

# A Cost-efficient Delay-Fault Monitor

Gaole Sai, Basel Halak, Mark Zwolinski

Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

Email:{g.sai, bh9, mz}@ecs.soton.ac.uk

**Abstract**—Delay-fault monitoring sensors are widely used for Dynamic Voltage and Frequency Scaling (DVFS) to compensate for intrinsic Process, Voltage, Temperature and Ageing (PVTa) variations. Such techniques are generally based on monitoring the circuit's critical paths. This paper presents a new delay-fault monitoring circuit, which is able to monitoring multiple paths simultaneously. The proposed circuitry has been designed and verified in a 32 bit MIPS processor using a 65nm technology. Our results indicate that the use of the proposed sensor for delay monitoring can lead to a significant saving in area and power overheads of two-thirds and one-third, respectively, compared to a canary flip-flop.

## I. INTRODUCTION

Integrated circuits are typically designed with a safety margin as the performance varies due to unavoidable PVTa variations [1]. This means the circuits are generally designed for a combination of worst-case conditions, which limits the system performance and leads to an increase in power consumption during the system's lifetime. DVFS schemes have been proposed to dynamically eliminate the unused safety margin to improve the power-efficiency [2]. Such schemes use in situ sensors to monitor the delay-fault from the longest delay path [1], [3]. The system can then dynamically scale the supply voltage and the operating frequency to compensate for the impact from PVTa variations.

Generally, the in situ delay monitoring sensors aim to detect data consistency before or after the clock rising edge to predict or detect delay faults [3], [4]. This can be done either by double sampling or by stability checking techniques [1], [5]. The sensors are usually implemented at the end of near critical paths. Each sensor only monitors delay faults from the paths that share the same end. Technology shrinking has exacerbated timing-dependent process and ageing variations, which may change the ranking of critical paths. This will led to a significant rise in the number of ageing and process variation Potential Critical Paths (PCPs) that are deemed to be vulnerable to delay faults [6], [7]. The cost of conventional in situ delay monitoring is thus becoming prohibitive.

In this paper, we propose a cost-efficient Parity Check Circuit (PCC) for delay fault prediction to mitigate the cost of in situ delay monitoring. PCC is able to monitor multiple paths simultaneously, which significantly reduces the number of sensors. The proposed sensor has been designed and verified in a 65nm technology. Our results indicate that using the proposed sensor for delay fault monitoring in a 32-bit MIPS can lead to a significant saving in area and power overheads, compared to the use of canary flip-flops [3]: by two-thirds and one-third, respectively.

The rest of the paper is organized as follows. Section II describes related work. Section III outlines the design principles of the PCC. Verification results and a cost analysis are discussed in section IV. Finally conclusions are drawn in section V.

## II. BACKGROUND AND PREVIOUS WORK

### A. Latch Type Sensors

A number of latch-type delay fault monitoring sensors have been proposed in recent years, [8], [9]. These replace the main FF with a latch, and therefore decrease the transition time from D to Q. The error signal is triggered by detecting transitions while the latch is enabled. However, replacing the FFs in a pipelined system may will cause further reliability issues as short paths may share the same end as PCPs. Two stages on both sides of a sensor will be connected to each other directly while the clock is at logic '1'. To compensate for this, buffers and clock duty cycle adjustment circuitry are generally implemented in the system where latch-type sensors are used [8]. This will increase the complexity of the system and means that latch-type sensors lose their advantage in terms of cost efficiency [9]. Delay fault recovery circuitry is also required when an actual error occurs.

### B. Canary FF

Unlike latch-type sensors, the Canary flip-flop [3] is a flip-flop-type delay fault prediction sensor that uses a double sampling technique. It predicts delay faults by detecting the consistency of the output signal from a PCP before the rising clock edge. Fig. 1 shows the architecture of the Canary FF. A Canary FF consists of a main flip-flop, a shadow flip-flop, one delay element and a comparator. The detection window of the Canary is defined by the delay element. The shadow FF receives delayed data, which is the data from the detection window interval before the current time. When a transition approaches the rising clock edge, because of the PVTa variations, the shadow FF will miss the sample before the main FF detects it. The Error signal will then be triggered as there is an inconsistency between the samples from the main FF and the shadow FF. As Fig.1 shows, a group of error signals is collected by a multiple-input OR gate. The Errors signal is asserted when one of more Error signals are triggered from the group of PCPs that the Canary FFs are monitoring.

