Timing violation reduction in the fpga prototyped design using failed path fixes and time borrowing techniques

Salahuddin Savugathali, Muslim Mustapa, Mohammed Sharazel Razali, Fazrul Faiz Zakaria
School of Computer and Communication Engineering, Univeristy Malaysia Perlis, Malaysia

Article Info

Article history:

Received Oct 1, 2018
Revised Dec 10, 2018
Accepted Jan 25, 2019

Keywords:

Failed path fixes on latches
FPGA
FPGA prototyped design
Gated clock conversion
Time borrowing
Timing violation

ABSTRACT

A fascinating property of a latch-based design is that the combinational path delay is allowed to be longer than the clock cycle as it can borrow time from the shorter paths in the subsequent logic states. Time borrowing technique is a common method used to satisfy timing violation in an FPGA prototyped design. The purpose of this paper is to review the current methodology involved in SoC design prototyping using a Synopsys Protocompiler and HAPS-80 platform and propose an approach by fixing the failed path in a latch due to the gated clock conversion (GCC) process during the synthesis stage which could lead to the timing violation. Two techniques are applied in this paper namely time borrowing technique and our proposed technique, Failed Path Fixes to reduce the timing violation in the FPGA prototyped design. The result shows that the applied techniques are able to close the timing violation in the design with an average of 90% improvement.

Corresponding Author:

Salahuddin Savugathali,
School of Computer and Communication Engineering,
Univeristy Malaysia Perlis, Malaysia.
Email: salahuddin@studentmail.unimap.edu.my

1. INTRODUCTION

Nowadays, the SoC design has become one of the mainstream in electronic design among the semiconductor industry [1]. The advance of technology and high competition rates between industries have indirectly led the electronic circuit developer to propose a more complex and large SoC design.

Due to the large size and high complexity of the modern SoC design, it caused a bottleneck in the validation stage. The main cause for the bottleneck is longer time taken to validate the design functionalities [1] and high-cost need to be invested by the semiconductor industry to fix the error [2]. One of the techniques used in the semiconductor industry to eliminate the challenges in the validation stage is by having an FPGA prototyping technique which is able to solve the bottleneck problem. The main idea of this technique is to prototype a SoC design on an FPGA so that, the design can be validated in a pre-silicon stage [2]. Figure 1 shows an FPGA prototyping flow using the Synopsys Protocompiler tool.

SoC design must be reworked to meet the FPGA based requirement due to the differences in architecture and memory modules before mapping into an FPGA. To obtain a prototyped design ready, few stages are visited on the prototyping flow which begins with a compilation of the RTL and subsequently through the pre-map, map and finally places and route stage on Vivado tool. The generated bit configuration file at the end of prototyping flow will be configured into the hardware before executing the validation test.

In FPGA prototyping flow, design performance is one of the main challenges to be faced in the semiconductor industry. One of the design performance challenges focused on this research is the negative slack generated during design prototyping which could lead to the FPGA prototyped design’s timing requirement violation [3].

Timing check is divided into two; setup and hold timing. The present of negative slacks which is less than an ideal value of 0ns could violate the functionalities of the FPGA prototyped design. It is necessary to verify that FPGA prototype design meets the timing requirement and able to work with the intended clock speed. Therefore, in this paper, a root cause for the violation of timing requirement in the design will be analyzed before applying the time borrowing technique and our proposed technique Failed Path Fixes to reduce the timing violation as modern SoC timing closure critically depends upon the effectiveness of the timing fixes and its implementation [4].

