

A Study on Self-Timed Asynchronous Subthreshold Logic

Niklas Lotze, Maurits Ortmanns, Yiannos Manoli
University of Freiburg

Chair of Microelectronics, Department of Microsystems Engineering - IMTEK
Georges-Koehler-Allee 102, 79110 Freiburg, Germany
{Niklas.Lotze, Maurits.Ortmanns, Yiannos.Manoli}@imtek.uni-freiburg.de

Abstract

This paper investigates self-timed asynchronous design techniques for subthreshold digital circuits. In this voltage range extremely high voltage-dependent delay uncertainties arise which make the use of synchronous circuits rather inefficient or their reliability doubtful. Delay-line controlled circuits face these difficulties with self-timed operation with the disadvantage of necessary timing margins for proper operation.

In this paper we discuss these necessary timing overheads and present our approach to their analysis and reduction to a minimum value by the use of circuit techniques allowing completion detection. Transistor-level simulation results for an entirely delay-adaptable counter under variable supply down to 200mV are presented. Additionally an analytical comparison and simulation of timing and energy consumption of more complex subthreshold asynchronous circuits is shown. The outcome is that a combination of delay-line based circuits with circuits using completion detection is promising for applications where the supply voltages are at extremely low levels.

1. Introduction

The reduction of energy consumption is one of the key challenges for interesting new devices and technologies. Subthreshold circuits therefore are an interesting design alternative due to their ultra-low power consumption, the fact that the point of minimum energy per operation typically lies in this operating region [14], and the possibility of using very low supply voltages. All of these properties also make subthreshold circuits very appealing for energy harvesting applications [12], where energy budgets typically are tight and available supply voltages can be very low. Well regulated supply voltages are rather tedious at these voltage levels though, therefore adaptability to temporarily changing supply voltages is desired.

A straightforward solution for this requirement is the use of matched delay lines to clock the circuits, thereby adapting clock speed to the current supply voltage, as realized e.g. in [2] or especially targeting energy harvesting applications in [1]. Due to thoroughly increasing delay variability with decreasing supply voltage, high safety margins in the delay lines are required when using this concept, degrading circuit speed and increasing leakage energy per computation.

There exist asynchronous circuit techniques which detect when an operation is finished, thus eliminating the need for extra timing overheads. These techniques have shown good self-adaptability to environmental and operating conditions in the superthreshold domain. In this paper, we show that these techniques can also be successfully applied to subthreshold circuits and we analyze when their use is advantageous, which is to the author's knowledge the first time being published. In [5], also asynchronous circuits for the subthreshold domain are addressed, but using a matched-delay concept.

This paper is organized as follows: After this introduction, the choice of methods is motivated by shortly explaining the circuit style used. The next section presents a proof-of-concept realizing a simple test circuit, simulated at transistor-level to ensure functionality. The rest of the paper focuses on a comparison between delay-line timing and circuits allowing completion detection, first determining the timing overhead necessary in a delay line realization, then explaining the modeling concept used for the comparison and finally presenting comparative results. The paper closes with an approximation determining where the use of the circuit styles shown is reasonable.

2. Technical Concept

The two main implementation styles for self-indicating datapath circuits - i.e. circuits which themselves can detect their completed operation - are determining circuit activity by sensing the supply currents or encoding data va-

lidity by use of multi-rail/dual-rail circuits. The former as used e.g. in [4] is not applicable in the subthreshold domain since switching and leakage currents are in the same order of magnitude there, making it impossible to determine when active currents have ceased. Multi-rail technologies follow the idea of one multi-rail value indicating data invalidity. Completion of an operation is shown by all circuit outputs changing to valid values (set phase). A complete cycle is finished by reset of the circuit outputs to the spacer value to separate adjacent data values (reset phase). Dual-rail circuits are the most prominent class of this concept using two rails to encode one bit, one to represent a valid false and the other a valid true value.

