

Computation-skip error resilient scheme for recursive CORDIC

Yanxiang Huang^{*†}, Meng Li^{*}, Chunshu Li^{*†}, Peter Debacker^{*}, Liesbet Van der Perre^{*†}

^{*}Interuniversity Microelectronics Centre (IMEC), Kapeldreef 75, 3001 Heverlee, Belgium

[†]Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Heverlee, Belgium

Abstract—Aggressive voltage and frequency scaling are widely utilized to exploit the design margin introduced by the process, voltage and environment variations. However, scaling beyond the critical voltage or frequency results to numerous timing errors, and hence unacceptable output quality. In this paper, a computation-skip (CS) scheme is proposed for recursive digital signal processors with a fixed cycles per instruction (CPI) to correct timing errors. A CORDIC processor with the proposed CS scheme still functions when scaling beyond the sub-critical voltage or frequency. It improves EVM by 47.9 dB at its most critical frequency or supply voltage, and extends the voltage scaling limit by 90 mV w.r.t the conventional CORDIC. Besides, it is more than 1.7X energy efficient w.r.t. the conventional high-speed CORDIC, which is designed for a more aggressive scaling.

I. INTRODUCTION

Modern integrated circuits (ICs) are facing an ever increasing challenge resulting from circuit process, voltage and temperature (PVT) variability. The variability result to randomized transistor parameters, e.g. gate width and length, channel mobility, threshold voltage V_{th} . Besides, supply voltage V_{dd} is also fluctuating due to noise or IR drop. As a consequence, the propagation delays of two theoretically identical transistors is different [1]. This phenomena becomes more significant with the scaling of the CMOS technology [2]. This is because i) in a scaled technology node, even tiny process parameter fluctuations have a 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 at the worst PVT corner, to ensure they always operate correctly. Nevertheless, ICs rarely operates at the worst case corner. Therefore, this worst-case approach introduces a design margin, leading to wasted performance capability and power consumption.

Recently, on-chip monitor techniques are proposed to reduce the design margin [3]–[10]. In those ICs, on-chip monitors, e.g. voltage and timing monitors, are equipped to detect the real operating conditions. Then those ICs accordingly adjust the operating situation, e.g. supply voltage V_{dd} , body-biasing V_{sb} , and operating frequency f . They try to find out the critical operating point per chip, where scaling beyond results to timing errors.

Timing check on flip-flops (FFs) have been proposed and widely utilized. This scheme modifies the application inde-

pendent FFs. Therefore, it is orthogonal for different signal processing algorithms. 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 the full exploit of the variability margin. Besides, the circuit is not capable of operating at the sub-critical frequency or V_{dd} situation since no error correction scheme is included.

In contrast, in-situ schemes [4], [5], [9], [10] are proposed to overcome this mismatch for micro-processors. A such scheme requires a timing-error detection scheme (EDS) and an error correction scheme (ECS). For EDS, Razor [4] detects timing error with a shadow latch. DSTB [9] and TDTB [9] replace the main FF in Razor with a latch to eliminate the meta-stability from the data-path. Bubble Razor [10] avoids the short-path constraints in previous designs by adapting the two-phase latch based design.

For ECS, counterflow [4] and instruction replay [9] are proposed. They correct error by issuing extra cycles, which results to multiple-cycle 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 latch, which is not mainstream. Global clock gating scheme [4] also achieves 1 cycle penalty. But the 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 achieve 0-cycle penalty in case of a timing error.

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 consumption and area. With ANT techniques, a remarkable amount of energy are saved at the cost of slightly reduced output quality [7], [8]. However, This scheme need to be tailored to each individual application, as it rely heavily on the characteristics of the target computation. Besides, for recursive algorithms, timing errors introduced by earlier iterations are usually amplified in later iterations, which greatly degrades the output quality.