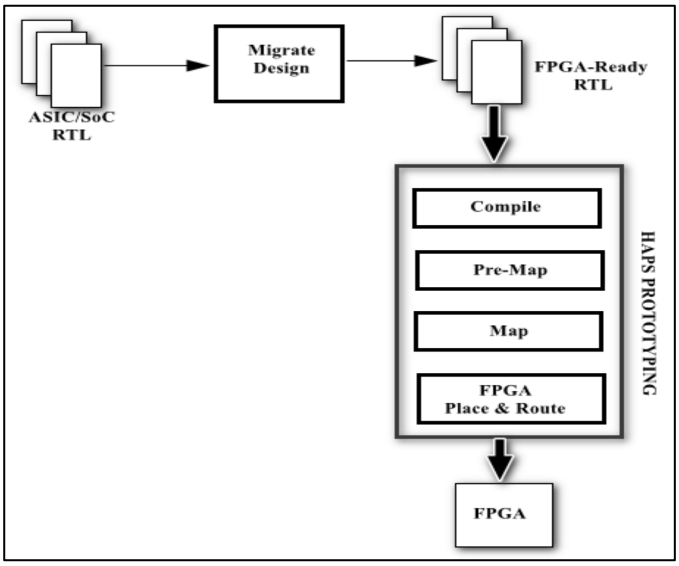


Figure 1. FPGA Prototyping Flow

This paper is organized as follow; Section 2 describes an existing technique to close the timing requirement violation.While Section 3 discusses the step implemented in this research from the analyzation of the failed design until the implementation of the approached techniques. Section 4 will summarize the approached techniques with the percentage of the improvement.

2. LITERATURE REVIEW

There are plenty of challenges had been highlighted in the previous reserach whilst in this research, reducing the timing violation will be focused on the single FPGA prototyping. In this section, an overview for timing analysis of latch-based circuit will be discussed. A basic fundamental of timing violation, the root cause for the failure of the timing to meet its specification for FPGA prototyping and also the techniques available and implied by previous researched will be discussed further.

2.1. Setup Timing Check

Setup time is a minimum amount of time that a data should hold stable before the capture event to ensure the data are reliably sampled by the clock. This check is approached in order to make sure the propagation signal from one point to another point is not taking more than the required time.

2.2. Hold Timing Check

Hold time is a minimum amount of time the data signal should hold stable after the capture event so that the data are reliably sampled by the clock. Purpose of this check is to assure that the data will not change while the sequential cell is in the process of capturing it.

Figure 1 1 concludes both setup and hold times of a latch are measured relative to the trailing edge of the clock. Therefore, the longest path a1 must arrive at next latch L2 before setup time and the shortest path a2 must reach next latch L3 after hold time in order to achieve the timing requirement.

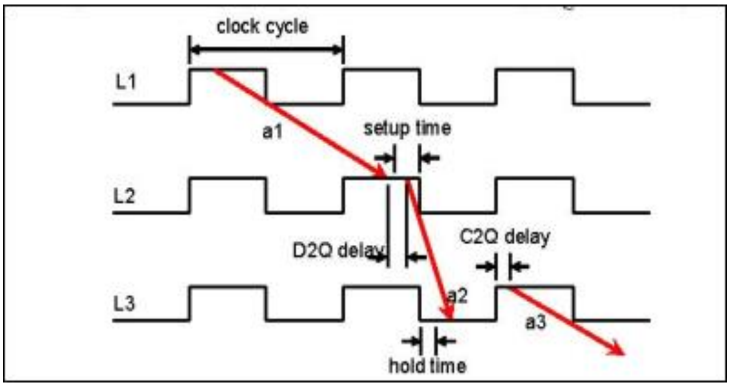


Figure 1. Latches timing diagram [5]

2.3. Gated Clock Conversion

Clock gating is an effective technique used mostly in the SoC design to save dynamic power, where the peculiar clock will be shut down when it is not required [6]. Clock gating can be divided into two approaches, which one of them is RTL clock gating. RTL clock gating is used for optimization and improving the efficiency of the SoC design. RTL clock gating is designed into the SoC architecture as a part of RTL functionality where all the functionalities and clock for the passive block will be blocked. As an effect of that approach, a block of logic is not required for switching for many cycles then substantial dynamic power is preserved [7].

Modern FPGA synthesis tools like Synopsys protocompiler perform this process automatically without having the user to change the RTL [8]. However, the synthesys tool is not able to perform gated clock conversion for complex gating logic and gated clock are derived based on multiple clocks [9]. Therefore, a modified netlist after gated clock conversion will be compared against the original RTL netlist to find the root cause of the timing violation.

