

A Novel Overlap-Based Logic Cell: An Efficient Implementation of Flip-Flops With Embedded Logic

Omid Sarbishei and Mohammad Maymandi-Nejad, *Member, IEEE*

Abstract—This paper presents several efficient architectures of dynamic/static edge-triggered flip-flops with a compact embedded logic. The proposed structure, which benefits from the overlap period, fixes most of the drawbacks of the dynamic logic family. The design issues of setting the appropriate overlap period for this architecture are explained. The proposed overlap-based approach is compared with several state-of-the-art dynamic/static logic styles in implementing a 4-bit shift register and an odd-even sort coprocessor using different CMOS technologies. The simulation results showed that the overlap-based logic cells become much more efficient when the complexity of their embedded logic function increases. Moreover, this approach improves static power consumption, which makes it even more efficient in below 0.18 μm CMOS technologies.

Index Terms—Clock overlap, digital ICs, edge-triggered flip-flops, static/dynamic logic family.

I. INTRODUCTION

AS DEEP submicrometer (DSM) CMOS technology has evolved during the last few years, researchers have developed several full-custom design techniques for digital circuits to improve design metrics like reliability, power consumption, performance, and area. These approaches can be divided into static and dynamic logic styles. Dynamic circuits are superior in terms of speed and area compared to static circuits. However, crosstalk and the clock routing issues are more troublesome in dynamic circuits.

During the past few decades, extensive studies have been done on the design of combinational circuits. The conventional CMOS logic (static or dynamic) and pass transistor logic (PTL) styles are two major architectures in implementing the logic circuits. The PTL family has been explored in the form of transmission gates (TGs) [1], complementary PTL (CPL), double PTL (DPL) [2], and gate diffusion input (GDI) [3] logic styles. In some logic functions like multiplexers, the TG logic is more efficient compared to the static CMOS architectures. However, the TG circuits become very slow in a large set of cascaded functions due to the RC delay and body effects. Under this condition, buffers must be inserted between cascaded TGs. The CPL

logic style is efficient in terms of area and dynamic power consumption due to the reduction of p-type MOS (pMOS) transistors and voltage swing in the internal nodes. However, they need intermediate buffers to repair the low-swing nodes and they suffer from static power consumption. Moreover the low-swing nodes make the circuit very prone to noise, and as a result, the low-swing PTL structures are typically avoided to maintain the robustness of the design. The DPL circuits add pMOS transistors to the CPL structures to provide full-swing nodes. The GDI technique uses n-type MOS (nMOS)–pMOS two-transistor cells to implement a logic function with reduced complexity. Like the CPL circuits, the voltage swing of the internal nodes is typically low, which makes the GDI circuits inefficient and prone to noise in new technologies.

In the design of sequential circuits, a major challenge is the design of an efficient D-flip-flop (DFF). Several static/dynamic DFF architectures have been proposed in [1] and [4]–[12]. The static GDI DFF proposed in [4], like the combinational GDI implementations, reduces the internal node voltage swings to improve the dynamic power consumption. However, it suffers from the aforementioned problems in GDI combinational circuits. The static DFF in [1] uses fewer pMOS transistors compared to the one in [4], and can be converted to a power-delay-product (PDP) efficient push-pull architecture proposed in [5]. All these structures are sensitive to the clock overlap. In order to eliminate the problem of clock overlap, several architectures are proposed. The dynamic single-clock DFFs in [1] are not sensitive to clock overlap, but suffer from charge-sharing problems. The dynamic single-clock charge-sharing-free DFF in [6] was proposed for binary ripple counters; however, this kind of DFF suffers from large set-up time and propagation delay. Another category of DFFs, which not only does not suffer but also benefits from the overlap period of the clock signals, is called hybrid latch flip-flop (HLFF) [7]. The DFFs in [7] are also able to have negative set-up times. Another structure for the overlap-based DFFs is proposed in [8], which is very efficient in terms of power consumption, delay, and area. Moreover, they have larger acceptable overlap periods and lower minimum allowable overlap times [8].

In this paper, a revised structure of the overlap-based DFFs in [8] is proposed. The new architecture is capable of embedding logic functions into the overlap-based FF and is an efficient architecture for designing control units and pipeline datapath structures. The proposed logic cell implements the logic functions more efficiently compared to the conventional dynamic logics. It performs a charge-sharing-free operation, while reducing area, power, and delay. The detailed design issues of these architectures are presented in the paper. The overlap-based

Manuscript received March 26, 2008; revised July 27, 2008. First published April 07, 2009; current version published January 20, 2010.

O. Sarbishei is with the Electrical Engineering Department, Sharif University of Technology, Tehran, Iran (e-mail: sarbishei@ee.sharif.edu; omid_sarbishei@yahoo.com).

M. Maymandi-Nejad is with the Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran (e-mail: maymandi@um.ac.ir).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>

Digital Object Identifier 10.1109/TVLSI.2008.2009453

logic cell is compared with several state-of-the-art dynamic/static FFs in implementing a 4-bit shift register. In order to emphasize the efficiency of this approach, the operation of the circuits are evaluated in all the process/temperature corners with several load capacitances and supply voltages using different CMOS technologies. We have also evaluated the robustness of the proposed dynamic logic cell with respect to cross coupling in 45 nm CMOS technology. Moreover, the proposed architecture has been evaluated in an odd–even sort coprocessor, which is widely used in many digital signal processing applications. The simulation results show that the proposed logic cells become more efficient when the complexity of their embedded logic function increases. Moreover, their power consumption is lower in advanced DSM technologies due to their superior operation in terms of leakage current.

The rest of the paper is organized as follows. In Section II, the proposed logic cell with its functionality and design issues are presented. Section III compares the efficiency of the proposed logic cell with several state-of-the-art dynamic/static architectures in a 4-bit shift register. The results of the postlayout simulations in 0.18 μm CMOS technology and schematic simulations in 90, 65, and 45 nm are provided to explore the impact of the leakage in more advanced technologies. In Section IV, the proposed logic cell is used to design a stable sort coprocessor and the improvements are explored. Finally, conclusions are drawn in Section V.

II. THE PROPOSED OVERLAP-BASED LOGIC CELL

In this section, the proposed logic cell is presented. The clock overlap is typically troublesome in sequential circuits. However, the proposed logic cell not only does not suffer but also benefits from the overlap period. Using this period, results in an efficient logic cell that has advantages over conventional dynamic logic styles.

A. Basic Operation

The proposed logic cell embeds a lookup table (LUT) into an overlap-based DFF as shown in Fig. 1. The LUT has been implemented by a pull-down network (PDN). Note that if the PDN is replaced with only one nMOS transistor, the logic cell operates as a single DFF.

The clock (clk) signal and its complement (clk') are used to latch and hold the data. In order for the circuit to operate properly, the clk must lead the clk' . As will be explained later, this structure makes a positive edge-triggered overlap-based FF with an embedded logic. The two clock signals are made in such a way as to have a 1–1 overlap. The 0–0 overlap has no effect on the operation of the circuit. It is possible to make a negative-edge-triggered cell by swapping the position of the clk and clk' signals.

The operation of the proposed logic cell can be divided into two modes as follows.

- 1) *Evaluation Mode*: This happens during 1–1 overlap of clk and clk' signals. During this phase, transistors M_2 , M_3 , and M_5 are all on. The second stage behaves like a simple inverter and the data can pass through the cell and reach the output. Note that before this phase starts, clk has been 0 and V_1 is V_{DD} at the beginning of this mode.

