Did this paper address an important issue? Explain.

DVS is becoming a widely used technique to reduce the energy consumption of high performance processors. However, the allowable voltages selected in DVS are chosen very conservatively, so that the processor functions correctly even under worst-case conditions. It is observed that the worst-case conditions occur only rarely. Thus, there is room to be more aggressive in selecting the allowed voltages i.e. reducing them further, to save more energy. However, this can lead to timing violations, and hence wrong results, when the worst-case conditions do occur. Thus, for this approach to be feasible, there has to be some reliable error detection and correction mechanism built into the system. This paper describes one such circuit (the Razor FF) using which the errors can be detected and corrected. This allows the aggressive reduction of $V_{DD}$ to save energy. Thus, this paper deals with an important issue.

Are the proposed approaches valid? Describe its strength and weakness.

The approach is valid, since it is able to reliably detect and correct the errors which will occur when timing violations occur due to worst-case conditions. (The probability that the error signal itself will go metastable is negligible.)

Strengths:

- The Razor FF is able to detect and correct errors with a very small penalty (one cycle)
- Circuit issues like power consumption and delay due to the introduction of Razor FFs are considered and minimized
- The delayed clock can be generated locally, and so the routing (additional metal layer) and power overhead normally associated with clocks can be avoided
- Metastability issues have been considered and the probability of metastability propagating through the design has been made negligibly small. So for all practical purposes, the design is quite reliable
- Less than 10% of the FFs in a design have to be converted to Razor FFs
- The performance overhead is quite acceptable (a few percent) considering the fact that impressive energy savings are obtained (40%)

Weaknesses:

- One of the main overheads is the overhead for the VLSI CAD tools due to the latch used in the Razor FF. Static timing analysis tools are not capable of dealing with latches, so the designer will have to manually insert false-paths. Also, the use of latches introduces the need to have a ‘short-path’ constraint, which will complicate the timing analysis and add additional burden on the VLSI CAD tools.
• The voltage controller response has to be kept sluggish, to avoid oscillations and instability. This causes a lot of errors when the voltage becomes very low, and thus the is significant performance degradation and energy loss (due to re-computations) during such periods. This reduces the overall effectiveness of the DVS approach.

• The work does not consider the effect of the low voltage on the system bus. The low voltages may cause the system bus to experience several timing errors and thus be a performance bottleneck.

• The performance degradation in real applications (benchmarks) is not gradual and varies between programs. This can put a firm lower limit on the acceptable voltage value.

Do the results support the conclusions? Explain.

The various experimental results do support the main hypothesis that the worst-cases conditions do not occur frequently. Their main conclusion is that significant energy savings are possible while maintaining good performance if a small number of errors can be reliably detected and corrected. Their results do support these conclusions, since they are able to achieve average energy savings of around 40% with an average performance hit of less than 2%.

Describe the potential future works?

The potential future works can include:

• Study of the voltage controller to find one which can give better response time, while being stable.

• Study of the effect of the low voltage on the system bus to determine if it can be a potential bottleneck.

• Can the Razor approach be used for hard real-time systems?

• Does using Razor approach for memory/cache have any benefit?

• Is this approach feasible for a complicated design (e.g. an OOO, speculative processor)? In such a design, there may be a lot of critical paths, and so a larger number for Razor FFs may have to be introduced. These FFs will cause power and delay overhead. What kind of trade-offs can be made in such a design?