

Trading Digital Accuracy for Power in an RSSI Computation of a Sensor Network Transceiver

Paul Detterer¹, Cumhur Erdin¹, Majid Nabi¹, José Pineda de Gyvez¹, Twan Basten^{1,2}, Hailong Jiao¹

¹Department of Electrical Engineering, Eindhoven University of Technology, the Netherlands

²ESI, TNO, Eindhoven, the Netherlands

{P.Detterer, C.Erdin, M.Nabi, J.Pineda.de.Gyvez, A.A.Basten, H.Jiao}@tue.nl

Abstract—To handle the rigid power and energy constraints in the Digital BaseBand (DBB) of Wireless Sensor Networks (WSN)s, we introduce approximate computing as a new power reduction method. The Received Signal Strength Indicator (RSSI) computation is a key element in DBB processing. We evaluate the trade-off in RSSI computation between Quality-of-Service (QoS) and power consumption through circuit-level approximation. RSSI elements are approximated in such a way that error propagation is minimized. In an industrial 40-nm CMOS technology, substantial energy savings up to 24% are achieved for every successfully transferred bit in DBB processing in a low-power listening WSN scenario.

Keywords—Digital Baseband, Approximate Computing, Clear Channel Assessment

I. INTRODUCTION

Energy consumption is the main bottleneck in many Wireless Sensor Network (WSN) applications. Low-power wireless nodes are often difficult to access and need to be very small and light-weight. Without access to strong power sources, those wireless nodes are required to operate without maintenance for long time period. This challenge is addressed in different domains, such as low power circuit designs and low power communication protocols.

Conventional power reduction techniques have already been extensively used in wireless nodes. Even with the combination of protocol and circuit-level power reduction techniques in the analog and digital domains of the WSN transceiver, power consumption is still a major challenge in many emerging WSN applications, such as extremely tiny body sensors. Thus, new energy reduction techniques are necessary. At the circuit level, Analog Front Ends (AFE) have been developed [1] requiring only a few milliwatts for RF signal conversion to the baseband and proper signal filtering. With the AFE being very power efficient, the digital domain starts to become a significant power consumer and needs more attention during the power planning of the wireless transceiver. Though typical energy reduction techniques in the digital domain such as power gating, clock gating, voltage scaling, and technology scaling are effective, it is a challenge to reduce power to micro-watt level. The micro-watt power levels enable new interesting WSN applications with extremely small and light sensors without need of maintenance. At the network level, power efficiency has been one of the main concerns in communication protocols and standards. Many state-of-the-art low-power communication techniques are incorporated in communication standards such as IEEE 802.15.4 [2], which is designed for low-power, low-cost, and low-rate

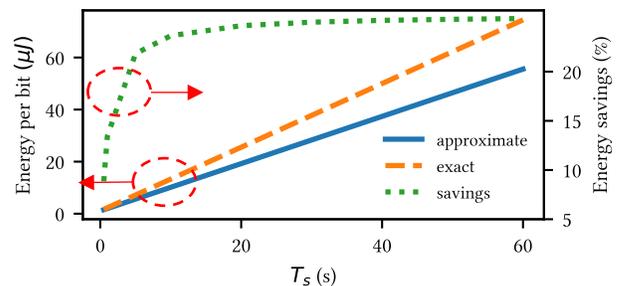


Fig. 1 Example of overall DBB energy reductions for every successfully transmitted and received bit over the sampling period T_s with approximate RSSI designed with adequate computing techniques.

WSN applications. Commercial IEEE 802.15.4 conformant transceivers, e.g. [3], exhibit outstanding energy efficiency, yet find their limitations in extremely small sensor nodes.

Meanwhile, inexact computing techniques, e.g. approximate computing, appear to provide a complementary way to reduce energy beyond conventional limits. Approximate computing emerges in domains such as image processing and provides significant power/energy benefits in exchange of insignificant degradation in image quality. Approximate computing typically involves inexact arithmetic circuits which sometimes produce incorrect computation results, but are simpler and hence, faster and more energy efficient. If the error behavior is bounded to tolerable values, the circuit can be used in computation with insignificant application-level performance degradation.

To push the energy consumption of the transceiver's digital circuitry to below the state-of-the-arts, we investigate the potential of *adequate computing*. Adequate computing applies inexact computations similar to approximate computing to the digital baseband in a WSN transceiver. Circuit-level inaccuracies can be hidden at the network level, trading off an acceptable, insignificant QoS degradation for additional power benefits (hence, the name *adequate computing*).

