Citation	<p>Hans Reyserhove, Wim Dehaene, (2017)</p> <p>A Differential Transmission Gate Design Flow for Minimum Energy Sub-10-pJ/cycle ARM Cortex-M0 MCUs</p> <p>IEEE Journal of Solid-State Circuits (JSSC), volume 52, issue 7, pp. 1904-1914, July 2017</p>
Archived version	<p>Author manuscript: the content is identical to the content of the published paper, but without the final typesetting by the publisher</p>
Published version	<p>https://doi.org/10.1109/JSSC.2017.2693241</p>
Journal homepage	<p>http://sscs.ieee.org/publications/ieee-journal-of-solid-state-circuits-jssc</p>
Author contact	<p>Hans.Reyserhove@esat.kuleuven.be</p> <p>+ 32 (0)16 321169</p>

(article begins on next page)

A Differential Transmission Gate Design Flow for Minimum Energy Sub-10pJ/Cycle ARM Cortex-M0 MCUs

Hans Reyserhove, *Student Member, IEEE*, and Wim Dehaene, *Senior
Member, IEEE*

Abstract

Ultra-low voltage operation is key in achieving energy-efficient operation for microcontroller systems. Variation resiliency, high speed operation and short design time are the most important challenges for these systems. This paper overcomes these challenges in a new design strategy that enables standard cell design with differential transmission gate logic. The commercial toolchain is extended with in-house developed add-ons and makes use of two custom libraries with different device length to allow high speed vs. low leakage trade-offs. The design flow is used to prototype two highly efficient 32b ARM Cortex-M0 microcontroller systems in 40nm CMOS. The core of the first prototype scales down to 190mV and 0.8MHz and reaches 16.07pJ/cycle at 31.2MHz and 440mV. The second prototype benefits from the dual libraries and reduces core energy consumption by 50% at the same speed performance. Minimum energy operation is thus achieved at an even lower voltage (370mV) with the M0 core consuming only 8.80pJ/cycle at 13.7MHz, breaking the sub-10pJ/cycle barrier for a 6-35MHz range.

I. INTRODUCTION

Design of mobile devices and IoT applications is challenged by mobility, performance and energy trade-offs. For the bigger part, these devices consist of sensors combined with a microcontroller (MCU). Because the increasing amount of sensed data almost always requires post-processing, the MCU is expected to handle a significant amount of workload [1], [2] and requires a memory large enough to handle this post-processing efficiently. As such, both these blocks

are among the dominant energy consumers in these SoCs. Low energy memory design as in [3] is therefore required for every IoT system, but falls out of the scope of this work. Considering the MCU, energy efficiency can be considered as the most important design parameter in such systems. Operating the microcontroller at ultra-low voltage (ULV) is a necessity to meet the imposed strict energy requirements. The goal is to operate at the Minimum Energy Point (MEP) [4]. The MEP balances active and static energy: through ULV operation active energy is reduced, while several other strategies allow static energy reduction. In this combination, optimal energy/cycle can be achieved at near-threshold voltages. This point is highly technology, implementation and application dependent. Therefore, every system requires careful tuning at design time to enable minimum energy operation at the desired speed performance. Although the ULV approach is attractive from an energy point of view, it imposes new challenges for the designer. Operation as low as the threshold voltage increases sensitivity to variations. Mismatch will have a significant effect on operation, as will operating in different process corners. In any case, speed degradation by 2 orders of magnitude or more can be expected when comparing with operation at nominal supply voltages: recent literature has trouble achieving MHz operation at the MEP [5]–[7]. Applications as in [1], [2] require a 1MHz speed performance or higher to operate ECG or other processing algorithms. Both variation tolerance and high speed performance can be achieved using a different logic structure, consisting of transmission gates [8]. Sensitivity to both intra-die and inter-die variation are significantly improved. By stacking the NMOS devices, sizing of PMOS devices can be reduced substantially. The high speed performance achieved in this way [9] seems promising, even for large designs with fewer pipeline stages. A major challenge in using ULV differential transmission gate logic is its full custom nature, requiring a large design effort and time.

Microcontrollers differ from DSP blocks similar to [9] in the fact that they are less active. A microcontroller, programmable to do any operation, is less efficient for a specific task than a DSP block designed for that task. With global activity factors in the order of a few percent [10], a different trade-off in dynamic vs. static energy is expected in microcontrollers. Fig. 1 shows this more clearly. Static energy depends on the supply voltage, the leakage current and the speed performance. Dynamic energy on the other hand is determined by activity, total capacitance and the supply voltage. If activity is very low, like in a microcontroller, dynamic energy decreases accordingly. As a consequence, static energy becomes relatively more important. To reduce

total energy consumption as much as possible, static energy reduction is necessary. Attempts to reduce static energy by reducing leakage current often result in a proportional reduction in drive current, severely reducing speed performance. A speed reduction, in its turn, actually increases static energy. Hence, when reducing static energy consumption, it is key to keep a high speed performance. This will result in a high speed performance and a low voltage MEP with a very low total energy consumption: dynamic energy is reduced by the ULV approach, while static energy is kept at a minimum by applying high speed low leakage circuits. A common strategy to combine low leakage and high speed performance (in the context of an IoT node) is the use of multiple standard cell libraries: slow low leakage cells are used on non-critical paths, while speed performance is kept high by using fast standard cells on the most critical paths. Current designs fail to achieve high performance at the lowest voltages, resulting in a higher voltage supply to achieve good-enough speed performance, leading to a MEP with higher static and dynamic energy.