Compared with delay fault detection sensors, Canary predicts a delay fault before an actual error occurs. Therefore, it does not require further circuitry to recover from a delay fault. However, implementing a shadow FF of the same size as the

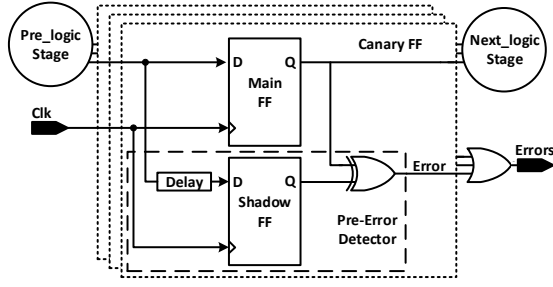


Fig. 1. Canary Flip-Flop [3]

main FF on every PCP will lead to a large area overhead. The area cost for delay fault monitoring in a system might therefore be prohibitive, as will be shown later.

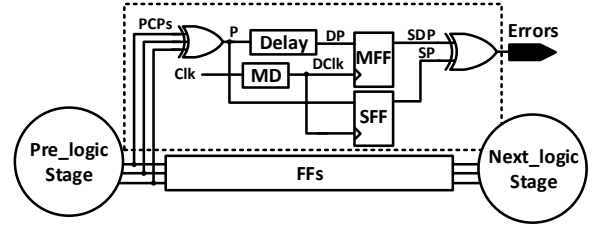
### III. DESIGN PRINCIPLES

This section outlines the operating principles of the proposed circuit. The main advantage over existing sensors is that the PCC can be used for multiple path delay fault prediction, hence it requires less area overhead. Furthermore, compared with latch-type sensors, implementing PCC will not influence the functionality of the original design, as it does not need to replace the FFs on the monitored paths nor add buffers to the short paths of the original design.

#### A. Operating Principles of PCC

Assume that a group of data from monitored PCPs is handled as a single number. Transitions on a PCP will change the parity of that number, from even parity to odd parity or from odd parity to even parity. A delay fault can be predicted when this change is captured before a clock rising edge. The architecture of the PCC is shown in Fig. 2. The PCC consists of one multiple-input XOR gate, a delay element, a matched delay element, one main FF, one shadow FF and a 2-input XOR gate. The multiple-input XOR gate checks the parity of the input from the PCPs. The output signal 'P' is not able to represent the parity at the current time because of the propagation delay of the multiple-input XOR gate. This will cause a phase shift in the detection window. The matched delay element matches the delay of the multiple-input XOR gate to compensate for this phase shift.

Compared with Canary, which detects the data consistency of a single PCP, the PCC checks the parity consistency to monitor multiple PCPs simultaneously. Hence, with an increase in the number of PCPs, the PCC implementation will not lead to a rapid growth in overheads. However, the parity of two sample point will remain the same if an even number of transitions from PCPs occur within the detection window and the delay fault behaviour will be unpredictable in this case. Nevertheless, PCC predicts the delay faults when transitions from PCPs approach the clock rising edge. Therefore it is not necessary to predict every single delay fault on the paths that the PCC is monitoring.



MD: Matched Delay element  
MFF: Main FF, SFF: Shadow FF

Fig. 2. Architecture of The Proposed Circuitry

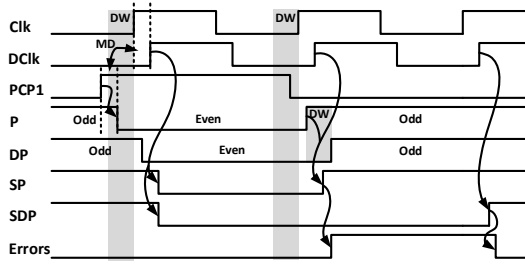
TABLE I  
PERCENTAGE OF TRANSITIONS FROM 4 PATHS