We develop a Received Signal Strength Indicator (RSSI) circuit as a proof of concept, illustrating how a typical baseband computation can be approximated, what the pitfalls are, and how error interaction and accumulation effects can be identified and mitigated. The approximate RSSI design is analyzed in a star network topology running the non-beacon enabled IEEE 802.15.4 protocol. Significant energy savings can be achieved as shown in Fig. 1. The energy is normed by all *successfully transmitted and received* bits of information, so that performance degradation is also included.

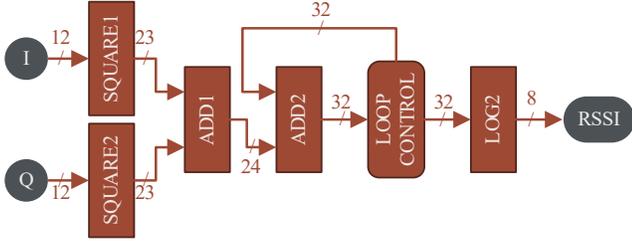


Fig. 2 RSSI computation data flow using In phase (I) and Quadrature (Q) values.

The rest of the paper is organized as follows. In Section II, relevant literature is surveyed. In Section III, RSSI computation and its reference design is discussed. Our approximation approach is presented in Section IV. The approximate RSSI analysis results are given in Section V. The paper is summarized in Section VI.

II. BACKGROUND

State-of-the-art WSN transceiver design: Without loss of generality, the focus of this work is on IEEE 802.15.4 conformant transceivers as they are the most representative transceivers in WSNs. The state-of-the-art transceiver is presented in [4], which can operate in an IEEE 802.15.4 or Bluetooth smart conformant network. The authors of [4] are one of the first few to focus on the digital part of a transceiver because there the flexibility is provided for the design of a multi-standard transceiver (Bluetooth smart and IEEE 802.15.4). Though the AFE is still the main power consumer in the transceiver of [4], it becomes less significant for very short communication ranges than the range specified by Bluetooth or IEEE 802.15.4 standards. This is because the power consumption in digital domain does not scale down by reducing the transmission power. In medical WSN applications, for instance, the typical communication range is two meters instead of the maximal 200 meters specified by IEEE 802.15.4. For further power reductions in such cases, an energy efficient digital part of transceiver is necessary.

State-of-the-art RSSI design: RSSI is a key element in state-of-the-art transceivers. RSSI is used for diverse purposes, such as energy efficient routing [5], hand-off [6], and indoor localization [7]. Traditionally, RSSI estimation is performed in the analog domain (e.g., in [8]). This approach is efficient for older technologies and flexible enough for conventional WSN applications. However, emerging applications demand more flexibility and have stricter energy constraints, while technology scaling is exhausted in the analog domain and cannot offer significant energy improvement [9]. Hence, state-of-the-art RSSI circuit implementations are gradually transferred to the digital domain. The state-of-the-art transceiver of [4] uses digitized In phase and Quadrature (I and Q) values from the main signal recovery path to calculate the RSSI signal entirely in the digital domain.

State-of-the-art approximation: Approximate computing is a technique that allows to trade the computation accuracy for power/energy/speed improvement by using inexact circuits or/and computational algorithms. These techniques are already successfully practiced in image processing. For example, in [10], a 2-bit multiplier is proposed as a building block for bigger

multipliers. The circuit complexity of the multiplier is reduced drastically by allowing an error only when both inputs are 3. The output image of a filter designed with those multipliers can be barely distinguished from the exact filter output by human eyes, while the approximate filter consumes only half the power compared to the exact filter. Since [10], approximate computing evolved significantly. The multipliers and adder described in [11] and [12] show significant power and speed benefits in exchange for negligible quality degradation in image processing. However, approximate computing techniques have not been studied in DBB processing yet. Though the computations are similar, the error impact on DBB processing is not straightforward, which is challenging yet interesting. We investigate approximation techniques to trade-off power consumption in RSSI for relaxed Quality-of-Service (QoS) margins for applications which can tolerate QoS degradation.

Adequate computing identifies error resilience and uses inexact circuits and algorithms, e.g. approximate computing, to trade accuracy for energy. The presented evaluation flow gives insight into approximation mechanics in DBB processing and gives better understanding how to optimize the given approximation.

