LOW POWER AND TIMING ERROR MITIGATION CIRCUIT DESIGN BASED ON CLOCK GATING

Parimala Gandhi Ayyavu[1], Jemima S[2]
Professor[1], ME Student (VLSI Design)
[1,2]Department of Electronics and Communication Engineering,
KIT-Kalaignarkarunanidhi Institute of Technology, Coimbatore-641402.
parimalagandhi@yahoo.com[1], jemima.262000@gmail.com[2]

Abstract:

Timing error is now getting increased attention due to the high rate of error-occurrence on semiconductors. Even slight external disturbance can threaten the timing margin between successive clocks since the latest semiconductor operates with high frequency and small supply voltage. To deal with a timing error, many techniques have been introduced. Nevertheless, existing methods that mitigate a timing error mostly have time-delaying mechanisms and too complex operation, resulting in a timing problem on clock-based systems and hardware overhead. In the proposed work a novel timing-error-tolerant method that can correct a timing error instantly through a simple mechanism is demonstrated. By modifying a clock in a flip-flop, the proposed system can recover a timing error without the loss of time in the clock-based system. Furthermore, in order to reduce power consumption in the stages were operation is not performed clock gating mechanism is used to reduce the unwanted transition. Look-Ahead Clock Gating (LACG) computes the clock enabling signals of each FF one cycle ahead of time, based on the present cycle data of those FFs on which it depends. It has however a big advantage of avoiding the tight timing constraints of earlier methods, by allotting a full clock cycle for the enabling signals to be computed and propagate to their gaters.

Corresponding Author: *R.Karthick*
Department of ECE,
KIT-Kalaignarkarunanidhi Institute of Technology, Coimbatore
parimalagandhi@yahoo.com

LACG mainly addresses three goals; stopping the clock pulse also in the master latch, making it applicable for large and general designs and avoiding the tight timing constraints. LACG is based on using the XOR output to generate clock enabling signals of other FFs in the system, whose data depend on that FF. The XOR output is valid only during a narrow window of around the clock rising edge, where and are the FF’s setup time and clock to output contamination delay, respectively. After a delay the XOR output is corrupted and turns eventually to zero. To be valid during the entire positive half cycle it must be latched and it is done using enhanced AGFF with the XOR output. Due to the compact mechanism, the proposed system has low hardware overhead in comparison with existing timing-error-tolerant systems that can recover the error instantly.

Key words: Look-Ahead Clock Gating (LACG), Error Mitigation, Soft-edge flip-flops (SEFF).

Introduction:

The timing-error rate is increased as the clock frequency is increased. Since the clock period is getting minimized, critical paths in the circuit are susceptible to timing errors [1-11]. Furthermore, variations on the CMOS process, power supply, and temperature impact the performance of modern integrated circuits, which results in the high incidence of timing errors. As the supply voltage decreases, the delay of the circuit can drastically change between the typical case and the worst case of process, voltage, and temperature (PVT) conditions [3]. With the 0.4-V operation, the logical path with the worst case is 12x slower than that with the typical case [4]. Moreover, transistor aging issues are critical for the occurrence of timing errors [5], [6]. Due to the negative-bias temperature instability (NBTI) in CMOS, the threshold voltage is lowered, which finally increases the path delays of logics. Hence, a timing-error-tolerant method is highly required in order to implement reliable systems.

Time Borrowing Process

Energy- and power-efficient designs are becoming very important in today’s electronic systems. The applications of today and the future require a system that not only consumes a low amount of power, but also preserves the high performance. Recently, designing in near-threshold voltages has been presented as a way to achieve this goal. Nevertheless, a serious challenge is the effect of process, voltage, and temperature variations on the system performance. Worst case design is too conservative as it imposes a severe limitation on the performance of every single product to ensure the correct operation of a small fraction of them. Using error detection, correction, and prevention techniques such as time borrowing (TB), the high performance could be preserved, while the errors in worst cases are prevented or corrected. This brief presents a novel TB method called dynamic flip-flop conversion (DFFC, Type B). The first type of DFFC (Type A) has been proposed. Hence, the main contributions of this brief are as follows.

1) A new structure is presented for DFFC, which automatically closes the transparency window after the arrival of the late data at critical FFs.

2) Statistical Monte Carlo analysis on the critical path of different benchmarks with different TB methods was performed.

3) The effects of frequency change and voltage scaling down to near-threshold region have been studied on different TB methods.

Master–slave FFs are widely used in digital circuits. If we apply the inverted clock signal to the master latch, the FF changes to two similar latches that behave like one. This is the main idea of DFFC. This way, an edge-sensitive register can be dynamically converted to a level-sensitive one and vice versa. The first type of DFFC (Type A) was previously presented. In this structure, a timing violation predictor (TVP) detects the STVs halfway in the critical path and, using an XOR gate, converts the FF to a latch so that the late data can still pass the critical FF [12-21].

The demand for high performance design with low power consumption 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, is computed based on the delay of the longest paths (critical paths) in the circuit. Such critical paths, which are rarely activated, make us keep the frequency low and that is why we lose the performance. It is obvious that by increasing the clock frequency we achieve better performance. However, the problem would be the fact that critical paths are not able to finish their job and therefore setup/hold time violation would occur. In order to address these timing violation issues three main approaches including error detection and correction, retiming and time borrowing have been proposed [22-30].

RELATED WORKS

1) Timing Error Tolerance in Small Core Designs for SoC Applications

Timing errors are an increasing reliability concern in nanometer technology, high complexity and multivoltage/ frequency integrated circuits. A local error detection and correction technique is presented in this work that is based on a new bit flipping flip-flop. Whenever a timing error is detected, it is corrected by complementing the output of the corresponding flip-flop. The proposed solution is characterized by very low silicon area and power requirements compared to previous design schemes in the open literature. To validate its efficiency, it has been applied in the design of a MIPS microprocessor core in a 90nm technology, while a demonstration version of the same core in an FPGA platform is presented.

2) Dynamic Flip-Flop Conversion: A Time-Borrowing Method for Performance Improvement of Low-Power Digital Circuits Prone to Variations

Dynamic flip-flop (FF) conversion is a method of time borrowing (TB) for improving the performance of digital systems prone to variations. The first type of this method (Type A), which was previously presented, suffers from a large inefficient transparency window. In this brief, we present an improved structure for this method (Type B) that mitigates this problem by automatically closing the window after the arrival of late data at timing critical FFs. This method was compared with soft edge FF and dynamic clock stretching through simulations on different ITC’99 benchmarks. We defined a parameter called improvement efficiency, which is the ratio of the timing yield improvement to the power overhead of TB. According to the simulation results, the efficiency of Type A is on average 269% more than the best results of other methods when considering only the setup time violations. But, when taking both the setup time and the hold time violations into account, Type B is on average 46% more efficient than the best results of other methods. The simulations also show that the yield improvement of this method increases in higher clock frequencies and it remains the most efficient method when reducing the voltage down to near-threshold region.

3) A Timing Error Mitigation Technique for High Performance Designs

Dynamic flip-flop conversion (DFFC) is a time borrowing method which converts the critical flip-flops into transparent latches to allow timing slacks pass between pipeline stages of given circuits. However, our previous DFFC methods [13] [14] suffer from false error prediction. It means even when there is no setup time violation, our previous method incorrectly issues timing error. In this paper we present an improved DFFC method which consumes less power and unlike previous DFFC methods does not suffer from false errors. Also we investigate the effectiveness of our proposed DFFC method on different benchmarks by considering all existing critical paths, instead of applying our method only to one of critical paths as done in [13][14]. The results show that our proposed method improves maximum allowable frequency 7.54% on average while the previousmethod in [13] [14] increases maximum allowable frequency 4.24% on average. Furthermore, it consumes on average 19.59 microwatt less power per critical path compared to previous DFFC methods [13][14].

4) iRazor: Current-Based Error Detection and Correction Scheme for PVT Variation in 40-nm ARM Cortex-R4 Processor

This paper presents iRazor, a lightweight error detection and correction approach, to suppress the cycle time margin that is traditionally added to very large scale integration systems to tolerate process, voltage, and temperature variations. iRazor is based on a novel current-based detector, which is embedded in flip-flops on potentially critical paths. The proposed iRazor flip-flop requires only three additional transistors, yielding only 4.3% area penalty over a standard D flip-flop. The proposed scheme is implemented in an ARM Cortex-R4 microprocessor in 40 nm through an automated iRazor flip-flop insertion flow. To gain an insight into the effectiveness of the proposed scheme, iRazor is compared to other popular techniques that mitigate the impact of variations, through the analysis of the worst case margin in 40 silicon dies. To the best of the authors’ knowledge, this is the first paper that compares the measured cycle time margin and the power efficiency improvements offered by frequency binning and various canary approaches. Results show that iRazor achieves 26%–34% performance gain and33%–41% energy reduction compared to a baseline design across the 0.6- to 1-V voltage range, at the cost of 13.6% area overhead.

