

**UCLA**

**Technical Reports**

**Title**

A Case Study of Logic Delay Fault Behaviors on General-Purpose Embedded Processor Under Voltage Overscaling

**Permalink**

<https://escholarship.org/uc/item/3967v8hw>

**Authors**

Lai, Liangzhen  
Gupta, Puneet

**Publication Date**

2014-08-07

# A Case Study of Logic Delay Fault Behaviors on General-Purpose Embedded Processor Under Voltage Overscaling

Liangzhen Lai

Dept. of Electrical Engineering  
University of California Los Angeles  
Los Angeles, CA 90095  
e-mail: liangzhen@ucla.edu

Puneet Gupta

Dept. of Electrical Engineering  
University of California Los Angeles  
Los Angeles, CA 90095  
e-mail: puneet@ee.ucla.edu

**Abstract**—Voltage overscaling has been an option for energy-reliability tradeoff. This work aims at exploring its efficiency and potential for logic delay faults. Our case study on an ARM Cortex-M0 processor with commercial 45nm libraries shows that the number of delay faults increases dramatically after the first failing operating point which implies that voltage overscaling will be inefficient after the critical operating point. This suggests the need of the monitoring schemes for tracking the critical operating point.

## I. INTRODUCTION

Voltage overscaling has been considered as an option for energy-reliability tradeoff [1]–[4]. Its efficiency, however, depends on how the circuit delay (and corresponding delay faults) reacts to the scaled voltage. Similar analysis has been performed on DSP [5] or as *Critical Operation Point Hypothesis* [6]

This work aims at conducting a case study of the properties of delay faults on a general-purpose, commercial embedded processor. The focus of this case study is exploring the impacts of circuit topology and workload on delay fault behaviors. Our observation based on the case study results includes:

- The endpoint-based criticality analysis may not be a good approach for delay fault analysis as multiple critical paths can end at the same endpoint as fan-in.
- Single-bit-flip may not be a good model for delay faults as most delay faults in our experiments are multi-bit failures, as one critical path can end at multiple endpoints as fan-out.
- Temporal distribution of delay faults has certain dependence on the software workload in our study. More throughout experiments are needed for generic workload.

The rest of the report is organized as follows. Section II explains in detail about our experiment setup. Section III presents and discusses the case study results. Section IV concludes the report..

## II. EXPERIMENT SETUP

This case study is performed on an ARM Cortex-M0 processor [7]. It is a 3-stage pipeline processor that supports Thumb/Thumb-2 ISA. The processor is implemented using

commercial 45nm process technology and libraries. Logic synthesis is performed using Cadence RTL Compiler [8]. Physical place and route is performed using Cadence Encounter [9]. The implemented design has about 10K gates. Detailed breakdown is shown in Fig. 1. The processor is implemented with a frequency target of 250 MHz, i.e., with critical path delay of 4 ns. The fraction of critical Flip-Flops(FFs) under different timing slack values are plotted in Fig. 2.

The case study is based on Verilog simulation with delay annotation. Synopsys VCS [10] is used as the verilog simulator. The Delay annotation information, i.e., Standard Delay Format (SDF) file, is generated with the physical synthesis tool. A verilog testbench is used to drive the simulated processor with all the memory transactions.

The sequential elements are configured to report any signal switching within certain time window before the incoming clock rising edge. The size of the time window is set to be  $k + t_{setup}$ , where  $t_{setup}$  is the setup time of that particular FF, and  $k$  is a user-defined value which is the same for all FFs. When we over-scale the clock period, the data signals start arriving within the time window and being reported as warnings. The additional  $k$  is used to preserve the correct operation while scaling the clock period. In this case study, we set  $k$  to be 1 ns. Therefore, the designed no-fault operating clock period becomes 5 ns.

