Partial Computation-skip Scheme for Power Supply Voltage Scaling

Yanxiang Huang*†, Meng Li*, Chunshu Li*, Liesbet Van der Perre*†
*imec, Kapeldreef 75, 3001 Heverlee, Belgium
†KU Leuven, Kasteelpark Arenberg 10, 3001 Heverlee, Belgium

***Abstract*—Aggressive power supply voltage V_{dd} scaling is widely utilized to exploit the design margin introduced by the process, voltage and environment variations. However, scaling beyond the critical V_{dd} results to numerous setup timing errors, and hence to an unacceptable output quality. In this paper, we propose partial computation-skip (CS) scheme to mitigate setup timing errors, for recursive digital signal processors with a fixed cycles per instruction (CPI). A coordinate rotation digital computer (CORDIC) with the proposed partial CS scheme still functions when scaling beyond the error-free voltage. It achieves 1.7 X energy saving w.r.t. nominal V_{dd} condition, and another 1.3 X energy saving when sacrificing 10.5 dB output quality.**

I. Introduction

In modern deep scaled CMOS ICs, the process, voltage and temperature (PVT) variations result to a randomized variation of the transistors parameters. For instance, for each transistors, the gate width and length, channel mobility, threshold voltage V_{th} might be altered because of the process imperfection. Power supply voltage V_{dd} is also fluctuating due to noise or IR drop. Besides, environmental temperature and transistor aging change the propagation delay of a specific transistor. As a consequence, the propagation delays of two theoretically identical transistors are different. This phenomena becomes more significant with the advancement of the CMOS technology [1], [2]. This is because i) in a scaled technology node, even tiny process parameter fluctuations have relative large impact on the parameters, e.g. random dopant fluctuation; and ii) the extensively scaling of supply voltage increases the circuit instability, which result to a higher vulnerability to environmental fluctuations.

Conventionally, to cope with this variability challenge, ICs are designed using the worst PVT corner, to ensure they always operate correctly. For the 28nm CMOS technology, electronic design automation (EDA) tools perform signal setup timing check in the worst-case scenario, i.e. in the slowest process parameter corner, with 10% V_{dd} drop margin, and at 0 degrees Celsius. Nevertheless, ICs rarely operate at the worst-case corner. Therefore, this worst-case approach introduces a design margin, leading to wasted performance capability and power consumption. For instance, at the 28nm CMOS technology, the speed margin between the typical-case situation and the worst-case design metric can be as high as 2.2 times. (Fig.1)

Power supply voltage V_{dd} is usually being scaled down to exploit the design margin, since power consumption is proportional to V_{dd}^2. The side effect of V_{dd} scaling is the increased propagation delay, which is proportional to $1/V_{dd}$.

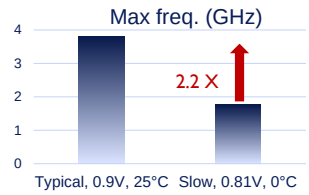


Fig. 1. Speed comparison of a given circuit estimated in worst-case corner and in typical-case scenario, in 28nm CMOS technology.

Therefore, designer must ensure the setup timing constraints for the critical paths are not violated, even at the scaled V_{dd}. Recently, on-chip monitor techniques are proposed to estimate the critical path setup timing situations during V_{dd} scaling, to reduce the design margin [3]–[10]. Those monitors also act as error correctors once setup timing error is detected, using error correction schemes, at the expense of throughput penalty.

For some real-time streaming applications, e.g. in communication systems, to eliminate storage buffer for the mixed signal circuits, a constant throughput is usually required. For those applications, conventional monitor-based error correction schemes are not suitable, since they lead to throughput penalty. From another perspective, the approximate computing opportunity in those applications leaves room for sacrificing the output quality to meet the stringent throughput requirement.

Algorithmic noise-tolerance (ANT) signal processing techniques [6] tackles the energy overhead from other perspectives. Since DSPs can live with a certain amount of errors in nature, by detecting and correcting the errors on the algorithmic level, they can trade-off output quality, in an acceptable degree, for power efficiency and chip area. With ANT techniques, a remarkable amount of energy is saved at the cost of slightly reduced output quality [7], [8]. Nevertheless, this scheme need to be tailored to each individual application, as it relies heavily on the characteristics of the target computation. Another drawback for ANT is the introduction of approximation error. For recursive algorithms, errors introduced by earlier iterations are usually amplified in later iterations, which greatly decreases the output quality.