2.4. Previous Research

Many researches have been devoted previously to address the timing violation problem in an FPGA prototyping design. Work in [10] has been focused on the finding an optimal time borrowing solution by formulating time borrowing problem as a linear programming problem. While research in [11] presenting a dynamic timing control techniques to prevent timing failure by applying time borrowing and elastic clock stretching approaches. In [12], the problem has been identified, where a frequency is being capped by the constrained path between flip-flop stages in a digital circuit. This is due to guard-bonding used by the designer to prevent the timing violation. In order to tackle this problem, two strategies have been proposed in this research which is Time Borrowing Flip Flop (TBFF) which is also proposed by [11] in different research on optimizing the timing violation and Alternative Path Activation (APA). In [13], the researcher has proposed hybrid time borrowing techniques by utilizing a DFFC unitedly with dynamic clock stretching techniques which is to overcome the drawback from the previously proposed techniques in [14].

Therefore, in this research, to reduce the timing violation in FPGA prototyped design, a time borrowing techniques will be used. In addition, since the gated clock conversion is the main root for the existence of the timing violation in our design, then it will be optimized by setting an addition latches created by the protocompiler tool as a false path through the design constraints.

3. RESEARCH MEHODOLOGY

In this section, all the steps taken in order to reduce the timing violation in an FPGA prototyped design will be discussed. Two designs have been used in this research to verify the proposed techniques which are design_1 and design_2 with a different number of logic blocks. Figure 2 shows the flow chart of impelementation step to reduce the timing violation in a prototyped design. A successful mapped design from the protocompiler tool will be exported to the Vivado tool for place and route stage.

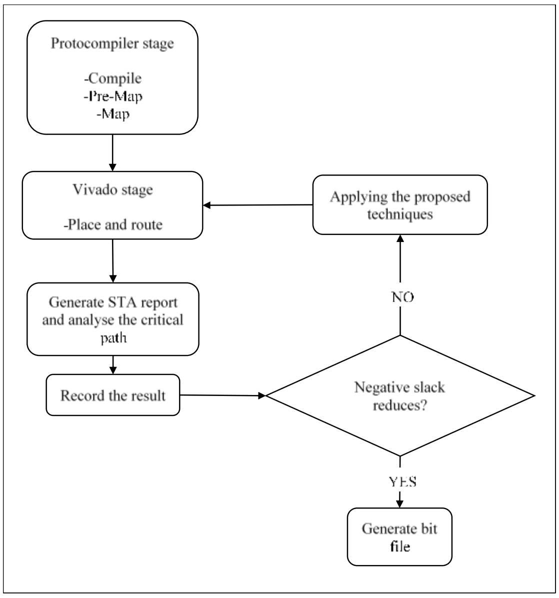
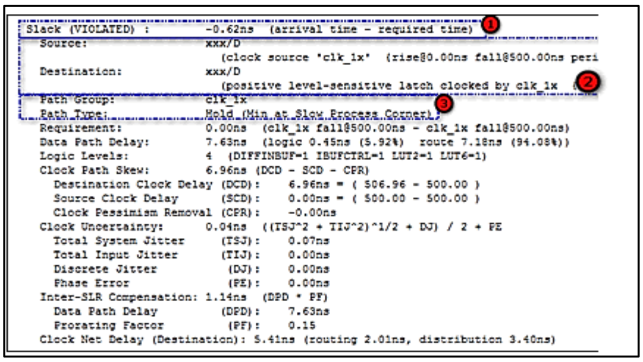


