# Low Energy Asynchronous Architectures

*Ilya Obridko and Ran Ginosar*
VLSI Systems Research Center
Technion —Israel Institute of Technology, Haifa 32000, Israel
[oilya@tx.technion.ac.il]

**Abstract:** Asynchronous circuits are often presented as a means of achieving low power operation. We investigate their suitability for low-energy applications, where long battery life and delay tolerance is the principal design goal, and where performance is not a critical requirement. Three existing adder circuits are studied — two dynamic and one based on pass-transistor logic. All adders combine dual-rail and bundled-data circuits. The circuits are simulated at a wide supply-voltage range, down to their minimal operating point. Leakage energy (at 0.18μm) is found negligible. Transistor count is found to be an unreliable predictor of energy dissipation. A set of the energy minimization rules is defined and two novel adders are proposed, based on these rules – a dynamic circuit and a pass-transistor logic adder. The new adders consume less energy and achieve better performance, confirming the proposed concepts.

**Index terms**

Low energy, adder, asynchronous logic.

## 1 Introduction

Asynchronous logic has been promoted as a means of achieving low power design [1][2][6]. A number of advantages of asynchronous logic that make it appropriate for low power operation have been sited: Asynchronous circuits can stop computing when there is no new input, without the extra complexity of clock-gating logic and without the need to wait for clock restart delays. Power dissipation in large clock distribution trees is eliminated, though partly replaced by local handshake power [10]. If the circuit is speed-independent, supply voltage can be reduced when lower performance can be tolerated without having to retune clock frequencies [5]. More recently, asynchronous low energy (rather than low power) has been addressed [6][7], as this is more appropriate a design goal for extending battery life for mobile and other devices, as well as minimizing the efforts for heat dissipation and cooling expenses. Low power and low energy techniques for asynchronous systems are typically based on minimizing the number of transitions [1]. Other approaches include voltage scaling [5], early-open latch controllers, and data-dependent enabling of the logic [1][3][7].

We focus on simple computing circuits that must dissipate as little energy as possible in applications where performance is non-limiting and the time to complete any computing task is immaterial. A secondary goal is to be able to operate over a very wide range of supply voltage, as is typically the case with some battery-operated devices where voltage regulation is not desirable. The principal implication of a varying supply voltage is a wide range of delays, calling for the speed-independence feature of asynchronous circuits. The most robust speed-independent circuit methodology is based on dual-rail encoding and on quasi-delay-insensitive (qDI) design [1]. Unfortunately, qDI circuits are not necessarily the most energy efficient ones.

Four-phase qDI data signaling is based on alternating valid and null values. Each data bit must toggle from valid to null and back again on every successive data value, even if the data on both sides of the null have the exact same value. Two-phase qDI protocols help reduce delays but complicate and enlarge the logic and consequently increase energy consumption. Bundled data signaling (in both synchronous and asynchronous circuits) eliminates data switching when data values do not change. However, bundled data speed independent logic may not be as tolerant to wide delay variations as qDI circuits, since most bundled data schemes require matched delays and are exposed to the risk of not being long enough on one hand, while always incurring a worst-case delay on the other hand. Bundled data control paths consume additional energy.

Another low energy technique prefers large combinational blocks and minimizes the use of pipeline registers. Minimizing the number of blocks leads to eliminating some internal signals and buffers. Purely combinational logic could achieve minimum energy per computation, as long as redundant transitions are avoided.

As a basic test case we consider 32-bit adders, which are built either as regular ripple carry (using single-bit full adders) or larger blocks of two-bit adders. Adders are the basic blocks used in a wide range of applications. Minimizing energy dissipation in adders contributes to the energy savings in the entire system. Moreover, we can apply the same techniques to other blocks and even in high speed designs.

We investigate a hybrid bundled data / dual rail approach [1]. The dual-rail part provides completion indication, while the bundled data parts help minimize energy dissipation. As an example, we apply the design methodology to a large adder, and compare it with other published low energy adders [2][4][9]. The existing low energy adders are presented in Section 2. The proposed set of energy minimization rules and two implemented novel adders are described in Section 3, and energy dissipation and simulations results and conclusion are discussed in Section 4.

## 2 Low Energy Adder Architectures

In order to achieve high performance in wide adders, carry look-ahead circuits are usually employed. However, such circuits dissipate extra energy. In low-energy applications when performance is not an issue, no look-ahead circuits are used. Thus, we consider only ripple-carry adders. We also employ those hazard-free asynchronous techniques that block spurious transitions and perform their computations only after all inputs have arrived.