Most methods commonly used for the implementation of dual-rail circuits can hardly be transferred to the subthreshold domain as they either use complex (e.g. Null Convention Logic (NCL) as in [9]), pseudostatic or dynamic (e.g. Precharge Half Buffer (PCHB) [6]) gates, all of which can hardly be implemented reliably at the supply voltages desired. Circuit techniques found applicable for subthreshold realization are Delay Insensitive Minterm Synthesis (DIMS), NCL with separate completion detection (NCL-X) and NCL-X with reduced completion detection. DIMS [11] uses C-elements and OR-gates for the implementation of a sum-of-products form of a function for each rail. NCL-X [10] implements the functions for each rail with standard gates, but requires a completion detection (CD) after each pair of gates to detect completion of the reset phase, resulting in rather large and time-consuming CD circuits. This is why NCL-X with reduced completion detection [3] only uses CD at gate outputs before state-holding elements to trigger these when valid data is available, strongly reducing the completion detection overhead and allowing quick reset and a slightly more efficient mapping onto basic gates. This method however does not allow an unambiguous detection of when the reset phase has ended and therefore introduces a weak delay dependency in this phase. Fig. 1 shows a simple example for each of the techniques mentioned.

For a proof-of-concept, DIMS and NCL-X with reduced completion have been used to implement one very safe, high-overhead technique and one less conservative one. For the comparison with delay-line methods in section 4, we focus on NCL-X with reduced completion detection as for the other techniques, it is less likely to get timing advantages since they have higher circuit complexity and a slower reset phase.

3. Proof-of-Concept Circuit

The realized circuit for proof-of-concept is a 4 bit counter with a multiplexer for preset on startup. For the DIMS design, the counter is implemented as a 3-stage ring structure (the minimum number of stages in a ring in this de-

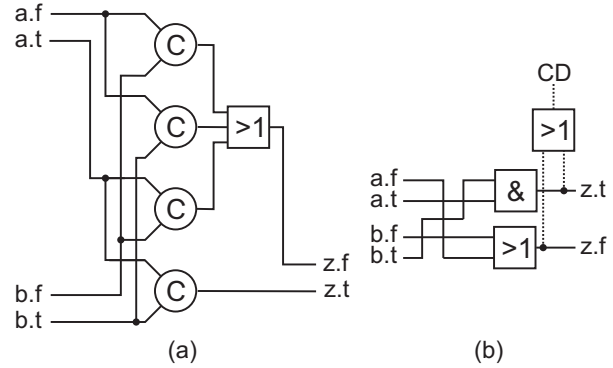


Figure 1. DIMS (a) and NCL-X (b) circuits implementing AND function

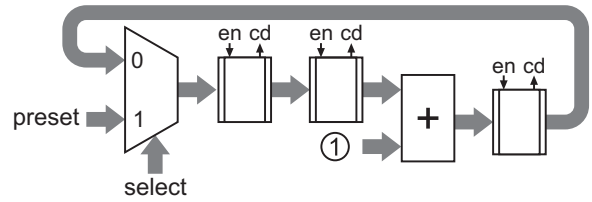


Figure 2. DIMS counter

sign style) using C-elements in latches and completion detectors. Fig. 2 illustrates the structure of the circuit realized, showing latches (controlled by *en*) combined with the respective completion detectors producing the completion detection signal *cd*. Both *en* and *cd* are connected to the control circuitry (not shown here). The NCL-X design uses latches realized with RS-latches connected to the next stage via AND-gates for a quick reset of the input values. As each stage can thus insert reset values by itself, only two stages are required in the ring. The control logic is implemented using simple speed-independent circuits in both designs.

3.1. Gate Design

The gates used in these circuits had to be sized carefully for reliable operation in the subthreshold domain, namely the active-to-leakage current ratios at all process corners being high enough to produce safe output levels. We target a minimum supply voltage of 200mV using regular-V_t transistors in a 130nm technology with typical threshold voltages around 300mV. The gates were sized following the guidelines given by Wang et al. [13]. Besides the standard gates covered there, the C-element is a central building block of asynchronous circuits. Among the common implementation styles for the C-Element, we identified two being feasible for subthreshold design: standard static and symmetric. The symmetric design is clearly superior to the

parameter	standard	symm.
transistor area [μm^2]	0.230	0.290
avg. t_{pd} at 250mV [ns]	38.92	25.56
avg. $E_{switching}$ at 250mV [fJ]	0.177	0.182
avg. I_{Leak} at 250mV [pA]	112.94	140.07

Table 1. Comparison of C-element implementations

standard one for normal operation [8], but this is ambiguous in the subthreshold domain: Even though the symmetric design is advantageous in terms of speed, it has a longer critical transistor chain which requires a bigger transistor sizing for safe low voltage operation. Furthermore, it exhibits higher average leakage currents. Tab. 1 presents an overview of the main parameters, indicating that the standard design should be preferred over the symmetric one for non-critical paths in asynchronous subthreshold circuits.