III. RSSI REFERENCE DESIGN

The RSSI computation in digital baseband is based on the estimation of energy sensed at the targeted channel. This computation can be implemented in different ways. In this paper, the approach of [4] is pursued and implemented in the digital domain with digitized I and Q values as input. For a fair evaluation of the approximation effect, first, a reference is designed and evaluated as the exact digital RSSI computation algorithm. We assume 12 bits for the bit width quantization for I and Q values, and 16 MHz for the DBB clock frequency. Those design assumptions satisfy the dynamic range and minimum sampling rates, to receive and decode an IEEE 802.15.4 signal with conventional communication techniques [2], [13]. Without loss of generality, for one computation period, an IEEE 802.15.4 symbol duration of 16 μ s is taken. The energy estimation from I and Q values can be computed by (1).

$$RSSI(I, Q, N) = 10 \log_{10} \left(\frac{1}{N} \sum_{i=0}^{N-1} [I_i^2 + Q_i^2] \right). \quad (1)$$

The sum $I^2 + Q^2$ is averaged over N samples. The result is expressed in dB which is the conventional unit for the communication domain. (1) is optimized to reduce the cost of the RSSI algorithm. The cost of \log_{10} operation is decreased by using \log_2 , because gate-level arithmetic works with binary numbers. When the number of samples (N) is known, division by N can be simply implemented by using subtraction with a constant number. These optimizations yield (2).

$$RSSI_{bb}(I, Q, N) = \log_2 \left(\sum_{i=0}^{N-1} [I_i^2 + Q_i^2] \right) - \log_2 N. \quad (2)$$

The RSSI dataflow is shown in Fig. 2. Considering that I and Q are integer values, the minimum RSSI value ($RSSI_{bb,min}$) is $-N$, while the maximum ($RSSI_{bb,max}$) is $\log_2(2I_{max}^2)$. Note that the average of 0 is treated specially by \log_2 computation and hence has a special value 0 in our implementation.

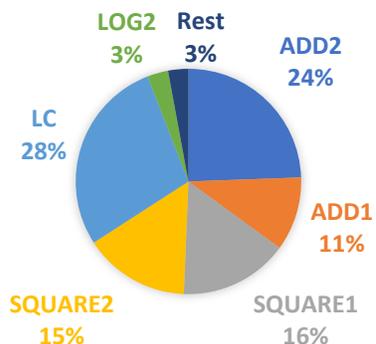


Fig. 3 RSSI computation power breakdown.

In case of a sampling rate of 16 MHz, the averaging for the whole symbol period demands $N = 256$ samples. If I and Q inputs are signed and have a bit width of 12 bits, then the precise adder in the accumulation loop has a bit width of 24 bits at one input and 32 bits at the other input. The dynamic range of 80 dB specified by IEEE 802.15.4 is quantized with an 8-bit integer. For the conversion between $RSSI$ and $RSSI_{bb}$, the physical range of recoverable signals (-80 dBm, 0 dBm) from IEEE 802.15.4 is mapped to (-3, 23), with $RSSI_{bb,max} = 23$ and $RSSI_{bb,min} = -3$ as the smallest recoverable signal strength. For the implementation of \log_2 operation, the optimized logarithm approximation is used from [14]. The exact RSSI computation circuit is described in HDL and implemented in an industrial 40-nm technology. The power breakdown extracted from the synthesis report is shown in Fig. 3. Most of the power (28%) is consumed by the Loop Control (LC) circuitry. Note that the power consumption of LC is significantly dependent on the preceding and following computational blocks, because intermediate and final results are saved in LC. ADD2 is the second biggest power consumer, followed by the SQUARE blocks and the I/Q merging adder ADD1. Note that the logarithm is only computed when accumulation is complete. Hence, the \log_2 activity is $1/N$ of the other blocks. The power contribution of \log_2 operation is therefore only 3%.

IV. APPROXIMATION APPROACH

The power breakdown information shown in Fig. 3 is used to choose which blocks to approximate. The approximation of \log_2 operation cannot bring substantial power savings. The precision of \log_2 operation is therefore kept at the maximum for the assumed bit width of 8 bits. Approximations of ADD1, ADD2, SQUARE1, or SQUARE2 result in reduced effective bit width of intermediate results and reduce power consumption of LC. Therefore, the approximation focus is on the addition and square computation blocks. We use combinations of adders from the open source EvoApprox8b adder library [12] and the multiplier from [11] for our implementations. *The target of the approach is to exploit the error-interaction effects.* Observe that the EvoApprox8b adders are designed separately from each other using genetic algorithms. *Though the circuits are optimized individually, in combination, their error behavior is unpredictable.* This is due to the error interaction effects. The error behavior at the output of an approximate element integrated in a system depends on error behaviors of all approximate elements in the fan-in cone. To investigate how several approximate elements are functioning within the RSSI computation, a template is created from the exact RSSI design