Another energy-related advantage of asynchronous ripple carry adders is their small and relatively simple completion-detection; in the circuits below, the carry-out of the last stage is considered as the indication of completion, and all sum outputs are assumed to be ready by the time the carry-out becomes valid (Figure 1).

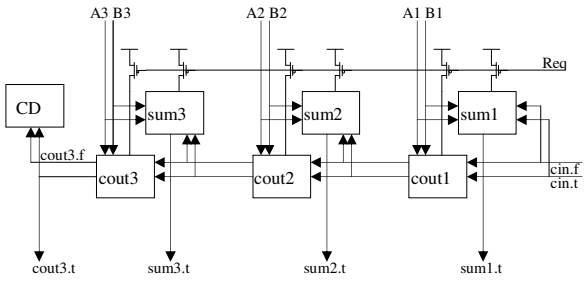


**Figure 1 Adder Based on Dynamic FA**

## 2.1 Dynamic (Nielsen) Full-Adder

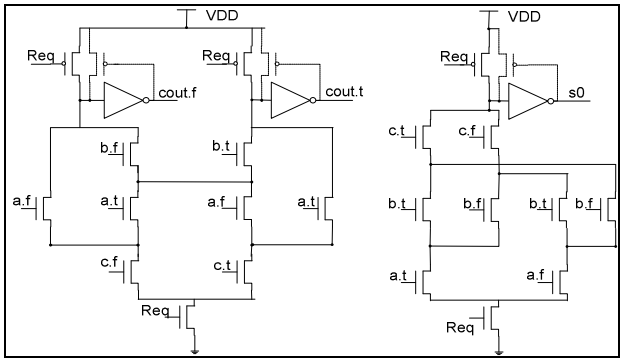


**Figure 2 Dynamic (Nielsen) FA Circuit**

The ripple-carry adder, used in the upper half of the Nielsen's adder [3], is modified by removing some logically redundant transistors that were employed for timing balance. The Nielsen's FA uses a single-rail sum, dual-rail inputs, and dual-rail carry-out. A complete adder using this dynamic FA is shown in Figure 1. Note that the adder is reset by *Req.* This allows quick execution of the return-to-zero part of the handshake. Likewise, when *Req* rises, all the stages in the chain start their calculations simultaneously. *Sum1,2,3* and *cout1,2,3* contain only NMOS pull down logic. The FA circuit is shown in Figure 2.

## 2.2 Dynamic Two-Bit Chong's Full-Adder

Chong *et al.* [4] introduce low-energy adders that also produce dual-rail carry-out and single-rail sum outputs. Two bits are combined and complex gates are employed to further minimize energy. The schematic description is depicted in Figure 3 and circuit implementation in Figure 4; note that completion detection depends only on the last carry-out.

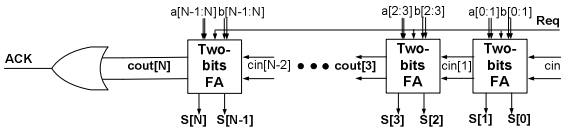


**Figure 3 Chong Adder**

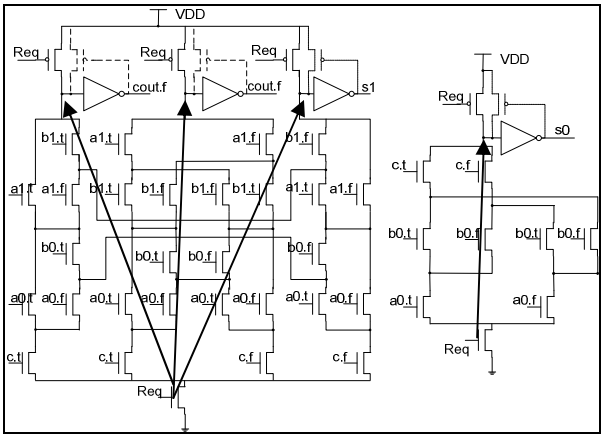


**Figure 4 Dynamic two-bit Chong's implementation and transistor reorder.**

The advantage of Chong's adder is a reduced number of transistors, thanks to using a complex gate. But transistor ordering is not optimal. *Req* signal triggers evaluation only after all input signals are ready. *Carry* inputs are calculated during the evaluation and arrive last. Eight transistors are connected to the precharged nodes, and any high inputs open additional transistors and further increase the load that must be precharged. If we "move" the *Req* transistors according to the four arrows in Figure 4, the capacitance charged during precharge is decreased, saving about 10% energy. *Req* transistors provide partial shielding between $V_{DD}$ and the rest of the pull-down logic (*Carry* provides additional better shielding – see Section 3.2). The delay and energy dissipation of the reordered adder is shown in Figure 9 and Figure 10 respectively.