To exploit the design margin for those constant throughput, soft error constraint, recursive digital signal applications, this paper proposes a partial computation-skip (CS) scheme to maintain the cycles per instruction (CPI) without throughput penalty during V_{dd} scaling. The in-situ timing checks are performed with the Double Sampling with Time Borrowing

(DSTB) [9]. The error correction is a mixture of algorithm and circuit level scheme: it maintains the constant CPI and mitigates the error by double sampling and by approximation logic. When timing violations are detected, part of computations in the next cycle are skipped as an approximation. A benefit of the proposed scheme is that, error introduced by skipping will not propagates to later iterations. This results to a gracefully reduced output quality, which is acceptable for wireless communication and video processing applications, where a good enough result is sufficient.

The proposed partial CS scheme is demonstrated in a coordinate rotation digital computer (CORDIC) working in vectoring mode, of which a typical application is a Cartesian to polar coordinate vector translator for polar transmitters.

II. Power Supply Voltage Scaling Methods

A stereotypical V_{dd} scaling scheme is illustrated in Fig.2. Alongside the core circuit, the error detection unit checks whether the critical paths in the core circuit violates the setup timing constraint. The errors are corrected by the error correction unit. Besides, the setup timing violation information is fed to supply voltage controller unit to adjust the V_{dd}. If no or very few errors are detected, the V_{dd} will be scaled down to reduce power consumption. The adjusting frequency of the supply voltage controller is much slower than the core circuit, for two main reasons: i) to save computation power dissipation in the supply voltage controller; ii) the transition delay for changing the V_{dd} is inherently much larger than the clock period of the core circuit.

Fig.3 summarized various V_{dd} scaling methods. The worst corner method is the conventional solution. It sets a fixed V_{dd} to above the require V_{dd} for safety computation, to meet the rarely existed worst-case corner IC. Therefore, its power consumption is the highest. Dynamical scaling methods find the most optimal V_{dd} for each chip, at chip setup stage, or periodically adjusts to environmental changes. This method

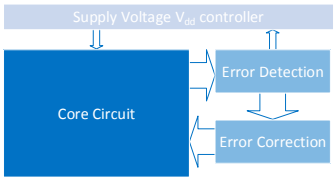


Fig. 2. V_{dd} scaling scheme for a core circuit, equipped with in-situ error detection, error correction, and V_{dd} controller unit.

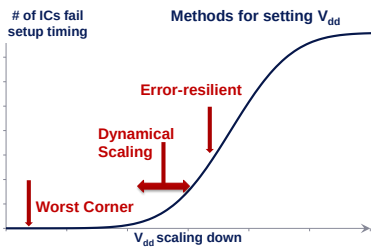


Fig. 3. Various methods in selecting the V_{dd} per chip.

reduces the V_{dd} and hence reduces the power consumption. However, because of transient degradation, e.g. noise related IR drop, V_{dd} will drop occasionally. To ensure no timing violation occurs at run-time, designer should leave a V_{dd} margin, which is also a energy waste. The third approach is error resilient method, which eliminates the V_{dd} margin by mitigating or correcting the infrequent setup timing errors.

For error detection, timing checks on flip-flops (FFs) have been proposed and widely utilized. This scheme modifies the FF in the circuit. A circuit for adaptive operating control with critical path replica monitor is presented in [3]. However, it suffers from the delay mismatch between the replica and the actual critical path caused by within-die variations. Canary FF [5] predicts timing errors before they actually happen, which requires no extra recovery cycles. Nevertheless, the inaccuracy of the timing prediction limits fully exploit the variability design margin.

In contrast, in-situ schemes [4], [5], [9], [10] are proposed to overcome this mismatch for micro-processors. Razor I [4] detects timing error with a shadow latch. However, the RAZOR I monitor exhibits meta-stability problem in the data path. The meta-stability problem occurs when the input signal changes along with the rising edge of the clock signal. In this case, the FF in the RAZOR circuit are pulled to 1 and 0 simultaneously. The circuit will then require an unbounded duration before resolving to a final state (1 or 0), according to the environmental noise. In the Razor I, this happens to the FF which is located in the data-path. As a consequence, the micro-processor runs into meta-stable state, which is difficult to resolve.

