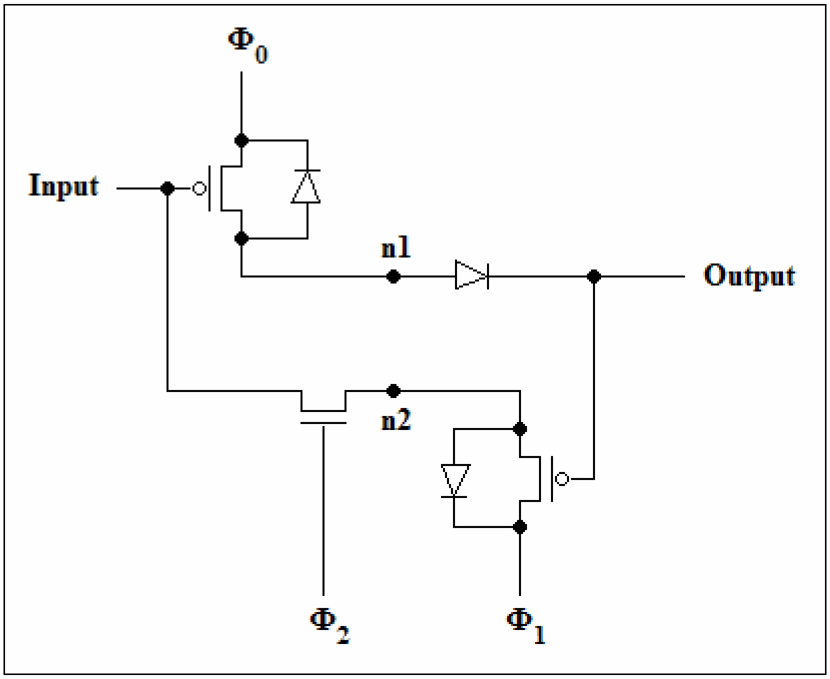

Figure 7: Hybrid-RERL Inverter

Overall, the total non-adiabatic loss from the circuit shown in Figure 7 is (Power Cost = $V_{th,n}$), which is significantly lower than the non-adiabatic loss from the static-CMOS inverter. If V_{dd} = 5V and $V_{th,n}$ = 0.7V, then the non-adiabatic loss from the static-CMOS inverter is approximately 14 times greater than the non-adiabatic loss from the hybrid-RERL inverter. If a transmission gate was used in place of the NMOS transistor in Figure 7, the non-adiabatic loss from the clock generator (as it generates both the signals and their inverted versions) would add to the overall loss of the hybrid-RERL circuit. The motivation to use a transmission gate instead is if a large number of hybrid-RERL circuits are present. Then, because the clock signals are being shared, the non-adiabatic loss from the clock generator becomes insignificant.

The performance of the hybrid-RERL inverter in Figure 7 is related by the transition time, $T_{\Phi 0}$, of the clock signal Φ_0. This is because once the input is charged to a valid value, the time it takes for the output to contain a valid value is the time it takes the clock signal, Φ_0, to rise from 0V to V_{dd}. Then, the output is sampled by the next circuit while the inverter uses the reverse logic path to discharge its input. In theory, the performance of a hybrid-RERL circuit is worse than a static-CMOS inverter because the transition time of the clock signal Φ_0 may be large to reduce adiabatic losses, unlike the static-CMOS inverter which abruptly charges its output, which usually takes a very short amount of time.