## 2.3 Path Transistor Logic Adder

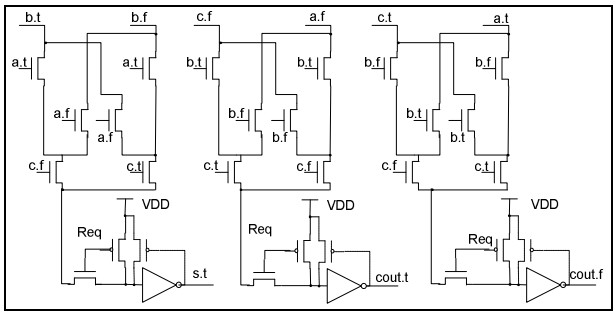


**Figure 5 PTL FA Circuit**

Single-ended pass transistor logic (SPL) and complementary pass transistor logic (CPL) are advocated for low energy arithmetic functions [8][9]. The main reason is that arithmetic functions are based on many XOR gates, and pass-transistor logic enables efficient implementations of XOR gates. CPL and SPL methods (named PTL below) contribute to energy minimization by the small number of pass transistors, which are usually NMOS, and produce very compact and regular designs.

Another advantage of PTL is that Vdd-to-GND paths, which may lead to short-circuit energy dissipation, are eliminated.

The drawback is the degradation of voltage swing to one $V_{TH}$ away from the supply. Voltage swing restoration buffers are required, increasing transistor count and energy dissipation. On the other hand, we charge the internal nodes to the lower voltage and thus reduce the charge amount and energy dissipation.

The original PTL FA of [9] has been appended with the Request-enabled output inverter and adapted to produce dual-rail carry-out to fit the asynchronous environment (**Error! Reference source not found.**).

# 3 Novel Low Energy Adder Circuits

We define a set of the rules for energy savings in asynchronous adders. First, hybrid dual- and single-rail circuits should be used. Dual-rail is used only where completion detection is necessary. Larger blocks of combinational logic save energy thanks to the buffer elimination and logic reduction. The latest transitioning signal has to be connected as close as possible to the precharged output node to reduce the number of nodes being charged.

Two new adders were designed using the above rules, as follows.

## 3.1 Novel Asynchronous Dual-Bit PTL Adder

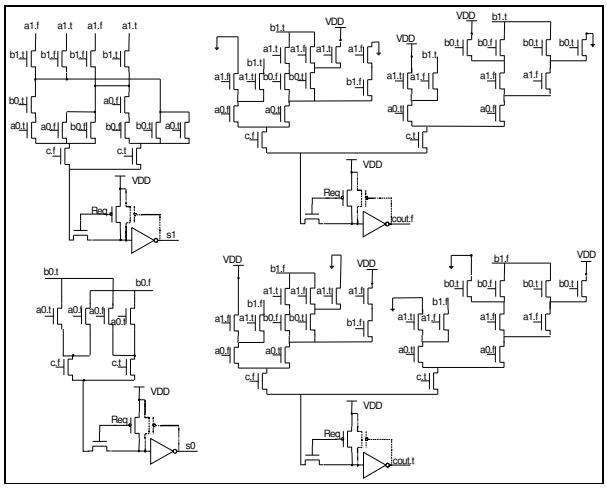


**Figure 6 Novel asynchronous dual-bit PTL cell.**

We designed a novel asynchronous dual-bit PTL adder (ADB-PTL), presented in Figure 6. *Req* NMOS transistor is combined with the keeper at the output, minimizing the precharged load. ADB-PTL adder cell calculates both *s0* and *s1*, eliminating internal carry calculations. Carry out provides completion indication. The ADB-PTL saves 15% energy relative to using single-bit PTL adders.

## 3.2 Novel Dual-Bit Dynamic Adder

A novel asynchronous dual-bit dynamic (ADBD) adder cell is shown in Figure 7. We take advantage of the fact that typically the latest arriving input signal is *Carry*, and insert the *Carry* transistors between the output and the rest of the pull-down nodes. This, as well as the NMOS *Req* transistor, minimize charge distribution in the internal nodes. In addition, the ADBD adder cell is dual-bit, employing a complex gate to produce *s0* and *s1*. The cell combines dual- and single-rail - only *Carry_out* provides completion indication.