The DSTB monitor proposes to swap the position of the FF and latch to eliminate the disadvantages of the RAZOR I monitor in the data-path. In a DSTB detector, the data signal arrives later than the required timing constraint (i.e. time margin), e.g. the rising edge of the clock signal, the data signal will still be captured by the latch. The latch circuit is sensitive even after the rising edge of the clock signal in contrary to a flip-flop. The DSTB circuit detects a timing violation by comparing the results from the latch and the FF. As the signal from latch can be used in the next cycle as the correct input from the previous cycle, the DSTB can find the exact timing slack and hence utilize it to reduce the design margin. Recently, Bubble Razor [10] is proposed to avoid the short-path constraints in previous designs by adapting the two-phase latch based design.

In scaled V_{dd} scenarios, infrequent timing violations occur. If V_{dd} is scaled up, those timing violations can be avoided. However, power consumption will increase. Error correction schemes handle those infrequent timing violations and provide reliable output. Counter-flow [4] and instruction replay [9] are proposed to correct the error by issuing extra cycles. They results to multiple-cycle throughput penalty once a timing error is detected. Bubble Razor [10] reduces the throughput penalty to 1 cycle. However, the design is based on two-phase latches, which is difficult for incorporating with mainstream EDA tools. Global clock gating scheme [4] also achieves 1 cycle

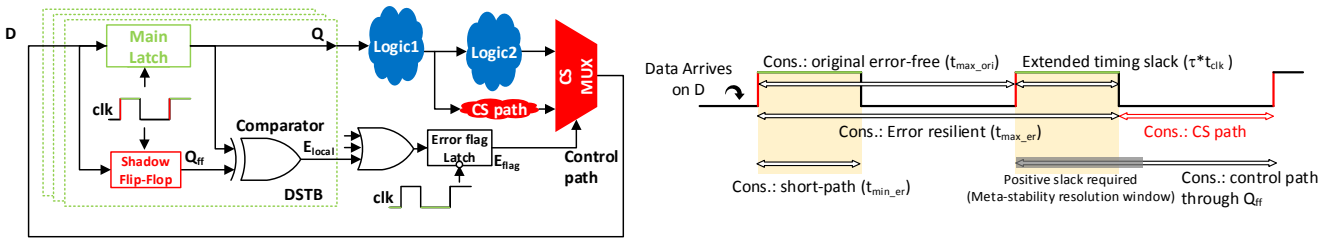


Fig. 4. Left: partial CS scheme implemented in a recursive processor. It consists of logic stage (Logic1 and Logic2), double sampling with time borrowing (DSTB) [9], and a CS MUX. Right: timing constraints for the processor with partial CS scheme.

penalty. Nevertheless, it is difficult to implement for large area high-speed circuits. Recently, a local 1-cycle error correction scheme is proposed [11]. However, the timing constraint for its error signal becomes very challenging for multiple fan-in situations. In sum, none of the previous proposed schemes achieves 0-cycle penalty in case of a timing error.

III. Partial CS Error Correction Scheme

To achieve 0-cycle penalty error correction, we propose this partial CS error correction scheme. The concept of the proposed partial CS error correction scheme is an advanced version of the CS error correction scheme presented previously [12].

In this section, the partial error correction scheme is presented in a recursive processor (see Fig.4). The error detection circuit consists of an array of DSTBs [9]. In each DSTB, the main latch captures the input signal D and produces an output Q, even if it arrives later than the clock rising-edge. In the unlikely event of a timing violation, the rising-edge sensitive shadow FF captures incorrect data after the clock rising-edge, and produces Q_{ff}, an incorrect sample of D. Note that Q is still correct since the main latch is transparent throughout the clock high period. By comparing Q and Q_{ff}, an error indicator E_{local} is identified. A setup timing violation in any DSTB is regarded as a setup timing failing for whole circuit. Therefore, all E_{local} are connected with a *OR* gate to produce a global error flag E_{flag}. A low sensitive latch is inserted after combing all the E_{local} through the 'OR' gate, so that the E_{flag} only changes after the shadow latch is stable. This reduces the glitching of the E_{flag}.