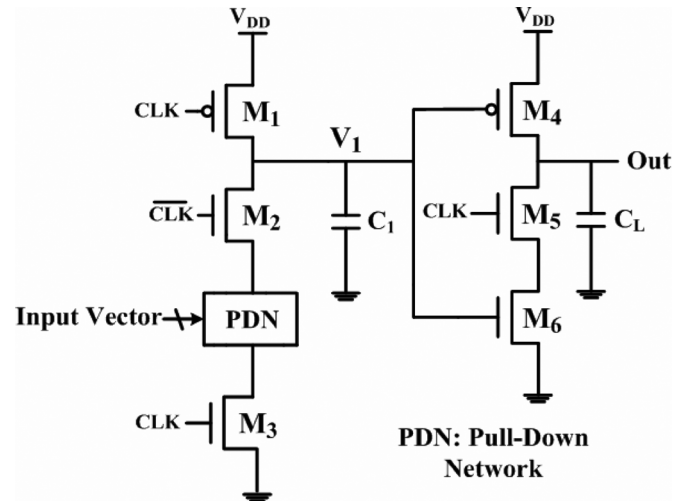


Fig. 1. Proposed overlap-based logic cell—a positive-edge-triggered FF with embedded logic.

- 2) *Holding Mode*: During this mode, the internal node V_1 is disconnected from the input PDN, while clk and clk' can have the following values:

- a) The 1–0 sequence ($\text{clk} = 1$ and $\text{clk}' = 0$): During this period, V_1 has a value that is the inverse of the input PDN and has been stored during the evaluation mode. Moreover, V_1 is inverted and passed to the output.
- b) The 0–X sequence ($\text{clk} = 0$ and $\text{clk}' = X$): Under this condition V_1 is disconnected from both the input PDN and the output node and is pre-charged to V_{DD} .

The previous discussion implies that the proposed structure can have a negative set-up time like the HLFFs.

B. Comparison With Conventional Dynamic Logics

It may seem that the proposed architecture operates the same as a dynamic combinational circuit, which has been merged with a DFF. However, there are some major drawbacks in dynamic circuits that are not troublesome in the proposed architecture. These issues are discussed as follows.

- 1) *Complementary Input Signals*: The dynamic circuits can be cascaded in domino or zipper styles. For these architectures, the input signals cannot be changed more than once during the evaluation mode. Moreover, the transitions of input signals must occur in only one direction. As an example, in cascaded PDN domino logic circuits, only the low-to-high transitions are allowed during the evaluation mode, because the output capacitance of the dynamic circuit may be wrongly discharged and it would not be possible to recharge it during the same evaluation phase. Based on the aforesaid discussion, if a PDN/pull-up network (PUN) needs complementary input signals (X and X'), it cannot be implemented with a dynamic architecture in cascaded dynamic circuits. The reason is that it is not possible to provide a similar transition direction for two complementary signals during the evaluation mode. As an example, consider the circuits in Fig. 2. In this figure, the logic function of a 2×1 multiplexer has been implemented by a PDN and it is merged with a DFF. The same circuit using overlap-based logic cell is also shown in Fig. 2. The select signal and its complement (SEL and

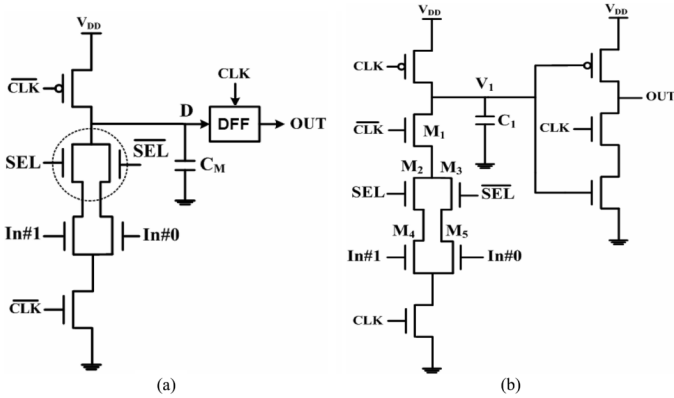


Fig. 2. (a) Unacceptable cascade of a dynamic multiplexer and a positive edge-triggered DFF. (b) A multiplexer embedded into an overlap-based DFF.

SEL') are necessary in realizing a 2×1 multiplexer with a PDN. For the circuit in Fig. 2(a), the two input signals SEL and SEL' should not have a high-to-low transition during the evaluation mode, because the output load capacitance of the multiplexer (C_M) may be wrongly discharged and it cannot be recharged in the same clock cycle. Providing only a low-to-high transition for both SEL and SEL' during the evaluation mode is not achievable in cascaded circuits. Note that regarding this issue, it does not matter whether SEL and SEL' are realized by a dynamic or static cell. One way to use the circuit in Fig. 2(a) is to have static realizations for SEL and SEL', while adding a timing constraint of having all the complementary input signals of the dynamic PDN fixed during the evaluation phase [i.e., during $\text{clk}' = 1$ for the circuit in Fig. 2(a)]. This issue implies having a hold time of about $T/2$ for the input signals of the PDN, where T is the clock period. Allocating about 50% of the clock period to only one dynamic combinational stage widely impacts the maximum achievable frequency and performance. Therefore, the architectures like XOR gates or any LUT, which need complementary inputs, are implemented with static CMOS logics or PTL styles.

On the other hand, the proposed PDN-based logic cell can implement any logic function in a cascade circuit, because the transistor M_2 in Fig. 1 limits the evaluation phase of the circuit to the 1-1 overlap and the inevitable changes of input signals during other clock sequences (0-0, 1-0, and 0-1) do not wrongly discharge V_1 . In other words the input signals of the PDN must be fixed only during the 1-1 overlap period (instead of half of the clock period). This issue can easily be achieved by considering a nonzero hold time for the circuit. This hold time is equal to the overlap period of the clocks and it is much lower than $T/2$. Hence, in contrast to the conventional dynamic logics, adding this timing constraint does not much impact the performance. As will be explained in the following sections, in contrast to the conventional overlap-based architectures like HLFFs, the proposed logic cells can be designed to function with very short overlap periods (hold times). They also accept larger overlap ranges.

2) *Charge Sharing*: Charge sharing between the internal nodes of the PDN and the output node of a dynamic circuit can be problematic in dynamic circuits. It can be resolved by adding a few transistors to the output node. However, this

approach leads to an overhead in area, power, and performance. Another solution is to provide a hold time of $T/2$ during the evaluation phase, while having a negligible positive set-up time to precharge the internal capacitances of the PDN before the evaluation phase starts. As explained before, having a hold time of about $T/2$ is not a good approach in terms of performance. In the case of the proposed cell, charge sharing can be avoided by simply considering a negligible positive set-up time, while having a hold time equal to the overlap period. This issue is discussed next.

If we consider a positive set-up-time for the cell, before the end of the 0-1 sequence, the input signals of the PDN are fixed, and as a result, the internal capacitances in the PDN are charged to $V_{DD} - V_T$, where V_T is the threshold voltage of the nMOS transistors. In this way, when the 1-1 overlap period begins, these precharged capacitances do not impact V_1 and the charge sharing issue is avoided.

It can be deduced from the aforesaid discussion that overlap-based logic cells make it possible to implement any logic function with a PDN, while avoiding charge sharing and keeping the maximum achievable frequency fixed.

C. Design Considerations

As the load capacitance on the clock circuitry is typically high, the clock overlap is an inevitable issue. However, in some design approaches, the nonoverlapping clocks are generated. This issue needs complex clock generators, and it is not suitable for applying the clock gating technique in large processors [13]. As a result the single-clock FFs or overlap-insensitive FFs are mostly used in chip-level designs to maintain reliability issues. It is worth noting that for overlap-insensitive FFs like C^2 MOS, the overlap-time is a compulsory overhead in set-up time and clock-to-output propagation delay. However, the proposed logic cell benefits from this overhead period. In this section, the design issues of the proposed logic cell are discussed in a general pipeline structure. It will be seen that the proposed cell is able to accept a wide overlap period within process/temperature variations. Moreover, the impacts of clock skew and clock jitter are included in the discussion.

