

# Dynamic Flip-Flop Conversion in Digital Circuits with Hybrid Time Borrowing and Critical Feedback Path Applications

K.Nanthakumar<sup>1</sup>, S.Thejaswini<sup>2</sup>

AP/ECE, M.P.Nachimuthu M.Jaganathan Engineering College, Erode, Tamil Nadu<sup>1</sup>

M.E.VLSI Design, M.P.Nachimuthu M.Jaganathan Engineering College, Erode, Tamil Nadu<sup>2</sup>

**Abstract:** Dynamic Flip-Flop Conversion (DFFC) is a time borrowing method for improving the performance of digital circuits. Existing types of DFFC [11], [12] suffer from successive critical and critical feedback paths that are frequently seen in digital circuits. Moreover, they are unable to increase the performance of the designs with short sequential depth. In this paper, we introduce a hybrid technique which utilizes DFFC together with a dynamic clock stretching mechanism. Our technique is able to mitigate the problems of successive critical and critical feedback path structures even in the presence of process variations. The results show that our hybrid technique is able to increase the performance of some ITC'99 and ISCAS'89 benchmarks by 24.4% on average while DFFC Type C increases the performance only by 8.4% on average. Furthermore, we have shown that our hybrid technique is able to tolerate process variations, 18% power supply variation, and 100 °C temperature variations, 27.3%, 16.4%, and 13.3% better than the state-of-the-art methods on average, respectively.

**Keywords:** Dynamic flip-flop conversion, hold time violation (HTV), process variation, setup time violation (STV), time borrowing, transparency window.

## I. INTRODUCTION

The demand for high performance designs which are not susceptible to variations has been significantly increasing over the past few years. Traditionally, in worst case design methodology the maximum allowable frequency (MAF), in which a circuit works correctly, is computed based on the delay of the longest paths (critical paths) in the circuit. Such critical paths, which are rarely activated, make the designer keep the frequency low while losing the performance. On the other hand, the performance degradation is even worse when process variation is taken into account. Such a variation makes some paths work faster or more slowly [1] and forces the designer to include timing margins in the design which leads to the performance degradation. A simple idea to achieve better performance is to increase the operational frequency of the circuit beyond the MAF. Obviously, in this case timing violations occur because critical paths would not be able to finish their job in a single clock period.

In order to address these timing violation issues three main approaches including error detection and correction, retiming and time borrowing have been proposed. In error detection and correction technique a timing error is detected at each flip-flop using a transition detector circuit and then a recovery mechanism corrects the data after some clock cycles [2]. Such techniques, however, do not prevent timing errors. Retiming is a technique to prevent timing violations by moving flip-flops backwards or forwards in the paths. Nevertheless, this technique is not efficient because it may increase the number of flip-flops [3]. A well-known and commonly used technique to avoid timing violation is time borrowing technique [4], [5], [8]. The main idea is the fact that a critical stage in which the setup time is violated can borrow some time from the successive stage if it has enough timing slack. In [5] and [6] the timing yield has been improved by replacing some flip-flops with latches. Since latches are level sensitive, the late data from critical paths can cross the latches when they are transparent. However, latch based designs suffer from large transparency windows and large delay elements that should be inserted in short paths to avoid hold time violation (HTV). In [7] and [19] a latch is placed at the destination flip-flop to perform time borrowing. A transition detector block is used to detect time borrowing in critical latch and issues a warning signal. This warning signal is used to raise the supply voltage of the next stage to speed up the next stage and prevent further timing violations. In [8] and [9], soft-edge flip-flops (SEFF) have been used which have a small window of transparency instead of a hard edge, allowing limited cycle stealing on critical paths, and thus compensating for delay variations.

Sequential circuits often include multiple pipeline stages. In pipeline stages where successive critical paths exist, time borrowing from the following critical stage is not possible and borrowed time cannot be compensated through successive stages. In [4] a special soft-edge flip-flop together with a clock shifter help prevent timing violations. When a timing error happens, this flip-flop allows time borrowing during a time-borrowing window (TBW) and generates a flag that shows time borrowing has occurred. Then the clock shifter elastically stretches the clock period in the next clock cycle to pay back for the borrowed time and prevents the next stage in pipeline to be-come critical. In [10] the pulsed-latch flip-flop with additional circuits is presented for time borrowing.

The additional circuit detects signal transition on input of flip-flop during the time-borrowing window and issues a time borrowing (TB) flag.

## II. DFFC MICROARCHITECTURES AND THEIR CHALLENGES

In order to make this work self-contained, we briefly explain existing dynamic flip-flop conversion (DFFC) microarchitectures. DFFC is a time borrowing method which takes the advantage of transparency window of latches in a flip-flop based design. Whenever setup time violation (STV) is predicted, the critical flip-flop is dynamically converted into a transparent latch and the late data is stored in the destination flip-flop. So far three different DFFC methods have been proposed [1], [11], [12]. Fig. 1(a) shows the DFFC microarchitecture in which the destination flip-flop is a master-slave flip-flop as shown in Fig. 1(b). The timing violation is predicted based on the fact that if half of a path cannot finalize its value in half of the clock period, then it cannot finalize the value at the destination flip-flop's input by the end of the clock period and therefore setup time violation is expected to happen. In order to predict timing error, Timing Violation Predictor (TVP) block shown in Fig. 1(c) detects any transition on the middle point (*Mid*) in the second half of the clock period.

Using information provided by timing analysis tool (e.g., Prime Time) the critical paths of the circuit and the delay information of each element in these paths is determined. We make use of such information to choose the midpoint based on the method explained in [1]. As mentioned in [1], Monte Carlo simulation is performed to find the best choice for the midpoint. According to the achieved results, if it is possible to cut the circuit into half it is better to choose the midpoint exactly at the middle of the critical path. Otherwise, it is chosen on the node after half delay of the critical path. If the value of *Mid* changes, the *Err* signal goes high and it stays high until the next falling edge of the clock. In the following subsections we explain existing DFFC methods in more details.

### A. DFFC Types A and B

In DFFC Type A, once the error is predicted, master latch clock is inverted using an XOR gate. So the flip-flop is turned into two latches that act like one transparent latch when the clock is high and late data from critical path can cross the flip-flop during the transparency window. In this type of DFFC the transparency window is large and data from short paths may overwrite the correct value so HTV may occur. In DFFC Types B and C a Data Arrival Detector (DAD) block is added to the circuit which detects the transition on destination flip-flop input and consequently closes the transparency window. If the data from the critical-path arrives before short-path data, the

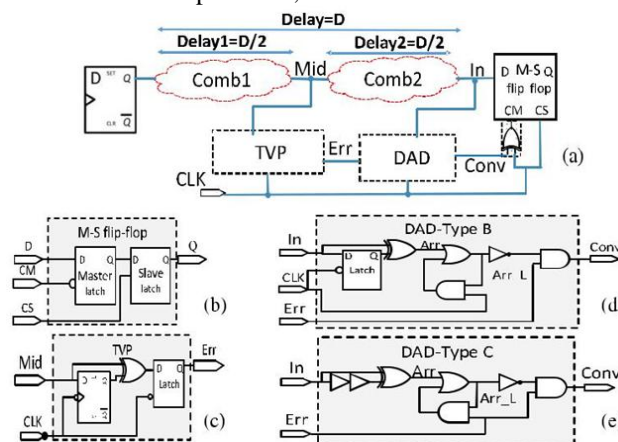


Fig. 1.