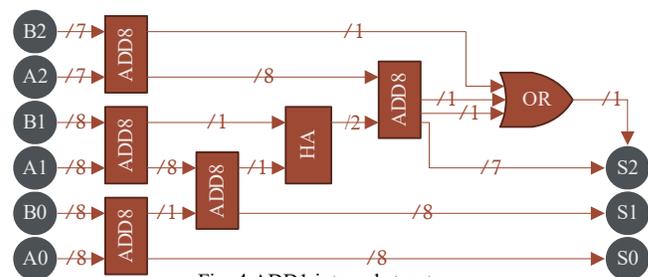


Fig. 4 ADD1 internal structure.

with place holders for approximate elements. The SQUARE blocks are implemented with 12-bit exact multipliers which can be replaced with multipliers from [11]. ADD1 and ADD2 blocks are designed using 8-bit adder blocks so that the 8-bit approximate adders from the EvoApprox8b library can be integrated. The schematic of the ADD1 block is shown in Fig. 4. ADD8 is a place holder for any 8-bit adder with two 8-bit inputs and a 9-bit output. The adders with such an interface cannot be chained in carry ripple fashion because there is no carry-in bit. This is why two additional adders and one Half Adder (HA) are used to connect the structures. The numbers in the figure indicate the bit-width. The ADD8 outputs are separated in 1-bit MSB (carry-out) and 8 lower bits. Carry out is added in higher order row using additional adders. The structure is synthesized flat so that the commercial synthesizers have freedom to optimize the non-idealities, if there are any. The structure is synthesized with 8-bit exact adders and is compared against the 24-bit adder synthesized by commercial circuit synthesis tool. The power differences are below 2%. Therefore, the proposed structures can be used as a reference for fair comparison. The OR gate is a way of easy error compensation. Because B2 and A2 inputs are 7-bit, carry-out of the 8-bit adder with those inputs is always zero. If exact adders are used, the commercial synthesis compiler removes the gate in the optimization process. For inexact adders, however, the OR gate stays and works as a light-weight error correction circuit. The circuit for ADD2 is designed similarly as illustrated in Fig. 5.

ADD8 combination approach: EvoApprox8b library has 500 adder designs. ADD1 needs five 8-bit adders while ADD2 needs six. Note that *the commutativity rule does not apply for approximate adders.* That is, an adder behaves differently when inputs are swapped. This results in a design space of $500^{2 \times (6+5)}$ possible combinations. Because EvoApprox8b adders are created randomly, an analytical approach is challenging. Therefore, we evaluate a selection of combinations using pareto optimality. Based on several criteria explained below, the design space is reduced to roughly 10^{20} options. These combinations are evaluated using heuristics.

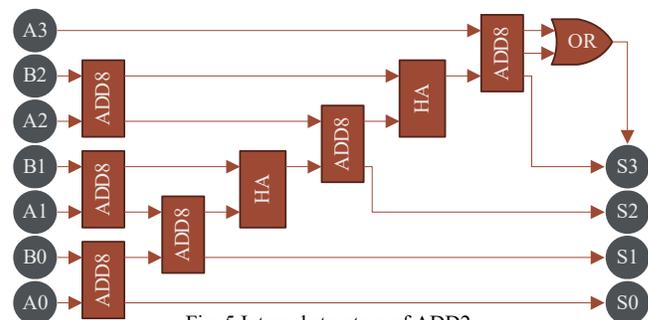


Fig. 5 Internal structure of ADD2.

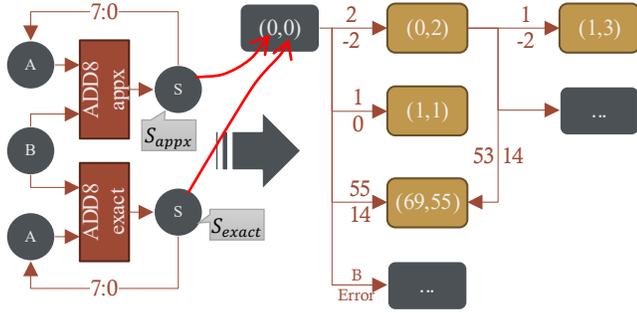


Fig. 6 Verification principle of self-accumulation criterion.

Power criterion: From the promising approximate components, the most suitable and promising sets are built and used to synthesize and evaluate different combinations of the approximate elements in those sets. The first disqualifying criterion is the power savings. The EvoApprox8b library is designed for two objectives: lowest power and lowest delay. However, the circuit delay constraint is not challenging in the RSSI computation. After synthesis with constraints for 16 MHz as the target clock frequency, the baseline RSSI implementation shows large timing slack, more than 90% of the clock period. Consequently, the library is filtered, searching for adders with promising power benefits. A carry ripple structure implemented in the same technology is taken for comparison as the most power-efficient exact solution. All adders with a power consumption above 90% of the conventional carry ripple adder are discarded. Out of 500 EvoApprox8b adders, 300 remain. Furthermore, the remaining adders are sorted according to their power in such a way that the adders with more power benefits are chosen and investigated first.