The main design issue in the proposed logic cell is the overlap period between clk and clk' . For the logic cell in Fig. 1, the overlap time (T_{OV}) should be long enough such that the data can pass through the first inverter and discharge the parasitic capacitance C_1 ($T_{OV} > T_{C1}$). Otherwise transistor M_2 turns off and disconnects the input signals from V_1 . On the other hand, T_{OV} should not be too large; otherwise, the signal may flow through to the next cell during the evaluation mode. This means that T_{OV} should be smaller than the minimum time needed to charge the output capacitance C_L (T_{C1} plus the delay of M_4 in the second stage in Fig. 1 plus the minimum delay of the combinational circuit between the logic cells). As an example, consider the circuit in Fig. 3. In this figure, two logic cells are depicted by $\text{cell}\#i$ and $\text{cell}\#(i+1)$. The two parameters $T_{\text{clk}}(i)$ and $T_{\text{clk}b}(i)$ refer to the routing delays of clk and clk' for $\text{cell}\#i$. Hence, if T_{OV} is the overlap-period of the reference clocks, then the overlap period of $\text{cell}\#i$ is

$$T_{OV}(i) = T_{OV} + T_{\text{clk}b}(i) - T_{\text{clk}}(i). \quad (1)$$

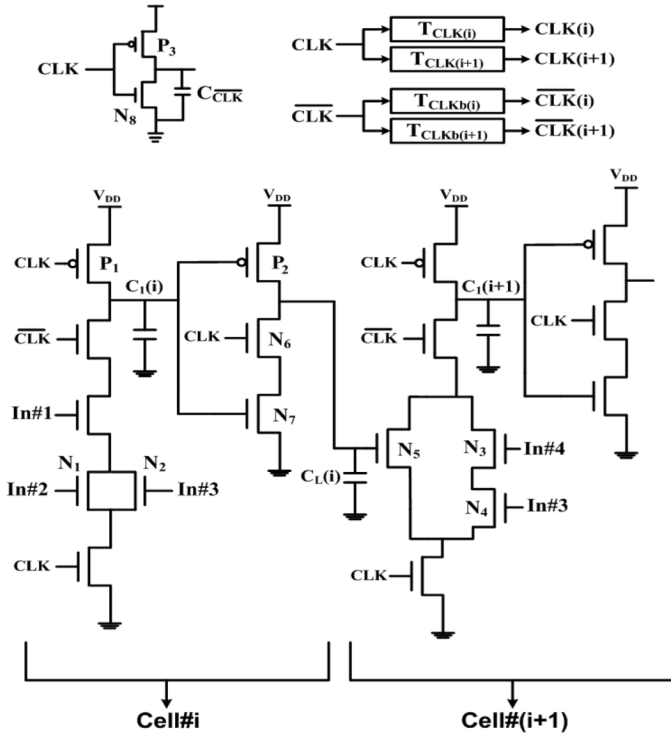


Fig. 3. Two-stage sequential circuit with overlap-based logic cells.

As explained earlier, the acceptable overlap range can be increased by the combinational delay between the logic cells. In this way, we did not add any combinational circuits between the two cells in Fig. 3 to address the minimum range of acceptable overlap periods. The acceptable range of $T_{OV}(i)$ is governed by the following equations:

$$\text{Max}\{T_{C1}(i)\} < T_{OV}(i), \quad i = 1, 2, \dots, n \quad (2)$$

$$T_{OV}(i+1) < \text{Min}\{T_{CL}(i)\} + \text{Min}\{T_D(i)\} + \hat{T}_{C1}(i+1) - T_{\text{skew}}(i) - 2T_{\text{jitter}}. \quad (3)$$

Equation (2)/(3) shows the minimum/maximum acceptable overlap period for each logic cell. In (2), $\text{Max}\{T_{C1}(i)\}$ is the maximum discharging delay of the parasitic capacitance C_1 in the i th logic cell. In the clock distribution network, if $T_{\text{clk}}(i)$ exceeds $T_{\text{clkb}}(i)$ so that the overlap period of $\text{cell}\#i$, i.e., $T_{OV}(i)$, is not long enough to discharge its internal capacitance C_1 , then we should generate a separate clk' from the received clk in $\text{cell}\#i$ to provide it an appropriate overlap period. Note that under such a condition, $T_{OV}(i)$ becomes independent of the routing delays and does not satisfy (1). In other words, for overlap-based logic cells, global routing of clk' is not an option in chip-level designs and local clk' signals are generated from the global clk for specific parts of the chip.

In addition to the minimum acceptable overlap period, (3) indicates that T_{OV} should not be too long to cause a flowthrough operation. This limitation is not just related to the proposed logic cell and is troublesome for almost every DFF and even for the single clocked circuits due to the existence of clock skew and clock jitter. For all sequential circuits, an equation like (3) is defined to determine the maximum hold time, which is equal to the

overlap period in the proposed logic cells. In (3), $\text{Min}\{T_{C1}(i)\}$ is equal to the minimum discharging delay of C_1 plus the delay of P_2 in charging the output of the second stage of the i th cell ($C_L(i)$)

$$\text{Min}\{T_{C1}(i)\} = \text{Min}\{T_{C1}(i)\} + T_{P2}(i). \quad (4)$$

In the circuit of Fig. 3, maximum/minimum value of $T_{C1}(i)$ happens when one/both of the nMOS transistors: N_1 and N_2 is/are turned on. The other parameter $\text{Min}\{T_D(i)\}$ in (3) is the combinational delay between the i th logic cell and its next stage ($i+1$)th cell. In the case of the circuit in Fig. 3, T_D is equal to zero. $\hat{T}_{C1}(i+1)$ is the discharging delay of C_1 in the ($i+1$)th logic cell. Note that the delay of $\hat{T}_{C1}(i+1)$ in the PDN of the ($i+1$)th cell must include a backward path to the i th cell. This is due to the fact that (3) is based on the possibility of a flow through operation between two cells, and as a result, there must be a valid path between the two cells to cause this issue during the overlap period.

The other parameter in (3) is $T_{\text{skew}}(i)$, which refers to the clock skew between two logic cells and is equal to $T_{\text{clk}}(i+1) - T_{\text{clk}}(i)$. Note that when data and clock have similar directions in the datapath, T_{skew} is positive, and as a result, the condition in (3) becomes more critical. Otherwise, it is negative, leading to a larger acceptable overlap range. Also, T_{jitter} in (3) refers to the clock jitter value, which indicates the maximum variation of clock period. Based on (1), the value of $T_{\text{clkb}}(i)$ may exceed $T_{\text{clk}}(i)$ so that $T_{OV}(i+1)$ becomes too big and leads to a flow-through operation. Under such a condition, we can generate a separate clk' for $\text{cell}\#(i+1)$ or increase the combinational delay, i.e., $T_D(i)$, between the two cells (e.g., by adding inverters) to increase the maximum acceptable overlap period and guarantee a proper operation.

In order to satisfy the acceptable T_{OV} in (2) and (3), we should set a proper falling delay for the clk' signal. This issue can be achieved by proper sizing of the nMOS transistor of the inverter making clk' from clk (N_8 in Fig. 3).