As the main latch is active throughout the clock high period, it is able to tolerate longer propagation delay in data-path than a nominal circuit without error correction schemes. As a consequence, the circuit is able to operate at V_{dd}s that lead to setup timing violations, or sub-critical V_{dd}s. By tuning the clock duty cycle factor τ (the percentage it stays high), designers can extend the nominal delay constraint $t_{max\_orignal}$ to the error resilient timing constraint t_{max_er}:

$$\begin{aligned} t_{max\_ori} &= t_{clk} - t_{setup\_FF} \\ t_{max\_er} &= t_{clk} + \tau \cdot t_{clk} - t_{setup\_latch} \end{aligned} \tag{1}$$

where t_{clk} is the clock duration, t_{setup_FF} represents the setup time for a normal FF, and t_{setup_latch} represents the setup time for the main latch. The maximum ratio R of speed degradation because of V_{dd} drop is denoted as:

$$R \triangleq \frac{t_{max\_er}}{t_{max\_ori}} \tag{2}$$

Substitute t_{max_CS} and t_{max_ori} from (1), we obtain

$$\begin{aligned} R &= \frac{t_{clk} + \tau \cdot t_{clk} - t_{setup\_latch}}{t_{clk} - t_{setup\_FF}} \\ &\approx 1 + \tau \end{aligned} \tag{3}$$

In [12], once a timing violation is detected by the DSTB, since re-computing the next logic with the late-arrived correct Q is impossible due to the setup timing constraint for the next cycle, Q is fed directly to the MUX as skipping. This skip can be regarded as a naive implementation for the approximated version of the logic. Since the quality degradation of skipping the whole logic can be very significant. Therefore, in this paper, we propose to skip partly of the next computation. As in Fig.4, only the second part, i.e. logic2, is skipped. As a result, the correct signals from the previous clock are preserved, and only a small portion of the next logic computations are skipped. For the CS path during error correction, signal through Q and through Q_{ff} have different timing constraints, as shown in Fig.4.

The shadow FFs will violate the timing constraint at sub-critical V_{dd} situations. So they might experience meta-stability. Therefore, those paths through Q_{ff} are guarded with extra slacks, serving as the resolution window for the FF to settle. For the selected 28nm CMOS technology in this paper, the resolution constant is measured as 20ps. Therefore, we set the slack to 700ps, which is sufficiently large to guarantee Mean Time Before Failure (MTBF) requirement for the system due to meta-stability. Note that the main latch are designed to never fail at even sub-critical situations, the data-path and CS path are immune from meta-stability, which is a big advantages for the DSTB.

As the main latch is still sensitive after the clock rising-edge throughout the first half of the clock, if it captures the new arrived signal too early, the signal from the previous cycle is flushed. In this situations, the error detection circuit might indicates a false error. Therefore, for the paths to main latch, a short path timing constraint is required:

$$t_{min\_er} = \tau \cdot t_{clk} + t_{hold\_latch}, \tag{4}$$

where t_{hold_latch} is the hold time for the main latch. This short time constraint is guaranteed by inserting buffers during placement & route.

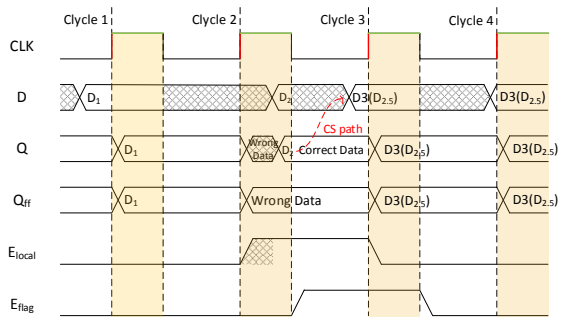


Fig. 5. An example timing diagram for the recursive processor with the partial CS scheme in Fig.4.

A timing diagram example is shown in Fig.5. When cycle 1 finishes, D receives the stable signal D_1 ahead of the rising edge, so both Q and Q_{ff} get the correct data, and hence E_{local} stays low. During cycle 2, new signals from logic is propagated to D. Suppose that the setup constraint is violated due to a slow logic during cycle 2. Q_{ff} captures an arbitrary wrong data since D is unstable when cycle 2 finishes. However, Q keeps glitching until D is stable, when the correct signal is captured. So a positive E_{flag} is captured by the comparator, which indicates a timing violation. As a consequence, Q is fed to D through the CS path for the next cycle. Therefore, the signal value in D_3 is annotated with a $D_{2.5}$, meaning that the computation of 2.5 cycles are performed for the given data. After cycle 3, all signals are returned to the normal state. In sum, 2.5 effective iterative calculations are performed during the first 3 cycles due to the timing violation during cycle 2.