0+0 criterion: Another disqualifying criterion for the adders integrated at higher bit-significance stages is an error for 0+0 in the adder building blocks. This 0+0 case is very common for the beginning of accumulation in the ADD2 block. Consequently, the ADD2 approximations with errors for 0+0 at the inputs cause significant error at the beginning, which is then difficult to tolerate or compensate. In ADD1, the 0+0 scenario has less effect but still significantly affects the final output precision. Those errors typically propagate to the final output and degrade performance significantly. The 0+0 criterion removes 175 adders of the remaining 300.

Self-accumulation criterion: The accumulation loop introduces an additional discarding criterion. The individual adders should not accumulate their own errors if connected in a loop. To assess this and other approximation effects we created an error propagation flow. The analysis principle is illustrated in Fig. 6. The adder under test and the exact reference are connected with one of the inputs (A) connected with the first 8 LSB of the accumulation value (S) as shown at the left side of the figure. The accumulation values of exact and approximate adders (S_{appx}, S_{exact}) are used as node values in the propagation graph which is shown at the right side of the figure. The edges correspond to one computation step with a specific value B on the other input of both adders. For an input combination of (A=0, B=2), the approximate error has a wrong result S=0 (Error = -2). This is transferred from (0, 0) to (0, 2). Note that the edges of the graph are a result of distinct combinations of A and B. For instance, node (69, 55) is a result of two other possible

Alg. 1 Error graph creation algorithm.

ALGORITHM *propErrors*:

INPUTS: A_{exact} , // Intermediate exact result of accumulation
 A_{appx} , // Intermediate approximate result of accumulation
 $F_{exact}: S_9 \leftarrow (A_8, B_8)$, // Exact Adder
 $F_{appx}: S_9 \leftarrow (A_8, B_8)$, // Approximate Adder
 $PathDepth$, // Current path depth
 $PathDepth_{max}$, // Maximal path depth
 $Error_{max}$, // Error boundary

OUTPUTS: G //Error propagation graph

BEGIN:

1. **FOR** all B in [0,255]
2. $S_{appx} = F_{appx}(A[7:0]_{appx}, b)$ //Computing the approximate result
3. $S_{exact} = F_{exact}(A[7:0]_{exact}, b)$ //Computing the exact reference result
4. $A_{appx,new} = A_{appx}[MSB:8] + S_{appx}$ //Accumulating all carry outs
5. $A_{exact,new} = A_{exact}[MSB:8] + S_{exact}$
6. $E = A_{appx,new} - A_{exact,new}$ //Computing of the accumulated error
7. **IF** $(A_{appx,new}, A_{exact,new}) \notin G(nodes)$ **THEN**
8. **ADD NODE** $(A_{appx,new}, A_{exact,new})$ **TO** G
9. $PathDepth = PathDepth + 1$ //Increase path depth
10. **IF** $E < Error_{max}$ **AND** $PathDepth < PathDepth_{max}$ **THEN**
11. **CALL** *propErrors* with $(A_{appx,new}, A_{exact,new})$ // Recursive call
12. **ENDIF**
13. **ENDIF**
14. **ADD EDGE** $(A_{appx}, A_{exact}) \rightarrow (A_{appx,new}, A_{exact,new})$ **TO** G
15. **ENDFOR**

END

computation scenarios. As we move deeper in the graph, the computed results are further propagated, and the corresponding errors are calculated accordingly, e.g. S(0, 2) yields S(1, 3). Hence, this approach is more scalable and faster than an exhaustive search alternative. We essentially used this approach to evaluate how probable and how fast an accumulated error can exceed a specified boundary. Formally, the error propagation graph is created recursively computing the approximate and exact results for every possible B and dropping the recursive calls at the (S_{appx}, S_{exact}) combinations which are already evaluated.

The algorithm is shown in Alg. 1. Excessive recursive calls are avoided by checking whether node (S_{appx}, S_{exact}) is already in the error propagation graph and adding it otherwise (line 8). Additionally, the recursion is stopped when the graph depth or the accumulated error exceeds the defined maximum (line 10). Information about error accumulation extracted from the graph is used as a discarding criterion.