To summarize, there are several design challenges in ULV operation: 1) A key challenge is to enable robust operation with the increased variability of advanced nanometer CMOS technologies; 2) the goal should be to achieve a MEP with an operating speed high enough to facilitate a wide range of applications; 3) the high speed MEP should occur at a supply voltage low enough to reduce dynamic energy as much as possible; 4) static energy should be reduced as much as possible; 5) all these challenges should be compatible with an efficient automated design flow, avoiding any full custom work and enabling fast design time and iterations if necessary. Current state-of-the-art low voltage microcontrollers do not fulfill these requirements. Speed performance in the MHz-range is not achieved at the MEP [5]–[7] and these papers exploit variation-sensitive building blocks. Other work requires full custom circuit design to ensure high speed performance and variation resilient operation [9].

The aim of this work is twofold and reflects the two prototypes discussed in this paper. A first goal is to use the efficiency of standard cell design with the ultra-low power, high speed performance and variation resilience of full custom work and combine it into a generic design flow, as offered by commercially available tools. The proposed design flow is applied to create a 32-bit microcontroller system with an industry-proven core, an ARM Cortex-M0, which meets all requirements: efficient design, high speed performance and variation resiliency. A second goal is to try and optimize energy consumption for this core to its bare minimum. This means

not only operate the MCU at its MEP, but also tune the standard cell library carefully so that the MEP is corresponding to a high speed performance with low supply voltage. Dynamic energy at this point is low and with high speed performance, static energy is limited, resulting in a very low energy/cycle, competing with state-of-the-art conventional standard CMOS designs.

This paper is organized as follows. Section II briefly revisits the transmission gate logic from [8] and describes how it is used to create a full standard cell library. Section III gives an overview of the proposed differential transmission gate design flow and describes the different steps taken to merge the flow with commercially available tools. Section IV discusses the microcontroller architecture used to prototype the proposed design flow. The steps taken to reduce energy consumption of the prototype even further, are explained in section V, followed by extensive measurements in section VI. A comparison with other state-of-the-art microcontrollers allows to evaluate the performance of the presented design flow with, among others, conventional static CMOS designs. Section VII concludes this paper.

II. TRANSMISSION GATE LOGIC

Transmission gates and pass gates have been used before for various reasons. [11] uses transmission gate logic (TGL) for very high speed applications though not at ULV. [12] does apply pass gate logic at ULV and focuses on leakage. This work uses TGL to realize reliable operation under variation at ULV, combined with high speed performance (considering the applied supply voltage). A more extensive analysis on this topic can be found in [8]. High speed performance at ULV is possible by maximizing the drive current of the applied logic. An important step is the use of low- V_T (LVT) devices. Since their V_T is lowest, LVT devices have a higher $V_{gs}-V_T$ at a low supply voltage, maximizing their drive current. Naturally, leakage current increases accordingly, but it can be managed by the circuit techniques further discussed in this section.

The inverter is the most basic digital building block and is therefore discussed first. To achieve ULV operation and guarantee correct operation under the influence of variations, it should be sized for maximal noise margin. Noise margin ¹, as defined in [13], provides a relatively good insight in how the gate functions under variations and different PVT conditions, especially at very low supply voltages. Although device strength at ULV can vary between technologies, PMOS

¹ $NM = \min(NM_H, NM_L)$ with $NM_H = V_{OH} - V_{IH}$ and $NM_L = V_{OL} - V_{IL}$

devices have significantly lower drive current in this region ([14], [15]). Band-engineered devices like in [16] can compensate for this but are not available in most technologies. Traditional CMOS inverters can thus not be applied at ultra-low voltage without a severe area penalty. NMOS stacking (Fig. 2) is employed to reduce NMOS drive and leakage current, which makes it easier to balance PMOS and NMOS strength. This significantly decreases the sensitivity to variation in the ULV range. Fig. 3 shows the variability of the propagation delay for an inverter employing NMOS stacking vs. a conventional static CMOS inverter. The CMOS inverter is sized either for equal speed, equal area or equal noise margin when compared to the stacked NMOS inverter. The stacked NMOS inverter is the most variation resilient topology, despite the better noise margin and smaller area when comparing to conventional static CMOS. Additionally, it provides significantly better noise margin across process corners, as shown on the right hand side of Fig. 3. This demonstrates its superior performance and robustness at ULV under the influence of variations. In this technology, area can be reduced by 50% through NMOS stacking while keeping maximal noise margin. The forthcoming speed degradation (27%) is of less importance when compared to the reduced area and leakage current (46% decrease).