- DFFC micro architecture,
- master-slave flip-flop,
- TVP block,
- DAD block of Type B [11], and (e) DAD block of Type C. Note that CM stands for Clock of Master latch and CS stands for Clock of Slave latch.

In [1] and [11], it is shown that DFFC Types A and B can result in a higher yield and better performance compared to other methods such as SEFF and dynamic clock stretching [13]. However, they both suffer from false error prediction of TVP block [11]. That is because the *Mid* is selected a little after the half delay in asymmetrical critical paths [1]. This may result in a wrong prediction of error while the path may not violate any timing constraint and data arrives before the rising edge of the clock. In this case, the transparency window falsely opens until either the data from short path overwrites the value in the destination flip-flop or at the next falling edge of the clock, *Err* signal goes down and the transparency window is closed. DFFC Type C [12] has solved this problem which will be explained in more detail in the following subsection.

### B. DFFC Type C

Fig. 1(e) shows DAD block of DFFC Type C which is an improved version of that of DFFC Type B to solve the false error problem without any additional hardware [12].

The basic idea is to detect data arrival on destination flip-flop's input before the rising edge of the clock and prevent the transparency window at destination flip-flop from opening. In normal operation mode *Arr*, *Arr\_L* and *Err* signals are low.

When error is predicted by TVP block (i.e., *Err* = 1) one of the following two cases happen:

### C. Gate Level Issues

In [12] it is mentioned that by directly inserting DFFC into critical paths of a given circuit, we may not be able to increase the performance due to challenging structures. We have investigated ITC'99 benchmarks [14] and found five special challenging structures as shown in Fig. 2. These structures are categorized as follows:

- *Short Feedback Path (SFP)*: Fig. 2(a) shows a short feedback path in which a critical path has a short feedback at its destination flip-flop. This short feedback can cause HTV in methods like latch based designs and SEFF or DFFC

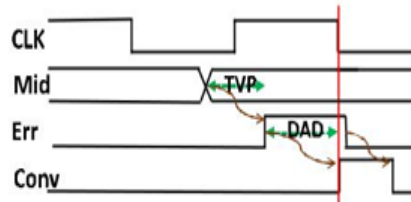


Fig. 3. Late Conversion (LC) problem waveforms

- *Critical Feedback Path (CFP)*: Fig. 2(b) shows a critical feedback structure. It is a feedback path which is also a critical path. Since in this structure the next path is also a critical path and it does not have enough time slack, time borrowing methods are not applicable.
- *Successive Critical Path (SCP)*: Fig. 2(c) shows a successive critical path structure. It is a critical path followed by another critical path in a pipelined design. Time borrowing in former critical path is not possible because the following critical path does not have enough time slack. However, time borrowing in the latter one is still possible.
- *Critical Fan-out Path (CFOP)*: Fig. 2(d) shows critical fan-out structure. In this structure a fan-out exists at the end of a critical path so that all destination flip-flops would be critical. DFFC can be very efficient when it comes to reconvergence fan-outs. In these structures only one TVP blocks is used and the *Err* signal generated by TVP can be shared between DAD blocks. So, area and power overheads are significantly reduced.
- *Critical Fan-in Path (CFIP)*: Fig. 2(e) shows critical fan-in structure. In such structures all branches that converge are critical and the midpoint (i.e., *Mid* signal) of each branch is before converging node. We have two options to implement DFFC method for these structures: 1) using one TVP block for each path which increases the power and area overhead, and 2) choosing converging node as the middle point which may cause false error prediction of TVP block. In these cases, DFFC Type C is very efficient because it is able to remove false errors.

In [12] it is mentioned that by employing the second choice, Late Conversion (LC) problem may occur. LC happens when the delay of the midpoint (*Mid*) plus the delays of DAD and TVP blocks become larger than 1.5 times of the clock period. This causes *Conv* signal ready to go high after the falling edge of the next clock cycle while at the falling edge of the clock *Err* signal goes down which causes *Conv* signal to fall immediately. In order to clarify it, let us consider the case where the critical path is asymmetrical and there is a large sub-circuit, e.g., full adder, in the middle of the path [in *Comb1* of Fig.

1(a)]. We have to choose *Mid* signal very close to the destination flip-flop and therefore the sub-circuit in *Comb1* of Fig. 1(a) makes  $Delay_1 > Delay_2$  and causes LC problem. In Fig. 3, by changing the *Mid* signal, *Err* signal goes high after the delay of TVP block and *Conv* signal goes high at the falling edge of the clock. The TVP block causes the *Err* signal to go down at the falling edge of the clock which makes *Conv* signal fall immediately. Note that, another reason for LC occurrence is the short sequential depth of the circuit.

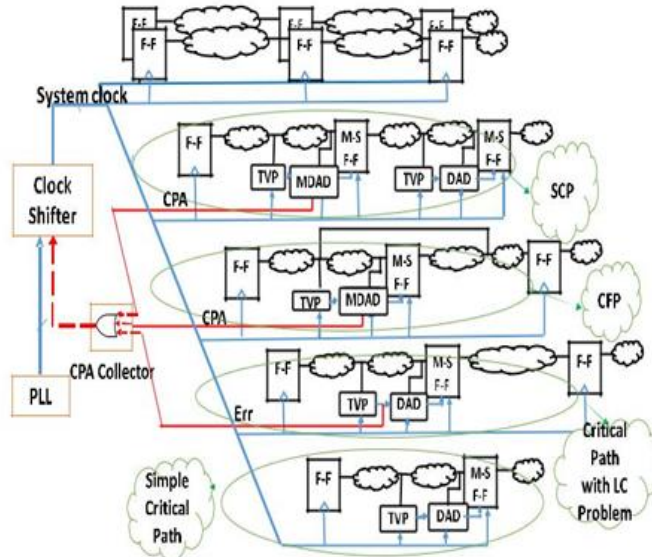


Fig. 4. Demonstration of applying hybrid technique on a complete digital circuit.

### III. PROPOSED HYBRID TIME BORROWING TECHNIQUE

As mentioned before, DFFC Type C [12] is not able to resolve the timing errors in CFP and SCP. In addition, due to LC problem it is unable to prevent timing errors in the circuits with short sequential depths. In order to alleviate these problems we have proposed a hybrid technique which first tries to prevent the timing error by borrowing the time from the following stages. If it is not possible due to CFP and SCP structures, it dynamically stretches the clock.