IV. CORDIC PROCESSOR

A CORDIC is a simple and efficient implementation to calculate trigonometric functions. It is usually equipped in ASIC polar transmitters, for which an otherwise hardware multiplication is relatively power hungry. In this paper, the CORDIC is used in polar transmitters to convert the Cartesian vectors, in I & Q format, to polar coordinate vectors.

A. algorithm

The CORDIC works in vectoring mode, to compute the magnitude M and initial angle ϕ of a input vector $[x_0, y_0]$, where x_0 is positive and y_0 is arbitrary. The output ϕ is represented in its trigonometric form ($Cos(\phi)$ and $Sin(\phi)$), as demanded by later stages of the DSP. By definition, the outputs are formulated as

$$\begin{aligned} M &\triangleq \sqrt{x_0^2 + y_0^2} \\ Cos(\phi) &\triangleq x_0 / \sqrt{x_0^2 + y_0^2} \\ Sin(\phi) &\triangleq y_0 / \sqrt{x_0^2 + y_0^2} \end{aligned} \tag{5}$$

Input vector $[x_0, y_0]$ is rotated recursively by iterative micro-rotations. For each iteration,

$$\begin{aligned} x_{i+1} &= x_i - y_i \cdot d_i \cdot 2^{-i} \\ y_{i+1} &= y_i + x_i \cdot d_i \cdot 2^{-i} \end{aligned} \tag{6}$$

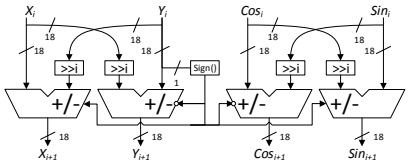


Fig. 6. Vectoring mode CORDIC cell for a CORIC iteration

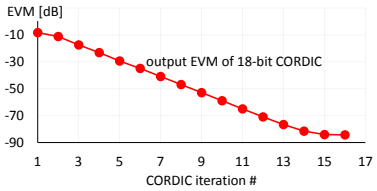


Fig. 7. CORDIC output EVM evolves with numbers of iterations n

where $d_i = 1$ if $y_i < 0$, $d_i = 0$ if $y_i >= 0$.

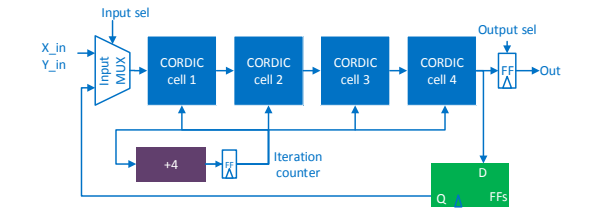
After n iterations, y_i reaches 0, and the resulted x_n represents the original magnitude with a pre-defined magnitude scaling factor. Fig.6 shows the hardware diagram of a CORDIC iteration, or CORDIC cell, with 18-bit width. Each signal (x_i, y_i, $sin(\phi)_i$, and $cos(\phi)_i$) is right shifted by a barrel shifter before added (or subtracted) by its corresponding summand signal. In this paper, the output quality of a CORDIC is measured as the error vector magnitude (EVM).

Fig.7 shows that the EVM for the CORDIC evolves with the iteration #. This evolution characteristics provides space for trading off the iteration number at V_{dd} over-scaling situations. Thus, we propose to apply the proposed partial CS scheme on the CORDIC application. Once a timing violation is detected, we skip part of the computation in the next cycle and adjust the iteration counter. At the requirements of constant CPI, the final iterations, which contribute less to the EVM, are skipped to ensure previous computations are guaranteed even at sub-critical situations.