Logic operations are realized by using transmission gates rather than conventional static CMOS logic. Similar problems as with the inverter occur: conventional static CMOS gates, especially the ones with stacked PMOS devices, can not be applied here without a severe area penalty. Transmission gate logic (TGL) offers a solution, as both PMOS and NMOS devices pass the logic level (see Fig. 4). Similar to the inverter, rise and fall time can be balanced by stacking the NMOS device. Close to minimal device sizes can be used. Balanced drive strength means an optimal noise margin and thus good performance under variation. Having both types of devices in the conducting path improves inter-die variation, especially in the difficult slow-fast or fast-slow corners. A thorough analysis is shown in Fig. 3. A TG NOR, extended with an inverter to regenerate the logic level, is compared to a conventional static CMOS NOR gate. Again, the NOR gate is sized for either equal speed, equal area or equal noise margin. When considering variability of propagation delay, the TG is not the most variation resilient building block. However, when noise margin is considered across process corners, the CMOS NOR gate performance degrades severely, especially in the slow-nmos fast-pmos and fast-pmos slow-nmos corners. This precludes the CMOS NOR gate from robust operation at ULV. Hence, the TG provides the best trade-off between speed, area, variation resiliency and noise margin across

process corners.

The structure of the transmission gate allows any logic function by rewiring the input signals. Inverting (NAND2, NOR2, XNOR2, NMUX2), non-inverting (AND2, OR2, XOR2, MUX2) or semi-inverting (IAND2, IOR2, INAND2, INOR2) gates can be realized with just 4 transistors (6 using stacked devices). Since TGL combines PMOS and NMOS devices, both signal polarities are necessary to control the gate. Two strategies are possible: either generate each complementary signal using an inverter, or propagate the differential signal and apply all logic fully differential. Since inverters have a severe area and energy impact, the TGL is applied in a fully differential way. Considering that the TG does not have to be up sized for ULV operation, area penalty is negligible. The combination of variation tolerance and differential signaling improves the robustness of the TGL library and thus ULV operation.

The TGs do not have gain and thus do not regenerate logic levels. In long logic paths, this results in signal loss. To realize gates with 3 or more inputs, multiple TGs are cascaded. Up to 8-input gates for any logic function are created by simply combining 2-input TGs in series or parallel. Typically difficult gates like NOR_x can be realized without excessive PMOS stacking. To continue to operate at ULV, no more than 3 TGs are cascaded, after which a regenerating element (inverter or flipflop) is placed. A complete cell can thus be considered as the combination of TGs and an inverter. An advantage of adding the inverter is that delay and power becomes dependent on the input slew and output load only, enabling characterization as any other standard cell (see section III).

To generate sequential logic, memory elements are required. [9] uses alternating positive and negative latches, allowing time borrowing through pipeline stages. Time borrowing is highly suited for ULV operation as it can average and therefore cancel timing imbalance between pipeline stages caused by process variations. However, full latch based designs are difficult for timing closure and require additional effort from the designer due to limited support in the commercial tool chain [17]. Positive and negative latches, consisting of TGs and inverters designed as discussed previously in this work, are thus combined into a master/slave-flipflop as shown in Fig. 2. These flipflops can be applied as in any other sequential design with a single clock domain, but with differential input/output.

By combining inverters, logic gates and flipflops, any logic or sequential function can be realized. This clears the way for a standard cell design flow, as discussed in section III.

III. DIFFERENTIAL TRANSMISSION GATE DESIGN FLOW

A major challenge in using TGs is their differential input. As mentioned in section II, the TGs are applied in a fully differential manner. In this section we propose an automated design flow, fully compatible with standard cell tools, that overcomes this challenge. Any RTL code can be synthesized to fully differential transmission gate logic in an automatic way, imposing almost no overhead to the designer and being fully compatible with the commercial tool chain for standard cell design. An overview of the proposed design flow can be found in Fig. 5. Starting from the elementary gates pictured in Fig. 2 a large amount of cells are created with varying functionality and drive strength. This library is characterized for timing and power, much like any other standard cell library, but in a differential way. This differential library can not be used for synthesis, since commercial synthesis tools do not support differential signaling. For this reason, the next step converts the differential library to a pseudo single ended library. Every cell is reduced to a single ended version of itself, keeping only the non-complementary signals. Although this library no longer has any physical meaning (the single ended cells are never implemented), it contains the real, worst case data from the differential library and can therefore be used as a temporary library replacement during synthesis. RTL code is synthesized with full tool chain capabilities, resulting in a single ended gate level netlist. In a last synthesis step this netlist is converted back to a differential netlist, coherent with the original differential library. At this point, the differential netlist can be used with the differential library for place and route, timing analysis and ultimately sign off. This flow, although requiring close to no additional effort from the designer, thus interferes in the conventional standard cell design flow on all 3 levels: the library level, the logic level and the physical level. All 3 interventions are now discussed in further detail.

