

Delay Efficient Kogge Stone Approach for Implementing Shift Registers By Using Pulsed Latches

Somarajupalli sudharani¹, M.Sumalatha²

¹PG Scholar, CEC, ECE Department, AP, India

²Assistant Professor, CEC, ECE Department, AP, India

Abstract:

This project proposes delay efficient architecture for shift registers by using pulsed latches instead of flip flops. By using latches instead of flip-flops the major factors area and power can be reduced. By considering the necessary delays in pulses for latches the timing problem latches can be reduced. For obtaining these delays counter has to incremented by 1. The proposed kogge stone adder architecture reduces the delay to maximum extent, and produces numerous variations between conventional adder architecture. The synthesis and simulation is carried out using XILINX ISE 12.3i and HDL is developed using VERILOG language.

Keywords: flip-flops, latches, kogge stone adder, VERILOG.

1. Introduction:

Across all elements of chip dynamic power is consumed. Clock network is one of the large consumers of dynamic power. Dynamic power significantly reduced by reducing the power in the clock network. By the usage of small clock buffers number of techniques are used to reduce the clock power. Clock network consumes large dynamic power as registers are used as state elements in the beginning. A register in general is known to be flip-flop. Based on STA for SoCs timing optimization is must. A methodology has been developed which uses latches triggered with pulse clock waveforms. Combinational logic circuit is the one whose output depends only on the present state of input. These are also known as time independent logic circuits i.e., they don't have the capability to store a state inside them. Memory elements are not necessary in combinational circuits. The data which is stored in the computer is used for performing the arithmetic operations by the combinational logic circuit .by using different devices such as multiplexers, de-multiplexers, encoders, decoders, halfadder, fulladder combinational logic circuits are logically implemented. The unit of computers that are

generally comprises of arithmetic and logic unit are nothing but combinational logic circuits.

It works independent of clock. As there are no clocks they don't need triggering in these digital logic circuits. By using a set of output functions combinational logic circuits are defined. Sum of products or products of sum methods will define the design of combinational logic circuit.

2. Combinational circuits

Half-adder is a circuit that adds two single bits. It has two inputs A and B and the outputs for the half adder circuit are sum (S) and carry (C0). The value of sum is calculated by xor operation and the carry value is calculated by and operation on two inputs. The truth table for half adder is shown below.

INPUTS		OUTPUTS	
A	B	SUM	CARRY
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Table.1: Truth of half adder.

The value of sum and carry are calculated by using the following equation

$$\text{Sum} = A \text{ xor } B$$

$$\text{Carry} = A \text{ and } B$$

The logic circuit is shown below

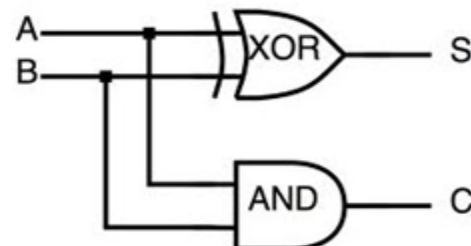


Figure.1: schematic of half adder.

Full Adder:

Full adder is a digital circuit that adds three single bits. The sum result is represented by S and the carry value is given as Cout or C0. The sum value is calculated by using xor operation on three inputs.

Sum = A xor B xor Cin

Carry = (A and B) Or (B and Cin) Or (Ci and A)

Input			Output	
A	B	Cin	Sum	Carry
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Table.2: Truth table of full adder

The logic diagram of a full adder with inputs A, B, Cin and the outputs are S and C0 is shown below

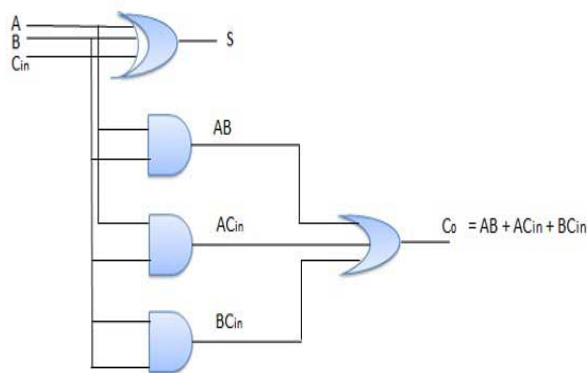


Figure.2: schematic of full adder.