PROPOSED METHOD

PROPOSED TIMING-ERROR-TOLERANT SYSTEM

A new timing-error-tolerant system that can correct timing error is developed. When a timing error causes a delayed arrival of an input on a flip-flop, the proposed system can detect a delayed input of the flip-flop, and the flip-flop passes through the data by making a transparent window. As described in Fig. 1(a), the proposed system consists of a "transition detector" and "master clock generator." The "transition detector" detects the input transition of a flip-flop, and it produces a pulse of the error-flagged signal. Based on the output of the transition detector, the "master clock generator" produces a pulse for a certain period only when a clock is high. Thus, the abnormal data that are stored in the flip-flop can be restored with delayed normal data. To avoid hold time violation, a pulse is generated with a minimum time, which is required for the setup time.

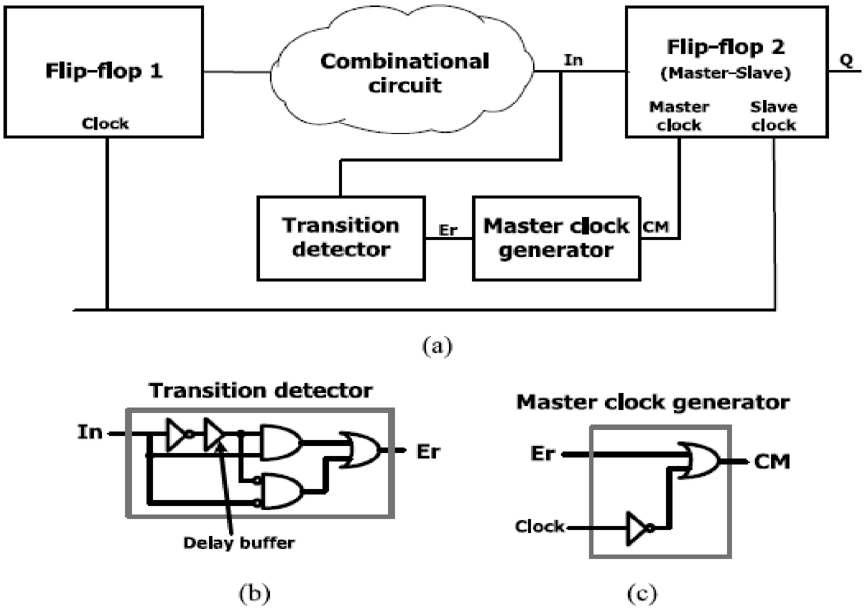


Fig. 3.1. Circuit of the proposed timing-error-tolerant system. (a) Circuit diagram of the proposed system. (b) Circuit of transition detector. (c) Circuit of the master clock generator.

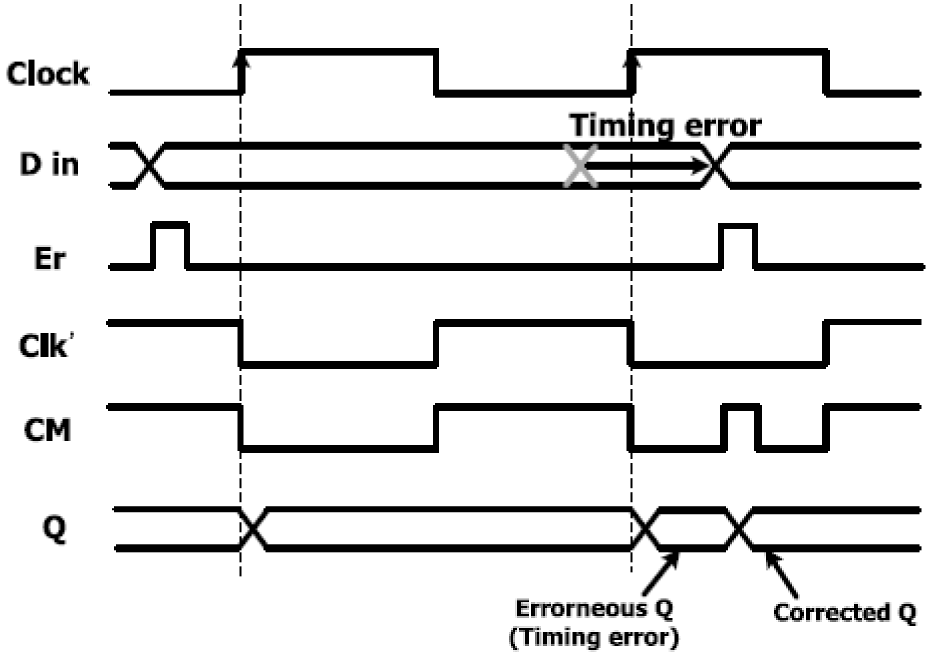
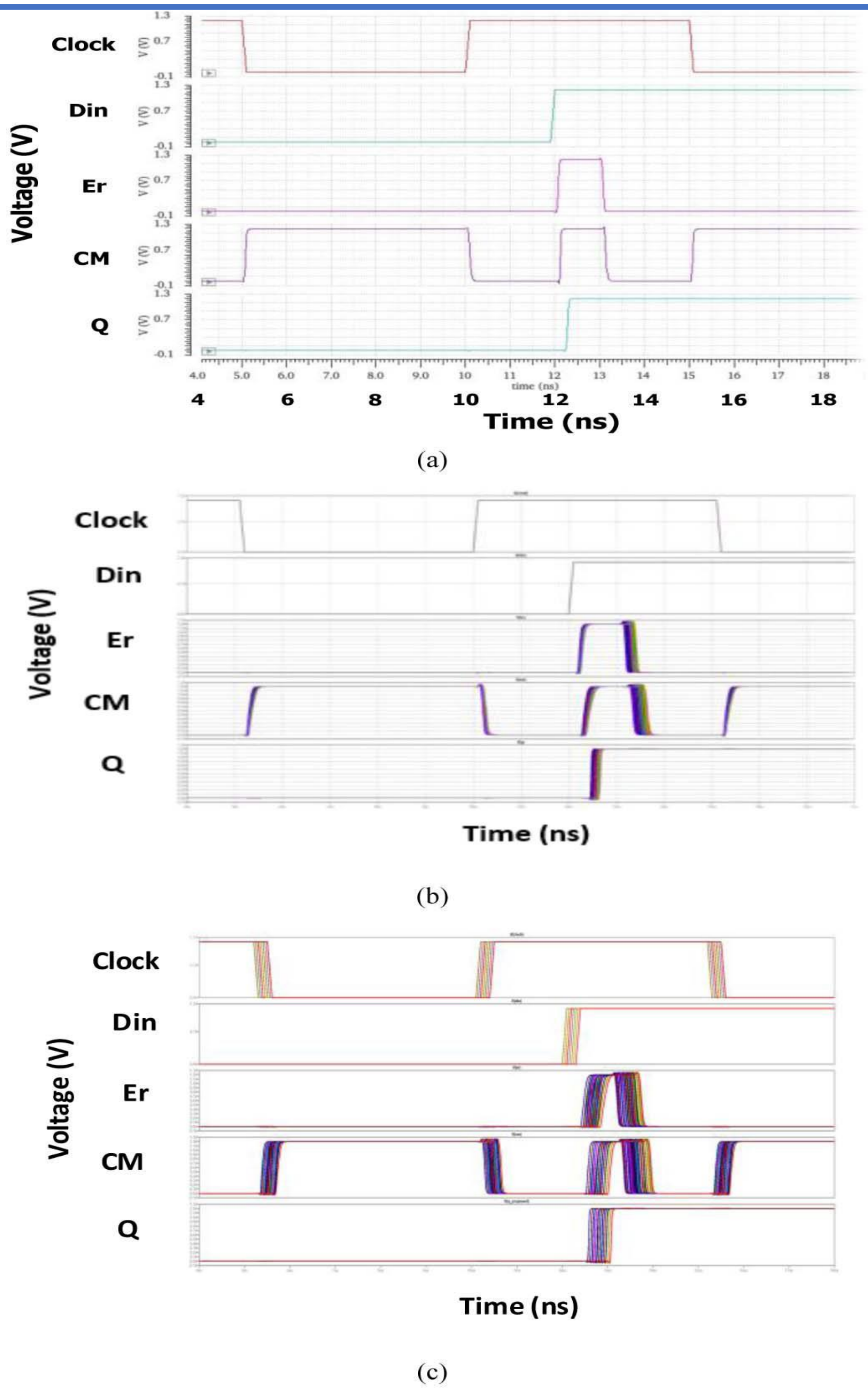


Fig. 3.2. Operation of the proposed system when a timing error occurs.

As shown in Fig. 2, the recovery process of a timing error in the proposed system is fully addressed. When a timing error occurs on the input data of the flip-flop 2, the flip-flop 2 stores abnormal data to the output Q due to the delayed input data. After the delayed normal data have arrived on the input of flip-flop 2, the transition detector that is located between a combinational circuit and the flip-flop 2 generates an error pulse. The transition detector detects both a rising edge of the data and a falling edge of the data by using an inverter and the AND gate. Furthermore, by implementing a delay buffer in the transition detector, the transition detector generates a customized period of a pulse, which maintains the transparent window for enough time. The master clock generator makes a clock of the master (CM) based on the error pulse and clock. Due to the OR gate and inverter in the master clock generator, a CM is high for a certain period of time after a timing error occurs. While the CM is high, the flip-flop 2 becomes transparent, and the delayed normal input is stored through flip-flop 2.