A. Library Level

The circuit techniques discussed in section II allow us to generate a large ULV standard cell library. All cells consist of a combination of TGs and inverters and are standalone in timing and power since they are terminated by an inverter regenerating the logic level and isolating the load from the TG. Complex gates are realized by cascading multiple TGs, effectively limiting the amount of power hungry inverters. Either 1, 2 or 3 levels of TGs can be used, providing up to 8-input logic gates with any combination of logic operations. Implementing

all possibilities would lead to a library of millions of cells. Although the TGs are structurally generic, such amount of library data would not be feasible. A subset of 241 different cells is selected, implemented and characterized across a wide voltage range for all process corners with the Synopsys SiliconSmart® tool. All cells are implemented over a range of different drive strengths.

To characterize the differential cells, start and end conditions of differential inputs are complementary. Since every cell ends in an inverter, it can be characterized unambiguously for a wide slew and load range. Despite being complementary, differential inputs are independently influenced by a variety of effects in the back-end design flow. e.g. routing length, drive strength, parasitic load and slew rate. To be able to account for these effects, differential signals transition independently during characterization, thus having independent drive strengths, slew rates and output loads. Hence timing mismatches between differential signals can be arbitrary large and is accounted for during timing and power analysis, allowing correct design optimization. Flipflops have differential input/output, controlled by a single clock domain. Launch and capture events are triggered by the intersection of the complementary signals. Although the complementary flipflop signals have identical paths, their loads can vary. Therefore, setup and hold times are characterized for both signals independently, enabling independent timing analysis and optimization.

All cells have a generic structure, existing of a combination of TGs and inverters. Layout of the complete library can thus be realized efficiently. The layout of a single TG is used hierarchically, automatically rewiring its inputs according to the desired functionality. Inverters are added corresponding to drive strength.

Differential characterization results in library files containing all the necessary data from the differential cells. To be compatible with and fully exploit the potential of commercial synthesis tools, single ended synthesis is preferred. To this end, the differential library is converted to a single ended library. An in-house developed tool reads the library, processes all the data and outputs a single ended library. The tool's operations are fairly straightforward: complementary signals are reduced to their single ended behavior with worst case timing and/or total added power information from the differential cell. Special consideration is given to some cells, e.g. flipflops already have differential output, both in the single ended and the differential library. The resulting single ended library is perfectly suited for synthesis, since it consists of fictive,

single ended cells with real timing and power information. The single ended library can be considered equivalent to the differential library, augmented with zero-delay and -power inverters at the interface to create the differential signals. The single ended library has no corresponding physical properties. Area properties from the differential library are transferred to the single ended library, so that the synthesis tool can correctly predict area implications and wiring.

B. Logic Level

At the logic level, the most important step is synthesis. Any RTL code can be synthesized with the single ended library. This results in a fictive gate level netlist: the allocated cells only exist in the single ended library, without having a physical meaning. The full potential of the synthesis tool can be used, e.g. clock gating, leakage optimization, multiple power domains, level shifter insertion, isolation cell insertion, ... During synthesis, the tool favors more complex cells (higher-level cascades of TGs) when speed performance is not stringent and functionality allows it. In a next step, the gate level netlist is processed to reconnect it with the original differential library. Again, conversion is clear, adding a complementary counterpart for each signal. Already differential signals, like the flipflop output, require caution not to create multiple independent duplicates of the same signal. Macro's or additional logic libraries are added to the design with full tool support. Dedicated level shifters with a wide voltage range are automatically inserted to interface between the ULV differential domain and other single ended domains.

The synthesis tool, considering every signal as being single ended, has no knowledge of the presence of complementary signals in the final differential gate level netlist. As a result, in logic combinations where it needs both polarities, it generates the complementary signal with an inverter. During conversion from single ended to differential, these redundant inverters are removed, reducing leakage and overall power.

Timing analysis is run on the differential gate level netlist with the actual differential library to verify whether all design constraints are met. If not, the process can be iterated requiring minimal effort from the designer.

C. Physical Level

The final steps in the proposed flow reach from place-and-route to full chip sign-off. All place-and-route steps are done with the differential library. Differential nets are routed independently

with the same detail as other single ended nets. Since start- and endpoints are physically close to each other, post-route analysis showed minimal routing length difference. Despite this fact, differential signals do experience different parasitic effects. Thanks to inclusion of these effects in the library characterization, the effect of this mismatch on timing and power consumption is accounted for by the optimizer. As such, the tool is able to optimize differential nets independently to meet the imposed design constraints. Because of the relatively low speed performance of the targeted applications (see section I), slight mismatch between complementary nets is acceptable. Differential routing would cause severe congestion and would vastly slow down the place-and-route design flow. In this strategy, differential routing is not necessary, which adds to the overall applicability of the design flow.

For clock tree synthesis, a single ended clock mesh is used rather than a normal clock tree. This is solely due to the ULV nature of the design process, not due to the differential logic. The carefully controlled clock mesh allows greater control over the clock delay slew and skew across the different voltage domains and process corners [18]. Timing analysis and sign-off can be done at the full chip level across different process corners and combines the differential transmission gate library with foundry libraries or other IP.

