A Sub-Threshold FPGA with Low-Swing Dual-V_{DD} Interconnect in 90nm CMOS

Joseph F. Ryan and Benton H. Calhoun

University of Virginia, Charlottesville, VA 22904, <jfr7p, bcalhoun>@virginia.edu

***Abstract*- This paper presents a sub-threshold Field Programmable Gate Array (FPGA) that uses a low-swing dual-V_{DD} global interconnect fabric to reduce energy and improve delay. A 90nm chip implements the FPGA with 1134 LUTs, which is 2.7X smaller, 14X faster, and 4.7X less energy than a sub-threshold FPGA using conventional interconnect and 22X less energy than an equivalent FPGA at full V_{DD}.**

I. INTRODUCTION

Ultra low power (ULP) miniature devices are enabling a new generation of applications for areas such as healthcare and wireless environmental control. Development of these applications is costly, however, as managing strict resource constraints can lead to high design complexity. For example, many require a significant amount of on-chip processing to extract information locally and thereby reduce the use of power hungry components like radios [1]. The large variety of ULP projects and frequent changes to their requirements make flexibility a desirable attribute in this design space.

Existing approaches fail to provide both flexibility and desired energy efficiency. Commercial processors and FPGAs are far too inefficient for the ULP space, but alternatives using sub-threshold (sub-V_T) operation have emerged as an energy efficient alternative. Sub-V_T microprocessors have shown very low energy per instruction [1][2], but their simple ISAs require many 1000s of instructions to perform significant computing. Sub-V_T ASICs give amazing efficiency (e.g. [3]) but are inflexible and thus expensive to produce for low-volume projects. We propose re-targeting FPGAs, which commercially compete mostly in the high performance space, to ULP for applications to take advantage of the balanced tradeoff they provide between hardware efficiency and flexibility. To do so, we design a sub-V_T FPGA.

Fig. 1 shows a typical look up table (LUT) based FPGA architecture. A configurable logic block (CLB) contains a cluster of 4 basic logic elements (BLEs), each holding a 4-input (16:1) LUT, mux, and flipflop (FF). SRAM bits (Cbits) configure the connectivity of the CLBs and the interconnect fabric between them to form *paths*. Each interconnect *net* connects CLBs using one or more wire *segments* and crossbar switch boxes (SBs). We have previously shown in simulation that mux-based FPGA structures using static CMOS circuits scale to sub-V_T voltages without significant functionality problems; however, tristate buffers are required in every switch box to combat variation [4]. We previously modified a traditional FPGA design (e.g. Fig. 1) to function in sub-V_T [4], and we use that design as a base case (BC) for comparison purposes. The global interconnect in the BC design dominates its energy and delay. In this paper, we present a new FPGA design that improves area, energy, and delay in sub-V_T.