The way of applying our proposed technique to a complete digital circuit (including four critical path structures: a CFP, a SCP, a critical path with LC problem and a simple critical path) is depicted in Fig. 4 where all critical flip-flops are shown as M-S flip-flop. The PLL creates two clock signals that are used in clock shifter to stretch the clock for half of a clock cycle. Simple critical paths as well as critical paths with LC problem are equipped with DAD and TVP blocks of Type C, while CFP and SCP are equipped with TVP and Modified Data Arrival Detector (MDAD) blocks (will be explained in Section III-A). Note that a simple critical path is a path that does not have any of the challenging structures and LC problem. As shown in Fig. 4, in our hybrid micro architecture all of CPA signals are collected into a block called “CPA Collector” (OR gates) which is fed to the clock shifter. On the rising edge of the clock, if one of CPA signals has been issued, the clock is stretched for half of a clock cycle giving more time to the critical paths to finish their job.

#### A. Modified DAD (MDAD) Block

Fig. 5 shows MDAD block which is similar to that of DFFC Type C with only a small additional circuit to dynamically generate a Challenging Path Activation (CPA) signal. This signal shows that time borrowing alone is not sufficient to solve the timing problem and the clock of the system needs to be stretched out as well. Whenever *Conv* signal is issued prior to the rising edge of CLK, CPA signal will be issued. The CLK will be stretched out for half of a clock cycle. Then with the rising edge of CLK, CPA signal goes down and the system goes back to the normal.

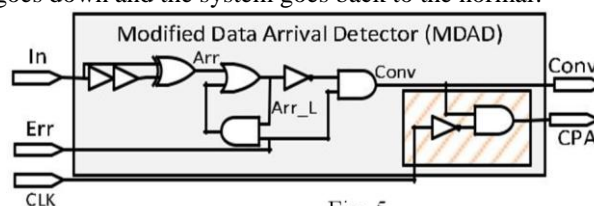


Fig. 5.

Modified Data Arrival Detector (MDAD) block which generates CPA signal to show that time borrowing alone is not sufficient to solve the timing problem.

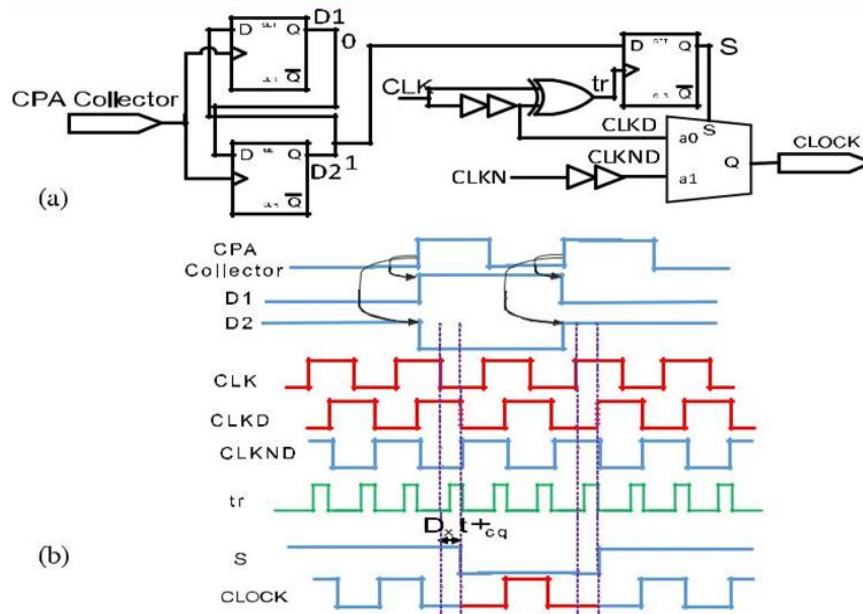


Fig. 6.

(a) Clock Shifter, and (b) its internal signals. CLK and CLKN are generated by PLL and have 180 degree phase difference. Moreover, in the case of LC problem, *Conv* and CPA signals in MDAD block are not issued before the rising edge of the clock, and therefore *Err* signal generated by TVP block is used as a CPA signal.

### B. Clock Shifter Unit

The proposed clock shifter is depicted in Fig. 6(a). The main reason behind using a clock shifter to stretch out the “System clock” for half of a period is that in the case of LC problem, we need more than half of a clock cycle to resolve the timing issue. The main idea is that if CLOCK is equal to one of the two clock signals from PLL block (i.e., CLK and CLKN that have half of the clock cycle delay compared to each other) and after the rising edge of CPA Collector, we switch CLOCK from that signal to the other one, we have delayed the rising edge of CLOCK for half of a clock cycle. To do so, two flip-flops (D1 and D2) are used to detect changes on CPA signal. At the rising edge of CPA Collector their outputs are switched. The output of one of the flip-flops is used to choose between two delayed clock signals (CLKD and CLKND) from PLL using a flip-flop and a XOR gate and a multiplexer. Suppose D1 and D2 flip-flops are initialized to 0 and 1, respectively, as shown in Fig. 6(b). If the system clock (CLOCK) is equal to CLKND signal, after transition on CPA Collector, D2 changes from 1 to 0 and the CLOCK changes from CLKND to CLKD. Again, after transition on CPA Collector, D2 changes from 0 to 1 and the CLOCK changes from CLKD to CLKND. Therefore, with each transition on CPA Collector, the period of CLOCK signal is equal to 1.5 times of that of CLK signal for one clock cycle.

Note that no hold time violation occurs in Clock Shifter circuit. That is because in Nand Gate open cell library 45 nm [17],  $t_{cq}$  delay of D1 and D2 flip flops is equal to 0.06 ns which is higher than the hold time of a latch that is 0.04 ns. Also, the S signal is generated after the edge of the clock with a delay that equals to the addition of XOR gate delay ( $D_x$ ) and  $t_{cq}$  of the S flip-flop. So, both inputs of the mux should be asserted with a delay that equals to that of the S signal to avoid glitches. To do so, we have inserted  $(D_x + t_{cq})/D_{buffer}$  number of buffers (which is 6 for NanGate open cell library 45 nm) between the inputs of the mux and the CLK signals from PLL.

### C. Upper Bound Limit for Clock Stretching

Clock stretching in its nature can cause performance deterioration. That is because by increasing the frequency of a circuit using our hybrid technique, the number of clock stretching may increase due to a larger number of timing violations. Therefore, in order not to lose any performance we would better to put an upper bound limit to the maximum number of clock stretching events. Let us consider a task that takes C cycles to be finished. If the circuit works at initial frequency of  $f_i$

with  $n_i$  number of clock stretching, then as mentioned before each clock stretching lasts for 1.5 clock cycles and the duration of the task with  $f_i$  frequency ( $D_i$ ) is calculated as follows:

$$D_i = (C - n_i) \times \frac{1}{f_i} + n_i * \frac{1.5}{f_i} \quad (1)$$

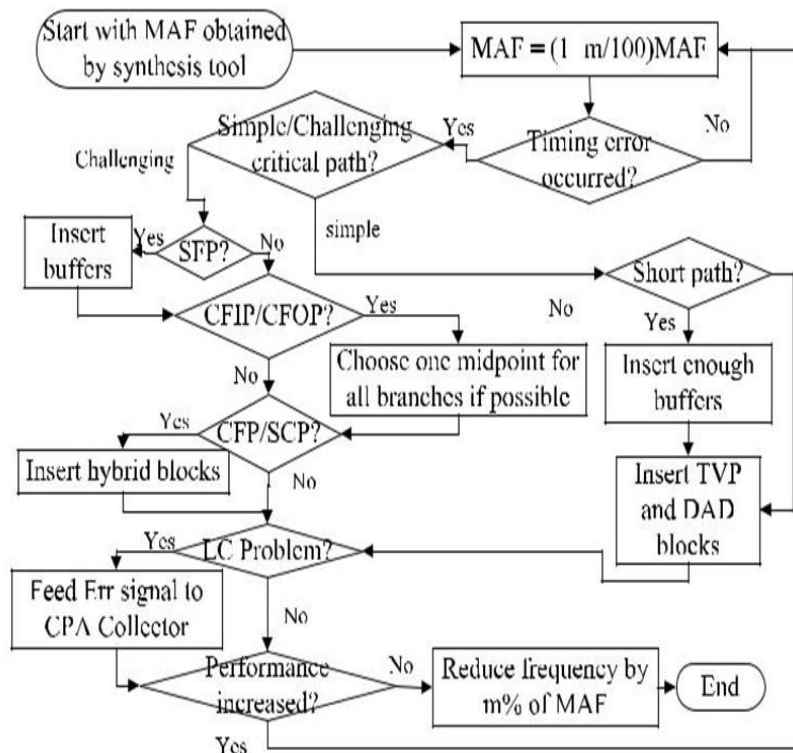
When the circuit operates at higher frequency  $f_{i+1}$  then  $n_{i+1}$  number of clock stretching ( $n_{i+1} \geq n_i$ ) may happen. In order to be sure that the performance is not lost, then the duration of the task with  $C$  cycles with frequency of  $f_i$  and  $n_i$  number of clock stretching must be larger than duration of finishing the job with higher frequency  $f_{i+1}$  and  $n_{i+1}$  number of clock stretching

$$D_i > D_{i+1} \Rightarrow (C - n_i) \times \frac{1}{f_i} + n_i * \frac{1.5}{f_i} > (C - n_{i+1}) \times \frac{1}{f_{i+1}} + n_{i+1} * \frac{1.5}{f_{i+1}} \quad (2)$$

Therefore, during  $m\%$  increase in frequency (i.e.,  $f_{i+1} = (1 + m/100)f_i$ ) an upper bound limit for  $n_{i+1}$  number of clock stretching in high performance designs is calculated as follows:

$$n_{i+1} < \frac{C * m}{50} + n_i + \frac{m * n_i}{100} \quad (3)$$

For example, a 100 cycle task with operating frequency of 1 GHz that needs 2 clock stretching takes 101 ns to accomplish. An increase of 5% in the frequency leads to more clock stretching. According to (3) the upper bound limit for clock stretching equals 12.1. If 25 number of clock stretching in 100 cycles occur, the task will take 107.14 ns to finish and obviously the expected performance improvement is lost.



7. Proposed flow of increasing the performance by adding our hybrid technique to the gate level circuits.

#### IV. EXPERIMENTAL SETUP AND RESULTS

In order to evaluate the effectiveness of our proposed hybrid technique, in the following subsections, first the performance improvement on complete digital circuits is investigated. Next, in order to show how much our technique can tolerate process variations, Spice net lists for problematic critical paths of various benchmarks are extracted and augmented with our hybrid blocks.

##### A. Performance Results

To evaluate the effectiveness of a time-borrowing technique, it is important to make sure that it is able to improve the performance of digital circuits when considering all paths inside them. Although in [12] DFFC Type C was implemented on complete digital circuits with real pipeline stages and feedback paths, in [1] and [11], DFFC was evaluated on single critical paths. In this paper, however, we have implemented our technique on five benchmark circuits from ITC'99 (i.e., b03, b05, b11, b12, b15) and three circuits from ISCAS'89 [15] (i.e., s5378, s38417, s38584) in order to show how much performance improvement is achievable. For doing so, the RTL code of the benchmark circuits are synthesized using Design Compiler [16]. The gate level netlist, critical path information reported by synthesis tool, and the MAF of the circuit are used by Tetramax [16] to generate test patterns which are then used by gate level simulators to evaluate the hybrid technique as well as other time borrowing methods.

To do so, Fig. 7 shows our proposed flow of increasing the performance using our hybrid technique. At the beginning of the flow the operational frequency of the circuit is increased by  $m\%$  of MAF. Note that, in our work, we have chosen  $m = 0.8$  which is equal to the longest frequency increasing that adds maximum of one critical path to the list of activated critical paths with timing violation. The frequency is increased by the steps of  $m\%$  until a timing violation occurs. Next, the critical path with timing violation is categorized as a simple or challenging critical path. If the path is a simple critical path, then enough buffers are added to short paths and TVP and DAD blocks are inserted. Otherwise, if the challenging path includes SFP structure, as in [12] enough buffers are inserted in the feedback path to avoid HTV. Moreover, in the case of CFIP structure, for all the branches one midpoint is chosen. Also, if the converging point in CFOP structure is placed after the half delay of critical paths, one midpoint is chosen for all branches which reduces area and power overhead. If not, each fan-out branch needs a TVP block. If the challenging path forms a CFP or SCP structure, hybrid blocks (i.e., TVP, MDAD, "CPA Collector," and "Clock Shifter") are added to the circuit. Next, if there is still a timing error due to LC problem, *Err* signal of TVP block is fed to "CPA Collector" block to be used for clock stretching. Finally, the circuit is simulated again to evaluate the performance considering both the frequency and the number of clock stretching. The number of clock stretching during  $m\%$  frequency increase must not exceed the upper bound limit of (3). It is worth noting that (3) states that when  $C = 1000$ ,  $n_i = 0$  and  $m = 0.8$ , the number of added clock stretching must be less than 16 in 1000 cycles on average to be sure that the performance is increasable without side-effect of clock stretching. In this case, the frequency is again increased and all the steps described above are repeated. Otherwise the clock frequency is reduced to the previously obtained value and the flow finishes its work.

For comparison purposes, we have implemented DFFC Type A [1], DFFC Type B [11], DFFC Type C [12], SEFF [8], SEFF with elastic clock stretching (ECS) [4], and Dynamic Clock stretching (DCS) on our benchmarks. Note that SEFFs are inserted at the destination of critical paths and master latches in SEFFs are fed by a delayed system clock which generates the transparency window. For implementing SEFF with ECS technique, we have inserted the modified time borrowing flip-flop of [4] at the destination of critical paths. Furthermore, clock stretching blocks of [4] are utilized to apply the necessary clock signals to the benchmark circuits. To apply DCS to the benchmark circuits the TVP blocks was inserted in the middle of each critical path and in the case of error prediction, delayed clock signal was fed to the critical flip-flop. Table I shows the MAF of eight benchmark circuits using DFFC Type A, DFFC Types B and C, and our hybrid technique. To compare our hybrid technique with other time borrowing techniques, Table II shows the MAF of eight benchmark circuits using SEFF, SEFF + ECS, DCS, and our hybrid technique.