(a)

(b)

(c)

Fig. 3.3 Simulated waveforms of the proposed system with CMOS design. (a) Simulated waveforms with a 90-nm process. (b) Simulated waveforms with process and mismatch variations. (c) Simulated waveforms with the process, clock, and data-arrival variations.

Proposed Timing-Error-Tolerant System with Time-Borrowing Technique

Using the information acquired from the timing analyzing tool, the critical paths of the circuit and the setup-time information of each element in the circuit are determined. We make full use of such results to choose the best location for our proposed system. If a single-stage error occurs, the delayed arriving data signal is recovered because the master latch is transparent for the pulse period. However, if a successive-stage error occurs due to the lack of setup time in the second-stage flip-flop, the delayed arriving data signal in the second stage cannot be stored because of the setup-time violation. To deal with the successive-stage error, we devised the time-borrowing technique. The time-borrowing circuit is provided in the second stage, as shown in Fig. 4.

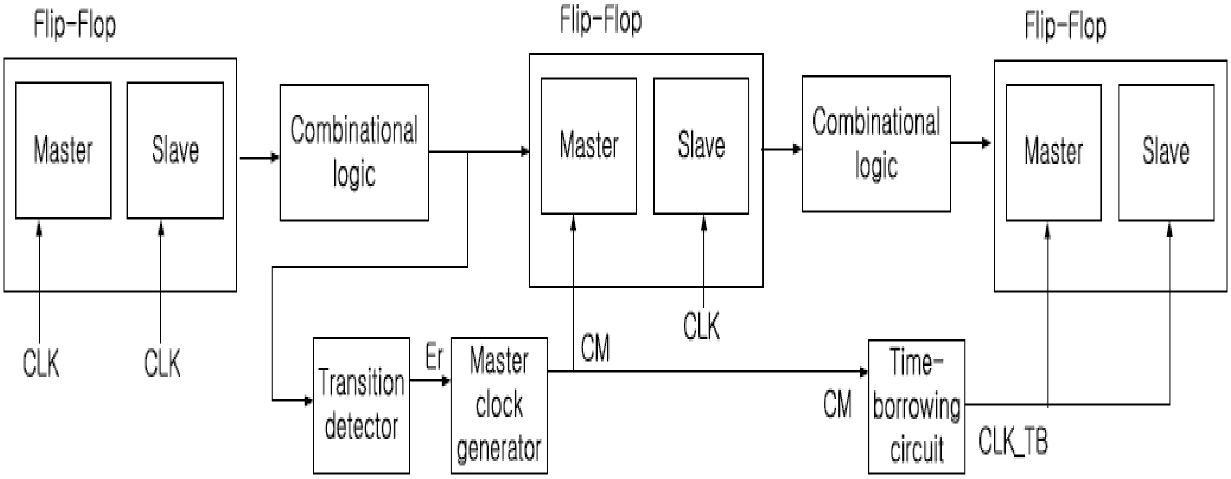
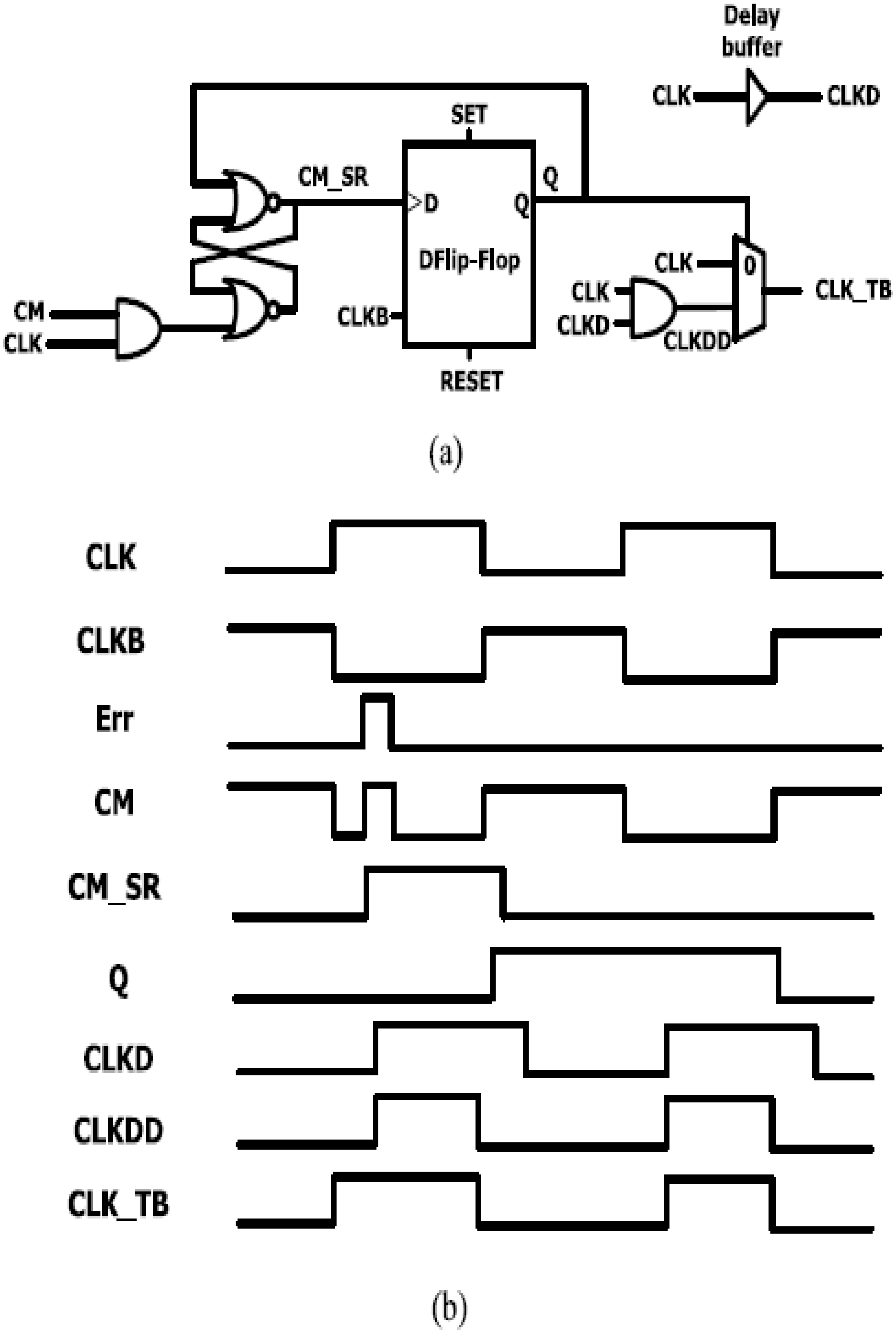


Fig.3.4. Timing-error-tolerant circuit with the time-borrowing circuit.

As shown in Fig. 5, the operation of a time-borrowing circuit is addressed. When a timing error occurs on the input data of the first stage, the CM signal is set to “1,” which also induces the CM_SR high level. After CLK is fallen, Q is set to “1.” While the Q signal sets the high period, the delayed CLK (CLKDD) is chosen as the main CLK for the second stage. After all, the new CLK maintains a transparent window for enough time. A CM is high for a certain period of time after a timing error occurs. Thus, the delayed data can be stored as normal data. The time-borrowing scheme can be used on any location that has a short setup time for the flip-flop.



(a)

(b)

Fig. 3.5. Time-borrowing technique. (a) Time-borrowing circuit structure. (b) Operation of the time-borrowing system.

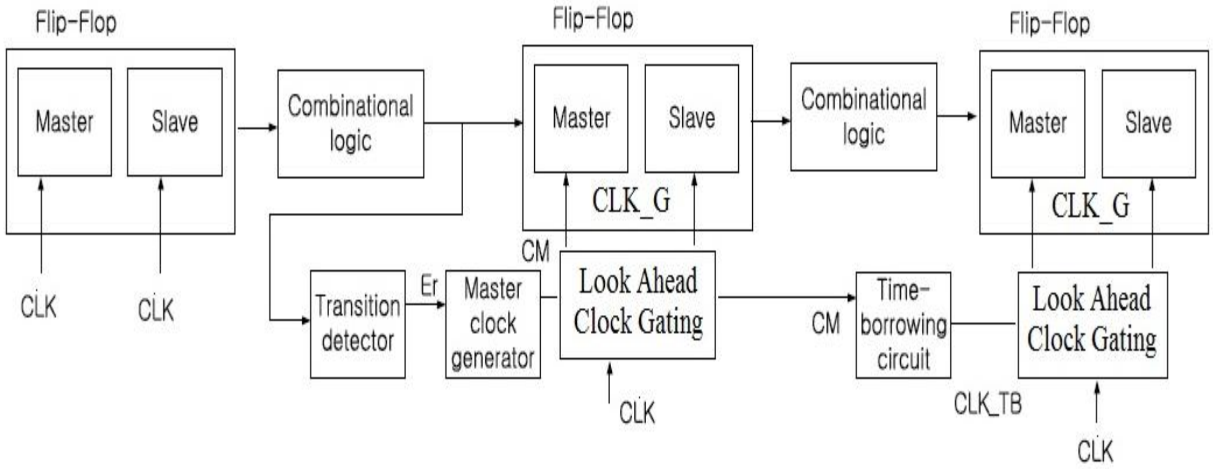


Fig.3.6. Proposed Timing-error-tolerant circuit with the time-borrowing and clock gating circuit.