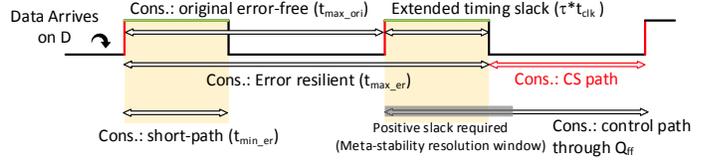
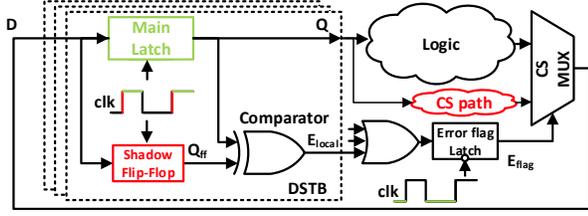


Fig. 1. (a) CS schme implemented in a recursive processor. It consists of logic stage, double sampling with time borrowing (DSTB) [9], and a CS MUX. (b) Timing constraints for the processor with CS schme.

For some real-time streaming applications, e.g. in communication system, a constant throughput is required for the mixed signal circuits. To exploit the design margin in those circuits, this paper propose a computation-skip (CS) scheme to maintain the cycles per instruction (CPI) for without throughput penalty at sub-critical situations. The in-situ timing check are performed in the DSTB [9]. The ECS is a mixture of algorithm and circuit level scheme: it mitigate the error at a very tight timing window to maintains the constant CPI. When timing violations are detected, the computations in the next cycle are skipped as an approximation. This results to a reduced output quality, which is acceptable for certain applications, e.g. wireless communication and video processing, where a good enough result is sufficient.

The proposed CS scheme is demonstrated in a coordinate rotation digital computer (CORDIC) operating in vectoring mode, of which a typical application is a Cartesian to polar coordinate vector translator for polar transmitters. This is the first paper to exploit the trade offs between skipping trivial iterations in recursive CORDIC with frequency and voltage over scaling, and energy efficiency.

II. CS ERROR RESILIENT SCHEME

The concept of the proposed error correction scheme is presented in a recursive processor (see Fig.1). The error detection circuit is a DSTB [9]. The main latch captures the input signal D and produces an output Q . In the unlikely event of timing violations, the positive sensitive shadow FF captures a 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 timing violation in any DSTB is regarded as a timing failing for a DSP. Therefore, all E_{local} are connected with a OR gate to produce a global error flag E_{flag} . A negative 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 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 error-free circuit. As a consequence, the circuit is able to operate at sub-critical frequency or V_{dd} . By tuning the clock duty cycle factor τ (the percentage it stays high), designers can extend the nominal delay constraint $t_{max_original}$

to the error resilient timing constraint t_{max_er} :

$$\begin{aligned} t_{max_ori} &= t_{clk} - t_{setup_FF} \\ t_{max_er} &= t_{clk} + \tau \cdot t_{clk} - t_{setup_latch} \end{aligned} \quad (1)$$

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 of frequency over scaling R_f is denoted as:

$$R_f \triangleq \frac{t_{max_er}}{t_{max_ori}} \quad (2)$$

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

$$\begin{aligned} R_f &= \frac{t_{clk} + \tau \cdot t_{clk} - t_{setup_latch}}{t_{clk} - t_{setup_FF}} \\ &\approx 1 + \tau \end{aligned} \quad (3)$$

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 a result, the correct signal from the previous clock is preserved, and the next logic computation is skipped. This skip can be regarded as a naive implementation for the approximated version of the logic. For the CS path during error correction, signal through Q and through Q_{ff} have different timing constraints (see Fig.5).

The shadow FF will violate the timing constraint at sub-critical situations, it 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 sub-critical situation in this paper, the resolution constant is measured as 15ps. Therefore, we set the slack to 600ps, which is sufficiently large to guarantee mean time before failure due to meta-stability. Note that the main latch can 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 always sensitive during 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:

$$t_{min_er} = \tau \cdot t_{clk} + t_{hold_latch}, \quad (4)$$

where t_{hold_latch} is the hold time for the main latch.