Figure 2. Flowchart of the implementation step

3.1. Timing Analysis

Timing is a major consideration in the synthesis and physical implementation of synchronous digital circuits. Static timing analysis is a method of validating the timing performance of a design by checking all possible paths for timing violations under worst-case conditions. A Xilinx Vivado Integrated Design Environment (IDE) tools provide a timing report based on Static Timing Analysis (STA) techniques in the completion of place and route stage [15]. Analyzation and debugging the timing issues can be done based on the timing summary report generated by the tool since enough information and overview of all timing checks have been reported. The areas covered in the timing summary report is setup area (Max Delay Analysis) and hold area (Min delay Analysis). Setup area of the timing summary will be reporting the worst negative slack (WNS) and total negative slack (TNS) where WNS represent the worst slack of all the timing paths for max delay analysis while TNS represent the sum of all WNS violation. For hold area, the worst hold slack (WHS) and total hold slack (THS) will be reported where WHS represent the worst slack of all timing path for min delay analysis while THS represent the sum of all WHS violation. The negative value of TNS and WHS correspondents to a violation exists in the design and positive value represent all timing constraint is met. Figure 4 represents an example of timing report generated by Vivado tool at the end of the place and route process.



```
Slack (VIOLATED) :        -0.62ns  (arrival time - required time)
  Source:                 xxx/D
                            (clock source 'clk_1x'  (rise@0.00ns fall@500.00ns peri
  Destination:            xxx/D
                            (positive level-sensitive latch clocked by clk_1x  (
  Path Group:             clk_1x
  Path Type:              Hold (Min at Slow Process Corner)
  Requirement:            0.00ns  (clk_1x fall@500.00ns - clk_1x fall@500.00ns)
  Data Path Delay:        7.63ns  (logic 0.45ns (5.92%)  route 7.18ns (94.08%))
  Logic Levels:           4  (DIFFINBUF=1 IBUFCTRL=1 LUT2=1 LUT6=1)
  Clock Path Skew:        6.96ns (DCD - SCD - CPR)
    Destination Clock Delay (DCD):    6.96ns = ( 506.96 - 500.00 )
    Source Clock Delay      (SCD):    0.00ns = ( 500.00 - 500.00 )
    Clock Pessimism Removal (CPR):    -0.00ns
  Clock Uncertainty:      0.04ns  ((TSJ^2 + TIJ^2)^1/2 + DJ) / 2 + PE
    Total System Jitter     (TSJ):    0.07ns
    Total Input Jitter      (TIJ):    0.00ns
    Discrete Jitter          (DJ):    0.00ns
    Phase Error              (PE):    0.00ns
  Inter-SLR Compensation: 1.14ns  (DPD * PF)
    Data Path Delay         (DPD):    7.63ns
    Prorating Factor         (PF):    0.15
  Clock Net Delay (Destination): 5.41ns (routing 2.01ns, distribution 3.40ns)
```

Figure 3. Violation path from the generated report

In the above report, detailed information about the timing violation caused by the transition of data from one path to another path can be obtained. Referring to Figure 4, box 1 and 2, a violated path has been reported from xxx/D to destination xxx/D with the negative slack of -0.62ns. While box 3 reporting a type of the negative slack which is hold type. Based on the information obtained from the timing summary report an optimization approaches will be taken to reduce the timing violation.

3.2. Time Borrowing Technique

Time borrowing technique is an approach to relaxing the tight timing requirement by allowing a failed path to borrow time from the next successive path. Both designs used in this research is a latch-based design, therefore borrowing time from one latch to the next can loosen the timing requirement for a particular path. Time borrowing technique has been applied to the violated design through the Vivado constraints. By referring to the example of timing report in Figure 4, timing violation is caused by the clk_1x with a period of 1000ns. Therefore, as Vivado tool allows time borrowing technique to be applied through the input constraints during place and route stage, the maximum value for time borrowing is set to 500ns which is half of the clock period of clk_1x. The circuit needs to be re-executed again through place and route stage once the input constraint is updated in order to replicate the changes in the constraints of the circuit. In order to diversify the obtained result, a few sets of experiments have been conducted based on these two types of circuit, deisgn_1 and design_2 with different design frequency which is 1MHz and 2MHz. Different design frequency is used to compare the effectiveness of the proposed time borrowing technique that satisfies the timing requirement in a circuit.

3.3. Failed Path Fixes Technique

Based on the result obtained for the timing path by implementing the time borrowing technique, a negative slack reduction average in the circuit is only around 30% which does not satisfy the timing requirement. A further analysis has been done on the time borrowed path and is identified to have an unwanted logic group. Therefore, a time was borrowed repeatedly until the timing for a signal to travel in a specific path exceeds clock period. In order to find the root cause for the above issue, a comparison has been made between mapped schematics against the RTL code. A problem has been noticed as after the synthesis process through protocompiler tool, a master/slave latch structure with a few XORs around the latch exists, while in the original RTL design only have one latch. In the altered netlist by the protocompiler tool, the master latch operates on the high phase of the clock and slave latch operates on the low phase of the clock, while in the original RTL codes, all latches only operate on a low phase of the clock. Another problem identified during the analyzation is, the gated clock enable signal is connected to the enable signal of the slave latch through the gated clock conversion logic. The comparison between the latches in the original netlist and the modified netlist is shown in

Figure 4 and Figure 5. As per found by [16], a Vivado will mimic an always transparent latch whenever the clock is gated. Consequences of this action, as the clock is enabled, only master latch will be working while the slave latch became irrelevant since it was XOR'd twice on the data path. Meanwhile, master and slave latches are opened on opposite phase of the clock when the clock was disabled in order to ensure that the output of structure followed the input.