Increasing the channel widths of the nMOS transistors in the first stage's PDN in an overlap-based cell reduces T_{C1} , and as a result, the minimum/maximum acceptable T_{OV} in (2)/(3) is decreased. In other words, stronger nMOS transistors in the PDN can reduce the minimum acceptable overlap time. Fig. 4 explores this issue for the circuit in Fig. 3 in 0.18 μm technology with the supply voltage of 1.8 V. In this figure, the minimum value of acceptable overlap time versus the size of nMOS transistors in the PDN of $\text{cell}\#i$ is illustrated. In these simulations, the minimum T_{OV} is equal to the maximum discharging delay of $C_1(i)$ in Fig. 3, which happens when $\text{In}\#1$ is high, $\text{In}\#2/\text{In}\#3$ is low, and $\text{In}\#3/\text{In}\#2$ is high. Since the value of the overlap is critical for proper operation of the logic cells, the simulation of Fig. 4 is done for all the process/temperature corners, but only the results of three process corners are depicted. Simulations show that the slow nMOS, slow pMOS (SS) process corner constitutes the worst case of minimum acceptable overlap time in (2). Therefore, if the requirement in (2) is fulfilled in the SS corner for all the logic cells, it is valid for other process/temperature corners too. Also, it can be seen in Fig. 4 that if we use

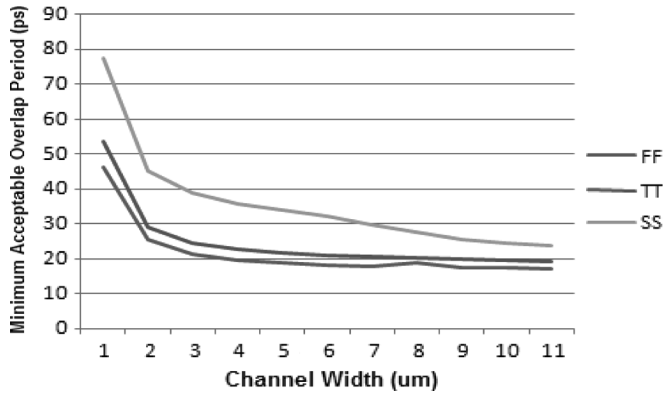


Fig. 4. T_{C1} versus the size of an nMOS transistor in the PDN of cell# i at three different process corners.

stronger transistors in the PDN, the variation of T_{OV} within different process/temperature corners is reduced.

Another evaluation has been made to check the acceptable overlap range in all the process/temperature corners for the circuit in Fig. 3. We did not add any external load capacitances to the two cells to cover the minimum value (worst case) of T_{cl} in (3). No clock skew has been considered between the cells. Note that if any skew exists between the cells, we can increase $T_D(i)$ so that it becomes dominant over $T_{skew}(i)$. Under such a condition, the following simulated acceptable overlap range becomes valid again.

Fixed sizing for typical nMOS and typical pMOS (TT) transistors have been taken into consideration to achieve the overlap time of $T_{OV} = 150$ ps at $T = 25^\circ\text{C}$. Then, several process/temperature corners have been simulated to track the variations of T_{OV} and the acceptable overlap periods in (2) and (3). The simulation results are summarized in Table I. In this table, the variation of maximum and minimum acceptable overlap time and also the variation of the initially set overlap time ($T_{OV} = 150$ ps) within process/temperature variations are shown. In order to obtain the maximum acceptable T_{OV} in (3), we need to calculate the values of $\text{Min}\{T_{cl}(i)\}$ and $\hat{T}_{C1}(i+1)$. The value of $\text{Min}\{T_{cl}(i)\}$ takes place when all the nMOS transistors in the PDN of the i th cell are turned on. Moreover, $\hat{T}_{C1}(i+1)$ happens when N_5 is turned on and N_3/N_4 is turned off. Table I shows that the fast nMOS and fast pMOS (FF) corner combined with the highest temperature constitutes the minimum acceptable overlap range and the minimum value of maximum acceptable overlap time. Therefore, if the requirement in (3) is fulfilled in the FF corner with the highest temperature for all the logic cells, it is valid for other process/temperature corners too. However, as the output load capacitance is typically high and also a combinational circuit exists between cells, the requirement in (3) is rarely a major issue and we have a much wider range of acceptable T_{OV} compared to the results in Table I.

The fact that the minimum allowable T_{OV} is almost independent of the load capacitance provides an advantage for the proposed logic cell compared to other HLFFs where the minimum T_{OV} is a function of the load capacitance. Hence, in the proposed cell, the overlap between clk and clk' can be realized by only one inverter. This is not the case for HLFFs where depending on the load capacitance, the number of required in-

TABLE I
VARIATION OF T_{OV} WITHIN PROCESS/TEMPERATURE VARIATIONS

Process Corner	T ($^\circ\text{C}$)	Minimum Acceptable T_{OV} (ps)	T_{OV} (ps)	Maximum Acceptable T_{OV} (ps)
SF	80	90	183	306
FS	-40	46	141	202
SS	80	96	207	444
SS	-40	66	185	270
FF	80	57	138	174
FF	-40	43	147	166
TT	25	64	150	236

verters increases. Another drawback of HLFFs is that their range of acceptable overlap period is much smaller than the proposed cells. This fact makes the HLFFs unreliable within process/temperature variations.

In order to avoid the flowthrough operation, the sizing of transistors N_6 and N_7 in Fig. 3 should be done so that

$$\hat{T}_{C1}(i+1) < T_{N6}(i) + T_{N7}(i) + T_D(i) \quad (5)$$

where " $T_{N6}(i) + T_{N7}(i)$ " is the time needed to discharge the output capacitance C_L of the i th cell. The requirement given in (5) is not a major issue for the logic cell since C_L is typically much larger than the internal capacitance C_1 . Moreover, the low-to-high propagation delay of a logic cell includes two stages (T_{C1} and T_{cl}), while the falling delay includes only one stage (N_6 and N_7). In this way, in order to have similar rising and falling propagation delays, we should have

$$T_{C1}(i) + T_{cl}(i) = T_{N6}(i) + T_{N7}(i). \quad (6)$$

The condition in (6) can be satisfied if N_6 and N_7 are weaker than the PDN transistors in the first stage. Using this typical sizing approach satisfies the condition in (5) as well.

The proposed cell has some other features presented next.

- 1) *Slack-Passing*: In conventional master-slave FFs, data must be flown through the FF in two stages, which constitutes two separate delays of set-up time and output delay. As a slack phase occurs between the two evaluation phases of master and slave, the maximum achievable frequency is much lower compared to an overlap-based logic cell that includes only one evaluation phase. The HLFFs also have this advantage.
- 2) *Restoring Low-Swing Input Signals*: In an overlap-based logic cell, the input signals are connected to the gate terminals of a PDN and no PUN is used. Therefore, the low-swing input signals can be restored without making statically turned-on pMOS transistors. This issue is not achievable in traditional master-slave FFs and HLFFs.
- 3) *Low Leakage Current*: Due to the full-swing nodes, and low complexity in realizing logic functions, the static power is efficient. In Section III, the simulation results address the leakage efficiency of the proposed logic cell.
- 4) *Compatibility With Clock-Gating and Clock-Tree Structures*: Clock gating reduces the number of clock-driven components in a design to improve the switching activity

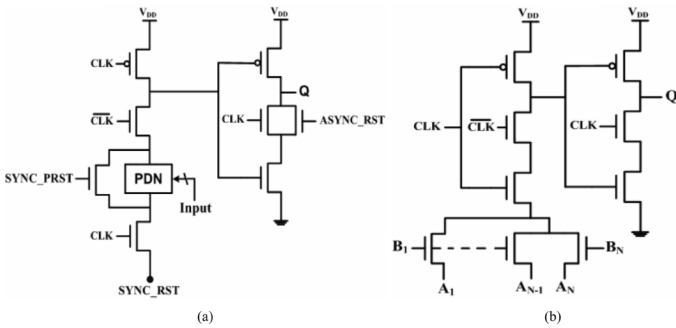


Fig. 5. Revision of overlap-based logic cells. (a) Logic cell with control signals. (b) Function specific PTL cell.

on the clock signal. The clock-tree architectures also distribute separate clock signals to reduce the clock skew. In this way, these two methods are suitable for applying the overlap-based techniques as they reduce n in (2) and (3) and make these conditions easier to satisfy. Therefore, the overlap-based logic cells are applicable to chip-level designs with clock-gating and clock-tree structures like the conventional FFs.