Activity Rate	$\alpha$	10%
One Paths Transition	$C(1, 4) \times \alpha \times (1 - \alpha)^3$	29.16%
Two Paths Transitions	$C(2, 4) \times \alpha^2 \times (1 - \alpha)^2$	4.86%
Three Paths Transitions	$C(3, 4) \times \alpha^3 \times (1 - \alpha)^1$	0.36%
Four Paths Transitions	$C(4, 4) \times \alpha^4$	0.01%
No Transition	$C(4, 4) \times (1 - \alpha)^4$	65.61%

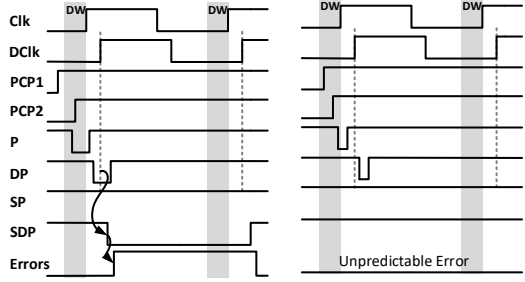
In reality, it very unlikely that the transitions from different PCPs are 100% correlated with each other. Table I shows the percentage of transitions from 4 monitoring points with a 10% activity rate [10]. As the Table shows, there is a 29.52% probability that an odd number of transitions occurs and 4.87% probability that an even number of transitions occurs during the system operating time. The delay fault prediction rate is 95.13% when a single PCC monitors 4 paths simultaneously. Moreover, the delay faults will be unpredictable if and only if an even number of transitions occur in the detection window, as shown in Fig. 3 (a) and (b). Therefore the actual delay-fault prediction rate would be higher than 95.13%. A delay fault will be eventually detected by the PCC when an odd number of transitions occurs.

Fig. 3 shows the timing diagram of the PCC when it monitors a group of paths simultaneously, where  $CLK$  is the clock signal of the system,  $DClk$  is the output signal of the matched delay element,  $PCP1$  and  $PCP2$  are the output signals from two different PCPs,  $P$  is the parity status of the output signals from the PCPs ('0' for even parity, '1' for odd parity),  $DP$  is the delayed parity status (which generates the detection window ( $DW$ )),  $SP$  and  $SDP$  are the output signals of the main FF and the shadow FF,  $MD$  is the delay generated by the matched delay element, and  $Errors$  is the delay fault prediction flag of the PCC. There are three typical operating cases during the PCC operation time, shown in Fig. 3 (a), (b) and (c), respectively.

(a) The PCPs might share the same ends with other short paths. In the first clock cycle, a short path exists and  $PCP1$  changes from '0' to '1' before the detection window (the other PCPs remains the same). This conversion causes the signal  $P$  to switch from odd parity to even parity. The error signal is not triggered as both  $P$  and  $DP$  switched to '0' before



(a) Odd Number of Transitions



(b) Even Number of Transitions (c) Unpredictable Error

Fig. 3. Timing Diagram of the PCC

DClk rises. In the second clock cycle, a PCP is asserted and PCP1 switches within the detection window, which results in the signal DP changing from even parity to odd parity after the DClk rising edge. The error signal is triggered due to inconsistent sampling between the main FF and the shadow FF. This error signal is then cleared in the next clock cycle.

(b) In this case, two PCPs are asserted in the same clock cycle. PCP2 reaches the detection window before PCP1. Transitions from those two paths trigger a pulse in signals P and DP. Signal P switches back to odd parity before DClk rises and signal DP remains at even parity when DClk is rising. The error signal is then triggered as the inconsistency is captured by the main FF and the shadow FF.

(c) When the transitions from two PCPs occur in the detection window, pulses on signals P and DP will be generated before and after the DClk rising edge. The inconsistency will not be captured by the main FF and the shadow FF, thus there is an unpredictable error. This is a rare situation, which will not arise every time that a group of PCPs are monitored. The error will be eventually predicted when an odd number of PCPs are asserted (see Table I).

In practice, the routing area overhead should also be considered. Compared with existing sensors such as Canary, there is more routing on the input side of the PCC as the multiple-input XOR gate needs to be connected to the output signals of the PCPs. However, existing sensors will have more routing on the output side to manage error signals. The PCC does not lose the advantage of cost-efficiency in terms of routing.