3.2. Simulation Results

Full transient transistor level simulations have been performed to ensure the usability of the discussed technologies for the subthreshold domain. The circuits have been tested at voltages between 200mV and 500mV, showing safe operation of the devices over the full voltage range and all process conditions. The circuits adapt autonomously to the maximum possible speed at the respective supply voltage, process corner and data input. Simulation results for delay and energy per operation of the ring counter over supply voltage are shown in Fig. 3. The expected exponential dependency between supply voltage and circuit speed could be observed, as well as the typical energy per operation vs. supply voltage behavior with a minimum energy point occurring at rather low supply voltages close to the minimum target V_{dd} . This can be addressed to the high activity present in an architecture with little circuitry within its pipeline stages as the counter exemplarily regarded here. The increased minimum energy point at the slow n - fast p corner observed in Fig. 3 is due to both leakage and cycle time being relatively high at this corner, shifting the optimum to a voltage with reduced cycle time.

Operation from a badly regulated supply as it might occur in energy-harvesting applications has also been simulated and shows a close tracking between supply voltage and circuit speed as seen in Fig. 4. Again, this shows the high robustness and adaptability of the regarded dual-rail asynchronous subthreshold circuits. All figures show the data for the NCLX design with reduced completion detection. Results for the DIMS design are comparable but show higher cycle times and energy-per-computation.

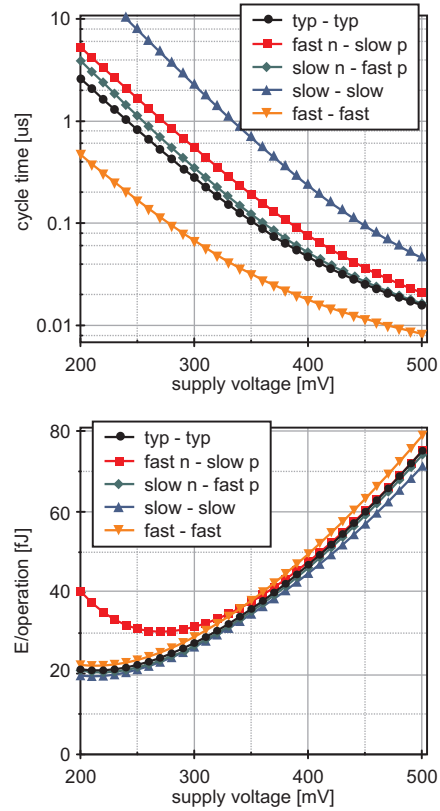


Figure 3. Cycle time and energy per operation for ring counter

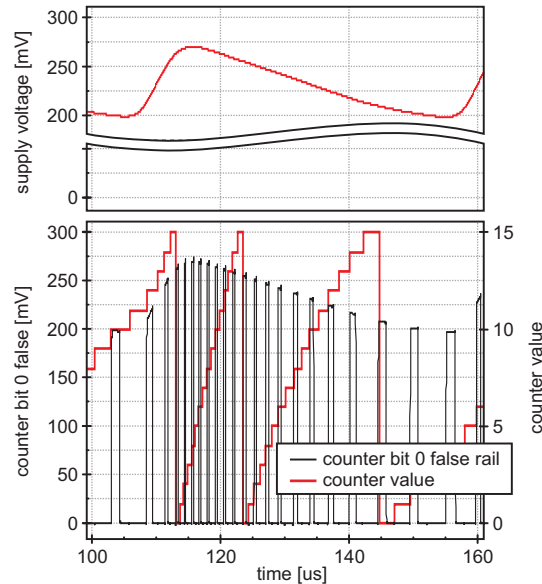


Figure 4. False-rail of LSB at changing supply voltage. Counter values are given to illustrate the dynamic speed change.

4. Timing and Energy in Subthreshold Dual-Rail Systems

After the proof-of-concept circuits have illustratively shown the functionality and robustness of dual-rail circuits in the subthreshold domain, we now investigate timing and energy gains achievable using this technique.