Kogge-stone adder:

It is one of the parallel-prefix adders. It has low fan-out at each stage that increases the performance of a typical process. It propagates and generates the carry bit at each stage in vertical position. The carriers are produced in the last stage (vertically), and these bits undergo XOR operation with the initial propagation after the input to produce the sum bits. The radix of the adder is founded by the number of computations that are performed. The number of required stages will reduce but the power consumption will increase. With the increase in sparsity the total need of computation and the amount of routing congestion can be reduced. The performance of speed can be improved by using these adders.

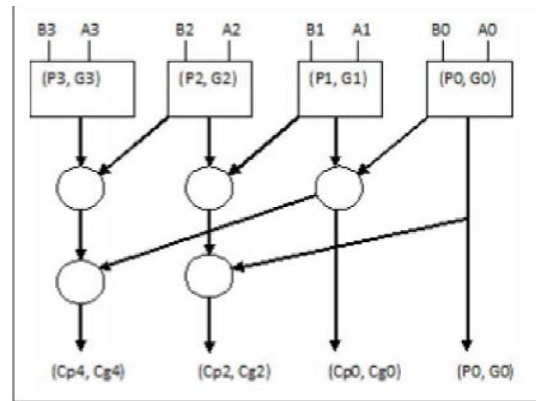


Figure.3: 4-bit kogge stone adder.

Sequential logic circuits are the one whose output depends on the present output as well as on the past output. These are capable to retain the state of a system that is based on the current input as well as on the earlier state. These devices are capable of storing the data which is not possible in the combinational logic circuits. In general when the circuit consists of more than one input with an input that is fed back to circuit is known as sequential circuit. Sequential circuits are used in maximum finite state machines and in memory elements, which are digital circuit models with finite possible states. For triggering the flops that use clock maximum devices would be in sequential model.

Synchronous circuits are the one in which all the circuits are not triggered at a time. The set of output functions and set of next state or memory functions are used to define the behavior of sequential logic circuits. We use both the circuits in practical.

3. Sequential circuits.

SR-Latch:

An SR-latch is a circuit that stores information. The signal can be either set or reset,. The circuit is said to be in “Reset” state when both the inputs are equal to 01 respectively. The circuit is said to be in “Set” state when the inputs are equal to 10. When both the inputs are equal to one then the circuit is said to be in “invalid state”. The logic diagram and truth table are shown below

EN	S	R	Q _n	Q _{n+1}	State
1	0	0	0	0	No change (NC)
1	0	0	1	1	
1	0	1	0	0	Reset
1	0	1	1	0	
1	1	0	0	1	Set
1	1	0	1	1	
1	1	1	0	X	Indeterminate
1	1	1	1	X	
0	X	X	0	0	No change (NC)
0	X	X	1	1	

Table.3: Truth table SR-latch.

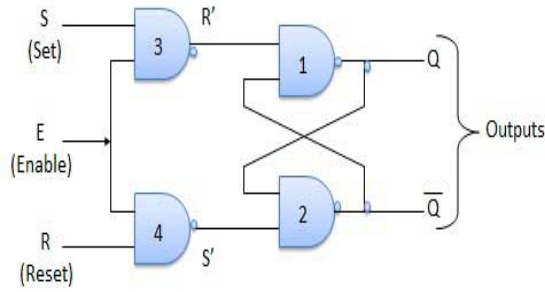


Figure.4: SR-latch

D-Latch:

The undesirable condition which occurred in the SR latch can be eliminated in D-latch. D-latch is the logic circuit with two inputs as en (enable) and D (data). Whenever the two inputs are given 1 the output goes to high. It is well known as a transparent latch as both the inputs are equal to one the data input will be present at output. In this state the circuit is said to be in “set” state. “Reset” state occurs by making the D (data) input as zero.

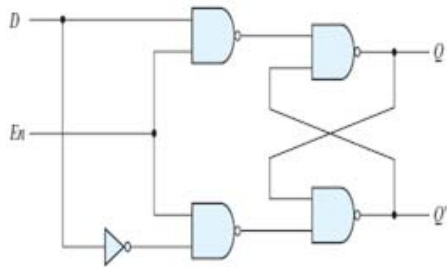


Figure.5: D-latch.