## B. Path Selection

The delay fault monitoring sensors would not be implemented on every path. A limited set of paths should be selected [1]. Timing dependent ageing and process variations should be considered as they may change the ranking of the critical paths. A PCP may become the critical path after fabrication due to process variations, or after a certain time because of ageing. Various ageing models are available for timing dependent ageing variations [6]; the potential critical paths after ageing can be identified using those models. The range of behaviours due to process variations can be estimated by applying the data provided for different technologies using worst and best case models.

As an external sensor, PCC would be implemented on the inputs of FFs on the PCPs. Therefore the load capacity of the last gate on the potential critical path may require adjustment. The safety margins need to be defined according to the variability analysis of each sensor implementation. The delay from different inputs of the multiple-input XOR gate should be balanced before implementation.

## IV. VERIFICATION AND COMPARATIVE ANALYSIS

This section first presents the results of functional verification at system level, then we summarise the cost of the proposed delay fault prediction technique when implemented in a 32-bit pipelined MIPS.

### A. System level Simulation Results

The first step in any delay-fault prediction technique is to identify the long delay paths to be monitored. These refer to the critical paths of the circuit under consideration, in addition to all long delay paths that may cause timing errors due to ageing and process variation induced delay degradation. In this example, four paths from the PCPs were selected after the timing analysis for a 32-bit pipelined MIPS in a 65nm technology. The PCC was used to monitor the 4 PCPs simultaneously for verification and evaluation.

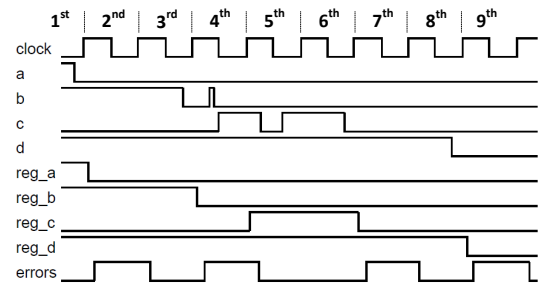


Fig. 4. 4-input PCC functional verification in a 32-bit MIPS

Fig. 4 shows the system level simulation results when a PCC monitored 4 PCPs simultaneously. Those four PCPs will be assigned when the MWB stage writes data '0' back to four different register files. The signals a, b, c and d are the output signals of those four PCPs and the signals reg\_a, reg\_b, reg\_c

and  $reg\_d$  are the data which is stored in the register files. As the figure shows, transitions occur in the first, third, fourth, fifth, sixth and eighth clock cycle. Those transitions can be divided into three categories (I), (II) and (III).

(I) The PCPs were asserted in the first, third, sixth and eighth clock cycles which triggers the late transitions at monitoring points  $a$ ,  $b$ ,  $c$  and  $d$ , respectively. The error signals are generated when the transitions are detected by the PCC without an actual error.

(II) In the fourth clock cycle, a transition is triggered at monitoring point  $c$  by a short path. The error signal is not triggered as the data settles before the detection window.

(III) The pulse signals on monitoring points  $b$  and  $c$  in the fourth and the fifth clock cycles occur due to the competition risk between logic circuits. The error signal is not triggered as the pulse signals occur before the detection window.

### B. Area and Power Overheads Comparison

To compare the area overhead of the proposed design PCC with Canary FF, we have considered a 32-bit pipelined MIPS. The PCC and Canary FF are designed by using exactly the same double sampling circuitry with the same width of detection window to produce an equitable comparison. The PCC and Canary FF were applied to monitor the same group of PCPs. The power and area overheads were estimated from Design Compiler using a 65 nm technology.