The proposed error tolerant circuit with time-borrowing circuit and clock gating circuit is shown in Fig. 6.

Clock Gating:

Clock gating is a popular technique used in many synchronous circuits for reducing dynamic power dissipation. Clock gating saves power by adding more logic to a circuit to prune the clock tree. Pruning the clock disables portions of the circuitry so that the flip-flops in them do not have to switch states. Switching states consumes power. When not being switched, the switching power consumption goes to zero, and only leakage currents are incurred. Clock gating works by taking the enable conditions attached to registers, and uses them to gate the clocks. A design must contain these enable conditions in order to use and benefit from clock gating. This clock gating process can also save significant die area as well as power, since it removes large numbers of muxes and replaces them with clock gating logic. This clock gating logic is generally in the form of "Integrated clock gating" (ICG) cells. However, note that the clock gating logic will change the clock tree structure, since the clock gating logic will sit in the clock tree. Clock gating logic can be added into a design in a variety of ways:

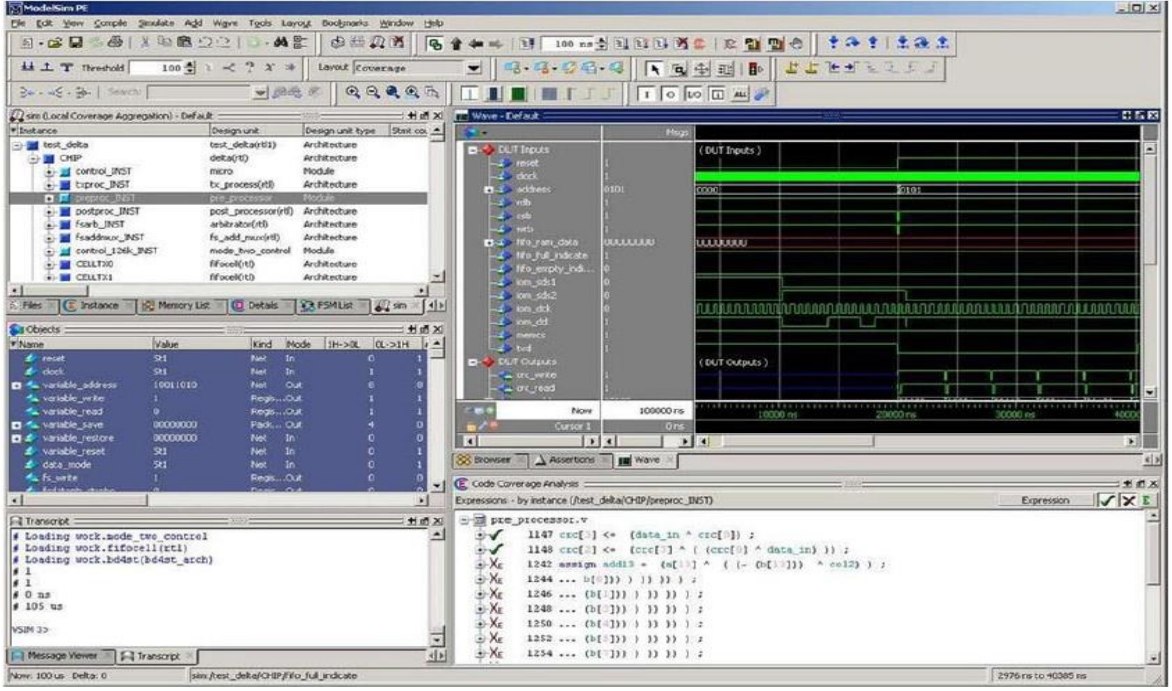
1. Coded into the RTL code as enable conditions that can be automatically translated into clock gating logic by synthesis tools (fine grain clock gating).
2. Inserted into the design manually by the RTL designers (typically as module level clock gating) by instantiating library specific ICG (Integrated Clock Gating) cells to gate the clocks of specific modules or registers.
3. Semi-automatically inserted into the RTL by automated clock gating tools. These tools either insert ICG cells into the RTL, or add enable conditions into the RTL code. These typically also offer sequential clock gating optimizations

RESULTS

MODELSIM

ModelSim is a multi-language HDL simulation environment by Mentor Graphics, for simulation of hardware description languages such as VHDL, Verilog and SystemC, and includes a built-in C debugger. ModelSim can be used independently, or in conjunction with Altera Quartus or XilinxISE. Simulation is performed using the graphical user interface (GUI), or automatically using scripts. The ModelSim environment is shown in Fig.4.2.ModelSim is offered in multiple editions, such as ModelSim PE, ModelSimSE, and ModelSim XE. ModelSim SE offers high-performance and advanced debugging capabilities, while ModelSim PE is the entry-level simulator for hobbyists and students. ModelSim SE is used in large multi-million gate designs, and is supported on Microsoft Windows and Linux, in 32-bit and 64-bit architectures.

ModelSim XE stands for Xilinx Edition, and is specially designed for integration with Xilinx ISE. ModelSim XE enables testing of HDL programs written for Xilinx Virtex/Spartan series FPGA's without needed physical hardware. ModelSim can also be used with MATLAB/Simulink, using Link for ModelSim. Link for ModelSim is a fast bidirectional co-simulation interface between Simulink and ModelSim.



sim (Local Coverage Aggregation) - Default

Instance	Design unit	Design unit type
test_delta	test_delta(rtl1)	Architecture
CHIP	delta(rtl)	Architecture
control_INST	micro	Module
txproc_INST	tx_process(rtl)	Architecture
preproc_INST	pre_processor	Module
postproc_INST	post_processor(rtl)	Architecture
fsarb_INST	arbitrator(rtl)	Architecture
fsaddmux_INST	fs_add_mux(rtl)	Architecture
control_126k_INST	mode_two_control	Module
CELLTX0	fifocell(rtl)	Architecture
CELLTX1	fifocell(rtl)	Architecture

Objects

Name	Value	Kind	Mode	1H->0L	0L->1H
reset	St1	Net	In	0	1
clock	St1	Net	In	1	1
variable_address	10011010	Net	Out	6	0
variable_write	1	Regis...	Out	1	1
variable_read	0	Regis...	Out	1	1
variable_save	00000000	Pack...	Out	4	0
variable_restore	00000000	Net	In	0	0
variable_reset	St1	Net	In	0	1
data_mode	St1	Net	In	0	1
fs_write	1	Regis...	Out	0	0

Transcript

```
# Loading work.mode_two_control
# Loading work.fifocell(rtl)
# Loading work.bd4st(bd4st_arch)
# 1
# 1
# 0 ns
# 105 us

VSIM 3>
```

Code Coverage Analysis

Expressions - by instance (/test_delta/CHIP/preproc_INST)

```
pre_processor.v
✓   1147 crc[3] <=  (data_in ^ crc[0]) ;
✓   1148 crc[2] <=  (crc[3] ^ ( (crc[0] ^ data_in) )) ;
XE  1242 assign add13 =  (a[13] ^  ( (~ (b[13]))  ^ co12) ) ;
XE  1244 ... b[0])) ) )) )) ) ;
XE  1246 ... (b[1])) ) )) )) ) ;
XE  1248 ... (b[3])) ) )) )) ) ;
XE  1250 ... (b[4])) ) )) )) ) ;
XE  1252 ... (b[5])) ) )) )) ) ;
XE  1254 ... (b[7])) ) )) )) ) ;
```