The first three columns of these tables contain the benchmark circuit information including the total number of gates (#gates) and flip-flops (#F F). The Next column (*MAF No TB*) gives the MAF of each benchmark circuit when no time borrowing technique is used (reported by the synthesis tool). The MAF of the circuits (minor column MAF) and their improvements compared to *MAF No TB* (minor column *Impr*) using DFFC Type A [1], DFFC Type B [11], DFFC Type C [12] and our hybrid technique are tabulated in DFFC – A, DFFC – B, DFFC – C, and *Our Hybrid* columns of Table I, respectively. It is worth mentioning that the provided time borrowing technique of [12] is enough for the circuit for frequencies up to MAF of DFFC Type C (MAF subcolumn of DFFC – C column in Table I) while clock stretching should be used for higher frequencies to avoid timing violations. For example, in the case of b03 circuit no time borrowing is

needed up to the frequency of 1.66 GHz (*MAF No TB* column in Table I). From 1.66 GHz to 1.73 GHz (*MAF* subcolumn of *DFFC – C* column in Table I) the time borrowing can solve timing violations. For frequencies from 1.73 GHz to 2.08 GHz (*MAF* subcolumn of *Our Hybrid* column in Table I) clock stretching must be used whenever SCP and CFP structures are activated.

**TABLE I**  
Performance Comparison (MAF) between DFFC TYPE A, DFFC TYPE B, DFFC TYPE C and our Hybrid Technique (All reported MAF IN GHZ)

Benchmarks	#gates	# FF	MAF No TB	DFFC-A[1]		DFFC-B [11]		DFFC-C [12]		Our Hybrid	
				MAF	Impr	MAF	Impr	MAF	Impr	MAF	Impr
b03	150	30	1.66	1.66	0%	1.73	3.3%	1.73	3.3%	2.08	25.3%
b05	608	34	1.28	1.35	8.8%	1.28	0%	1.39	7.7%	1.6	25%
b11	366	31	1.19	1.25	5%	1.31	10.8%	1.31	10.8%	1.56	31.3%
b12	1000	121	1.28	1.31	2.3%	1.32	3.1%	1.39	13.8%	1.58	29.6%
b15	8922	449	1.06	1.21	14.1%	1.11	4.7%	1.11	4.7%	1.38	30%
s5378	10098	179	1.09	1.09	0%	1.09	0%	1.09	0%	1.12	12.6%
s38417	22179	1636	1.47	1.61	9.5%	1.61	9.5%	1.61	9.5%	2.0	36%
s38584	19253	1426	1.79	1.79	0%	2.08	16.2%	2.08	16.2%	2.13	19%
Average			1.35	1.4	3.7%	1.44	6.7%	1.46	8.4%	1.68	24.4%

**TABLE II**  
Performance Comparison (MAF) between SEFF, SEFF + ECS, DCS, and our Hybrid Technique (ALL REPORTED MAF IN GHZ)

Benchmarks	#gates	# FF	MAF No TB	SEFF [8]					SEFF+ECS[4]		DCS[13]		Our Hybrid	
				MAF1	MAF2	MAF3	Ave MAF	Ave Impr	MAF	Impr	MAF	Impr	MAF	Impr
b03	150	30	1.66	1.66	1.66	1.66	1.66	0%	1.7	2.4%	1.66	0%	2.08	25.3%
b05	608	34	1.28	1.4	1.38	1.38	1.39	8.5%	1.4	9.3%	1.43	11.7%	1.6	25%
b11	366	31	1.19	1.25	1.25	1.31	1.27	6.7%	1.46	22.7%	1.19	0%	1.56	31.3%
b12	1000	121	1.28	1.28	1.41	1.28	1.32	3.1%	1.28	0%	1.35	5.4%	1.58	29.6%
b15	8922	449	1.06	1.25	1.32	1.06	1.21	14%	1.06	0%	1.11	4.7%	1.38	30%
s5378	10098	179	1.09	1.09	1.09	1.09	1.09	0%	1.09	0%	1.09	0%	1.12	12.6%
s38417	22179	1636	1.47	1.61	1.61	1.56	1.59	8.4	1.85	25.8%	1.61	9.5%	2.0	36%
s38584	19253	1426	1.79	2.08	1.85	1.85	1.92	7.3%	1.85	3.4%	1.86	3.4%	2.13	19%
Average			1.35	1.33	1.35	1.34	1.34	3.7%	1.39	6.9%	1.33	2.3%	1.68	24.4

In Table II the SEFF major column presents the MAF of the circuits using SEFF technique with three degrees of softness in terms of clock period (T), minor columns MAF1 (0.125 T), MAF2 (0.25 T) and MAF3 (0.375 T), while *Ave MAF* and *Ave Impr* minor columns shows the average MAF and its improvement of these three implementations compared to that in *MAF No TB* column. The MAF of the circuits (minor column MAF) and their improvements compared to *MAF No TB* (minor column *Impr*) using SEFF with ECS [4], DCS [13], and our hybrid technique are presented in SEFF – ECS, DCS and *Our Hybrid* columns of Table II, respectively. It can be seen that in terms of the performance, our new hybrid technique improves the performance of the circuits by 24.4% on average while using our previous DFFC methods including DFFC Type A, DFFC Type B, and DFFC Type C 3.7%, 6.7%, and 8.4% performance improvement is achieved, respectively. Moreover, using other time borrowing techniques such as SEFF, SEFF with ECS, and DCS, only 3.7%, 6.9%, and 2.3% improvement was attained. It should be noted that, in SEFF technique, HTV may happen because a fixed transparency window is created and therefore this technique is not good enough. Moreover, we need to run various test-vectors for different degrees of softness for a given circuit in order to find the best transparency window size. In SEFF with ECS technique, a special SEFF with 0.25 of clock period (0.25 T) softness is used. It stretches the clock of the system for 0.25 T and opens the transparency window even if the timing violation is removed by stretching the clock period. This fixed transparency window may result in HTV.