IV. MICROCONTROLLER ARCHITECTURE

The proposed design flow is applied to implement a microcontroller system consisting of an ARM Cortex-M0 core, an SRAM memory, a UART-interface, GPIO ports and a test/debug-interface. An overview of the system is shown in Fig. 6. All peripherals are memory-mapped to the AHB bus. With 32-bit instructions, 3 pipeline stages and 16 interrupts, the M0 core is a competitive architecture in industry that is optimized for small low power applications. The provided SRAM memory is used to emulate ROM, instruction and data memory. This division is allocated during start up. The core and peripherals are located in a separate ULV power domain, fully composed out of the proposed differential transmission gate library. The SRAM macro lies in a separate power domain, accessed through level shifters also performing single ended to differential conversion and vice versa.

The used microcontroller system is an ideal architecture as a proof-of-concept for the proposed design flow for several reasons. First, the design flow targets ULV applications and minimum energy operation. The Cortex-M0, being one of the smallest 32-bit microcontrollers, is an industry

standard for low power applications. Second, the Cortex-M0 core was obtained from the ARM DesignStart University Program, providing an obfuscated RTL netlist. The lack of insight in the RTL code reinforces the genericity of the design flow, since there was no control over the input RTL code for synthesis. Third, the Cortex-M0 is the subject of recent state-of-the-art literature ([5], [6]) showcasing high energy efficiency, using either conventional static CMOS logic or other logic families. This allows extensive evaluation of the proposed differential transmission gate design flow. Last, Fig. 6 shows the system is a hybrid system: it combines the ULV differential transmission gate library with both single ended IP blocks and foundry models (SRAM) as with differential - single ended interfacing (I/Os). Here lies the true potential of the design flow: combining foundry libraries or other IP and macros with dedicated ultra-low power sub-domains, consisting of differential transmission gate logic.

V. MINIMUM ENERGY OPTIMIZATION

This work includes two generations of the microcontroller architecture discussed in section IV. The first design acts as a pioneering design realized with the differential transmission gate design flow. It uses the transmission gate library discussed in section III-B to enable a high operating speed low power M0 microcontroller system. Using a single, high speed library, it sets the standard for speed performance and shows the baseline energy consumption. The M0 core consists of 10693 logic gates and 1332 flipflops. The synthesis tool makes a trade-off between complex functionality but slow, high level TGs and simple but fast, low level TGs. As a result, 5.8% of total logic gates are 3 level TGs (A222O, A33OI, ...), 36.7% are 2 level TGs (AND4, AOI, FA, ...), the rest are 1 level TG deep (AND2, NAND2, IAND2, MUX2, ...).

The second design is similar in architecture to the first, but uses a combination of two transmission gate libraries, allowing dual library leakage and speed optimization. This way static energy consumption can be decreased severely without compromising operating speed. Creating multiple libraries can be realized by either increasing the V_T or the gate length of one of the libraries. Fig. 7 shows the trade-off between length increase vs. V_T increase. Both low V_T and regular V_T devices for 40nm and 60nm gate length are compared for speed performance and leakage power. While the delay increase across devices is more or less equidistant, leakage power actually increases when a higher V_T device is equipped for the minimal length. Suffering from both a speed and leakage penalty, it makes no sense to use these devices for cells in any design

at this supply voltage. The main reason is the relatively large V_T increase compared to the low supply voltage; this dramatically decreases $V_{gs}-V_T$ compared to the LVT devices, which pushes the device further into weak inversion, making it even more sensitive to variations. Another disadvantage of increasing the V_T is the decrease in PMOS device strength, which results in a larger PMOS device for the same noise margin. One library thus uses 40nm gate length low V_T devices, while the other uses 60nm gate length low V_T devices. Device width was adjusted accordingly, allowing ULV operation and maximal noise margin.

By employing the fast library on timing critical paths, the second design can still achieve the required high operating speed, while major power savings are realized due to leakage reduction on the non-critical paths (see section VI). An identical RTL netlist of the microcontroller architecture was synthesized using the dual libraries. A smaller SRAM (64KB) was equipped to reduce overall area and power. The dual library design uses only 24.98% of the fast 40nm gate length cells, while all the other cells are replaced by slower 60nm gate length cells with less leakage current. No speed reduction is allowed, as both designs are synthesized at the same target frequency. The dual library optimization also impacts the use of higher level TG cells: only 28% of the cells are 1 level TGs, 58% are 2 level TGs and 14% are 3 level TGs. This can be explained by the addition of the low leakage library, allowing a more careful trade-off between gate complexity, speed and power.

In order to see the effect of the discussed improvements, Fig. 8 shows a preview of the measured energy performance of both designs. Total energy consists of static and dynamic energy and all are shown for both prototype 1 and prototype 2. The design that equips a single library (prototype 1) uses fast cells all-round, resulting in a good speed performance. Since speed performance is determined by the critical path, all non-critical paths are over designed in speed and thus energy. This adds significantly to static energy consumption. When considering the low activity of the microcontroller (see section I), the static energy has a relatively high contribution to the total energy consumption (up to 90%). Static energy decreases with increasing supply voltage due to a relatively large speed increase compared to leakage current increase. For prototype 1 this results in a MEP that is located at a relatively high supply voltage, undermining all original efforts to operate at ULV and decrease dynamic energy consumption. An improvement, realized in the dual library design (prototype 2), is to decrease static energy without compromising speed. A lower static energy curve moves the static vs. dynamic energy trade-off to the left, resulting

in a lower supply voltage MEP. In this scenario, both static and dynamic energy are decreased, resulting in a significantly lower total energy consumption, and the efforts done at library level to enable ULV operation pay off.