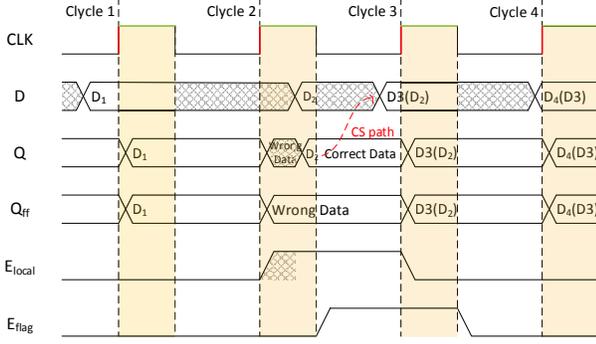


Fig. 2. An example timing diagram for the recursive processor with CS scheme in Fig.1.

A timing diagram example is shown in Fig.2. 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 directly fed to D through the CS path for the next cycle. After cycle 3, all signals are returned to the normal state. In sum, 2 effective iterative calculations are performed during the first 3 cycles due to the timing violation during cycle 2.

III. CORDIC PROCESSOR

CORDIC is a simple and efficient algorithm to calculate trigonometric functions. They are usually equipped in ASIC polar transmitters, for which an otherwise hardware multiplication is relatively power hungry.

A. algorithm

In this paper, we configure the CORDIC 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

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

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

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

where

$$d_i = \begin{cases} 1 & \text{if } y_i < 0 \\ -1 & \text{if } y_i \geq 0 \end{cases} \quad (7)$$

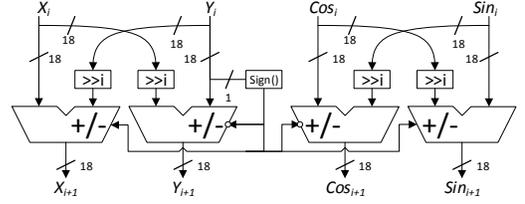


Fig. 3. Vectoring mode CORDIC cell for a CORIC iteration

After n iterations, y_i reaches 0, and the resulted x_n represents the original magnitude. Since x_i and y_i are scaled during each micro rotation, x_n are multiplied with scaling factor $K(n)$ to obtain M :

$$M = x_n \cdot K(n) \quad (8)$$

where $K(n)$ is a function of the number of total iterations n . $K(n)$ is calculated by:

$$K = K(n) = \prod_{i=0}^{n-1} 1/\sqrt{1+2^{-2i}} \quad (9)$$

After several iterations, $K(n)$ does not change much with n :

$$K = \lim_{n \rightarrow \infty} K(n) \approx 0.607252935 \dots \quad (10)$$

Since the number of total iterations are pre-defined, K is a fixed factor. Therefore, the computation in (8) is saved in hardware, since it can be integrated into later stages of the DSP.

Similar to (6), $Cos(\phi)$ and $Sin(\phi)$ are computed according to d_i :

$$\begin{aligned} Cos(\phi)_{i+1} &= Cos(\phi)_i + Sin(\phi)_i \cdot d_i \cdot 2^{-i} \\ Sin(\phi)_{i+1} &= Sin(\phi)_i - Cos(\phi)_i \cdot d_i \cdot 2^{-i} \end{aligned} \quad (11)$$

where the initial values are set as: $Cos(\phi)_0 = 1$ and $Sin(\phi)_0 = 0$.

Therefore, we couple the computation of $Cos(\phi)$ and $Sin(\phi)$ with the computation of x_i and y_i . Fig.3 shows the hardware diagram of a 18-bit CORDIC iteration, or a CORDIC cell. 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):

$$EVM(dB) = 10 \log_{10} \frac{P_{error}}{P_{signal}} \quad (12)$$

where the signal and error power are calculated as:

$$\begin{aligned} P_{signal} &= x_0^2 + y_0^2 \\ P_{error} &= (M \cdot Cos(\phi) - x_0)^2 + (M \cdot Sin(\phi) - y_0)^2 \end{aligned} \quad (13)$$