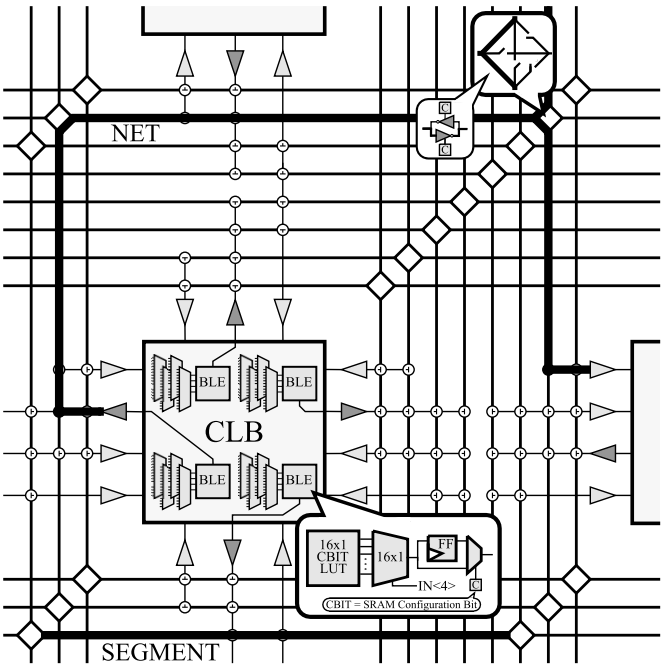

Fig. 1: Standard FPGA Tile / Base Case (BC) FPGA Tile

II. SUB-VT FPGA WITH LOW-SWING GLOBAL INTERCONNECT

The global interconnect fabric dominates the energy and delay of commercial FPGAs and also thus the base case sub-V_T FPGA. Almost 84% of its delay and 70% of its energy is consumed in the interconnect at 0.4 V [4]. Our new "custom case" (CC) sub-V_T FPGA design uses an architecture with a higher degree of clustering and a novel high-density, low-swing, dual-V_{DD} global interconnect fabric to improve area, energy, and delay.

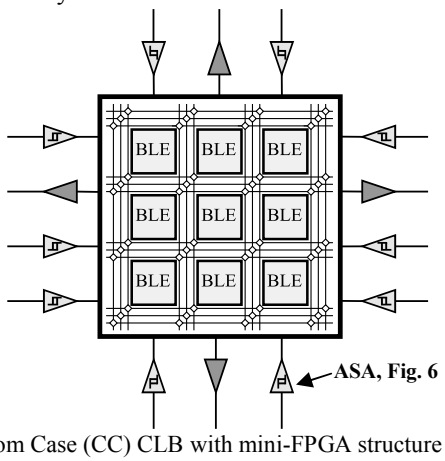

Fig. 2: Custom Case (CC) CLB with mini-FPGA structure for local (intra-CLB) interconnect.

A. Mini-FPGA for clustering inside the CLB

Since the global interconnect is costly in energy and delay, we can increase the utilization of local routing available inside clustered CLBs to reduce global signal traffic. Clustering more BLEs in each CLB requires extra area for local (e.g. intra-CLB) routing since every combination of BLE inputs, including shared CLB inputs and local BLE outputs, must be routable. Bigger CLBs also stretch global segments' physical length. Since more clustering increases local connections for any given configuration, it reduces the number of hops required on the global fabric, which is helpful to our low swing circuit approach (Section II.B). We cluster 9 BLEs per CLB and use a "mini-FPGA" configurable routing fabric in the CC CLB. The mini-FPGA uses transmission gate SBs for full swing signaling (Fig. 2) to provide the same intra-CLB connectivity as the BC muxes but with less area. BLEs have full connectivity with 6 local channels on the mini-FPGA. We designed a custom toolflow (Fig. 3), leveraging publicly available tools, e.g. [5], which allows us to program arbitrary configurations onto the new architecture. Over a suite of 43 benchmarks, the CC architecture lowers the burden on the global fabric due to a reduction of the number of CLBs on the critical path. The average CLBs/path drops from 7.6 to 6, and its σ shrinks from 2.9 to 2.1 across the benchmarks.

B. Low-swing global interconnect

To reduce energy and delay on the global interconnect, we design a low-swing signaling scheme. The bi-directional tristate drivers in the SBs (Fig. 1) are replaced with single passgate switches, reducing the number of FETs per SB by 3X and the average per-segment total load capacitance by 40%. The CC FPGA's passgate interconnect has very different transfer characteristics than the BC's buffers. The signal is marked first by a steep initial transition (~0.1V for 0.4V V_{DD}, e.g. Fig. 10) and then followed by a very slow tail that does not swing to full V_{DD}. The reduced swing helps to lower switching energy in the highly capacitive interconnect, but it also presents a functionality concern. In sub-V_T, the reduced output swing in a passgate network depends on the node's placement in the net (nodes further in the chain suffer a larger drop) and on the *total* parasitic leakage from every node along the entire net, including those on branching paths. We developed a set of custom tools that can quickly compute the delay and energy of an entire placed-and-routed Verilog design (e.g. Fig. 3) for both the CC and BC designs using a model of the FPGA fabric, similar to the models used for SRAM array design. The tool uses an analytical expression for the transfer characteristics of a passgate in sub-V_T [6], which we apply to a model of the regular structure of the FPGA fabric to compute the swing, delay, and energy of an arbitrary net. The tool also allows full SPICE simulations, using Monte-Carlo to account for process variations, to be substituted for the analytic model for higher accuracy results. The model gives estimates of the energy and delay in minutes compared with multiple-day SPICE simulations with only a few % error in energy and 10%-20% error in delay. Simulations show that even very deep nets (>50 segments) retain an output swing that is detectable in the initial steep transition region by a sense amp (SA). Our architecture with higher clustering helps reduce the maximum length of nets, limiting the extent of the voltage droop. However, we still need to detect this lower voltage with a SA.