Now: 100 us Delta: 0

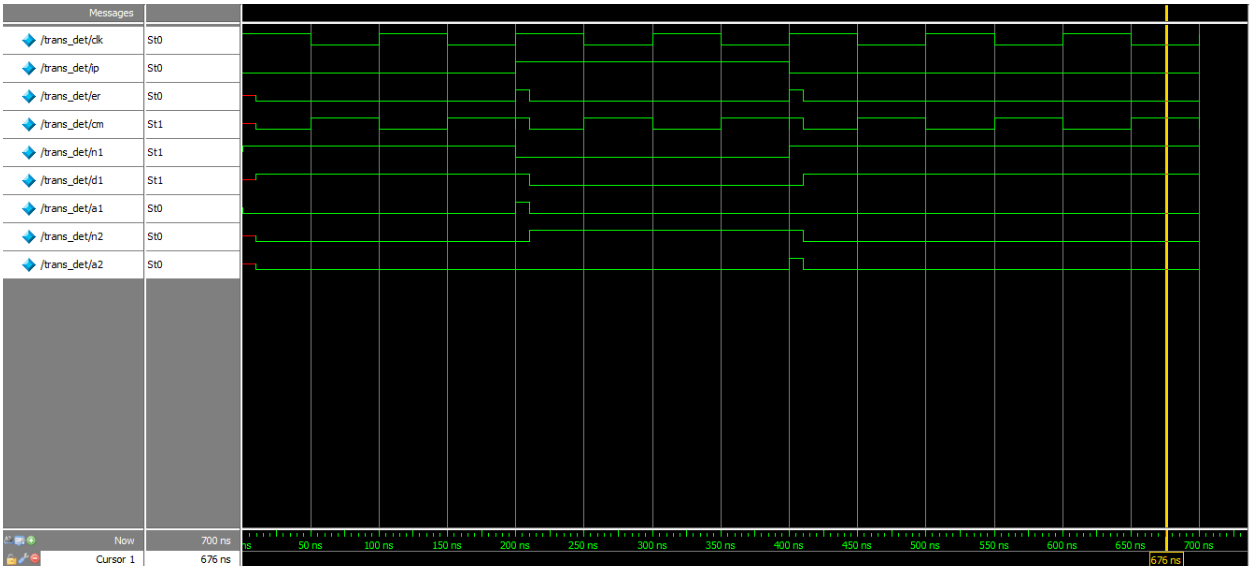
sim:/test_delta/CHIP/fifo_full_indicate

2976 ns to 40385 ns

Fig.4.2.ModelSim Environment

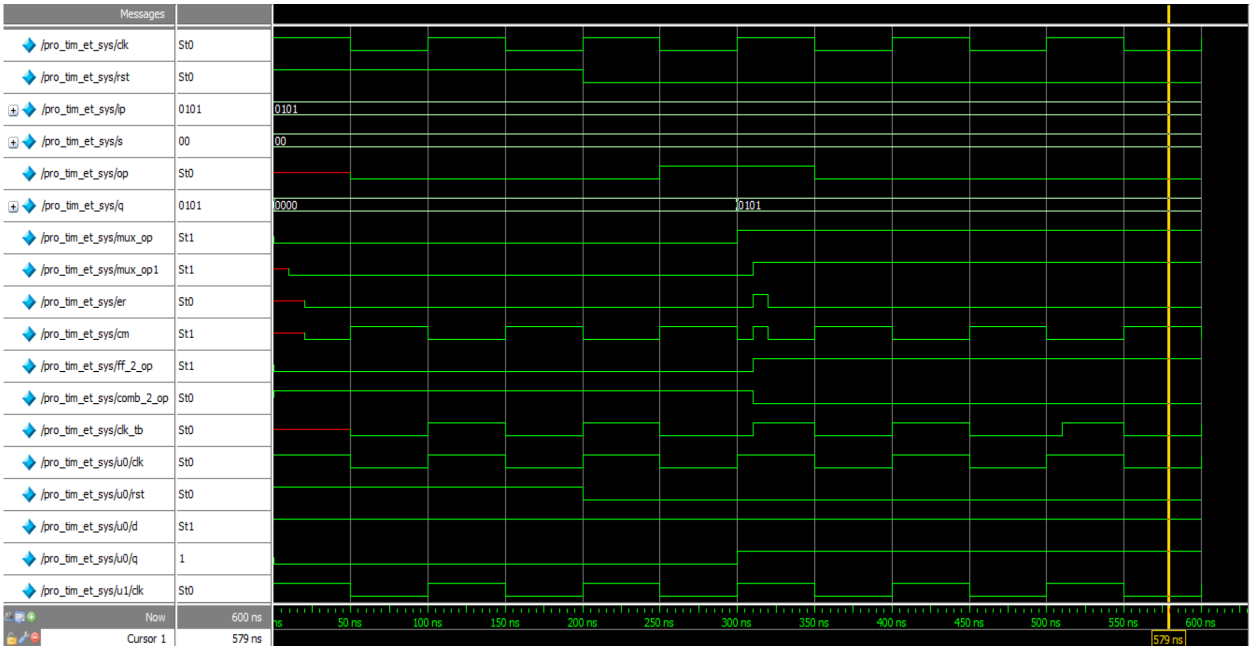
For such designs, MATLAB provides a numerical simulation toolset, while ModelSim provides tools to verify the hardware implementation & timing characteristics of the design. ModelSim eases the process of finding design defects with an intelligently engineered debug environment.

Simulation result of Transition detector



Messages	
/trans_det/clk	St0
/trans_det/ip	St0
/trans_det/er	St0
/trans_det/cm	St1
/trans_det/n1	St1
/trans_det/d1	St1
/trans_det/a1	St0
/trans_det/n2	St0
/trans_det/a2	St0
Now	700 ns
Cursor 1	676 ns

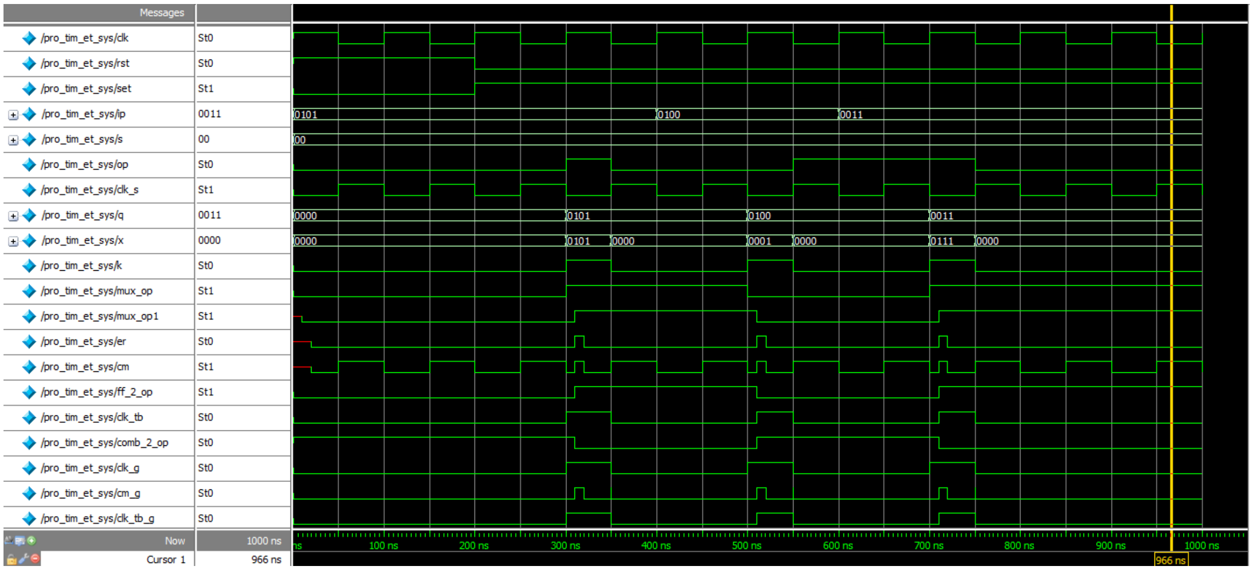
Simulation result of timing error tolerant system without clock gating



Messages	
/pro_tim_et_sys/clk	St0
/pro_tim_et_sys/rst	St0
/pro_tim_et_sys/ip	0101
/pro_tim_et_sys/s	00
/pro_tim_et_sys/op	St0
/pro_tim_et_sys/q	0101
/pro_tim_et_sys/mux_op	St1
/pro_tim_et_sys/mux_op1	St1
/pro_tim_et_sys/er	St0
/pro_tim_et_sys/cm	St1
/pro_tim_et_sys/ff_2_op	St1
/pro_tim_et_sys/comb_2_op	St0
/pro_tim_et_sys/clk_tb	St0
/pro_tim_et_sys/u0/clk	St0
/pro_tim_et_sys/u0/rst	St0
/pro_tim_et_sys/u0/d	St1
/pro_tim_et_sys/u0/q	1
/pro_tim_et_sys/u1/clk	St0
Now	600 ns
Cursor 1	579 ns