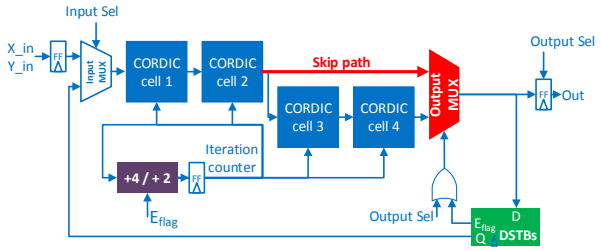
B. conventional CORDIC processor

A conventional implementation of the CORDIC algorithm is demonstrated in Fig.8(a). The internal word size is 18-bit. 16 iterations are performed for each operation. The processor contains 4 CORDIC cells, each performs a CORDIC iteration. Therefore, 4 clocks cycles are required to finish a CORDIC operation ($CPI = 4$). Computation results are stored in the FFs, whose outputs serves as the inputs for the next cycle. Iteration counter counts the CORDIC iterations, and controls the barrel shifter in each CORDIC cell. In each cycle, the iteration counter is counted up by 4, meaning that 4 CORDIC iterations are finished in this cycle.

The conventional CORDIC requires intensive gate-sizing to meet the speed requirement, due to two reasons: i) the frequency requirement from the application is challenging; ii) the V_{dd} drop margin, which is set to 70mv in this paper, tighten the setup-timing constraint.



(a) Output quality



(b) Conventional CORDIC processor

Fig. 8. Implementation of Conventional and Proposed partial CS CORDIC processors

C. partial CS CORDIC processor

To correct setup timing error and eliminate the V_{dd} drop margin, the proposed partial CS is deployed to the CORDIC processor. (see Fig.8(b)). The FFs for storing the computation output are replaced with DSTB for double sampling. In addition, The computation-skip (CS) MUX selects the signal from the nominal logic, or from the CS path, according to E_{flag}.

When a setup timing violation (for the previous cycle) is detected at the beginning of the current cycle, the last 2 CORDIC iterations are skipped because of insufficient processing time. Therefore, the iteration counter is counted up by 2 instead of by 4. As a consequence, the intended 4 CORDIC iteration computation will eventually conduct 2 iterations. Those skipped computations are actually performed in later stages. As a result, only the final computations are skipped because of the constraint of fixing the CPI to 4.

Since errors cannot be detected unless a cycle is finished, we cannot correct timing error for the last cycle. To solve this, the signal in the last cycle always takes the skip path, which reduces the propagation delay, and thus eliminates the occurrence of timing violations. This results to 3.4 dB EVM loss. However, this loss can be predicted at design time, and hence be compensated by other techniques, e.g. reducing quantization error by increasing the word width.

By checking the final iteration counter, we get the information of how many iterations are skipped. In other words, the iteration counter serves as a circuit quality monitor, which indicates how critical the setup timing constraint is. Therefore, V_{dd} can be adjusted at run time based on this quality monitor. Besides, iteration counter also severs as a output quality monitor, because of its strong correlation with EVM. As a consequence, algorithm-level error compensation is also possible based on this quality monitor.

Because of the partial CS scheme takes care of the V_{dd} drop related error. The V_{dd} drop margin during design time can be eliminated. As a result, the setup timing constraint in data-path (CORDIC cells) is relaxed and hence smaller and slower gates can be used. The timing constraint for control paths are still tight, to ensure the error correction circuit functions during V_{dd} over-scaling or V_{dd} drop. In sum, as the design is data-path dominated, the whole design is expected to be smaller and lower in power consumption.

V. Evaluation

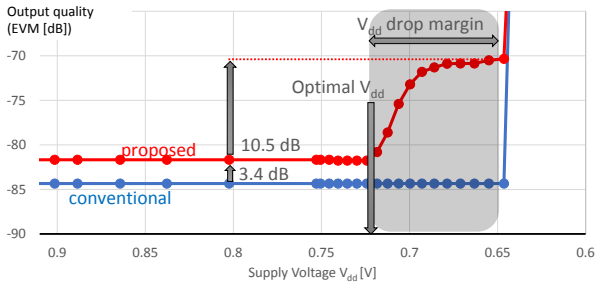
For benchmark, we compare the conventional and the proposed partial CS CORDIC during V_{dd} over-scaling. Both designs are synthesized and routed with standard cells from *TSMC 28nm hpm*.

Because of short-path constraint, the hold time fixing during placement & route is very challenging. Note that increasing duty cycle will increase the V_{dd} over-scaling capability. However, more buffers are inserted to meet the short path constraint, and thus more energy overhead is introduced. In this paper, the timing detection window ratio is set as 25%. The nominal V_{dd} is $0.9V$.