The evaluation metrics to assess the robustness of an adder are defined as the positive and negative Break Out Paths (BOPs). A BOP is a path in the error propagation graph that causes error accumulation above the set error boundary. Table 1 shows results for Add8_212 and Add8_212* which are the same adders but evaluated with swapped inputs. Add8_212 is less suitable for an in-loop operation. The error limit (64 for the analysis in Table 1) is exceeded in five cycles in 18816 cases. If Add8_212 is used, the inputs should be swapped. The number of BOP cases decreases, and the average BOP length increases. Furthermore, there is negative drift (the error is negative), which has the potential to improve precision in further loop cycles. The number of dropped paths is the number of paths which are dropped due to exceeded graph path depth. This is done to improve the scalability of the algorithm. In this circuit configuration, the graph path depth corresponds to the number of loop cycles. If the number is too large and full error bounding is not possible, the adders are evaluated for smaller loop cycle numbers and worst candidates are discarded. As listed in Table

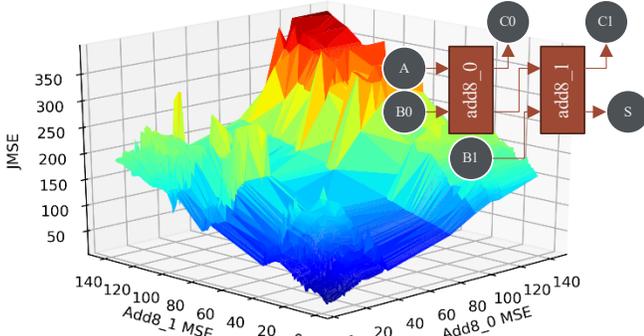


Fig. 7 MSE analysis for paired adder.

1, Add8_164 outperforms Add8_212 by exceeding the bound error only after 34 cycles in 3456 cases. Add8_164 is therefore chosen as a candidate for ADD2 approximations. With this criterion, an additional 60 adders are discarded from the design space, leaving 65 adders.

Table 1 Evaluated in-loop adder characteristics. For maximum error of 64 and path drop threshold of 64.

	Add8_212	Add8_212*	Add8_164
Positive BOPs	18816	12192	0
Negative BOPs	0	3344	3456
Mean BOP length	5	5.2	34
Dropped paths	14592	4054	11019

Pairing criterion: The error propagation flow used in the self-accumulation criterion is reused with different circuit configurations to assess adder pairing properties. Consider that two adders from the EvoApprox8b library are connected, e.g. the output of the first adder is input to the second adder. Then the adder pair is investigated for joint error characteristics. The final metrics for comparison are Mean Square Error (MSE) obtained from system-level simulation and power obtained after the back annotated gate-level simulation. MSE is computed using (3)

$$MSE = \sum_I \frac{(O_{apx} - O_{exact})^2}{|I|}. \quad (3)$$

O_{apx} and O_{exact} are approximate and exact computation results, respectively. I is the set of all tested input combinations. The MSE of the joint structure is assessed and illustrated as in Fig. 7. The Joint Mean Square Error (JMSE) is plotted against the individual MSE of the constituent add8_0 and add8_1 when analyzed alone. As shown in Fig. 7, some combinations result in better JMSE. Those combinations are preferred in our adder constructions. With the pairing criterion, more adders are discarded from the design space, resulting in a set of 10 candidate adders.

V. EVALUATION OF APPROXIMATE RSSI CIRCUITS

Evaluation approach: For the circuit analysis, we developed an error analysis framework that incorporates generation of realistic input stimuli, modeling of the approximate RSSI computation on system level and synthesis of the same designs with commercial synthesis tools in an industrial 40-nm technology. The approach is illustrated in Fig. 8. With Python scripts, a generic template is created with place holders for approximate or exact elements. Approximation is performed by

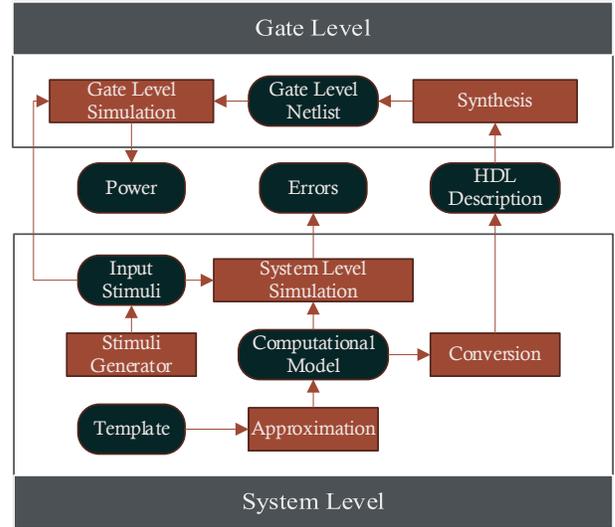


Fig. 8 Approximation and error analysis framework.