Measurements on the first prototype presented in [19] show that minimum energy operation does not occur at the lowest supply voltage. Operation in this region is clearly sub-optimal due to a low speed and a high static energy. Enabling operation at these very low supply voltage comes at an energy cost: device sizing is increased to enable equal noise margin at these voltages. By relaxing the constraints on noise margin, operation at the lowest supply voltage is given up. As a result, this energy cost can be retrieved: device sizes can be reduced, reducing both static and dynamic energy at the device level and by consequence total energy consumption. The overall strategy to decrease the $V_{DD,MEP}$ by lowering the leakage energy is partly counteracted by the dynamic energy decrease due to device downsizing, but energy/cycle benefits from both measures, as is shown in section VI. In general, the applied supply voltage is of less importance, as long as energy consumption is minimized and desired performance is achieved.

VI. MEASUREMENTS

The qualitative analysis from section V is made quantitative with measurements of both designs. Both generations of Cortex-M0 microcontrollers were prototyped in a 40nm General Purpose CMOS process. Chip micrographs are shown in Fig. 9: on the left prototype 1 which equips a single library; on the right prototype 2 which uses the dual library strategy. The larger device length used in prototype 2 does not impact core area (both 0.16mm^2). All measurements come from multiple dies, measured at 20°C and are realized while running the Dhrystone benchmark C-code, compiled with the commercial ARM tool chain. Speed performance as a function of supply voltage is shown in Fig. 10 for both prototypes. Speed difference is minimal and solely due to a difference in process corner of both MPW runs, as synthesis speed objectives were identical. The minimal difference in speed shows how the multi library optimization is able to achieve identical speed, while still using slower cells for a major part of the design (75.2%). Minimal supply voltage for correct operation has increased from 190mV to 200mV, as was expected. Although this increase is small, it is non-negligible considering the applied supply voltages. Fig. 11 shows energy/cycle for the M0 core of both designs. The dual library optimization clearly had a tremendous impact, reducing core energy consumption by 50% or

more across a wide voltage range. The contribution of static energy to the total energy is shown in Fig. 12: as expected from the analysis in section V, static energy contribution is significantly lower. Because dynamic energy has remained more or less identical, the MEP has moved to a lower supply voltage, frequency and energy/cycle. While the original MEP was at 440mV, 31MHz and 16.07pJ/cycle for the single library design, the MEP has now moved to 370mV, operating at 13.7MHz for a mere 8.80pJ/cycle. As both energy curves are quite flat, a wide performance range is possible for energy consumption differing little from that from the MEP, breaking the sub-10pJ barrier for a 6-35MHz range. Although the second prototype has a lower speed at the MEP, it outperforms the first prototype across the entire voltage and performance range. This is shown more clearly in Fig. 13: a better energy-delay product (EDP) is achieved for all V_{dd} signifying the design performs better under all conditions. Performance of a single die of prototype 2 under a wide temperature range is shown in Fig. 14. Energy consumption increases with increasing temperature. Voltage supply for constant speed performance only varies about 5%, as shown by the 20MHz line. The MEP line shows an equally small variation in supply voltage for the MEP. However, the MEP energy consumption does increase with almost 40% when temperature is increased to 70°C.

Both designs are compared with state-of-the-art 32bit microcontrollers in Table I. Referenced designs equip either conventional static CMOS logic [5], [7] or another logic library [6]. This allows to evaluate the speed and energy performance of the TGL against state-of-the-art conventional static CMOS design. As Table I shows, nor the differential signaling, nor the increased gate length for low leakage cells compromises area. Prototype 2 combines the MCU with a 64KB SRAM memory, which is large enough for most applications. As expected, the applied transmission gate logic results in a significantly higher speed than all other compared designs. This is a significant benefit over both conventional static CMOS designs and most other logic families, even when considering technology scaling. The single-library design (prototype 1) combines this speed with a low energy consumption, resulting in a very competitive speed-energy combination. The dual library design (prototype 2) improves this even further, resulting in an higher speed and better or comparable core energy consumption than the referenced designs. The smaller SRAM in prototype 2 significantly reduces total energy consumption compared to prototype 1, resulting in a complete low power architecture. However, it is still far larger than the referenced designs, adding significantly to total energy consumption. Additionally, the proposed

Cortex-M0 architecture is less energy efficient than the referenced Cortex-M0+ architecture (energy consumption increases with about 33% [20]). Despite the less efficient architecture and far larger SRAM of the proposed prototypes, the proposed work achieves state-of-the-art energy consumption, especially when the high operating speed is considered, comparing with either conventional static CMOS designs or other logic. Although measurements shown at the minimum supply voltage confirm the sub-optimal operation in this region, they act as a useful parameter for variation resilient operation. The speed-energy combination achieved for both designs is depicted in the EDP and is the best metric to compare both speed and energy performance. Prototype 2 outperforms all others by a factor 5 or more without considering the difference in memory size.