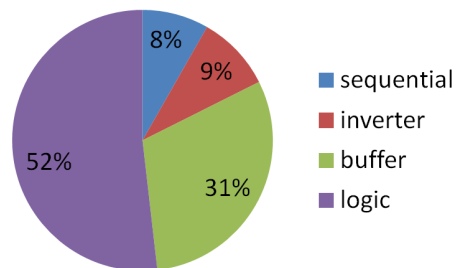


Fig. 1. Gate count breakdown for the processor.

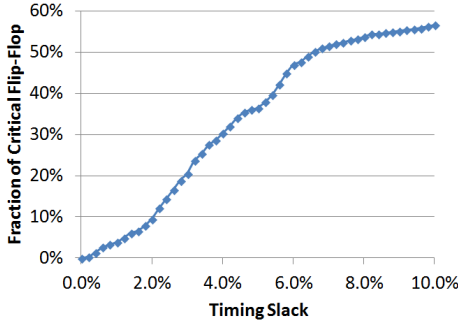


Fig. 2. Fraction of critical FF under different timing slack values

The software benchmark used in this case study is an FFT program from Mibench [11]. The benchmark is cross-compiled for the targeting processor and loaded into the verilog testbench as memory image. The input data is generated as random numbers with controllable seed values. The FFT program takes about 500K clock cycles on the processor.

### III. CASE STUDY RESULTS AND DISCUSSION

With the experiment setup described in Section II, we repeat the simulation with different clock period values. The reported timing warnings are recorded and used for the delay fault analysis in the rest of this section.

#### A. Delay Faults vs. Clock Period

There are two effects of changing clock period on the circuits:

- 1) The number of potential faulty FFs increases with scaled clock period.
- 2) The number of potential faulty paths ending at the same FF increases with scaled clock period.

The trend of 1) can be inferred from Fig. 2, if we ignore the path activation factor. But the effect of 2) depends on both how many paths ends at the FF and how frequent are the paths being activated. This depends on both the circuit topology and workload. Based on the FFT program running on the processor, the number of delay faults of one single FF with different clock period values is shown in Fig. 3. Since we are using static delay annotation in the experiments, the same critical path will have the same delay regardless of the clock period. Therefore, the increased delay fault count at reduced clock period implies that there are new critical paths ending at the same FF.

The number of delay faults of all FFs and one single FF with different clock period values is plotted in Fig. 4. As analyzed earlier, the effect of increased number of faulty FFs causes the increase of overall delay fault count from the red curve to the blue curve.

#### B. Number of Delay Faults in a Cycle

The correlation of the delay faults ending at different FFs is also important for correctly modeling of delay faults. We use the recorded timing warning log generated during the

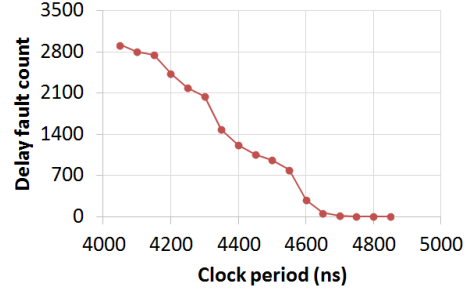


Fig. 3. Number of delay faults of one single FF with different clock period values

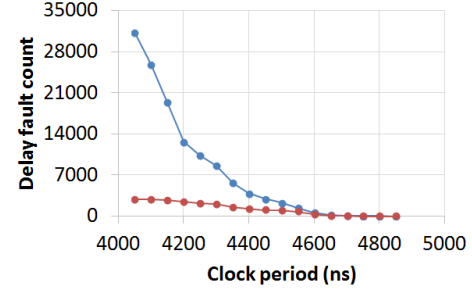


Fig. 4. Number of delay faults of all FFs(blue) and one single FF (red) with different clock period values

simulation to identify the number of delay faults in each clock cycle. The histogram of the number of delay faults at clock period of 4050 ps is shown in Fig. 5. The results imply that most delay faults occur in a single cycle are multi-bit faults. We also plot the histogram at a much slower clock period of 4650 ps (see Fig. 6). A significant fraction of faults are still multi-bit faults. This can be caused by a slow path fanning-out into multiple FFs.