It is obvious that circuits allowing completion detection can eliminate timing overheads, thereby increasing circuit speed, but since this detection requires significant overhead, it is not as obvious how this may decrease energy-per-computation. The energy used for a computation E_{comp} can be written as

$$E_{comp} = C_{eff} V_{dd}^2 + I_{leak,eff} V_{dd} t_{comp}$$

where C_{eff} is the capacitance switched per operation, V_{dd} the supply voltage, $I_{leak,eff}$ the average leakage current of the system and t_{comp} the average time required for an operation. The use of self-indicating circuits on the one hand may reduce t_{comp} , thus decreasing E_{comp} , on the other hand, it introduces switching and area overheads, increasing C_{eff} and $I_{leak,eff}$. This makes it rather unlikely to reduce the energy-per-computation when using a pure dual-rail implementation.

Our concept therefore is to use self-indicating techniques only in the most critical i.e. slowest parts of a system - this increases the system speed, reduces the computation time and therewith also the leakage current per computation. The surrounding circuit can either be asynchronous single-rail circuits using delay lines, rather seamlessly integrating the dual-rail part, or a synchronous system clocked from a single delay-line. In this case, the dual-rail part can e.g. stall an operation if not finished yet.

Our analysis strategy for the optimal use of self-indicating and surrounding, delay-line based circuitry is the following: The implementation strategies for the critical system part are analyzed, basically comparing the timing strategies shown in Fig. 5. This gives estimations on the possible timing gains and necessary overheads of a dual-rail implementation compared to a delay-line single-rail circuit, which can then be used for a discussion on a system incorporating both techniques. For a sound comparison basis, it is though necessary to first determine which timing overheads are necessary in a delay-line system.

4.1. Timing Overheads in a Delay-Line System

When using a standard single-rail circuit with a matched delay line (Fig. 5b) it is important to note that both, the delay line and the circuit, have a certain delay variability.

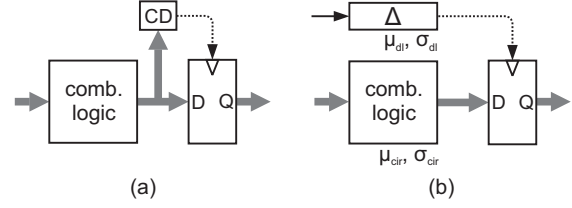


Figure 5. Timing strategies compared: using completion detection (a) or using delay line (b)

For a safe operation of the circuit, it is necessary that the probability:

$$P(T_{pd,dl} > T_{pd,cir}) > \alpha$$

Thereby, $T_{pd,dl}$ and $T_{pd,cir}$ are the propagation delays of the delay line and the circuit (including the necessary setup times for the state-holding elements) and α the desired safety that the circuit operates correctly (e.g. $3\sigma \rightarrow \alpha = 99,7\%$). As shown in [15] and also confirmed by our simulations, the delay variability of a circuit operated in the subthreshold region can approximately be modeled as log-normally distributed, thus having a probability density following

$$f(t, \mu_L, \sigma_L) = \frac{1}{t \sigma_L \sqrt{2\pi}} \exp\left(-\frac{(\ln t - \mu_L)^2}{2\sigma_L^2}\right)$$

with

$$\mu_L = \ln(\mu) - \frac{1}{2} \ln\left(1 + \frac{\sigma^2}{\mu^2}\right), \quad \sigma_L = \sqrt{\ln\left(1 + \frac{\sigma^2}{\mu^2}\right)}$$

where μ and σ are the mean value and variance of the circuit delay considered, namely μ_{dl}, σ_{dl} for the delay line and μ_{cir}, σ_{cir} for the circuit. The above stated problem can thus be expressed as

$$P(T_{pd,dl} > T_{pd,cir}) = \int_0^\infty f(t_{cir}, \mu_{L,cir}, \sigma_{L,cir}) \cdot \left(\int_{t_{cir}}^\infty f(t_{dl}, \mu_{L,dl}, \sigma_{L,dl}) dt_{dl}\right) dt_{cir} > \alpha$$

To our knowledge, there is no closed-form solution to this equation. To simplify the problem, we use $\mu_{dl} = K \mu_{cir}$ for the delay line with K expressing the necessary delay overhead - i.e. how much slower the delay line must be to always ensure reliable operation of the circuit. We use the approximation $\sigma_{dl} \approx \sqrt{K} \sigma_{cir}$ which is reasonable as the delay line is typically designed as critical-path replica leading to a close correlation between the delay time variance of the circuit and the delay line. With these simplifications the above equation only depends on the relative variance $\frac{\mu_{cir}}{\sigma_{cir}}$ of the circuit - the ratio of the mean value of the circuit delay $T_{pd,cir}$ and the standard deviation of this delay.