VII. CONCLUSION

This work presents a differential transmission gate standard cell design flow for ultra-low power applications. Commercial tools are used with carefully engineered add-ons to enable full standard cell design using fast, variation resilient transmission gate logic cells. The flow provides full support for the designer, including multiple power domains, multi-mode multi-corner and leakage optimization. To showcase the tool flow, two generations of prototypes of an ARM Cortex-M0 microcontroller architecture were designed, implemented and measured in a 40nm CMOS process. The tool flow succeeds in realizing both prototypes and provides enough control to make the correct design trade-offs for minimum energy optimization. Both prototypes achieve a state-of-the-art speed-energy combination, combined with low variability. Sub-10pJ/cycle operation was realized across a 6-35MHz range. The designs outperform similar architectures designed in conventional static CMOS logic by far in energy-delay product. The short design time, complexity and level of control that can be achieved using the standard cell flow is combined with performance comparable to full custom work in this state-of-the-art microcontroller, showing the potential of the proposed design flow approach.

ACKNOWLEDGMENT

This work was funded by the IWT-agency for Innovation by Science and Technology.

REFERENCES

- [1] H. Kim, S. Kim, N. V. Helleputte, A. Artes, M. Konijnenburg, J. Huisken, C. V. Hoof, and R. F. Yazicioglu, "A configurable and low-power mixed signal soc for portable ecg monitoring applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 8, no. 2, pp. 257–267, April 2014.
- [2] M. Ashouei, J. Hulsink, M. Konijnenburg, J. Zhou, F. Duarte, A. Breeschoten, J. Huisken, J. Stuyt, H. de Groot, F. Barat, J. David, and J. V. Ginderdeuren, "A voltage-scalable biomedical signal processor running ecg using 13pj/cycle at 1mhz and 0.4v," in *2011 IEEE International Solid-State Circuits Conference*, Feb 2011, pp. 332–334.
- [3] V. Sharma, S. Cosemans, M. Ashouie, J. Huisken, F. Catthoor, and W. Dehaene, "Ultra low-energy sram design for smart ubiquitous sensors," *IEEE Micro*, vol. 32, no. 5, pp. 10–24, Sept 2012.
- [4] A. Wang, A. P. Chandrakasan, and S. V. Kosonocky, "Optimal supply and threshold scaling for subthreshold cmos circuits," in *VLSI, 2002. Proceedings. IEEE Computer Society Annual Symposium on*, 2002, pp. 5–9.
- [5] J. Myers, A. Savanth, D. Howard, R. Gaddh, P. Prabhat, and D. Flynn, "An 80nm retention 11.7pj/cycle active subthreshold arm cortex-m0+ subsystem in 65nm cmos for wsn applications," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2015 IEEE International*, Feb 2015, pp. 144–145.
- [6] W. Lim, I. Lee, D. Sylvester, and D. Blaauw, "Batteryless sub-nw cortex-m0+ processor with dynamic leakage-suppression logic," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2015 IEEE International*, Feb 2015, pp. 146–147.
- [7] S. Luetkemeier, T. Jungeblut, M. Porrmann, and U. Rueckert, "A 200mv 32b subthreshold processor with adaptive supply voltage control," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, Feb 2012, pp. 484–486.
- [8] N. Reynders and W. Dehaene, "Variation-Resilient Building Blocks for Ultra-Low-Energy Sub-Threshold Design," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 59, no. 12, pp. 898–902, Dec 2012.
- [9] —, "A 210mv 5mhz variation-resilient near-threshold jpeg encoder in 40nm cmos," in *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2014 IEEE International*, Feb 2014, pp. 456–457.
- [10] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consumption in digital cmos circuits," *Proceedings of the IEEE*, vol. 83, no. 4, pp. 498–523, Apr 1995.
- [11] K. Yano, T. Yamanaka, T. Nishida, M. Saito, K. Shimohigashi, and A. Shimizu, "A 3.8-ns cmos 16 times;16-b multiplier using complementary pass-transistor logic," *IEEE Journal of Solid-State Circuits*, vol. 25, no. 2, pp. 388–395, Apr 1990.
- [12] L. P. Alarcn, T.-T. Liu, M. D. Pierson, and J. M. Rabaey, "Exploring very low-energy logic: A case study," *Journal of Low Power Electronics*, vol. 3, no. 3, pp. 223–233, 2007.
- [13] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits: A Design Perspective*. Prentice Hall, 2003, p.188-193.
- [14] A. Wang and A. Chandrakasan, "A 180mv fft processor using subthreshold circuit techniques," in *Solid-State Circuits Conference, 2004. Digest of Technical Papers. ISSCC. 2004 IEEE International*, Feb 2004, pp. 292–529 Vol.1.
- [15] B. H. Calhoun and A. P. Chandrakasan, "Ultra-dynamic voltage scaling (udvs) using sub-threshold operation and local voltage dithering," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 1, pp. 238–245, Jan 2006.
- [16] H. R. Harris, P. Kalra, P. Majhi, M. Hussain, D. Kelly, J. Oh, D. He, C. Smith, J. Barnett, P. D. Kirsch, G. Gebara, J. Jur, D. Lichtenwalner, A. Lubow, T. P. Ma, G. Sung, S. Thompson, B. H. Lee, H. H. Tseng, and R. Jammy, "Band-engineered low pmos vt with high-k/metal gates featured in a dual channel cmos integration scheme," in *2007 IEEE Symposium on VLSI Technology*, June 2007, pp. 154–155.