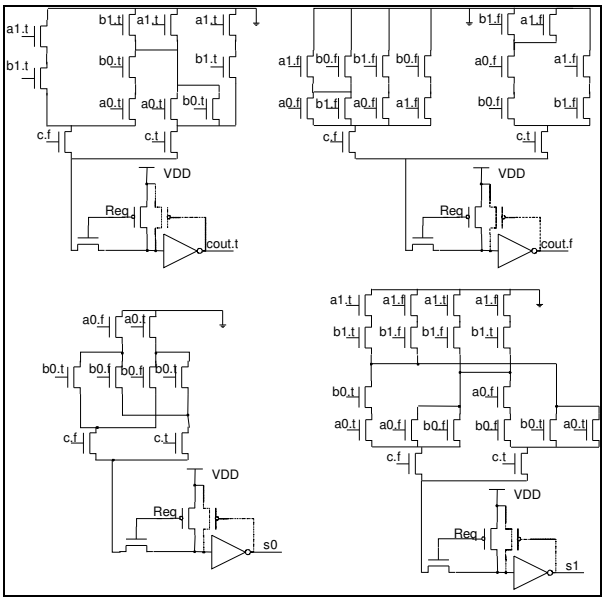


**Figure 7 Novel asynchronous dual-bit dynamic adder.**

# 4 Simulation Results and Conclusions

All the simulated adders were 32 bits wide, using as basic cell the adders described in Sections 2 and 3. All adder circuits were designed (at the schematic transistor level considering diffusions) for TSMC 0.18μm technology and simulated with Cadence Spectre. The simulated circuits included completion detection. All outputs were loaded by 10fF capacitors. Since voltage scaling serves as the principal means for energy reduction, all simulations were conducted by $V_{DD}$ sweeping over 0.7—1.5V (the nominal $V_{DD}$ for the technology is 1.8V). 100 random vectors were used for energy estimation of the adders at each voltage level. Energy and delay were measured over a full calculation cycle and averaged over the 100 random input vectors.

Figure 8 shows transistor counts for three 2-bit FA circuits. Based on Figure 8 and Figure 10 (energy), one can conclude that mere transistor count is not a sufficient predictor of energy dissipation. PTL FA requires the largest number of transistors (40% of them were employed in the Request-enabled output buffer that was required to make it “asynchronous”). Still, the PTL FA dissipates on average 14% less energy than Nielsen’s FA. Also, despite the fact that PTL FA requires 17% more transistors than Chong’s FA, it dissipates only about 6% more energy. Chong’s FA contains 8.5% fewer transistors but consumes 20% less energy than the Nielsen’s FA, thanks to producing only one carry-out signal. Nielsen’s FA calculates a carry-out signal per every bit, thus dissipating more energy.

The simulation results (Table 1) show that larger blocks of logic save 16-20% of energy thanks to the wire and transistor count reduction and thanks to fewer blocks. As a result, buffers are eliminated and consequentially the number of transitions is reduced. This can be seen in Figure 10, comparing Chong's and ADBD to Nielsen's adder, and ADB-PTL to single-bit PTL. These observations provide a strong incentive to design larger blocks of logic in order to gain maximal energy reduction.

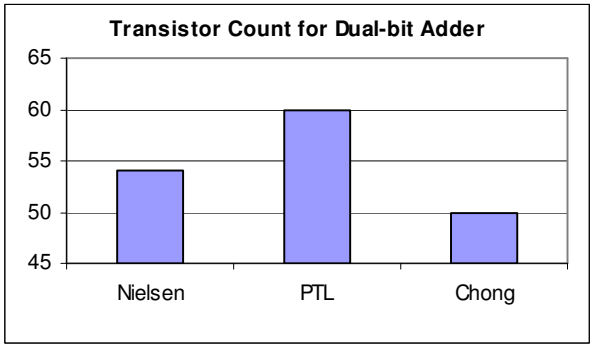


**Figure 8 Transistor Count Comparison**

Correct transistor ordering provides additional energy savings, thanks to decreasing the number of charged/discharged nodes. Thus, the reordered Chong adder consumes 10% less energy than the original one and the novel ABDB adder saves up to 20% energy, despite the fact that it was implemented using the same set of logic expressions [4].

Decreasing the supply voltage contributes about 20% energy savings per each 0.1V.

If we choose a certain supply voltage (higher than a minimal operating 0.7V for most of the adders), the two-bit basic adder cells in Chong and ADB_PTL reduce delay by 13% and 27% compared to their single-bit dynamic and PTL counterparts, thanks to fewer blocks and buffers. Optimal transistor ordering improves performance almost twofold, thanks to the fact that the amount of charge in the output nodes is smaller than in the original Chong's adder and charging and discharging are faster.