Fig.4 shows that for the 18-bit CORDIC, the EVM evolves with each iteration, and saturates at $n = 16$, where n is no longer the bottleneck for the EVM. Note that by keeping K constant to $K(16)$, insignificant quality is reduced comparing with multiplying the realistic $K(n)$, especially for the cases

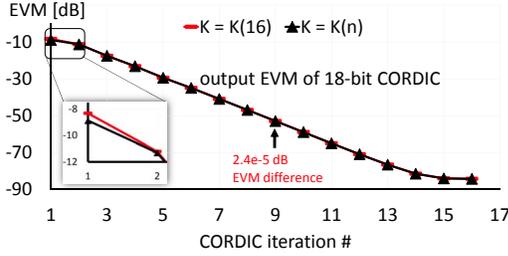


Fig. 4. CORDIC output EVM evolves with different numbers of iterations n when n is large. Therefore, in this paper, K is set to $K(16)$ for different number of total iterations cases, which is more suitable for hardware implementation.

This evolution characteristics from Fig.4 provides space for trading off the iteration number at sub-critical situations. Thus, we propose to apply the CS scheme on the CORDIC application. Once a timing violation is detected, we skip the next cycle and freeze 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. implementation

The trade off between the number of iteration and the EVM is demonstrated for CORDIC. When enhanced with CS scheme, by skipping the final iterations, timing violations are tolerated. The rest explains this method and exploits its trade off in hardware.

The implementation of the CS scheme on a CORDIC processor is shown in Fig.5. 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 ($CPI = 4$). C_{last} is the control signal that indicates whether it is the last cycle. It controls the input MUX and enables the output FF. After each cycle, computation outputs are stored in the DSTB, whose outputs serves as the inputs for the next cycle. C_{itr} counts the CORDIC iterations, and controls the barrel shifter in each CORDIC cell. The computation-skip (CS) MUX selects the signal from the nominal logic, or from the CS path, according to E_{flag} .

When a timing violation is detected, C_{itr} is frozen instead of counted up. As a consequence, the intended 4 CORDIC iteration computation will actually conduct one cycle later, which eats up one cycle duration. As the output CPI is kept constant, the last 4 CORDIC iterations are skipped due to this timing violation. We freeze C_{itr} , because the micro rotation angles in later iterations are smaller, and thus skipping them would less degrade EVM. Moreover, by checking the final C_{itr} of each CORDIC operation, we get the information of how many iterations are skipped. In other words, C_{itr} serves as a quality monitor, which indicates how critical the timing constraint is. Therefore, f , V_{dd} , and V_{th} , etc., can be adjusted at run time based on this quality monitor.

The data paths (CORDIC cells) are synthesized at a nominal frequency. In order to perform frequency or supply voltage

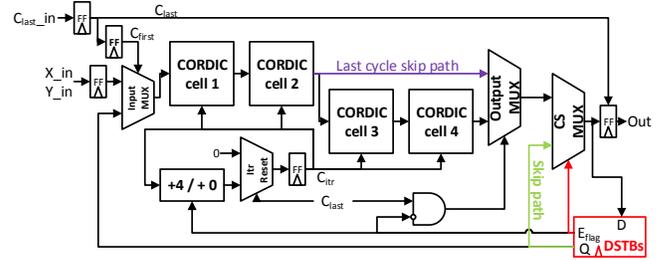


Fig. 5. Proposed CORDIC processor with CS timing error resilient scheme.

over scaling, the control paths are synthesized at a higher frequency (equal to the targeted frequency over scaling limit). Since errors cannot be detected unless a cycle is finished, we cannot correct timing error for the last cycle. To solve this problem, during the last cycle, the outputs from CORDIC cell 2 are utilized instead of cell 4, which greatly reduces the propagation delay, and thus eliminates the occurrence of timing violations. As the last 2 CORDIC iterations are skipped, 3.4 dB EVM loss is expected. However, this loss can be predicted during design time, and hence be compensated by other techniques, e.g. reducing quantization error by increasing the word width.