En	D	Next state of Q
0	X	No change
1	0	Q = 0; reset state
1	1	Q = 1; set state

Table.4: truth table

JK-Latch:

The operation of JK-Latch is same as SR-latch the only change is done with the invalid state that occurs in SR-latch. In case of JK-latch when both the inputs are equal to one the output will be in toggle condition i.e., 1 to 0 and 0 to 1. Latch will toggle continuously throughout the level. It has four states no change, Reset, Set and toggle.

	Input		Output		Description
	J	K	Q	Q̄	
same as for the SR Latch	0	0	0	0	Memory no change
	0	0	0	1	
	0	1	1	0	Reset Q >> 0
	0	1	0	1	
	toggle action	1	0	0	1
1		0	1	0	
toggle action	1	1	0	1	Toggle
	1	1	1	0	

Figure.6: JK-Latch truth table

D flip-flop:

At some definite portions of clock cycle the input data will be captured by the output. The captured value is nothing but the output. In D flip-flop the condition S = 1 and R = 1 is resolved. The following shows the truth tables for S and R configurations.

Flip flops are the essential part of many electronic devices. Shift registers are nothing but group of flip-flops. Whenever the device is clocked the flop captures the data and send it to output so, it is known as “transparent latch”.

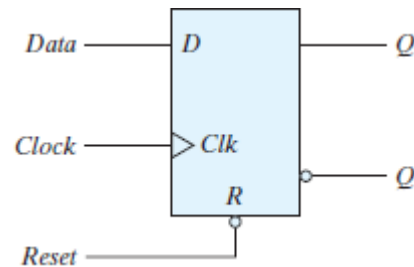


Figure.7: graphic symbol.

R	Clk	D	Q	Q'
0	X	X	0	1
0	↑	0	0	1
0	↑	1	1	0

Table.5:truth table

Pulse latch:

The pulse generator will generate the waveform by using a source clock. The transition is made by suitable pulse width. A simple pulse generator and the associated waveforms are shown in below figures. To satisfy several rules during clock-tree synthesis pulse generators are automatically inserted. It also uses a number of matching delays cells to allow for match clock insertion delays with or without pulse generators.

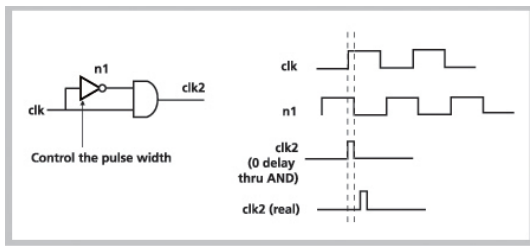


Figure.8: pulse controller.

4. Shift registers:

Registers are made up of group of flip flops and are used to store multiple bits of data. The input to the registers may be either serial or parallel that depends on the application. The data is stored and for every clock cycle it is shifted by one bit.

Serial in serial out Shift Registers:

The data given to these registers are in serial fashion. The data is given bit by bit on a single line. The output is also collected in the serially. Though the data that is entering bit by bit and it is collected at the output in same way as it is entering it is said to be serial in and serial out shift register. The shift register shifts the data to left or right that depends on application. In the above figure the data is shifted to left whenever the clock signal is triggered. This register is mainly used as temporary storage device.

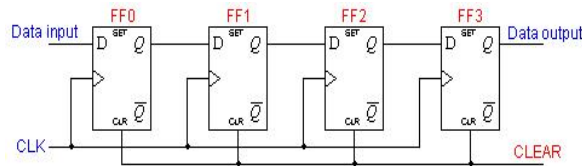


Figure.9: Serial in Serial out Shift Register

Serial in Parallel out Shift Register:

The data is entered in serial way but the output is collected in parallel in serial in parallel out shift register. In order to reset all the flip flops a clear signal is connected in additional to the clock signal. First flip flop output of is connected as input to the next flip flop. All the flip flops are driven by clock simultaneously. All the four outputs of flip flops are Q1, Q2, Q3 and Q4 respectively. The output data of the four flip flops are collected at a time.