#### C. Temporal Distribution of Delay Faults

The temporal dependence of delay faults is also important for exploring potential system-level and software-level mechanisms for handling delay faults, as they tend to have longer turnaround time. The temporal distribution of delay faults at clock period of 4050 ps is shown in Fig. 7, where each dot represents the number of delay faults within 200 cycles. The

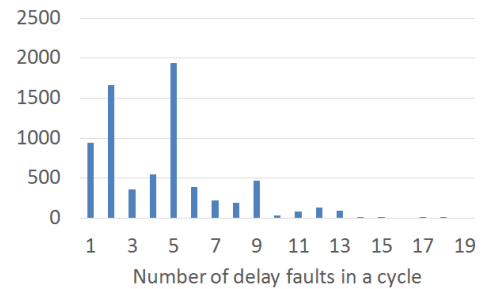


Fig. 5. Histogram of the number of delay faults in a cycle at clock period of 4050 ps

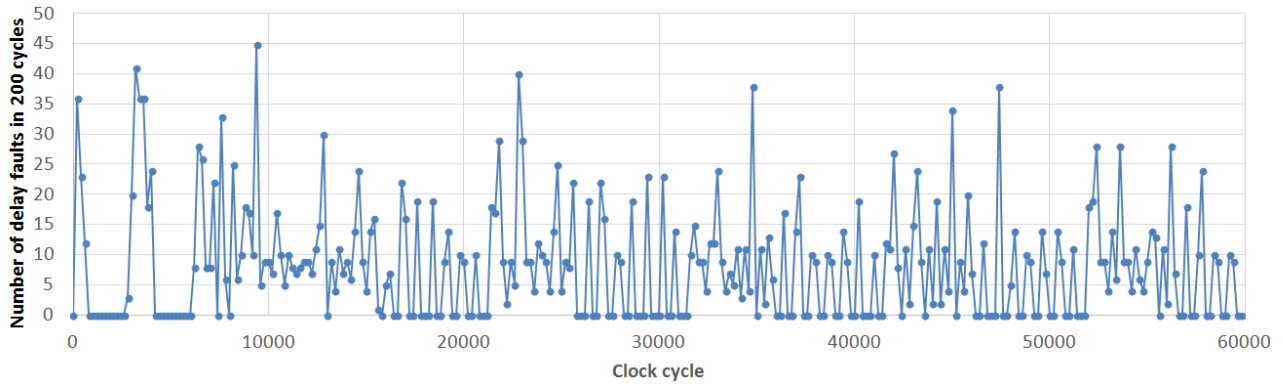


Fig. 7. Temporal distribution of delay faults at clock period of 4050 ps. Each dot represents the number of delay faults within 200 cycles