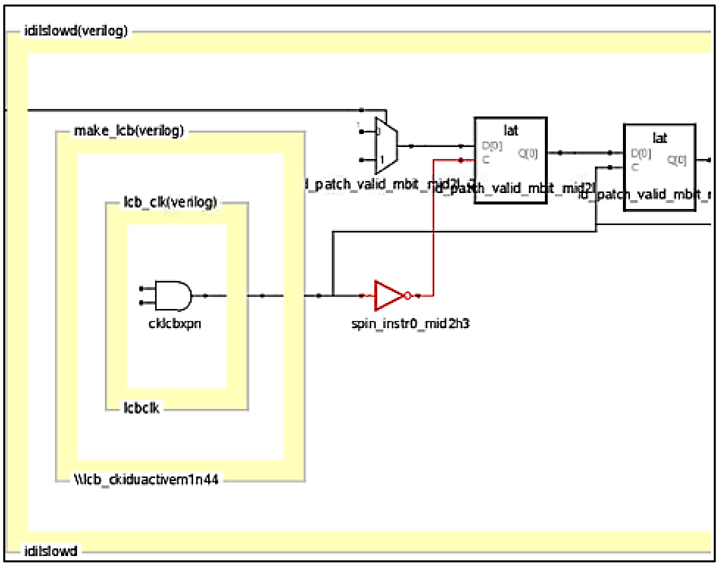


Figure 4. Original latches in netlist [16]

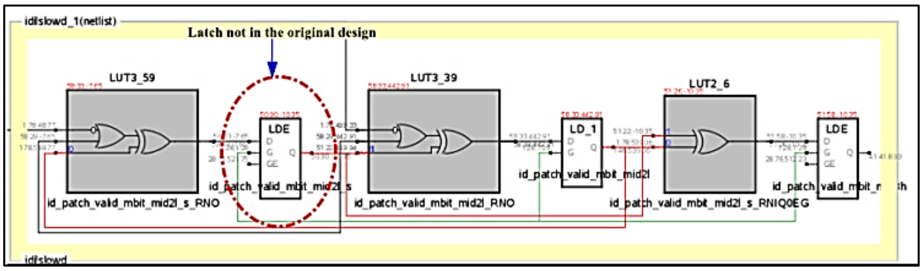

idlslowd_1(netlist)
Latch not in the original design
LUT3_59
LDE
LUT3_39
LD_1
LUT2_6
LDE
id_patch_valid_mbit_mid2l_s_RNO
id_patch_valid_mbit_mid2l_s
id_patch_valid_mbit_mid2l_RNO
id_patch_valid_mbit_mid2l
id_patch_valid_mbit_mid2l_s_RNIQ0EG
idlslowd

Figure 5. Latches on modified netlist after GCC[16]

Figure 6 shows differences on schematic before and after gated clock conversion on a flip-flop and latches in Xilinx Vivado implementation. Clock gate module on the clock path for flip-flop can be converted either to data path or to enable pin regardless of the clock edges. However, for latches, only high phase latches able to be converted, while for low phase latches, an always transparent latch is required if the clock is gated to be always low, which is not able to be implemented by Xilinx LDCE instances.