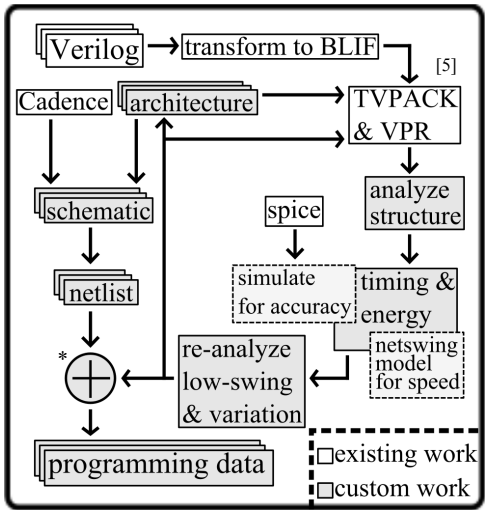

Fig. 3: Toolflow, including both custom and existing tools.

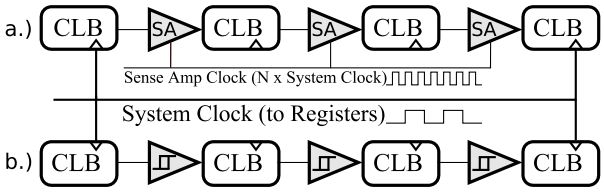

Fig. 4: (a) Synchronous vs. (b) asynchronous sense amps (SAs) for a low swing interconnect scheme. Synch timing forces high overhead.

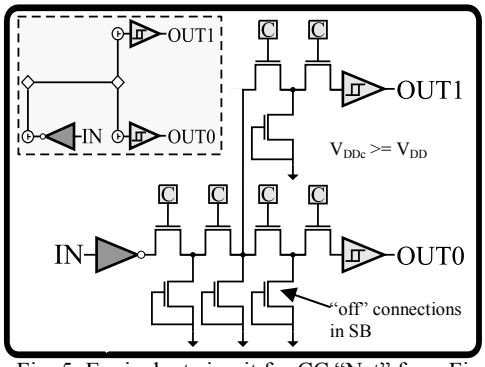

Fig. 5: Equivalent circuit for CC "Net" from Fig. 1.

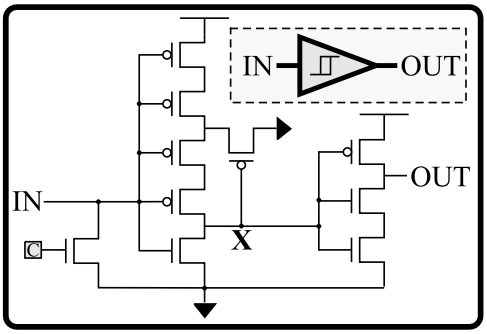

Fig. 6: Schematic of asynchronous sense amp (ASA).

Fig. 4 shows how a logical path on an FPGA can originate at one CLB FF (or pad) and pass across many nets and through many other CLBs before terminating at another FF (or pad). The system clock is thus set by the length of the longest path. Signals must take several hops on the global interconnect to span this path, which influences the design of the SA. Fig. 4(a) shows that a synchronous SA scheme essentially pipelines a path, requiring a SA clock with a higher frequency that is a fixed multiple of the system clock to drive SAs in each CLB. Even though the per-net delay for most nets along the path may be short, the longest CLB-to-CLB net on the path will set the period of the SA clock, so imbalanced net lengths lead to extra unnecessary delay by imposing slack on shorter nets. Using asynchronous SAs (ASAs) removes the need for a costly SA clock and allows signals to propagate along each path at their natural pace (Fig. 4(b)). Note that the FPGA remains synchronous with its original clock.