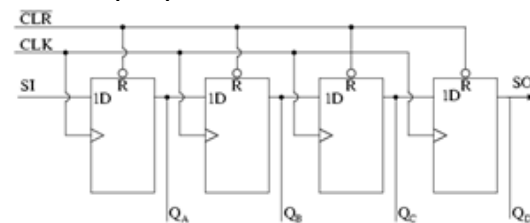


Figure.10: serial in parallel out.

The main use of this register is conversion of serial to parallel data. These types of registers are widely used in communication lines.

Parallel in Serial Out shift Registers:

The data will be loaded to all the flip flops simultaneously at a time when the clock is triggered. The data is shifted one bit for each clock pulse and is collected at the output bit by bit in serial way.

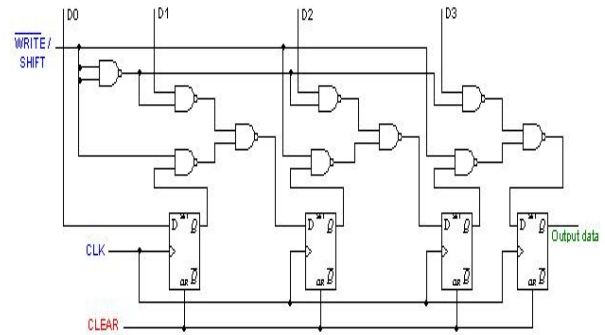


Figure.11: parallel in serial out.

Parallel in Parallel out Shift Registers:

Parallel in parallel out shift register acts as temporary storage device and as delay element.

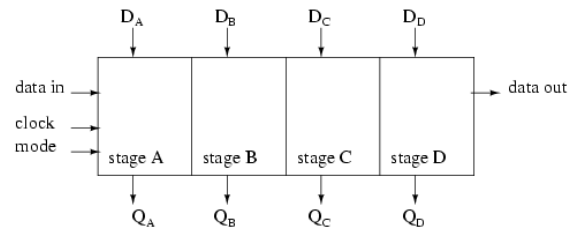


Figure.12: parallel in parallel out.

The data is fed to each flip flop individually and the output is collected individually from each flop.

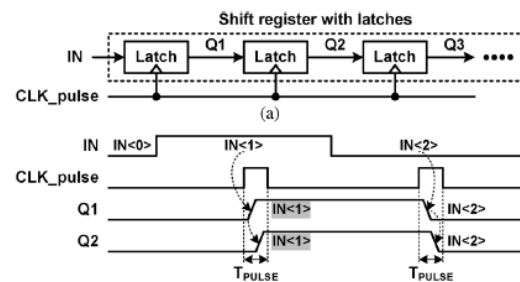


Figure.13: shift registers with pulsed latches and clock signals

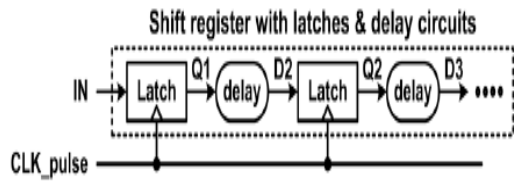


Figure.14: shift registers with delay circuits

Master slave flip flop consists of two latches cascaded in parallel. The pulse generator generates the pulse clock signal that is shared by all the flip flops.

The main use of the pulse latch when compared to master slave flip flop is the major factors area and power are very less. Pulse latch circuit will work well for the application of small area and less power consumption. The waveform for the pulse latch circuit is shown in below figure.

In above drawn waveforms we can observe the timing problem. Delay circuit are used to eliminate the timing problem in the pulsed latches. After a clock pulse the output signals of both the latches become equal. Before the clock pulse the input signals will remain constant.

Pulsed Latch architecture:

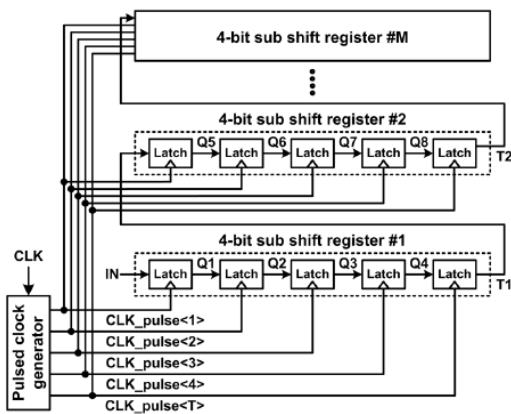


Figure.15: Pulsed Latch Architecture.