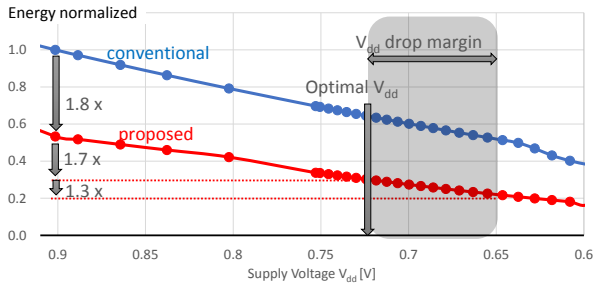
EVM result and energy consumption is estimated from the back-annotated simulation with realistic high bandwidth stimuli, with typical case circuit, i.e. typical process corner at 25 degrees Celsius. Accurate energy consumption during V_{dd} scaling requires re-characterization of the entire library at different fine grained V_{dd}, which takes much computation time. Therefore, in this paper, the timing violation effects of voltage over-scaling is obtained by adjusting the clock frequency during simulation. As the circuit speed is proportional to $(V_{dd}-V_{th})^2/V_{dd}$, it decreases approximately linearly with V_{dd} scaling (If V_{dd} is high above V_{th}). So the clock frequency scaling effects is translated into V_{dd} scaling by the following interpolation method. The gradient of their relation is characterized by Spice simulations on one critical path of the conventional CORDIC, at different coarse-grained V_{dd}s. For instance, the simulation that at 0.9V, 10% reduction on V_{dd} corresponds to 24% reduction on speed.

Fig.9(a) shows the EVM during V_{dd} over-scaling. The EVM for proposed partial CS CORDIC is always 3.4 dB worse than conventional CORDIC, because the last two iterations in the last cycle is always skipped, even no timing violations are detected.

Because of the worst-case design margin and IR-drop design margin, both conventional CORDIC and the proposed partial CS CORDIC produces error-free results until 0.72V. When scaled beyond 0.72V, the conventional CORDIC is still error free. However, the partial CS CORDIC reports error because its setup timing constraint at design time is more relaxed. This leads to timing violations and hence skipped iterations and reduced output quality. Lower V_{dd} results to longer propagation delay and hence more frequent setup timing violations. A maximum of 10.5 dB EVM drop is observed because of timing violation related skipping. For wireless communication applications, the channel quality is varying a lot. Therefore, this EVM drop mode could be enabled once



(a) Output quality. Proposed design reduces quality at V_{dd} drop margin region.



(b) Energy consumption per CORDIC operation. Proposed design saves energy because of relaxed constraint.

Fig. 9. Voltage over-scaling for the converntional CORDIC, and the CORDIC with the proposed partial CS setup timing error correction scheme.

the channel condition is good and ready to compensate the EVM degradation. As a result, automatic control loop based on channel condition is possible to reduce the CORDIC's power consumption.

For the conventional CORDIC, once the V_{dd} is scaled beyond its minimum V_{dd} requirements, i.e. 0.65V, timing violations occur and results to totally unacceptable output quality. For V_{dd} lower than 0.65V, the control paths in the partial CS CORDIC fail. Therefore, the setup timing errors cannot be corrected and yields unacceptable EVM performance.

In sum, the proposed approach yields comparable EVM above 0.73V, and only gracefully degradation when scaling beyond 0.73V. In contrast to conventional CORDICs that have a rigid dependency between V_{dd} and output quality, the partial CS CORDIC operates still functions if supply voltage is above its critical V_{dd} (0.72V). Therefore, the constraint at design time is more relaxed to save energy. From another perspective, it is more robust to V_{dd} over-scaling or accidentally noise related voltage drop.

The benefit of the proposed approach is observed by the energy consumption during V_{dd} scaling (in Fig.9(b)). Since the V_{dd} drop margin can be handled by the error correction unit, the setup timing constraint for the data-path is more relaxed than the conventional design. As a consequence, the EDA tool use gate downsizing to select slower but power efficient cells. Therefore, partial CS CORDIC achieves 1.8 X energy saving at nominal voltage. The error-free design margin related V_{dd} scaling results to 1.7 X energy saving. If output quality degradation is permitted, another 1.3 X energy saving is observed.