C. Asynchronous Sense Amp (ASA)

We have designed a single ended ASA for our low swing interconnect. Fig. 5 shows an equivalent circuit for a net, similar to the one highlighted in Fig. 1, using the ASA. A standard inverter drives the entire net, with the passgate switches inherently limiting the swing arriving at the ASAs. The driver can be upsized as needed, as there are few overall drivers compared with SB channel intersections. In contrast, each channel intersection in a SB in the BC design has 12 tristates, which must be small for area reasons. One major advantage of the CC structure (Fig. 5) is that it decouples V_G of the passgates from V_{DD}; the Cbit voltage (V_{DDc}) can be set higher than V_{DD} to increase the drive of the passgates without incurring an active energy penalty. The leakage penalty of raising V_{DDc} is minor, because we use a high-V_T 5T bitcell for Cbits in both the BC and CC. In sub-V_T, voltage is a stronger knob than size, so the dual-V_{DD} nature of our design gives a strong knob for increasing speed without affecting energy, as we show later.

Fig. 6 shows the schematic of our single ended ASA, which is based on a conventional Schmitt trigger (ST) but designed to trip earlier in a rising input transition. When unused, a Cbit places it into a low leakage mode. Fig. 7 shows a Monte-Carlo (M-C) sim of the ASA and ST trip points. The ASA is faster than the ST since it trips earlier in the slow transition on the interconnect. Fig. 8 shows an additional problem with a normal ST; any nets that settle to low swing voltages near $V_{DD}/2$ will incur large static current. The ASA on-current peak occurs at lower input values, so its static current is much smaller when the global nets settle to their final values. Fig. 9 shows the total leakage (including static current) for one representative benchmark (780 LUTs). The CC leakage is slightly higher (40%) at V_{DD}=0.4V for the active CLBs. Fig. 10 shows a M-C simulation of one net (10 segments deep, with branching loads). The 10 series tristates from the SBs in the BC design create large delay variation due to mismatch, so the BC output (BC out) is substantially spread in time. The ASA output (CC ASA out) triggers sharply early in the interconnect transition (CC ASA in) and with far less impact of variation.

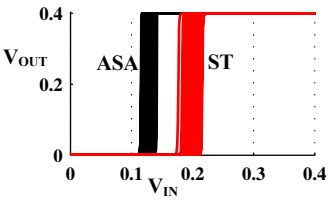

Fig. 7: VTC Monte-Carlo (M-C) sim of ASA and normal ST.

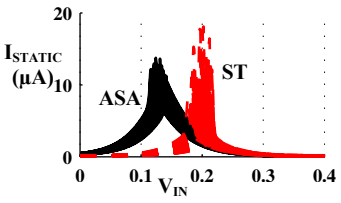

Fig. 8: Static current M-C sim of ASA and ST vs. input voltage.

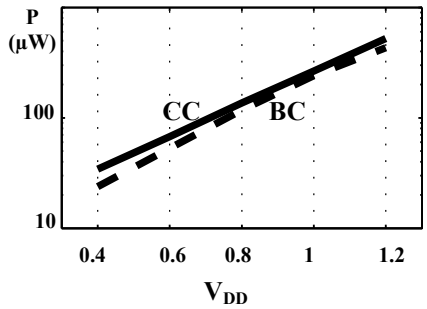

Fig. 9: Sim of benchmark leakage (780 LUTs). $V_{DDc} = V_{DD}+0.4V$.