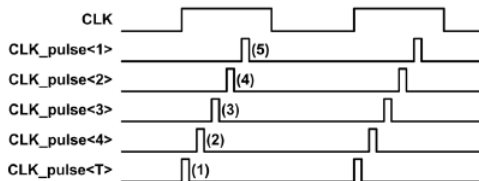


Figure.16: delayed clock pulses

Area and Power are the two major factors in any design. Optimization of Area and Power gives better result. In our design we use a latch which consumes more power. The power is consumed while the data

is transmitting and the clock is loading in clock circuit. Clock buffers are not considered in selection. The clock buffer size and its increase ratio are inversely proportional i.e., as the number of clock buffer increases their size will decrease. The increase ratio of clock buffers can be neglected by their size.

The minimum clock time delay is the delay of the rising edge of main clock signal to the rising edge first clock pulse. The delay time is selected as per maximum clock frequency of target application. At different intervals pulsed clock signals will be present at each sub shift registers due to pulse skew. The wire distance of delayed clock pulse generator increases as the pulse skew increases. Pulse skew will be different for pulsed clock signals as they arrive at different sub registers in different intervals, but pulse skew is same for those who arrive at same sub register in same time. The pulse skews which are larger for the clock pulses are cancelled out. The difference between sub shift registers does not cause any problem as they are connected to a long clock pulse interval.

Delayed clock pulse generator:

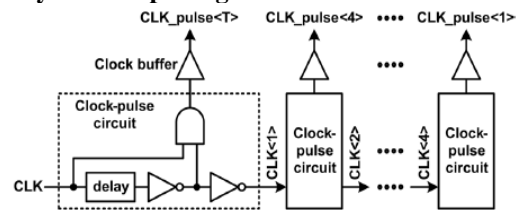


Figure.17: Delayed clock pulse generator

The shape of clock pulse is degraded due to wire capacitance and resistance. By increasing the clock pulse width we can get the exact shape of clock pulse without any degradation. But the maximum clock frequency will decrease as clock width increases for maintaining the shape of clock pulse.

5. Results:

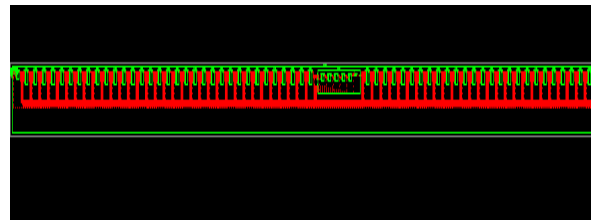


Figure.18: RTL Schematic.

Figure.19: RTL Internal Structure

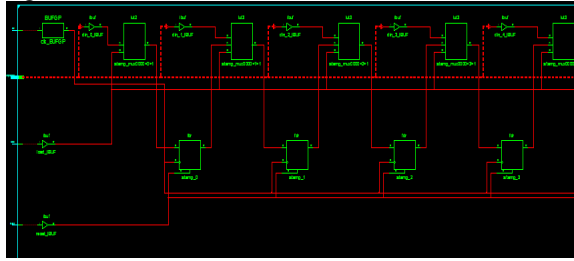


Figure.20: Technology Schematic

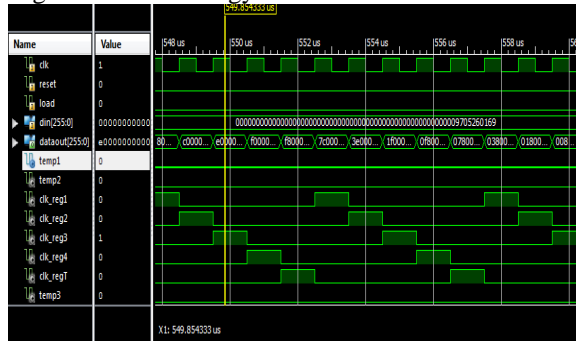


Figure.21: Simulation results and related timing diagrams

Conclusion:

This project proposes delay efficient by the use of pulsed latches. Pulsed latches give the same results but the hardware resources are reduced. Flip flops need two latches. By using a clocked latch with pulse signal as clock input the same results are obtained. This reduces the hardware requirement, in addition to this for producing different clock pulses for latches this project uses counter architecture for activating the latches. In this, counter is the circuit that increments its previous value by 1. In the increasing process several adders are implemented.