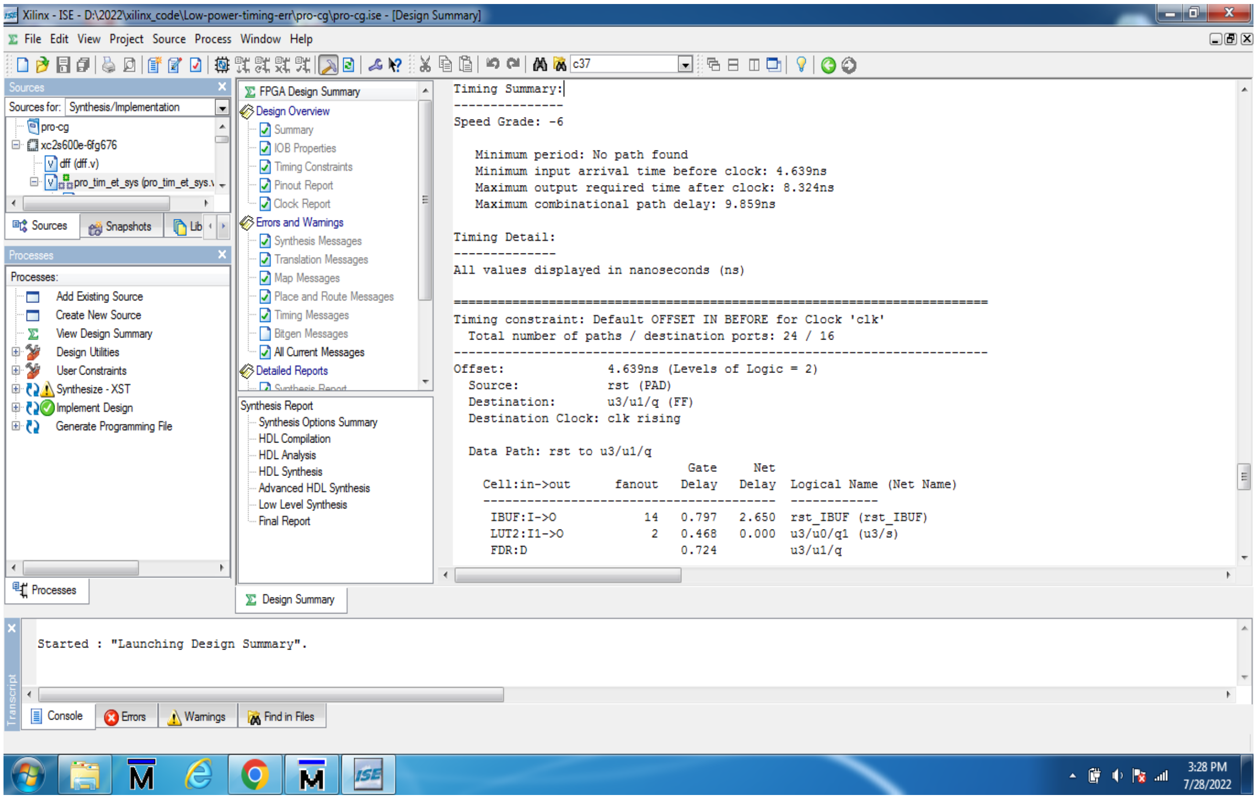
Simulation result of timing error tolerant system with clock gating

\



Messages	
/pro_tim_et_sys/clk	St0
/pro_tim_et_sys/rst	St0
/pro_tim_et_sys/set	St1
/pro_tim_et_sys/ip	0011
/pro_tim_et_sys/s	00
/pro_tim_et_sys/op	St0
/pro_tim_et_sys/clk_s	St1
/pro_tim_et_sys/q	0011
/pro_tim_et_sys/x	0000
/pro_tim_et_sys/k	St0
/pro_tim_et_sys/mux_op	St1
/pro_tim_et_sys/mux_op1	St1
/pro_tim_et_sys/er	St0
/pro_tim_et_sys/cm	St1
/pro_tim_et_sys/ff_2_op	St1
/pro_tim_et_sys/clk_tb	St0
/pro_tim_et_sys/comb_2_op	St0
/pro_tim_et_sys/clk_g	St0
/pro_tim_et_sys/cm_g	St0
/pro_tim_et_sys/clk_tb_g	St0
Now	1000 ns
Cursor 1	966 ns

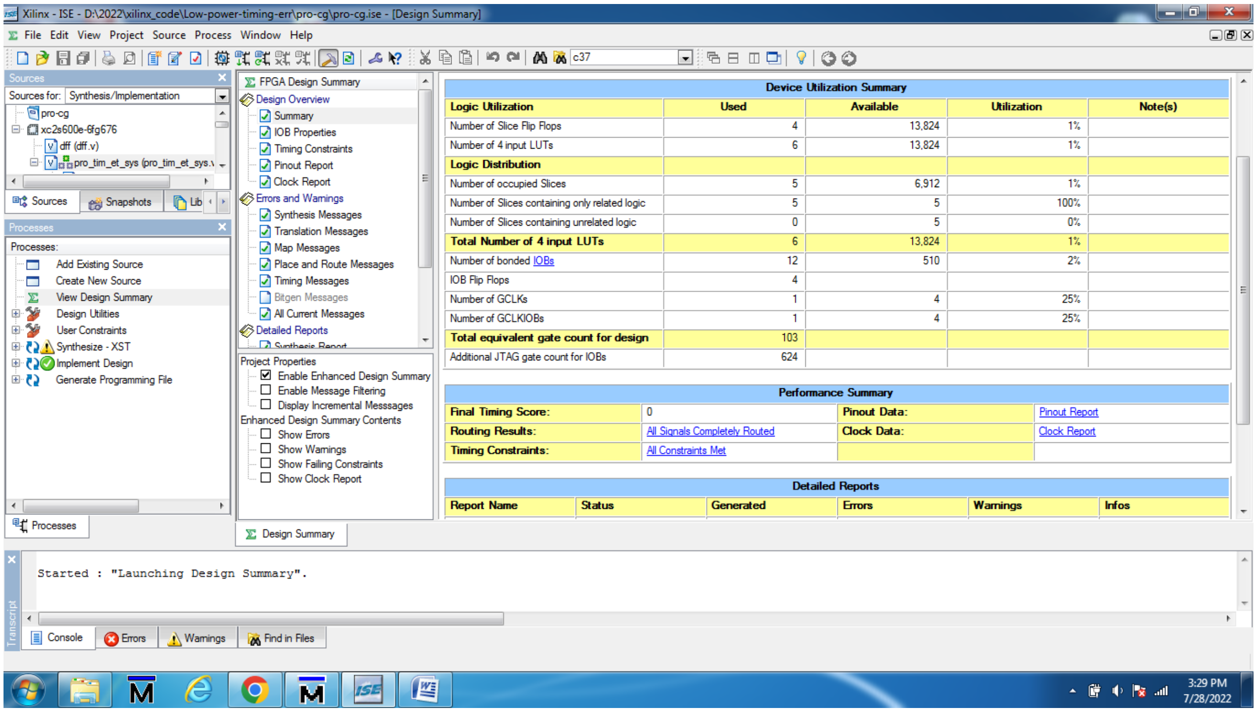
Delay report of Proposed method



```
Timing Summary:
---------------
Speed Grade: -6

   Minimum period: No path found
   Minimum input arrival time before clock: 4.639ns
   Maximum output required time after clock: 8.324ns
   Maximum combinational path delay: 9.859ns

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 24 / 16
-------------------------------------------------------------------------
Offset:              4.639ns (Levels of Logic = 2)
  Source:            rst (PAD)
  Destination:       u3/u1/q (FF)
  Destination Clock: clk rising

  Data Path: rst to u3/u1/q
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O            14   0.797   2.650  rst_IBUF (rst_IBUF)
     LUT2:I1->O            2   0.468   0.000  u3/u0/q1 (u3/s)
     FDR:D                     0.724          u3/u1/q
```

```
Started : "Launching Design Summary".
```

Area report of Proposed method



Project Properties
☑ Enable Enhanced Design Summary
☐ Enable Message Filtering
☐ Display Incremental Messages
Enhanced Design Summary Contents
☐ Show Errors
☐ Show Warnings
☐ Show Failing Constraints
☐ Show Clock Report