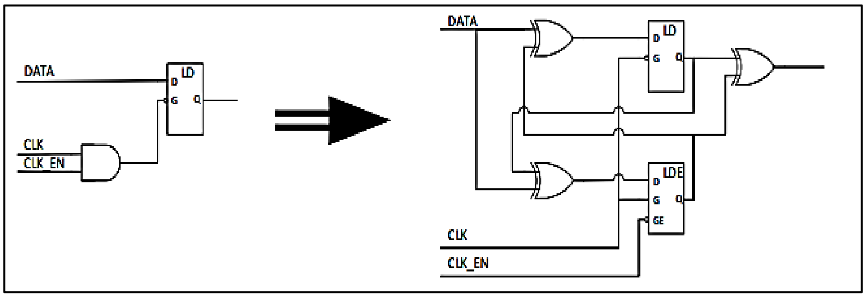


Figure 6. Comparison between RTL codes and GCC converted schematics

Figure 7 represent a conversion schematic where the protocompiler implements two different structures for this always transparent behavior which is master/slave latch as shown in Figure 5 and multiplexer (mux) at the output of the original latch controlled by the clock gate enable signal. Therefore, from a static timing perspective, both of these implementations create paths between the latches on the same clock phase which is causing a timing problem in a design.

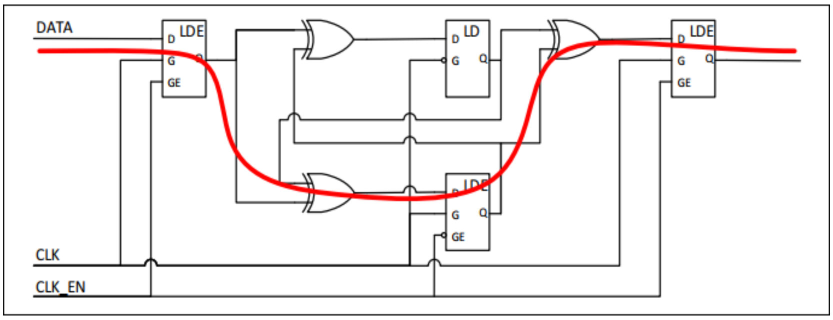
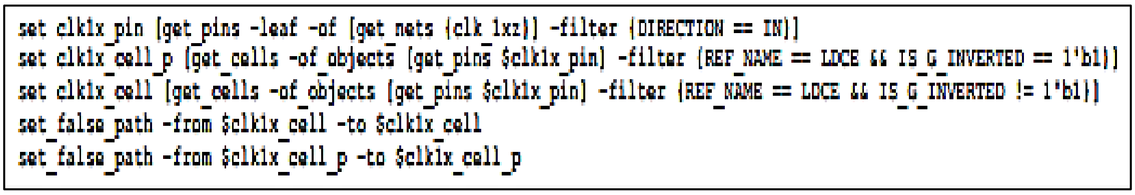


Figure 7. Timing path through 3 latches

Based on the analysis made, all the timing path were created by a low clock phase latches on a clock network which is when gated, would stay high. Therefore, all path through transparent latch were identified while the clock is gated and marked as a false path since slave latch in a master/slave structure was only used when the clock is gated.

Figure 8 shows the Vivado constraint that has been added to clock path clk_1 to solve path through the transparent latch which will reduce the negative slack as shown in Figure 3.



```
set clk1x_pin [get_pins -leaf -of [get_nets {clk_1xz}] -filter {DIRECTION == IN}]
set clk1x_cell_p [get_cells -of_objects [get_pins $clk1x_pin] -filter {REF_NAME == LDCE && IS_G_INVERTED == 1'b1}]
set clk1x_cell [get_cells -of_objects [get_pins $clk1x_pin] -filter {REF_NAME == LDCE && IS_G_INVERTED != 1'b1}]
set_false_path -from $clk1x_cell -to $clk1x_cell
set_false_path -from $clk1x_cell_p -to $clk1x_cell_p
```

Figure 8. additional Vivado constraint

These techniques will be applied on both design_1 and design_2 respectively, where a constraint file will be prepared according to the clock that causing the timing issues in each of the design. The entire clock in the same phase will be identified and marked as a false path as shown in the first and last two lines respectively in

Figure 8 the result obtained after implementation of Time Borrowing and Failed Path Fixes techniques in the design will be compared to an original timing violation without implementing any techniques to measure the percentage of the improvement of the timing violation.