IV. EVALUATION

A. Evaluation condition

The proposed error resilient CS CORDIC processor (Fig.5) is implemented to evaluate the benefit w.r.t. conventional CORDIC design. For the conventional CORDIC, DSTBs are converted back to normal FFs. Besides, CS MUX and the last iteration protection MUX are also eliminated. All design are synthesized with standard cells from *TSMC 28nm hpmwbp*. They use ULVT to meet the high speed requirement. The nominal V_{dd} is 0.9V. The slow-slow process variation corner is used to ensure the chip operates correctly at the worst case. Energy consumption is estimated with the toggling rate derived with realistic high bandwidth input vectors.

The nominal operating frequency of the conventional CORDIC processors is 750 MHz, which is the best frequency for energy delay product (EDP), as shown in Fig.6. At this frequency, synthesis tool utilizes carry look-ahead adders in the CORDIC cells. So timing errors would occur at all the high order bits when scaling beyond the critical situation [12]. Therefore, the conventional low-speed CORDIC that synthesized at 750 MHz suffers from heavy quality loss. A conventional high-speed CORDIC, which is synthesized at 1000 MHz is utilized as a conventional solution to cope with this loss. It is capable of error-free operation at more aggressively scaled situations. However, it significantly increases energy consumption as the hardware compiler tries very hard in gate up-sizing to meet the setup timing constraint (see Fig.6).

The proposed CS CORDIC, whose data-path is also synthesized at 750 MHz, is however capable of operating at sub-critical situations with a limited amount of errors. Without losing generality, we set the clock duty cycle factor τ to 0.33.

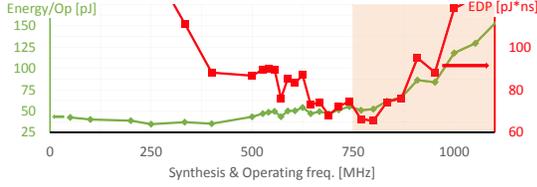


Fig. 6. Energy consumption per CORDIC operation and energy delay product (EDP) for conventional CORDIC with different speeds.

According to (3), the frequency over scaling limit is 1.33X of the original frequency, i.e. 1000 MHz. The following investigates the output quality (in EVM) and energy consumption from post synthesis gate level netlist.

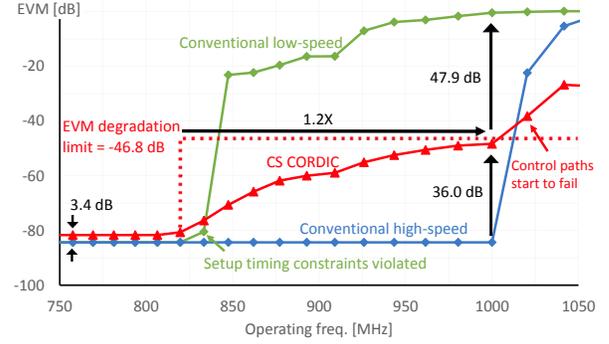
The short path constraints for CS CORDIC is fixed by inserting buffers during post clock tree synthesis optimization, along with other hold violation fixing. Note that increasing duty cycle will increase the over scaling capability, however, more buffers are inserted to meet the short path constraint, and thus more energy overhead is introduced.

During simulation, the meta-stable output state of a setup time or hold time violated FF is interpreted to a be random state of 0 or 1 (or to the previous state). The reason of this approximation is to avoids the situation where the output is unknown, which is impossible for EVM evaluation. This actually improves the simulated EVM for conventional CORDICs, where some timing violated or meta-stable signals are accidentally interpreted to be correct. However, the proposed CS CORDIC do not take advantage of this simulation approximation. This is because, similar as for the real chip, during simulation, the approximation might suggest a false non-error indicator, which let the timing error propagates without mitigation.