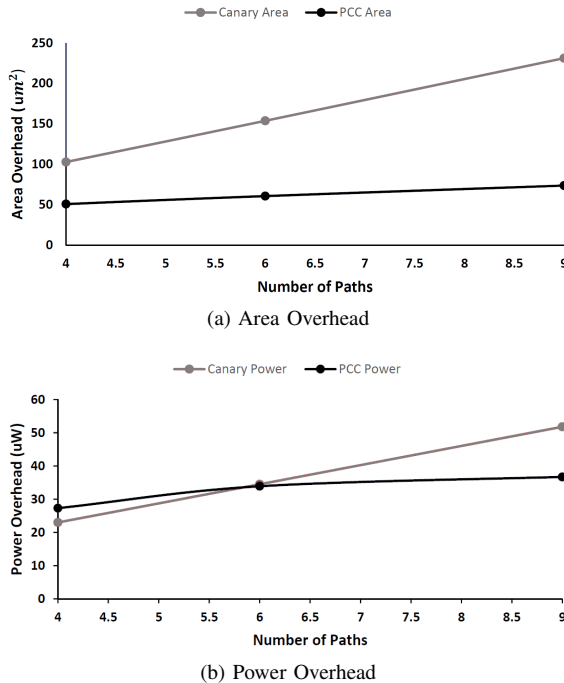


Fig. 5. Area and Power Overheads Comparison

Fig. 5 shows the trends of area and power overheads for 4, 6 and 9 path monitoring when Canary and PCC were applied to the MIPS respectively at the highest operating frequency (800 MHz) with 1.05V supply voltage. As the Fig.5 (a) shows, the area overhead of PCC is generally smaller than that of the Canary FF. Compared with Canary FF, the

growth in area overhead is more than 6 times slower when the PCC was applied to the MIPS. The PCC has a higher power overhead compared with the Canary FF when fewer than 6 paths are monitored, shown in Fig. 5 (b). This is due to the dynamic power overhead produced by the matched delay element, as the matched delay element is connected to the clock signal. However, the power overhead of Canary might be underestimated as Canary FF will require more clock buffers after place and route. PCC saves two-thirds area overhead and one-third power overhead compared with Canary FF when PCC monitors 9 paths simultaneously.

### V. CONCLUSIONS AND FUTURE WORK

In this paper, we have a new delay-fault prediction circuitry, named PCC. PCC is a multiple delay-fault predictor which improves the cost and energy efficiency. Compared with Canary FF, PCC saves two-thirds and one-third of area and power overheads respectively in a 32-bit MIPS. The design was implemented and verified in a 32-bit MIPS in a 65 nm technology. Future research will focus on the full PCC implementation for a DVFS system on a high-performance processor.

### REFERENCES

- [1] D. Bull, S. Das, K. Shivashankar, G. S. Dasika, K. Flautner, and D. Blaauw, "A power-efficient 32 bit arm processor using timing-error detection and correction for transient-error tolerance and adaptation to pvt variation," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 1, pp. 18–31, 2011.
- [2] Y. Huang, M. Li, C. Li, P. Debacker, and L. Van der Perre, "Computation-skip error mitigation scheme for power supply voltage scaling in recursive applications," *Journal of Signal Processing Systems*, pp. 1–12, 2016.
- [3] H. Fuketa, M. Hashimoto, Y. Mitsuyama, and T. Onoye, "Adaptive performance compensation with in-situ timing error predictive sensors for subthreshold circuits," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 20, no. 2, pp. 333–343, 2012.
- [4] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*. IEEE, 2003, pp. 7–18.
- [5] J. Semiao, D. Saraiva, C. Leong, A. Romao, M. Santos, I. Teixeira, and J. Teixeira, "Performance sensor for tolerance and predictive detection of delay-faults," in *Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2014 IEEE International Symposium on*. IEEE, 2014, pp. 110–115.
- [6] E. Mintarno, J. Skaf, R. Zheng, J. B. Velamala, Y. Cao, S. Boyd, R. W. Dutton, and S. Mitra, "Self-tuning for maximized lifetime energy-efficiency in the presence of circuit aging," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 30, no. 5, pp. 760–773, 2011.
- [7] A. F. Gomez and V. Champac, "Early selection of critical paths for reliable nbtI aging-delay monitoring," *IEEE Transactions on very large scale integration (VLSI) systems*, vol. 24, no. 7, 2016.
- [8] S. Das, C. Tokunaga, S. Pant, W.-H. Ma, S. Kalaiselvan, K. Lai, D. M. Bull, and D. T. Blaauw, "RazorII: In situ error detection and correction for pvt and ser tolerance," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32–48, 2009.
- [9] Y. Zhang, M. Khayatzadeh, K. Yang, M. Saligane, N. Pinckney, M. Alioto, D. Blaauw, and D. Sylvester, "8.8 irazor: 3-transistor current-based error detection and correction in an arm cortex-r4 processor," in *2016 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2016, pp. 160–162.
- [10] Y. Lin and M. Zwolinski, "A cost-efficient self-checking register architecture for radiation hardened designs," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2014, pp. 149–152.