- [17] 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.
- [18] M. Seok, D. Blaauw, and D. Sylvester, "Clock network design for ultra-low power applications," in *Low-Power Electronics and Design (ISLPED), 2010 ACM/IEEE International Symposium on*, Aug 2010, pp. 271–276.
- [19] H. Reyserhove and W. Dehaene, "A 16.07pj/cycle 31mhz fully differential transmission gate logic arm cortex m0 core in 40nm cmos," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, Sept 2016, pp. 257–260.
- [20] "ARM Cortex-M Series Processors Overview," <https://www.arm.com/products/processors/cortex-m>, accessed: 2016-12-01.

LIST OF FIGURES

1	Energy consumption in a digital system, divided in static and dynamic energy. Analysis before and after activity reduction (1 vs 2) results in a lower MEP.	19
2	Overview of the building blocks of the differential transmission gate standard cell library.	20
3	Left: Propagation delay variability analysis of stacked inverter and TG vs. conventional static CMOS logic sized for either equal speed, equal area or equal noise margin (1000MC). Right: Normalized static noise margin across process corners at 300mV for stacked inverter and TG vs. conventional static CMOS logic.	21
4	Transmission gate building block allowing every logic function and easy layout. Combinations of multiple transmission gates are joined with inverters to form logic cells.	21
5	Proposed digital design flow with ultra-low voltage differential transmission gate standard cells.	22
6	Block diagram of the ARM Cortex-M0 system.	23
7	Leakage - delay trade-off for device length and V_T at 300mV.	23
8	Static vs. dynamic energy trade-off comparison for prototype 1 vs. prototype 2. Static energy reduction by dual library leakage optimization results in a lower MEP at a lower supply voltage.	24
9	Chip micrograph of both generations, implemented in 40nm CMOS. Left: prototype 1 which equips a single library. Right: prototype 2 which improves static energy by employing two libraries.	24
10	Mean of the measured maximum operation frequency as function Vdd for both designs.	25
11	Boxplot of the M0 core measured energy/cycle as a function of Vdd for both designs.	26
12	Leakage contribution to M0 core energy consumption as a function of Vdd for both designs.	27
13	Energy-delay product of the M0 core as a function of Vdd for both designs.	28
14	Energy consumption of the M0 core as a function of Vdd for a wide temperature range.	28

Fig. 1

Fig. 2

Fig. 3

Fig. 4

Fig. 5

Fig. 6

Fig. 7

Fig. 8

Fig. 9

Fig. 10

Fig. 11

Fig. 12

Fig. 13

Fig. 14

LIST OF TABLES

I	Performance summary and state-of-the-art comparison.	30
---	--	----

TABLE I

	This work [19] Prototype 1	This work Prototype 2	[5] ISSCC'15	[6] ISSCC'15	[7] ISSCC'12
Technology	40nm CMOS	40nm CMOS	65nm CMOS	180nm CMOS	65nm CMOS
Core	ARM Cortex M0	ARM Cortex M0	ARM Cortex M0+	ARM Cortex M0+	32b RISC
Die Area (core)	0.16mm²	0.16mm²	0.16mm ²	1.1mm ²	0.36mm ²
Die Area (total)	2.55mm²	1.80mm²	3.76mm ²	2.04mm ²	2.7mm ²
Memory	256KB SRAM @ 0.6V	64KB SRAM @ 0.6V	2KB ROM, 8KB ULV SRAM, 16KB SRAM	256B	0.6KB ULV SRAM, 32KB SRAM
# dies measured	25	10	160	28	37
V _{dd,MEP}	0.44V	0.37V	0.35V	0.55V	0.325V
V _{dd,min}	0.19V	0.20V	0.25V	0.16V	0.20V
Speed @ MEP	31.2MHz	13.7MHz	750kHz	7Hz	133kHz
Speed @ V _{dd,min}	0.8MHz	0.8MHz	27kHz	16Hz	10kHz
Core E/cycle @ MEP	16.07pJ	8.80pJ	/	/	9.94pJ
Core E/cycle @ V _{dd,min}	60.55pJ	21.09pJ	/	/	20pJ*
Total E/cycle @ MEP	100.34pJ	43.22pJ	11.7pJ	92.04pJ	/
Total E/cycle @ V _{dd,min}	2551pJ	179.01pJ	32pJ	2275pJ*	/
	CPU, Periph, 256KB	CPU, Periph, 64KB	CPU, 4KB SRAM	CPU, 256B	
EDP [pJ/MHz]	0.5 (core) 81.2 (total)	0.6 (core) 3.2 (total)	/	/	74.4 (core)
			15.6 (total)	10 ⁷ (total)	/

* = estimated