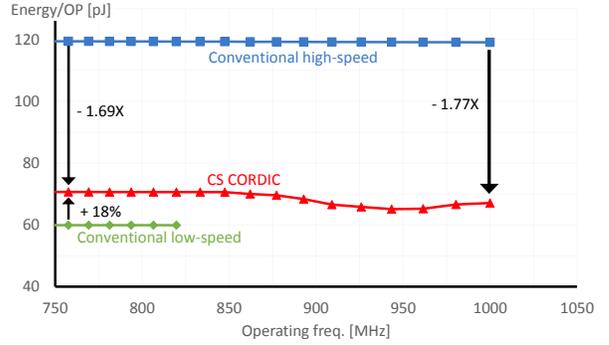
B. Frequency over scaling

From Fig.7, The EVM for the low speed CORDIC (synthesized at 750 MHz) starts to reduce dramatically when frequency increases. Its maximum error-free operating frequency is 833 MHz, rather than the targeted 750 MHz, due to the timing violation approximation during simulation. The conventional high-speed CORDIC functions perfectly until scaling above 1000 MHz.

For the CS CORDIC, during the timing error-free operating, a 3.4 dB penalty in EVM is measured, which is caused by the last cycle protection. The EVM degrades gracefully with the increase of clock frequency, as more timing paths fail and more iterations are hence skipped. The CS error mitigation scheme ensures its correctly operating at 1000 MHz, which is 1.33X of the frequency limit for the conventional low-speed CORDIC. When the operating frequency approximates the frequency scaling limit (1000 MHz), almost all the shadow FFs fail and thus exactly half of the CORDIC iterations are skipped. So the degraded EVM tends to saturate at -46.8 dB. Beyond 1000 MHz, the EVM of the CS CORDIC degrades dramatically, due to the failing of the control paths and the CS path. At 1000 MHz, the proposed CS CORDIC achieves a 47.9 dB EVM improvement w.r.t. the conventional low-speed one.



(a) Output quality



(b) Energy consumption per CORDIC operation

Fig. 7. Frequency over scaling for the conventional low-speed and high-speed CORDICs (synthesized at 750 MHz and 1000 MHz, respectively), and the proposed error resilient CS CORDIC, when $V_{dd} = 0.9V$.

TABLE I
CRITICAL PATH TIMING OF THE CONVENTIONAL LOW-SPEED CORDIC

V_{dd}	0.9V	0.81V	0.72V
Max. Frequency	750 MHz	561 MHz	371 MHz

The energy consumption during frequency over scaling is shown in Fig.7(b). The energy numbers correspond to EVM worse than -46.8 dB are left out because those output qualities are regarded as unacceptable. For high frequency operation, the energy are mostly consumed by the active current. Thus, the energy consumption is proportional to V_{dd}^2 . Thus, the energy consumption is expected to be constant at different operating frequency. However, when the frequency is too high that timing violations happens frequently, e.g. beyond 830 MHz for the CS CORDIC, not all transitions are finished in the limited clock period, so energy consumption drops.

At the 750 MHz (timing error-free frequency), the CS CORDIC has 18% power overhead: 8% penalty is due to the replacement of DSTBs, 8% is introduced by the extra buffers that fix the short path problem, and 2% is due to the tighter time constraint introduced by those MUXs. The CS CORDIC is 1.69X better at energy efficiency at nominal frequency w.r.t. conventional high-speed CORDIC. At 1000 MHz, it reduces energy consumption by 1.77X, at the penalty of 36.0 dB EVM loss. In sum, 1.2X frequency improvement is achieved by the CS CORDIC from the simulation.