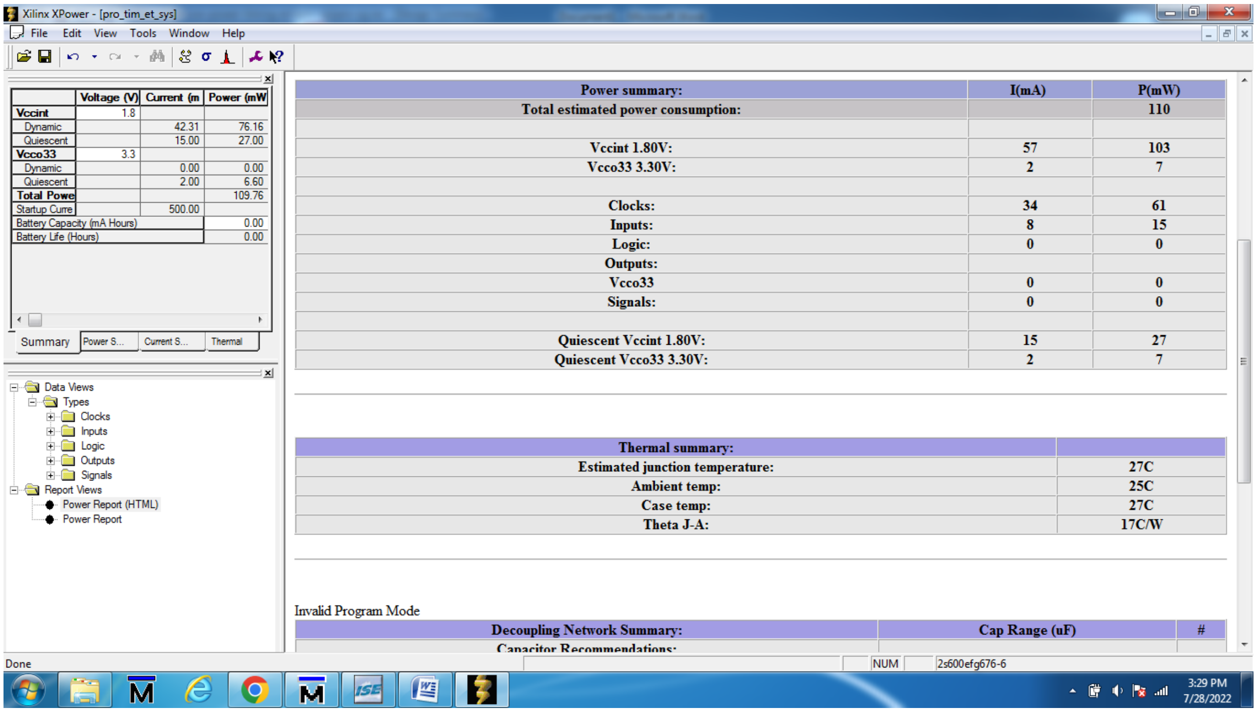
Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Flip Flops	4	13,824	1%	
Number of 4 input LUTs	6	13,824	1%	
Logic Distribution				
Number of occupied Slices	5	6,912	1%	
Number of Slices containing only related logic	5	5	100%	
Number of Slices containing unrelated logic	0	5	0%	
Total Number of 4 input LUTs	6	13,824	1%	
Number of bonded IOBs	12	510	2%	
IOB Flip Flops	4			
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	103			
Additional JTAG gate count for IOBs	624			

Performance Summary			
Final Timing Score:	0	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

Detailed Reports					
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**

```
Started : "Launching Design Summary".
```

Power report of Proposed method



	Voltage (V)	Current (m	Power (mW
Vccint	1.8		
Dynamic		42.31	76.16
Quiescent		15.00	27.00
Vcco33	3.3		
Dynamic		0.00	0.00
Quiescent		2.00	6.60
Total Powe			109.76
Startup Curre		500.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

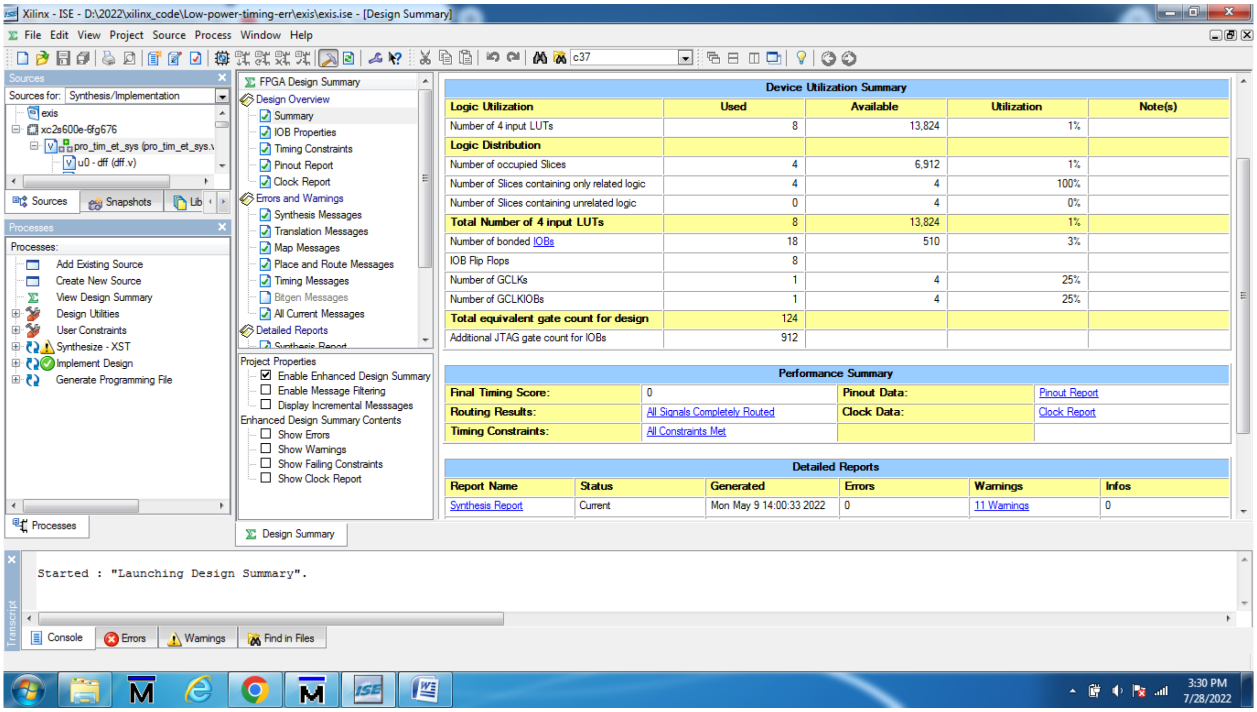
Power summary:	I(mA)	P(mW)
Total estimated power consumption:		110
Vccint 1.80V:	57	103
Vcco33 3.30V:	2	7
Clocks:	34	61
Inputs:	8	15
Logic:	0	0
Outputs:		
Vcco33	0	0
Signals:	0	0
Quiescent Vccint 1.80V:	15	27
Quiescent Vcco33 3.30V:	2	7

Thermal summary:	
Estimated junction temperature:	27C
Ambient temp:	25C
Case temp:	27C
Theta J-A:	17C/W

Invalid Program Mode

Decoupling Network Summary:	Cap Range (uF)	#
Capacitor Recommendations:		

Area report of existing method



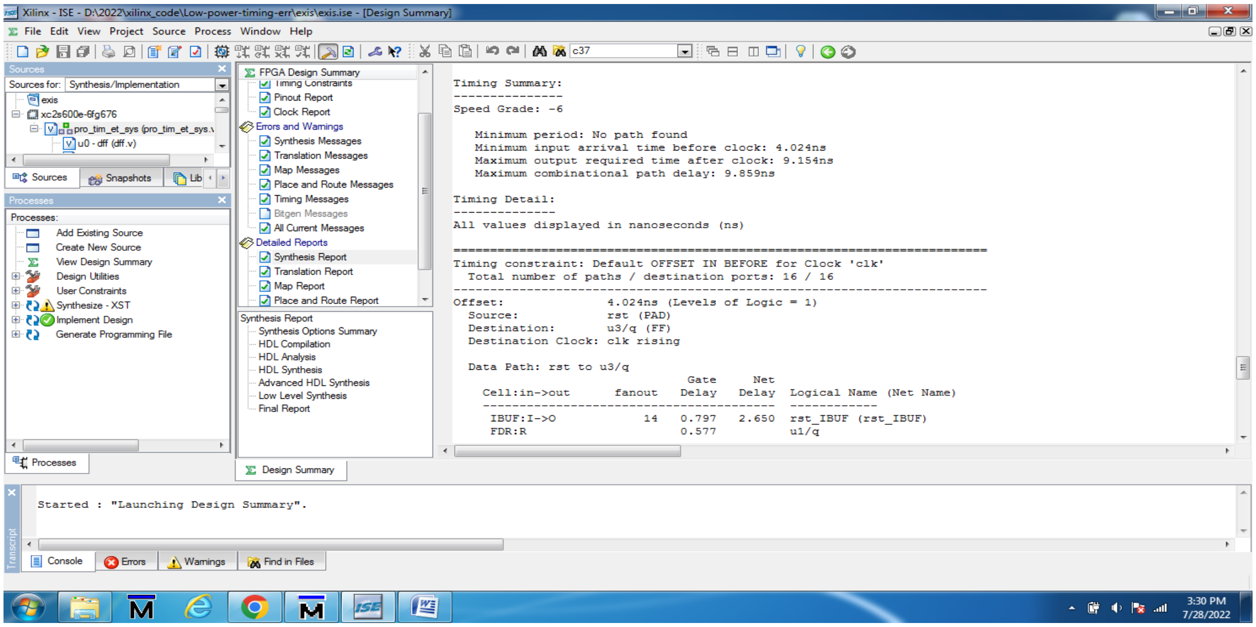
Device Utilization Summary				
Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of 4 input LUTs	8	13,824	1%	
Logic Distribution				
Number of occupied Slices	4	6,912	1%	
Number of Slices containing only related logic	4	4	100%	
Number of Slices containing unrelated logic	0	4	0%	
Total Number of 4 input LUTs	8	13,824	1%	
Number of bonded IOBs	18	510	3%	
IOB Flip Flops	8			
Number of GCLKs	1	4	25%	
Number of GCLKIOBs	1	4	25%	
Total equivalent gate count for design	124			
Additional JTAG gate count for IOBs	912			