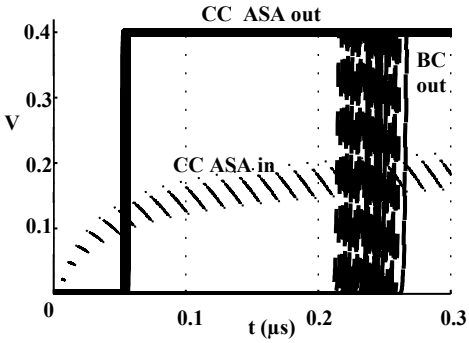

Fig. 10: Sim comparing CC and BC 10-length segment path. 100pt M-C. $V_{DDc} = V_{DD}+0.2V$ for CC.

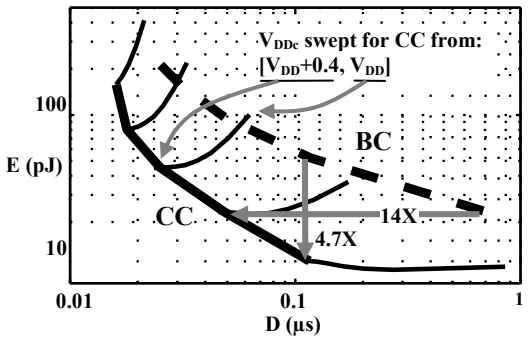

Fig. 11: Energy vs. delay sim for a representative benchmark (780 LUTs) with V_{DD} swept.

Fig. 11 compares the energy-delay (E-D) curves for the BC and CC FPGAs at V_{DD}=0.4V for a 780 LUT benchmark whose results are typical for the suite of 43 benchmarks. V_{DDc} for the CC varies from V_{DD} to V_{DD} + 0.4V, and the thin lines show how V_{DDc} affects the CC's E-D at constant V_{DD}. At $V_{DDc}=V_{DD}$, the performance of the CC is not much different than the BC. However, as V_{DDc} is raised, both delay and

energy are reduced. The energy savings are due to increased signal swing at the input of the ASA, which speeds up ASA operation and reduces static current. For example, a final value of approximately 0.2V at the CC ASA input translates into a ~5µA static/leakage draw of the ASA (Fig 8) as well as significant energy lost by static current during the slow transition. If V_{DDc} is raised by 0.2V, then the new final-time voltage increases to 0.32V, which reduces short-circuit and leakage power, especially if multiple ASAs connect to the same long net. V_{DDc} cannot be raised indefinitely, however, as eventually the swing for even the worst-case nets will hit V_{DD}.

The savings in sub-V_T are most significant. At constant energy, the CC is 14X faster than the BC in sub-V_T. At constant delay, the CC uses 4.7X less energy than the BC. The CC at V_{DD}=0.4V and V_{DDc}=0.8V uses 22X (20X average across all benchmarks) less energy than the BC at 1.2V with only a 5X delay penalty, which is much less than most sub-V_T designs. Fig. 12 shows that the CC gives these improvements by reducing the percentage of energy and delay in the interconnect at low voltage. The CLBs, which use full-swing logic, do not scale differently in proportion to each other.

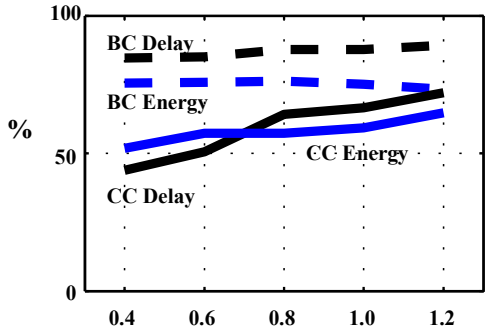

Fig. 12: % of delay and energy in the interconnect fabric for benchmark. V_{DDc} =V_{DD}+0.4 for the CC.

III. TEST CHIP AND MEASURED RESULTS

We implemented a CC FPGA array with 1134 LUTs in 90nm bulk CMOS (Fig. 13, die photo). The array of 6x21CLBs has routing tracks with 36 channels. For direct comparison on the same die, we also implemented a 912 LUT (12x19 CLB) BC array with 36-track channels. Per LUT, the CC is 2.7X less area due to the low swing passgate interconnect. The global interconnect uses 75% of the BC area and only 43% of the CC area. Measurements of the test chip confirm proper functionality of the CC including ASAs, mini-FPGA CLB, and BLEs down to V_{DD}=0.2V with raised V_{DDc}. Fig. 14 shows a Shmoo plot of the functional range of the CC FPGA across V_{DD} and V_{DDc}. Measured energy and delay curves for a fully-programmed benchmark design are pending completion of the automated bitstream mapping from vpr to the physical config bits (marked with * in Fig 3).