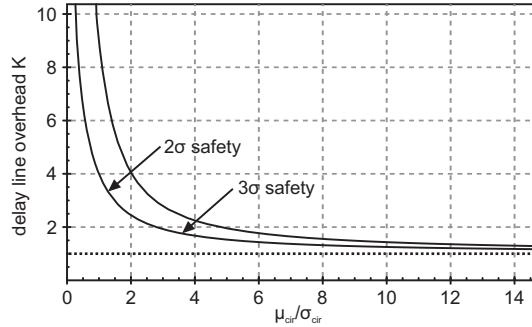


Figure 6. Delay line overhead

The numeric solutions for the necessary overhead at safety levels of 3σ and 2σ are shown in Fig. 6.

Realistic values for $\frac{\mu_{cir}}{\sigma_{cir}}$ seen in subthreshold circuits range between 1 for single gates and 5-15 for larger circuits, depending on circuit depth. Even larger circuits thus require approx. 30% delay line overhead. For practical applications, a 3rd order polynomial fit for 3σ safety is used with

$$K \left(\frac{\sigma_{cir}}{\mu_{cir}} \right) \approx 1 + 3.860 \frac{\sigma_{cir}}{\mu_{cir}} + 4.831 \left(\frac{\sigma_{cir}}{\mu_{cir}} \right)^2 - 0.708 \left(\frac{\sigma_{cir}}{\mu_{cir}} \right)^3$$

introducing an error of less than 2% for $1 < \frac{\mu_{cir}}{\sigma_{cir}} < 100$.

4.2. Modeling

The proof-of-concept circuit in section 3 uses transistor-level modeling, a concept severely restricting circuit size due to otherwise huge simulation runtimes. Modeling of larger circuits is required though, as small circuits are dominated by delays in control and state-holding elements and not the datapath delay we aim to improve. It is however key to accurately represent the delay variability prevalent in the subthreshold regime. Statistical static timing analysis could be used for static purposes, but is not capable of determining dynamic effects like the switching activities in the circuit.

Our approach therefore is to shift the burden of Monte-Carlo-Simulations from transistor level to logic level. As previously mentioned, the variability of gate delays can be modeled as lognormal distribution. Consequently, we performed extensive Monte-Carlo-simulations at transistor level for each of the gates used to get the respective mean and variance values for the propagation delays from any input to the output. Therefrom, custom VHDL models have been implemented which choose a random value - following the lognormal distribution - for each input-output propagation delay at each gate used during the simulation. An extra signal is used in the model to reset to a new configuration, thus allowing quick reconfiguration during a running VHDL simulation.

4.3. Reference Circuit

A multiplier circuit has been used as reference circuit for a comparison of a single-rail (SR) and dual-rail (DR) implementation. The multiplier implementation is aimed at minimum depth to avoid a comparison where the SR circuit could be accelerated more easily by an optimized architecture. The circuit realized is made up of a standard NAND-matrix and a Wallace-Tree-Adder which is terminated by a Kogge-Stone-Adder. The input word width is varied from 4 to 8 bits to analyze the effects of changing circuit depth and size.

4.4. Simulation Results on the Reference Circuit

The results shown are based on Monte-Carlo-Simulations on gate level by using the custom VHDL models outlined in section 4.2. The simulation scheme is to set a random timing configuration and simulate it with 5k random input patterns, repeated for 5k timing configurations. The data used for mean and variance of gate delays is the one for the minimum targeted supply voltage of 200mV, as delay variability is maximized there and the circuits need to be designed for these worst-case conditions.

TIMING: As seen in Fig. 7 (a) for the SR implementation, the propagation delays at a single timing configuration show a roughly normal distribution, representing the dependency of the delay on the input data. It exhibits obvious peaks which can be ascribed to the discrete nature of the input data (the various input patterns to the multiplier). This distribution is analyzed for each of the 5k timing configurations simulated, whereupon the maximum value has to be considered in the SR case (as the delay line needs to be matched to the input data causing maximum delay) whereas the mean value is taken in the DR case. These values form another distribution which is not exactly lognormal, but can be rather well approximated as lognormal distribution, as proposed in [15]. This is shown in Fig. 7 (b) for SR, thus considering maximum values. Fig. 8 shows the mean values of the respective distributions for SR delay, DR set time, completion detection delay and the propagation delay for the reset pulse from the input to the CD output, along with the safety margin necessary for the delay line in the SR case. A complete reset - which safely sets all gate outputs to the spacer value - would take even longer than the reset propagation delay shown. But even for these values a complete set-reset cycle would take about as long as the SR delay line, nullifying the timing advantage of the DR circuit (except if it was not required in every cycle, allowing reset during the idle cycles).