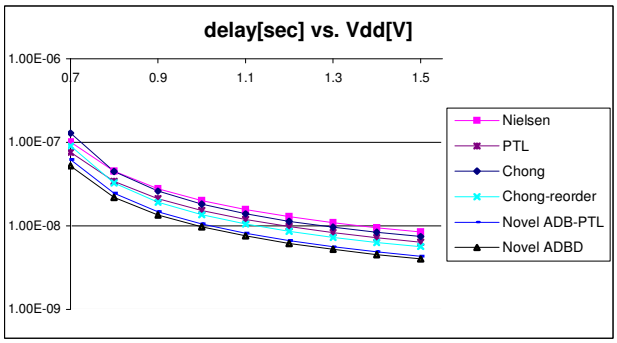


**Figure 9 Delay vs. Vdd Comparison**

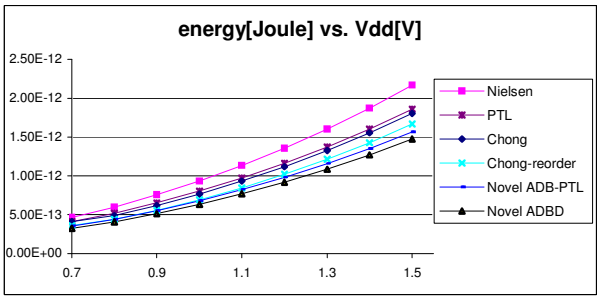


**Figure 10 Energy vs. Vdd Comparison**

| **Energy** | | | | | | |
|---|---|---|---|---|---|---|
| **Vdd [V]** | **Nielsen Δ%** | **PTL Δ%** | Chong | Chong-reorder Δ% | Novel ADB-PTL Δ% | Novel ADBD Δ% |
| 0.7 | 13.02% | -0.26% | 1 | -12.81% | -14.64% | -21.13% |
| 1.1 | 21.70% | 4.52% | 1 | -9.21% | -11.62% | -17.53% |
| 1.5 | 20.15% | 3.09% | 1 | -7.75% | -13.30% | -18.41% |
| **Delay** | | | | | | |
| 0.7 | -21.40% | -40.78% | 1 | -29.67% | -51.53% | -59.24% |
| 1.1 | 12.19% | -14.39% | 1 | -24.87% | -41.50% | -45.90% |
| 1.5 | 13.12% | -14.21% | 1 | -24.36% | -42.02% | -46.10% |

**Table 1. Adders comparison (Relative to Chong's)**

## Acknowledgment

We are grateful to Dana Amburg, Michael Moreinis, Yevgeny Perelman and Akadiy Morgenshtein, who have helped with ideas and CAD tools.

## References

[1] J. Sparso and S. Furber, *Principles of Asynchronous Circuit Design: A Systems Perspective*: Kluwer Academic Publishers, 2001.

[2] L.S. Nielsen and J. Sparsø, "A Low-power Asynchronous Data-path for a FIR Filter Bank," *Int. Symp. Adv. Res. Async. Circuits and Systems (ASYNC '96)*, pp. 18 - 21, 1996

[3] L.S. Nielsen, "Low-power Asynchronous VLSI Design," *Ph.D. Thesis, Department of Information Technology*, Technical University of Denmark, 1997.

[4] K.S. Chong, B.H. Gwee and J.S. Chang, "Low-voltage Asynchronous Adders for Low Power and High Speed Applications," *Int. Symp. Circuits and Systems (ISCAS)*, 2002.

[5] L. S. Nielsen, C. Niessen, J. Sparso, "Low-power operation using self-timed and adaptive scaling of the supply voltage," *IEEE Trans. VLSI Systems*, 2:391-397, 1994.

[6] A.J. Martin, "Remarks on low-power advantages of asynchronous circuits," *Europ. Solid-State Circuits Conf. (ESSCIRC)*, 1998.

[7] A.J. Martin, M. Nyström *et al.*, "The Lutonium: A Sub-Nanojoule Asynchronous 8051 Microcontroller," *IEEE Int. Symp. Async. Systems and Circuits*, May 2003.

[8] M. Munteanu, I. Bogdan *et al.*, "Single-Ended Pass Transistor Logic for Low-Power Design," *IEEE Asilomar Conf. Signals Systems and Computing*, pp. 364-368, 1999.

[9] D. Soudris, C. Piguet and C. Goutis, "Designing CMOS Circuits for Low-Power," *Kluwer Academic Publishers*, 2002.

[10] A.J. Martin, "An asynchronous approach to energy-efficient computing and communication," *SSGRR 2000*, August 2000.