This project proposes koggle stone adder which takes 0.932 nano seconds where as the conventional adder takes 3.497 nano seconds which reduces thirds fourth of the conventional adder delay .

References:

[1] P. Reyes, P. Reviriego, J. A. Maestro, and O. Ruano, “New protection techniques against SEUs for moving average filters in a radiation environment,” *IEEE Trans. Nucl. Sci.*, vol. 54, no. 4, pp. 957–964, Aug. 2007.

[2] M. Hatamian et al., “Design considerations for gigabit ethernet 1000 base-T twisted pair transceivers,” *Proc. IEEE Custom Integr. Circuits Conf.*, pp. 335–342, 1998.

[3] H. Yamasaki and T. Shibata, “A real-time image-feature-extraction and vector-generation vlsi employing arrayed-shift-register architecture,” *IEEE J. Solid-State Circuits*, vol. 42, no. 9, pp. 2046–2053, Sep. 2007.

[4] H.-S. Kim, J.-H. Yang, S.-H. Park, S.-T. Ryu, and G.-H. Cho, “A 10-bit column-driver IC with parasitic-insensitive iterative charge-sharing based capacitor-string interpolation for mobile active-matrix LCDs,” *IEEE J. Solid-State Circuits*, vol. 49, no. 3, pp. 766–782, Mar. 2014.

[5] S.-H. W. Chiang and S. Kleinfelder, “Scaling and design of a 16-megapixel CMOS image sensor for electron microscopy,” in *Proc. IEEE Nucl. Sci. Symp. Conf. Record (NSS/MIC)*, 2009, pp. 1249–1256.

[6] S. Heo, R. Krashinsky, and K. Asanovic, “Activity-sensitive flip-flop and latch selection for reduced energy,” *IEEE Trans. Very Large Scale Integer. (VLSI) Syst.*, vol. 15, no. 9, pp. 1060–1064, Sep. 2007.

[7] S. Naffziger and G. Hammond, “The implementation of the nextgeneration 64 b titanium microprocessor,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2002, pp. 276–504.

[8] H. Partovi et al., “Flow-through latch and edge-triggered flip-flop hybrid elements,” *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, pp. 138–139, Feb. 1996.

[9] E. Consoli, M. Alioto, G. Palumbo, and J. Rabaey, “Conditional push-pull pulsed latch with 726 fJops energy delay product in 65 nm CMOS,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2012, pp. 482–483.

[10] V. Stojanovic and V. Oklobdzija, “Comparative analysis of masterslave latches and flip-flops for high-performance and low-power systems,” *IEEE J. Solid-State Circuits*, vol. 34, no. 4, pp. 536–548, Apr. 1999.

[11] J. Montanaro et al., “A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor,” *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1703–1714, Nov. 1996.

[12] S. Nomura et al., “A 9.7 mW AAC-decoding, 620 mW H.264 720p 60fps decoding, 8-core media processor with embedded forwardbody- biasing and power-gating circuit in 65 nm CMOS technology,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2008, pp. 262–264.

[13] Y. Ueda et al., “6.33 mW MPEG audio decoding on a multimedia processor,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2006, pp. 1636–1637.

[14] B.-S. Kong, S.-S. Kim, and Y.-H. Jun, “Conditional-capture flip-flop for statistical power reduction,” *IEEE J. Solid-State Circuits*, vol. 36, pp. 1263–1271, Aug. 2001.

[15] C. K. Teh, T. Fujita, H. Hara, and M. Hamada, “A 77% energy-saving 22-transistor single-phase-clocking D-flip-flop with adaptive-coupling configuration in 40 nm CMOS,” in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2011, pp. 338–339.