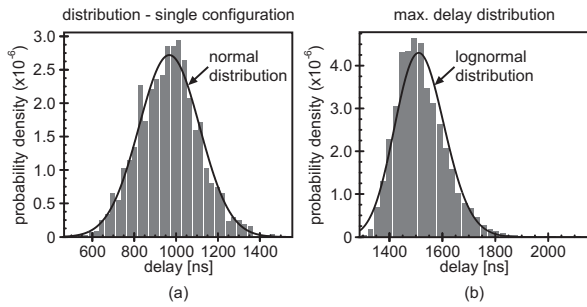


Figure 7. Delay probability distributions for SR circuit.

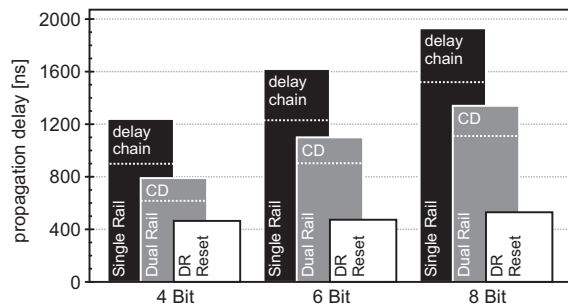


Figure 8. Delay times for SR and DR implementations.

RESET TIME: The time for reset can be significantly reduced though, as it is sufficient to provide a safe reset wavefront traveling through the circuit, securely separating two sets of data. The following needs to be ensured:

- a reset pulse of sufficient length has to occur at the CD output
- when the CD output sets, the data has to be valid

Fig. 9 shows the minimum reset pulse width at the CD output, the probability of a CD set error (CD setting without the output data being valid) and the worst-case CD-to-output slack (delay from the moment when CD sets to the output data being valid) plotted for changing input reset pulse width. The confidence levels used for the SR circuit timing are applied here as well. These results indicate that the reset time can be reduced by approx. 50% without sacrificing reliability of operation compared to the SR circuit.

COMPLETION DETECTOR SIZE: Further optimizations are possible for the completion detection circuit, as some outputs may be omitted from completion detection: those which never are latest outputs and those which can be latest outputs, but whose maximum slack (delay from last changing output considered for completion detection to regarded output) is certainly smaller than the CD delay.

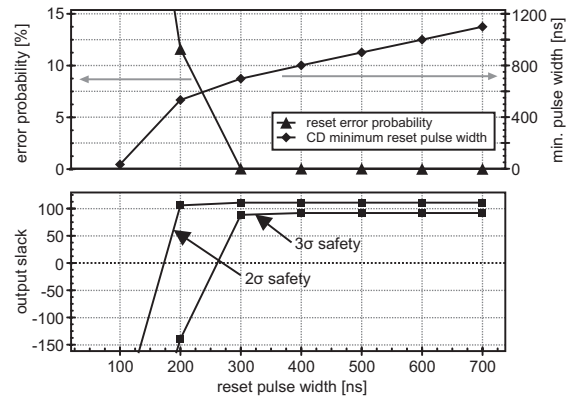


Figure 9. Effect of changing reset pulse width.

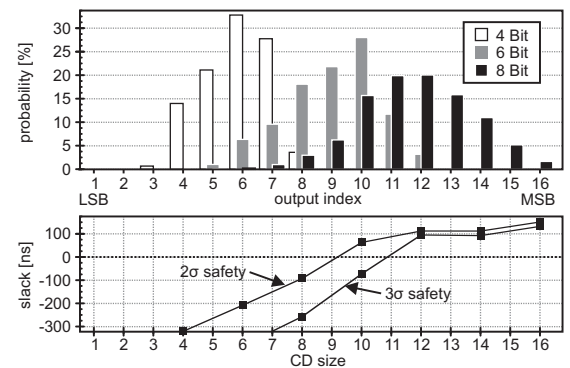


Figure 10. Probabilities for output to be last changing output and effect of changing CD size for the 8 bit case.