4. RESULT

In this research, the proposed Time Borrowing techniques are implemented in the synthesis level using the tool without any modification in the RTL level of the SoC design. In the meantime, failed path in the latches are also fixed to reduce the timing violation of the design. While all the proposed techniques and the implementation parameters were covered in the previous section, this section presents the analysis and discussion of numerical results gained from the implementation of the proposed techniques. Each design will be run with different FPGA clock frequency that are 1 MHz and 2MHz. These two frequency fall in the range of a workable frequency for FPGA prototyped design. In addition, the percentage of the Total Reduction (TR) is defined as in the (1), where Previous Slack (PS) and New Slack (NS) represent a slack before and after the implementation of the approached techniques respectively.

$$TR\ (\%) = \frac{PS-NS}{PS} \times 100 \qquad (1)$$

4.1. Timing Violation Result

The performance of the proposed techniques to reduce the timing violation of an FPGA prototyped model will be measured empirically and compared in the percentage of improvement achieved with the design without any implementation of the techniques.

Table 1 represents a violation in a design before and after implementation of the proposed techniques which is using Time Borrowing and Failed Path Fixes in a latch in a design_1 with an FPGA clock frequency of 1MHz. While Table 2 represents a result obtained for the same experiment on design_1 with a different clock frequency which is 2MHz. Table 3 and Table 4 represents a result obtained for the experiment executed on the design_2 with a clock frequency of 1 Mhz and 2 MHz respectively.

The obtained result is extracted from the timing summary report which is generated by the Vivado tool at the end of the place and route stage based on STA techniques. As per mentioned before, with the STA techniques all path including the false path will be covered in the report. Therefore, failed path in latches able to be detected and fixed and resulting in a huge reduction in negative slack.

Table 1. Timing Violation for A *Design_1* with A Clock Frequency of 1mhz

Parameter	Timing Violation		Total Reduction
	Previous Slack (ns)	New Slack (ns)	(%)
WNS	-11.33	-0.31	97.26%
TNS	-184329.88	-1.72	99.99%
WHS	-11.70	-8.25	29.49%
THS	-139286.38	-7249.80	94.80%

Table 2. Timing Violation for A *Design_1* with A Clock Frequency of 2mhz

Parameter	Timing Violation		Total Reduction (%)
	Previous Slack (ns)	New Slack (ns)	
WNS	-24.76	-0.29	99.97%
TNS	-428248.53	-0.90	99.99%
WHS	-11.70	-8.34	28.72%
THS	-139287.91	-7224.67	94.81%

Table 3. Timing Violation for A *Design_2* with A Clock Frequency of 1mhz

Parameter	Timing Violation		Total Reduction (%)
	Previous Slack (ns)	New Slack (ns)	
WNS	-9.81	0.00	100.00%
TNS	-910029.94	0.00	100.00%
WHS	-4.19	0.02	100.48%
THS	-16.26	0.00	100.00%

Table 4. Timing Violation for A *Design_2* with A Clock Frequency of 2mhz

Parameter	Timing Violation		Total Reduction (%)
	Previous Slack (ns)	New Slack (ns)	
WNS	-12.50	-0.027	99.78%
TNS	-915614.81	-0.027	99.99%
WHS	-13.72	-0.29	97.89%
THS	-1611.78	-66.73	95.85%

4.2. Summary for Clock Frequency

A comparison has been made between two clock frequencies within a workable range for each FPGA prototyped design before implementation of the proposed techniques to identify the impact of increasing a clock frequency to the timing violation. Referring to Table 1 and Table 2, the early reading shows that timing violation rises directly proportional to an FPGA clock frequency. To ensure this condition, the same parameter is tested on a different design named as design_2 which can be referred from the Table 3 and Table 4. As the timing violation in a design increases directly proportional to the FPGA clock frequency, then a conclusion has been made from this experiment that as the speed of the design is increased the timing requirement become tighter to be met by latches in a design.

4.3. Summary on the Result for the Proposed Techniques

Since the violation in the design is increases due to the rises in clock frequency, the proposed techniques have been applied which is Time Borrowing through a constraint by at least half of the clock period and fixes in the failed path of latches for both speeds of a design for each design, design_1, and design_2. Based on the result recorded for a design_1 in Table 1 and Table 2, average more than 90% of the negative slack has been eliminated for the setup slack, worst negative slack (WNS) and total negative slack (TNS). However, challenges have been identified in reducing a hold slack, Worst Hold Slack (WHS) as only an average of 29% is reduced, but for Total Hold Slack (THS), more than 90% of negative slack is reduced. Therefore, to ensure the effectiveness of these techniques, another experiment has been done on different design, named design_2 using the same techniques. Table 3 and Table 4 represent a result obtained for design_2 with a clock frequency of 1MHz and 2 MHz respectively. Design_2 with a clock frequency of 1MHz achieve an average 100% of the timing violation elimination. While for a design_2 with 2 MHz, reduction in a timing violation also is satisfied as achieving more than 90% average.