IV. CONCLUSION

Our sub-V_T FPGA uses a low swing, dual-V_{DD} interconnect scheme to reduce area per LUT (by 2.7X), delay at a constant energy (by 14X), and energy at a constant delay (by 4.7X) relative to a conventional design at low voltage. These improvements are made possible by a custom

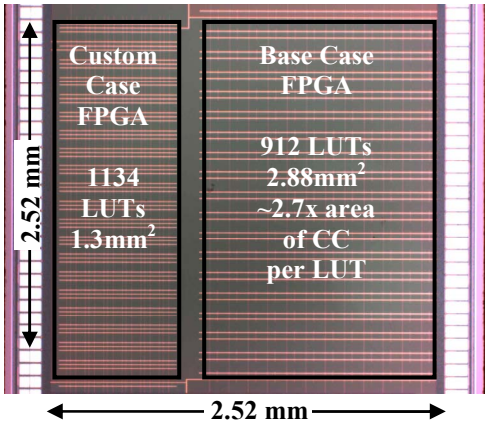

Figure 13: Annotated Die Photo

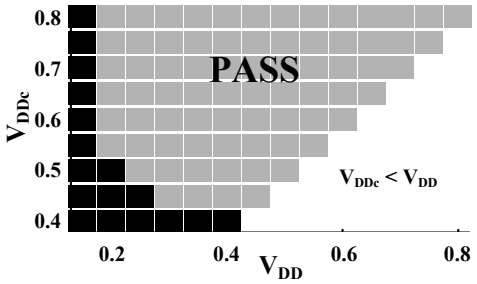

Fig. 14: Measured shmoo of CC FPGA across V_{DD} and V_{DDc}.

asynchronous sense amp and a separated Cbit voltage to optimize the delay in the routing with low energy overhead. Compared to a conventional design at a nominal voltage of 1.2V, the new sub-threshold FPGA consumes 22X less energy. This sub-threshold FPGA design enables energy efficient and cost effective configurable logic for a wide variety of ULP applications.

V. ACKNOWLEDGEMENTS

The authors thank DARPA (Young Faculty Award) for funding and E. Labadibi, S. Wooters, and K. Ringgenberg for helping with the chip.

VI. REFERENCES

[1] S. Jocke, J. Bolus, S. N. Wooters, A. D. Jurik, A. C. Weaver, T. N. Blalock, and B. H. Calhoun, "A 2.6-µW Sub-threshold Mixed-signal ECG SoC," Symposium on VLSI Circuits, 2009.

[2] B. Zhai, L. Nazhandali, J. Olson, A. Reeves, M. Minuth, R. Helfand, S. Pant, D. Blaauw, and T. Austin, "A 2.60pJ/Inst Subthreshold Sensor Processor for Optimal Energy Efficiency," Symposium on VLSI Circuits, 2006.

[3] Y. Pu, J.P. de Gyvez, H. Corporaal, and Y. Ha, "An Ultra-Low-Energy/Frame Multi-Standard JPEG Co-Processor in 65nm CMOS with Sub/Near-Threshold Power Supply," ISSCC, 2009.

[4] B. H. Calhoun, J. Ryan, S. Khanna, M. Putic, and J. Lach, "Flexible Circuits and Architectures for Ultra Low Power," Proc. of IEEE, Feb. 2010.

[5] V. Betz, J. Rose, and A. Marquardt, Architecture and CAD for Deep-Submicron FPGAs, Kluwer, 1999.

[6] J. F. Ryan and B. H. Calhoun, "Minimizing Offset for Latching Voltage-Mode Sense Amplifiers for Sub-threshold Operation," ISQED, 2008.