Table III compares the number of DAD and MDAD blocks when the circuits are equipped with DFFC Type C and hybrid techniques. The *#DAD C*, *#MDAD Hybrid*, and *#DAD Hybrid* columns report the total number of DAD blocks using DFFC Type C (operating at DFFC – C MAF reported in Table I), the number of MDAD and DAD blocks using our hybrid technique (operating at *Our Hybrid MAF* reported in Table I), respectively. Note that the number of critical paths equipped with DFFC Type C is equal to the number of DAD blocks (*#DAD C*). Also, in our hybrid technique MDAD blocks are only used in SCP and CFP structures and the *#MDAD Hybrid* represents the number of SCP and CFP structures in each benchmark circuit. Additionally, DAD blocks are inserted in other critical path structures and *#DAD Hybrid* shows their



**TABLE III**  
Hardware and Cycle Overhead of DFFC TYPE C and hybrid technique (NA: NOT APPLICABLE)

Bench mark	#DAD C	#MDAD Hybrid	#DAD Hybrid	#Stretch Cycles	Extra C		Extra H	
					FF	G	FF	G
b03	3	1	0	25	6	24	5	28
b05	6	2	0	21.5	12	42	7	38
b11	3	1	2	52.5	6	24	9	44
b12	10	1	7	7.35	20	80	19	84
b15	2	1	1	24.7	4	16	7	36
s5378	NA	2	0	11.5	NA	NA	7	38
s38417	11	2	1	46.4	22	88	9	46
s38584	13	2	5	445	26	104	17	78
Ave.	6.8	1.5	2	79	14	49	10	50

number. It can be seen that, using hybrid technique in comparison with DFFC Type C, the average number of critical paths needed time borrowing decreases by 48% on average. That is because using hybrid method the clock stretching technique prevents subsequent noncritical paths to become critical. In order to show that the clock stretching technique does not cause the performance loss, we have also calculated the number of clock stretching in 1000 cycles when the circuits are operating at MAF of our hybrid technique reported in Table I. The *#Stretch Cycles* column shows that in seven benchmark circuits, 87 clock stretching on average has occurred in 1000 cycles for 24.4% increase in the clock frequency. Based on (3), the upper bound limit of clock stretching for an increase of 24.4% in clock frequency is 488 which is much higher than 87 number of clock stretching occurred in our benchmark circuits. The *Extra C*, *Extra H* major columns in this table report the overall extra number of flip-flops (subcolumn *FF*) and gates (subcolumn *G*) that are added to the benchmarks by considering all TVP, DAD, MDAD, and clock shifter blocks using DFFC Type C and hybrid techniques, respectively. In the case of s5378 circuit no improvement is reported in Table I for DFFC type C. So, in Table III adding extra hardware is reported to be not applicable (*NA*). According to subcolumn *FF*, the average number of added flip-flops using DFFC Type C and hybrid technique are 14 and 10 which are 2.8% and 2% of average number of flip-flops in the benchmarks. Subcolumn *G* shows that 49 and 50 gates on average have been added to the benchmarks using DFFC Type C and hybrid technique which are 0.62% and 0.63% of total number of gates in the benchmarks on average, respectively.

### B. Process, Power Supply, and Temperature Variations Results

In order to tolerate various kinds of variations such as process, power supply, and temperature variations, designers always have to include timing margins in the design to prevent HTV and STV which causes the loss of performance. To make an evaluation of our technique in the presence of these kinds of variations, we have performed Hspice simulation. We have implemented both hybrid and DFFC Type C techniques on challenging critical paths with transistor level description. To do so, we have synthesized the benchmark circuits in previous subsection using NanGate open cell library with 45 nm technology. After performing timing analysis on the synthesized benchmarks, the critical paths were determined using Design Compiler. Hspice net lists of the critical paths in circuits with short sequential depth (paths with LC problem) and critical paths that create SCP or CFP were extracted. Then we added the TVP, MDAD, Clock shifter, and PLL blocks to the net list of each critical path. We used 45 nm predictive models for the transistors. In previous subsection we observed that our method is able to improve MAF of benchmark circuits for 24.4% on average with a maximum of 36% improvement for benchmark circuit s38417. So we performed our simulations for 5 different frequencies ranging from the MAF (column *MAF No TB* in Table I) of the circuit to  $1.33 \times \text{MAF}$  which is the frequency range of our interest.

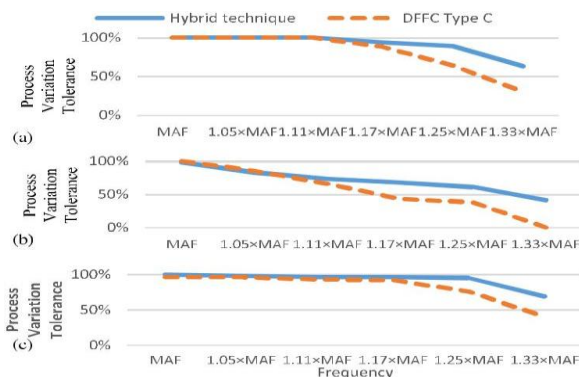


Fig. 8.

Variation tolerance for different path structures (a) short sequential depth (LC problem), (b) CFP, and (c) SCP. Note that,  $1.m \times \text{MAF}$  represents  $m\%$  increasing of MAF.

1) **Process Variation:** To apply process variations, both random global and local variations are modeled on the threshold voltage of the transistors of all circuit elements and blocks and the Monte Carlo simulation is performed. It should be noted that in corner-based analysis sources of variations have worst case values for all parameters and it is clearly too pessimistic. On the other hand, in Monte Carlo simulation all values of parameters within the distribution range are considered during simulation. The amount of variations is usually demonstrated with a parameter called  $3\sigma/\mu$ . It is the ratio of three standard deviations to the mean value. This parameter was assumed 15% in [18] on the sources of variation like threshold voltage, and 20% in [13] on the gate delays. In [1] and [11] a 21% variation from the nominal value on the threshold voltage of transistors ( $3\sigma/\mu = 21\%$ ) was considered. In this work, we set the variation equal to 21% to make sure that we have considered the worst case design in our results.

In Fig. 8 the percentage of variation tolerance in 1000 iterations for both DFFC Type C and our hybrid technique is presented for short sequential, CFP, and SCP structures. Fig. 8(a) and (c) show the results for the critical paths with short sequential depth and SCP, respectively. In these path structures both methods seem to be effective for operational frequencies close to MAF. However, by increasing the frequency from  $11 \times \text{MAF}$  to  $1.17 \times \text{MAF}$  or higher frequencies the hybrid technique becomes more effective. At operating frequency of  $1.33 \times \text{MAF}$ , the hybrid technique can tolerate variation 33% and 24% better than DFFC Type C method for short sequential depth and SCP structures, respectively.

The simulation results for four cycles of activation of CFP structure is depicted in Fig. 8(b). Note that since our chosen

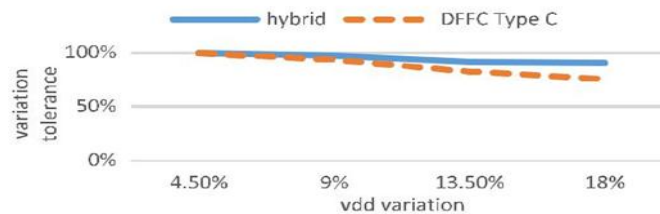


Fig. 9.

Average Power supply variation tolerance for different path structures including short sequential depth (LC problem), CFP, and SCP

CFP is not the most critical path in the circuit, DFFC Type C method can be a better solution when working at MAF (column *MAF No TB* in Table I). The reason for that is the fact that the variation in “Clock Shifter” block makes the clock period shorter or longer and this causes timing violations. By increasing the frequency up to  $1.11 \times \text{MAF}$  the SCP becomes critical and that is when our hybrid technique seems to be more effective than DFFC Type C method. At operational frequency of  $1.33 \times \text{MAF}$  the DFFC Type C completely fails to tolerate variations while our hybrid technique can still tolerate 41% of variations. In summary, our hybrid technique can tolerate the variations 27.3% on average better than DFFC Type C method for short sequential depth, CFP, and SCP challenging path structures when the circuit operates at  $1.33 \times \text{MAF}$ .