- 5) There is no need for additional inverters at the output of the cell since its second stage is evaluating only during the high period of clk (no flowthrough operation). This is not achievable in the HLFF flip-flops.

D. Revised Overlap-Based Logic Cells

The proposed logic cell in Fig. 1 can be revised into similar architectures to obtain specific functionalities. These revised structures are discussed in this section.

1) *Logic Cells With Control Signals*: Fig. 5(a) shows a positive edge-triggered logic cell with several control signals. The active-high synchronous reset (SYNC_RST) and preset (SYNC_PRST) signals have been added to the input PDN. In this figure, SYNC_RST is more prior than SYNC_PRST, which means that if both of these signals are activated, the cell will be reset. However, this priority can easily be changed by modifying the PDN. There is also a pseudoasynchronous reset signal (ASYNC_RST), which can be active only during the low period of clock. Based on the application, some of these control signals may be used for the cell. Note that the simple procedure of adding these control signals does not create a static short-circuit path.

2) *PTL Logic Cells*: Fig. 5(b) illustrates a revised PTL logic cell, which implements: $F = A'_1 B_1 + A'_2 B_2 + \dots + A'_n B_n$. As the input A_i signals are not connected to a gate terminal, there must be a nonzero combinational delay between an A_i signal and its previous sequential stage to hold the logic level of the input data in the evaluation mode and avoid the flowthrough operation (typically an inverter is sufficient for the intermediate combinational delay).

3) *Split-Merge Cells*: Some logic cells may need to share a PDN. Fig. 6(a) illustrates an implementation of two cells sharing a PDN. Due to the quadratic increment of propagation delay with respect to the number of stacked transistors in a PDN, we can split a deep-stacked PDN in a logic cell into smaller PDNs to improve performance. Then, the split PDNs will be merged

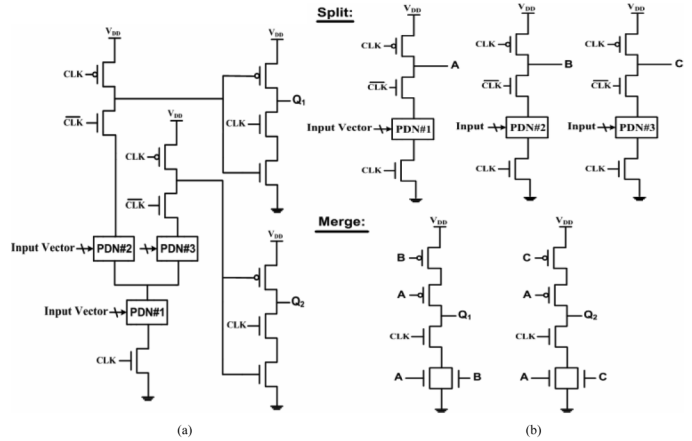


Fig. 6. Revision of overlap-based logic cells. (a) Sharing a PDN between two logic cells. (b) The equivalent split-merge process.

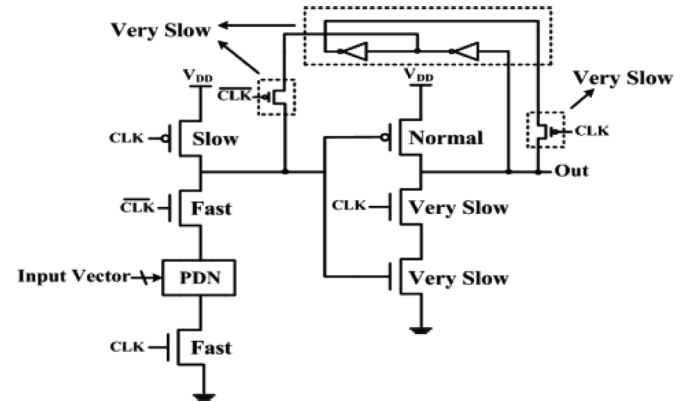


Fig. 7. Pseudostatic overlap-based logic cell with its typical transistor sizing.

by adding transistors to the second stage of the cell. A sample splitting process is shown in Fig. 6(b). In this figure, the two logic cells in Fig. 6(a) have been split into smaller PDNs. The second stage of each cell merges its input PDNs with a two-input NOR gate. This technique is not only suitable for reducing the number of stacked transistors, but also improves the acceptable overlap range in (2) and (3) as it makes the first stage of the logic cell faster, while the second stage becomes slower.

4) *Pseudostatic Logic Cells*: The proposed dynamic logic cells can be revised into an efficient pseudostatic architecture to increase noise immunity and also be applicable for low-frequency clock-gated FFs. The revised pseudostatic architecture of the logic cell in Fig. 1 is shown in Fig. 7. In this figure, six weak transistors are added to the circuit to hold the logic value at the output and internal nodes that may otherwise float during some period of time. It is worth noting that these transistors are typically small.

III. COMPARISON WITH OTHER LOGIC STYLES

In this section, the proposed logic cell is compared with several state-of-the-art dynamic/static logic styles in a 4-bit shift register with synchronous reset (SYNC_RST) in different CMOS technologies. Several implementations of dynamic and static DFFs are shown in Figs. 8 and 9, respectively. For the PTL DFFs depicted in Fig. 8(b), and Fig. 9(a) and (c), the

TABLE II
POSTLAYOUT SIMULATION RESULTS OF A 4-bit SHIFT REGISTER WITH SYNCHRONOUS RESET USING DYNAMIC DFFS

Dynamic DFF Architecture	Area (μm^2)	$C_L=10fF, V_{DD}=1.1V$			$C_L=10fF, V_{DD}=1.3V$			$C_L=20fF, V_{DD}=1.5$			$C_L=50fF, V_{DD}=1.8V$		
		Power (μW)	Total Delay (ps)	PDP (fJ)	Power (μW)	Total Delay (ps)	PDP (fJ)	Power (μW)	Total Delay (ps)	PDP (fJ)	Power (μW)	Total Delay (ps)	PDP (fJ)
Overlap-Based (Proposed)	125.6	40.6	198.1	11.5	59.3	153	9.07	93.7	175.7	16.4	182.7	250.2	45.7
C ² MOS in [1]	244.8	Failed in 1GHz			72.2	454.7	32.8	112.9	390.5	44.1	189.9	469.6	89.2
Single CLK in [6]	334.7	Failed in 1GHz			158.8	483.7	76.8	251.3	400.6	100.1	481.6	423.3	204
Classic in [1]	252.2	Failed in 1GHz			Failed in 1GHz			188.9	419.7	79.3	378.6	433.4	164

TABLE III
POSTLAYOUT SIMULATION RESULTS OF A 4-bit SHIFT REGISTER WITH SYNCHRONOUS RESET USING STATIC DFFS

Static DFF Architecture	Area (μm^2)	$C_L=10fF, V_{DD}=1.1V$			$C_L=10fF, V_{DD}=1.3V$			$C_L=20fF, V_{DD}=1.5V$			$C_L=50fF, V_{DD}=1.8V$		
		Power (μW)	Total Delay (ps)	PDP (fJ)	Power (μW)	Total Delay (ps)	PDP (fJ)	Power (μW)	Total Delay (ps)	PDP (fJ)	Power (μW)	Total Delay (ps)	PDP (fJ)
Overlap-Based (Proposed)	331.8	70	222.1	15.5	110.4	153.4	16.9	169.5	155.4	26.3	321.3	198.2	63.6
HLFF in [7]	393.9	Failed in 1GHz			162.6	215.2	32.8	241.1	198.3	47.8	445.9	221.2	98.6
Classic in [1]	416.9	Failed in 1GHz			Failed in 1GHz			211.2	276.6	58.4	376.3	397.9	149

synchronous reset function is realized by a PTL structure as shown in Fig. 10(a). For other conventional DFFs, the reset logic has been realized by expanding the internal PDN/PUN as shown in Fig. 10(b).