For the final chip, the power supply controller periodically adjusts the V_{dd} in low frequency. For the conventional CORDIC, it sets the V_{dd} to the optimal point (0.72V), leaving 70mv (0.72V-0.65V) as the high-frequency V_{dd} drop margin. For the proposed partial CS CORDIC, the V_{dd} is also set to 0.72V. The 70mv design margin is handled by the error correction scheme. As a consequence, the proposed partial CS method enables operating at lower V_{dd} with relaxed setup timing constraint at design time.

VI. Conclusion

In this paper, a partial computation-skip scheme is proposed to mitigate timing errors introduced by supply voltage over-scaling. The proposed partial CS scheme is suitable for evolutionary algorithms. We implemented this scheme in a recursive CORDIC processor. Effectively, the last CORDIC iterations are skipped once timing errors are detected in this proposed partial CS CORDIC. Based on simulation result on a typical-case circuit V_{dd} scaling, we observe 1.7 X energy saving because of design margin, and another 1.3 X energy saving at the sacrifice of 10.5 dB output quality.

References

[1] H. De Man, "Ambient intelligence: gigascale dreams and nanoscale realities," in *Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International*. IEEE, 2005, pp. 29–35.

[2] K. J. Kuhn, "CMOS transistor scaling past 32nm and implications on variation," in *Advanced Semiconductor Manufacturing Conference (ASMC), 2010 IEEE/SEMI*, Jul. 2010, pp. 241–246.

[3] J. T. Kao, M. Miyazaki, and A. P. Chandrakasan, "A 175-MV multiply-accumulate unit using an adaptive supply voltage and body bias architecture," *Solid-State Circuits, IEEE Journal of*, vol. 37, no. 11, pp. 1545–1554, Nov. 2002.

[4] D. Ernst, N. S. Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge, "Razor: a low-power pipeline based on circuit-level timing speculation," in *Microarchitecture, 2003. MICRO-36. Proceedings. 36th Annual IEEE/ACM International Symposium on*, Dec. 2003, pp. 7–18.

[5] B. H. Calhoun and A. P. Chandrakasan, "Standby power reduction using dynamic voltage scaling and canary flip-flop structures," *Solid-State Circuits, IEEE Journal of*, vol. 39, no. 9, pp. 1504–1511, Sep. 2004.

[6] R. Hegde and N. R. Shanbhag, "Energy-efficient signal processing via algorithmic noise-tolerance," in *Low Power Electronics and Design, 1999. Proceedings. 1999 International Symposium on*, Aug. 1999, pp. 30–35.

[7] R. Abdallah and N. Shanbhag, "Error-resilient low-power viterbi decoder architectures," *Signal Processing, IEEE Transactions on*, vol. 57, no. 12, pp. 4906–4917, 2009.

[8] E. P. Kim and N. R. Shanbhag, "Energy-Efficient LDPC Decoders Based on Error-Resiliency," *2012 IEEE Workshop on Signal Processing Systems*, pp. 149–154, Oct. 2012.

[9] K. A. Bowman, J. W. Tschanz, N. S. Kim, J. C. Lee, C. B. Wilkerson, S. L. Lu, T. Karnik, and V. K. De, "Energy-efficient and metastability-immune resilient circuits for dynamic variation tolerance," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 1, pp. 49–63, 2009.

[10] M. Fojtik, D. Fick, Y. Kim, N. Pinckney, D. M. Harris, D. Blaauw, and D. Sylvester, "Bubble Razor: Eliminating Timing Margins in an ARM Cortex-M3 Processor in 45 nm CMOS Using Architecturally Independent Error Detection and Correction," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 66–81, Jan. 2013.

[11] I. Shin, J.-j. Kim, Y.-s. Lin, and Y. Shin, "Architecture with 1-Cycle Timing Error for Low Voltage Operations Correction for Low Voltage Operations," in *Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on*, vol. 3, 2013, pp. 199–204.

[12] Y. Huang, M. Li, C. Li, P. Debacker, and L. V. D. Perre, "Computation-Skip Error Resilient Scheme for Recursive CORDIC," in *2014 SIPS, Workshop on Signal Processing Systems, in press*, 2014.