2) **Power Supply Variation:** In order to investigate how well the design equipped with our hybrid technique can tolerate power supply variations we have applied 18% variation from our 1.1 volts nominal value in power supply of critical paths from our benchmarks with short sequential depth, CFP, and SCP structures. Fig. 9 shows the percentage of power supply variation tolerance in 1000 iterations for critical paths using both DFFC Type C and our hybrid technique. The results show that with 4.5% variation in power supply both techniques seem to be very effective. However, by increasing the power supply variation, hybrid technique seems to be more effective than DFFC Type C and for 18% variation in power supply voltage, the hybrid technique is 17% more effective than DFFC Type C on average. Also we investigated the MAF of the critical paths equipped with DFFC Type C, hybrid technique and no time borrowing for different values of power supply voltages. Our experiment results are depicted in Fig. 10. Our experiments show that critical paths equipped with our hybrid technique have 19.1% and 38.3% higher MAF than those equipped with DFFC Type C and no time borrowing respectively.

3) **Temperature Variation:** In another experiment, we have applied four different values of temperature variations from the nominal temperature value of  $25^\circ\text{C}$  to the paths of circuits equipped by DFFC Type C and our hybrid technique to investigate the temperature variation tolerance of the circuits. Fig. 11 shows the percentage of temperature variation tolerance in 1000 iterations for 9 critical path including short sequential depth, CFP, and SCP structures. Our experiments show that our hybrid technique is able to tolerate temperature variation 6% more than DFFC Type C on average.

Furthermore, we linearly increased the temperature from  $-75\text{ }^{\circ}\text{C}$  to  $125\text{ }^{\circ}\text{C}$  for critical paths with no time borrowing as well as those equipped with DFFC Type C, and our hybrid

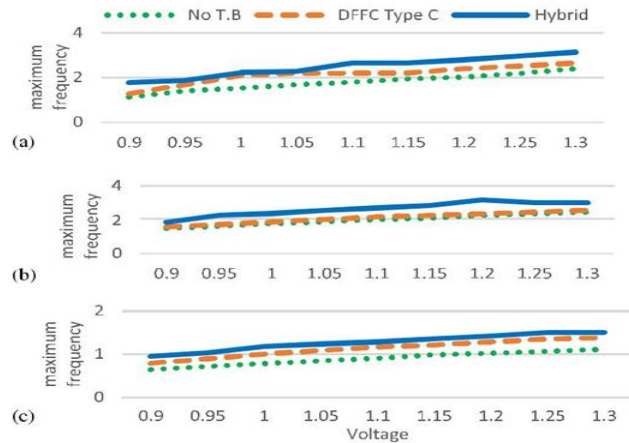


Fig. 10.

Power supply variation tolerance for different path structures: (a) short sequential depth (LC problem), (b) CFP, and (c) SCP.

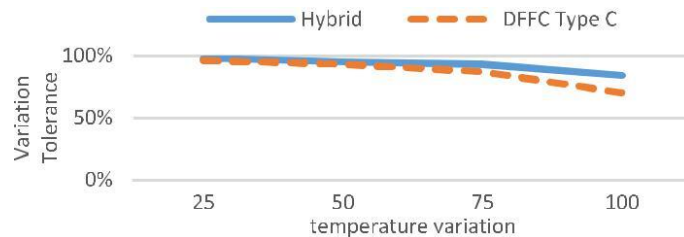


Fig. 11.

Temperature variation tolerance for different path structures including short sequential depth (LC problem), CFP, and SCP. The horizontal axis is in centigrade degree.

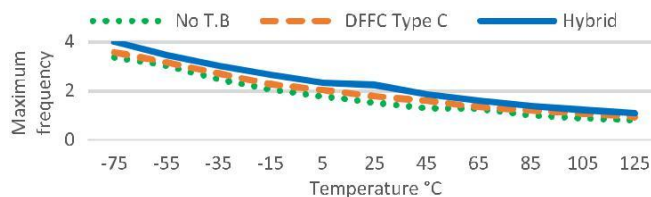


Fig. 12. Temperature variation tolerance for different path structures: short sequential depth (LC problem), CFP, and SCP.

technique. Then we have investigated the MAF of critical paths in the presence of temperature variation. Fig. 12 shows the average MAF in GHz for three path structures including short sequential depth, CFP, and SCP structures. As can be seen in Fig. 12, the MAF of hybrid technique is 13.3% and 22.8% higher than that of DFFC Type C and no time borrowing on average, respectively.

C. Power Analysis

In the third experiment, we have used Power Compiler to estimate power consumption. To make a fair comparison, the power of benchmark circuits using different techniques was calculated for the same number of clock cycles at the same clock frequency. Table IV reports the summation of dynamic and leakage powers of the benchmark circuits equipped with DFFC Type C [12], SEFF with elastic clock stretching [4], and our hybrid method. The first column (*MAF No TB*) gives the MAF of each benchmark circuit when no time borrowing technique is used. The second column (*F req*) is the MAF of the circuit in which the power of benchmarks is calculated.

TABLE IV  
THE POWER CONSUMPTION COMPARISON OF DFFC TYPE C, SEFF + ECS, HYBRID TECHNIQUES  
(ALL REPORTED POWER IN  $\mu$ W; NA: NOT APPLICABLE)

Benchmarks	MAF No TB	Freq	Power No TB	PowerC	Power SEFF+ECS	PowerH		PowerH AtSpeed	
						DUT	ClockShifter	DUT	ClockShifter
b05	1.28	1.39	408	584	449	488	85	616.24	89
b11	1.19	1.31	315.6	514	375	529	158	635	193
b12	1.28	1.39	1123	1926	NA	1689	175	1764	187
b15	1.06	1.11	4547	4990	NA	4538	136.5	5250	157.7
s5378	1.09	NA	3093	NA	NA	NA	NA	4241	271
s38417	1.47	1.61	18432	21347	22978	20807	396.5	25372	484.6
s38584	1.79	1.85	14868	21369	19239	16711	239	20601	257
Average	1.308	1.39	6112.4	8451	7951	7460	198.33	8354	234.18
Power increase compared to Power No TB %				38%	30%	22.04%	NA	36.67%	NA

This frequency is the minimum MAF between three techniques (i.e., DFFC Type C, SEFF with elastic clock stretching, and hybrid technique reported on Tables I and II) so that all benchmark circuits equipped with three selected techniques are able to operate at this frequency. The *Power No TB* column shows the power of the benchmark circuits when working at *MAF No TB*. The next three columns present the power consumption of the benchmarks working at *F req* when equipped with DFFC Type C (*Power C*), SEFF with elastic clock stretching (*PowerSEFF + ECS*), and our hybrid method (*PowerH*), respectively. The last column (*PowerH AtSpeed*) shows the power consumption of the circuits equipped by hybrid technique when working at the highest frequency reported in Table I. In the last two columns, the subcolumns *DUT* and *ClockShifter* present the power consumption of the benchmark circuit and the power of Clock Shifter block, respectively. Note that in the case of b12 and b15 circuits SEFF + ECS technique is not able to work at 1.39 and 1.11 GHz (Freq column) and therefore no power consumption is reported (NA in the table). Also, in the case of s5378 circuit neither of DFFC Type C nor SEFF + ECS techniques seems to be effective. So, in Table IV choosing a value for *F req* is also not applicable and therefore no power results are reported [Not Applicable (NA)]. At working frequency of *F req* the average power consumption of hybrid technique considering both ClockShifter and DUT is 25.29% higher than the average of *Power No TB*, while using DFFC Type C and SEFF + ECS techniques, 38% and 30% increase in the power consumption is reported. Even at higher frequencies the power consumption of the hybrid technique considering both ClockShifter and DUT remains reasonably low so that at the highest frequency obtained by the hybrid technique (reported in Table I) only 40.5% more power on average is consumed while the speed increases by 24.4%.