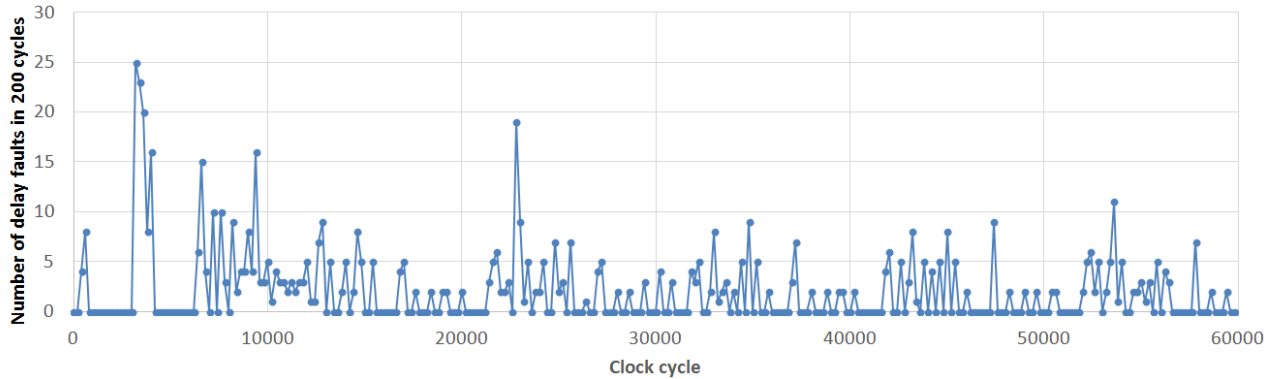


Fig. 8. Temporal distribution of delay faults at clock period of 4650 ps. Each dot represents the number of delay faults within 200 cycles

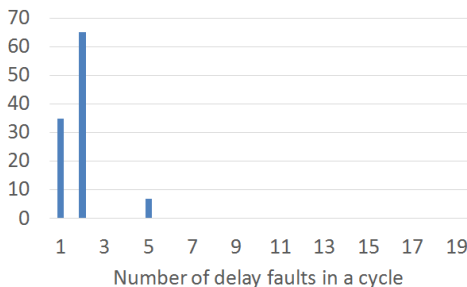


Fig. 6. Histogram of the number of delay faults in a cycle at clock period of 4650 ps

delay faults are distributed along the software execution and the length of fault-free segments are typically small.

As comparison, the temporal distribution of delay faults at clock period of 4650 ps is shown in Fig. 8. We see similar distribution of delay faults as in Fig. 7. The fault-free segments are with larger length and occur with certain patterns. Further analysis is required to identify the connection between these patterns and corresponding software code segments or phases.

#### IV. CONCLUSION

We perform a case study of delay faults on a processor with overscaled clock period. The results show that circuit topology plays an important role in how the delay faults occur.

The results also show that workload has certain impact on the temporal distribution of delay faults. Further analysis is required to identify the cause of workload dependence.

#### REFERENCES

- [1] A. B. Kahng, S. Kang, R. Kumar, and J. Sartori, "Slack redistribution for graceful degradation under voltage overscaling," in *Design Automation Conference (ASP-DAC), 2010 15th Asia and South Pacific*. IEEE, 2010, pp. 825–831.
- [2] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *Proceedings of the 1999 international symposium on Low power electronics and design*. ACM, 1999, pp. 30–35.
- [3] T. Austin, V. Bertacco, D. Blaauw, and T. Mudge, "Opportunities and challenges for better than worst-case design," in *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*. ACM, 2005, pp. 2–7.
- [4] C. Tokunaga, J. F. Ryan, T. Karnik, and J. W. Tschanz, "Resilient and adaptive circuits for voltage, temperature, and reliability guardband reduction," in *Reliability Physics Symposium, 2014 IEEE International*, June 2014, pp. 3D.3.1–3D.3.5.
- [5] Y. Liu, T. Zhang, and K. K. Parhi, "Computation error analysis in digital signal processing systems with overscaled supply voltage," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 18, no. 4, pp. 517–526, 2010.
- [6] J. Patel, "Cmos process variations: A critical operation point hypothesis," in *Online Presentation*, 2008.
- [7] "ARM Cortex-M0." [Online]. Available: <http://www.arm.com/products/processors/cortex-m/cortex-m0.php>
- [8] "Cadence rtl compiler." [Online]. Available: [http://www.cadence.com/products/ld/rtl\\_compiler/pages/default.aspx](http://www.cadence.com/products/ld/rtl_compiler/pages/default.aspx)
- [9] "Cadence encounter." [Online]. Available: [http://www.cadence.com/products/di/edi\\_system/pages/default.aspx](http://www.cadence.com/products/di/edi_system/pages/default.aspx)

- [10] "Synopsys vcs." [Online]. Available: <http://www.synopsys.com/Tools/Verification/FunctionalVerification/Pages/VCS.aspx>
- [11] "Mibench." [Online]. Available: <http://www.eecs.umich.edu/mibench/>