To compare the proposed architecture with conventional structures, the circuit in Fig. 5(b) is used for implementing the dynamic overlap-based DFFs with synchronous reset. In this figure, we put $N = 1$, $A_N = \text{SYNC_RST}$, and $B_N = D$ to achieve a DFF with synchronous reset. The static revision technique in Fig. 7 is also applied to this dynamic DFF to provide the static-overlap-based DFF structure. The circuits are implemented with several supply voltages and load capacitances in 1 GHz clock frequency. The clock rise time is 100 ps. The transistor-sizing procedure in [1] is used to obtain the same rising and falling delays for all the DFFs. All the process/temperature corners are taken into consideration. The postlayout simulation results in 0.18 μm CMOS technology are summarized in Tables II and III. In these tables, *total delay* is the clock-to-output delay plus the set-up time.

The C²MOS DFF in Fig. 8(a) is a power-delay efficient dynamic structure. However, it is not as fast as the other dynamic FFs under higher load capacitances and supply voltages. The classic dynamic DFF in Fig. 8(b) reduces the number of transistors. However, it suffers from internal low-swing nodes that widely impact static power consumption. Moreover, these low-swing nodes drastically increase the propagation delay in lower supply voltages and make the circuit really prone to noise. It can be seen that this FF could not operate at 1 GHz frequency with the supply voltages lower than 1.5 V. One solution to this issue is to add restoring pMOS transistors to the circuit to improve performance and static power consumption. The inverting single-clock DFF in Fig. 8(c) is suitable for implementing binary ripple counters as explored in [6]. However, in the case of other types of sequential circuits, it is not as efficient as the C²MOS DFF in terms of PDP.

The HLFF in Fig. 9(b) is an efficient architecture that provides a slack passing operation to achieve higher performance, but it needs a wide overlap period to operate properly. Therefore, at least three inverters are necessary to realize T_{OV} for this circuit. Moreover this FF is not power-efficient. The classic DFF in Fig. 9(a) is the pseudostatic version of the FF in Fig. 8(b). The GDI DFF in Fig. 9(c) is also a PTL structure that provides differential output signals. These two FFs suffer from the low-swing node problems like the FF in Fig. 8(b). On the other hand, the proposed logic cell embeds the reset function into an overlap-based DFF and performs the superior operation in terms of power, delay, and area in all the process/temperature corners and it can also operate in lower supply voltages.

In another experiment, we evaluated the leakage efficiency of the proposed dynamic overlap-based 4-bit shift register and the C²MOS architecture in below 0.18 μm CMOS technologies. The SPICE schematic model for 90, 65, and 45 nm CMOS technologies has been used to evaluate the average power consumption of the circuits. The supply voltages were 1, 0.8, and 0.6 V, respectively. In the simulations, we performed the sizing of transistors to achieve a clock-to-output delay of 70 ps in 1 GHz clock frequency and a load capacitance of $C_L = 20$ fF. Moreover, the set-up time of the C²MOS DFF was about 110 ps, while the overlap-based shift register had a zero set-up-time. The average power consumption of the circuits is illustrated in Fig. 11. As can be seen in more scaled-down technologies in which the leakage power becomes dominant over switching power, the proposed architecture becomes even more efficient.

We have also evaluated the cross-coupling robustness of the proposed dynamic shift register compared to conventional dynamic C²MOS in [1] and the single-clock FF in [6] in 45 nm CMOS technology. In order to address this issue, we added some wiring coupling capacitances to the FFs with the same transistor sizing compared to the previous experiment and then tracked the impacts of cross-coupling capacitances on perfor-

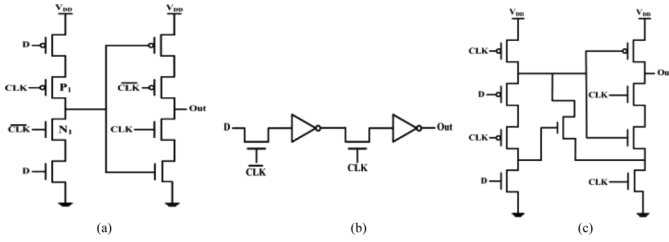


Fig. 8. Conventional dynamic positive edge-triggered DFFs. (a) C²MOS in [1]. (b) Classic dynamic FF in [1]. (c) Inverting single-clock FF in [6].

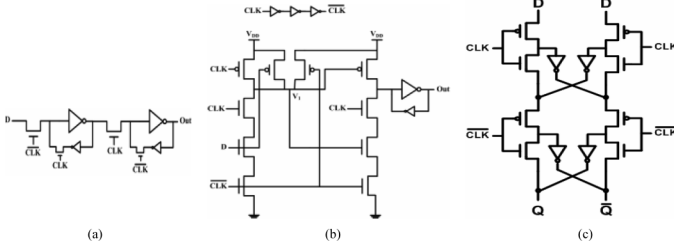


Fig. 9. Different implementations of static positive edge-triggered DFFs. (a) Classic FF in [1]. (b) HLFF in [7]. (c) GDI in [4].

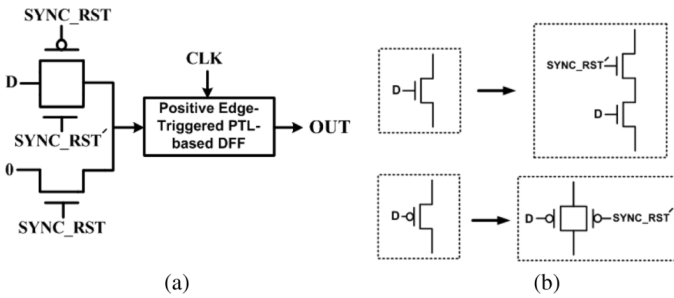


Fig. 10. Adding synchronous reset to the conventional FFs. (a) PTL-based DFFs. (b) PDN/PUN-based DFFs.

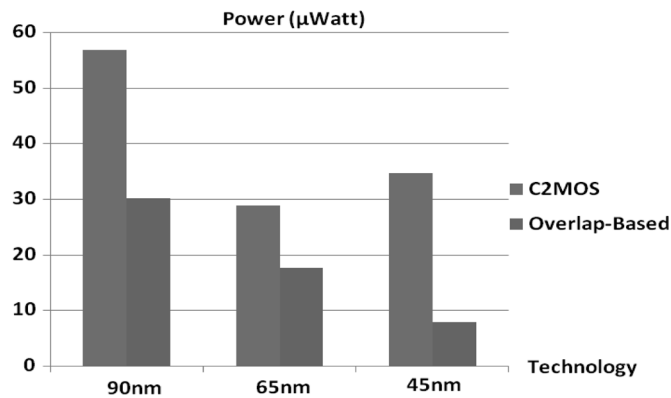


Fig. 11. Leakage efficiency of the proposed overlap-based dynamic 4-bit shift register in new DSM technologies.

mance and noise immunity of each circuit. The following wiring coupling capacitances have been added to the parasitic capacitances of the FFs.

- 1) $C_{W1} = 8$ fF between clk and the output node of FF.
- 2) $C_{W2} = 3$ fF between clk' and the output node.

Note that the second capacitance does not exist in the single-clock FF. Using the previous coupling capacitances impacts the

TABLE IV
IMPACT OF CROSS-COUPPLING ISSUES ON PERFORMANCE AND VOLTAGE SWING OF DYNAMIC DFFS IN 45 nm CMOS TECHNOLOGY WITH $V_{DD} = 0.6$ V

FF Structure	V_{OL} (mV)	V_{OH} (mV)	Power (μ W)	T_{Setup} (ps)	T_{Delay} (ps)
Overlap-Based	0	570	13.84	0	209.1
C ² MOS in [1]	55	520	43.95	115.5	215.6
Single Clock in [6]	0	430	85.48	191.5	219.8

performance and voltage swing of the circuits leading to the reduction of noise immunity. Table IV addresses the impacts of the previous coupling capacitances on performance and noise immunity of the circuits. In this table, V_{OH}/V_{OL} refers to the minimum/maximum static high/low voltage. The supply voltage is 600 mV. It can be seen that the overlap-based FF is more robust under cross-coupling issues as it provides a higher performance and a much better voltage swing. Note that in order to solve the problem of low-swing signals, we can use stronger transistors to provide more robust circuits. However, increasing the channel widths of the transistors not only impacts the power and area of the circuit, but also it may fail to keep full-swing nodes when the coupling capacitances increase. More information about the delay sensitivity analysis of FFs under cross-coupling issues can be found in [12] and [14].