Performance Summary			
Final Timing Score:	0	**Pinout Data:**	Pinout Report
Routing Results:	All Signals Completely Routed	**Clock Data:**	Clock Report
Timing Constraints:	All Constraints Met		

Detailed Reports					
Report Name	**Status**	**Generated**	**Errors**	**Warnings**	**Infos**
Synthesis Report	Current	Mon May 9 14:00:33 2022	0	11 Warnings	0

```
Started : "Launching Design Summary".
```

Delay report of existing method



```
Timing Summary:
---------------
Speed Grade: -6

   Minimum period: No path found
   Minimum input arrival time before clock: 4.024ns
   Maximum output required time after clock: 9.154ns
   Maximum combinational path delay: 9.859ns

Timing Detail:
--------------
All values displayed in nanoseconds (ns)

=========================================================================
Timing constraint: Default OFFSET IN BEFORE for Clock 'clk'
  Total number of paths / destination ports: 16 / 16
-------------------------------------------------------------------------
Offset:              4.024ns (Levels of Logic = 1)
  Source:            rst (PAD)
  Destination:       u3/q (FF)
  Destination Clock: clk rising

  Data Path: rst to u3/q
                                Gate     Net
    Cell:in->out      fanout   Delay   Delay  Logical Name (Net Name)
    ----------------------------------------  ------------
     IBUF:I->O            14   0.797   2.650  rst_IBUF (rst_IBUF)
     FDR:R                     0.577          u1/q
```

```
Started : "Launching Design Summary".
```

CONCLUSION

A timing-error-tolerant method that can correct a timing error immediately with a compact circuit structure is proposed. In the critical path, the abnormal data transition after the edge of the clock can be detected and corrected by controlling the transparent window of the clock. The timing error is corrected directly through a minimum number of logics. Furthermore, our time-borrowing technique that deals with the successive-stage error is introduced. If the timing error occurs in the second stage successively, modified CLK maintains the transparent window during enough period of time for timing-error tolerance without changing system CLK. Further to eliminate unwanted transition look ahead clock gating is incorporated into the error tolerant design. The proposed circuit was extensively simulated with CMOS circuit design to address PVT variations. In terms of hardware overhead and performance, comparisons with other conventional methods were also conducted. Compared with existing systems, it was proven that the proposed system has a better performance with a lower hardware overhead in large size of circuits. Since the whole circuit consists of compact logics, it results in such low hardware overhead compared with existing timing-error mitigating systems that can recover the error instantly.

REFERENCES

[1] J. W. McPherson, “Reliability challenges for 45 nm and beyond,” in Proc. 43rd ACM/IEEE Design Automat. Conf., Jul. 2006, pp. 176–181.

[2] S. Mitra, N. Seifert, M. Zhang, Q. Shi, and K. S. Kim, “Robust system design with built-in soft-error resilience,” Computer, vol. 38, no. 2, pp. 43–52, Feb. 2005.

[3] S. P. Park, K. Roy, and K. Kang, “Reliability implications of bias temperature instability in digital ICs,” IEEE Des. Test Comput., vol. 23, no. 6, pp. 8–17, Nov./Dec. 2009.

[4] M. Seok, G. Chen, S. Hanson, M. Wieckowski, D. Blaauw, and D. Sylvester, “CAS-FEST 2010: Mitigating variability in near-threshold computing,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 1, no. 1, pp. 42–49, Mar. 2011.

[5] M. Agarwal, B. C. Paul, M. Zhang, and S. Mitra, “Circuit failure prediction and its application to transistor aging,” in Proc. 25th IEEE VLSI Test Symmposium (VTS), May 2007, pp. 277–284.

[6] M. Agarwal et al., “Optimized circuit failure prediction for aging: Practicality and promise,” in Proc. IEEE Int. Test Conf., Oct. 2008, pp. 1–10.

[7] M. Nicolaidis, “Time redundancy based soft-error tolerance to rescue nanometer technologies,” in Proc. 17th IEEE VLSI Test Symp., Apr. 1999, pp. 86–94.

[8] L. Anghel and M. Nicolaidis, “Cost reduction and evaluation of a temporary faults detecting technique,” in Proc. Design, Automat. TestEur. Conf. Exhib., Mar. 2000, pp. 591–598.

[9] S. Valadimas, A. Floros, Y. Tsiatouhas, A. Arapoyanni, and X. Kavousianos, “The time dilation technique for timing error tolerance,” IEEE Trans. Comput., vol. 63, no. 5, pp. 1277–1286, May 2014.

[10] S. Valadimas, Y. Tsiatouhas, and A. Arapoyanni, “Timing error tolerance in small core designs for SoC applications,” IEEE Trans. Comput., vol. 65, no. 2, pp. 654–663, Feb. 2016.

[11] M. R. Choudhury, V. Chandra, R. C. 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, Feb. 2014.

[12] 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.

[13] M. Ahmadi, B. Alizadeh, and B. Forouzandeh, “A timing error mitigation technique for high performance designs,” in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, Jul. 2015, pp. 428–433.

[14] C. Metra, R. Degiampietro, M. Favalli, and B. Ricco, “Concurrent detection and diagnosis scheme for transient, delay and crosstalk faults,”in Proc. IEEE Int. On-Line Test. Workshop, Apr. 1999, pp. 66–70.

[15] Y. Tsiatouhas and T. Haniotakis, “A zero aliasing built-in self test technique for delay fault testing,” in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Syst. (EFT), Nov. 1999, pp. 95–100.

[16] K. Kang, S. P. Park, K. Kim, and K. Roy, “On-chip variability sensor using phase-locked loop for detecting and correcting parametric timing failures,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 2, pp. 270–280, Feb. 2010.

[17] D. Bull, S. Das, K. Shivashankar, G. S. Dasika, K. Flautner, and D. T. Blaauw, “A power-efficient 32 bit ARM processor using timing error detection and correction for transient-error

tolerance and adaptation to PVT variation," IEEE J. Solid-State Circuits, vol. 46, no. 1, pp. 18–31, Jan. 2011.

[18] M. Omana, D. Rossi, N. Bosio, and C. Metra, "Low cost NBTI degradation detection and masking approaches," IEEE Trans. Comput., vol. 62, no. 3, pp. 496–509, Mar. 2013.

[19] K. A. Bowman et al., "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," IEEE J. Solid-State Circuits, vol. 44, no. 1, pp. 49–63, Jan. 2009.

[20] J. S. Wang, "Dynamic voltage scaling system having time borrowing and local boosting capability," U.S. Patent 8 933 726, Jun. 26, 2014.

[21] W. Shan, X. Shang, X. Wan, H. Cai, C. Zhang, and J. Yang, "A widevoltage- range half-path timing error-detection system with a 9-transistor transition-detector in 40-nm CMOS," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 6, pp. 2288–2297, Jun. 2019.

[22] Rajagopal, R., Karthick, R., Meenalochini, P., & Kalaichelvi, T. (2023). Deep Convolutional Spiking Neural Network optimized with Arithmetic optimization algorithm for lung disease detection using chest X-ray images. Biomedical Signal Processing and Control, 79, 104197.

[23] Srinivas, T. Aditya Sai, B. Ravindra Babu, Miskir Solomon Tsige, R. Rajagopal, S. Devi, and Subrata Chowdhury. "Effective implementation of the Prototype of a digital stethoscope using a Smartphone." In 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), pp. 1-8. IEEE, 2022.

[24] Rajagopal, R., M. Karthik, M. Soni, Narayan Krishan Vyas, S. Hemavathi, and M. R. Arun. "Monitoring the high-speed engine application using ferro magnetic system." Materials Today: Proceedings (2022).

[25] Babu, P. Ramesh, Vemuri Kusuma Priya, N. Drawin, R. Thiagarajan, and R. Krishnamoorthy. "Enhanced Hybrid Resource Scheduler for an Institution Employing NB-RR Scheduling." In 2022 8th International Conference on Smart Structures and Systems (ICSSS), pp. 1-6. IEEE, 2022.

[26] Jose, S. Edwin, R. Lal Raja Singh, and R. Rajagopal. "Automatic and real time classification of power quality disturbance using statistical moments." AIP Conference Proceedings. Vol. 2327. No. 1. AIP Publishing LLC, 2021.

[27] Rajagopal, R., and S. Edwin Jose. "An Efficient Framework for Locating Stroke in Brain MRI Images Using Radon Transform and Convolutional Neural Networks." In Next Generation of Internet of Things, pp. 385-395. Springer, Singapore, 2021.

[28] Rajagopal, R., and P. Subbaiah. "A survey on liver tumor detection and segmentation methods." ARPN Journal of Engineering and Applied Sciences 10, no. 6 (2015): 2681-2685

[29] Rajagopal, R., & Subbiah, P. (2014). Computer aided detection of liver tumor using SVM classifier. International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 3(6), 10170-7.

[30] Rajagopal, R. (2019). Glioma brain tumor detection and segmentation using weighting random forest classifier with optimized ant colony features. International Journal of imaging systems and technology, 29(3), 353-359.