5. CONCLUSION

In this paper, Time Borrowing technique together with our proposed technique, Failed Path Fixes has been applied to reduce the timing vioaltion in the FPGA prototyped design. Two FPGA prototyped design with different number of logic blocks have been used. A failed path in the latches has been fixed by setting it as a false path through the Vivado constraint. Both techniques applied in this paper improves the total reduction of the negative slack averaging more than 90%. This is an acceptable range in the industries.

REFERENCES

[1] X. Li, L. Hou, S. Geng, J. Wang, and H. Zhang, "The FPGA prototyping implementation of LEON3 SoC," *Proc. 2012 Int. Conf. Ind. Control Electron. Eng. ICICEE 2012*, pp. 1643–1646, 2012.

[2] Y. Abarbanel, E. Singerman, and M. Y. Vardi, "Validation of SoC Firmware-Hardware Flows," *Proc. 51st Annu. Des. Autom. Conf. Des. Autom. Conf. - DAC '14*, pp. 1–4, 2014.

[3] B. Li, N. Chen, and U. Schlichtmann, "Statistical timing analysis for latch-controlled circuits with reduced iterations and graph transformations," *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 31, no. 11, pp. 1670–1683, 2012.

[4] V. Bhargava, G. Kapoor, S. S. Iqbal, and F. S. June, "Enhanced timing closure using latches," no. Figure 1, 2014.

[5] S. X. Shi, A. Ramalingam, D. Wang, and D. Z. Pan, "Latch modeling for statistical timing analysis," *Proc. -Design, Autom. Test Eur. DATE*, vol. 2, pp. 1136–1141, 2008.

[6] N. Srinivasan, N. S. Prakash, Shalakha D., Sivaranjani D., S. Sri Lakshmi G., and B. B. T. Sundari, "Power Reduction by Clock Gating Technique," *Procedia Technol.*, vol. 21, pp. 631–635, 2015.

[7] D. Amos, "Clock gating," in *FPGA-Based Prototyping Methodology Manual*, D. Amos, Ed. Synopsys, Inc., 2011, pp. 165–179.

[8] "HAPS ProtoCompiler User Guide," no. September. Synopsys, Inc., 2016.

[9] D. Amos, "Automatic gated-clock conversion," in *FPGA-Based Prototyping Methodology Manual*, Synopsys, Inc., 2011, pp. 165–179.

[10] S.-Z. E. Lin, C. Changfan, Y.-C. Hsu, and F.-S. Tsai, "Optimal time borrowing analysis and timing budgeting optimization for latch-based designs," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 7, no. 1, pp. 217–230, 2002.

[11] K. Chae, C. H. Lee, and S. Mukhopadhyay, "Timing error prevention using elastic clocking," *2011 IEEE Int. Conf. Integr. Circuit Des. Technol. ICICDT 2011*, pp. 9–12, 2011.

[12] A. Silva and F. Sill, "Mitigation of Aging Effects Through Selective Time-borrowing and Alternative Path Activation," *ACM/IEEE Prof SBCCI*, pp. 210–216, 2017.

[13] M. Ahmadi, B. Alizadeh, and B. Forouzandeh, "A Hybrid Time Borrowing Technique to Improve the Performance of Digital Circuits in the Presence of Variations," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 64, no. 1, pp. 100–110, 2017.

[14] M. Ahmadi, B. Alizadeh, and B. Forouzandeh, "A Timing Error Mitigation Technique for High Performance Designs," *2015 IEEE Comput. Soc. Annu. Symp. VLSI*, pp. 428–433, 2015.

[15] VivadoUG1, "Vivado Design Suite User Guide: System-Level Design Entry," vol. 901, pp. 1–120, 2013.

[16] J. Zhang and I. M. Nadon, "Latch-based CPU Prototyping with HAPS Platform," pp. 1–10, 2016.