### V. CONCLUSIONS AND FUTURE WORK

In this paper a new hybrid technique was presented which utilizes dynamic clock stretching together with our previous time borrowing method, i.e., DFFC Type C. DFFC allows time borrowing in critical paths while in those critical paths that do not have enough timing slacks in their subsequence stages, clock stretching provides enough time to avoid timing violations. We have shown that when the circuit operating at the frequency of  $1.33 \times \text{MAF}$ , in challenging path structures, our new hybrid technique is able to tolerate process variations up to 27.3% better than state-of-the-art techniques. Moreover, with 18% variation in the power supply and 100 °C variation in the temperature, our hybrid technique is 16.4% and 13.3% more effective than DFFC Type C, respectively.

Furthermore, our simulation results show that our new hybrid technique improves the performance of the circuits by 24.4% on average while using our previous DFFC Types B and C methods an average of 6.7% and 8.4% increase in the performance is possible, respectively. The results also show that at the same operational frequency, our hybrid method has 4.71% and 12.71% less power consumption on average compared to SEFF + ECS and DFFC method, respectively. As our future work, we are going to investigate new hybrid techniques that utilize a combination of different techniques applicable to various critical paths. Another avenue of future work is to implement our hybrid technique on FPGA architectures.

### ACKNOWLEDGMENT

we would like to thank **Mr.K.G.Parthiban**, ASP &HoD/ECE for the support.

**REFERENCES**

- [1] M. Nejat, B. Alizadeh, and A. Afzali-Kusha, "Dynamic flip-flop conversion to tolerate process variation in low power circuits," in *Proc. DATE*, 2014, pp. 1–4.
- [2] M. Choudhury, V. Chandra, R. Aitken, and K. Mohanram, "Time-borrowing circuit designs and hardware prototyping for timing error resilience," *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 497–509, Jan. 2014.
- [3] L. Seonggwan, S. Paik and Y. Shin, "Retiming and time borrowing: Optimizing high-performance pulsed-latch-based circuits," in *Proc. ICCAD*, 2009, pp. 375–380.
- [4] K. Chae, C. H. Lee, and S. Mukhopadhyay, "Timing error prevention using elastic clocking," in *Proc. ICICDT*, 2011, pp. 1–4.
- [5] Y. Chen and Y. Xie, "Tolerating process variations in high-level synthesis using transparent latches," in *Proc. ASP-DAC*, 2009, pp. 73–78.
- [6] J. H. Anderson and B. Teng, "Latch-based performance optimization for field-programmable gate arrays," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 5, pp. 667–680, Apr. 2013.
- [7] J. S. Wang, K. J. Chang, T. J. Lin, R. W. Prasojo, and C. Yeh, "A 0.36 V, 33.3  $\mu$ W 18-band ANSI S1. 11 1/3-octave filter bank for digital hearing aids in 40 nm CMOS," in *Proc. VLSIC*, 2013, pp. 54–C255.
- [8] M. Wieckowski, M. P. Young, C. Tokunaga, D. W. Kim, Z. Foo, D. Sylvester, and D. Blaauw, "Timing yield enhancement through soft edge flip-flop based design," in *Proc. CICC*, 2008, pp. 543–546.
- [9] V. Joshi, "Soft-edge flip-flops for improved timing yield: Design and optimization," in *Proc. ICCAD*, 2007, pp. 667–673.
- [10] Chae, Kwanyeob, and S. Mukhopadhyay, "Resilient pipeline under supply noise with programmable time borrowing and delayed clock gating," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 61, no. 3, pp. 173–177, Mar. 2014.
- [11] M. Nejat, B. Alizadeh, and A. Afzali-Kusha, "Dynamic flip-flop conversion: A time-borrowing method for performance improvement of low-power digital circuits prone to variations," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 11, pp. 2724–2727, Nov. 2015.
- [12] M. Ahmadi, B. Alizadeh, and B. Forouzandeh, "A timing error mitigation technique for high performance designs," in *Proc. ISVLSI*, 2015, pp. 428–433.
- [13] V. Mahalingam, "Dynamic clock stretching for variation compensation in VLSI circuit design," *J. Emerg. Technol. Comput. Syst.*, vol. 8, no. 16, pp. 1–13, Aug. 2012.
- [14] (Accessed: Jul. 2015) ITC'99 benchmarks. [Online]. Available: <http://www.cad.polito.it/downloads/tools/itc99.html>
- [15] (Accessed: Jul. 2015) ISCAS'89 benchmark. [Online]. Available: <http://www.pld.ttu.edu/~maksim/benchmarks/iscas89/verilog/>
- [16] (Accessed: Jul. 2015). [Online]. Available: <http://www.synopsys.com/Tools/Implementation/RTLsynthesis/Pages/default.aspx>
- [17] (Accessed: Jun. 2013). NanGate FreePDK45 Generic Open Cell Library\_v1.3\_2010\_12. [Online]. Available: <http://www.si2.org/openeda.si2.org/projects/nangatelib/>
- [18] M. Ghasemazar and M. Pedram, "Optimizing the power-delay product of a linear pipeline by opportunistic time borrowing," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, pp. 1493–1506, 2011.
- [19] J. S. Wang, "Dynamic voltage scaling system having time borrowing and local boosting capability," U.S. Patent 8933726, 06 26, 2014.

**BIOGRAPHIES**

**K.NANTHAKUMAR** completed his M.E VLSI DESIGN in M.P.Nachimuthu M.Jaganathan Engineering College, Erode and Completed B.E Electrical and Electronics Engineering under Anna University, Chennai and has 5 years of teaching experience. Now working as a assistant professor in ECE department in M.P.Nachimuthu M.Jaganathan Engineering College,Erode.



**S.THEJASWINI** received B.E degree in Electronics and Communication Engineering from Anna University, Chennai, Tamil Nadu, and India in 2015. She is currently doing M.E.VLSI DESIGN in Anna University, Chennai, Tamil Nadu, and India. Her areas of interest include Digital circuit design, logic synthesis and test, variable latency designs, especially low power and high performance VLSI designs.