IV. A NEW MEMORY CELL FOR A SCALABLE SORT COPROCESSOR BASED ON THE PROPOSED LOGIC CELLS

The superiority of the proposed logic cells can also be magnified in more complex architectures in which embedding logic functions become more efficient. We have evaluated the efficiency of the proposed architecture in implementing an odd–even stable sort coprocessor that is widely used in communication system researches, electronic triggering applications, and high-energy physics experiments [15]. Several register-transfer-level (RTL) stable sorters are proposed [15]–[19]. We have evaluated these sorters and it is seen that combining the two sorter architectures in [17] and [18] results in the most efficient implementation in terms of area, power, and performance. In the rest of this section, we first present the operation of this sorter, and then, its transistor-level implementation is evaluated using several logic styles including the proposed logic cells.

The selected sorter is able to perform an overlapping operation of insert, sort, and extract of n elements in n clock cycles on average. The architecture of this sorter when designed for sorting up to N key values with the bit length of m and N data records with the bit length of k consists of $N/2$ m -bit comparators and $(m+k) \times N$ memory cells with the structure in Fig. 12. These memory cells perform the storing process (the operation of FFs) as well as swapping and shifting elements (multiplexing logic). Note that the signal $SEL(i)$ and $SEL'(i)$ are the output results of the i th comparator ($i = 1, 2, \dots, N/2$). The other signal SEQ determines whether we are in the insertion or extraction phase. More information about these types of sorters can be found in [17]. The memory cell in Fig. 12 consists of three 2×1 multiplexers that form a *combinational block* and an edge-triggered DFF with synchronous preset as a *sequential*

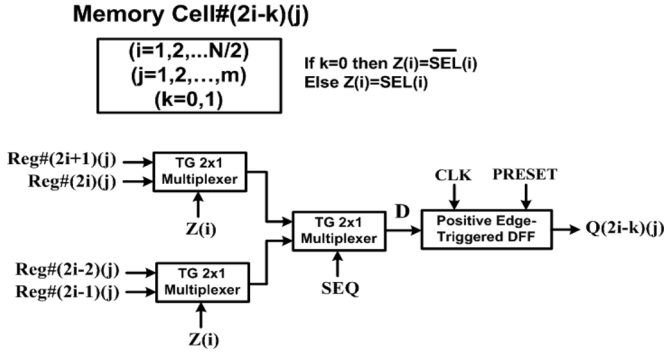


Fig. 12. Bit-level structure of a memory cell in the selected sorter.

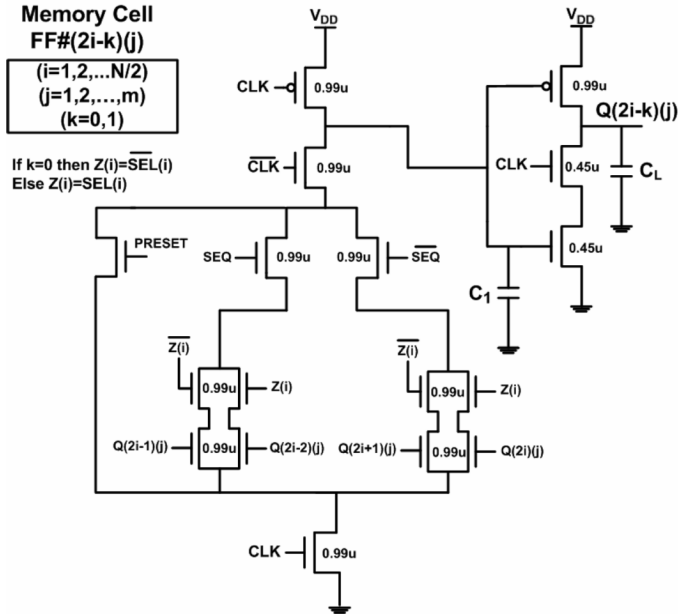


Fig. 13. Proposed overlap-based memory cell for the stable sorter.

block. The multiplexers used in a memory cell need complementary input signals and must be implemented with a static CMOS architecture. The TG logic is an efficient implementation for realizing the multiplexers in memory cells.

It is possible to integrate the combinational and sequential block of this memory cell into an efficient overlap-based logic cell with the same functionality. Fig. 13 illustrates the proposed memory cell. The channel widths of the transistors are addressed in this figure while all the channel lengths are 180 nm. This memory cell embeds the combinational block into a LUT with nMOS transistors. This LUT consists of five discharging paths for the capacitance C_1 . The control signals SEQ and $Z(i)$ are used to separate the discharging paths.

The proposed memory cell is compared with several state-of-the-art architectures. For the alternative memory cells, the TG logic is used to implement the combinational block of memory cells and the sequential block is implemented with several DFF architectures with synchronous preset to make the comparisons. Based on the schematic of the memory bank, a similar load capacitance of $C_L = 10$ fF is added to the parasitic capacitances of memory cells. The operating supply voltage is 1.8 V in 0.18 μm

TABLE V
POSTLAYOUT SIMULATION RESULTS OF DIFFERENT MEMORY CELL ARCHITECTURES IN A STABLE SORT COPROCESSOR

Memory Cell Structure	$C_{Z(i)}$ (fF)	Area (μm^2)	Power (μW)	TG Delay (ps)	Total Delay (ps)	PDP (fJ)
Overlap-Based	0.81	72.85	8.27	-	146.8	1.21
TG + HLFF in [7]	2.24	160.2	35.2	61.7	197.7	6.96
TG + C^2 MOS in [1]	2.24	122.9	24.9	61.7	352.5	8.78
TG + Classic	2.24	124.8	32.2	61.7	273.8	8.81
Dynamic FFin [1]						

CMOS technology. A clock frequency of 250 MHz with 150 ps rise/fall time is used. All the process/temperature corners are taken into consideration. The postlayout simulation results are summarized in Table V. The parameter " $C_{Z(i)}$ " refers to the input load capacitance on $Z(i)/Z'(i)$ produced by one memory cell in the i th sorting unit. Therefore, the load capacitance on the comparator block of each sorting unit is $2M \times C_{Z(i)}$, because $Z(i)/Z'(i)$ is connected to $2M$ memory cells. It can be seen that the proposed memory cell provides a much lower load capacitance as it does not need to route $Z(i)/Z'(i)$ through pMOS transistors. This approach makes the comparator block more efficient in terms of power and performance.

The TG delay column is the combinational delay of the TG multiplexers in the conventional memory cells in Fig. 12. The critical path delay of this block starts from $Z(i)$, which is produced by the comparator block and ends with the input signal of the FF. The TG delay is zero in the proposed memory cell. The total delay column in Table V is equal to the setup-time of the FF plus its propagation delay plus the TG delay in the case of conventional memory cells. On the other hand, in the proposed memory cell, total delay is equal to the propagation delay of the logic cell in Fig. 13. It can be seen that about 41% of area and 82% of PDP are saved by using the proposed architecture. Note that a significant amount of switching power is consumed within signal glitches of the vector multiplexers. However, in the case of the proposed memory cell, the operation of the embedded multiplexers is limited to the 1–1 overlap period of clock, and as a result, the signal glitches in other clock sequences will not lead to transient charge/discharge of the internal capacitances. This issue is not achievable in the conventional static multiplexer structures that widely suffer from signal glitches and switching power. Note that the simulation results of the proposed memory cell in Table V will be improved in more scaled-down technologies. The results in Table V can be used to estimate the efficiency of a sorting unit using different memory cell architectures. As an example, in a 256×32 sort processor ($N = 256$, $M = 32$) with 16-bit key values, the conventional memory cells provide a load capacitance of about $C_Z = 143$ fF on $Z(i)/Z'(i)$ for each comparator, while this value is about $C_Z = 52$ fF in the case of the proposed memory cell. Therefore, the performance and power consumption of the comparator block will be much improved if we use the proposed memory cells. However, regarding area consumption, if we use the dynamic 16-bit CMOS comparators in [20], the proposed memory cell will approximately save 34% of total area in each sorting unit.