filling the template spaces with approximate elements. The EvoApprox8b functional description of adders in the C programming language is used for creation of the computational model based on the template. The computational model is then used for error analysis in a system-level simulation and conversion to the HDL description of the design and the following synthesis. The synthesized netlist is used for activity-based power estimation using commercial power analysis tools. The gate-level activity is elaborated from a digital simulation of the synthesized delay-back-annotated netlist using the same stimuli files which are used for error analysis. The stimuli files are generated by a self-made flexible stimuli generator which modulates the raw payload data and converts it to the RF signal with addition of the controlled channel noise effects. Both MSE and power depend on the incoming signal RSSI. To address this dependency, MSE and power values are computed for a range of input signal stimuli with different RSSI values, and then averaged (MSE_{AVR}).

ADD1 only approximation results: The MSE versus power plot of the RSSI design with only ADD1 approximated is shown in Fig. 9 (A). Every ‘x’ point corresponds to a synthesized and analyzed RSSI design that is selected in the way discussed in the previous section. The red ‘+’ corresponds to the exact RSSI. Some approximate designs consume more power for small RSSI values because the error propagates to higher significance bits and causes gate switching there that is not present in the exact design. However, most designs indicate power savings, which are significant. Pareto-optimal adders are highlighted through the grey line. The approximations are only in the ADD1 block though which consumes only 11% of the total power.

Approximating both adders: We also perform ADD2-only approximation and select again Pareto-optimal adders, as a basis for more aggressively approximated RSSI designs in which both adders are approximated (Fig. 9 (B)). Most designs operate with average power values below 75% of the exact RSSI power. The MSE degradation is clearly noticeable but can be tolerated for some applications. Note that an MSE of 30 corresponds to the squared variance of the errors in the signal with values between 2^8 and 0.

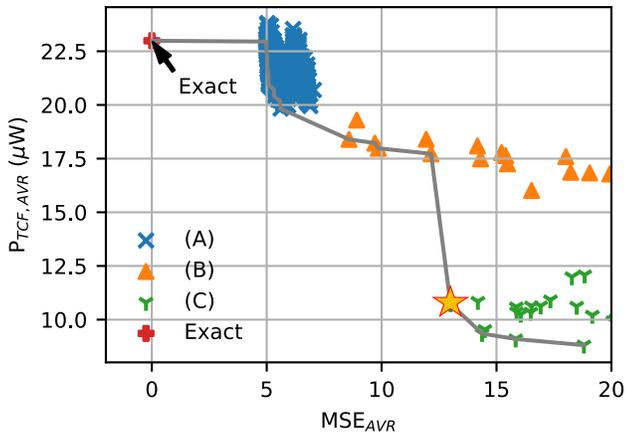


Fig. 9 Average power vs average MSE of RSSI with approximated ADD1 (A), approximated ADD1 and ADD2 (B), approximated ADD1 and ADD2, SQUARE1 and SQUARE2 (C).

Square approximation: As the most aggressive approximation step, the SQUARE1 and SQUARE2 blocks are approximated with the multiplier from [11] (Fig. 9 (C)). Further power savings are achieved with a limited increase in MSE. The designs at the Pareto front have significant power savings of over 50% with the most aggressive design at 62%.

The impact on network level: The significant power savings reported above are achieved at a cost of precision, which has impact on network level performance, and needs to be carefully assessed there. In [15], the impact of circuit-level inaccuracies on the network level is investigated. We integrated one of the RSSI designs (marked with a star in Fig. 9) into a star network (described in detail in [15]). Eight sensor nodes and one coordinator communicate periodically through the non-beacon enabled IEEE 802.15.4 protocol with default settings. The result of our simulations is shown in Fig. 1. The approximate RSSI is used for Clear Channel Assessment (CCA) and Low Power Listening (LPL) [16] operations. LPL is a technique that enables a node to sleep, though an incoming packet is expected to be detected by periodically checking the state of the channel. Because of frequent use of CCA in this scenario, the channel acquisition circuitry such as RSSI becomes a dominant power consumer, and needs to be considered in power planning. We applied the analysis method of [15] in this work, and analyzed the impact of the chosen design on three network performance metrics, namely Packet Reception Ratio (PRR), latency, and energy per bit. The results show that the chosen approximate RSSI design saves 24% of the energy per transmitted and received bit in the baseband. PRR and latency are kept the same by smart readjustment of parameters (e.g. the CCA RSSI threshold). This is possible by sacrificing network flexibility through reduction of effective communication range. This may or may not be possible depending on the application.