Figure 10 shows the probability for each of the outputs of the exemplary multiplier circuit to be the last changing output and the maximum slack seen when omitting the n least probable outputs from completion detection. The results indicate that the output's LSBs can be omitted from completion detection for the regarded circuit. The number and position of negligible outputs for CD strongly depends on the structure of a particular circuit though, in our case being due to the fact that the negligible LSBs are direct outputs of the Wallace Tree and not of the Kogge-Stone-Adder. Omitting outputs with significant probability of being latest output leads to high slack values and therefore is not advisable, revealing that mostly all outputs are critical.

Recapitulating the timing results shown, we can expect about 15-20% cycle time reduction for the DR circuit, adding up DR set time, optimized DR reset time and CD delay for the reduced CD size. This value may change for a real implementation as the control circuits for the regarded timing alternatives are not considered yet. Nonetheless, this

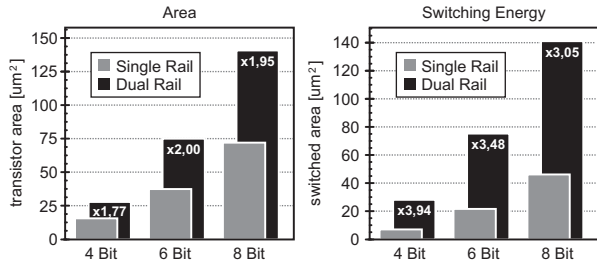


Figure 11. Overhead of DR implementation.

figure is a good estimate for the achievable cycle time reduction and also will be used in the energy per operation analysis in section 4.5.

AREA AND ENERGY OVERHEAD: Area and energy overhead for the DR compared to the SR implementation is shown in Fig. 11. The factor of 2 for the area overhead is what we expect from the structure of the DR implementation. The overhead in switching energy is explained as follows: The activity factor at each input of the SR circuit is $\frac{1}{2}$ for random input signals. Each SR input corresponds to a pair of inputs in the DR implementation, one of which certainly is set and reset in each cycle, corresponding to an activity factor of 2 for the input pair, resulting in the overhead of 4 seen in the simulation. The best case for SR energy consumption has been assumed in this analysis, though: if the single rail inputs change to a definite value after each computation, e.g. because they are connected to a bus or a multiplexer, SR activity doubles and the overhead decreases by a factor of 2. The decreasing energy overhead with increasing circuit size can be attributed to the glitching activity in the SR circuits, amplified by the high delay uncertainties. In bigger SR circuits, these glitches spread further, thus increasing energy consumption. The DR circuit, in contrast, is glitch-free by design.

4.5. Minimum System Size for Energy-per-Operation Reduction

From the last statement in section 4.4 it is concluded that switching energy and area overhead are high in the DR implementation, thus analysis on when its use as part of a larger system pays out in terms of energy-per-computation is necessary. Qualitatively speaking, energy-per-operation can be reduced if a sufficiently large system is substantially accelerated by the use of a small DR part, thus overcoming the switching energy overhead in the DR implementation by reduction of leakage energy per operation. The critical parameters thus are the size ratio of the critical part to the complete system, sf , the area and active energy overheads in the DR part, c_{area} and c_{act} , the cycle time reduction c_{comp} and the fraction of active energy per operation ae indicating how much energy can be saved by circuit speed-up.

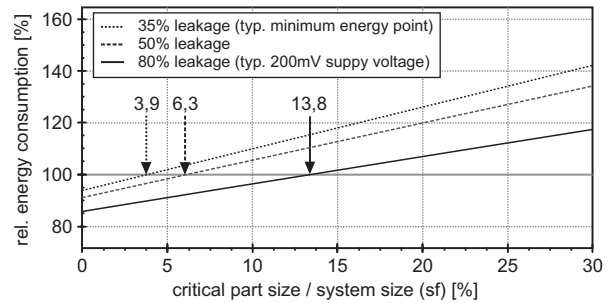


Figure 12. Impact of system size on DR energy balance.

Using the approximation of an area-proportional leakage and an approximately uniform activity in the SR circuit, we get the following equation relating energy-per-operation in a part DR implementation to the one in a pure SR circuit :

$$\frac{E_{comp,DR}}{E_{comp,SR}} = \frac{(c_{act} + \frac{1}{sf} - 1) ae + (c_{area} + \frac{1}{sf} - 1) c_{comp} (1 - ae)}{\frac{1}{sf}}$$

The factor ae depends on the system architecture and the supply voltage used. Fig. 12 shows plots for some typical active-to-leakage ratios as reported in literature for larger systems (e.g. [7]) and using the c_{act} , c_{area} and c_{comp} values acquired for the 8-Bit multiplier.