The proposed logic cells are not only suitable for pipeline datapath structures, but also they can be used to implement state machines in the control units of processors. The more the complexity of logic functions in the state machine is, the more efficiently it can be implemented by the proposed cells.

V. CONCLUSION AND FUTURE WORK

An efficient architecture for implementing DFFs with embedded logic is proposed. This structure benefits from the overlap period of clocks. The design issues of setting an appropriate overlap period for the proposed logic cells are discussed in this paper. Several advantages of this approach have been explored and it has been shown that this architecture becomes more efficient when the complexity of its embedded logic function increases. Moreover, the static power dominance over switching power in new DSM technologies constitutes a more efficient operation for the proposed circuit in state-of-the-art CMOS technologies. The efficiency of the overlap-based logic cells has been evaluated in a 4-bit shift-register as well as an odd-even sort coprocessor. The proposed logic cells provide an efficient memory cell structure for the sorter. The proposed memory cell could improve up to 34% and 82% of the area consumption and power-delay-product of each scalable sorting unit in 0.18 μm CMOS technology. This improvement will be magnified in more scaled-down technologies. The overlap-based logic cell approach can efficiently be applied to the design of control units (state machines) as well as datapath structures in several deep pipeline applications. Moreover, we may be able to benefit from an overlap-based logic cell with a programmable embedded LUT in FPGA designs.

ACKNOWLEDGMENT

This work was initially developed in the Integrated Circuits Laboratory (ISL) at Ferdowsi University of Mashhad, while the final revisions were fulfilled in the Digital Systems Design Center at Sharif University of Technology, Tehran, Iran.

REFERENCES

- [1] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 2002.
- [2] I. S. Abu-Khater, A. Bellaouar, and M. I. Elmasry, "Circuit techniques for CMOS low-power high-performance multipliers," *IEEE J. Solid-State Circuits*, vol. 31, no. 10, pp. 1535–1546, Oct. 1996.
- [3] A. Morgenstein, A. Fish, and I. A. Wagner, "Gate-diffusion input (GDI)—A power efficient method for digital combinatorial circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 5, pp. 566–581, Oct. 2002.
- [4] A. Morgenshtein, A. Fish, and I. A. Wagner, "An efficient implementation of D-flip-flop using the GDI technique," in *Proc. Int. Symp. Circuits Syst. (ISCAS 2004)*, May, vol. 2, pp. II-673–II-676.
- [5] U. Ko and P. T. Balsara, "High-performance energy-efficient D-flip-flop circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 8, no. 1, pp. 94–98, Feb. 2000.
- [6] S. H. Yang, C. H. Lee, and K. R. Cho, "A CMOS dual-modulus prescaler based on a new charge sharing free D-flip-flop," in *Proc. IEEE ASIC/SOC Conf.*, Sep. 2001, pp. 276–280.
- [7] H. Partovi, R. Burd, U. Salim, F. Weber, L. DiGregorio, and D. Draper, "Flow-through latch and edge-triggered flip-flop hybrid elements," in *IEEE ISSCC Dig. Tech. Papers*, 1996, pp. 138–139.
- [8] O. Sarbishei and M. Maymandi-Nejad, "Power-delay efficient overlap-based charge-sharing free pseudo-dynamic D flip-flops," in *Proc. IEEE ISCAS*, 2007, pp. 637–640.
- [9] J. Yuan and C. Svensson, "New single-clock CMOS latches and flip flops with improved speed and power savings," *IEEE J. Solid-State Circuits*, vol. 32, no. 1, pp. 62–69, Jan. 1997.
- [10] V. Stojanovic and V. G. Oklobdzija, "Comparative analysis of master-slave 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] X. Cheng and R. Duane, "0.6 V D flip-flop utilising negative differential resistance device," *Electron. Lett.*, vol. 42, no. 7, pp. 388–390, Mar. 2006.
- [12] P. M. Wai, "High-performance output feedback flip-flop for ultra low-power applications," *Int. J. Electr. Eng.*, vol. 13, no. 1, pp. 29–39, Feb. 2006.
- [13] V. Tirumalashetty and H. Mahmoodi, "Clock gating and negative edge triggering for energy recovery clock," in *Proc. IEEE ISCAS*, 2007, pp. 1141–1144.
- [14] W. L. Goh, "Delay sensitivity study on process, supply voltage and temperature variations of single edge-triggered flip-flops," in *Proc. Symp. Low-Power High-Speed Chips*, Apr. 2007, pp. 151–162.
- [15] I. E. Mumolo, G. Capello, and M. Nolic, "VHDL design of a scalable VLSI sorting device based on pipelined computation," *J. Comput. Inf. Technol. (CIT 2004)*, vol. 12, no. 1, pp. 1–14, 2004.
- [16] A. A. Colavita, A. Cicutin, F. Fratnik, and G. Capello, "SORTCHIP: A VLSI implementation of a hardware algorithm for continuous data sorting," *IEEE J. Solid-State Circuits*, vol. 38, no. 36, pp. 1076–1079, Jun. 2003.
- [17] S. W. Moore and B. T. Graham, "Tagged up/down sorter—A hardware priority queue," *Comput. J.*, vol. 38, pp. 695–703, Sep. 1995.
- [18] N. Takagi and C. K. Wong, "A hardware sort-merge system," *IBM J.*, vol. 29, no. 1, pp. 49–67, Jan. 1985.
- [19] I. Hatirnaz, F. K. Gurkaynak, and Y. Leblebici, "A compact modular architecture for high-speed binary sorting," in *Proc. IEEE ISCAS*, 2000, vol. 4, pp. 685–688.
- [20] C.-H. Huang and J.-S. Wang, "High-performance and power-efficient CMOS comparators," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 254–262, Feb. 2003.



Omid Sarbishei received the B.Sc. degree in electrical engineering from Ferdowsi University of Mashhad, Mashhad, Iran, in 2007. He is currently working toward the M.Sc. degree in electrical engineering at Sharif University of Technology, Tehran, Iran.

He has been a Teaching and Research Assistant. His current research interests include those dedicated to formal verification, low-power and high-performance digital very large-scale integration (VLSI) design, and high-level synthesis. Other

research interests include hardware-software codesign and physical design methodologies.



Mohammad Maymandi-Nejad (M'02) received the B.Sc. degree from Ferdowsi University of Mashhad, Mashhad, Iran, in 1990 and the M.Sc. degree from Khajeh Nassir Tossi University of Technology, Tehran, Iran, in 1993, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2005, all in electronics engineering.

From 1994 to 2001, he was a Lecturer in the Department of Electrical Engineering, Ferdowsi University of Mashhad, where he was engaged in teaching and research and also conducted several industrial projects in the field of automation and computer interfacing. He is currently an Assistant Professor in the same department. His current research interests include low-voltage, low-power analog ICs and their applications in biomedical circuits and systems.

Dr. Maymandi-Nejad received the Strategic Microelectronics Council of Information Technology Academia Collaboration (ITAC) Industrial Collaboration Award in 2005 for his work on a wireless bioimplantable device for monitoring blood pressure of transgenic mice.