VI. CONCLUSION

This paper is the first to present a detailed investigation of approximation in one of the DBB computations. A generic approach is presented that targets to use various circuit-level approximations of blocks within a design and match them for adequate and tolerable results. Furthermore, the approximation analysis flow is presented. With this flow, the inexact baseband

computation circuits, such as RSSI, are analyzed for average error with representative environment conditions using realistic stimuli. The internal error behavior is analyzed, and scalable criteria are presented for design space reduction of an approximate design with bounded error. The promising approximation possibilities are implemented, synthesized and analyzed in terms of the resulting power-error Pareto graph, which can be used for QoS-Energy matching on higher abstraction layers. In an illustrative network setup, the proposed approximate RSSI improves overall network-level energy efficiency of DBB processing by 24%, preserving PRR and latency performance, at the cost of a reduced effective communication range.

ACKNOWLEDGEMENTS

This work is partially supported by Semiconductor Research Corporation (SRC) under the QoS-AB project (GRC Task 2681.001).

REFERENCES

- [1] Y. H. Liu *et al.*, "A 1.9nJ/b 2.4GHz multistandard (Bluetooth low energy/Zigbee/IEEE802.15.6) transceiver for personal/body-area networks," in *Dig. Tech. Pap. IEEE Int. Solid. State. Circuits Conf.*, 2013.
- [2] IEEE Computer Society, "IEEE 802.15.4 Specifications," 2015.
- [3] NXP, "NXP Kinetis® KW41Z 2.4 GHz." [Online]. Available: <https://www.nxp.com/docs/en/data-sheet/MKW41Z512.pdf>. [Accessed: 17-May-2018].
- [4] Y. H. Liu *et al.*, "A 3.7mW-RX 4.4mW-TX fully integrated Bluetooth Low-Energy/IEEE802.15.4/proprietary SoC with an ADPLL-based fast frequency offset compensation in 40nm CMOS," in *Dig. Tech. Pap. IEEE Int. Solid. State. Circuits Conf.*, 2015, vol. 58, pp. 236–237.
- [5] M. O. Farooq *et al.*, "MR-LEACH: Multi-hop routing with low energy adaptive clustering hierarchy," in *Proc. of the IEEE Int. Conf. Sens. Technol. Appl.*, 2010, pp. 262–268.
- [6] K. Itoh *et al.*, "Performance of handoff algorithm based on distance and RSSI measurements," *IEEE Trans. Veh. Technol.*, vol. 51, no. 6, pp. 1460–1468, Nov. 2002.
- [7] S. Mazuelas *et al.*, "Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks," *IEEE J. Sel. Top. Signal Process.*, vol. 3, no. 5, pp. 821–831, Oct. 2009.
- [8] H. Ito *et al.*, "An ultra-low-power RF transceiver with a 1.5-pJ/bit maximally-digital impulse-transmitter and an 89.5-uW super-regenerative RSSI," in *IEEE Asian Solid. State Circuits Conf.*, 2014, pp. 265–268.
- [9] "International Technology Roadmap for Semiconductors 2.0: Executive Report." [Online]. Available: <http://www.itrs2.net/>. [Accessed: 12-Nov-2018].
- [10] P. Kulkarni, P. Gupta, and M. Ercegovic, "Trading accuracy for power with an underdesigned multiplier architecture," in *Proc. of the IEEE Int. Conf. on VLSI Design*, 2011, pp. 346–351.
- [11] K. Y. Kyaw *et al.*, "Low-power high-speed multiplier for error-tolerant application," *IEEE Int. Conf. Electron Devices and Solid-State Circuits*, 2010.
- [12] V. Mrazek *et al.*, "EvoApprox8b: Library of approximate adders and multipliers for circuit design and benchmarking of approximation methods," in *Proc. of the IEEE/ACM Design, Automation & Test in Europe Conf. & Exhibition*, 2017, pp. 258–261.
- [13] S. Haykin, *Digital communication systems*. Hoboken, Nj: J. Wiley & Sons, 2014.
- [14] V.-T. Sai and V.-P. Hoang, "An optimized implementation of logarithm hardware generator for digital signal processing," in *IEEE Int. Conf. on Communications and Electronics*, 2016, pp. 153–157.
- [15] P. Deterer *et al.*, "Understanding the impact of circuit-level inaccuracy on sensor network performance," in *Proc. of the 15th ACM Sym. on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks (PE-WASUN)*, 2018, pp. 107-114.
- [16] A. Dunkels, "The ContikiMAC radio duty cycling protocol," Swedish Institute of Computer Science, SICS Technical Report T2011:13, 2011.