These results indicate that energy-per-operation can be reduced if the critical part realized in DR accounts for less than $\sim 14\%$ of the overall system when operating at very low supply voltages, whereas it is limited to $\sim 4\%$ at the minimum energy point. The shift of the minimum energy point due to changed activity is not taken into account as this effect is small if the DR part is small compared to the overall system. The maximum achievable reduction in energy-per-operation ranges between 5% for operation at the minimum energy point and 15% when working at very low voltages.

5. Conclusions

This paper presents our work on evaluation and optimization of self-timing in asynchronous subthreshold circuits. The feasibility of dual-rail circuits in the subthreshold domain has been successfully shown for the first time to our knowledge. If not overly conservative designed this technique has the potential to significantly increase the asynchronous circuit speed. In order to also reduce the combined energy per computation, descriptive and analytical expressions are presented which show a strong dependency of the size of the used dual-rail circuitry to the overall circuit and in addition to the supply voltage. Especially, for very low, subthreshold supply - as commonly seen in energy harvesting devices - our approach becomes obviously advantageous.

It can therefore be concluded that the combination of delay-line based and self-indicating asynchronous circuits in the subthreshold domain is a promising circuit technique for applications where either very reliable circuit timing is required or where the supply voltages are at extremely low levels.

Acknowledgments. This work is supported by the Deutsche Forschungsgemeinschaft (German Research Foundation - DFG) under Grant Number 1103.

References

- [1] R. Amirtharajah, J. Wenck, J. Collier, J. Siebert, and B. Zhou. Circuits for energy harvesting sensor signal processing. In *Proc. DAC*, pages 639–644, 2006.
- [2] B. Calhoun and A. Chandrakasan. Ultra-dynamic voltage scaling (udvs) using sub-threshold operation and local voltage dithering. *Solid-State Circuits, IEEE Journal of*, 41(1):238–245, January 2006.
- [3] J. Cortadella, A. Kondratyev, L. Lavagno, and C. Sotiriou. Coping with the variability of combinational logic delays. *Proc. ICCD*, pages 505–508, October 2004.
- [4] M. E. Dean, D. L. Dill, and M. Horowitz. Self-timed logic using current-sensing completion detection (cscd). *J. VLSI Signal Process. Syst.*, 7(1-2):7–16, 1994.
- [5] N. Jayakumar et al. A pla based asynchronous micropipelining approach for subthreshold circuit design. In *Proc. DAC*, pages 419–424, 2006.
- [6] A. Martin et al. The lutonium: a sub-nanojoule asynchronous 8051 microcontroller. *Proc. ASYNC*, pages 14–23, May 2003.
- [7] L. Nazhandali et al. Energy optimization of subthreshold-voltage sensor network processors. *SIGARCH Comput. Archit. News*, 33(2):197–207, 2005.
- [8] M. Shams, J. C. Ebergen, and M. I. Elmasry. Modeling and comparing cmos implementations of the c-element. *IEEE Trans. Very Large Scale Integr. Syst.*, 6(4):563–567, 1998.
- [9] G. Sobelman and K. Fant. Cmos circuit design of threshold gates with hysteresis. *Proc. ISCAS*, 2(31):61–64, June 1998.
- [10] D. Sokolov. *Automated synthesis of asynchronous circuits using direct mapping for control and data path*. PhD thesis, University of Newcastle upon Tyne, 2005.
- [11] J. Sparsø and J. Staunstrup. Delay-insensitive multi-ring structures. *Integr. VLSI J.*, 15(3):313–340, 1993.
- [12] D. Spreemann, Y. Manoli, B. Folkmer, and D. Mintenbeck. Non-resonant vibration conversion. *J. micromech. microeng.*, 16(9):169–173, 2006.
- [13] A. Wang and A. Chandrakasan. A 180mv fft processor using subthreshold circuit techniques. *Tech. Dig. ISSCC*, 1:292–529, February 2004.
- [14] A. Wang, A. Chandrakasan, and S. Kosonocky. Optimal supply and threshold scaling for subthreshold cmos circuits. In *Proc. ISVLSI*, pages 7–11, 2002.
- [15] B. Zhai, S. Hanson, D. Blaauw, and D. Sylvester. Analysis and mitigation of variability in subthreshold design. *Proc. ISLPED*, pages 20–25, August 2005.