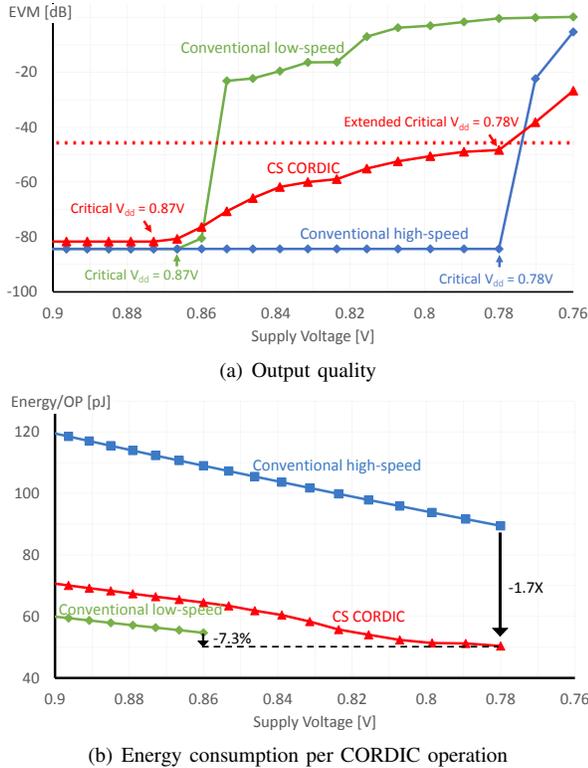


Fig. 8. Voltage over scaling for the conventional low-speed and high-speed CORDICs, and the proposed CS CORDIC, at 750 MHz

C. Voltage over scaling

Fig.8 shows the EVM and energy consumption of supply voltage scaling At 750 MHz. Accurate power consumption and EVM data requires re-characterization of the entire library at different fine grained V_{dd} , which takes much computation time. Therefore, in this paper, voltage over scaling is mapped to frequency over scaling, since they have similar effects on the timing violation situation. As the circuit speed is proportional to $(V_{dd} - V_{th})^2/V_{dd}$, it desegregates approximately linearly with V_{dd} scaling for a high V_{dd} . The gradient is characterized by the spice simulation on one critical path of the conventional low-speed CORDIC, at 3 different V_{dd} (see Table I). It suggests that at 0.9V, 10% reduction on V_{dd} corresponds to 24% reduction on speed.

For conventional CORDICs, scaling the V_{dd} lower than their critical V_{dd} would results to totally unacceptable outputs. In contrast to conventional CORDICs that have a rigid dependency between V_{dd} and output quality, the CS CORDIC operates still functions if supply voltage is above the critical $V_{dd} = 0.87V$. This property makes the CS CORDIC more robust to voltage over scaling or accidentally noise related voltage drop w.r.t. the conventional low-speed CORDIC at $V_{dd} \leq 0.87V$. At the retirements of no control path or CS paths fails, the new critical V_{dd} is extended by 90 mV (from 0.87V to 0.78V). Besides, by scaling V_{dd} to 0.78V, the CS CORDIC reduces energy by 7.3% w.r.t. the conventional low-speed CORDIC. The conventional high-speed CORDIC can also operate at 0.78V since large design margin is guaranteed. However, it consumes 1.7X more energy.

V. CONCLUSION

In this paper, a computation-skip scheme is proposed to mitigate timing errors introduced by frequency and supply voltage over scaling. This scheme is implemented in a recursive CORDIC processor. Effectively, the last CORDIC iterations are skipped once timing errors are detected in this proposed CS CORDIC. The simulated frequency over scaling limit is extended by $R_f = 1.2X$. Moreover, the critical voltage is reduced from 0.87V to 0.78V, resulting to 7.3% energy consumption improvement. With regarding to the conventional high-speed CORDIC, which adds large design margin to protect error, the proposed CS CORDIC is 1.7X energy efficient.

The proposed CS scheme can also be generalized to other evolutionary algorithms, e.g. LDPC. In the future, we expect to add more sophisticated logic to the CS path to exploit the CS path slack, e.g. skipping part of the computation instead of the whole, and to apply this scheme to other non-evolutionary applications.

REFERENCES

- [1] A. Asenov, A. R. Brown, J. H. Davies, S. Kaya, and G. Slavcheva, "Simulation of intrinsic parameter fluctuations in decanometer and nanometer-scale MOSFETs," *Electron Devices, IEEE Transactions on*, vol. 50, no. 9, pp. 1837–1852, Sep. 2003.
- [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. Liu and T. Zhang, "On the selection of arithmetic unit structure in voltage overscaled soft digital signal processing," *Proceedings of the 2007 international symposium on Low power electronics and design - ISLPED '07*, pp. 250–255, 2007.