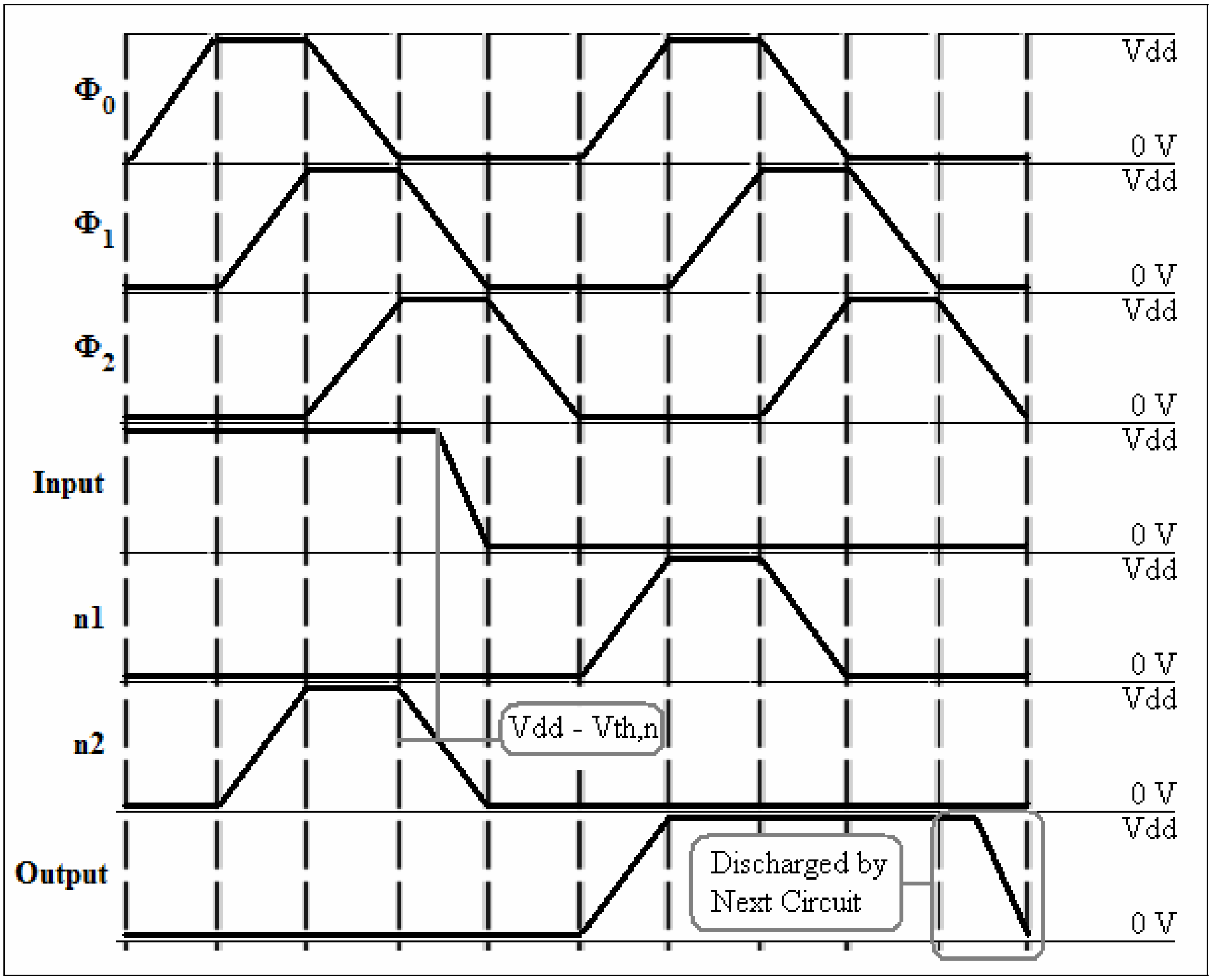

Figure 8: A timing diagram of the hybrid-RERL circuit from Figure 7 is shown. The circled text in the picture illustrates the non-adiabatic loss that occurs whenever the input to the hybrid-RERL circuit is discharged.

Finally, the size of the hybrid-RERL circuit is three transistors, three diodes, and five literals, which is significantly larger than the static-CMOS inverter due to the extra literal. Also, the size of the clock generator circuit adds to complexity of the hybrid-RERL inverter.

Conclusion:

A theoretical comparison between a static-CMOS inverter and a hybrid-RERL inverter is made. The result is that while the static-CMOS inverter performs better and takes up less real estate than the hybrid-RERL inverter, the hybrid-RERL inverter consumes about 5-10% of the power than the static-CMOS inverter. Also, Lim et. al. proven that designing their circuit using the RERL architecture reduced the power consumption by 95% [1].

With a theoretical study of the hybrid-RERL circuit, further analysis is demanded. That is, a measured and simulated comparison between conventional circuit and a circuit using the hybrid-RERL architecture is yet to be performed. Until then, the motivation for new designs using reversible logic will increase.

References:

[1] Lim, Joonho, Kim Dong-Gyu, and Soo-Ik Chae. “nMOS Reversible Energy Recovery Logic for Ultra-Low-Energy Applications.” IEEE Journal of Solid-State Circuits 35.6 (2000): 865-75.

[2] Lim, Joonho, Kim Dong-Gyu, and Soo-Ik Chae. “Reversible Energy Recovery Logic Circuits and Its 8-Phase Clocked Power Generator for Ultra-Low-Power Applications.” IECE Trans. Electronics E82-C.4 (1999): 646-53.

[3] Merkle, Ralph C. “Reversible Electronic Logic Using Switches.” Nanotechnology 4 (1993): 21-40.

[4] Miller, Michael, Dmitriv Maslov, and Gerhard W. Dueck. “A Transformation Based Algorithm for Reversible Logic Synthesis.” Annual ACM IEEE Design Automation Conference 20.1 (2003): 318-23.

[5] Ye, Yibin, Kaushik Roy, and Georgios I. Stamoulis. “Quasi-Static Energy Recovery Logic and Supply-Clock Generation Circuits.” International Symposium on Low Power Electronics and Design 20 Aug. 1997: 96-9.

[6] Younis, Saed G., and Thomas F. Knight, Jr. "Asymptotically Zero Energy Split-Level Charge Recovery Logic." International Workshop on Low Power Design Mar. 1994: 177-82.